MICROCHIP

microID® 125 kHz RFID System Design Guide

DATA SHEET MARKINGS

Microchip uses various data sheet markings to designate each document phase as it relates to the product development stage. The markings appear at the bottom of the data sheet, between the copyright and document and page numbers. The definitions for each marking are provided below for your use.

Marking	Description
Advance Information	The information is on products in the design phase. Your designs should not be finalized with this information as revised information will be published when the product becomes available.
Preliminary	This is preliminary information on new products in production but not yet fully characterized. The specifications in these data sheets are subject to change without notice. Before you finalize your design, please ensure that you have the most current revision of the data sheet by contacting your Microchip sales office.
No Marking	Information contained in the data sheet is on products in full production.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Table of Contents

PAGE

AN680 PASSIVE RFID BASICS

MCRF200 DATA SHEET

MCRF250 DATA SHEET

Table of Contents (Continued)

PAGE

TB023 CONTACT PROGRAMMING SUPPORT

AN710 ANTENNA CIRCUIT DESIGN FOR RFID APPLICATION

FSK READER REFERENCE DESIGN

PSK READER REFERENCE DESIGN

ASK READER REFERENCE DESIGN

Table of Contents

PAGE

Table of Contents

AN680

Passive RFID Basics

Author: Youbok Lee and Pete Sorrells
Microchip Technology Inc.

INTRODUCTION

Radio Frequency Identification (RFID) systems use radio frequency to identify, locate and track people, assets and animals. Passive RFID systems are composed of three components – a reader (interrogator), passive tag and host computer. The tag is composed of an antenna coil and a silicon chip that includes basic modulation circuitry and non-volatile memory. The tag is energized by a time-varying electromagnetic radio frequency (RF) wave that is transmitted by the reader. This RF signal is called a *carrier signal*. When the RF field passes through an antenna coil, there is an AC voltage generated across the coil. This voltage is rectified to result in a DC voltage for the device operation. The device becomes functional when the DC voltage reaches a certain level. The information stored in the device is transmitted back to the reader. This is often called backscattering. By detecting the backscattering signal, the information stored in the device can be fully identified.

There are two classes of RFID device depending on type of memory cell : (a) read only device and (b) read and write device. The memory cell can be made of EEPROM or FRAM. EEPROM is based on CMOS silicon and FRAM is based on ferroelectric memory. Since CMOS process technology has been matured, the EEPROM can be produced relatively at lower cost than the FRAM device. However, FRAM based RFID device consumes less power which is desirable for low power device. Therefore, it is known as a good candidate for the future RFID device, if its manufacturing cost becomes compatible to that of the CMOS technology.

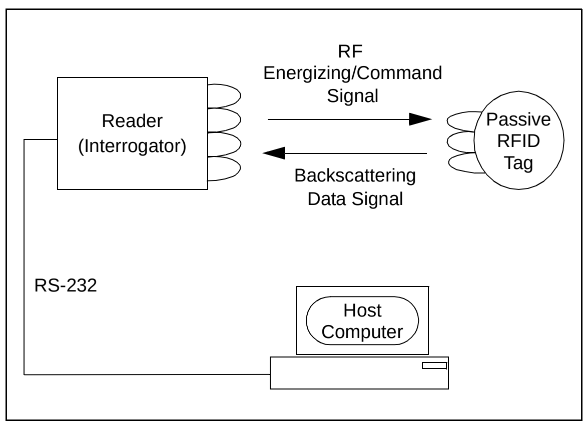
Because of its simplicity for use, the passive RFID system has been used for many years in various RF remote sensing applications. Specifically in access control and animal tracking applications.

In recent years, there have been dramatic increases in application demands. In most cases, each applications uses a unique packaging form factor, communication protocol, frequency, etc. Because the passive tag is remotely powered by reader's RF signal, it deals with very small power (~ µw). Thus, the read range (communication distance between reader and tag) is typically limited within a proximity distance. The read range varies with design parameters such as frequency, RF power level, reader's receiving sensitivity, size of antenna, data rate, communication protocol, current consumptions of the silicon device, etc.

Low frequency bands (125 kHz-400 kHz) were traditionally used in RFID applications. This was because of the availability of silicon devices. Typical carrier frequency (reader's transmitting frequency) in today's applications range from 125 kHz-2.4 GHz.

In recent years, the applications with high frequency (4-20 MHz) and microwave (2.45 GHz) bands have risen with the advent of new silicon devices. Each frequency band has advantages and disadvantages. The 4-20 MHz frequency bands offer the advantages of low (125 kHz) frequency and microwave (2.4 GHz) bands. Therefore, this frequency band becomes the most dominant frequency band in passive RFID applications.

FIGURE 1: SIMPLE CONFIGURATION OF RFID SYSTEMS



DEFINITIONS

READER, INTERROGATOR

RFID reader is used to activate passive tag with RF energy and to extract information from the tag.

For this function, the reader includes RF transmission, receiving and data decoding sections. In addition, the reader includes a serial communication (RS-232) capability to communicate with the host computer. Depending on the complexity and purpose of applications, the reader's price range can vary from ten dollars to a few thousand dollar worth of components and packaging.

The RF transmission section includes an RF carrier generator, antenna and a tuning circuit. The antenna and its tuning circuit must be properly designed and tuned for the best performance. See Application Note AN710 (DS00710) for the antenna circuit design.

Data decoding for the received signal is accomplished using a microcontroller. The firmware algorithm in the microcontroller is written in such a way to transmit the RF signal, decode the incoming data and communicate with the host computer.

Typicall, reader is a read only device, while the reader for read and write device is often called interrogator. Unlike the reader for read only device, the interrogator uses command pulses to communicate with tag for reading and writing data.

TAG

Tag consists of a silicon device and antenna circuit.

The purpose of the antenna circuit is to induce an energizing signal and to send a modulated RF signal. The read range of tag largely depends upon the antenna circuit and size.

The antenna circuit of tag is made of LC resonant circuit or E-field dipole antenna, depending on the carrier frequency. The LC resonant circuit is used for the frequency of less than 100 MHz. In this frequency band, the communication between the reader and tag takes place with magnetic coupling between the two antennas through the magnetic field. The antenna utilizing the inductive coupling is often called magnetic dipole antenna.

The antenna circuit must be designed such a way to maximize the magnetic coupling between them. This can be achieved with the following parameters:

a) LC circuit must be tuned to the carrier frequency of the reader
b) Maximize Q of the tuned circuit
c) Maximize antenna size within physical limit of application requirement.

See Application Note AN710 for more details.

When the frequency goes above 100 MHz, the requirement of LC values for its resonant frequency becomes too small to realize with discrete L and C components. As frequency increases, the wavelength is getting shorter. In this case, a true E-field antenna can be made of a simple conductor that has linear dimension less than or equivalent to half (½) the wavelength of the signal. The antenna that is made of a simple conductor is called electric dipole antenna. The electric dipole antenna utilizes surface current that is generated by an electric field (E-Field). The surface current on the conductor produces voltage at load. This voltage is used to energize the silicon device. Relatively simple antenna structure is formed for the higher frequency compared to the lower frequency.

READ-ONLY DEVICE, READ/WRITE DEVICE:

For the read only device, the information that is in the memory can't be changed by RF command once it has been written.

Read only devices are programmed as follows: (a) in the factory as a part of manufacturing process, (b) contactlessly programmed one time after the manufacturing (MCRF200 and MCRF250) or (c) can be programmed and also reprogrammed in contact mode (MCRF355 and MCRF360).

A device with memory cells that can be reprogrammed by RF commands is called read/write device. The information in the memory can be reprogrammed by Interrogator command.

Memory in today's RFID device is made of (a) CMOS or (b) FRAM array. The CMOS memory cell needs higher voltage for writing than reading. In the passive read/write device, the programming voltage is generated by multiplying the rectified voltage. The voltage multiplier circuit is often called a charge pumper. In addition to the programming voltage, the read/write device needs command decoder and other controller logics. As a result, the read/write device needs more circuit building blocks than that of the read only device. Therefore, the device size is larger and cost more than a read only device. The FRAM device needs the same voltage for reading and writing. However, its manufacturing cost is much higher than CMOS technology. Most of RFID device available today's market place are CMOS based device.

READ/WRITE RANGE

Read/write range is a communication distance between the reader (Interrogator) and tag. Specifically, the read range is a maximum distance to read data out from the tag and write range is a maximum distance to write data from interrogator to tag.

The read/write range is related to:

(1) Electromagnetic coupling of the reader (interrogator) and tag antennas, (2) RF Output power level of reader (interrogator), (3) Carrier frequency bands, (4) Power consumption of device, etc.

The electromagnetic coupling of reader and tag antennas increases using similar size of antenna with higher Q in both sides. The read range is improved by increasing the carrier frequency. This is due to the gain in the radiation efficiency of the antenna as the frequency increases. However, the disadvantage of high frequency (900 MHz-2.4 GHz) application are shallow skin depth and narrower antenna beam width. These cause less penetration and more directional problem, respectively. Low frequency application, on the other hand, has advantage in the penetration and directional, but a disadvantage in the antenna performance.

The read range increases by reducing the current consumptions in the silicon device. This is because additional radiating power is available by reducing the power dissipation in the silicon device.

MODULATION PROTOCOL

The passive RFID tag uses backscattering of the carrier frequency for sending data from the tag to reader. The amplitude of backscattering signal is modulated with modulation data of the tag device. The modulation data can be encoded in the form of ASK (NRZ or Manchester), FSK or PSK. Therefore, the modulation signal from the tag is Amplitude-Amplitude, Amplitude-FSK and Amplitude-PSK. See MicroID 125 kHz Design Guide for Amplitude, Amplitude-FSK and Amplitude-PSK reader.

CARRIER

Carrier is the transmitting radio frequency of reader (interrogator). This RF carrier provides energy to the tag device, and is used to detect modulation data from the tag using backscattering. In read/write device, the carrier is also used to deliver interrogator's command and data to the tag.

Typical passive RFID carrier frequencies are:

a) 125 kHz-400 kHz
b) 4 MHz-24 MHz
c) 900 MHz-2.45 GHz.

The frequency bands must be selected carefully for applications because each one has its own advantages and disadvantages. Table 1 shows the characteristic of each frequency bands.

TABLE 1:

Frequency Bands	Antenna Components	Read Range (typical)	Penertration (skin depth)	Orientation (Directionality)	Usability in metal or humid environment	Applications (typical)
Low Frequency (125 - 400) kHz	Coil (> 100 turns) and capacitor	Proximity (8")	Best	Least	Possible	Proximity
Medium Frequency (4 MHz - 24 MHz)	Coil (< 10 turns) and capacitor	Medium (15")	Good	Not much	Possible	Low cost and high volume
High Frequency (>900 MHz)	E-field dipole (a piece of conductor)	Long (> 1 m)	Poor	Very high	Difficult	Line of sight with long range

SYSTEM HANDSHAKE

Typical handshake of a tag and reader (interrogator) is as follows:

A. Read Only Tag

1. The reader continuously transmits an RF signal and watches always for modulated backscattering signal.
2. Once the tag has received sufficient energy to operate correctly, it begins clocking its data to a modulation transistor, which is connected across the antenna circuit.
3. The tag's modulation transistor shorts the antenna circuit, sequentially corresponding to the data which is being clocked out of the memory array.
4. Shorting and releasing the antenna circuit accordingly to the modulation data causes amplitude fluctuation of antenna voltage across the antenna circuit.
5. The reader detects the amplitude variation of the tag and uses a peak-detector to extract the modulation data.

B. Read and Write Tag

(Example: MCRF45X devices with FRR and Reader Talks First mode)

1. The interrogator sends a command to initiate communication with tags in the fields. This command signal is also used for energizing the passive device.
2. Once the tag has received sufficient energy and command, it responses back with its ID for acknowledgment.
3. The interrogator now knows which tag is in the field. The interrogator sends a command to the identified tag for what to do next: processing (read or write) or sleep.
4. If the tag receive processing and reading commands, it transmits a specified block data and waits for the next command.
5. If the tag receives processing and writing commands along with block data, it writes the block data into the specified memory block, and transmits the written block data for verification.
6. After the processing, the interrogator sends an "end" command to send the tag into the sleep ("silent") mode.
7. If the device receives "end" command after processing, it sends an acknowledgement (8-bit preamble) and stays in sleep mode. During the sleep mode, the device remains in non-modulating (detuned) condition as long as it remains in the power-up. This time the handshake is over.
8. The interrogator is now looking for the next tag for processing, establishes an handshake and repeats the processing.
9. See Fig.4-1 in MCRF45x Data sheet for more details.

BACKSCATTER MODULATION

This terminology refers to the communication method used by a passive RFID tag to send data to the reader using the same reader's carrier signal. The incoming RF carrier signal to the tag is transmitted back to the reader with tag's data.

The RF voltage induced in the tag's antenna is amplitude-modulated by the modulation signal (data) of tag device. This amplitude-modulation can be achieved by using a modulation transistor across the LC resonant circuit or partially across the resonant circuit.

The changes in the voltage amplitude of tag's antenna can affect on the voltage of the reader antenna. By monitoring the changes in the reader antenna voltage (due to the tag's modulation data), the data in the tag can be reconstructed.

The RF voltage link between reader and tag antennas are often compared to a weakly coupled transformer coils; as the secondary winding (tag coil) is momentarily shunted, the primary winding (reader coil) experiences a momentary voltage drop.

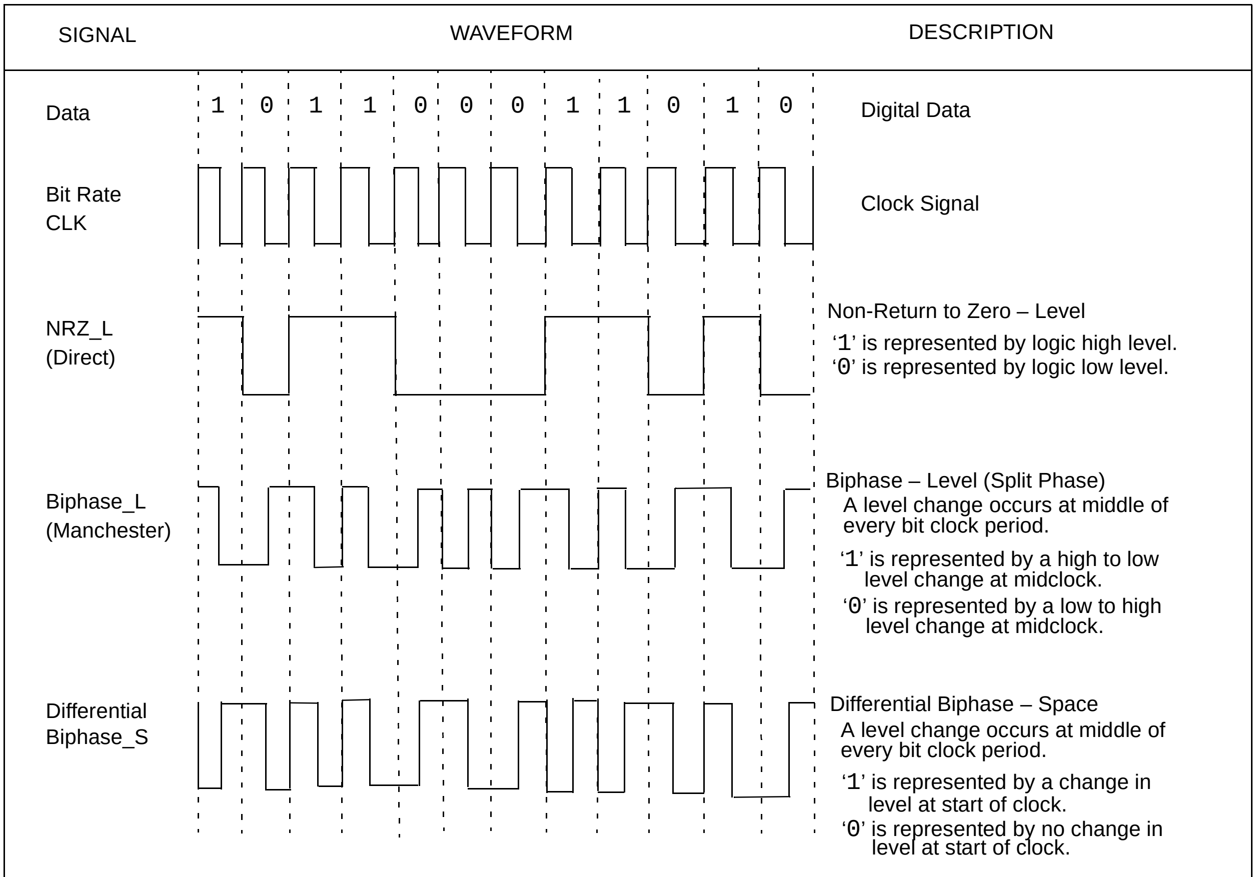
DATA ENCODING

Data encoding refers to processing or altering the data bitstream in-between the time it is retrieved from the RFID chip's data array and its transmission back to the reader. The various encoding algorithms affect error recovery, cost of implementation, bandwidth, synchronization capability and other aspects of the system design. Entire textbooks are written on the subject, but there are several popular methods used in RFID tagging today:

1. **NRZ (Non-Return to Zero) Direct.** In this method no data encoding is done at all; the `1`'s and `0`'s are clocked from the data array directly to the output transistor. A low in the peak-detected modulation is a '`0`' and a high is a '`1`'.
2. **Differential Biphase.** Several different forms of differential biphase are used, but in general the bitstream being clocked out of the data array is modified so that a transition always occurs on every clock edge, and `1`'s and `0`'s are distinguished by the transitions within the middle of the clock period. This method is used to embed clocking information to help synchronize the reader to the bitstream. Because it always has a transition at a clock edge, it inherently provides some error correction capability. Any clock edge that does not contain a transition in the data stream is in error and can be used to reconstruct the data.

3. **Biphase_L (Manchester).** This is a variation of biphase encoding in which there is not always a transition at the clock edge. **The MCRF355/360 and MCRF45X devices use this encoding method.**

FIGURE 2: VARIOUS DATA CODING WAVEFORMS



Note: Manchester coding is used for the MCRF355/360 and MCRF45X

DATA MODULATION FOR 125 kHz DEVICES (MCRF2XX)

Although all the data is transferred to the host by amplitude-modulating the carrier (backscatter modulation), the actual modulation of 1's and 0's is accomplished with three additional modulation methods:

1. Direct. In direct modulation, the Amplitude Modulation of the backscatter approach is the only modulation used. A high in the envelope is a '1' and a low is a '0'. Direct modulation can provide a high data rate but low noise immunity.
2. **FSK (Frequency Shift Keying).** This form of modulation uses two different frequencies for data transfer; the most common FSK mode is FC/8/10. In other words, a '0' is transmitted as an amplitude-modulated clock cycle with period corresponding to the carrier frequency divided by 8, and a '1' is transmitted as an amplitude-modulated clock cycle period corresponding to the carrier frequency divided by 10. The amplitude modulation of the carrier thus switches from FC/8 to FC/10 corresponding to 0's and 1's in the bitstream, and the reader has only to count cycles between the peak-detected clock edges to decode the data. FSK allows for a simple reader design, provides very strong noise immunity, but suffers from a lower data rate than some other forms of data modulation. In Figure 3, FSK data modulation is used with NRZ encoding.
3. **PSK (Phase Shift Keying).** This method of data modulation is similar to FSK, except only one frequency is used, and the shift between 1's and 0's is accomplished by shifting the phase of the backscatter clock by 180 degrees. Two common types of PSK are:
   - Change phase at any '0', or
   - Change phase at any data change (0 to 1 or 1 to 0).

   PSK provides fairly good noise immunity, a moderately simple reader design, and a faster data rate than FSK. Typical applications utilize a backscatter clock of FC/2, as shown in Figure 4.

FIGURE 3: FSK MODULATED SIGNAL, FC/8 = 0, FC/10 = 1

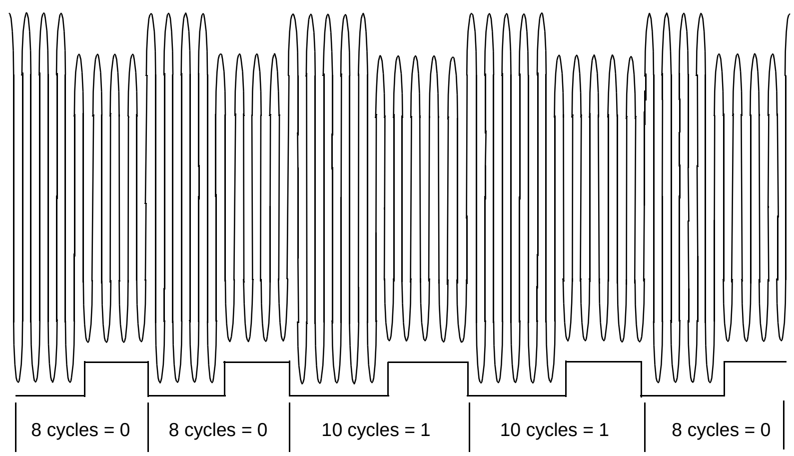
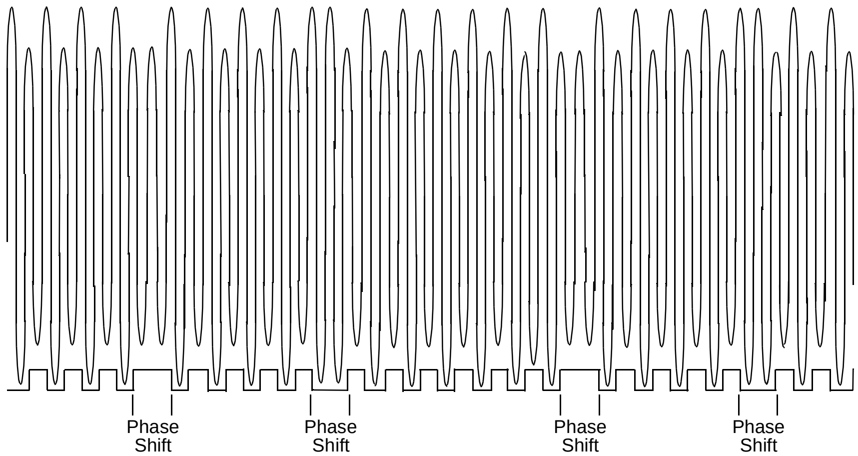


FIGURE 4: PSK MODULATED SIGNAL



ANTI-COLLISION

In many existing applications, a single-read RFID tag is sufficient and even necessary: animal tagging and access control are examples. However, in a growing number of new applications, the simultaneous reading of several tags in the same RF field is absolutely critical: library books, airline baggage, garment and retail applications are a few.

In order to read multiple tags simultaneously, the tag and reader must be designed to detect the condition that more than one tag is active. Otherwise, the tags will all backscatter the carrier at the same time and the amplitude-modulated waveforms shown in Figure 3 and Figure 4 would be garbled. This is referred to as a *collision*. No data would be transferred to the reader. The tag/reader interface is similar to a serial bus, even though the “bus” travels through the air. In a wired serial bus application, arbitration is necessary to prevent bus contention. The RFID interface also requires arbitration so that only one tag transmits data over the “bus” at one time.

A number of different methods are in use and in development today for preventing collisions; most are patented or patent pending. Yet, all are related to making sure that only one tag “talks” (backscatters) at any one time. See the *MCRF250* (DS21267), *MCRF355/360* (DS21287) and *MCRF45X* (DS40232) data sheets for various anti-collision algorithms.

NOTES:

MCRF200

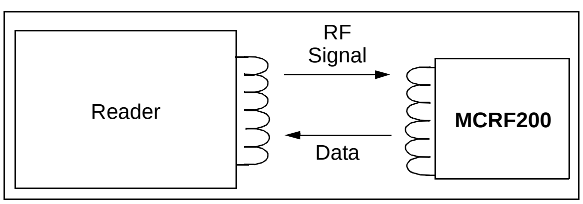
125 kHz microID® Passive RFID Device

Features:

- Factory programming and memory serialization (SQTP[SM])
- One-time contactless programmable (developer kit only)
- Read-only data transmission after programming
- 96 or 128 bits of One-Time Programmable (OTP) user memory (also supports 48 and 64-bit protocols)
- Typical operation frequency: 100 kHz-400 kHz
- Ultra low-power operation (5 µA @ V_{CC} = 2V)
- Modulation options:
  - ASK, FSK, PSK
- Data Encoding options:
  - NRZ Direct, Differential Biphase, Manchester Biphase
- Die, wafer, COB, PDIP or SOIC package options
- Factory programming options

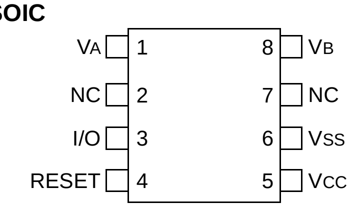
Application:

- Low-cost alternative for existing low-frequency RFID devices
- Access control and time attendance
- Security systems
- Animal tagging
- Product identification
- Industrial tagging
- Inventory control



Package Type

PDIP/SOIC



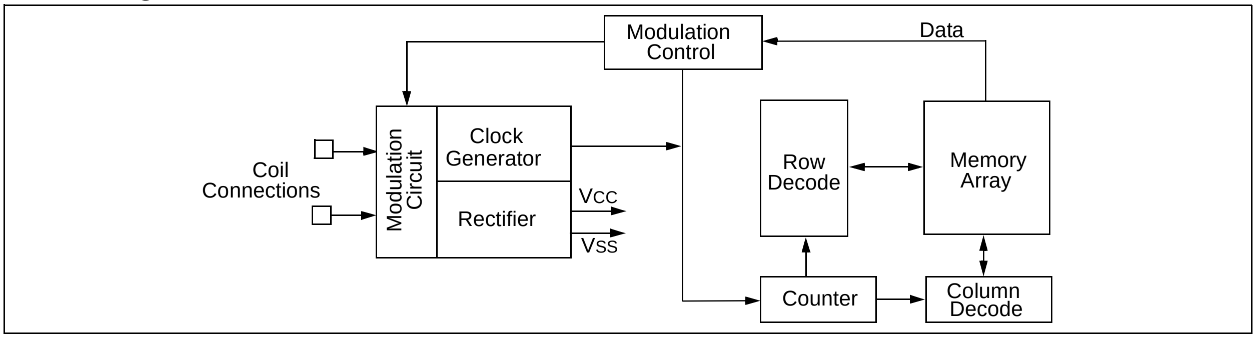
Note: Pins 3, 4, 5 and 6 are for device test purposes only.
Pins 1 and 8 are for antenna connections.
DO NOT ground pin 5.

Description:

The microID® 125 kHz Design Guide is a passive Radio Frequency Identification (RFID) device for low-frequency applications (100 kHz-400 kHz). The device is powered by rectifying an incoming RF signal from the reader. The device requires an external LC resonant circuit to receive the incoming RF signal and to send data. The device develops a sufficient DC voltage for operation when its external coil voltage reaches approximately 10 V_{PP}.

This device has a total of 128 bits of user programmable memory and an additional 12 bits in its configuration register. The user can manually program the 128 bits of user memory by using a contactless programmer in a microID developer kit such as DV103001 or PG103001. However, in production volume the MCRF200 is programmed at the factory (Microchip SQTP – see Technical Bulletin TB023). The device is a One-Time Programmable (OTP) integrated circuit and operates as a read-only device after programming.

Block Diagram



The configuration register includes options for communication protocol (ASK, FSK, PSK), data encoding method, data rate, and data length.These options are specified by customer and factory programmed during assembly. Because of its many choices of configuration options, the device can be easily used as an alternative or second source for most of the existing low frequency passive RFID devices available today.

The device has a modulation transistor between the two antenna connections (V_A and V_B). The modulation transistor damps or undamps the coil voltage when it sends data. The variation of coil voltage controlled by the modulation transistor results in a perturbation of voltage in reader antenna coil. By monitoring the changes in reader coil voltage, the data transmitted from the device can be reconstructed.

The device is available in die, wafer, Chip-on-Board (COB) modules, PDIP, or SOIC packages. Factory programming and memory serialization (SQTP) are also available upon request. See TB023 for more information on contact programming support.

The DV103001 developer’s kit includes Contactless Programmer, ASK, FSK, PSK reference readers, and reference design guide. The reference design guide includes schematics for readers and contactless programmer as well as in-depth document for antenna circuit designs.

1.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings (†)

Storage temperature .... - 65°C to +150°C
Ambient temperature with power applied .... -40°C to +125°C
Maximum current into coil pads .... 50 mA

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 1-1: AC AND DC CHARACTERISTICS

All parameters apply across the specified operating ranges unless otherwise noted.	Industrial (I): T_A = -40°C to +85°C					
Parameter	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
Clock frequency	F_{CLK}	100	—	400	kHz	
Contactless programming time	T_{WC}	—	2	—	sec	For all 128-bit array
Data retention		200	—	—	Years	at 25°C
Coil current (Dynamic)	I_{CD}	—	50		µA	
Operating current	I_{DD}	—	5		µA	V_{CC} = 2V
Turn-on-voltage (Dynamic) for modulation	V_AV_B V_{CC}	10 2	— —	— —	V_{PP} V_{DC}	
Input Capacitance	C_{IN}	—	2	—	pF	Between V_A and V_B

2.0 FUNCTION DESCRIPTION

The device contains three major building blocks. They are RF front-end, configuration and control logic, and memory sections. The Block Diagram is shown on page 1.

2.1 RF Front-End

The RF front-end of the device includes circuits for rectification of the carrier, VDD (operating voltage) and high-voltage clamping. This section also includes a clock generator and modulation circuit.

2.1.1 RECTIFIER – AC CLAMP

The rectifier circuit rectifies RF voltage on the external LC antenna circuit. Any excessive voltage on the tuned circuit is clamped by the internal circuitry to a safe level to prevent damage to the IC.

2.1.2 POWER-ON RESET

This circuit generates a Power-on Reset when the tag first enters the reader field. The Reset releases when sufficient power has developed on the VDD regulator to allow correct operation.

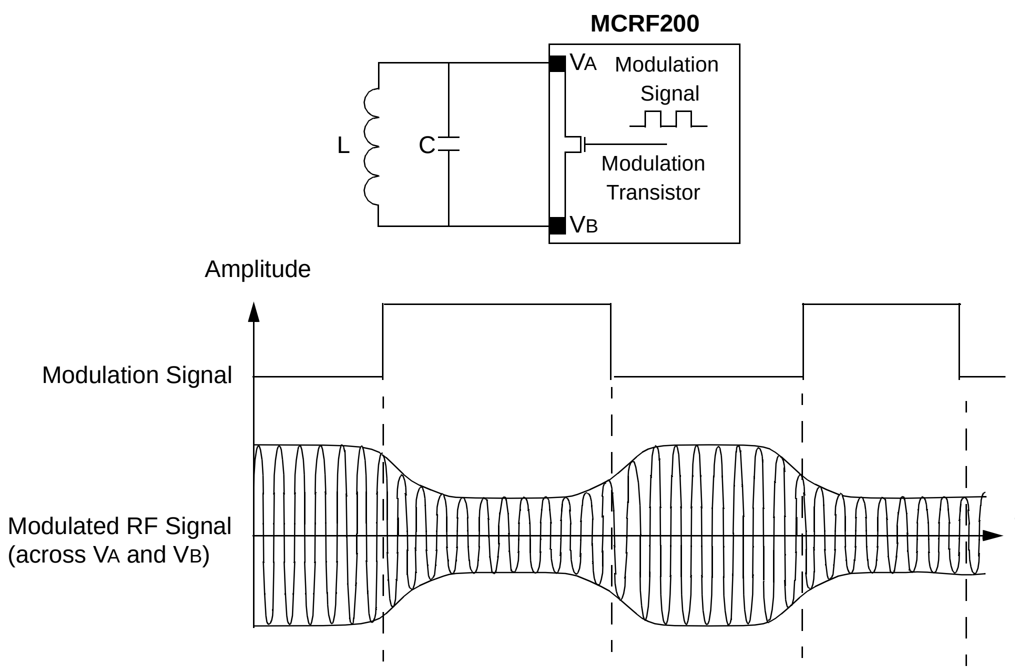
2.1.3 CLOCK GENERATOR

This circuit generates a clock based on the carrier frequency from the reader. This clock is used to derive all timing in the device, including the baud rate and modulation rate.

2.1.4 MODULATION CIRCUIT

The device sends the encoded data to the reader by AM-modulating the coil voltage across the tuned LC circuit. A modulation transistor is placed between the two antenna coil pads (VA and VB). The transistor turns on and off based on the modulation signal. As a result, the amplitude of the antenna coil voltage varies with the modulation signal. See Figure 2-1 for details.

FIGURE 2-1: MODULATION SIGNAL AND MODULATED SIGNAL



2.2 Configuration Register and Control Logic

The configuration register determines the operational parameters of the device. The configuration register can not be programmed contactlessly; it is programmed during wafer probe at the Microchip factory. CB11 is always a zero; CB12 is set when successful contact or contactless programming of the data array has been completed. Once CB12 is set, device programming and erasing is disabled. Table 2-4 contains a description of the bit functions of the control register.

2.2.1 BAUD RATE TIMING OPTION

The chip will access data at a baud rate determined by bits CB2, CB3 and CB4 of the configuration register. For example, MOD32 (CB2 = 0, CB3 = 1, CB4 = 1) has 32 RF cycles per bit. This gives the data rate of 4 kHz for the RF carrier frequency of 128 kHz.

The default timing is MOD128 (FCLK/128), and this mode is used for contact and contactless programming. Once the array is successfully programmed, the lock bit CB12 is set. When the lock bit is set, programming and erasing the device becomes permanently disabled. The configuration register has no effect on device timing until the EEPROM data array is programmed (CB12 = 1).

2.2.2 DATA ENCODING OPTION

This logic acts upon the serial data being read from the EEPROM. The logic encodes the data according to the configuration bits CB6 and CB7. CB6 and CB7 determine the data encoding method. The available choices are:

- Non-return to zero-level (NRZ_L)
- Biphase Differential, Biphase Manchester
- Inverted Manchester

2.2.3 MODULATION OPTION

CB8 and CB9 determine the modulation protocol of the encoded data. The available choices are:

- ASK
- FSK
- PSK_1
- PSK_2

When ASK (direct) option is chosen, the encoded data is fed into the modulation transistor without change.

When FSK option is chosen, the encoded data is represented by:

a) Sets of 10 RF carrier cycles (first 5 cycles → higher amplitude, the last 5 cycles → lower amplitude) for logic “high” level.

b) Sets of 8 RF carrier cycles (first 4 cycles → higher amplitude, the last 4 cycles → lower amplitude) for logic “low” level.

For example, FSK signal for MOD40 is represented:

a) 4 sets of 10 RF carrier cycles for data ‘1’.

b) 5 sets of 8 RF carrier cycles for data ‘0’.

Refer to Figure 2-2 for the FSK signal with MOD40 option.

The PSK_1 represents change in the phase of the modulation signal at the change of the encoded data. For example, the phase changes when the encoded data is changed from ‘1’ to ‘0’, or from ‘0’ to ‘1’.

The PSK_2 represents change in the phase at the change on ‘1’. For example, the phase changes when the encoded data is changed from ‘0’ to ‘1’, or from ‘1’ to ‘1’.

FIGURE 2-2: ENCODED DATA AND FSK OUTPUT SIGNAL FOR MOD40 OPTION

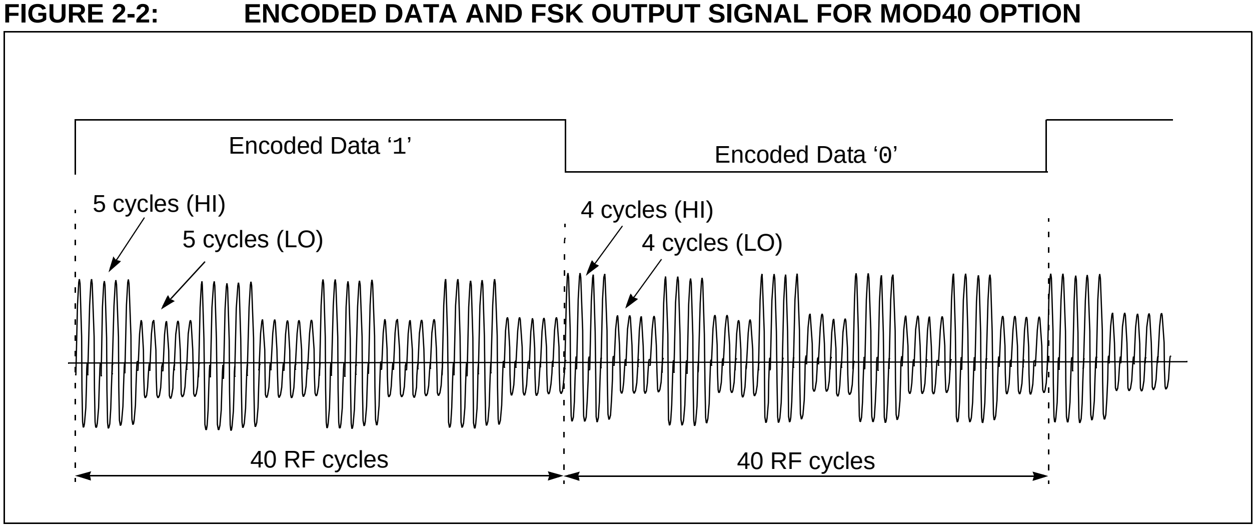
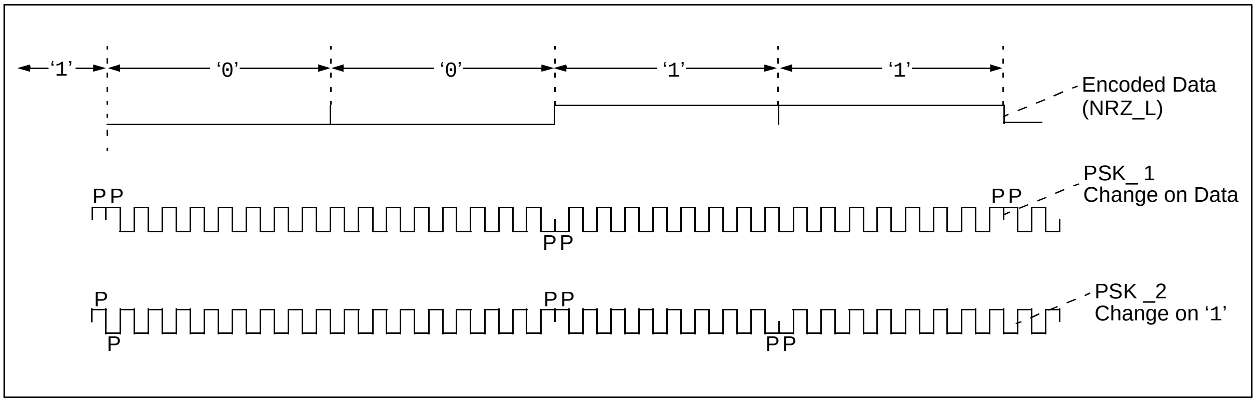


FIGURE 2-3: PSK DATA MODULATION



2.2.4 MEMORY ARRAY LOCK BIT (CB12)

The CB12 must be ‘0’ for contactless programming (Blank). The bit (CB12) is automatically set to ‘1’ as soon as the device is programmed contactlessly.

2.3 Memory Section

The device has 128 bits of one-time programmable (OTP) memory. The user can choose 96 or 128 bits by selecting the CB1 bit in the configuration register. See Table 2-4 for more details.

2.3.1 COLUMN AND ROW DECODER LOGIC AND BIT COUNTER

The column and row decoders address the EEPROM array at the clock rate and generate a serial data stream for modulation. This data stream can be up to 128 bits in length. The size of the data stream is user programmable with CB1 and can be set to 96 or 128 bits. Data lengths of 48 and 64 bits are available by programming the data twice in the array, end-to-end.

The column and row decoders route the proper voltage to the array for programming and reading. In the programming modes, each individual bit is addressed serially from bit 1 to bit 128.

2.4 Examples of Configuration Settings

EXAMPLE 2-1: “08D” CONFIGURATION

The “08D” (hex) configuration is interpreted as follows:

CB12 ↓ ... CB1 ↓

“08D” → `0000-1000-1101`

Referring to Table 2-4, the “08D” configuration represents:

Modulation = PSK_1
PSK rate = rf/2
Data encoding = NRZ_L (direct)
Baud rate = rf/32 = MOD32
Memory size 128 bits

EXAMPLE 2-2: “00A” CONFIGURATION

The “00A” (hex) configuration is interpreted as follows:

CB12 ↓ ... CB1 ↓

“00A” → `0000-0000-1010`

The MSB corresponds to CB12 and the LSB corresponds to CB1 of the configuration register. Therefore, we have:

CB12=0	CB11=0	CB10=0	CB9=0
CB8=0	CB7=0	CB6=0	CB5=0
CB4=1	CB3=0	CB2=1	CB1=0

Referring to Table 2-4, the “00A” configuration represents:

Not programmed device (blank), anti-collision: disabled, FSK protocol, NRZ_L (direct) encoding, MOD50 (baud rate = rf/50), 96 bits.

EXAMPLE 2-3: MCRF200 CONFIGURATION FOR FDX-B ISO ANIMAL STANDARD PROTOCOL (ASP)

The FDX-B ISO Specification is:

Modulation = ASK
Data encoding = Differential biphase
Baud rate = rf/32 = 4 Kbits/sec for 128 kHz
Memory size = 128 bits

Referring to Table 2-4, the equivalent MCRF200 configuration is: “14D”.

TABLE 2-4: CONFIGURATION REGISTER

CB12	CB11	CB10	CB9	CB8	CB7	CB6	CB5	CB4	CB3	CB2	CB1

MEMORY SIZE
CB1 = 1 128-bit user memory array
CB1 = 0 96-bit user memory array

BAUD RATE

CB2	CB3	CB4	Rate
0	0	0	MOD128
0	0	1	MOD100
0	1	0	MOD80
0	1	1	MOD32
1	0	0	MOD64
1	0	1	MOD50
1	1	0	MOD40
1	1	1	MOD16

SYNC WORD
CB5 = 0 (Always)

DATA ENCODING
CB6 = 0; CB7 = 0 NRZ_L (Direct)
CB6 = 0; CB7 = 1 Biphase_S (Differential)
CB6 = 1; CB7 = 0 Biphase_L (Manchester)
CB6 = 1; CB7 = 1 (Inverted Manchester)

MODULATION OPTIONS
CB8 = 0; CB9 = 0 FSK 0 = Fc/8, 1 = Fc/10
CB8 = 0; CB9 = 1 Direct (ASK)
CB8 = 1; CB9 = 0 PSK_1
(phase change on change of data)
CB8 = 1; CB9 = 1 PSK_2
(phase change at beginning of a one)

PSK RATE OPTION
CB10 = 1 Carrier/4
CB10 = 0 Carrier/2

(ANTI-COLLISION OPTION (Read-only)
CB11 = 0 Disabled (Always)

MEMORY ARRAY LOCK BIT (Read-only)
CB12 = 0 User memory array not locked (Blank)
CB12 = 1 User memory array is locked
(Programmed)

3.0 MODES OF OPERATION

The device has two basic modes of operation: Native mode and Read mode.

3.1 Native Mode

Every unprogrammed blank device (CB12 = 0) operates in Native mode, regardless of configuration register settings:

FCLK/128, FSK, NRZ_L (direct)

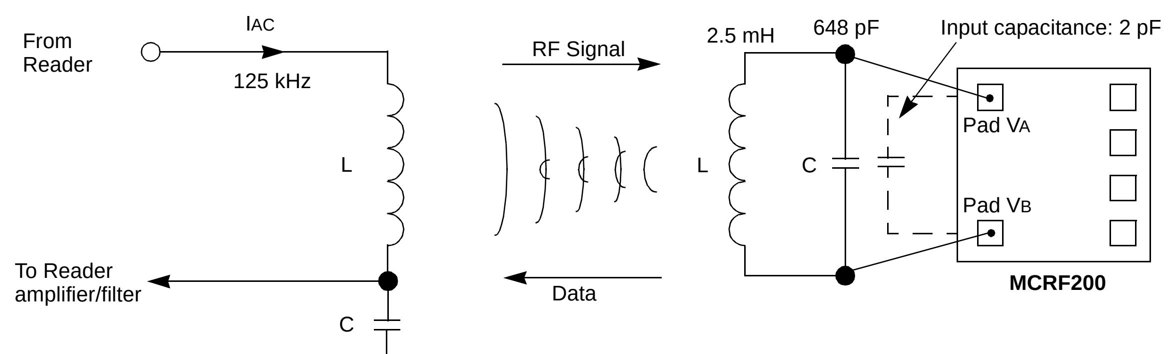
Once the user memory is programmed, the lock bit is set (CB12 = 1) which causes the MCRF200 to switch from Native mode to the Communication mode defined by the configuration register.

Refer to Figure 4-1 for contactless programming sequence. Also see the *microID® 125 kHz RFID System Design Guide* (DS51115) for more information.

3.2 Read Mode

After the device is programmed (CB12 = 1), the device is operated in the Read-only mode. The device transmits its data according to the protocol in the configuration register.

FIGURE 3-1: TYPICAL APPLICATION CIRCUIT



$$f_{res} = \frac{1}{2\pi\sqrt{LC}} = 125\ kHz$$

4.0 CONTACTLESS PROGRAMMING

The contactless programming of the device is possible for blank devices (CB12 = 0) only and is recommended for only low-volume, manual operation during development. In volume production, the MCRF200 is normally used as a factory programmed device only. The contactless programming timing sequence consists of:

a) RF power-up signal.
b) Short gap (absence of RF field).
c) Verify signal (continuous RF signal).
d) Programming signal.
e) Device response with programmed data.

The blank device (CB12 = 0) understands the RF power-up followed by a gap as a blank checking command, and outputs 128 bits of FSK data with all ‘1’s after the short gap. To see this blank data (verify), the reader/programmer must provide a continuous RF signal for 128 bit-time. (The blank (unprogrammed) device has all ‘F’s in its memory array. Therefore, the blank data should be all ‘1’s in FSK format). Since the blank device operates at Default mode (MOD128), there are 128 RF cycles for each bit. Therefore, the time requirement to complete this verify is 128 bits x 128 RF cycles/bit x 8 use/cycles = 131.1 msec for 125 kHz signal.

As soon as the device completes the verify, it enters the programming mode. The reader/programmer must provide RF programming data right after the verify. In this programming mode, each bit lasts for 128 RF cycles. Refer to Figure 4-1 for the contactless programming sequence.

Customer must provide the following specific voltage for the programming:

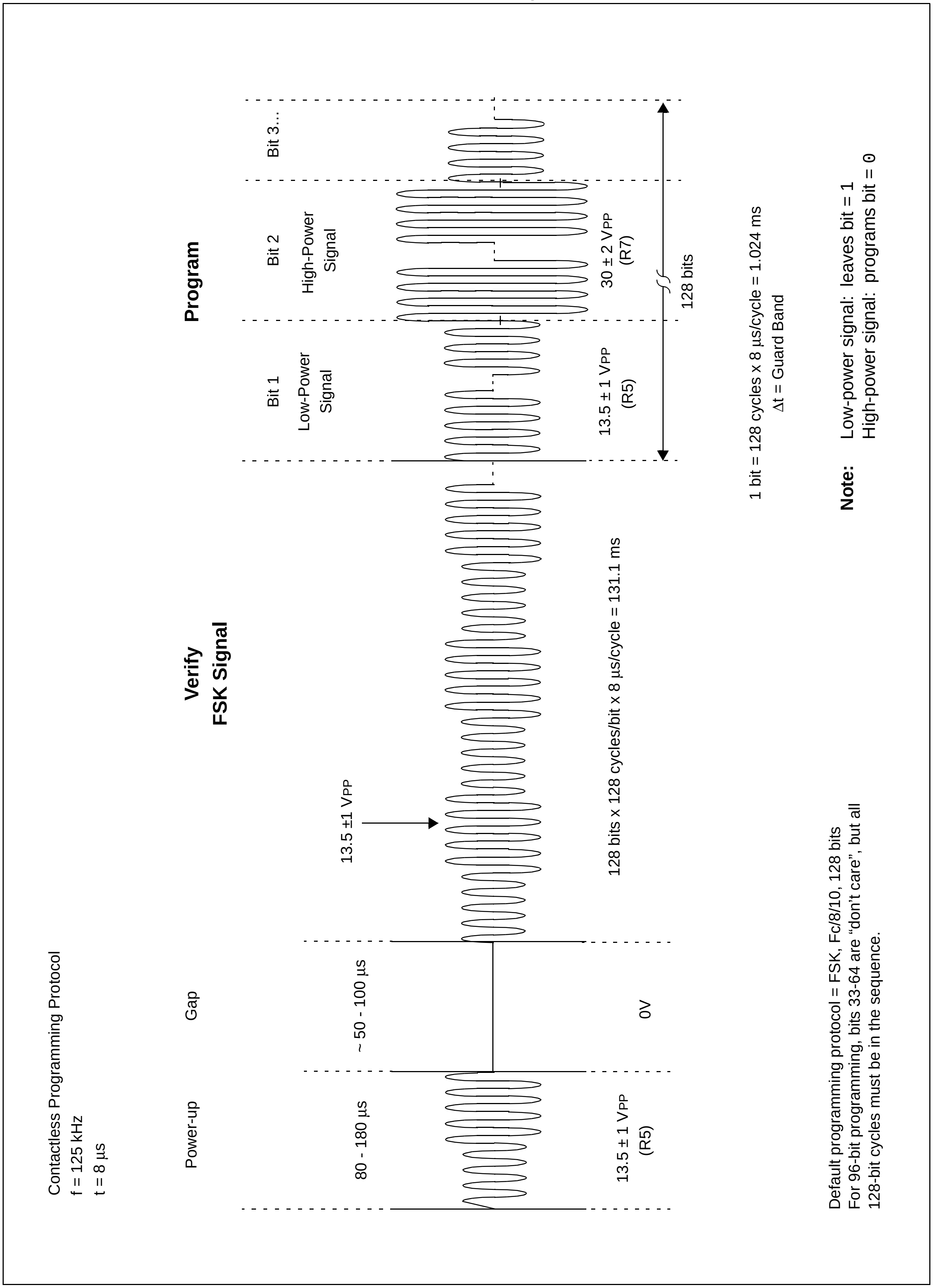
1. Power-up and verify signal = 13.5V ±1 V_{PP}
2. Programming voltage:
   - To program bit to ‘1’: 13.5V ±1 V_{PP}
   - To program bit to ‘0’: 30V ±2 V_{PP}

After the programming cycle, the device outputs programmed data (response). The reader/programmer can send the programming data repeatedly after the device response until the programming is successfully completed. The device locks the CB12 as soon as the programming mode (out of field) is exited and becomes a read-only device.

Once the device is programmed (CB12 = 1), the device outputs its data according to the configuration register.

The PG103001 (Contactless Programmer) is used for the programming of the device. The voltage level shown in Figure 4-1 is adjusted by R5 and R7 in the contactless programmer. Refer to the *MicroID® 125 kHz RFID System Design Guide* (DS51115) for more information.

FIGURE 4-1: CONTACTLESS PROGRAMMING SEQUENCE



5.0 MECHANICAL SPECIFICATIONS FOR DIE AND WAFER

FIGURE 5-1: DIE PLOT

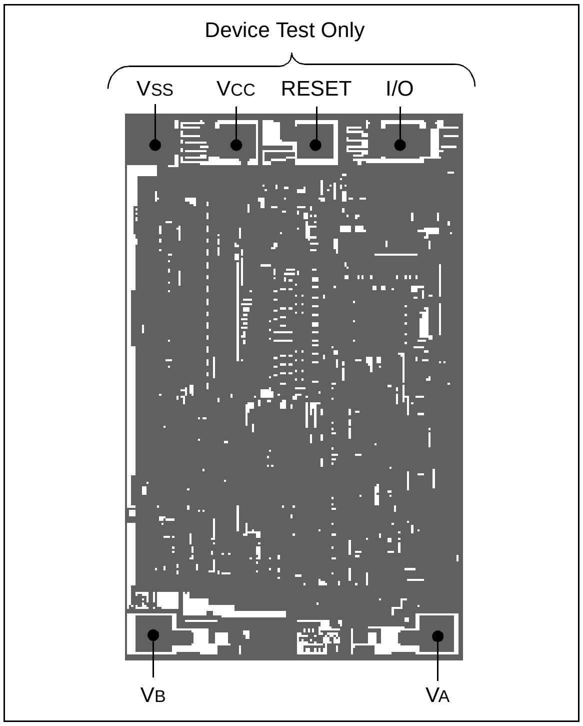


TABLE 5-1: PAD COORDINATES (μm)

Pad Name	Passivation Openings: Pad Width	Passivation Openings: Pad Height	Pad Center X	Pad Center Y
VA	90.0	90.0	427.50	-734.17
VB	90.0	90.0	-408.60	-734.17

Note 1: All coordinates are referenced from the center of the die.

2: Die size: 1.1215 mm x 1.7384 mm
44.15 mils x 68.44 mils

TABLE 5-2: PAD FUNCTION TABLE

Name	Function
VA	Antenna Coil connection
VB	
VSS	For device test only Do Not Connect to Antenna
VCC	
RESET	
I/O	

TABLE 5-3: DIE MECHANICAL DIMENSIONS

Specifications	Min	Typ	Max	Unit	Comments
Bond pad opening	— —	3.5 x 3.5 89 x 89	— —	mil µm	**Note 1, Note 2**
Die backgrind thickness	— —	7 177.8	— —	mil µm	Sawed 6” wafer on frame (option = WF) **Note 3**
	— —	11 279.4	— —	mil µm	Unsawed wafer (option = W) **Note 3**
Die backgrind thickness tolerance	— —	— —	±1 ±25.4	mil µm	
Die passivation thickness (multilayer)	—	0.9050	—	µm	**Note 4**
Die Size: Die size X*Y before saw (step size) Die size X*Y after saw	 — —	 44.15 x 68.44 42.58 x 66.87	 — —	 mil mil	 — —

Note 1: The bond pad size is that of the passivation opening. The metal overlaps the bond pad passivation by at least 0.1 mil.

2: Metal Pad Composition is 98.5% Aluminum with 1% Si and 0.5% Cu.

3: As the die thickness decreases, susceptibility to cracking increases. It is recommended that the die be as thick as the application will allow.

4: The Die Passivation thickness (0.905 µm) can vary by device depending on the mask set used. The passivation is formed by:
- Layer 1: Oxide (undoped oxide 0.135 µm)
- Layer 2: PSG (doped oxide, 0.43 µm)
- Layer 3: Oxynitride (top layer, 0.34 µm)

5: The conversion rate is 25.4 µm/mil.

Notice: Extreme care is urged in the handling and assembly of die products since they are susceptible to mechanical and electrostatic damage.

TABLE 5-4: WAFER MECHANICAL SPECIFICATIONS

Specifications	Min	Typ	Max	Unit	Comments
Wafer Diameter	—	8	—	inch	150 mm
Die separation line width	—	80	—	µm	
Dice per wafer	—	14,000	—	die	
Batch size	—	24	—	wafer	

6.0 FAILED DIE IDENTIFICATION

Every die on the wafer is electrically tested according to the data sheet specifications and visually inspected to detect any mechanical damage such as mechanical cracks and scratches.

Any failed die in the test or visual inspection is identified by black colored ink. Therefore, any die covered with black ink should not be used.

The ink dot specification:

- Ink dot size: minimum 20 µm x 20 µm
- Position: central third of die
- Color: black

7.0 WAFER DELIVERY DOCUMENTATION

Each wafer container is marked with the following information:

- Microchip Technology Inc. MP Code
- Lot Number
- Total number of wafers in the container
- Total number of good dice in the container
- Average die per wafer (DPW)
- Scribe number of wafers with number of good dice

8.0 NOTICE ON DIE AND WAFER HANDLING

The device is very susceptible to Electrostatic Discharge (ESD). ESD can cause critical damage to the device. Special attention is needed during the handling process.

Any untraviolet (UV) light can erase the memory cell contents of an unpackaged device. Flourescent lights and sun light can also erase the memory cell although it takes more time than UV lamps. Therefore, keep any unpackaged devices out of UV light and also avoid direct exposure from strong flourescent lights and sun light.

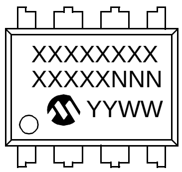
Certain integrated circuit (IC) manufacturing, chip-on-board (COB) and tag assembly operations may use UV light. Operations such as backgrind, de-tape, certain cleaning operations, epoxy or glue cure should be done without exposing the die surface to UV light.

Using x-ray for die inspection will not harm the die, nor erase memory cell contents.

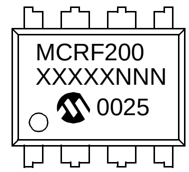
9.0 PACKAGING INFORMATION

9.1 Package Marking Information

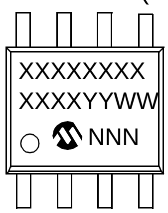
8-Lead PDIP (300 mil)



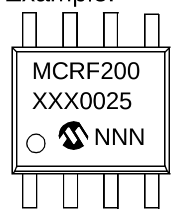
Example:



8-Lead SOIC (150 mil)



Example:

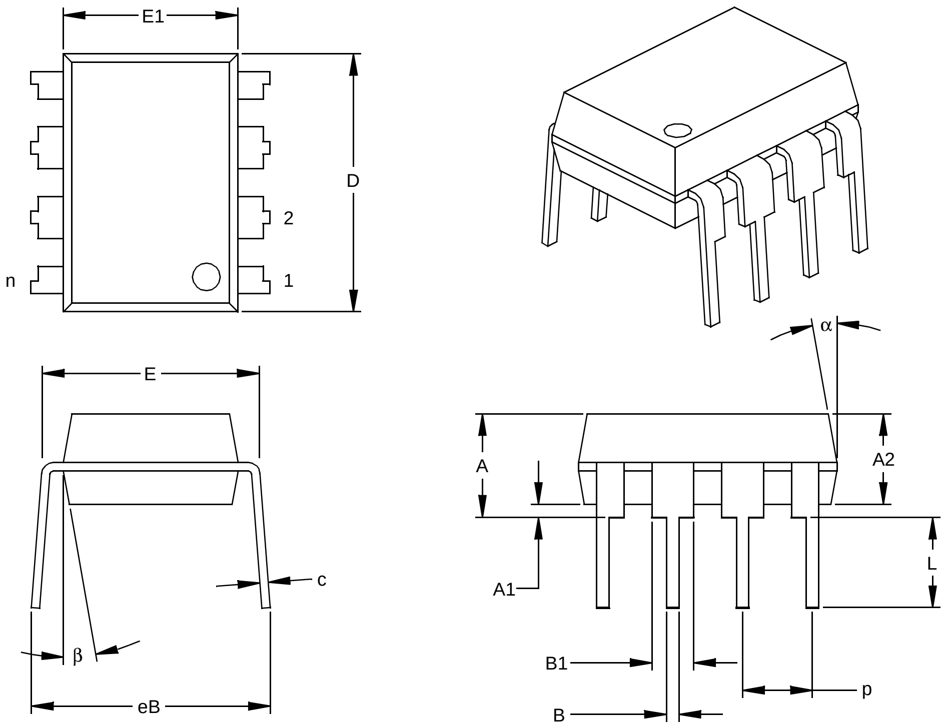


Legend:	XX...X	Customer specific information*
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard device marking consists of Microchip part number, year code, week code, and traceability code.

8-Lead Plastic Dual In-line (P) – 300 mil (PDIP)

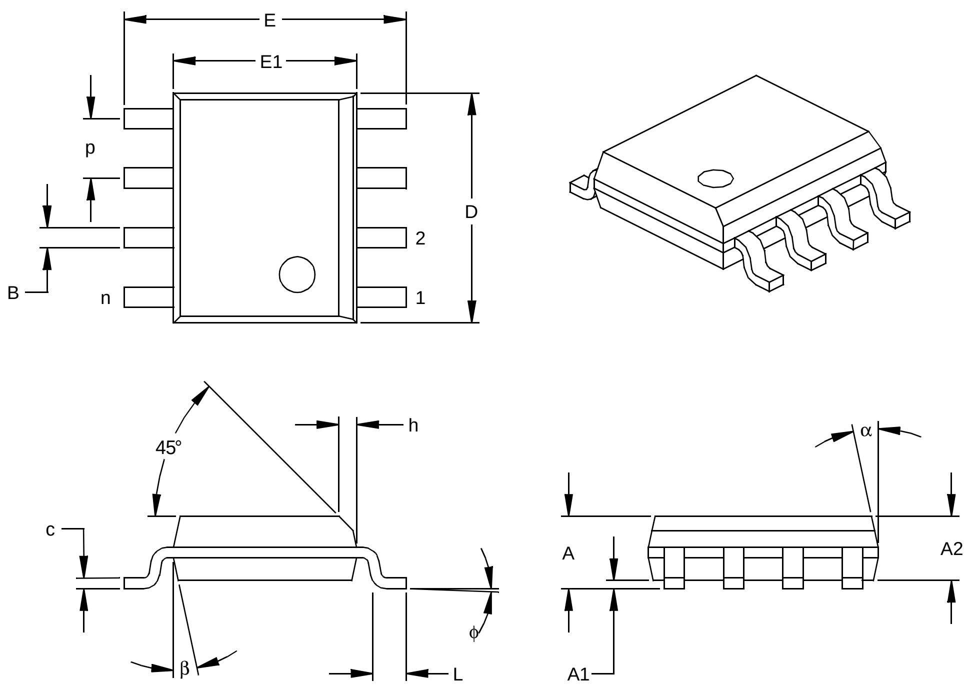


UNITS		INCHES*			MILLIMETERS		
DIMENSION LIMITS		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.360	.373	.385	9.14	9.46	9.78
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-018

8-Lead Plastic Small Outline (SN) – Narrow, 150 mil (SOIC)

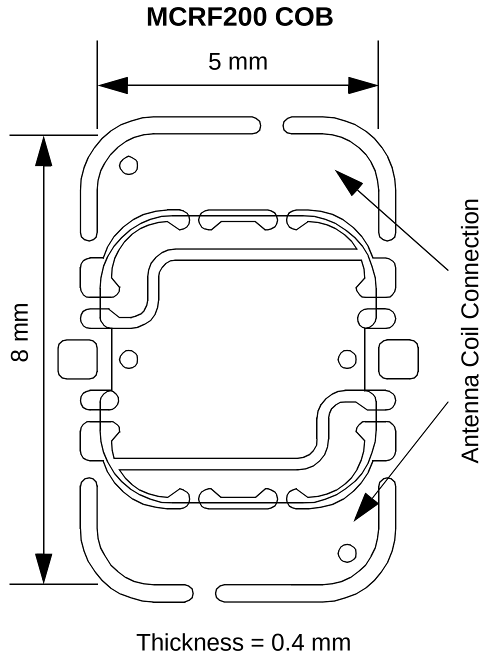


UNITS		INCHES*			MILLIMETERS		
DIMENSION LIMITS		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.050			1.27	
Overall Height	A	.053	.061	.069	1.35	1.55	1.75
Molded Package Thickness	A2	.052	.056	.061	1.32	1.42	1.55
Standoff §	A1	.004	.007	.010	.10	.18	.25
Overall Width	E	.228	.237	.244	5.79	6.02	6.20
Molded Package Width	E1	.146	.154	.157	3.71	3.91	3.99
Overall Length	D	.189	.193	.197	4.80	4.90	5.00
Chamfer Distance	h	.010	.015	.020	.25	.38	.51
Foot Length	L	.019	.025	.030	.48	.62	.76
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.008	.009	.010	.20	.23	.25
Lead Width	B	.013	.017	.020	.33	.42	.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.
JEDEC Equivalent: MS-012
Drawing No. C04-057

1M/3M COB (IOA2)



PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Configuration/SQTP code

Device	MCRF200 = 125 kHz Contactless Programmable MicroID® tag, 96/128-bit
Temperature Range	I = -40°C to +85°C (Industrial)
Package	WF = Sawed wafer on frame (7 mil backgrind) W = Wafer (11 mil backgrind) S = Dice in waffle pack P = Plastic PDIP (300 mil Body) 8-lead SN = Plastic SOIC (150 mil Body) 8-lead 1M = 0.40 mm (I0A2 package) COB Module w/1000 pF capacitor 3M = 0.40 mm (I0A2 package) COB Module with 330 pF capacitor
Configuration	Three-digit HEX value to be programmed into the configuration register. Three HEX characters correspond to 12 binary bits. These bits are programmed into the configuration register MSB first (CB12, CB11...CB1). Refer to example.
SQTP Code	An assigned custom, 3-digit code used for tracking and controlling production and customer data files for factory programming. In this case the configuration code is not shown in the part number, but is captured in the SQTP documentation.

Examples:

a) MCRF200-I/W00A = 125 kHz, industrial temperature, wafer package, contactlessly programmable, 96 bit, FSK Fc/8 Fc/10, direct encoded, Fc/50 data return rate tag.

b) MCRF200-I/WFQ23 = 125 kHz, industrial temperature, wafer sawn and mounted on frame, factory programmed.

The configuration register is:

CB12	CB11	CB10	CB9	CB8	CB7	CB6	CB5	CB4	CB3	CB2	CB1
0	0	0	0	0	0	0	0	1	0	1	0

Sales and Support

Data Sheets
Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 792-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System
Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

MCRF250

125 kHz microID® Passive RFID Device with Anti-Collision

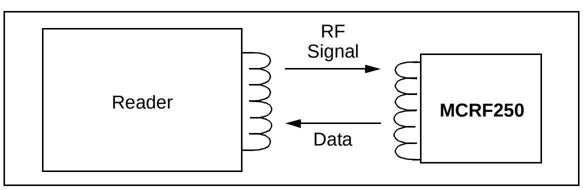
Features:

Factory programming and memory serialization (SQTP℠)

- Anti-collision feature to read multiple tags in the same RF field.
- One-time contactless programmable (developer kit only)
- Read-only data transmission after programming
- 96 or 128 bits of One-Time Programmable (OTP) user memory (also supports 48- and 64-bit protocols)
- Typical operation frequency: 100 kHz-400 kHz
- Ultra low-power operation (5 µA @ V_{CC} = 2V)
- Modulation options:
  - ASK, FSK, PSK
- Data Encoding options:
  - NRZ Direct, Differential Biphase, Manchester Biphase
- Die, wafer, COB or SOIC package options
- Factory programming options

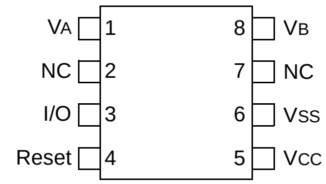
Applications:

- Access control and time attendance
- Security systems
- Animal tagging
- Product identification
- Industrial tagging
- Inventory control
- Multiple item tagging



Package Type

PDIP/SOIC



Note: Pins 3, 4, 5 and 6 are for device test purposes only. Pins 1 and 8 are for antenna connection.

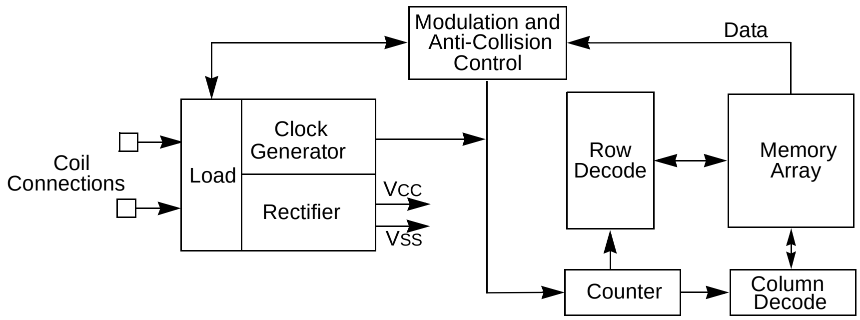
Description:

The MCRF250 is equipped with an anti-collision feature that allows multiple tags in the same field to be read simultaneously. This revolutionary feature eliminates the issue of data corruption due to simultaneous transmissions from multiple tags.

The microID® 125 kHz Design Guide is a passive Radio Frequency Identification (RFID) device for low frequency applications (100 kHz-400 kHz). The device is powered by rectifying an incoming RF signal from a reader interrogator. The device requires an external LC resonant circuit to receive the incoming energizing signal and to send data. The device develops a sufficient DC voltage for operation when it’s external coil voltage reaches approximately 10 V_{PP}.

This device has a total of 128 bits of user programmable memory and an additional 12 bits in its configuration register. The user can manually program the 128 bits of user memory by using a contactless programmer in a microID developer kit such as DV103001 or PG103001. However, in production volume the MCRF250 is programmed at the factory (Microchip SQTP – see Technical Bulletin TB023).The device is a One-Time Programmable (OTP) integrated circuit and operates as a read-only device after programming.

Block Diagram



The configuration register includes options for communication protocol (ASK, FSK, PSK), data encoding method, data rate and data length.These options are specified by customer and are factory programmed during production.

The device has a modulation transistor between the two antenna connections (V_A and V_B). The modulation transistor damps or undamps the coil voltage when it sends data. The variation of coil voltage controlled by the modulation transistor results in a perturbation of voltage in reader antenna coil. By monitoring the changes in reader coil voltage, the data transmitted from the device can be reconstructed.

The device is available in die, wafer, Chip-on-Board (COB) modules, PDIP or SOIC packages. Factory programming and memory serialization (SQTP) are also available upon request. See TB023 for more information on contact programming support.

The DV103002 Developer's Kit includes Contactless Programmer, MCRF250 Anti-Collision FSK reference reader, and reference design guide. The reference design guide includes schematics for readers and contactless programmer as well as in-depth documentation for antenna circuit designs.

1.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings(†)

Storage temperature .... -65°C to +150°C
Ambient temperature with power applied .... -40°C to +125°C
Maximum current into coil pads .... 50 mA

† **NOTICE**: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 1-1: AC AND DC CHARACTERISTICS

All parameters apply across the specified operating ranges unless otherwise noted.	Industrial (I): T_A = -40°C to +85°C					
Parameter	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
Clock frequency	F_{CLK}	100	—	400	kHz	
Contactless programming time	T_{WC}	—	2	—	sec	For all 128-bit array
Data retention		200	—	—	Years	at 25°C
Coil current (Dynamic)	I_{CD}	—	50		µA	
Operating current	I_{DD}	—	5		µA	V_{CC} = 2V
Turn-on-voltage (Dynamic) for modulation	V_AV_B V_{CC}	10 2	— —	— —	V_{PP} V_{DC}	
Input Capacitance	C_{IN}	—	2	—	pF	Between V_A and V_B

2.0 FUNCTIONAL DESCRIPTION

The device contains three major building blocks. They are RF front-end, configuration and control logic, and memory sections. The Block Diagram is shown on page 1.

2.1 RF Front-End

The RF front-end of the device includes circuits for rectification of the carrier, V_{DD} (operating voltage), and high-voltage clamping to prevent excessive voltage from being applied to the device. This section also generates a system clock from the incoming carrier signal and modulates the carrier signal to transmit data to the reader.

2.1.1 RECTIFIER – AC CLAMP

The rectifier circuit rectifies RF voltage on the external LC antenna circuit. Any excessive voltage on the tuned circuit is clamped by the internal circuitry to a safe level to prevent damage to the IC.

2.1.2 POWER-ON RESET

This circuit generates a Power-on Reset when the tag first enters the reader field. The Reset releases when sufficient power has developed on the V_{DD} regulator to allow correct operation.

2.1.3 CLOCK GENERATOR

This circuit generates a clock based on the carrier frequency from the reader. This clock is used to derive all timing in the device, including the baud rate and modulation rate.

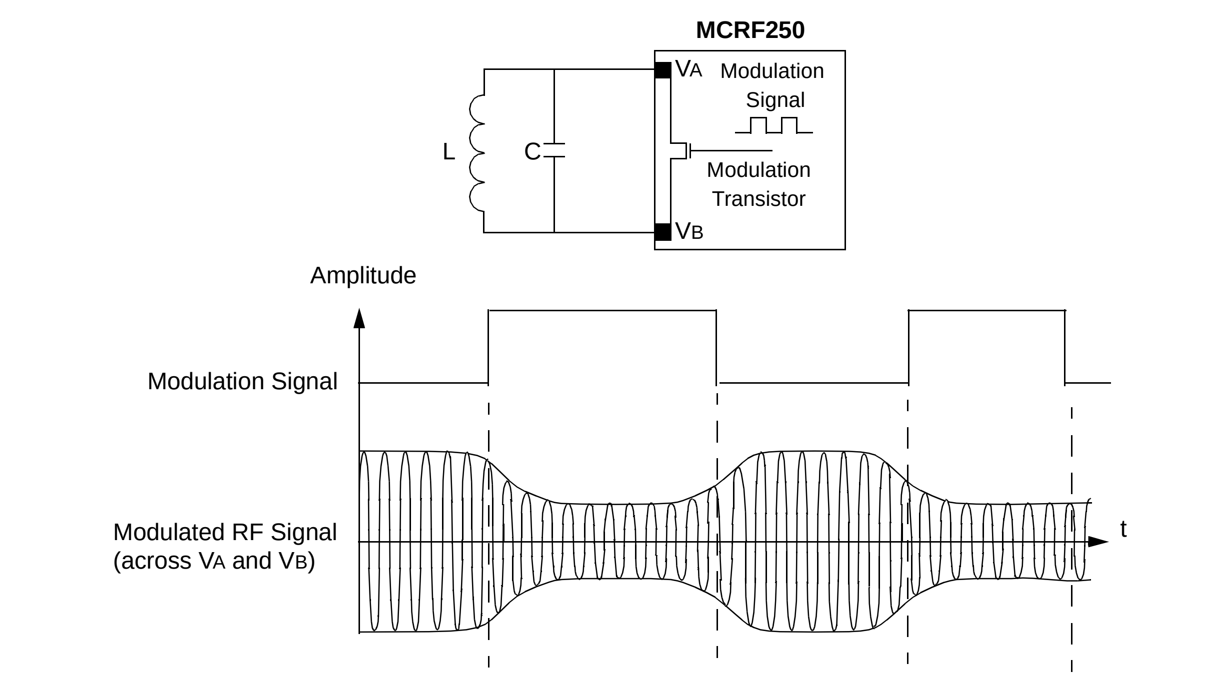
2.1.4 IRQ DETECTOR

This circuitry detects an interrupt in the continuous electromagnetic field of the interrogator. An IRQ (interrupt request) is defined as the absence of the electromagnetic field for a specific number of clock cycles. Detection of an IRQ will trigger the device to enter the Anti-collision mode. This mode is discussed in detail in **Section 5.0 “Anti-Collision”**.

2.1.5 MODULATION CIRCUIT

The device sends the encoded data to the reader by AM-modulating the coil voltage across the tuned LC circuit. A modulation transistor is placed between the two antenna coil pads (V_A and V_B). The transistor turns on and off based on the modulation signal. As a result, the amplitude of the antenna coil voltage varies with the modulation signal. See Figure 2-1 for details.

FIGURE 2-1: MODULATION SIGNAL AND MODULATED SIGNAL



2.2 Configuration Register and Control Logic

The configuration register determines the operational parameters of the device. The configuration register can not be programmed contactlessly; it is programmed during wafer probe at the Microchip factory. CB11 is always a one; CB12 is set when successful contact or contactless programming of the data array has been completed. Once CB12 is set, device programming and erasing is disabled. Table 2-1 contains a description of the bit functions of the control register.

2.2.1 BAUD RATE TIMING OPTION

The chip will access data at a baud rate determined by bits CB2, CB3, and CB4 of the configuration register. For example, MOD32 (CB2 = 0, CB3 = 1, CB4 = 1) has 32 RF cycles per bit. This gives the data rate of 4 kHz for the RF carrier frequency of 128 kHz.

The default timing is MOD 128 (FCLK/128), and this mode is used for contact and contactless programming. Once the array is successfully programmed, the lock bit CB12 is set. When the lock bit is set, programming and erasing the device becomes permanently disabled. The configuration register has no effect on device timing until the EEPROM data array is programmed (CB12 = 1).

2.2.2 DATA ENCODING OPTION

This logic acts upon the serial data being read from the EEPROM. The logic encodes the data according to the configuration bits CB6 and CB7. CB6 and CB7 determine the data encoding method. The available choices are:

- Non-return to zero-level (NRZ_L)
- Biphase_S (Differential)
- Biphase_L (Manchester)
- Inverted Manchester

2.2.3 MODULATION OPTION

CB8 and CB9 determine the modulation protocol of the encoded data. The available choices are:

- ASK
- FSK
- PSK_1
- PSK_2

When ASK (direct) option is chosen, the encoded data is fed into the modulation transistor without change.

When FSK option is chosen, the encoded data is represented by:

a) Sets of 10 RF carrier cycles (first 5 cycles → higher amplitude, the last 5 cycles → lower amplitude) for logic "high" level.

b) Sets of 8 RF carrier cycles (first 4 cycles → higher amplitude, the last 4 cycles → lower amplitude) for logic "low" level.

For example, FSK signal for MOD40 is represented:

a) 4 sets of 10 RF carrier cycles for data '1'.

b) 5 sets of 8 RF carrier cycles for data '0'.

Refer to Figure 2-2 for the FSK signal with MOD40 option.

The PSK_1 represents change in the phase of the modulation signal at the change of the encoded data. For example, the phase changes when the encoded data is changed from '1' to '0', or from '0' to '1'.

The PSK_2 represents change in the phase at the change on '1'. For example, the phase changes when the encoded data is changed from '0' to '1', or from '1' to '1'.

FIGURE 2-2: ENCODED DATA AND FSK OUTPUT SIGNAL FOR MOD40 OPTION

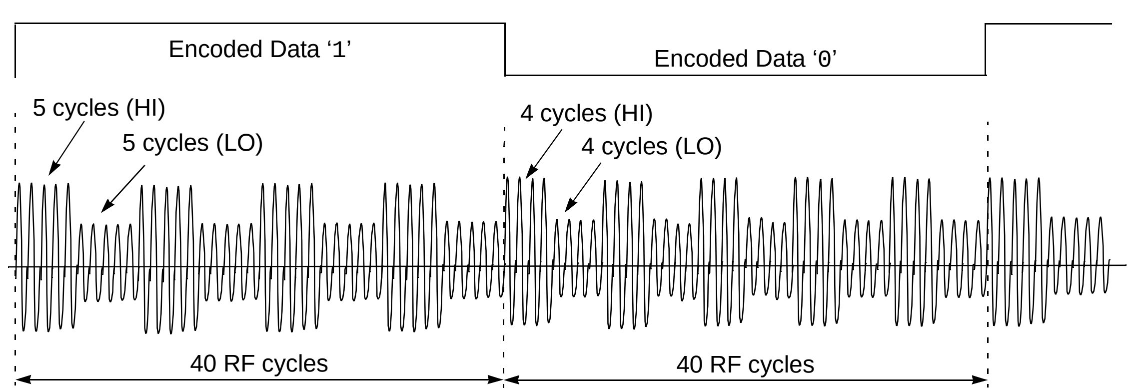
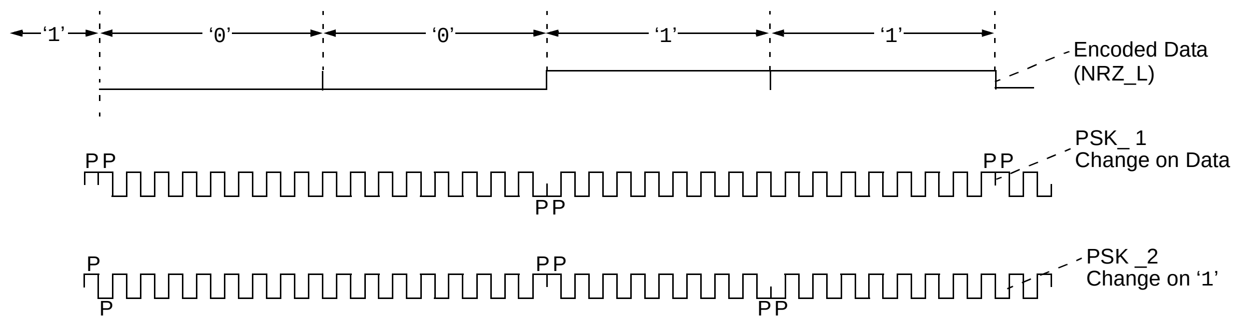


FIGURE 2-3: PSK DATA MODULATION



2.2.4 MEMORY ARRAY LOCK BIT (CB12)

The CB12 must be '0' for contactless programming (Blank). The bit (CB12) is automatically set to '1' itself as soon as the device is programmed contactlessly.

2.3 Memory Section

The device has 128 bits of one-time programmable (OTP) memory. The user can choose 96 or 128 bits by selecting the CB1 bit in the configuration register. See Table 2-1 for more details.

2.3.1 COLUMN AND ROW DECODER LOGIC AND BIT COUNTER

The column and row decoders address the EEPROM array at the clock rate and generate a serial data stream for modulation. This data stream can be up to 128 bits in length. The size of the data stream is user programmable with CB1 and can be set to 96 or 128 bits. Data lengths of 48 and 64 bits are available by programming the data twice in the array, end-to-end.

The column and row decoders route the proper voltage to the array for programming and reading. In the programming modes, each individual bit is addressed serially from bit 1 to bit 128.

2.4 Examples of Configuration Settings

EXAMPLE 2-1: "48D" CONFIGURATION

The "48D" (hex) configuration is interpreted as follows:

CB12 … CB1

"48D" → 0100-1000-1101

Referring to Table 2-1, the "48D" configuration represents:

Blank (not programmed) Device
Anti-Collision
Modulation = PSK_1
PSK rate = rf/2
Data encoding = NRZ_L (direct)
Baud rate = rf/32 = MOD32
Memory size: 128 bits

EXAMPLE 2-2: "40A" CONFIGURATION

The "40A" (hex) configuration is interpreted as follows:

CB12 … CB1

"40A" → 0100-0000-1010

The MSB corresponds to CB12 and the LSB corresponds to CB1 of the configuration register. Therefore, we have:

CB12=0	CB11=1	CB10=0	CB9=0
CB8=0	CB7=0	CB6=0	CB5=0
CB4=1	CB3=0	CB2=1	CB1=0

Referring to Table 2-1, the "40A" configuration represents:

Not programmed device (blank), anti-collision, FSK protocol, NRZ_L (direct) encoding, MOD50 (baud rate = rf/50), 96 bits.

Note: The sample cards in the DV103002 kit are configured to "40A".

TABLE 2-1: CONFIGURATION REGISTER

CB12	CB11	CB10	CB9	CB8	CB7	CB6	CB5	CB4	CB3	CB2	CB1

MEMORY SIZE
CB1 = 1 128-bit user memory array
CB1 = 0 96-bit user memory array

BAUD RATE

CB2	CB3	CB4	Rate
0	0	0	MOD128
0	0	1	MOD100
0	1	0	MOD80
0	1	1	MOD32
1	0	0	MOD64
1	0	1	MOD50
1	1	0	MOD40
1	1	1	MOD16

SYNC WORD
CB5 = 0 (Always)

DATA ENCODING
CB6 = 0; CB7 = 0 NRZ_L (Direct)
CB6 = 0; CB7 = 1 Biphase_S (Differential)
CB6 = 1; CB7 = 0 Biphase_L (Manchester)
CB6 = 1; CB7 = 1 (Inverted Manchester)

MODULATION OPTIONS
CB8 = 0; CB9 = 0 FSK 0 = Fc/8, 1 = Fc/10
CB8 = 0; CB9 = 1 Direct (ASK)
CB8 = 1; CB9 = 0 PSK_1
(phase change on change of data)
CB8 = 1; CB9 = 1 PSK_2
(phase change at beginning of a one)

PSK RATE OPTION
CB10 = 1 Carrier/4
CB10 = 0 Carrier/2

(ANTI-COLLISION OPTION (Read only)
CB11 = 1 (Always)

MEMORY ARRAY LOCK BIT (Read only)
CB12 = 0 User memory array not locked (Blank)
CB12 = 1 User memory array is locked
(Programmed)

3.0 MODES OF OPERATION

The device has two basic modes of operation: Native Mode and Read Mode.

3.1 Native Mode

Every unprogrammed blank device (CB12 = 0) operates in Native mode, regardless of configuration register settings:

Baud rate = FCLK/128, FSK, NRZ_L (direct)

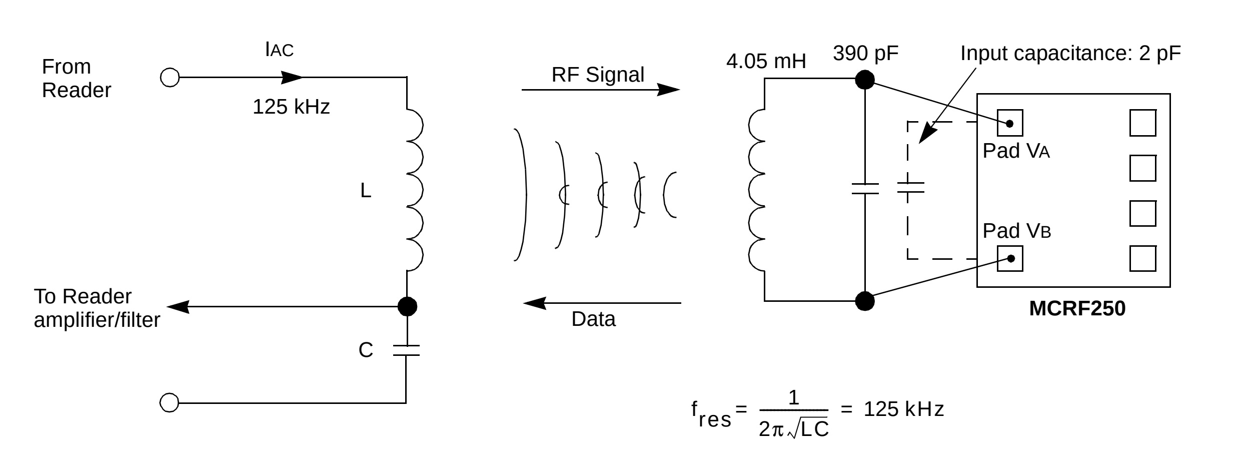
Once the user memory is programmed, the lock bit is set (CB12 = 1) which causes the MCRF250 to switch from Native mode to Communication mode defined by the configuration register.

Refer to Figure 4-1 for contactless programming sequence. Also see the *microID® 125 kHz RFID System Design Guide* (DS51115) for more information.

3.2 Read Mode

After the device is programmed (CB12 = 1), the device is operated in the Read-only mode. The device transmits its data according to the protocol in the configuration register.

FIGURE 3-1: TYPICAL APPLICATION CIRCUIT



4.0 CONTACTLESS PROGRAMMING

The contactless programming of the device is possible for a blank device (CB12 = 0) only, and is recommended for only low-volume, manual operation during development. In volume production, the MCRF250 is normally used as a factory programmed device only. The contactless programming timing sequence consists of:

a) RF Power-up signal.
b) Short gap (absence of RF field).
c) Verify signal (continuous RF signal).
d) Programming signal.
e) Device response with programmed data.

The blank device (CB12 = 0) understands the RF power-up followed by a gap as a blank checking command, and outputs 128 bits of FSK data with all ‘1’s after the short gap. To see this blank data (verify), the reader/programmer must provide a continuous RF signal for 128 bit-time. (The blank (unprogrammed) device has all ‘F’s in its memory array. Therefore, the blank data should be all ‘1’s in FSK format). Since the blank device operates at Default mode (MOD128), there are 128 RF cycles for each bit. Therefore, the time requirement to complete this verify is 128 bits x 128 RF cycles/bit x 8 use/cycles = 131.1 msec for 125 kHz signal.

As soon as the device completes the verify, it enters the programming mode. The reader/programmer must provide RF programming data right after the verify. In this programming mode, each bit lasts for 128 RF cycles. Refer to Figure 4-1 for the contactless programming sequence.

Customer must provide the following specific voltage for the programming:

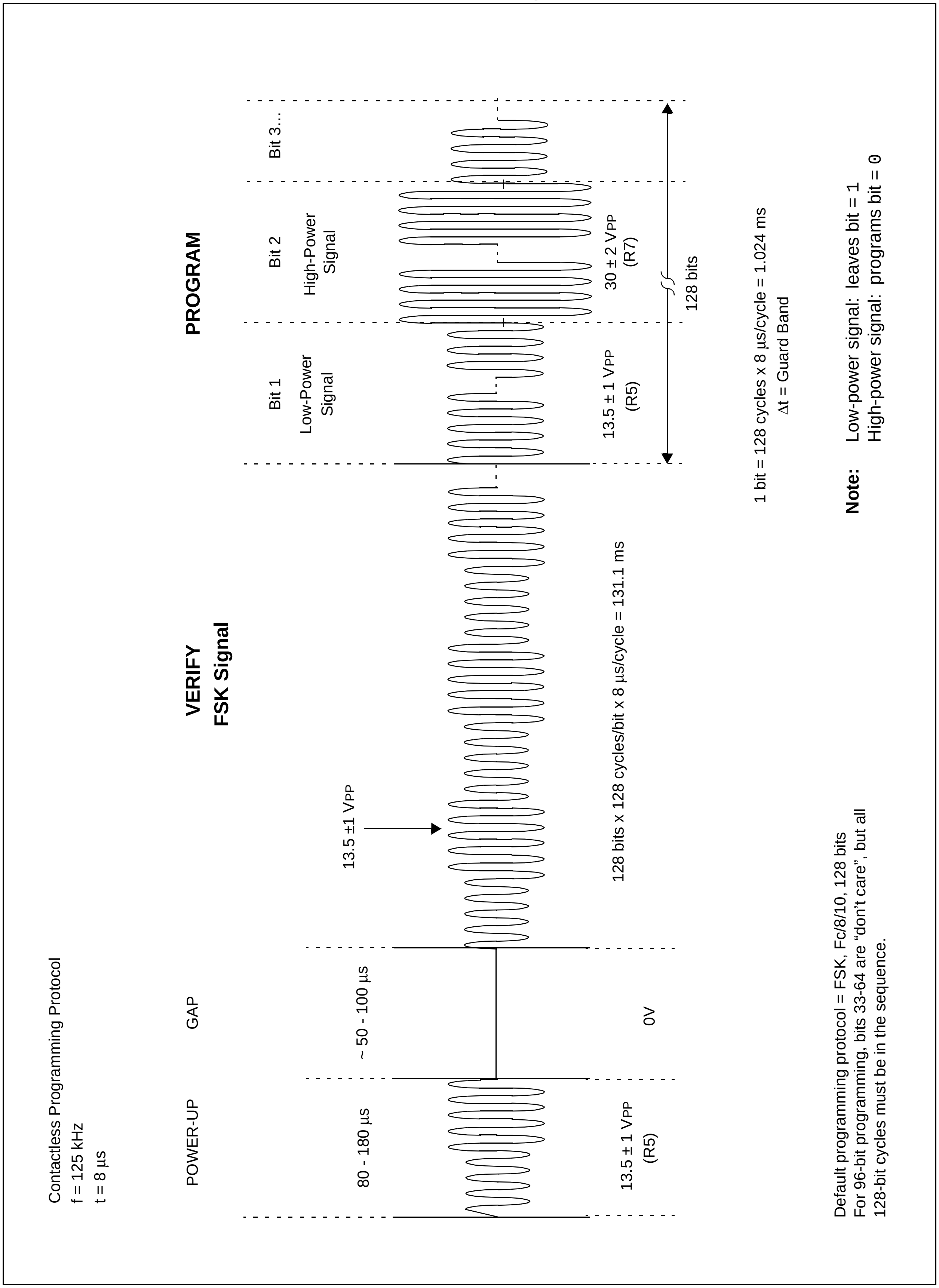
1. Power-up and verify signal = 13.5 V_{PP} ±1 V_{PP}
2. Programming voltage:
   - To program bit to ‘1’: 13.5 V_{PP} ±1 V_{PP}
   - To program bit to ‘0’: 30 V_{PP} ±2 V_{PP}

After the programming cycle, the device outputs programmed data (response). The reader/programmer can send the programming data repeatedly after the device response until the programming is successfully completed. The device locks the CB12 as soon as the programming mode (out of field) is exited and becomes a read-only device.

Once the device is programmed (CB12 = 1), the device outputs its data according to the configuration register.

The PG103001 (Contactless Programmer) is used for the programming of the device. The voltage level shown in Figure 4-1 is adjusted by R5 and R7 in the contactless programmer. Refer to the *MicroID® 125 kHz RFID System Design Guide* (DS51115) for more information.

FIGURE 4-1: CONTACTLESS PROGRAMMING SEQUENCE



5.0 ANTI-COLLISION

The anti-collision feature is enabled after the array lock bit (CB12) is set. This feature relies on internal random number oscillator/counter and special gap pulses (= turn off RF field) provided by a reader. Figure 5-1 shows the anti-collision flow chart.

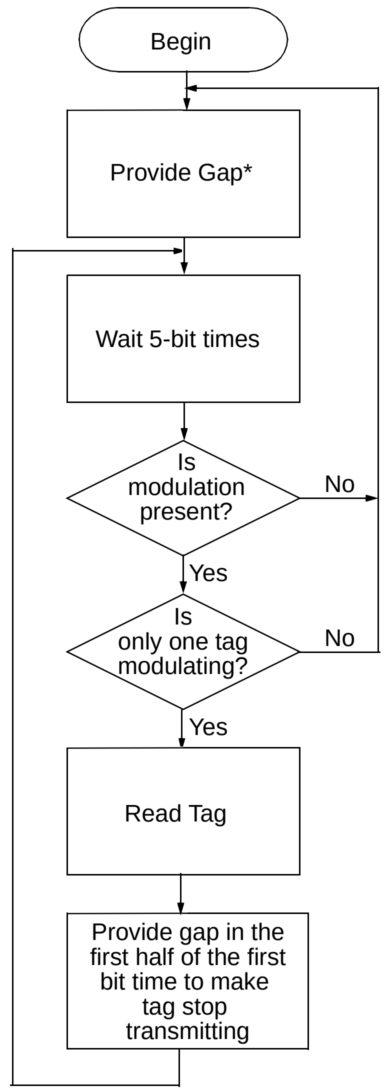
The MCRF250 works with the following anti-collision features:

1. The device does not output data until it sees the first gap. (no RF field for about 60 µsec.)
2. When the device sees the first gap, the internal random number oscillator starts clocking immediately after the gap.
3. At the same time, the internal random number counter starts counting the random number clocks.
4. The device waits for 5 bit times (about 5 msec. for MOD128 configuration).

   Example: 1 bit time=RF/128=1 msec for 128 kHz for MOD128
5. After the 5 bit times, the device sends data.
6. At this time, the random number counter is still running. If multiple tags in the field send data at the same time, the reader will see a data collision.
7. When the reader sees the data collision, it sends the second gap pulse. (no RF field for about 60 µsec.)
8. After the second gap pulse, there is a chance that the random number counter of each tag may have a different value due to a random variation in the oscillator’s starting time, etc.
9. After the second gap, the random number oscillator stops and the random number counter will decrement at each subsequent gap.
10. The device will transmit data when its random number counter reaches ‘0’.
11. The device repeats this sequence (as shown in the flow chart in Figure 5-1) according to the proper gap pulses provided by the reader.

Note:	Each device will output data in different time frames since each random number counter will arrive at ‘0’ at different times. As a result, the reader can receive clean data from a different tag in each time frame.

FIGURE 5-1: ANTI-COLLISION FLOW CHART



Note: *Gap = lack of RF carrier signal = 60 µs ± 20%.

6.0 MECHANICAL SPECIFICATIONS FOR DIE AND WAFER

FIGURE 6-1: DIE PLOT

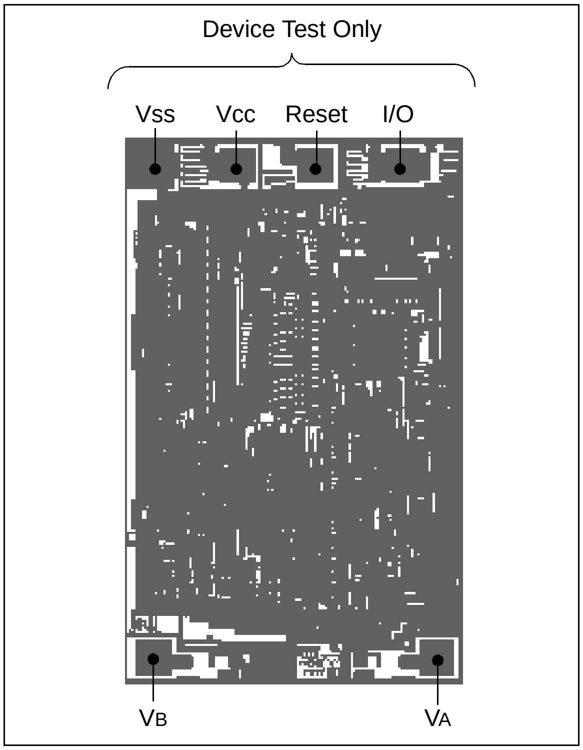


TABLE 6-1: PAD COORDINATES (µm)

Pad Name	Passivation Openings		Pad Center X	Pad Center Y
	Pad Width	Pad Height		
VA	90.0	90.0	427.50	-734.17
VB	90.0	90.0	-408.60	-734.17

Note 1: All coordinates are referenced from the center of the die.

2: Die size: 1.1215 mm x 1.7384 mm.
44.15 mils x 68.44 mils

TABLE 6-2: PAD FUNCTION TABLE

Name	Function
VA	Antenna Coil connections
VB	
VSS	For device test only Do Not Connect to Antenna
VCC	
Reset	
I/O	

TABLE 6-3: DIE MECHANICAL DIMENSIONS

Specifications	Min	Typ	Max	Unit	Comments
Bond pad opening	— —	3.5 x 3.5 89 x 89	— —	mil µm	**Note 1, Note 2**
Die backgrind thickness	— —	7 177.8	— —	mil µm	Sawed 6” wafer on frame (option = WF) **Note 3**
	— —	11 279.4	— —	mil µm	Unsawed wafer (option = W) **Note 3**
Die backgrind thickness tolerance	— —	— —	±1 ±25.4	mil µm	
Die passivation thickness (multilayer)	—	0.9050	—	µm	**Note 4**
Die Size: Die size X*Y before saw (step size) Die size X*Y after saw	 — —	 44.15 x 68.44 42.58 x 66.87	 — —	 mil mil	 — —

Note 1: The bond pad size is that of the passivation opening. The metal overlaps the bond pad passivation by at least 0.1 mil.

2: Metal Pad Composition is 98.5% Aluminum with 1% Si and 0.5% Cu.

3: As the die thickness decreases, susceptibility to cracking increases. It is recommended that the die be as thick as the application will allow..

4: The Die Passivation thickness can vary by device depending on the mask set used:

- Layer 1: Oxide (undopped oxide, 0.135 µm)
- Layer 2: PSG (dopped oxide, 0.43 µm)
- Layer 3: Oxynitride (top layer, 0.34 µm)

5: The conversion rate is 25.4 µm/mil.

Notice: Extreme care is urged in the handling and assembly of die products since they are susceptible to mechanical and electrostatic damage.

TABLE 6-4: WAFER MECHANICAL SPECIFICATIONS

Specifications	Min	Typ	Max	Unit	Comments
Wafer Diameter	—	8	—	inch	150 mm
Die separation line width	—	80	—	µm	
Dice per wafer	—	14,000	—	die	
Batch size	—	24	—	wafer	

7.0 FAILED DIE IDENTIFICATION

Every die on the wafer is electrically tested according to the data sheet specifications and visually inspected to detect any mechanical damage such as mechanical cracks and scratches.

Any failed die in the test or visual inspection is identified by black colored inking. Therefore, any die covered with black ink should not be used.

The ink dot specification:

- Ink dot size: minimum 20 µm x 20 µm
- Position: central third of die
- Color: black

8.0 WAFER DELIVERY DOCUMENTATION

Each wafer container is marked with the following information:

- Microchip Technology Inc. MP Code
- Lot Number
- Total number of wafer in the container
- Total number of good dice in the container
- Average die per wafer (DPW)
- Scribe number of wafer with number of good dice.

9.0 NOTICE ON DIE AND WAFER HANDLING

The device is very susceptible to Electrostatic Discharge (ESD). ESD can cause critical damage to the device. Special attention is needed during the handling process.

Any ultraviolet (UV) light can erase the memory cell contents of an unpackaged device. Fluorescent lights and sun light can also erase the memory cell although it takes more time than UV lamps. Therefore, keep any unpackaged devices out of UV light and also avoid direct exposure from strong fluorescent lights and sun light.

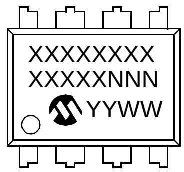
Certain integrated circuit (IC) manufacturing, chip-on-board (COB) and tag assembly operations may use UV light. Operations such as backgrind, de-tape, certain cleaning operations, epoxy or glue cure should be done without exposing the die surface to UV light.

Using x-ray for die inspection will not harm the die, nor erase memory cell contents.

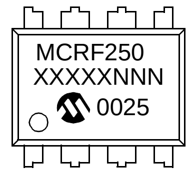
10.0 PACKAGING INFORMATION

10.1 Package Marking Information

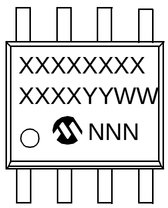
8-Lead PDIP (300 mil)



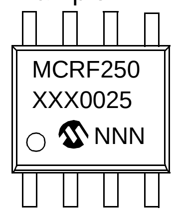
Example:



8-Lead SOIC (150 mil)



Example:

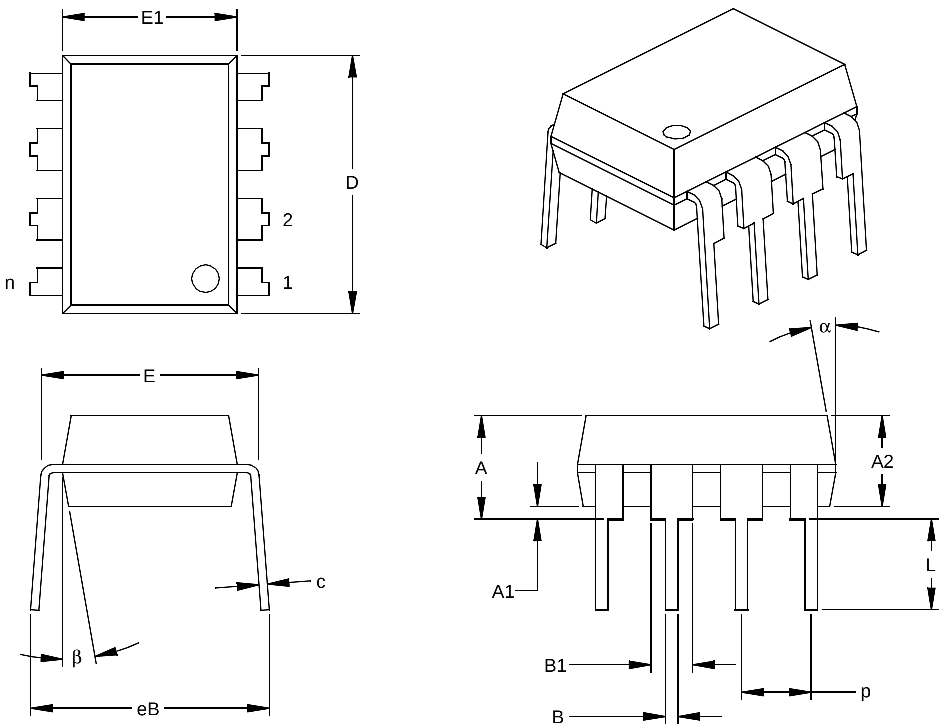


Legend:	XX...X	Customer specific information*
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard device marking consists of Microchip part number, year code, week code, and traceability code.

8-Lead Plastic Dual In-line (P) – 300 mil (PDIP)

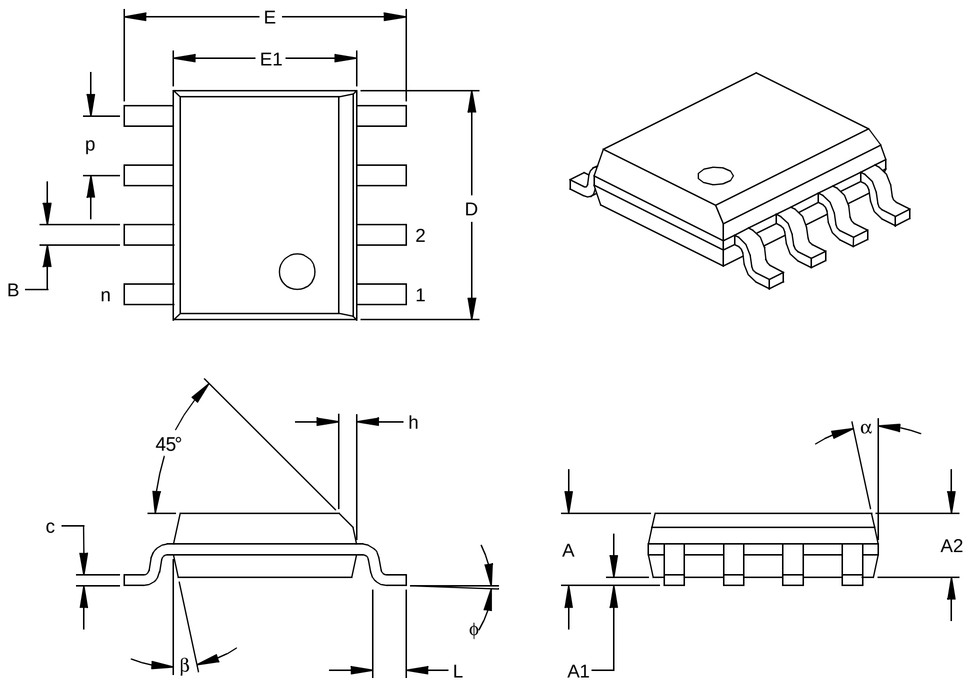


UNITS		INCHES*			MILLIMETERS		
DIMENSION LIMITS		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.360	.373	.385	9.14	9.46	9.78
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-018

8-Lead Plastic Small Outline (SN) – Narrow, 150 mil (SOIC)

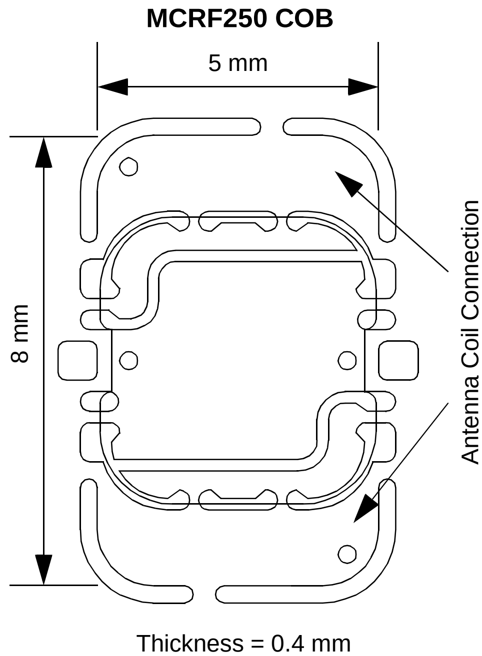


UNITS		INCHES*			MILLIMETERS		
DIMENSION LIMITS		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.050			1.27	
Overall Height	A	.053	.061	.069	1.35	1.55	1.75
Molded Package Thickness	A2	.052	.056	.061	1.32	1.42	1.55
Standoff §	A1	.004	.007	.010	.10	.18	.25
Overall Width	E	.228	.237	.244	5.79	6.02	6.20
Molded Package Width	E1	.146	.154	.157	3.71	3.91	3.99
Overall Length	D	.189	.193	.197	4.80	4.90	5.00
Chamfer Distance	h	.010	.015	.020	.25	.38	.51
Foot Length	L	.019	.025	.030	.48	.62	.76
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.008	.009	.010	.20	.23	.25
Lead Width	B	.013	.017	.020	.33	.42	.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-012
Drawing No. C04-057

1M/3M COB (IOA2)



PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	-X	/XXX	XXX
Device	Temperature Range	Package	Configuration/SQTP Code

Device:	MCRF250 = 125 kHz Anti-collision MicroID tag, 96/128-bit
Temperature Range:	I = -40°C to +85°C
Package:	WF = Sawed wafer on frame (7 mil backgrind) W = Wafer (11 mil backgrind) S = Dice in waffle pack P = Plastic PDIP (300 mil Body) 8-lead SN = Plastic SOIC (150 mil Body) 8-lead
Configuration:	Three-digit hex value to be programmed into the configuration register. Three hex characters correspond to 12 binary bits. These bits are programmed into the configuration register MSB first (CB12, CB11...CB1). Refer to example.
SQTP Code:	An assigned, customer 3-digit code used for tracking and controlling production and customer data files for factory programming. In this case the configuration code is not shown in the part number, but is captured in the SQTP documention.

Examples:

a) MCRF250-I/W40A = 125 kHz, industrial temperature, wafer package, contactlessly programmable, 96 bit, FSK Fc/8 Fc/10, direct encoded, Fc/50 data return rate tag.

b) MCRF250-I/WFQ23 = 125 kHz, industrial temperature, wafer sawn and mounted on frame, factory programmed.

The configuration register is:

CB12	CB11	CB10	CB9	CB8	CB7	CB6	CB5	CB4	CB3	CB2	CB1
0	1	0	0	0	0	0	0	1	0	1	0

Sales and Support

Data Sheets
Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 792-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System
Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

NOTES:

TB023

Serialized Quick Turn Programming^{SM} (SQTP^{SM})

INTRODUCTION

Factory programming of MCRF200, MCRF202 or MCRF250 is performed by Microchip Technology Inc. upon customer request. The customer can choose any ID code suitable to the application subject to a minimum order quantity.

DEFINITIONS

First, the configuration register code must be determined in order to configure the following operation options of the MCRF200, MCRF202 and MCRF250 (refer to individual data sheets DS21219, DS21308 and DS21267 respectively):

- Bit rate — Defined as clocks per bit (e.g., Fc/16, Fc/32, Fc/40, Fc/50, Fc/64, Fc/80, Fc/100, and Fc/128)
- Modulation — FSK, PSK1, PSK2, ASK Direct
- Encoding — NRZ_L (Direct), Biphase_L (Manchester), Differential Biphase_S
- Code length — 32, 48, 64, 96, and 128 bits

Second, the ID codes and series numbers must be supplied by the customer on floppy disk, CD or via email. The codes should conform to the SQTP format below:

FILE SPECIFICATION

SQTP codes supplied to Microchip must comply with the following format:

The ID code file is a plain ASCII text file on floppy disk, CD or email (no headers).

Please provide zipped (.zip) files, no self-extracting (.exe) files.

The code files are used in alphabetical order of their file names (including letters and numbers).

Used (i.e., programmed) code files are discarded by Microchip after use.

Each line of the code file must contain one ID code for one IC.

The code is in hexadecimal format.

The code line is exactly as long as the selected code length (e.g., for a code length = 64, the ID code = 16 hex characters = 64-bit number).

Each line must end with a carriage return.

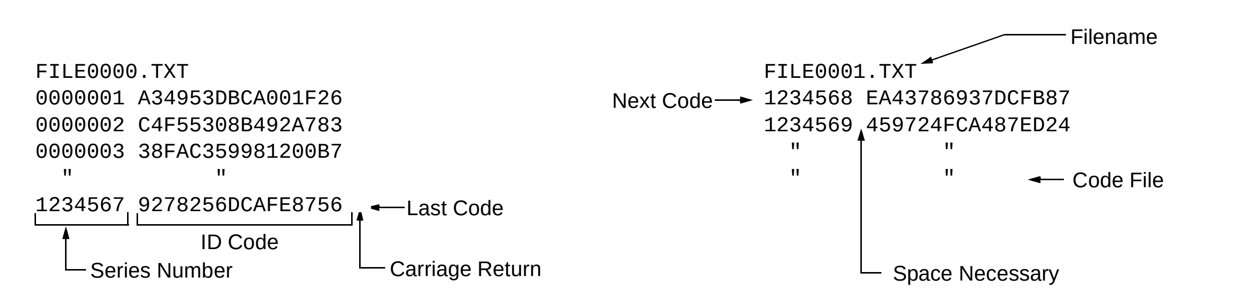
Each hexadecimal ID code must be preceded by a decimal series number.

Series number and ID code must be separated by a space.

The series number must be unique and ascending to avoid double programming.

The series numbers of consecutive files must also increment serially for proper linking. The series number may contain five, six or seven digits.

FIGURE 1: EXAMPLE OF TWO CODE FILES, CODE LENGTH = 64 BITS



NOTES:

AN710

Antenna Circuit Design for RFID Applications

Author: Youbok Lee, Ph.D.
Microchip Technology Inc.

INTRODUCTION

Passive RFID tags utilize an induced antenna coil voltage for operation. This induced AC voltage is rectified to provide a voltage source for the device. As the DC voltage reaches a certain level, the device starts operating. By providing an energizing RF signal, a reader can communicate with a remotely located device that has no external power source such as a battery. Since the energizing and communication between the reader and tag is accomplished through antenna coils, it is important that the device must be equipped with a proper antenna circuit for successful RFID applications.

An RF signal can be radiated effectively if the linear dimension of the antenna is comparable with the wavelength of the operating frequency. However, the wavelength at 13.56 MHz is 22.12 meters. Therefore, it is difficult to form a true antenna for most RFID applications. Alternatively, a small loop antenna circuit that is resonating at the frequency is used. A current flowing into the coil radiates a near-field magnetic field that falls off with r^{-3}. This type of antenna is called a *magnetic dipole antenna.*

For 13.56 MHz passive tag applications, a few microhenries of inductance and a few hundred pF of resonant capacitor are typically used. The voltage transfer between the reader and tag coils is accomplished through inductive coupling between the two coils. As in a typical transformer, where a voltage in the primary coil transfers to the secondary coil, the voltage in the reader antenna coil is transferred to the tag antenna coil and vice versa. The efficiency of the voltage transfer can be increased significantly with high Q circuits.

This section is written for RF coil designers and RFID system engineers. It reviews basic electromagnetic theories on antenna coils, a procedure for coil design, calculation and measurement of inductance, an antenna tuning method, and read range in RFID applications.

REVIEW OF A BASIC THEORY FOR RFID ANTENNA DESIGN

Current and Magnetic Fields

Ampere's law states that current flowing in a conductor produces a magnetic field around the conductor. The magnetic field produced by a current element, as shown in Figure 1, on a round conductor (wire) with a finite length is given by:

EQUATION 1:

$$B_\phi = \frac{\mu_o I}{4\pi r}(\cos\alpha_2 - \cos\alpha_1) \qquad (\text{Weber}/m^2)$$

where:

I = current

r = distance from the center of wire

μ_0 = permeability of free space and given as $4\pi \times 10^{-7}$ (Henry/meter)

In a special case with an infinitely long wire where:

α_1 = -180°

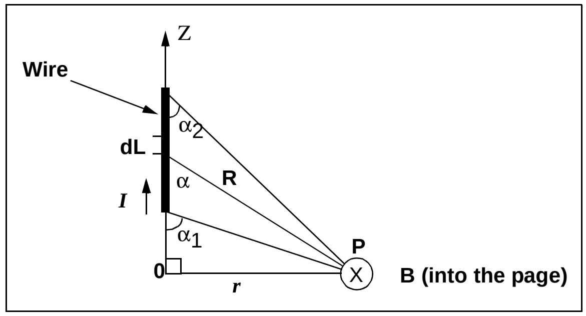
α_2 = 0°

Equation 1 can be rewritten as:

EQUATION 2:

$$B_\phi = \frac{\mu_o I}{2\pi r} \qquad (\text{Weber}/m^2)$$

FIGURE 1: CALCULATION OF MAGNETIC FIELD B AT LOCATION P DUE TO CURRENT I ON A STRAIGHT CONDUCTING WIRE



The magnetic field produced by a circular loop antenna is given by:

EQUATION 3:

$$B_z = \frac{\mu_o I N a^2}{2(a^2 + r^2)^{3/2}}$$

$$= \frac{\mu_o I N a^2}{2}\left(\frac{1}{r^3}\right) \quad \text{for} \quad r^2 >> a^2$$

where

- I = current
- a = radius of loop
- r = distance from the center of loop
- μ_0 = permeability of free space and given as $4\pi \times 10^{-7}$ (Henry/meter)

The above equation indicates that the magnetic field strength decays with $1/r^3$. A graphical demonstration is shown in Figure 3. It has maximum amplitude in the plane of the loop and directly proportional to both the current and the number of turns, N.

Equation 3 is often used to calculate the ampere-turn requirement for read range. A few examples that calculate the ampere-turns and the field intensity necessary to power the tag will be given in the following sections.

FIGURE 2: CALCULATION OF MAGNETIC FIELD B AT LOCATION P DUE TO CURRENT I ON THE LOOP

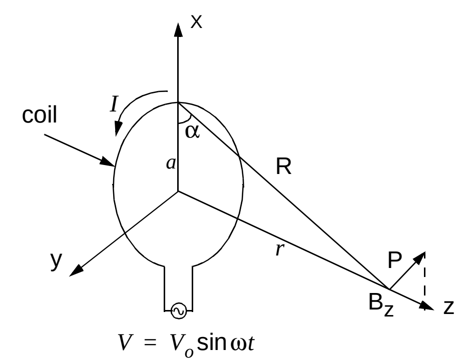
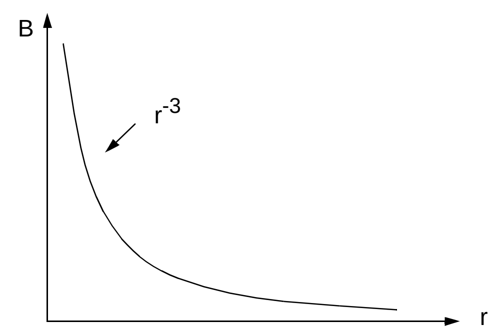


FIGURE 3: DECAYING OF THE MAGNETIC FIELD B VS. DISTANCE *r*



INDUCED VOLTAGE IN AN ANTENNA COIL

Faraday's law states that a time-varying magnetic field through a surface bounded by a closed path induces a voltage around the loop.

Figure 4 shows a simple geometry of an RFID application. When the tag and reader antennas are in close proximity, the time-varying magnetic field B that is produced by a reader antenna coil induces a voltage (called electromotive force or simply EMF) in the closed tag antenna coil. The induced voltage in the coil causes a flow of current on the coil. This is called Faraday's law. The induced voltage on the tag antenna coil is equal to the time rate of change of the magnetic flux Ψ.

EQUATION 4:

$$V = -N\frac{d\Psi}{dt}$$

where:

N = number of turns in the antenna coil

Ψ = magnetic flux through each turn

The negative sign shows that the induced voltage acts in such a way as to oppose the magnetic flux producing it. This is known as Lenz's law and it emphasizes the fact that the direction of current flow in the circuit is such that the induced magnetic field produced by the induced current will oppose the original magnetic field.

The magnetic flux Ψ in Equation 4 is the total magnetic field B that is passing through the entire surface of the antenna coil, and found by:

EQUATION 5:

$$\psi = \int B \cdot dS$$

where:

B = magnetic field given in Equation 2

S = surface area of the coil

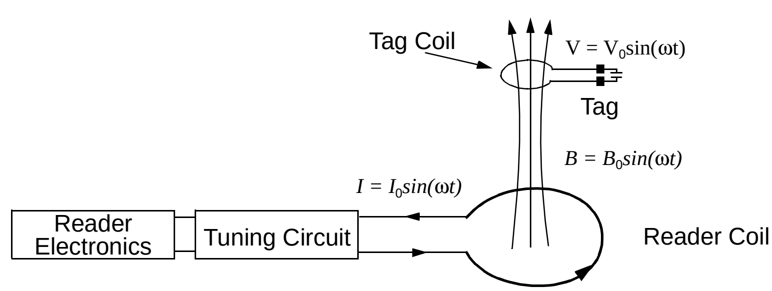
$\bullet$ = inner product (*cosine angle between two vectors*) of vectors B and surface area S

Note:	Both magnetic field B and surface S are vector quantities.

The presentation of inner product of two vectors in Equation 5 suggests that the total magnetic flux ψ that is passing through the antenna coil is affected by an orientation of the antenna coils. The inner product of two vectors becomes minimized when the cosine angle between the two are 90 degrees, or the two (B field and the surface of coil) are perpendicular to each other and maximized when the cosine angle is 0 degrees.

The maximum magnetic flux that is passing through the tag coil is obtained when the two coils (reader coil and tag coil) are placed in parallel with respect to each other. This condition results in maximum induced voltage in the tag coil and also maximum read range. The inner product expression in Equation 5 also can be expressed in terms of a mutual coupling between the reader and tag coils. The mutual coupling between the two coils is maximized in the above condition.

FIGURE 4: A BASIC CONFIGURATION OF READER AND TAG ANTENNAS IN RFID APPLICATIONS



Using Equations 3 and 5, Equation 4 can be rewritten as:

EQUATION 6:

$$V = -N_2\frac{d\Psi_{21}}{dt} = -N_2\frac{d}{dt}\left(\int B\cdot dS\right)$$
$$= -N_2\frac{d}{dt}\left[\int\frac{\mu_o i_1 N_1 a^2}{2(a^2+r^2)^{3/2}}\cdot dS\right]$$
$$= -\left[\frac{\mu_o N_1 N_2 a^2(\pi b^2)}{2(a^2+r^2)^{3/2}}\right]\frac{di_1}{dt}$$
$$= -M\frac{di_1}{dt}$$

where:

- V = voltage in the tag coil
- i_1 = current on the reader coil
- a = radius of the reader coil
- b = radius of tag coil
- r = distance between the two coils
- M = mutual inductance between the tag and reader coils, and given by:

EQUATION 7:

$$M = \left[\frac{\mu_o \pi N_1 N_2 (ab)^2}{2(a^2+r^2)^{3/2}}\right]$$

The above equation is equivalent to a voltage transformation in typical transformer applications. The current flow in the primary coil produces a magnetic flux that causes a voltage induction at the secondary coil.

As shown in Equation 6, the tag coil voltage is largely dependent on the mutual inductance between the two coils. The mutual inductance is a function of coil geometry and the spacing between them. The induced voltage in the tag coil decreases with r^{-3}. Therefore, the read range also decreases in the same way.

From Equations 4 and 5, a generalized expression for induced voltage V_o in a tuned loop coil is given by:

EQUATION 8:

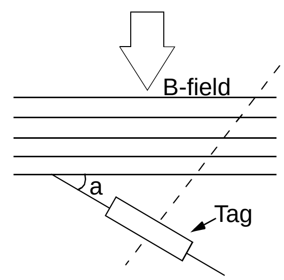
$$V_0 = 2\pi f N S Q B_o \cos\alpha$$

where:

- f = frequency of the arrival signal
- N = number of turns of coil in the loop
- S = area of the loop in square meters (m^2)
- Q = quality factor of circuit
- B_o = strength of the arrival signal
- α = angle of arrival of the signal

In the above equation, the quality factor Q is a measure of the selectivity of the frequency of the interest. The Q will be defined in Equations 43 through 59.

FIGURE 5: ORIENTATION DEPENDENCY OF THE TAG ANTENNA



The induced voltage developed across the loop antenna coil is a function of the angle of the arrival signal. The induced voltage is maximized when the antenna coil is placed in parallel with the incoming signal where $\alpha = 0$.

EXAMPLE 1: CALCULATION OF B-FIELD IN A TAG COIL

The MCRF355 device turns on when the antenna coil develops 4 VPP across it. This voltage is rectified and the device starts to operate when it reaches 2.4 VDC. The B-field to induce a 4 VPP coil voltage with an ISO standard 7810 card size (85.6 x 54 x 0.76 mm) is calculated from the coil voltage equation using Equation 8.

EQUATION 9:

$$V_o = 2\pi fNSQB_o\cos\alpha = 4$$

and

$$B_o = \frac{4/(\sqrt{2})}{2\pi fNSQ\cos\alpha} = 0.0449 \qquad (\mu wbm^{-2})$$

where the following parameters are used in the above calculation:

Tag coil size	=	(85.6 x 54) mm^2 (ISO card size) = 0.0046224 m^2
Frequency	=	13.56 MHz
Number of turns	=	4
Q of tag antenna coil	=	40
AC coil voltage to turn on the tag	=	4 VPP
cosα	=	1 (normal direction, α = 0).

EXAMPLE 2: NUMBER OF TURNS AND CURRENT (AMPERE-TURNS)

Assuming that the reader should provide a read range of 15 inches (38.1 cm) for the tag given in the previous example, the current and number of turns of a reader antenna coil is calculated from Equation 3:

EQUATION 10:

$$(NI)_{rms} = \frac{2B_z(a^2+r^2)^{3/2}}{\mu a^2}$$

$$= \frac{2(0.0449\times10^{-6})(0.1^2+(0.38)^2)^{3/2}}{(4\pi\times10^{-7})(0.1^2)}$$

$$= 0.43(\text{ampere - turns})$$

The above result indicates that it needs a 430 mA for 1 turn coil, and 215 mA for 2-turn coil.

EXAMPLE 3: OPTIMUM COIL DIAMETER OF THE READER COIL

An optimum coil diameter that requires the minimum number of ampere-turns for a particular read range can be found from Equation 3 such as:

EQUATION 11:

$$NI = K\frac{(a^2+r^2)^{\frac{3}{2}}}{a^2}$$

where: $K = \frac{2B_z}{\mu_o}$

By taking derivative with respect to the radius a,

$$\frac{d(NI)}{da} = K\frac{3/2(a^2+r^2)^{1/2}(2a^3)-2a(a^2+r^2)^{3/2}}{a^4}$$

$$= K\frac{(a^2-2r^2)(a^2+r^2)^{1/2}}{a^3}$$

The above equation becomes minimized when:

The above result shows a relationship between the read range versus optimum coil diameter. The optimum coil diameter is found as:

EQUATION 12:

$$a = \sqrt{2}r$$

where:

a = radius of coil

r = read range.

The result indicates that the optimum loop radius, a, is 1.414 times the demanded read range r.

WIRE TYPES AND OHMIC LOSSES

DC Resistance of Conductor and Wire Types

The diameter of electrical wire is expressed as the American Wire Gauge (AWG) number. The gauge number is inversely proportional to diameter, and the diameter is roughly doubled every six wire gauges. The wire with a smaller diameter has a higher DC resistance. The DC resistance for a conductor with a uniform cross-sectional area is found by:

EQUATION 13: DC Resistance of Wire

$$R_{DC} = \frac{l}{\sigma S} = \frac{l}{\sigma \pi a^2} \qquad (\Omega)$$

where:

l = total length of the wire
σ = conductivity of the wire (mho/m)
S = cross-sectional area = πr^2
a = radius of wire

For a The resistance must be kept small as possible for higher Q of antenna circuit. For this reason, a larger diameter coil as possible must be chosen for the RFID circuit. Table 5 shows the diameter for bare and enamel-coated wires, and DC resistance.

AC Resistance of Conductor

At DC, charge carriers are evenly distributed through the entire cross section of a wire. As the frequency increases, the magnetic field is increased at the center of the inductor. Therefore, the reactance near the center of the wire increases. This results in higher impedance to the current density in the region. Therefore, the charge moves away from the center of the wire and towards the edge of the wire. As a result, the current density decreases in the center of the wire and increases near the edge of the wire. This is called a *skin effect*. The depth into the conductor at which the current density falls to 1/e, or 37% (= 0.3679) of its value along the surface, is known as the *skin depth* and is a function of the frequency and the permeability and conductivity of the medium. The net result of skin effect is an effective decrease in the cross sectional area of the conductor. Therefore, a net increase in the AC resistance of the wire. The skin depth is given by:

EQUATION 14:

$$\delta = \frac{1}{\sqrt{\pi f \mu \sigma}}$$

where:

f = frequency
μ = permeability (F/m) = $\mu_o \mu_r$
μ_o = Permeability of air = $4 \pi \times 10^{-7}$ (h/m)
μ_r = 1 for Copper, Aluminum, Gold, etc
= 4000 for pure Iron
σ = Conductivity of the material (mho/m)
= 5.8×10^7 (mho/m) for Copper
= 3.82×10^7 (mho/m) for Aluminum
= 4.1×10^7 (mho/m) for Gold
= 6.1×10^7 (mho/m) for Silver
= 1.5×10^7 (mho/m) for Brass

EXAMPLE 4:

The skin depth for a copper wire at 13.56 MHz and 125 kHz can be calculated as:

EQUATION 15:

$$\delta = \frac{1}{\sqrt{\pi f (4\pi \times 10^{-7})(5.8 \times 10^{7})}}$$

$$= \frac{0.0661}{\sqrt{f}} \qquad (m)$$

$$= 0.018 \,(mm) \qquad \text{for 13.56 MHz}$$

$$= 0.187 \,(mm) \qquad \text{for 125 kHz}$$

As shown in Example 4, 63% of the RF current flowing in a copper wire will flow within a distance of 0.018 mm of the outer edge of wire for 13.56 MHz and 0.187 mm for 125 kHz.

The wire resistance increases with frequency, and the resistance due to the skin depth is called an AC resistance. An approximated formula for the AC resistance is given by:

EQUATION 16:

$$R_{ac} = \frac{l}{\sigma A_{active}} \approx \frac{l}{2\pi a \delta \sigma} \qquad (\Omega)$$

$$= \frac{l}{2a}\sqrt{\frac{f\mu}{\pi\sigma}} \qquad (\Omega)$$

$$= (R_{dc})\ \frac{a}{2\delta} \qquad (\Omega)$$

where the skin depth area on the conductor is,

$$A_{active} \approx 2\pi a \delta$$

The AC resistance increases with the square root of the operating frequency.

For the conductor etched on dielectric, substrate is given by:

EQUATION 17:

$$R_{ac} = \frac{l}{\sigma(w+t)\delta} \quad = \frac{l}{(w+t)}\sqrt{\frac{\pi f \mu}{\sigma}} \qquad (\Omega)$$

where w is the width and t is the thickness of the conductor.

Resistance of Conductor with Low Frequency Approximation

When the skin depth is almost comparable to the radius of conductor, the resistance can be obtained with a low frequency approximation[5]:

EQUATION 18:

$$R_{low\ freq} \approx \frac{l}{\sigma\pi a^2}\left[1 + \frac{1}{48}\left(\frac{a}{\delta}\right)^2\right] \qquad (\Omega)$$

The first term of the above equation is the DC resistance, and the second term represents the AC resistance.

TABLE 5: AWG WIRE CHART

Wire Size (AWG)	Dia. in Mils (bare)	Dia. in Mils (coated)	Ohms/ 1000 ft.
1	289.3	—	0.126
2	287.6	—	0.156
3	229.4	—	0.197
4	204.3	—	0.249
5	181.9	—	0.313
6	162.0	—	0.395
7	166.3	—	0.498
8	128.5	131.6	0.628
9	114.4	116.3	0.793
10	101.9	106.2	0.999
11	90.7	93.5	1.26
12	80.8	83.3	1.59
13	72.0	74.1	2.00
14	64.1	66.7	2.52
15	57.1	59.5	3.18
16	50.8	52.9	4.02
17	45.3	47.2	5.05
18	40.3	42.4	6.39
19	35.9	37.9	8.05
20	32.0	34.0	10.1
21	28.5	30.2	12.8
22	25.3	28.0	16.2
23	22.6	24.2	20.3
24	20.1	21.6	25.7
25	17.9	19.3	32.4
26	15.9	17.2	41.0
27	14.2	15.4	51.4
28	12.6	13.8	65.3
29	11.3	12.3	81.2
30	10.0	11.0	106.0
31	8.9	9.9	131
32	8.0	8.8	162
33	7.1	7.9	206
34	6.3	7.0	261
35	5.6	6.3	331
36	5.0	5.7	415
37	4.5	5.1	512
38	4.0	4.5	648
39	3.5	4.0	847
40	3.1	3.5	1080
41	2.8	3.1	1320
42	2.5	2.8	1660
43	2.2	2.5	2140
44	2.0	2.3	2590
45	1.76	1.9	3350
46	1.57	1.7	4210
47	1.40	1.6	5290
48	1.24	1.4	6750
49	1.11	1.3	8420
50	0.99	1.1	10600

Note: mil = 2.54 x 10^{-3} cm

INDUCTANCE OF VARIOUS ANTENNA COILS

An electric current element that flows through a conductor produces a magnetic field. This time-varying magnetic field is capable of producing a flow of current through another conductor – this is called *inductance*. The inductance L depends on the physical characteristics of the conductor. A coil has more inductance than a straight wire of the same material, and a coil with more turns has more inductance than a coil with fewer turns. The inductance L of inductor is defined as the ratio of the total magnetic flux linkage to the current I through the inductor:

EQUATION 19:

$$L = \frac{N\Psi}{I} \quad \text{(Henry)}$$

where:

N = number of turns
I = current
Ψ = the magnetic flux

For a coil with multiple turns, the inductance is greater as the spacing between turns becomes smaller. Therefore, the tag antenna coil that has to be formed in a limited space often needs a multilayer winding to reduce the number of turns.

Calculation of Inductance

Inductance of the coil can be calculated in many different ways. Some are readily available from references[1-7]. It must be remembered that for RF coils the actual resulting inductance may differ from the calculated true result because of distributed capacitance. For that reason, inductance calculations are generally used only for a starting point in the final design.

INDUCTANCE OF A STRAIGHT WOUND WIRE

The inductance of a straight wound wire shown in Figure 1 is given by:

EQUATION 20:

$$L = 0.002l\left[\log_e\frac{2l}{a} - \frac{3}{4}\right] \quad (\mu H)$$

where:

l and a = length and radius of wire in cm, respectively.

EXAMPLE 6: INDUCTANCE CALCULATION FOR A STRAIGHT WIRE:

The inductance of a wire with 10 feet (304.8cm) long and 2 mm in diameter is calculated as follows:

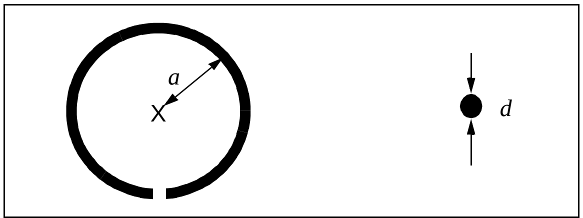
EQUATION 21:

$$L = 0.002(304.8)\left[\ln\left(\frac{2(304.8)}{0.1}\right) - \frac{3}{4}\right]$$
$$= 0.60967(7.965)$$
$$= 4.855(\mu H)$$

INDUCTANCE OF A SINGLE TURN CIRCULAR COIL

The inductance of a single turn circular coil shown in Figure 6 can be calculated by:

FIGURE 6: A CIRCULAR COIL WITH SINGLE TURN



EQUATION 22:

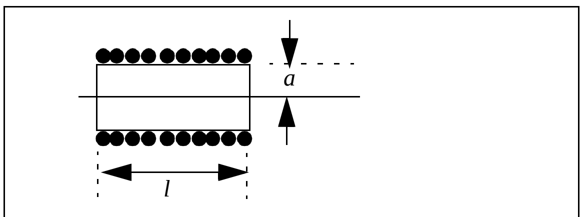
$$L = 0.01257(a)\left[2.303\log_{10}\left(\frac{16a}{d} - 2\right)\right] \qquad (\mu H)$$

where:

a = mean radius of loop in (cm)
d = diameter of wire in (cm)

INDUCTANCE OF AN N-TURN SINGLE LAYER CIRCULAR COIL

FIGURE 7: A CIRCULAR COIL WITH SINGLE TURN



EQUATION 23:

$$L = \frac{(aN)^2}{22.9a + 25.4l} \qquad (\mu H)$$

where:

N = number of turns
l = length in cm
a = the radius of coil in cm

INDUCTANCE OF N-TURN MULTILAYER CIRCULAR COIL

FIGURE 8: N-TURN MULTILAYER CIRCULAR COIL

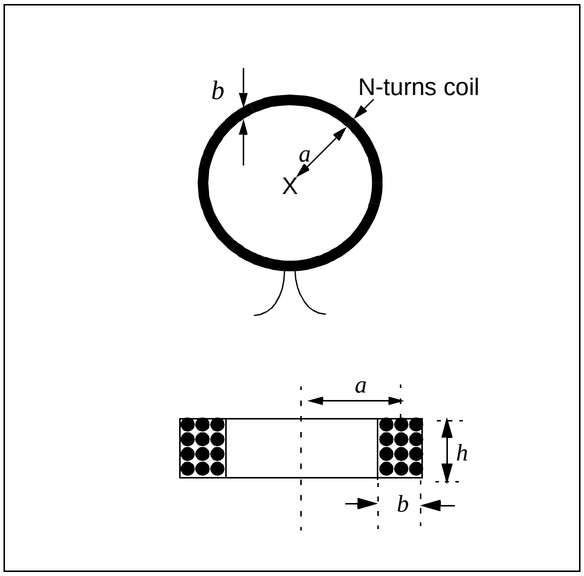


Figure 8 shows an N-turn inductor of circular coil with multilayer. Its inductance is calculated by:

EQUATION 24:

$$L = \frac{0.31(aN)^2}{6a + 9h + 10b} \qquad (\mu H)$$

where:

a = average radius of the coil in cm
N = number of turns
b = winding thickness in cm
h = winding height in cm

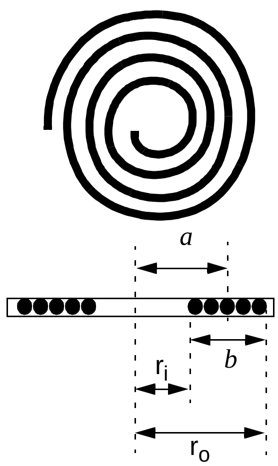
INDUCTANCE OF SPIRAL WOUND COIL WITH SINGLE LAYER

The inductance of a spiral inductor is calculated by:

EQUATION 25:

$$L = \frac{(0.3937)(aN)^2}{8a + 11b} \qquad (\mu H)$$

FIGURE 9: A SPIRAL COIL



where:

- a = $(r_i + r_o)/2$
- b = $r_o - r_i$
- r_i = Inner radius of the spiral
- r_o = Outer radius of the spiral

Note: All dimensions are in cm

INDUCTANCE OF N-TURN SQUARE LOOP COIL WITH MULTILAYER

Inductance of a multilayer square loop coil is calculated by:

EQUATION 26:

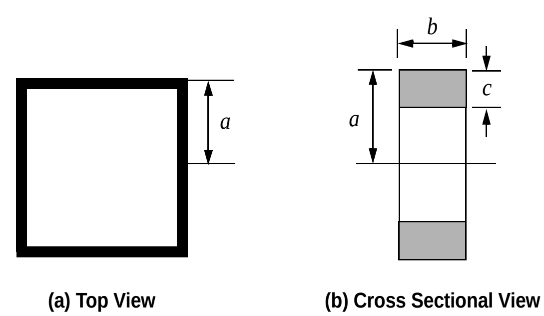
$$L = 0.008aN^2\left\{2.303\log_{10}\left(\frac{a}{b+c}\right) + 0.2235\frac{b+c}{a} + 0.726\right\}(\mu H)$$

where:

- N = number of turns
- a = side of square measured to the center of the rectangular cross section of winding
- b = winding length
- c = winding depth as shown in Figure 10

Note: All dimensions are in cm

FIGURE 10: N-TURN SQUARE LOOP COIL WITH MULTILAYER



INDUCTANCE OF N-TURN RECTANGULAR COIL WITH MULTILAYER

Inductance of a multilayer rectangular loop coil is calculated by:

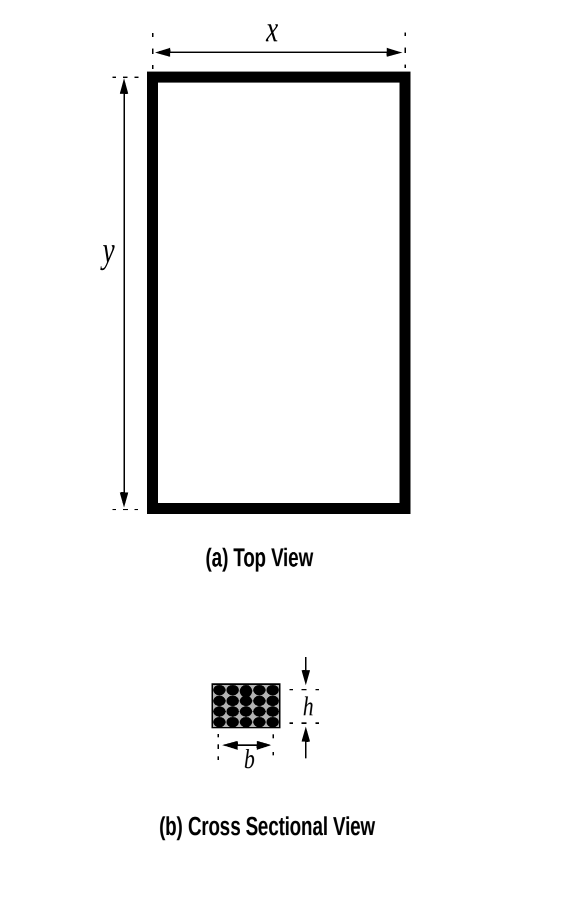
EQUATION 27:

$$L = \frac{0.0276\ (CN)^2}{1.908C + 9b + 10h} \qquad (\mu H)$$

where:

N = number of turns
C = x + y + 2h
x = width of coil
y = length of coil
b = width of cross section
h = height (coil build up) of cross section
Note: All dimensions are in cm

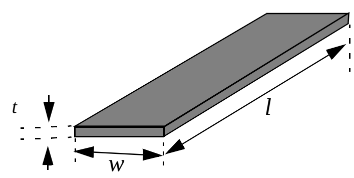
FIGURE 11: N-TURN SQUARE LOOP COIL WITH MULTILAYER



INDUCTANCE OF THIN FILM INDUCTOR WITH A RECTANGULAR CROSS SECTION

Inductance of a conductor with rectangular cross section as shown in Figure 12 is calculated as:

FIGURE 12: A STRAIGHT THIN FILM INDUCTOR



EQUATION 28:

$$L = 0.002l\left\{\ln\left(\frac{2l}{w+t}\right) + 0.50049 + \frac{w+t}{3l}\right\} \qquad (\mu H)$$

where:

w = width in cm
t = thickness in cm
l = length of conductor in cm

INDUCTANCE OF A FLAT SQUARE COIL

Inductance of a flat square coil of rectangular cross section with N turns is calculated by[2]:

EQUATION 29:

$$L = 0.0467aN^2\left\{\log_{10}\left(2\frac{a^2}{t+w}\right) - \log_{10}(2.414a)\right\} + 0.02032aN^2\left\{0.914 + \left[\frac{0.2235}{a}(t+w)\right]\right\}$$

where:

L = in µH

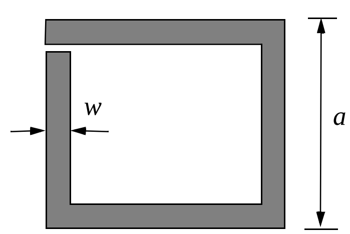
a = side length in inches

t = thickness in inches

w = width in inches

N = total number of turns

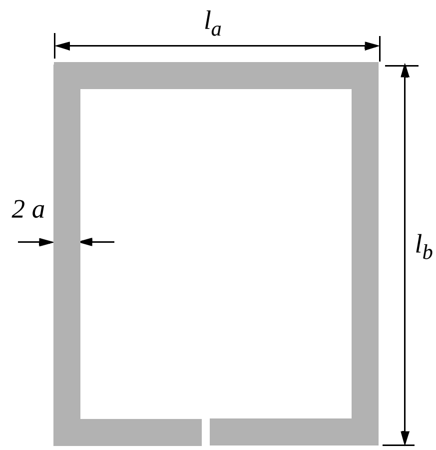
FIGURE 13: SQUARE LOOP INDUCTOR WITH A RECTANGULAR CROSS SECTION



EXAMPLE ON ONE TURN READER ANTENNA

If reader antenna is made of a rectangular loop composed of a thin wire or a thin plate element, its inductance can be calculated by the following simple formula [5]:

FIGURE 14: ONE TURN READER ANTENNA



EQUATION 30:

$$L = 4\left\{l_b \ln\left(\frac{2A}{a(l_b + l_c)}\right) + l_a \ln\left(\frac{2A}{a(l_a + l_c)}\right) + 2\left[a + l_c - (l_a + l_b)\right]\right\} \qquad (nH)$$

where

units are all in cm, and a = radius of wire in cm.

$$l_c = \sqrt{l_a^2 + l_b^2}$$
$$A = l_a \times l_b$$

Example with dimension:

One-turn rectangular shape with l_a = 18.887 cm, l_b = 25.4 cm, width a = 0.254 cm gives 653 (nH) using the above equation.

INDUCTANCE OF N-TURN PLANAR SPIRAL COIL

Inductance of planar structure is well calculated in Reference [4]. Consider an inductor made of straight segments as shown in Figure 15. The inductance is the sum of self inductances and mutual inductances[4]:

EQUATION 31:

$$L_T = L_o - M_+ - M_- \qquad (\mu H)$$

where:

- L_T = Total Inductance
- L_o = Sum of self inductances of all straight segments
- M_+ = Sum of positive mutual inductances
- M_- = Sum of negative mutual inductances

The mutual inductance is the inductance that is resulted from the magnetic fields produced by adjacent conductors. The mutual inductance is positive when the directions of current on conductors are in the same direction, and negative when the directions of currents are opposite directions. The mutual inductance between two parallel conductors is a function of the length of the conductors and of the geometric mean distance between them. The mutual inductance of two conductors is calculated by:

EQUATION 32:

$$M = 2lF \qquad (nH)$$

where l is the length of conductor in centimeter. F is the mutual inductance parameter and calculated as:

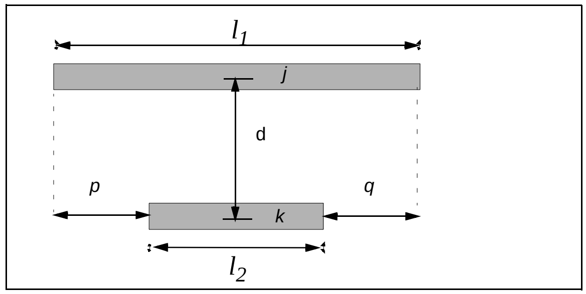
EQUATION 33:

$$F = \ln\left\{\left(\frac{l}{d}\right) + \left[1 + \left(\frac{l}{d}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{l}{d}\right)^2\right]^{1/2} + \left(\frac{d}{l}\right)$$

where d is the geometric mean distance between two conductors, which is approximately equal to the distance between the track center of the conductors.

Let us consider the two conductor segments shown in Figure 15:

FIGURE 15: TWO CONDUCTOR SEGMENTS FOR MUTUAL INDUCTANCE CALCULATION



j and k in the above figure are indices of conductor, and p and q are the indices of the length for the difference in the length of the two conductors.

The above configuration (with partial segments) occurs between conductors in multiple turn spiral inductor. The mutual inductance of conductors j and k in the above configuration is:

EQUATION 34:

$$M_{j,k} = \frac{1}{2}\{(M_{k+p} + M_{k+q}) - (M_p + M_q)\}$$

$$= \frac{1}{2}\{(M_j + M_k) - M_q\} \qquad for\ p = 0 \qquad \text{(a)}$$

$$= \frac{1}{2}\{(M_j + M_k) - M_p\} \qquad for\ q = 0 \qquad \text{(b)}$$

$$= M_{k+p} - M_p \qquad for\ p = q \qquad \text{(c)}$$

$$= M_k \qquad for\ p = q = 0 \qquad \text{(d)}$$

If the length of l_1 and l_2 are the same ($l_1 = l_2$), then Equation 34 (d) is used. Each mutual inductance term in the above equation is calculated as follows by using Equations 33 and 34:

EQUATION 35:

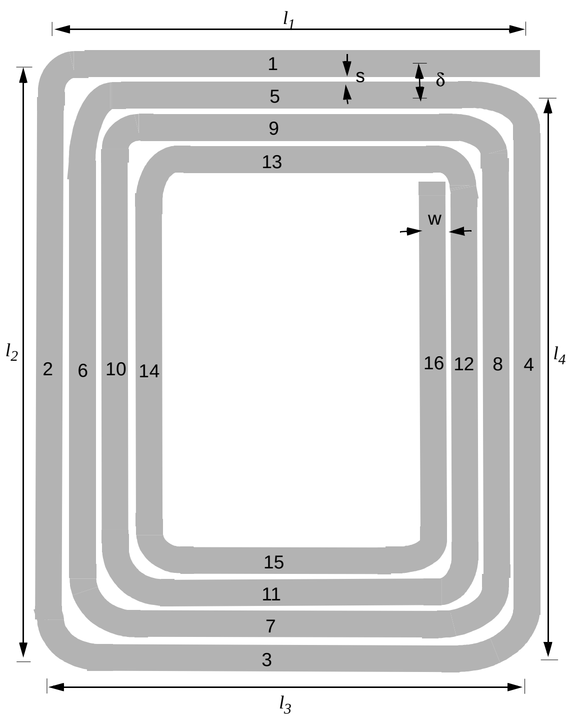
$$M_{k+p} = 2l_{k+p}F_{k+p}$$

where

$$F_{k+p} = \ln\left\{\left(\frac{l_{k+p}}{d_{j,k}}\right) + \left[1 + \left(\frac{l_{k+p}}{d_{j,k}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{j,k}}{l_{k+p}}\right)^2\right]^{1/2} + \left(\frac{d_{j,k}}{l_{k+p}}\right)$$

The following examples shows how to use the above formulas to calculate the inductance of a 4-turn rectangular spiral inductor.

EXAMPLE 7: INDUCTANCE OF RECTANGULAR PLANAR SPIRAL INDUCTOR



1, 2, 3, ... ,16 are indices of conductor. For four full turn inductor, there are 16 straight segments. *s* is the spacing between conductor, and δ (= s + w) is the distance of track centers between two adjacent conductors. l_1 is the length of conductor 1, l_2 is the length of conductor 2, and so on. The length of conductor segments are:

$$l_3 = l_1,\ l_4 = l_2 - \delta,\ l_5 = l_1 - \delta,\ l_6 = l_4 - \delta,$$
$$l_7 = l_5 - \delta,\ l_8 = l_6 - \delta,\ l_9 = l_7 - \delta,$$
$$l_{10} = l_8 - \delta,\ l_{11} = l_9 - \delta,\ l_{12} = l_{10} - \delta,$$
$$l_{13} = l_{11} - \delta,\ l_{14} = l_{12} - \delta,\ l_{15} = l_{13} - \delta,$$
$$l_{16} = l_{14} - \delta$$

The total inductance of the coil is equal to the sum of the self inductance of each straight segment (L_0 = L1 + L2 + L3 + L4 +.....+ L16) plus all the mutual inductances between these segments as shown in Equation 31.

The self inductance is calculated by Equation (28), and the mutual inductances are calculated by Equations (32) - (34).

For the four-turn spiral, there are both positive and negative mutual inductances. The positive mutual inductance (M_+) is the mutual inductance between conductors that have the same current direction. For example, the current on segments 1 and 5 are in the same direction. Therefore, the mutual inductance between the two conductor segments is positive. On the other hand, the currents on segments 1 and 15 are in the opposite direction. Therefore, the mutual inductance between conductors 1 and 15 is negative term.

The mutual inductance is maximized if the two segments are in parallel, and minimum if they are placed in orthogonal (in 90 degrees). Therefore the mutual inductance between segments 1 and 2, 1 and 6, 1 and 10, 1 and 14, etc, are negligible in calculation.

In Example 7, the total positive mutual inductance terms are:

EQUATION 36:

$$\begin{aligned} M_+ = {} & 2(M_{1,5} + M_{1,9} + M_{1,13}) \\ & +2(M_{5,9} + M_{5,13} + M_{9,13}) \\ & +2(M_{3,7} + M_{3,11} + M_{3,15}) \\ & +2(M_{7,11} + M_{7,15} + M_{11,15}) \\ & +2(M_{2,6} + M_{2,10} + M_{2,14}) \\ & +2(M_{6,10} + M_{6,14} + M_{10,14}) \\ & +2(M_{4,8} + M_{4,12} + M_{4,16}) \\ & +2(M_{8,12} + M_{8,16} + M_{12,16}) \end{aligned}$$

The total negative mutual inductance terms are:

EQUATION 37:

$$\begin{aligned} M_- = {} & 2(M_{1,3} + M_{1,7} + M_{1,11} + M_{1,15}) \\ & +2(M_{5,3} + M_{5,7} + M_{5,11} + M_{5,15}) \\ & +2(M_{9,3} + M_{9,7} + M_{9,11} + M_{9,15}) \\ & +2(M_{13,15} + M_{13,11} + M_{13,7} + M_{13,3}) \\ & +2(M_{2,4} + M_{2,8} + M_{2,12} + M_{2,16}) \\ & +2(M_{6,4} + M_{6,8} + M_{6,12} + M_{6,16}) \\ & +2(M_{10,4} + M_{10,8} + M_{10,12} + M_{10,16}) \\ & +2(M_{14,4} + M_{14,8} + M_{14,12} + M_{14,16}) \end{aligned}$$

See Appendix A for calculation of each individual mutual inductance term in Equations (36) - (37).

EXAMPLE 8: INDUCTANCE CALCULATION INCLUDING MUTUAL INDUCTANCE TERMS FOR A RECTANGULAR SHAPED ONE TURN READER ANTENNA

Let us calculate the Inductance of one turn loop etched antenna on PCB board for reader antenna (for example, the MCRF450 reader antenna in the DV103006 development kit) with the following parameters:

l_2 = $l4$ = 10” = 25.4 cm

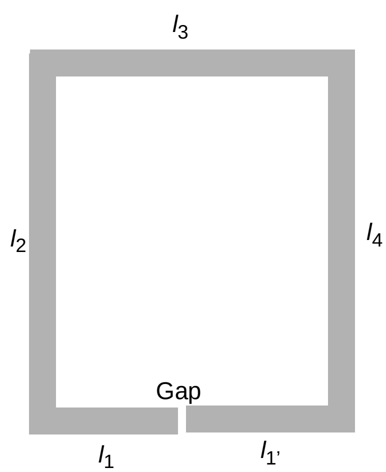
l_3 = 7.436” = 18.887 cm

l_1 = $l_{1'}$ = 3” = 7.62

gap = 1.4536” = 3.692 cm

trace width (w) = 0.508 cm

trace thickness (t) = 0.0001 cm



In the one turn rectangular shape inductor, there are four sides. Because of the gap, there are a total of 5 conductor segments. In one-turn inductor, the direction of current on each conductor segment is all opposite directions to each other. For example, the direction of current on segment 2 and 4, 1 and 3, 1’ and 3 are opposite. There is no conductor segments that have the same current direction. Therefore, there is no positive mutual inductance.

From Equation 31, the total inductance is:

EQUATION 38:

$$L_T = L_o + M_+ - M_- \quad (\mu H)$$
$$= L_o - M_- \quad (\mu H)$$

where

M+ = 0 since the direction of current on each segment is opposite with respect to the currents on other segments.

$$L_o = L_1 + L_{1'} + L_2 + L_3 + L_4$$

By solving the self inductance using Equation (28),

$$L_1 = L_{1'} = 59.8 \quad (nH)$$
$$L_2 = L_4 = 259.7 \quad (nH)$$
$$L_3 = 182 \quad (nH)$$
$$L_0 = 821 \quad (nH)$$

Negative mutual inductances are solved as follows:

$$M_- = 2(M_{1,3} + M_{1',3} + M_{2,4})$$
$$M_{2,4} = 2 l_2 F_{2,4}$$
$$M_{1,3} = \frac{1}{2}(M_3 + M_1 - M_{1'+\text{gap}})$$
$$M_{1',3} = \frac{1}{2}(M_3 + M_{1'} - M_{1+\text{gap}})$$

$$F_{2,4} = \ln\left\{\frac{l_2}{d_{2,4}} + \left[1 + \left(\frac{l_2}{d_{2,4}}\right)^2\right]^{\frac{1}{2}}\right\} - \left[1 + \left(\frac{d_{2,4}}{l_2}\right)^2\right]^{\frac{1}{2}} + \frac{l_2}{d_{2,4}}$$

$$F_3 = \ln\left\{\frac{l_3}{d_{1,3}} + \left[1 + \left(\frac{l_3}{d_{1,3}}\right)^2\right]^{\frac{1}{2}}\right\} - \left[1 + \left(\frac{d_{1,3}}{l_3}\right)^2\right]^{\frac{1}{2}} + \frac{l_3}{d_{1,3}}$$

$$F_1 = \ln\left\{\frac{l_1}{d_{1,3}} + \left[1 + \left(\frac{l_1}{d_{1,3}}\right)^2\right]^{\frac{1}{2}}\right\} - \left[1 + \left(\frac{d_{1,3}}{l_1}\right)^2\right]^{\frac{1}{2}} + \frac{l_1}{d_{1,3}}$$

$$F_{1'} = \ln\left\{\frac{l_{1'}}{d_{1',3}} + \left[1 + \left(\frac{l_{1'}}{d_{1',3}}\right)^2\right]^{\frac{1}{2}}\right\} - \left[1 + \left(\frac{d_{1',3}}{l_{1'}}\right)^2\right]^{\frac{1}{2}} + \frac{l_{1'}}{d_{1',3}}$$

$$M_1 = 2 l_1 F_1$$
$$M_{1'} = 2 l_{1'} F_1$$
$$M_{1'+gap} = 2 l_{1'+gap} F_{1'+gap}$$

$$F_{1'+gap} = \ln\left\{\frac{l_{1'+gap}}{d_{1'+gap,3}} + \left[1 + \left(\frac{l_{1'+gap}}{d_{1'+gap,3}}\right)^2\right]^{\frac{1}{2}}\right\} + \frac{l_{1'}}{d_{1',3}} - \left[1 + \left(\frac{d_{1'+gap,3}}{l_{1'+gap}}\right)^2\right]^{\frac{1}{2}}$$

By solving the above equation, the mutual inductance between each conductor are:

$M_{2,4} = 30.1928$ (nH),

$M_{1,3} = 5.1818$ (nH) $= M_{1',3}$

Therefore, the total inductance of the antenna is:

$L_T = L_o - M_- = L_o - 2(M_{2,4} + M_{1,3}) =$

$= 797.76 - 81.113 = 716.64$ (nH)

It has been found that the inductance calculated using Equation (38) has about 9% higher than the result using Equation (30) for the same physical dimension. The resulting difference of the two formulas is contributed mainly by the mutual inductance terms. Equation (38) is recommended if it needs very accurate calculation while Equation (30) gives quick answers within about 10 percent of error.

The computation software using Mathlab is shown in Appendix B.

The formulas for inductance are widely published and provide a reasonable approximation for the relationship between inductance and the number of turns for a given physical size[1–7]. When building prototype coils, it is wise to exceed the number of calculated turns by about 10% and then remove turns to achieve a right value. For production coils, it is best to specify an inductance and tolerance rather than a specific number of turns.

CONFIGURATION OF ANTENNA CIRCUITS

Reader Antenna Circuits

The inductance for the reader antenna coil for 13.56 MHz is typically in the range of a few microhenries (µH). The antenna can be formed by air-core or ferrite core inductors. The antenna can also be formed by a metallic or conductive trace on PCB board or on flexible substrate.

The reader antenna can be made of either a single coil, that is typically forming a series or a parallel resonant circuit, or a double loop (transformer) antenna coil. Figure 16 shows various configurations of reader antenna circuit. The coil circuit must be tuned to the operating frequency to maximize power efficiency. The tuned LC resonant circuit is the same as the band-pass filter that passes only a selected frequency. The Q of the tuned circuit is related to both read range and bandwidth of the circuit. More on this subject will be discussed in the following section.

Choosing the size and type of antenna circuit depends on the system design topology. The series resonant circuit results in minimum impedance at the resonance frequency. Therefore, it draws a maximum current at the resonance frequency. Because of its simple circuit topology and relatively low cost, this type of antenna circuit is suitable for proximity reader antenna.

On the other hand, a parallel resonant circuit results in maximum impedance at the resonance frequency. Therefore, maximum voltage is available at the resonance frequency. Although it has a minimum resonant current, it still has a strong circulating current that is proportional to Q of the circuit. The double loop antenna coil that is formed by two parallel antenna circuits can also be used.

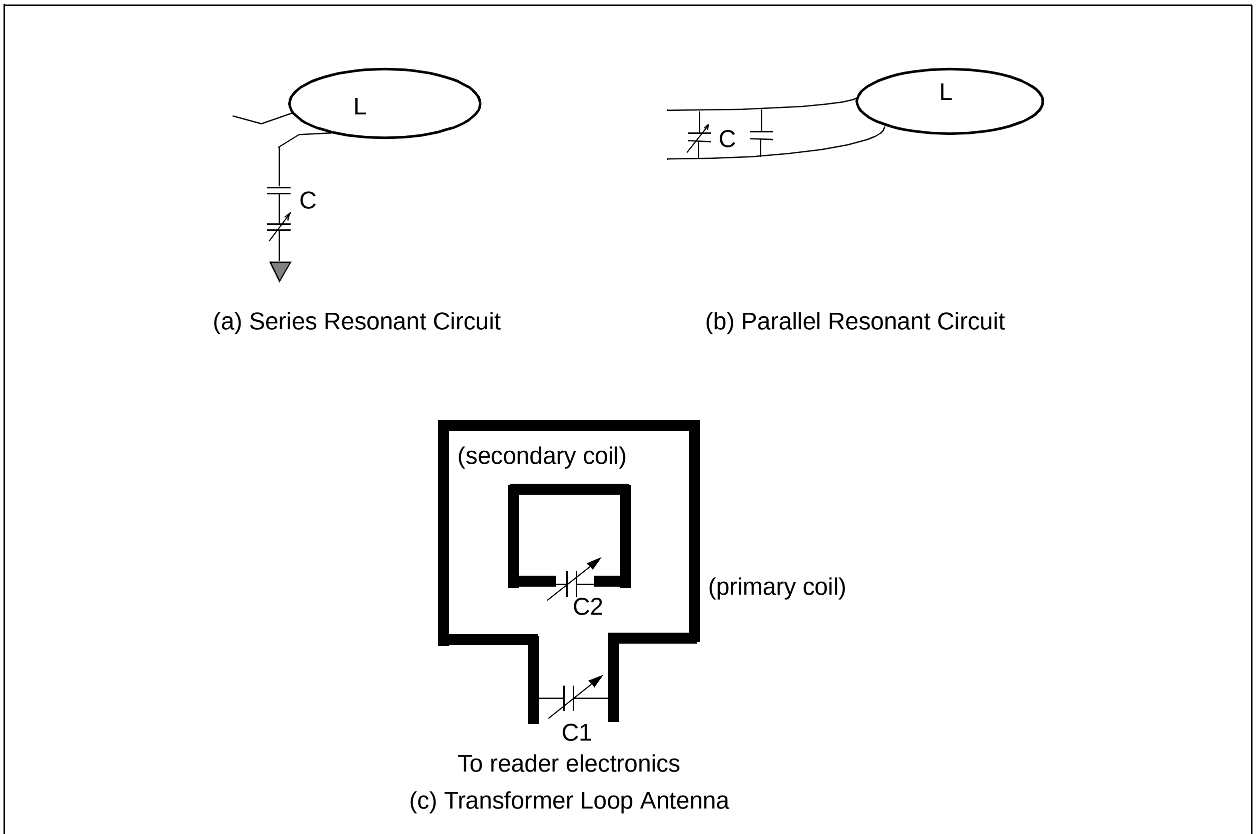
The frequency tolerance of the carrier frequency and output power level from the read antenna is regulated by government regulations (e.g., FCC in the USA).

FCC limits for 13.56 MHz frequency band are as follows:

1. Tolerance of the carrier frequency: 13.56 MHz +/- 0.01% = +/- 1.356 kHz.
2. Frequency bandwidth: +/- 7 kHz.
3. Power level of fundamental frequency: 10 mv/m at 30 meters from the transmitter.
4. Power level for harmonics: -50.45 dB down from the fundamental signal.

The transmission circuit including the antenna coil must be designed to meet the FCC limits.

FIGURE 16: VARIOUS READER ANTENNA CIRCUITS



Tag Antenna Circuits

The MCRF355 device communicates data by tuning and detuning the antenna circuit (see AN707). Figure 17 shows examples of the external circuit arrangement.

The external circuit must be tuned to the resonant frequency of the reader antenna. In a detuned condition, a circuit element between the antenna B and VSS pads is shorted. The frequency difference (delta frequency) between tuned and detuned frequencies must be adjusted properly for optimum operation. It has been found that maximum modulation index and maximum read range occur when the tuned and detuned frequencies are separated by 3 to 6 MHz.

The tuned frequency is formed from the circuit elements between the antenna A and VSS pads without shorting the antenna B pad. The detuned frequency is found when the antenna B pad is shorted. This detuned frequency is calculated from the circuit between antenna A and VSS pads excluding the circuit element between antenna B and VSS pads.

In Figure 17 (a), the tuned resonant frequency is:

EQUATION 39:

$$f_o = \frac{1}{2\pi\sqrt{L_T C}}$$

where:

- L_T = $L_1 + L_2 + 2L_M$ = Total inductance between antenna A and VSS pads
- L_1 = inductance between antenna A and antenna B pads
- L_2 = inductance between antenna B and VSS pads
- M = mutual inductance between coil 1 and coil 2
  = $k\sqrt{L_1 L_2}$
- k = coupling coefficient between the two coils
- C = tuning capacitance

and detuned frequency is:

EQUATION 40:

$$f_{detuned} = \frac{1}{2\pi\sqrt{L_1 C}}$$

In this case, $f_{detuned}$ is higher than f_{tuned}.

Figure 17(b) shows another example of the external circuit arrangement. This configuration controls C_2 for tuned and detuned frequencies. The tuned and untuned frequencies are:

EQUATION 41:

$$f_{tuned} = \frac{1}{2\pi\sqrt{\left(\frac{C_1 C_2}{C_1 + C_2}\right)L}}$$

and

EQUATION 42:

$$f_{detuned} = \frac{1}{2\pi\sqrt{LC_1}}$$

A typical inductance of the coil is about a few microhenry with a few turns. Once the inductance is determined, the resonant capacitance is calculated from the above equations. For example, if a coil has an inductance of 1.3 µH, then it needs a 106 pF of capacitance to resonate at 13.56 MHz.

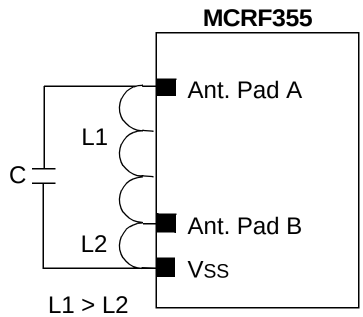
CONSIDERATION ON QUALITY FACTOR Q AND BANDWIDTH OF TUNING CIRCUIT

The voltage across the coil is a product of quality factor Q of the circuit and input voltage. Therefore, for a given input voltage signal, the coil voltage is directly proportional to the Q of the circuit. In general, a higher Q results in longer read range. However, the Q is also related to the bandwidth of the circuit as shown in the following equation.

EQUATION 43:

$$Q = \frac{f_o}{B}$$

FIGURE 17: VARIOUS EXTERNAL CIRCUIT CONFIGURATIONS



$$f_{tuned} = \frac{1}{2\pi\sqrt{L_T C}}$$

$$f_{detuned} = \frac{1}{2\pi\sqrt{L_1 C}}$$

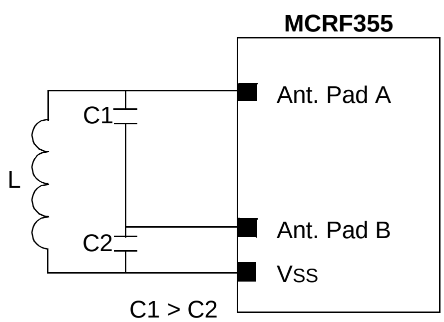
$$L_T = L_1 + L_2 + 2L_m$$

where:

L_m = mutual inductance
= $K\sqrt{L_1 L_2}$

K = coupling coefficient of two inductors
$0 \le K \le 1$

(a) Two inductors and one capacitor

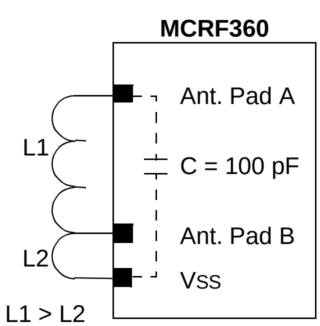


$$f_{tuned} = \frac{1}{2\pi\sqrt{L C_T}}$$

$$f_{detuned} = \frac{1}{2\pi\sqrt{L C_1}}$$

$$C_T = \frac{C_1 C_2}{C_1 + C_2}$$

(b) Two capacitors and one inductor



$$f_{tuned} = \frac{1}{2\pi\sqrt{L_T C}}$$

$$f_{detuned} = \frac{1}{2\pi\sqrt{L_1 C}}$$

$$L_T = L_1 + L_2 + 2L_m$$

(c) Two inductors with one internal capacitor

Bandwidth requirement and limit on circuit Q for MCRF355

Since the MCRF355 operates with a data rate of 70 kHz, the reader antenna circuit needs a bandwidth of at least twice of the data rate. Therefore, it needs:

EQUATION 44:

$$B_{minimum} = 140 \text{ kHz}$$

Assuming the circuit is turned at 13.56 MHz, the maximum attainable Q is obtained from Equations 43 and 44:

EQUATION 45:

$$Q_{max} = \frac{f_o}{B} = 96.8$$

In a practical LC resonant circuit, the range of Q for 13.56 MHz band is about 40. However, the Q can be significantly increased with a ferrite core inductor. The system designer must consider the above limits for optimum operation.

RESONANT CIRCUITS

Once the frequency and the inductance of the coil are determined, the resonant capacitance can be calculated from:

EQUATION 46:

$$C = \frac{1}{L(2\pi f_o)^2}$$

In practical applications, parasitic (distributed) capacitance is present between turns. The parasitic capacitance in a typical tag antenna coil is a few (pF). This parasitic capacitance increases with operating frequency of the device.

There are two different resonant circuits: parallel and series. The parallel resonant circuit has maximum impedance at the resonance frequency. It has a minimum current and maximum voltage at the resonance frequency. Although the current in the circuit is minimum at the resonant frequency, there are a circulation current that is proportional to Q of the circuit. The parallel resonant circuit is used in both the tag and the high power reader antenna circuit.

On the other hand, the series resonant circuit has a minimum impedance at the resonance frequency. As a result, maximum current is available in the circuit. Because of its simplicity and the availability of the high current into the antenna element, the series resonant circuit is often used for a simple proximity reader.

Parallel Resonant Circuit

Figure 18 shows a simple parallel resonant circuit. The total impedance of the circuit is given by:

EQUATION 47:

$$Z(j\omega) = \frac{j\omega L}{(1-\omega^2 LC) + j\frac{\omega L}{R}} \quad (\Omega)$$

where ω is an angular frequency given as $\omega = 2\pi f$.

The maximum impedance occurs when the denominator in the above equation is minimized. This condition occurs when:

EQUATION 48:

$$\omega^2 LC = 1$$

This is called a resonance condition, and the resonance frequency is given by:

EQUATION 49:

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

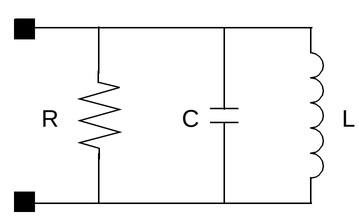
By applying Equation 48 into Equation 47, the impedance at the resonance frequency becomes:

EQUATION 50:

$$Z = R$$

where R is the load resistance.

FIGURE 18: PARALLEL RESONANT CIRCUIT



The R and C in the parallel resonant circuit determine the bandwidth, B, of the circuit.

EQUATION 51:

$$B = \frac{1}{2\pi RC} \quad (Hz)$$

The quality factor, Q, is defined by various ways such as:

EQUATION 52:

$$Q = \frac{\text{Energy Stored in the System per One Cycle}}{\text{Energy Dissipated in the System per One Cycle}}$$

$$= \frac{\text{reactance}}{\text{resistance}}$$

$$= \frac{\omega L}{r} \qquad \text{For inductance}$$

$$= \frac{1}{\omega c r} \qquad \text{For capacitance}$$

$$= \frac{f_0}{B}$$

where:

- ω = $2\pi f$ = angular frequency
- f_0 = resonant frequency
- B = bandwidth
- r = ohmic losses

By applying Equation 49 and Equation 51 into Equation 52, the Q in the parallel resonant circuit is:

EQUATION 53:

$$Q = R\sqrt{\frac{C}{L}}$$

The Q in a parallel resonant circuit is proportional to the load resistance R and also to the ratio of capacitance and inductance in the circuit.

When this parallel resonant circuit is used for the tag antenna circuit, the voltage drop across the circuit can be obtained by combining Equations 8 and 53:

EQUATION 54:

$$V_o = 2\pi f_o NQSB_o \cos\alpha$$

$$= 2\pi f_0 N\left(R\sqrt{\frac{C}{L}}\right) SB_0 \cos\alpha$$

The above equation indicates that the induced voltage in the tag coil is inversely proportional to the square root of the coil inductance, but proportional to the number of turns and surface area of the coil.

Series Resonant Circuit

A simple series resonant circuit is shown in Figure 19. The expression for the impedance of the circuit is:

EQUATION 55:

$$Z(j\omega) = r + j(X_L - X_C) \qquad (\Omega)$$

where:

- r = a DC ohmic resistance of coil and capacitor
- X_L *and* X_C = the reactance of the coil and capacitor, respectively, such that:

EQUATION 56:

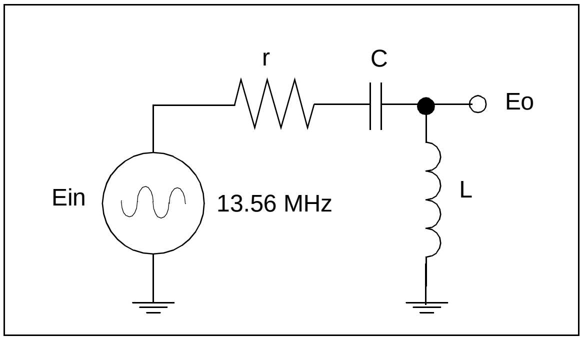
$$X_L = 2\pi f_o L \qquad (\Omega)$$

EQUATION 57:

$$X_c = \frac{1}{2\pi f_o C} \qquad (\Omega)$$

The impedance in Equation 55 becomes minimized when the reactance component cancelled out each other such that $X_L = X_C$. This is called a resonance condition. The resonance frequency is same as the parallel resonant frequency given in Equation 49.

FIGURE 19: SERIES RESONANCE CIRCUIT



The half power frequency bandwidth is determined by r and L, and given by:

EQUATION 58:

$$B = \frac{r}{2\pi L} \qquad (Hz)$$

The quality factor, Q, in the series resonant circuit is given by:

$$Q = \frac{f_0}{B} = \frac{\omega L}{r} = \frac{1}{r\omega C}$$

The series circuit forms a voltage divider, the voltage drops in the coil is given by:

EQUATION 59:

$$V_o = \frac{jX_L}{r + jX_L - jX_c}V_{in}$$

When the circuit is tuned to a resonant frequency such as $X_L = X_C$, the voltage across the coil becomes:

EQUATION 60:

$$V_o = \frac{jX_L}{r}V_{in}$$

$$= jQV_{in}$$

The above equation indicates that the coil voltage is a product of input voltage and Q of the circuit. For example, a circuit with Q of 40 can have a coil voltage that is 40 times higher than input signal. This is because all energy in the input signal spectrum becomes squeezed into a single frequency band.

EXAMPLE 9: CIRCUIT PARAMETERS

If the DC ohmic resistance r is 5 Ω, then the L and C values for 13.56 MHz resonant circuit with Q = 40 are:

EQUATION 61:

$$X_L = Qr_s = 200\Omega$$

$$L = \frac{X_L}{2\pi f} = \frac{200}{2\pi(13.56MHz)} = 2.347 \qquad (\mu H)$$

$$C = \frac{1}{2\pi f X_L} = \frac{1}{2\pi(13.56\ \text{MHz})(200)} = 58.7\ \ (\text{pF})$$

TUNING METHOD

The circuit must be tuned to the resonance frequency for a maximum performance (read range) of the device. Two examples of tuning the circuit are as follows:

- **Voltage Measurement Method:**
  a) Set up a voltage signal source at the resonance frequency.
  b) Connect a voltage signal source across the resonant circuit.
  c) Connect an Oscilloscope across the resonant circuit.
  d) Tune the capacitor or the coil while observing the signal amplitude on the Oscilloscope.
  e) Stop the tuning at the maximum voltage.
- **S-Parameter or Impedance Measurement Method using Network Analyzer:**
  a) Set up an S-Parameter Test Set (Network Analyzer) for S11 measurement, and do a calibration.
  b) Measure the S11 for the resonant circuit.
  c) Reflection impedance or reflection admittance can be measured instead of the S11.
  d) Tune the capacitor or the coil until a maximum null (S11) occurs at the resonance frequency, f_o. For the impedance measurement, the maximum peak will occur for the parallel resonant circuit, and minimum peak for the series resonant circuit.

FIGURE 20: VOLTAGE VS. FREQUENCY FOR RESONANT CIRCUIT

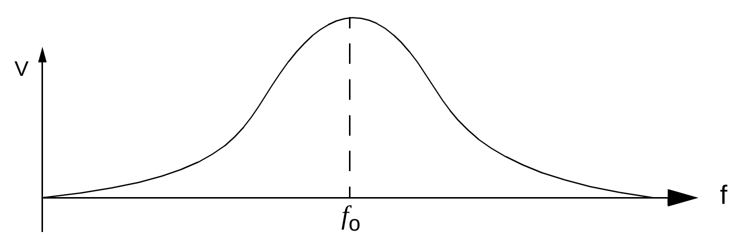
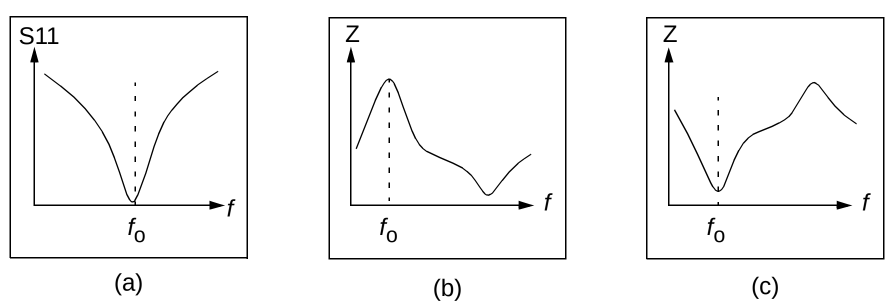


FIGURE 21: FREQUENCY RESPONSES FOR RESONANT CIRCUIT



Note 1: (a) S11 Response, (b) Impedance Response for a Parallel Resonant Circuit, and (c) Impedance Response for a Series Resonant Circuit.

2: In (a), the null at the resonance frequency represents a minimum input reflection at the resonance frequency. This means the circuit absorbs the signal at the frequency while other frequencies are reflected back. In (b), the impedance curve has a peak at the resonance frequency. This is because the parallel resonant circuit has a maximum impedance at the resonance frequency. (c) shows a response for the series resonant circuit. Since the series resonant circuit has a minimum impedance at the resonance frequency, a minimum peak occurs at the resonance frequency.

READ RANGE OF RFID DEVICES

Read range is defined as a maximum communication distance between the reader and tag. In general, the read range of passive RFID products varies, depending on system configuration and is affected by the following parameters:

a) Operating frequency and performance of antenna coils
b) Q of antenna and tuning circuit
c) Antenna orientation
d) Excitation current
e) Sensitivity of receiver
f) Coding (or modulation) and decoding (or demodulation) algorithm
g) Number of data bits and detection (interpretation) algorithm
h) Condition of operating environment (electrical noise), etc.

The read range of 13.56 MHz is relatively longer than that of 125 kHz device. This is because the antenna efficiency increases as the frequency increases. With a given operating frequency, the conditions (a – c) are related to the antenna configuration and tuning circuit. The conditions (d – e) are determined by a circuit topology of reader. The condition (f) is a communication protocol of the device, and (g) is related to a firmware software program for data detection.

Assuming the device is operating under a given condition, the read range of the device is largely affected by the performance of the antenna coil. It is always true that a longer read range is expected with the larger size of the antenna with a proper antenna design. Figures 22 and 23 show typical examples of the read range of various passive RFID devices.

FIGURE 22: READ RANGE VS. TAG SIZE FOR TYPICAL PROXIMITY APPLICATIONS*

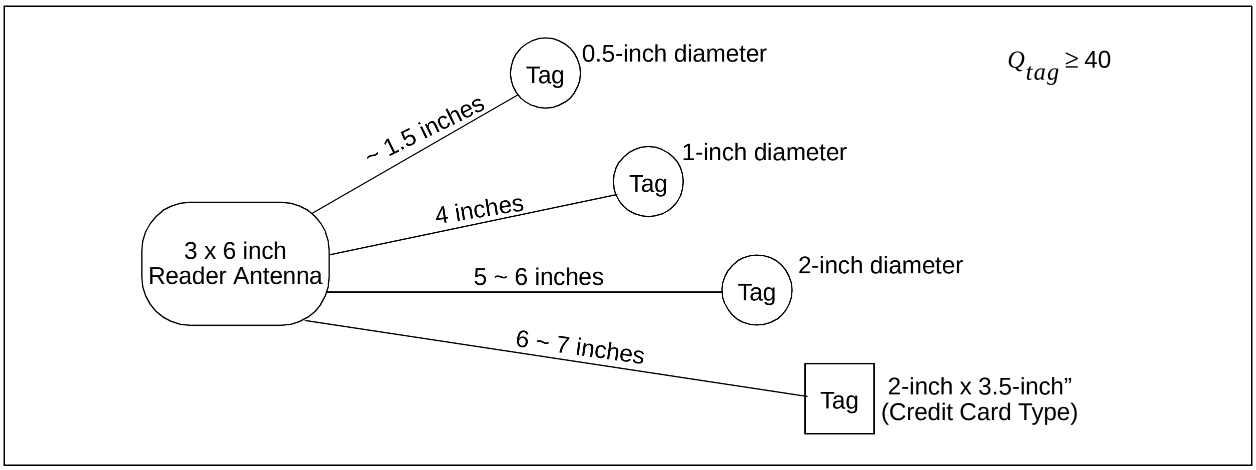
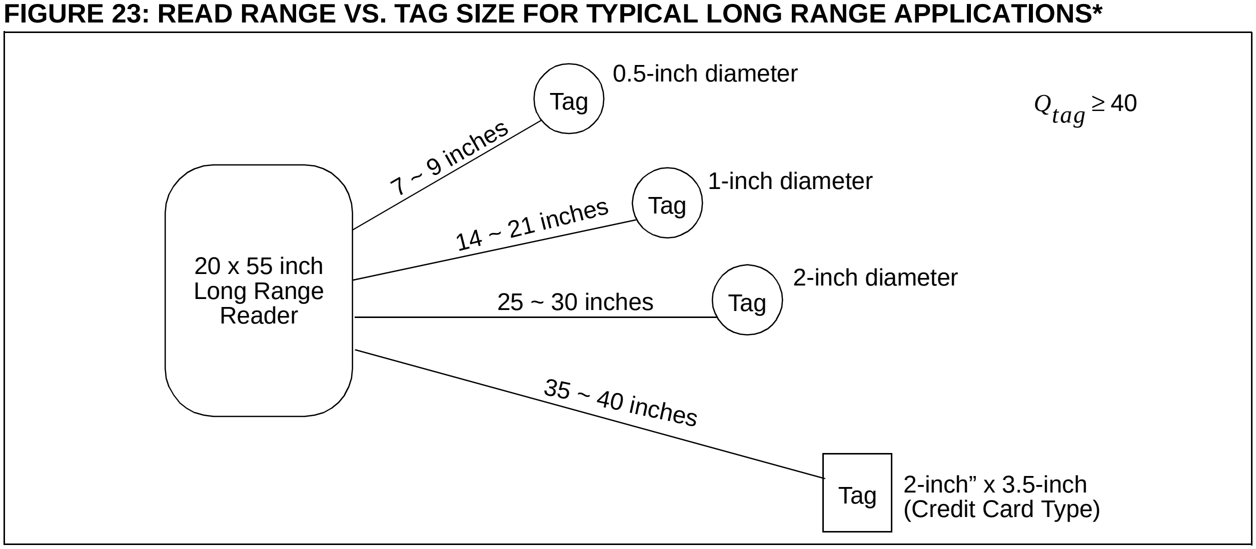


FIGURE 23: READ RANGE VS. TAG SIZE FOR TYPICAL LONG RANGE APPLICATIONS*



Note: Actual results may be shorter or longer than the range shown, depending upon factors discussed above.

APPENDIX A: CALCULATION OF MUTUAL INDUCTANCE TERMS IN EQUATIONS 36 AND 37

Positive Mutual Inductance Terms:

EQUATION A.1 Mutual inductance between conductors 1 and 5

$$M_{1,5} = \frac{1}{2}\left\{\left(M_1^{1,5} + M_5^{1,5}\right) - M_\delta^{1,5}\right\}$$

where:

$$M_1^{1,5} = 2l_1F_1^{1,5}$$

$$M_5^{1,5} = 2l_5F_5^{1,5}$$

$$M_\delta^{1,5} = 2d_{1,5}F_\delta^{1,5}$$

$$F_1^{1,5} = \ln\left\{\frac{l_1}{d_{1,5}} + \left[1 + \left(\frac{l_1}{d_{1,5}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,5}}{l_1}\right)^2\right]^{1/2} + \left(\frac{d_{1,5}}{l_1}\right)$$

$$F_5^{1,5} = \ln\left\{\frac{l_5}{d_{1,5}} + \left[1 + \left(\frac{l_5}{d_{1,5}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,5}}{l_5}\right)^2\right]^{1/2} + \left(\frac{d_{1,5}}{l_5}\right)$$

$$F_\delta^{1,5} = \ln\left\{\frac{l_\delta}{d_{1,5}} + \left[1 + \left(\frac{l_\delta}{d_{1,5}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,5}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{1,5}}{l_\delta}\right)$$

$$\delta = w + s$$

$$l_\delta = \delta$$

where $d_{1,5}$ is the distance between track centers of conductor l_1 and l_5. S is the interspacing between conductors l_1 and l_5, w is the width of track, δ is $s + w$.

$F_1^{1,5}$ is the mutual inductance parameter between conductor segments 1 and 5 by viewing from conductor 1.

$F_5^{1,5}$ is the mutual inductance parameter between conductor segments 1 and 5 by viewing from conductor 5.

$F_\delta^{1,5}$ is the mutual inductance parameter between conductor segments 1 and 5 by viewing from the length difference between the two conductors.

EQUATION A.2 Mutual inductance between conductors 1 and 9

$$M_{1,9} = \frac{1}{2}\left\{\left(M_{9+2\delta}^{1,9} + M_{9+\delta}^{1,9}\right) - \left(M_{2\delta}^{1,9} + M_\delta^{1,9}\right)\right\}$$

where:

$$M_{9+2\delta}^{1,9} = 2l_{9+2\delta}F_{9+2\delta}^{1,9}$$

$$M_{9+\delta}^{1,9} = 2l_{9+\delta}F_{9+\delta}^{1,9}$$

$$M_{2\delta}^{1,9} = 2d_{1,9}F_{2\delta}^{1,9}$$

$$M_\delta^{1,9} = 2d_{1,9}F_\delta^{1,9}$$

$$F_{9+2\delta}^{1,9} = \ln\left\{\frac{l_{9+2\delta}}{d_{1,9}} + \left[1 + \left(\frac{l_{9+2\delta}}{d_{1,9}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,9}}{l_{9+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,9}}{l_{9+2\delta}}\right)$$

$$F_{9+\delta}^{1,9} = \ln\left\{\frac{l_{9+\delta}}{d_{1,9}} + \left[1 + \left(\frac{l_{9+\delta}}{d_{1,9}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,9}}{l_{9+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,9}}{l_{9+\delta}}\right)$$

$$F_{2\delta}^{1,9} = \ln\left\{\frac{l_{2\delta}}{d_{1,9}} + \left[1 + \left(\frac{l_{2\delta}}{d_{1,9}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,9}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,9}}{l_{2\delta}}\right)$$

$$F_\delta^{1,9} = \ln\left\{\frac{l_\delta}{d_{1,9}} + \left[1 + \left(\frac{l_\delta}{d_{1,9}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,9}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{1,9}}{l_\delta}\right)$$

EQUATION A.3 Mutual inductance between conductors 1 and 13

$$M_{1,13} = \frac{1}{2}\left\{\left(M^{1,13}_{13+3\delta} + M^{1,13}_{13+2\delta}\right) - \left(M^{1,13}_{3\delta} + M^{1,13}_{2\delta}\right)\right\}$$

where:

$$M^{1,13}_{13+3\delta} = 2l_{13+3\delta}F^{1,13}_{13+3\delta}$$

$$M^{1,13}_{13+2\delta} = 2l_{13+2\delta}F^{1,13}_{13+\delta}$$

$$M^{1,13}_{3\delta} = 2d_{1,13}F^{1,13}_{3\delta}$$

$$M^{1,13}_{2\delta} = 2d_{1,13}F^{1,13}_{2\delta}$$

$$F^{1,13}_{13+3\delta} = \ln\left\{\frac{l_{13+3\delta}}{d_{1,13}} + \left[1 + \left(\frac{l_{13+3\delta}}{d_{1,13}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,13}}{l_{13+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,13}}{l_{13+3\delta}}\right)$$

$$F^{1,13}_{13+\delta} = \ln\left\{\frac{l_{13+\delta}}{d_{1,13}} + \left[1 + \left(\frac{l_{13+\delta}}{d_{1,13}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,13}}{l_{13+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,13}}{l_{13+\delta}}\right)$$

$$F^{1,13}_{3\delta} = \ln\left\{\frac{l_{3\delta}}{d_{1,13}} + \left[1 + \left(\frac{l_{3\delta}}{d_{1,13}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,13}}{l_{3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,13}}{l_{3\delta}}\right)$$

$$F^{1,13}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{1,13}} + \left[1 + \left(\frac{l_{2\delta}}{d_{1,13}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,13}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,13}}{l_{2\delta}}\right)$$

EQUATION A.4 Mutual inductance between conductors 5 and 9

$$M_{5,9} = M^{5,9}_{9+\delta} - M^{5,9}_{\delta}$$

where:

$$M^{5,9}_{9+\delta} = 2l_1F^{5,9}_{9+\delta}$$

$$M^{5,9}_{\delta} = 2l_1F^{5,9}_{\delta}$$

$$F^{5,9}_{9+\delta} = \ln\left\{\frac{l_{9+\delta}}{d_{5,9}} + \left[1 + \left(\frac{l_{9+\delta}}{d_{5,9}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,9}}{l_{9+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,9}}{l_{9+\delta}}\right)$$

$$F^{5,9}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{5,9}} + \left[1 + \left(\frac{l_{\delta}}{d_{5,9}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,9}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,9}}{l_{\delta}}\right)$$

EQUATION A.5 Mutual inductance between conductors 5 and 13

$$M_{5,13} = M^{5,13}_{13+2\delta} - M^{5,13}_{2\delta}$$

where:

$$M^{5,13}_{13+2\delta} = 2l_{13+2\delta}F^{5,13}_{13+2\delta}$$

$$M^{5,13}_{2\delta} = 2l_{2\delta}F^{5,13}_{2\delta}$$

$$F^{5,13}_{13+2\delta} = \ln\left\{\frac{l_{13+2\delta}}{d_{5,13}} + \left[1 + \left(\frac{l_{13+2\delta}}{d_{5,13}}\right)^2\right]^{1 \S 2}\right\} - \left[1 + \left(\frac{d_{5,13}}{l_{13+2\delta}}\right)^2\right]^{1 \S 2} + \left(\frac{d_{5,13}}{l_{13+2\delta}}\right)$$

$$F^{5,13}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{5,13}} + \left[1 + \left(\frac{l_{2\delta}}{d_{5,13}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,13}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,13}}{l_{2\delta}}\right)$$

EQUATION A.6 Mutual inductance between conductors 9 and 13

$$M_{9,13} = M^{9,13}_{13+\delta} - M^{9,13}_{\delta}$$

where:

$$M^{9,13}_{13+\delta} = 2l_{13+\delta}F^{9,13}_{13+\delta}$$

$$M^{9,13}_{\delta} = 2l_{\delta}F^{9,13}_{\delta}$$

$$F^{9,13}_{13+\delta} = \ln\left\{\frac{l_{13+\delta}}{d_{9,13}} + \left[1 + \left(\frac{l_{13+\delta}}{d_{9,13}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,13}}{l_{13+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,13}}{l_{13+\delta}}\right)$$

$$F^{9,13}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{9,13}} + \left[1 + \left(\frac{l_{\delta}}{d_{9,13}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,13}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,13}}{l_{\delta}}\right)$$

EQUATION A.7 Mutual inductance between conductors 3 and 7

$$M_{3,7} = M^{3,7}_{7+\delta} - M^{3,7}_{\delta}$$

where:

$$M^{3,7}_{7+\delta} = 2l_{7+\delta}F^{3,7}_{7+\delta}$$

$$M^{3,7}_{\delta} = 2l_{\delta}F^{3,7}_{\delta}$$

$$F^{3,7}_{7+\delta} = \ln\left\{\frac{l_{7+\delta}}{d_{3,7}} + \left[1 + \left(\frac{l_{7+\delta}}{d_{3,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{3,7}}{l_{7+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{3,7}}{l_{7+\delta}}\right)$$

$$F^{3,7}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{3,7}} + \left[1 + \left(\frac{l_{\delta}}{d_{3,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{3,7}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,13}}{l_{\delta}}\right)$$

EQUATION A.8 Mutual inductance between conductors 3 and 11

$$M_{3,11} = M^{3,11}_{11+2\delta} - M^{3,11}_{2\delta}$$

where:

$$M^{3,11}_{11+2\delta} = 2l_{11+2\delta}F^{3,11}_{11+2\delta}$$

$$M^{3,11}_{2\delta} = 2l_{2\delta}F^{3,11}_{2\delta}$$

$$F^{3,11}_{11+2\delta} = \ln\left\{\frac{l_{11+2\delta}}{d_{3,11}} + \left[1 + \left(\frac{l_{11+2\delta}}{d_{3,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{3,11}}{l_{11+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{3,11}}{l_{11+2\delta}}\right)$$

$$F^{3,11}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{3,11}} + \left[1 + \left(\frac{l_{2\delta}}{d_{3,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{3,11}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{3,11}}{l_{2\delta}}\right)$$

EQUATION A.9 Mutual inductance between conductors 3 and 15

$$M_{3,15} = M^{3,15}_{15+3\delta} - M^{3,15}_{3\delta}$$

where:

$$M^{3,15}_{15+3\delta} = 2l_{15+3\delta}F^{3,15}_{15+3\delta}$$

$$M^{3,15}_{3\delta} = 2l_{3\delta}F^{3,15}_{3\delta}$$

$$F^{3,15}_{3+3\delta} = \ln\left\{\frac{l_{15+3\delta}}{d_{3,15}} + \left[1 + \left(\frac{l_{15+3\delta}}{d_{3,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{3,15}}{l_{15+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{3,15}}{l_{15+3\delta}}\right)$$

$$F^{3,15}_{3\delta} = \ln\left\{\frac{l_{3\delta}}{d_{3,15}} + \left[1 + \left(\frac{l_{3\delta}}{d_{3,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{3,15}}{l_{3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{3,15}}{l_{3\delta}}\right)$$

$$\delta = w + s$$

EQUATION A.10 Mutual inductance between conductors 7 and 11

$$M_{7,11} = M^{7,11}_{11+\delta} - M^{7,11}_{\delta}$$

where:

$$M^{7,11}_{11+\delta} = 2l_{11+\delta}F^{7,11}_{11+\delta}$$

$$M^{7,11}_{\delta} = 2l_{\delta}F^{7,11}_{\delta}$$

$$F^{7,11}_{11+\delta} = \ln\left\{\frac{l_{11+\delta}}{d_{7,11}} + \left[1 + \left(\frac{l_{11+\delta}}{d_{7,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{7,11}}{l_{11+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{7,11}}{l_{11+\delta}}\right)$$

$$F^{7,11}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{7,11}} + \left[1 + \left(\frac{l_{\delta}}{d_{7,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{7,11}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{7,11}}{l_{\delta}}\right)$$

EQUATION A.11 Mutual inductance between conductors 7 and 15

$$M_{7,15} = M^{7,15}_{15+2\delta} - M^{7,15}_{2\delta}$$

where:

$$M^{7,15}_{15+2\delta} = 2l_{15+2\delta}F^{7,15}_{15+2\delta}$$

$$M^{7,15}_{2\delta} = 2l_{2\delta}F^{7,15}_{2\delta}$$

$$F^{7,15}_{15+2\delta} = \ln\left\{\frac{l_{15+2\delta}}{d_{7,15}} + \left[1 + \left(\frac{l_{15+2\delta}}{d_{7,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{7,15}}{l_{15+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{7,15}}{l_{15+2\delta}}\right)$$

$$F^{7,15}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{7,15}} + \left[1 + \left(\frac{l_{2\delta}}{d_{7,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{7,15}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{7,15}}{l_{2\delta}}\right)$$

EQUATION A.12 Mutual inductance between conductors 11 and 15

$$M_{11,15} = M^{11,15}_{15+\delta} - M^{11,15}_{\delta}$$

where:

$$M^{11,15}_{15+\delta} = 2l_{15+\delta}F^{11,15}_{15+\delta}$$

$$M^{11,15}_{\delta} = 2l_{\delta}F^{11,15}_{\delta}$$

$$F^{11,15}_{15+\delta} = \ln\left\{\frac{l_{15+\delta}}{d_{11,15}} + \left[1 + \left(\frac{l_{15+\delta}}{d_{11,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{11,15}}{l_{15+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{11,15}}{l_{15+\delta}}\right)$$

$$F^{11,15}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{11,15}} + \left[1 + \left(\frac{l_{\delta}}{d_{11,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{11,15}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{11,15}}{l_{\delta}}\right)$$

EQUATION A.13 Mutual inductance between conductors 2 and 6

$$M_{2,6} = M^{2,6}_{6+\delta} - M^{2,6}_{\delta}$$

where:

$$M^{2,6}_{6+\delta} = 2l_{6+\delta}F^{2,6}_{6+\delta}$$

$$M^{2,6}_{\delta} = 2l_{\delta}F^{2,6}_{\delta}$$

$$F^{2,6}_{6+\delta} = \ln\left\{\frac{l_{6+\delta}}{d_{2,6}} + \left[1 + \left(\frac{l_{6+\delta}}{\delta_{2,6}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{\delta_{2,6}}{l_{6+\delta}}\right)^2\right]^{1/2} + \left(\frac{\delta_{2,6}}{l_{6+\delta}}\right)$$

$$F^{2,6}_{\delta} = \ln\left\{\frac{l_{\delta}}{\delta_{2,6}} + \left[1 + \left(\frac{l_{\delta}}{d_{2,6}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,6}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,6}}{l_{\delta}}\right)$$

EQUATION A.14 Mutual inductance between conductors 2 and 10

$$M_{2,10} = M^{2,10}_{10+2\delta} - M^{2,10}_{2\delta}$$

where:

$$M^{2,10}_{10+2\delta} = 2l_{10+2\delta}F^{2,10}_{10+2\delta}$$

$$M^{2,10}_{2\delta} = 2l_{2\delta}F^{2,10}_{2\delta}$$

$$F^{2,10}_{10+2\delta} = \ln\left\{\frac{l_{10+2\delta}}{d_{2,10}} + \left[1 + \left(\frac{l_{10+2\delta}}{d_{2,10}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,10}}{l_{10+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,10}}{l_{10+2\delta}}\right)$$

$$F^{2,10}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{2,10}} + \left[1 + \left(\frac{l_{2\delta}}{d_{2,10}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,10}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,10}}{l_{2\delta}}\right)$$

EQUATION A.15 Mutual inductance between conductors 2 and 14

$$M_{2,14} = M^{2,14}_{14+3\delta} - M^{2,14}_{3\delta}$$

where:

$$M^{2,14}_{14+3\delta} = 2l_{14+3\delta}F^{2,14}_{14+3\delta}$$

$$M^{2,14}_{3\delta} = 2l_{3\delta}F^{2,14}_{3\delta}$$

$$F^{2,14}_{14+3\delta} = \ln\left\{\frac{l_{14+3\delta}}{d_{2,14}} + \left[1 + \left(\frac{l_{14+3\delta}}{d_{2,14}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,14}}{l_{14+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,14}}{l_{14+3\delta}}\right)$$

$$F^{2,14}_{3\delta} = \ln\left\{\frac{l_{3\delta}}{d_{2,14}} + \left[1 + \left(\frac{l_{3\delta}}{d_{2,14}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,14}}{l_{3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,14}}{l_{3\delta}}\right)$$

EQUATION A.16 Mutual inductance between conductors 6 and 10

$$M_{6,10} = M^{6,10}_{10+\delta} - M^{6,10}_{\delta}$$

where:

$$M^{6,10}_{10+\delta} = 2l_{10+\delta}F^{6,10}_{10+\delta}$$

$$M^{6,10}_{\delta} = 2l_{\delta}F^{6,10}_{\delta}$$

$$F^{6,10}_{10+\delta} = \ln\left\{\frac{l_{10+\delta}}{d_{6,10}} + \left[1 + \left(\frac{l_{10+\delta}}{d_{6,10}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,10}}{l_{10+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,10}}{l_{10+\delta}}\right)$$

$$F^{6,10}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{6,10}} + \left[1 + \left(\frac{l_{\delta}}{d_{6,10}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,10}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,10}}{l_{\delta}}\right)$$

EQUATION A.17 Mutual inductance between conductors 6 and 14

$$M_{6,14} = M^{6,14}_{14+2\delta} - M^{6,14}_{2\delta}$$

where:

$$M^{6,14}_{14+2\delta} = 2l_{14+2\delta}F^{6,14}_{14+2\delta}$$

$$M^{6,14}_{2\delta} = 2l_{2\delta}F^{6,14}_{2\delta}$$

$$F^{6,14}_{14+2\delta} = \ln\left\{\frac{l_{14+2\delta}}{d_{6,14}} + \left[1 + \left(\frac{l_{14+2\delta}}{d_{6,14}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,14}}{l_{14+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,14}}{l_{14+2\delta}}\right)$$

$$F^{6,14}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{6,14}} + \left[1 + \left(\frac{l_{2\delta}}{d_{6,14}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,14}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,14}}{l_{2\delta}}\right)$$

EQUATION A.18 Mutual inductance between conductors 10 and 14

$$M_{10,14} = M^{10,14}_{14+\delta} - M^{10,14}_{\delta}$$

where:

$$M^{10,14}_{14+\delta} = 2l_{10+\delta}F^{10,14}_{10+\delta}$$

$$M^{10,14}_{\delta} = 2l_{\delta}F^{10,14}_{\delta}$$

$$F^{10,14}_{14+\delta} = \ln\left\{\frac{l_{14+\delta}}{d_{10,14}} + \left[1 + \left(\frac{l_{14+\delta}}{d_{10,14}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,14}}{l_{14+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,14}}{l_{14+\delta}}\right)$$

$$F^{10,14}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{10,14}} + \left[1 + \left(\frac{l_{\delta}}{d_{10,14}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,14}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,14}}{l_{\delta}}\right)$$

EQUATION A.19 Mutual inductance between conductors 4 and 8

$$M_{4,8} = M^{4,8}_{8+\delta} - M^{4,8}_{\delta}$$

where:

$$M^{4,8}_{8+\delta} = 2l_{8+\delta}F^{4,8}_{8+\delta}$$

$$M^{4,8}_{\delta} = 2l_{\delta}F^{4,8}_{\delta}$$

$$F^{4,8}_{8+\delta} = \ln\left\{\frac{l_{8+\delta}}{d_{4,8}} + \left[1 + \left(\frac{l_{8+\delta}}{d_{4,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{4,8}}{l_{8+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{4,8}}{l_{8+\delta}}\right)$$

$$F^{4,8}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{4,8}} + \left[1 + \left(\frac{l_{\delta}}{d_{4,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{4,8}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{4,8}}{l_{\delta}}\right)$$

EQUATION A.20 Mutual inductance between conductors 4 and 12

$$M_{4,12} = M^{4,12}_{12+2\delta} - M^{4,12}_{2\delta}$$

where:

$$M^{4,12}_{12+2\delta} = 2l_{12+2\delta}F^{4,12}_{12+2\delta}$$

$$M^{4,12}_{2\delta} = 2l_{2\delta}F^{4,12}_{2\delta}$$

$$F^{4,12}_{12+2\delta} = \ln\left\{\frac{l_{12+2\delta}}{d_{4,12}} + \left[1 + \left(\frac{l_{12+2\delta}}{d_{4,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{4,12}}{l_{12+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{4,12}}{l_{12+2\delta}}\right)$$

$$F^{4,12}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{4,12}} + \left[1 + \left(\frac{l_{2\delta}}{d_{4,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{4,12}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{4,12}}{l_{2\delta}}\right)$$

EQUATION A.21 Mutual inductance between conductors 4 and 16

$$M_{4,16} = M^{4,16}_{16+3\delta} - M^{4,16}_{3\delta}$$

where:

$$M^{4,16}_{16+3\delta} = 2l_{16+3\delta}F^{4,16}_{16+3\delta}$$

$$M^{4,16}_{3\delta} = 2l_{3\delta}F^{4,16}_{3\delta}$$

$$F^{4,16}_{16+3\delta} = \ln\left\{\frac{l_{16+3\delta}}{d_{4,16}} + \left[1 + \left(\frac{l_{16+3\delta}}{d_{4,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{4,16}}{l_{16+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{4,16}}{l_{16+3\delta}}\right)$$

$$F^{4,16}_{3\delta} = \ln\left\{\frac{l_{3\delta}}{d_{4,16}} + \left[1 + \left(\frac{l_{3\delta}}{d_{4,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{4,16}}{l_{3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{4,16}}{l_{3\delta}}\right)$$

EQUATION A.22 Mutual inductance between conductors 8 and 12

$$M_{8,12} = M^{8,12}_{14+\delta} - M^{8,12}_{\delta}$$

where:

$$M^{8,12}_{12+\delta} = 2l_{10+\delta}F^{8,12}_{12+\delta}$$

$$M^{8,12}_{\delta} = 2l_{\delta}F^{8,12}_{\delta}$$

$$F^{8,12}_{12+\delta} = \ln\left\{\frac{l_{12+\delta}}{d_{8,12}} + \left[1 + \left(\frac{l_{12+\delta}}{d_{8,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{8,12}}{l_{12+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{8,12}}{l_{12+\delta}}\right)$$

$$F^{8,12}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{8,12}} + \left[1 + \left(\frac{l_{\delta}}{d_{8,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{8,12}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{8,12}}{l_{\delta}}\right)$$

EQUATION A.23 Mutual inductance between conductors 8 and 16

$$M_{8,16} = M^{8,16}_{14+2\delta} - M^{8,16}_{2\delta}$$

where:

$$M^{8,16}_{16+2\delta} = 2l_{16+2\delta}F^{8,16}_{16+2\delta}$$

$$M^{8,16}_{2\delta} = 2l_{2\delta}F^{8,16}_{2\delta}$$

$$F^{8,16}_{16+2\delta} = \ln\left\{\frac{l_{16+2\delta}}{d_{8,16}} + \left[1+\left(\frac{l_{16+2\delta}}{d_{8,16}}\right)^2\right]^{1/2}\right\} - \left[1+\left(\frac{d_{8,16}}{l_{16+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{8,16}}{l_{16+2\delta}}\right)$$

$$F^{8,16}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{8,16}} + \left[1+\left(\frac{l_{2\delta}}{d_{8,16}}\right)^2\right]^{1/2}\right\} - \left[1+\left(\frac{d_{8,16}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{8,16}}{l_{2\delta}}\right)$$

EQUATION A.24 Mutual inductance between conductors 12 and 16

$$M_{12,16} = M^{12,16}_{14+\delta} - M^{12,16}_{\delta}$$

where:

$$M^{12,16}_{16+\delta} = 2l_{16+\delta}F^{12,16}_{16+\delta}$$

$$M^{12,16}_{\delta} = 2l_{\delta}F^{12,16}_{\delta}$$

$$F^{12,16}_{16+\delta} = \ln\left\{\frac{l_{16+\delta}}{d_{12,16}} + \left[1+\left(\frac{l_{16+\delta}}{d_{12,16}}\right)^2\right]^{1/2}\right\} - \left[1+\left(\frac{d_{12,16}}{l_{16+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{12,16}}{l_{16+\delta}}\right)$$

$$F^{12,16}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{12,16}} + \left[1+\left(\frac{l_{\delta}}{d_{12,16}}\right)^2\right]^{1/2}\right\} - \left[1+\left(\frac{d_{12,16}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{12,16}}{l_{\delta}}\right)$$

EQUATION A.25 Mutual inductance between conductors 1 and 3

$$M_{1,3} = M^{1,3}_{1} = M^{1,3}_{3} = 2l_1F^{1,3}_{1}$$

where:

$$F^{1,3}_{1} = \ln\left\{\frac{l_1}{d_{1,3}} + \left[1+\left(\frac{l_1}{d_{1,3}}\right)^2\right]^{1/2}\right\} - \left[1+\left(\frac{d_{1,3}}{l_1}\right)^2\right]^{1/2} + \left(\frac{d_{1,3}}{l_1}\right)$$

EQUATION A.26 Mutual inductance between conductors 1 and 7

$$M_{1,7} = M^{1,7}_{7+\delta} - M^{1,7}_{\delta}$$

where:

$$M^{1,7}_{7+\delta} = 2l_{7+\delta}F^{1,7}_{7+\delta}$$

$$M^{1,7}_{\delta} = 2l_{\delta}F^{1,7}_{\delta}$$

$$F^{1,7}_{7+\delta} = \ln\left\{\frac{l_{7+\delta}}{d_{1,7}} + \left[1+\left(\frac{l_{7+\delta}}{d_{1,7}}\right)^2\right]^{1/2}\right\} - \left[1+\left(\frac{d_{1,7}}{l_{7+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,7}}{l_{7+\delta}}\right)$$

$$F^{1,7}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{1,7}} + \left[1+\left(\frac{l_{\delta}}{d_{1,7}}\right)^2\right]^{1/2}\right\} - \left[1+\left(\frac{d_{1,7}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,7}}{l_{\delta}}\right)$$

EQUATION A.27 Mutual inductance between conductors 1 and 11

$$M_{1,11} = M^{1,11}_{11+2\delta} - M^{1,11}_{2\delta}$$

where:

$$M^{1,11}_{11+2\delta} = 2l_{11+2\delta}F^{1,11}_{11+2\delta}$$

$$M^{1,11}_{2\delta} = 2l_{2\delta}F^{1,11}_{2\delta}$$

$$F^{1,11}_{11+2\delta} = \ln\left\{\frac{l_{11+2\delta}}{d_{1,11}} + \left[1 + \left(\frac{l_{11+2\delta}}{d_{1,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,11}}{l_{11+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,11}}{l_{11+2\delta}}\right)$$

$$F^{1,11}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{1,11}} + \left[1 + \left(\frac{l_{2\delta}}{d_{1,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,11}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,11}}{l_{2\delta}}\right)$$

EQUATION A.28 Mutual inductance between conductors 1 and 15

$$M_{1,15} = M^{1,15}_{15+3\delta} - M^{1,15}_{3\delta}$$

where:

$$M^{1,15}_{15+3\delta} = 2l_{15+3\delta}F^{1,15}_{15+3\delta}$$

$$M^{1,15}_{3\delta} = 2l_{3\delta}F^{1,15}_{3\delta}$$

$$F^{1,15}_{15+3\delta} = \ln\left\{\frac{l_{15+3\delta}}{d_{1,15}} + \left[1 + \left(\frac{l_{15+3\delta}}{d_{1,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,15}}{l_{15+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,15}}{l_{15+3\delta}}\right)$$

$$F^{1,15}_{3\delta} = \ln\left\{\frac{l_{3\delta}}{d_{1,15}} + \left[1 + \left(\frac{l_{3\delta}}{d_{1,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{1,15}}{l_{3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{1,15}}{l_{3\delta}}\right)$$

EQUATION A.29 Mutual inductance between conductors 5 and 3

$$M_{5,3} = \frac{1}{2}\left\{\left(M^{5,3}_5 + M^{5,3}_3\right) - M^{5,3}_\delta\right\}$$

where:

$$M^{5,3}_5 = 2l_5F^{5,3}_5$$

$$M^{5,3}_5 = 2l_5F^{5,3}_5$$

$$M^{5,3}_\delta = 2l_\delta F^{5,3}_\delta$$

$$F^{5,3}_5 = \ln\left\{\frac{l_5}{d_{5,3}} + \left[1 + \left(\frac{l_5}{d_{5,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,3}}{l_5}\right)^2\right]^{1/2} + \left(\frac{d_{5,3}}{l_5}\right)$$

$$F^{5,3}_3 = \ln\left\{\frac{l_3}{d_{5,3}} + \left[1 + \left(\frac{l_3}{d_{5,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,3}}{l_3}\right)^2\right]^{1/2} + \left(\frac{d_{5,3}}{l_3}\right)$$

$$F^{5,3}_\delta = \ln\left\{\frac{l_\delta}{d_{5,3}} + \left[1 + \left(\frac{l_\delta}{d_{5,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,3}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{5,3}}{l_\delta}\right)$$

EQUATION A.30 Mutual inductance between conductors 5 and 7

$$M_{5,7} = \frac{1}{2}\left\{\left(M^{5,7}_5 + M^{5,7}_7\right) - M^{5,7}_\delta\right\}$$

where:

$$M^{5,7}_5 = 2l_5F^{5,7}_5$$

$$M^{5,7}_7 = 2l_7F^{5,7}_7$$

$$M^{5,7}_\delta = 2l_\delta F^{5,7}_\delta$$

$$F^{5,7}_5 = \ln\left\{\frac{l_5}{d_{5,7}} + \left[1 + \left(\frac{l_5}{d_{5,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,7}}{l_5}\right)^2\right]^{1/2} + \left(\frac{d_{5,7}}{l_5}\right)$$

$$F^{5,7}_7 = \ln\left\{\frac{l_7}{d_{5,7}} + \left[1 + \left(\frac{l_7}{d_{5,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,7}}{l_7}\right)^2\right]^{1/2} + \left(\frac{d_{5,7}}{l_7}\right)$$

$$F^{5,7}_\delta = \ln\left\{\frac{l_\delta}{d_{5,7}} + \left[1 + \left(\frac{l_\delta}{d_{5,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,7}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{5,7}}{l_\delta}\right)$$

EQUATION A.31 Mutual inductance between conductors 5 and 11

$$M_{5,11} = \frac{1}{2}\left\{\left(M_{11+2\delta}^{5,11} + M_{11+\delta}^{5,11}\right) - \left(M_{2\delta}^{5,11} + M_{\delta}^{5,11}\right)\right\}$$

where:

$$M_{11+2\delta}^{5,11} = 2l_{11+2\delta}F_{11+2\delta}^{5,11}$$

$$M_{11+\delta}^{5,11} = 2l_{11+\delta}F_{11+\delta}^{5,11}$$

$$M_{2\delta}^{5,11} = 2l_{2\delta}F_{2\delta}^{5,11}$$

$$M_{\delta}^{5,11} = 2l_{\delta}F_{\delta}^{5,11}$$

$$F_{11+2\delta}^{5,11} = \ln\left\{\frac{l_{11+2\delta}}{d_{5,11}} + \left[1 + \left(\frac{l_{11+2\delta}}{d_{5,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,11}}{l_{11+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,11}}{l_{11+2\delta}}\right)$$

$$F_{11+\delta}^{5,11} = \ln\left\{\frac{l_{11+\delta}}{d_{5,11}} + \left[1 + \left(\frac{l_{11+\delta}}{d_{5,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,11}}{l_{11+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,11}}{l_{11+\delta}}\right)$$

$$F_{2\delta}^{5,11} = \ln\left\{\frac{2\delta}{d_{5,11}} + \left[1 + \left(\frac{2\delta}{d_{5,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,11}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,11}}{l_{2\delta}}\right)$$

$$F_{\delta}^{5,11} = \ln\left\{\frac{\delta}{d_{5,11}} + \left[1 + \left(\frac{\delta}{d_{5,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,11}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{5,11}}{\delta}\right)$$

EQUATION A.32 Mutual inductance between conductors 2 and 4

$$M_{2,4} = \frac{1}{2}\left\{\left(M_{2}^{2,4} + M_{4}^{2,4}\right) - M_{\delta}^{2,4}\right\}$$

where:

$$M_{2}^{2,4} = 2l_{2}F_{2}^{2,4} \qquad M_{4}^{2,4} = 2l_{4}F_{4}^{2,4}$$

$$M_{\delta}^{2,4} = 2l_{\delta}F_{\delta}^{2,4}$$

$$F_{2}^{2,4} = \ln\left\{\frac{l_{2}}{d_{2,4}} + \left[1 + \left(\frac{l_{2}}{d_{2,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,4}}{l_{2}}\right)^2\right]^{1/2} + \left(\frac{d_{2,4}}{l_{2}}\right)$$

$$F_{4}^{2,4} = \ln\left\{\frac{l_{4}}{d_{2,4}} + \left[1 + \left(\frac{l_{4}}{d_{2,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,4}}{l_{4}}\right)^2\right]^{1/2} + \left(\frac{d_{2,4}}{l_{4}}\right)$$

$$F_{\delta}^{2,4} = \ln\left\{\frac{l_{\delta}}{d_{2,4}} + \left[1 + \left(\frac{l_{\delta}}{d_{2,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,4}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,4}}{l_{\delta}}\right)$$

EQUATION A.33 Mutual inductance between conductors 5 and 15

$$M_{5,15} = \frac{1}{2}\left\{\left(M^{5,15}_{15+3\delta} + M^{5,15}_{15+2\delta}\right) - \left(M^{5,15}_{3\delta} + M^{5,15}_{2\delta}\right)\right\}$$

where:

$$M^{5,15}_{15+3\delta} = 2l_{15+3\delta}F^{5,15}_{15+3\delta}$$

$$M^{5,15}_{15+2\delta} = 2l_{15+2\delta}F^{5,15}_{15+2\delta}$$

$$M^{5,15}_{3\delta} = 2l_{3\delta}F^{5,15}_{3\delta} \quad , \quad M^{5,15}_{2\delta} = 2l_{2\delta}F^{5,15}_{\delta}$$

$$F^{5,15}_{15+3\delta} = \ln\left\{\frac{l_{15+3\delta}}{d_{5,15}} + \left[1 + \left(\frac{l_{15+3\delta}}{d_{5,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,15}}{l_{15+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,15}}{l_{15+3\delta}}\right)$$

$$F^{5,15}_{15+2\delta} = \ln\left\{\frac{l_{15+2\delta}}{d_{5,15}} + \left[1 + \left(\frac{l_{15+2\delta}}{d_{5,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,15}}{l_{15+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,15}}{l_{15+2\delta}}\right)$$

$$F^{5,15}_{2\delta} = \ln\left\{\frac{2\delta}{d_{5,15}} + \left[1 + \left(\frac{2\delta}{d_{5,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,15}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{5,15}}{l_{2\delta}}\right)$$

$$F^{5,15}_{\delta} = \ln\left\{\frac{\delta}{d_{5,15}} + \left[1 + \left(\frac{\delta}{d_{5,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{5,15}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{5,15}}{\delta}\right)$$

EQUATION A.34 Mutual inductance between conductors 9 and 7

$$M_{9,7} = \frac{1}{2}\left\{\left(M^{9,7}_{9} + M^{9,7}_{7}\right) - M^{9,7}_{\delta}\right\}$$

where:

$$M^{9,7}_{9} = 2l_9F^{9,7}_{9} \qquad M^{9,7}_{7} = 2l_7F^{9,7}_{7}$$

$$M^{9,7}_{\delta} = 2l_{\delta}F^{9,7}_{\delta}$$

$$F^{9,7}_{9} = \ln\left\{\frac{l_9}{d_{9,7}} + \left[1 + \left(\frac{l_9}{d_{9,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,7}}{l_9}\right)^2\right]^{1/2} + \left(\frac{d_{9,7}}{l_9}\right)$$

$$F^{9,7}_{7} = \ln\left\{\frac{l_7}{d_{9,7}} + \left[1 + \left(\frac{l_7}{d_{9,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,7}}{l_7}\right)^2\right]^{1/2} + \left(\frac{d_{9,7}}{l_7}\right)$$

$$F^{9,7}_{\delta} = \ln\left\{\frac{l_\delta}{d_{9,7}} + \left[1 + \left(\frac{l_\delta}{d_{9,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,7}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{9,7}}{l_\delta}\right)$$

EQUATION A.35 Mutual inductance between conductors 9 and 3

$$M_{9,3} = \frac{1}{2}\left\{\left(M^{9,3}_{9+2\delta} + M^{9,3}_{9+\delta}\right) - \left(M^{9,3}_{2\delta} + M^{9,3}_{\delta}\right)\right\}$$

where:

$$M^{9,3}_{9+2\delta} = 2l_{9+2\delta}F^{9,3}_{9+2\delta} \quad , \quad M^{9,3}_{9+\delta} = 2l_{9+\delta}F^{9,3}_{9+\delta}$$

$$M^{9,3}_{2\delta} = 2l_{2\delta}F^{9,3}_{2\delta} \quad , \quad M^{9,3}_{\delta} = 2l_{\delta}F^{9,3}_{\delta}$$

$$F^{9,3}_{9+2\delta} = \ln\left\{\frac{l_{9+2\delta}}{d_{9,3}} + \left[1 + \left(\frac{l_{9+2\delta}}{d_{9,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,3}}{l_{9+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,3}}{l_{9+2\delta}}\right)$$

$$F^{9,3}_{9+\delta} = \ln\left\{\frac{l_{9+\delta}}{d_{9,3}} + \left[1 + \left(\frac{l_{9+\delta}}{d_{9,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,3}}{l_{9+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,3}}{l_{9+\delta}}\right)$$

$$F^{9,3}_{2\delta} = \ln\left\{\frac{2\delta}{d_{9,3}} + \left[1 + \left(\frac{2\delta}{d_{9,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,3}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,3}}{l_{2\delta}}\right)$$

$$F^{9,3}_{\delta} = \ln\left\{\frac{\delta}{d_{9,3}} + \left[1 + \left(\frac{\delta}{d_{9,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,3}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{9,3}}{\delta}\right)$$

EQUATION A.36 Mutual inductance between conductors 9 and 11

$$M_{9,11} = \frac{1}{2}\left\{\left(M^{9,11}_{9} + M^{9,11}_{11}\right) - M^{9,11}_{\delta}\right\}$$

where:

$$M^{9,11}_{9} = 2l_9F^{9,11}_{9} \qquad M^{9,11}_{7} = 2l_7F^{9,11}_{7}$$

$$M^{9,11}_{\delta} = 2l_{\delta}F^{9,11}_{\delta}$$

$$F^{9,11}_{9} = \ln\left\{\frac{l_9}{d_{9,11}} + \left[1 + \left(\frac{l_9}{d_{9,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,11}}{l_9}\right)^2\right]^{1/2} + \left(\frac{d_{9,11}}{l_9}\right)$$

$$F^{9,11}_{11} = \ln\left\{\frac{l_{11}}{d_{9,11}} + \left[1 + \left(\frac{l_{11}}{d_{9,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,11}}{l_{11}}\right)^2\right]^{1/2} + \left(\frac{d_{9,11}}{l_{11}}\right)$$

$$F^{9,11}_{\delta} = \ln\left\{\frac{l_\delta}{d_{9,11}} + \left[1 + \left(\frac{l_\delta}{d_{9,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,11}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{9,11}}{l_\delta}\right)$$

EQUATION A.37 Mutual inductance between conductors 9 and 15

$$M_{9,15} = \frac{1}{2}\left\{\left(M^{9,15}_{15+2\delta} + M^{9,15}_{15+\delta}\right) - \left(M^{9,15}_{2\delta} + M^{9,15}_{\delta}\right)\right\}$$

where:

$$M^{9,15}_{15+2\delta} = 2l_{15+2\delta}F^{9,15}_{15+2\delta} \quad , \quad M^{9,15}_{15+\delta} = 2l_{15+\delta}F^{9,15}_{15+\delta}$$

$$M^{9,15}_{2\delta} = 2l_{2\delta}F^{9,15}_{2\delta} \quad , \quad M^{9,15}_{\delta} = 2l_{\delta}F^{9,15}_{\delta}$$

$$F^{9,15}_{9+2\delta} = \ln\left\{\frac{l_{9+2\delta}}{d_{9,15}} + \left[1 + \left(\frac{l_{9+2\delta}}{d_{9,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,15}}{l_{9+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,15}}{l_{9+2\delta}}\right)$$

$$F^{9,15}_{9+\delta} = \ln\left\{\frac{l_{9+\delta}}{d_{9,15}} + \left[1 + \left(\frac{l_{9+\delta}}{d_{9,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,15}}{l_{9+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,15}}{l_{9+\delta}}\right)$$

$$F^{9,15}_{2\delta} = \ln\left\{\frac{2\delta}{d_{9,15}} + \left[1 + \left(\frac{2\delta}{d_{9,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,15}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{9,15}}{l_{2\delta}}\right)$$

$$F^{9,15}_{\delta} = \ln\left\{\frac{\delta}{d_{9,15}} + \left[1 + \left(\frac{\delta}{d_{9,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{9,15}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{9,15}}{\delta}\right)$$

EQUATION A.38 Mutual inductance between conductors 13 and 15

$$M_{13,15} = \frac{1}{2}\left\{\left(M^{13,15}_{13} + M^{13,15}_{15}\right) - M^{13,15}_{\delta}\right\}$$

where:

$$M^{13,15}_{13} = 2l_{13}F^{13,15}_{13} \quad M^{13,15}_{15} = 2l_{15}F^{13,15}_{15}$$

$$M^{13,15}_{\delta} = 2l_{\delta}F^{13,15}_{\delta}$$

$$F^{13,15}_{13} = \ln\left\{\frac{l_{13}}{d_{13,15}} + \left[1 + \left(\frac{l_{13}}{d_{13,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,15}}{l_{13}}\right)^2\right]^{1/2} + \left(\frac{d_{13,15}}{l_{13}}\right)$$

$$F^{13,15}_{13} = \ln\left\{\frac{l_{15}}{d_{13,15}} + \left[1 + \left(\frac{l_{15}}{d_{13,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,15}}{l_{15}}\right)^2\right]^{1/2} + \left(\frac{d_{13,15}}{l_{15}}\right)$$

$$F^{13,15}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{13,15}} + \left[1 + \left(\frac{l_{\delta}}{d_{13,15}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,15}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{13,15}}{l_{\delta}}\right)$$

EQUATION A.39 Mutual inductance between conductors 13 and 11

$$M_{13,11} = \frac{1}{2}\left\{\left(M^{13,11}_{13} + M^{13,11}_{11}\right) - M^{13,11}_{\delta}\right\}$$

where:

$$M^{13,11}_{13} = 2l_{13}F^{13,11}_{13} \quad M^{13,11}_{11} = 2l_{11}F^{13,11}_{11}$$

$$M^{13,11}_{\delta} = 2l_{\delta}F^{13,11}_{\delta}$$

$$F^{13,11}_{13} = \ln\left\{\frac{l_{13}}{d_{13,11}} + \left[1 + \left(\frac{l_{13}}{d_{13,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,11}}{l_{13}}\right)^2\right]^{1/2} + \left(\frac{d_{13,11}}{l_{13}}\right)$$

$$F^{13,11}_{11} = \ln\left\{\frac{l_{11}}{d_{13,11}} + \left[1 + \left(\frac{l_{11}}{d_{13,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,11}}{l_{11}}\right)^2\right]^{1/2} + \left(\frac{d_{13,11}}{l_{11}}\right)$$

$$F^{13,11}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{13,11}} + \left[1 + \left(\frac{l_{\delta}}{d_{13,11}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,11}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{13,11}}{l_{\delta}}\right)$$

EQUATION A.40 Mutual inductance between conductors 13 and 7

$$M_{13,7} = \frac{1}{2}\left\{\left(M^{13,7}_{13+2\delta} + M^{13,7}_{13+\delta}\right) - \left(M^{13,7}_{2\delta} + M^{13,7}_{\delta}\right)\right\}$$

where:

$$M^{13,7}_{13+2\delta} = 2l_{13+2\delta}F^{13,7}_{13+2\delta} \quad , \quad M^{13,7}_{13+\delta} = 2l_{13+\delta}F^{13,7}_{13+\delta}$$

$$M^{13,7}_{2\delta} = 2l_{2\delta}F^{13,7}_{2\delta} \quad , \quad M^{13,7}_{\delta} = 2l_{\delta}F^{13,7}_{\delta}$$

$$F^{13,7}_{13+2\delta} = \ln\left\{\frac{l_{13+2\delta}}{d_{13,7}} + \left[1 + \left(\frac{l_{13+2\delta}}{d_{13,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,7}}{l_{13+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{13,7}}{l_{13+2\delta}}\right)$$

$$F^{13,7}_{13+\delta} = \ln\left\{\frac{l_{13+\delta}}{d_{13,7}} + \left[1 + \left(\frac{l_{13+\delta}}{d_{13,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,7}}{l_{13+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{13,7}}{l_{13+\delta}}\right)$$

$$F^{13,7}_{2\delta} = \ln\left\{\frac{2\delta}{d_{13,7}} + \left[1 + \left(\frac{2\delta}{d_{13,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,7}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{13,7}}{l_{2\delta}}\right)$$

$$F^{13,7}_{\delta} = \ln\left\{\frac{\delta}{d_{13,7}} + \left[1 + \left(\frac{\delta}{d_{13,7}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,7}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{13,7}}{\delta}\right)$$

EQUATION A.41 Mutual inductance between conductors 13 and 3

$$M_{13,3} = \frac{1}{2}\left\{\left(M^{13,3}_{13+3\delta} + M^{13,3}_{13+2\delta}\right) - \left(M^{13,3}_{3\delta} + M^{13,3}_{2\delta}\right)\right\}$$

where:

$$M^{13,3}_{13+3\delta} = 2l_{13+3\delta}F^{13,3}_{13+3\delta}$$

$$M^{13,3}_{13+2\delta} = 2l_{13+2\delta}F^{13,3}_{13+2\delta}$$

$$M^{13,3}_{3\delta} = 2l_{3\delta}F^{13,3}_{3\delta} \quad , \quad M^{13,3}_{2\delta} = 2l_{2\delta}F^{13,3}_{\delta}$$

$$F^{13,3}_{13+3\delta} = \ln\left\{\frac{l_{13+3\delta}}{d_{13,3}} + \left[1 + \left(\frac{l_{13+3\delta}}{d_{13,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,3}}{l_{13+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{13,3}}{l_{13+3\delta}}\right)$$

$$F^{13,3}_{13+2\delta} = \ln\left\{\frac{l_{13+2\delta}}{d_{13,3}} + \left[1 + \left(\frac{l_{13+2\delta}}{d_{13,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,3}}{l_{13+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{13,3}}{l_{13+2\delta}}\right)$$

$$F^{13,3}_{2\delta} = \ln\left\{\frac{2\delta}{d_{13,3}} + \left[1 + \left(\frac{2\delta}{d_{13,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,3}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{13,3}}{l_{2\delta}}\right)$$

$$F^{13,3}_{\delta} = \ln\left\{\frac{\delta}{d_{13,3}} + \left[1 + \left(\frac{\delta}{d_{13,3}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{13,3}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{13,3}}{\delta}\right)$$

EQUATION A.42 Mutual inductance between conductors 6 and 4

$$M_{6,4} = \frac{1}{2}\left\{\left(M^{6,4}_{6} + M^{6,4}_{4}\right) - M^{6,4}_{\delta}\right\}$$

where:

$$M^{6,4}_{6} = 2l_6F^{6,4}_{6}$$

$$M^{6,4}_{4} = 2l_4F^{6,4}_{4}$$

$$M^{6,4}_{\delta} = 2l_\delta F^{6,4}_{\delta}$$

$$F^{6,4}_{6} = \ln\left\{\frac{l_6}{d_{6,4}} + \left[1 + \left(\frac{l_6}{d_{6,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,4}}{l_6}\right)^2\right]^{1/2} + \left(\frac{d_{6,4}}{l_6}\right)$$

$$F^{6,4}_{4} = \ln\left\{\frac{l_4}{d_{6,4}} + \left[1 + \left(\frac{l_4}{d_{6,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,4}}{l_4}\right)^2\right]^{1/2} + \left(\frac{d_{6,4}}{l_4}\right)$$

$$F^{6,4}_{\delta} = \ln\left\{\frac{l_\delta}{d_{6,4}} + \left[1 + \left(\frac{l_\delta}{d_{6,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,4}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{6,4}}{l_\delta}\right)$$

EQUATION A.43 Mutual inductance between conductors 2 and 8

$$M_{2,8} = \frac{1}{2}\left\{\left(M^{2,8}_{8+2\delta} + M^{2,8}_{8+\delta}\right) - \left(M^{2,8}_{2\delta} + M^{2,8}_{\delta}\right)\right\}$$

where:

$$M^{2,8}_{8+2\delta} = 2l_{8+2\delta}F^{2,8}_{8+2\delta} \quad , \quad M^{2,8}_{8+\delta} = 2l_{8+\delta}F^{2,8}_{8+\delta}$$

$$M^{2,8}_{2\delta} = 2l_{2\delta}F^{2,8}_{2\delta} \quad , \quad M^{2,8}_{\delta} = 2l_{\delta}F^{2,8}_{\delta}$$

$$F^{2,8}_{8+2\delta} = \ln\left\{\frac{l_{8+2\delta}}{d_{2,8}} + \left[1 + \left(\frac{l_{8+2\delta}}{d_{2,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,8}}{l_{8+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,8}}{l_{8+2\delta}}\right)$$

$$F^{2,8}_{8+\delta} = \ln\left\{\frac{l_{8+\delta}}{d_{2,8}} + \left[1 + \left(\frac{l_{8+\delta}}{d_{2,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,8}}{l_{8+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,8}}{l_{8+\delta}}\right)$$

$$F^{2,8}_{2\delta} = \ln\left\{\frac{2\delta}{d_{2,8}} + \left[1 + \left(\frac{2\delta}{d_{2,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,8}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,8}}{l_{2\delta}}\right)$$

$$F^{2,8}_{\delta} = \ln\left\{\frac{\delta}{d_{2,8}} + \left[1 + \left(\frac{\delta}{d_{2,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,8}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{2,8}}{\delta}\right)$$

EQUATION A.44 Mutual inductance between conductors 10 and 8

$$M_{10,8} = \frac{1}{2}\left\{\left(M^{10,8}_{10} + M^{10,8}_{8}\right) - M^{10,8}_{\delta}\right\}$$

where:

$$M^{10,8}_{6} = 2l_6F^{10,8}_{6} \quad , \quad M^{10,8}_{8} = 2l_8F^{10,8}_{8}$$

$$M^{10,8}_{\delta} = 2l_\delta F^{10,8}_{\delta}$$

$$F^{10,8}_{6} = \ln\left\{\frac{l_6}{d_{10,8}} + \left[1 + \left(\frac{l_6}{d_{10,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,8}}{l_6}\right)^2\right]^{1/2} + \left(\frac{d_{10,8}}{l_6}\right)$$

$$F^{10,8}_{8} = \ln\left\{\frac{l_8}{d_{10,8}} + \left[1 + \left(\frac{l_8}{d_{10,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,8}}{l_8}\right)^2\right]^{1/2} + \left(\frac{d_{10,8}}{l_8}\right)$$

$$F^{10,8}_{\delta} = \ln\left\{\frac{l_\delta}{d_{10,8}} + \left[1 + \left(\frac{l_\delta}{d_{10,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,8}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{10,8}}{l_\delta}\right)$$

EQUATION A.45 Mutual inductance between conductors 2 and 12

$$M_{2,12} = \frac{1}{2}\left\{\left(M^{2,12}_{12+3\delta} + M^{2,12}_{12+2\delta}\right) - \left(M^{2,12}_{3\delta} + M^{2,12}_{2\delta}\right)\right\}$$

where:

$$M^{2,12}_{12+3\delta} = 2l_{12+3\delta}F^{2,12}_{12+3\delta}$$

$$M^{2,12}_{12+2\delta} = 2l_{15+2\delta}F^{2,12}_{12+2\delta}$$

$$M^{2,12}_{3\delta} = 2l_{3\delta}F^{2,12}_{3\delta} \quad , \quad M^{2,12}_{2\delta} = 2l_{2\delta}F^{2,12}_{2\delta}$$

$$F^{2,12}_{12+3\delta} = \ln\left\{\frac{l_{12+3\delta}}{d_{2,12}} + \left[1 + \left(\frac{l_{12+3\delta}}{d_{2,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,12}}{l_{12+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,12}}{l_{12+3\delta}}\right)$$

$$F^{2,12}_{12+2\delta} = \ln\left\{\frac{l_{12+2\delta}}{d_{2,12}} + \left[1 + \left(\frac{l_{12+2\delta}}{d_{2,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,12}}{l_{12+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,12}}{l_{12+2\delta}}\right)$$

$$F^{2,12}_{2\delta} = \ln\left\{\frac{2\delta}{d_{2,12}} + \left[1 + \left(\frac{2\delta}{d_{2,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,12}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,12}}{l_{2\delta}}\right)$$

$$F^{2,12}_{2\delta} = \ln\left\{\frac{\delta}{d_{2,12}} + \left[1 + \left(\frac{\delta}{d_{2,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,12}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{2,12}}{\delta}\right)$$

EQUATION A.46 Mutual inductance between conductors 6 and 8

$$M_{6,8} = \frac{1}{2}\left\{\left(M^{6,8}_{6} + M^{6,8}_{4}\right) - M^{6,8}_{\delta}\right\}$$

where:

$$M^{6,8}_{6} = 2l_6F^{6,8}_{6}$$

$$M^{6,8}_{8} = 2l_8F^{6,8}_{8}$$

$$M^{6,8}_{\delta} = 2l_\delta F^{6,8}_{\delta}$$

$$F^{6,8}_{6} = \ln\left\{\frac{l_6}{d_{6,8}} + \left[1 + \left(\frac{l_6}{d_{6,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,8}}{l_6}\right)^2\right]^{1/2} + \left(\frac{d_{6,8}}{l_6}\right)$$

$$F^{6,8}_{8} = \ln\left\{\frac{l_8}{d_{6,8}} + \left[1 + \left(\frac{l_8}{d_{6,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,8}}{l_8}\right)^2\right]^{1/2} + \left(\frac{d_{6,8}}{l_8}\right)$$

$$F^{6,8}_{\delta} = \ln\left\{\frac{l_\delta}{d_{6,8}} + \left[1 + \left(\frac{l_\delta}{d_{6,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,8}}{l_\delta}\right)^2\right]^{1/2} + \left(\frac{d_{6,8}}{l_\delta}\right)$$

EQUATION A.47 Mutual inductance between conductors 2 and 16

$$M_{2,16} = \frac{1}{2}\left\{\left(M^{2,16}_{16+3\delta} + M^{2,16}_{16+4\delta}\right) - \left(M^{2,16}_{3\delta} + M^{2,16}_{4\delta}\right)\right\}$$

where:

$$M^{2,16}_{16+3\delta} = 2l_{16+3\delta}F^{2,16}_{16+3\delta}$$

$$M^{2,16}_{16+4\delta} = 2l_{16+4\delta}F^{2,16}_{16+4\delta}$$

$$M^{2,16}_{3\delta} = 2l_{3\delta}F^{2,16}_{3\delta} \quad , \quad M^{2,16}_{4\delta} = 2l_{4\delta}F^{2,16}_{4\delta}$$

$$F^{2,16}_{16+3\delta} = \ln\left\{\frac{l_{16+3\delta}}{d_{2,16}} + \left[1 + \left(\frac{l_{16+3\delta}}{d_{2,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,16}}{l_{16+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,16}}{l_{16+3\delta}}\right)$$

$$F^{2,16}_{16+4\delta} = \ln\left\{\frac{l_{16+4\delta}}{d_{2,16}} + \left[1 + \left(\frac{l_{16+4\delta}}{d_{2,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,16}}{l_{16+4\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,16}}{l_{16+4\delta}}\right)$$

$$F^{2,16}_{2\delta} = \ln\left\{\frac{2\delta}{d_{2,16}} + \left[1 + \left(\frac{2\delta}{d_{2,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,16}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{2,16}}{l_{2\delta}}\right)$$

$$F^{2,16}_{\delta} = \ln\left\{\frac{\delta}{d_{2,16}} + \left[1 + \left(\frac{\delta}{d_{2,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{2,16}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{2,16}}{\delta}\right)$$

EQUATION A.48 Mutual inductance between conductors 14 and 16

$$M_{14,16} = \frac{1}{2}\left\{\left(M_{14}^{14,16} + M_{12}^{14,16}\right) - M_{\delta}^{14,16}\right\}$$

where:

$$M_{14}^{14,16} = 2l_{14}F_{14}^{14,16} \quad , \quad M_{12}^{14,16} = 2l_{12}F_{12}^{14,16}$$

$$M_{\delta}^{14,16} = 2l_{\delta}F_{\delta}^{14,16}$$

$$F_{14}^{14,16} = \ln\left\{\frac{l_{14}}{d_{14,16}} + \left[1 + \left(\frac{l_{14}}{d_{14,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,16}}{l_{14}}\right)^2\right]^{1/2} + \left(\frac{d_{14,16}}{l_{14}}\right)$$

$$F_{16}^{14,16} = \ln\left\{\frac{l_{16}}{d_{14,16}} + \left[1 + \left(\frac{l_{16}}{d_{14,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,16}}{l_{16}}\right)^2\right]^{1/2} + \left(\frac{d_{14,16}}{l_{16}}\right)$$

$$F_{\delta}^{14,16} = \ln\left\{\frac{l_{\delta}}{d_{14,16}} + \left[1 + \left(\frac{l_{\delta}}{d_{14,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,16}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,16}}{l_{\delta}}\right)$$

EQUATION A.49 Mutual inductance between conductors 6 and 12

$$M_{6,12} = \frac{1}{2}\left\{\left(M_{12+2\delta}^{6,12} + M_{12+\delta}^{6,12}\right) - \left(M_{2\delta}^{6,12} + M_{\delta}^{6,12}\right)\right\}$$

where:

$$M_{12+2\delta}^{6,12} = 2l_{12+2\delta}F_{12+2\delta}^{6,12} \quad , \quad M_{12+\delta}^{6,12} = 2l_{12+\delta}F_{12+\delta}^{6,12}$$

$$M_{2\delta}^{6,12} = 2l_{2\delta}F_{2\delta}^{6,12} \quad , \quad M_{\delta}^{6,12} = 2l_{\delta}F_{\delta}^{6,12}$$

$$F_{12+2\delta}^{6,12} = \ln\left\{\frac{l_{12+2\delta}}{d_{6,12}} + \left[1 + \left(\frac{l_{12+2\delta}}{d_{6,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,12}}{l_{12+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,12}}{l_{12+2\delta}}\right)$$

$$F_{12+\delta}^{6,12} = \ln\left\{\frac{l_{12+\delta}}{d_{6,12}} + \left[1 + \left(\frac{l_{12+\delta}}{d_{6,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,12}}{l_{12+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,12}}{l_{12+\delta}}\right)$$

$$F_{2\delta}^{6,12} = \ln\left\{\frac{2\delta}{d_{6,12}} + \left[1 + \left(\frac{2\delta}{d_{6,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,12}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,12}}{l_{2\delta}}\right)$$

$$F_{\delta}^{6,12} = \ln\left\{\frac{\delta}{d_{6,12}} + \left[1 + \left(\frac{\delta}{d_{6,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,12}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{6,12}}{\delta}\right)$$

EQUATION A.50 Mutual inductance between conductors 10 and 12

$$M_{10,12} = \frac{1}{2}\left\{\left(M_{10}^{10,12} + M_{12}^{10,12}\right) - M_{\delta}^{10,12}\right\}$$

where:

$$M_{10}^{10,12} = 2l_{10}F_{10}^{10,12} \quad , \quad M_{12}^{10,12} = 2l_{12}F_{12}^{10,12}$$

$$M_{\delta}^{10,8} = 2l_{\delta}F_{\delta}^{10,12}$$

$$F_{10}^{10,12} = \ln\left\{\frac{l_{10}}{d_{10,12}} + \left[1 + \left(\frac{l_{10}}{d_{10,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,12}}{l_{10}}\right)^2\right]^{1/2} + \left(\frac{d_{10,12}}{l_{10}}\right)$$

$$F_{12}^{10,12} = \ln\left\{\frac{l_{12}}{d_{10,12}} + \left[1 + \left(\frac{l_{12}}{d_{10,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,12}}{l_{12}}\right)^2\right]^{1/2} + \left(\frac{d_{10,12}}{l_{12}}\right)$$

$$F_{\delta}^{10,12} = \ln\left\{\frac{l_{\delta}}{d_{10,12}} + \left[1 + \left(\frac{l_{\delta}}{d_{10,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,12}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,12}}{l_{\delta}}\right)$$

EQUATION A.51 Mutual inductance between conductors 6 and 16

$$M_{6,16} = \frac{1}{2}\left\{\left(M^{6,16}_{16+3\delta} + M^{6,16}_{16+2\delta}\right) - \left(M^{6,16}_{3\delta} + M^{6,16}_{2\delta}\right)\right\}$$

where:

$$M^{6,16}_{16+3\delta} = 2l_{16+3\delta}F^{6,16}_{16+3\delta}$$

$$M^{6,16}_{16+2\delta} = 2l_{16+2\delta}F^{6,16}_{16+2\delta}$$

$$M^{6,16}_{3\delta} = 2l_{3\delta}F^{6,16}_{3\delta} \quad , \quad M^{6,16}_{2\delta} = 2l_{2\delta}F^{6,16}_{2\delta}$$

$$F^{6,16}_{16+3\delta} = \ln\left\{\frac{l_{16+3\delta}}{d_{6,16}} + \left[1 + \left(\frac{l_{16+3\delta}}{d_{6,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,16}}{l_{16+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,16}}{l_{16+3\delta}}\right)$$

$$F^{6,16}_{16+2\delta} = \ln\left\{\frac{l_{16+2\delta}}{d_{6,16}} + \left[1 + \left(\frac{l_{16+2\delta}}{d_{6,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,16}}{l_{16+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,16}}{l_{16+2\delta}}\right)$$

$$F^{6,16}_{3\delta} = \ln\left\{\frac{l_{3\delta}}{d_{6,16}} + \left[1 + \left(\frac{l_{3\delta}}{d_{6,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,16}}{l_{3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,16}}{l_{3\delta}}\right)$$

$$F^{6,16}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{6,16}} + \left[1 + \left(\frac{l_{2\delta}}{d_{6,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{6,16}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{6,16}}{l_{2\delta}}\right)$$

EQUATION A.52 Mutual inductance between conductors 10 and 4

$$M_{10,4} = \frac{1}{2}\left\{\left(M^{10,4}_{12+2\delta} + M^{10,4}_{12+\delta}\right) - \left(M^{10,4}_{2\delta} + M^{10,4}_{\delta}\right)\right\}$$

where:

$$M^{10,4}_{10+2\delta} = 2l_{10+2\delta}F^{10,4}_{10+2\delta} \quad , \quad M^{10,4}_{\delta} = 2l_{\delta}F^{10,4}_{\delta}$$

$$M^{10,4}_{2\delta} = 2l_{2\delta}F^{10,4}_{2\delta} \quad , \quad M^{10,4}_{10+\delta} = 2l_{10+\delta}F^{10,4}_{10+\delta}$$

$$F^{10,4}_{10+2\delta} = \ln\left\{\frac{l_{10+2\delta}}{d_{10,4}} + \left[1 + \left(\frac{l_{10+2\delta}}{d_{10,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,4}}{l_{10+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,4}}{l_{10+2\delta}}\right)$$

$$F^{10,4}_{10+\delta} = \ln\left\{\frac{l_{10+\delta}}{d_{10,4}} + \left[1 + \left(\frac{l_{10+\delta}}{d_{10,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,4}}{l_{10+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,4}}{l_{10+\delta}}\right)$$

$$F^{10,4}_{2\delta} = \ln\left\{\frac{2\delta}{d_{10,4}} + \left[1 + \left(\frac{2\delta}{d_{10,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,4}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,4}}{l_{2\delta}}\right)$$

$$F^{10,4}_{\delta} = \ln\left\{\frac{\delta}{d_{10,4}} + \left[1 + \left(\frac{\delta}{d_{10,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,4}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{10,4}}{\delta}\right)$$

EQUATION A.53 Mutual inductance between conductors 10 and 8

$$M_{14,12} = \frac{1}{2}\left\{\left(M_{14}^{14,12} + M_{12}^{14,12}\right) - M_{\delta}^{14,12}\right\}$$

where:

$$M_{14}^{14,12} = 2l_{14}F_{14}^{14,12} \quad , \quad M_{12}^{14,12} = 2l_{12}F_{12}^{14,12}$$

$$M_{\delta}^{14,12} = 2l_{\delta}F_{\delta}^{14,12}$$

$$F_{14}^{14,12} = \ln\left\{\frac{l_{14}}{d_{14,12}} + \left[1 + \left(\frac{l_{14}}{d_{14,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,12}}{l_{14}}\right)^2\right]^{1/2} + \left(\frac{d_{14,12}}{l_{14}}\right)$$

$$F_{12}^{14,12} = \ln\left\{\frac{l_{12}}{d_{10,8}} + \left[1 + \left(\frac{l_{12}}{d_{14,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,12}}{l_{12}}\right)^2\right]^{1/2} + \left(\frac{d_{14,12}}{l_{12}}\right)$$

$$F_{\delta}^{14,12} = \ln\left\{\frac{l_{\delta}}{d_{14,12}} + \left[1 + \left(\frac{l_{\delta}}{d_{14,12}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,12}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,12}}{l_{\delta}}\right)$$

EQUATION A.54 Mutual inductance between conductors 10 and 16

$$M_{10,16} = \frac{1}{2}\left\{\left(M_{16+2\delta}^{10,16} + M_{16+\delta}^{10,16}\right) - \left(M_{2\delta}^{10,16} + M_{\delta}^{10,16}\right)\right\}$$

where:

$$M_{16+2\delta}^{10,16} = 2l_{16+2\delta}F_{16+2\delta}^{10,16} \, , \quad M_{16+\delta}^{10,16} = 2l_{16+\delta}F_{16+\delta}^{10,16}$$

$$M_{2\delta}^{10,16} = 2l_{2\delta}F_{2\delta}^{10,16} \quad , \quad M_{\delta}^{10,16} = 2l_{\delta}F_{\delta}^{10,16}$$

$$F_{16+2\delta}^{10,16} = \ln\left\{\frac{l_{16+2\delta}}{d_{10,16}} + \left[1 + \left(\frac{l_{16+2\delta}}{d_{10,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,16}}{l_{16+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,16}}{l_{16+2\delta}}\right)$$

$$F_{16+\delta}^{10,16} = \ln\left\{\frac{l_{16+\delta}}{d_{10,16}} + \left[1 + \left(\frac{l_{16+\delta}}{d_{10,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,16}}{l_{16+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,16}}{l_{16+\delta}}\right)$$

$$F_{2\delta}^{10,16} = \ln\left\{\frac{l_{2\delta}}{d_{10,16}} + \left[1 + \left(\frac{l_{2\delta}}{d_{10,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,16}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{10,16}}{l_{2\delta}}\right)$$

$$F_{\delta}^{10,16} = \ln\left\{\frac{\delta}{d_{10,16}} + \left[1 + \left(\frac{\delta}{d_{10,16}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{10,16}}{\delta}\right)^2\right]^{1/2} + \left(\frac{d_{10,16}}{\delta}\right)$$

EQUATION A.55 Mutual inductance between conductor 1 and other conductors

$$M_{1,3} = M_1^{1,3} = M_3^{1,3} = 2l_1F_1^{1,3}$$

$$M_{1,5} = \frac{1}{2}\{(M_1^{1,5} + M_5^{1,5}) - M_d^{1,5}\}$$

$$M_{1,7} = M_{7+d}^{1,7} - M_d^{1,7}$$

$$M_{1,9} = \frac{1}{2}\{(M_{9+2d}^{1,9} + M_{9+d}^{1,9}) - (M_{2d}^{1,9} + M_d^{1,9})\}$$

$$M_{1,11} = M_{11+2d}^{1,11} - M_{2d}^{1,11}$$

$$M_{1,13} = \frac{1}{2}\{(M_{13+3d}^{1,13} + M_{13+2d}^{1,13}) - (M_{3d}^{1,13} + M_{2d}^{1,13})\}$$

$$M_{1,15} = M_{15+3d}^{1,15} - M_{3d}^{1,15}$$

EQUATION A.56 Mutual inductance between conductors 14 and 4

$$M_{14,4} = \frac{1}{2}\left\{\left(M^{14,4}_{14+3\delta} + M^{14,4}_{14+2\delta}\right) - \left(M^{14,4}_{3\delta} + M^{14,4}_{2\delta}\right)\right\}$$

where:

$$M^{14,4}_{16+3\delta} = 2l_{16+3\delta}F^{14,4}_{16+3\delta}$$

$$M^{14,4}_{16+2\delta} = 2l_{16+2\delta}F^{14,4}_{16+2\delta}$$

$$M^{14,4}_{3\delta} = 2l_{3\delta}F^{14,4}_{3\delta} \quad , \quad M^{14,4}_{2\delta} = 2l_{2\delta}F^{14,4}_{2\delta}$$

$$F^{14,4}_{14+3\delta} = \ln\left\{\frac{l_{14+3\delta}}{d_{14,4}} + \left[1 + \left(\frac{l_{14+3\delta}}{d_{14,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,4}}{l_{14+3\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,4}}{l_{14+3\delta}}\right)$$

$$F^{14,4}_{14+2\delta} = \ln\left\{\frac{l_{14+2\delta}}{d_{14,4}} + \left[1 + \left(\frac{l_{14+2\delta}}{d_{14,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,4}}{l_{14+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,4}}{l_{14+2\delta}}\right)$$

$$F^{14,4}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{14,4}} + \left[1 + \left(\frac{l_{2\delta}}{d_{14,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,4}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,4}}{l_{2\delta}}\right)$$

$$F^{14,4}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{14,4}} + \left[1 + \left(\frac{l_{2\delta}}{d_{14,4}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,4}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,4}}{l_{2\delta}}\right)$$

EQUATION A.57 Mutual inductance between conductor 2 and other conductors

$$M_{2,6} = M^{2,6}_{6+d} - M^{2,6}_{d}$$

$$M_{2,4} = \frac{1}{2}\{(M^{2,4}_{2} + M^{2,4}_{4}) - M^{2,4}_{d}\}$$

$$M_{2,10} = M^{2,10}_{10+2d} - M^{2,10}_{2d}$$

$$M_{2,12} = \frac{1}{2}\{(M^{2,12}_{12+2d} + M^{2,12}_{12+3d}) - (M^{2,12}_{2d} + M^{2,12}_{3d})\}$$

$$M_{2,14} = M^{2,14}_{14+3d} - M^{2,14}_{3d}$$

$$M_{2,16} = \frac{1}{2}\{(M^{2,16}_{16+3d} + M^{2,16}_{16+4d}) - (M^{2,16}_{3d} + M^{2,16}_{2d})\}$$

$$M_{2,8} = \frac{1}{2}\{(M^{2,8}_{8+2\delta} + M^{2,8}_{8+\delta}) - (M^{2,8}_{2\delta} + M^{2,8}_{\delta})\}$$

EQUATION A.58 Mutual inductance between conductors 14 and 8

$$M_{14,8} = \frac{1}{2}\left\{\left(M^{14,8}_{14+2\delta} + M^{14,8}_{14+\delta}\right) - \left(M^{14,8}_{2\delta} + M^{14,8}_{\delta}\right)\right\}$$

where:

$$M^{14,8}_{14+2\delta} = 2l_{14+2\delta}F^{14,8}_{14+2\delta} \quad , \quad M^{14,8}_{14+\delta} = 2l_{14+\delta}F^{14,8}_{14+\delta}$$

$$M^{14,8}_{2\delta} = 2l_{2\delta}F^{14,8}_{2\delta} \quad , \quad M^{14,8}_{\delta} = 2l_{\delta}F^{14,8}_{\delta}$$

$$F^{14,8}_{14+2\delta} = \ln\left\{\frac{l_{14+2\delta}}{d_{14,8}} + \left[1 + \left(\frac{l_{14+2\delta}}{d_{14,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,8}}{l_{14+2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,8}}{l_{14+2\delta}}\right)$$

$$F^{14,8}_{14+\delta} = \ln\left\{\frac{l_{14+\delta}}{d_{14,8}} + \left[1 + \left(\frac{l_{14+\delta}}{d_{14,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,8}}{l_{14+\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,8}}{l_{14+\delta}}\right)$$

$$F^{14,8}_{2\delta} = \ln\left\{\frac{l_{2\delta}}{d_{14,8}} + \left[1 + \left(\frac{l_{2\delta}}{d_{14,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,8}}{l_{2\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,8}}{l_{2\delta}}\right)$$

$$F^{14,8}_{\delta} = \ln\left\{\frac{l_{\delta}}{d_{14,8}} + \left[1 + \left(\frac{l_{\delta}}{d_{14,8}}\right)^2\right]^{1/2}\right\} - \left[1 + \left(\frac{d_{14,8}}{l_{\delta}}\right)^2\right]^{1/2} + \left(\frac{d_{14,8}}{l_{\delta}}\right)$$

EQUATION A.59 Mutual inductance between conductor 5 and other conductors

$$M_{5,9} = M^{5,9}_{9+d} - M^{5,9}_{d}$$

$$M_{5,7} = \frac{1}{2}\{(M^{5,7}_{5} + M^{5,7}_{7}) - M^{5,7}_{d}\}$$

$$M_{5,3} = \frac{1}{2}\{(M^{5,3}_{5} + M^{5,3}_{3}) - M^{5,3}_{d}\}$$

$$M_{5,11} = \frac{1}{2}\{(M^{5,11}_{11+d} + M^{5,11}_{11+2d}) - (M^{5,11}_{d} + M^{5,11}_{2d})\}$$

$$M_{5,13} = M^{5,13}_{13+2d} - M^{5,13}_{2d}$$

$$M_{5,15} = \frac{1}{2}\{(M^{5,15}_{15+2d} + M^{5,15}_{15+3d}) - (M^{5,15}_{3d} + M^{5,15}_{2d})\}$$

EQUATION A.60 Mutual inductance between conductor 9 and other conductors

$$M_{9,3} = \frac{1}{2}\{(M_{9+2d}^{9,3} + M_{9+d}^{9,3}) - (M_{2d}^{9,3} + M_{d}^{9,3})\}$$
$$M_{9,7} = \frac{1}{2}\{(M_{9}^{9,7} + M_{7}^{9,7}) - M_{d}^{9,7}\}$$
$$M_{9,11} = \frac{1}{2}\{(M_{9}^{9,11} + M_{11}^{9,11}) - M_{d}^{9,11}\}$$
$$M_{9,13} = M_{13+d}^{9,13} - M_{d}^{9,13}$$

EQUATION A.61 Mutual inductance between conductor 13 and other conductors

$$M_{13,3} = \frac{1}{2}\{(M_{13+3d}^{13,3} + M_{13+2d}^{13,3}) - (M_{3d}^{13,3} + M_{2d}^{13,3})\}$$
$$M_{13,7} = \frac{1}{2}\{(M_{13+2d}^{13,7} + M_{13+d}^{13,7}) - (M_{2d}^{13,7} + M_{d}^{13,7})\}$$
$$M_{13,11} = \frac{1}{2}\{(M_{13}^{13,11} + M_{11}^{13,11}) - M_{d}^{13,11}\}$$
$$M_{13,15} = \frac{1}{2}\{(M_{13}^{13,15} + M_{15}^{13,15}) - M_{d}^{13,15}\}$$

EQUATION A.62 Mutual inductance between conductors 15, 11, 7 and other conductors

$$M_{15,11} = M_{15+d}^{15,11} - M_{d}^{15,11}$$
$$M_{15,7} = M_{15+2d}^{15,7} - M_{2d}^{15,7}$$
$$M_{15,3} = M_{15+3d}^{15,3} - M_{3d}^{15,3}$$
$$M_{11,7} = M_{11+d}^{11,7} - M_{d}^{11,7}$$
$$M_{11,3} = M_{11+2d}^{11,3} - M_{2d}^{11,3}$$
$$M_{7,3} = M_{7+d}^{7,3} - M_{d}^{7,3}$$

EQUATION A.63 Mutual inductance between conductor 6 and other conductors

$$M_{6,10} = M_{10+d}^{6,10} - M_{d}^{6,10}$$
$$M_{6,14} = M_{14+2d}^{6,14} - M_{2d}^{6,14}$$
$$M_{6,16} = \frac{1}{2}\{(M_{16+2d}^{6,16} + M_{16+3d}^{6,16}) - (M_{2d}^{6,16} + M_{3d}^{6,16})\}$$
$$M_{6,12} = \frac{1}{2}\{(M_{12+2\delta}^{6,12} + M_{12+\delta}^{6,12}) - (M_{2\delta}^{6,12} + M_{\delta}^{6,12})\}$$
$$M_{6,8} = \frac{1}{2}\{(M_{6}^{6,8} + M_{8}^{6,8}) - M_{d}^{6,8}\}$$
$$M_{6,4} = \frac{1}{2}\{(M_{6}^{6,4} + M_{4}^{6,4}) - M_{d}^{6,4}\}$$

EQUATION A.64 Mutual inductance between conductor 10 and other conductors

$$M_{10,14} = M_{14+d}^{10,14} - M_{d}^{10,14}$$
$$M_{10,16} = \frac{1}{2}\{(M_{16+2d}^{10,16} + M_{16+d}^{10,16}) - (M_{2d}^{10,16} + M_{d}^{10,16})\}$$
$$M_{10,12} = \frac{1}{2}\{(M_{10}^{10,12} + M_{12}^{10,12}) - M_{d}^{10,12}\}$$
$$M_{10,8} = \frac{1}{2}\{(M_{10}^{10,8} + M_{8}^{10,8}) - M_{d}^{10,8}\}$$
$$M_{10,4} = \frac{1}{2}\{(M_{10+d}^{10,4} + M_{10+2d}^{10,4}) - (M_{d}^{10,4} + M_{2d}^{10,4})\}$$

EQUATION A.65 Mutual inductance between conductors 16, 12, 8 and other conductors

$$M_{16,12} = M_{16+d}^{16,12} - M_{d}^{16,12}$$
$$M_{16,8} = M_{16+2d}^{16,8} - M_{2d}^{16,8}$$
$$M_{16,4} = M_{16+3d}^{16,4} - M_{3d}^{16,4}$$
$$M_{12,8} = M_{12+d}^{12,8} - M_{d}^{12,8}$$
$$M_{12,4} = M_{12+2d}^{12,4} - M_{2d}^{12,4}$$
$$M_{8,4} = M_{8+d}^{8,4} - M_{d}^{8,4}$$

APPENDIX B: MATHLAB PROGRAM EXAMPLE FOR EXAMPLE 8

```
% One_turn.m
% Inductance calculation with mutual inductance terms
% for 1 turn rectangular shape.
% Inductor type = Etched MCRF450 reader antenna
%
% Youbok Lee
%
% Microchip Technology Inc.
%------------------------------------------
% L_T = L_o + M_+ M_-  (nH)
% unit = cm
% where
% L_o = L1 + L2 + L3+ L4 = (self inductance)
% M_- = Negative mutual inductance
% M_+ = positive mutual inductance = 0 for 1 turn coil
%
%------------- Length of each conductor -------------------
% l_1a = l_1b = 3" = 7.62 Cm
% l_2 = l_4 = 10" = 25.4 Cm
% l_4 = 7.436" = 18.887 Cm
% gap = 3.692 cm
%-------Define segment length (cm) ---------------------
w = 0.508
t = 0.0001
gap = 3.692
l_1A = 7.62 - w/2.
l_1B = 7.62 - w/2.
l_2 = 25.4  - w
l_3 = 18.887  - w
l_4 = 25.4  - w
%---------- distance between branches (cm) -------
d13 = l_2
d24 = l_3
%----------calculate self inductance ---------
L1A = 2*l_1A*(log((2*l_1A)/(w+t)) + 0.50049 + (w+t)/(3*l_1A))
L1B = 2*l_1B*(log((2*l_1B)/(w+t)) + 0.50049 + (w+t)/(3*l_1B))

L2 = 2*l_2*(log((2*l_2)/(w+t)) + 0.50049 + (w+t)/(3*l_2))
L3 = 2*l_3*(log((2*l_3)/(w+t)) + 0.50049 + (w+t)/(3*l_3))

L4 = 2*l_4*(log((2*l_4)/(w+t)) + 0.50049 + (w+t)/(3*l_4))
```

```
L_o = L1A + L1B + L2 + L3 + L4
%-------- calculate mutual inductance parameters ----
Q1A_3 =log((l_1A/d13)+(1+(l_1A/d13)^2)^0.5)-(1+(d13/l_1A)^2)^0.5 + (d13/l_1A)
Q1B_3 =log((l_1B/d13)+(1+(l_1B/d13)^2)^0.5)-(1+(d13/l_1B)^2)^0.5 + (d13/l_1B)

Q_1A_gap =log(((l_1A+gap)/d13)+(1+((l_1A+gap)/d13)^2)^0.5)-(1+(d13/(l_1A+gap))^2)^0.5 + (d13/(l_1A+gap))
Q_1B_gap =log(((l_1B+gap)/d13)+(1+((l_1B+gap)/d13)^2)^0.5)-(1+(d13/(l_1B+gap))^2)^0.5 + (d13/(l_1B+gap))


Q3 =log((l_3/d13)+(1+(l_3/d13)^2)^0.5)-(1+(d13/l_3)^2)^0.5 + (d13/l_3)

Q2_4 =log((l_2/d24)+(1+(l_2/d24)^2)^0.5)-(1+(d24/l_2)^2)^0.5 + (d24/l_2)

%--------- calculate negative mutual inductance --------
%
M1A = 2*l_1A*Q1A_3
M1B = 2*l_1B*Q1B_3
M1A_gap = 2*(l_1A+gap)*Q_1A_gap
M1B_gap = 2*(l_1B+gap)*Q_1B_gap
M3 = 2*l_3*Q3

M1A_3 = (M1A+M3 - M1B_gap)/2.
M1B_3 = (M1B+M3 - M1A_gap)/2.
M2_4 = 2* (l_2*Q2_4)

M_T = 2* (M1A_3 + M1B_3 + M2_4)
%--------- Total Inductance (nH) -----------
L_T = L_o - M_T
```

REFERENCES

[1] V. G. Welsby, The Theory and Design of Inductance Coils, John Wiley and Sons, Inc., 1960.

[2] Frederick W. Grover, Inductance Calculations Working Formulas and Tables, Dover Publications, Inc., New York, NY., 1946.

[3] Keith Henry, Editor, Radio Engineering Handbook, McGraw-Hill Book Company, New York, NY., 1963.

[4] H.M. Greenhouse, IEEE Transaction on Parts, Hybrid, and Packaging, Vol. PHP-10, No. 2, June 1974.

[5] K. Fujimoto, A. Henderson, K. Hirasawa, and J.R. James, Small Antennas, John Wiley & Sons Inc., ISBN 0471 914134, 1987

[6] James K. Hardy, High Frequency Circuit Design, Reston Publishing Company, Inc.Reston, Virginia, 1975.

[7] Simon Ramo, Fields and Waves in Communication Electronics, John Wiley, 1984.

NOTES:

FSK Reader Reference Design

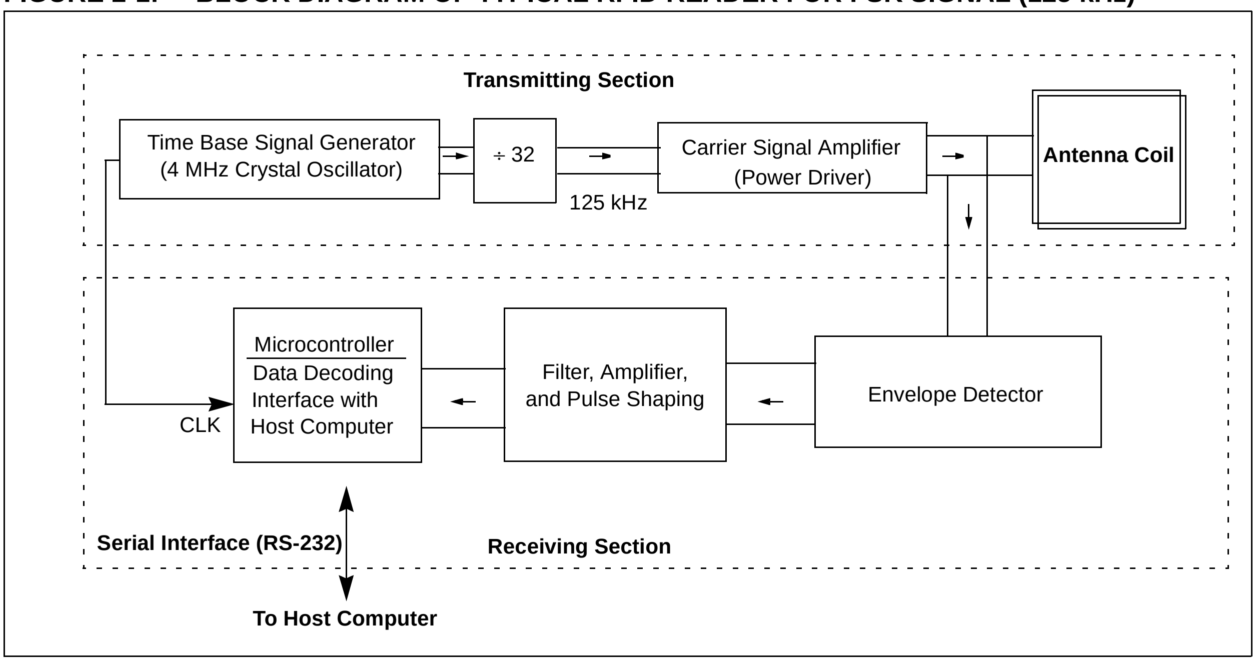
1.0 INTRODUCTION

This application note is written as a reference guide for FSK reader designers. Microchip Technology Inc. provides basic reader electronics circuitry for the MCRF200 customers as a part of this design guide. The circuit is designed for a read range of 3 ~ 5 inches with an access control card. The microID FSK Reader (demo unit), which is built based on the FSK reference design, is available in the microID Designers Kit (DV103001). The circuit can be modified for longer read range or other applications with the MCRF200. An electronic copy of the FSK microID PICmicro® source code is available upon request.

2.0 READER CIRCUITS

The RFID reader consists of transmitting and receiving sections. It transmits a carrier signal, receives the backscattering signal, and performs data processing. The reader also communicates with an external host computer. A basic block diagram of the typical RFID reader is shown in Figure 2-1.

FIGURE 2-1: BLOCK DIAGRAM OF TYPICAL RFID READER FOR FSK SIGNAL (125 kHz)



PICmicro is a registered trademark of Microchip Technology Inc.

2.1 Transmitting Section

The transmitting section contains circuitry for a carrier signal (125 kHz), power amplifiers, and a tuned antenna coil.

The 125 kHz carrier signal is typically generated by dividing a 4 MHz (4 MHz/32 = 125 kHz) crystal oscillator signal. The signal is amplified before it is fed into the antenna tuning circuit. A complementary power amplifier circuit is typically used to boost the transmitting signal level.

An antenna impedance tuning circuit consisting of capacitors is used to maximize the signal level at the carrier frequency. This tuning circuit is also needed to form an exact LC resonant circuit for the carrier signal. The tuning compensates the variations in the component values and the perturbation of coil inductance due to environment effect. A design guide for the antenna coil is given in *AN710, Antenna Circuit Design for RFID Applications* (DS00710).

2.1.1 LIMITS ON TRANSMITTING SIGNAL LEVEL (FCC PART 15) IN THE USA

Each country limits the signal strength of the RF wave that is intentionally radiated by a device. In the USA, the signal strength of the carrier signal (125 kHz) radiating from the antenna coil must comply with the FCC (Federal Communications Commission) part 15 regulation. The signal level is specified by the 47 CFR Part 15.209a of the federal regulation. For a 125 kHz signal, the FCC limits the signal level to 19.2 μv per meter, or 25.66 dBμV (i.e., 20 log(19.2) = 25.66 dBμV), at 300 meters away from the antenna. For a close distance measurement, an extrapolation rule (40 dB per decade) is applied (Part 15.31.f.2). For example, the signal level at 30 meters away from the device must not exceed:

$$25.66\ dB\mu V + 40\ dB\mu V = 65.66\ dB\mu V$$

2.2 Receiving Section

The receiving section consists of an antenna coil, demodulator, filters, amplifiers, and microcontroller. In applications for close proximity read range, a single coil is often used for both transmitting and receiving. For long read-range applications, however, separated antennas may be used. More details on the antenna coil are given in *AN710, Antenna Circuit Design for RFID Applications* (DS00710).

In the FSK communication protocol, a ‘0’ and a ‘1’ are represented by two different frequencies. In the MCRF200, a ‘0’ and a ‘1’ are represented by Fc/8 and Fc/10, respectively. Fc is the carrier frequency. The MCRF200 sends this FSK signal to the reader by an amplitude modulation of the carrier signal.

The FSK reader needs two steps for a full recovery of the data. The first step is demodulating the backscattering signal, and the second step is detecting the frequency (or period) of the demodulation signal.

The demodulation is accomplished by detecting the envelope of the carrier signal. A half-wave capacitor-filtered rectifier circuit is used for the demodulation process. A diode detects the peak voltage of the backscattering signal. The voltage is then fed into an RC charging/discharging circuit. The RC time constant must be small enough to allow the voltage across C to fall fast enough to keep in step with the envelope. However, the time constant must not be so small as to introduce excessive ripple. The demodulated signal must then pass through a filter and signal shaping circuit before it is fed to the microcontroller. The microcontroller performs data decoding and communicates with the host computer through an RS-232 or other serial interface protocols.

3.0 microID FSK READER

The electronic circuitry for an FSK reader is shown in Figure 3-1. The reader needs +9 VDC power supply. The 125 kHz carrier signal is generated by dividing the 4 MHz time base signal that is generated by a crystal oscillator. A 16-stage binary ripple counter (74HC4060) is used for this purpose. The 74HC4060 also provides a clock signal for the PIC16C84 microcontroller. The 125 kHz signal is passed to an RF choke (L1) and filter before it is fed into a power amplifier that is formed by a pair of complementary bipolar transistors (Q2 and Q3).

For long read-range applications, this power amplifier circuit can be modified. Power MOSFETs may be used instead of the bipolar transistors (2N2222). These power MOSFETs can be driven by +24 VDC power supply. A push-pull predriver can be added at the front of the complementary circuit. This modification will enhance the signal level of the carrier signal.

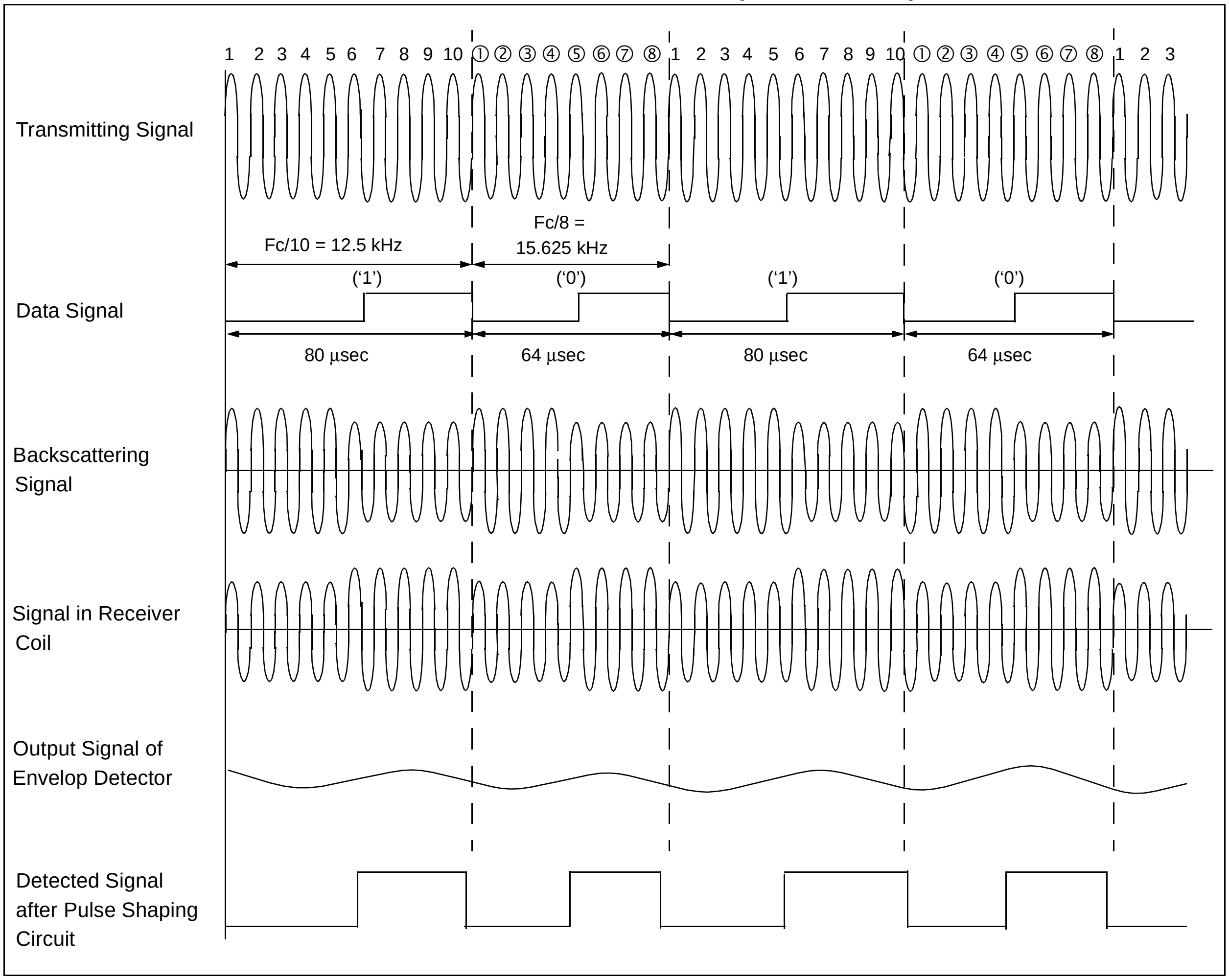
The reader circuit uses a single coil for both transmitting and receiving signals. An antenna coil (L2: 1.62 mH) and a resonant capacitor (C2: 1000 pF) forms a series resonant circuit for a 125 kHz resonance frequency. Since the C2 is grounded, the carrier signal (125 kHz) is filtered out to ground after passing the antenna coil. The circuit provides a minimum impedance at the resonance frequency. This results in maximizing the antenna current, and therefore, the magnetic field strength is maximized.

L2, C15, D7, and the other bottom parts in the circuit form a signal receiving section. The voltage drop in the antenna coil is a summation (superposition) of transmitting signal and backscattering signal. The D7 is a demodulator which detects the envelope of the backscattering signal. The FSK signal waveforms are shown in Figure 3-1.

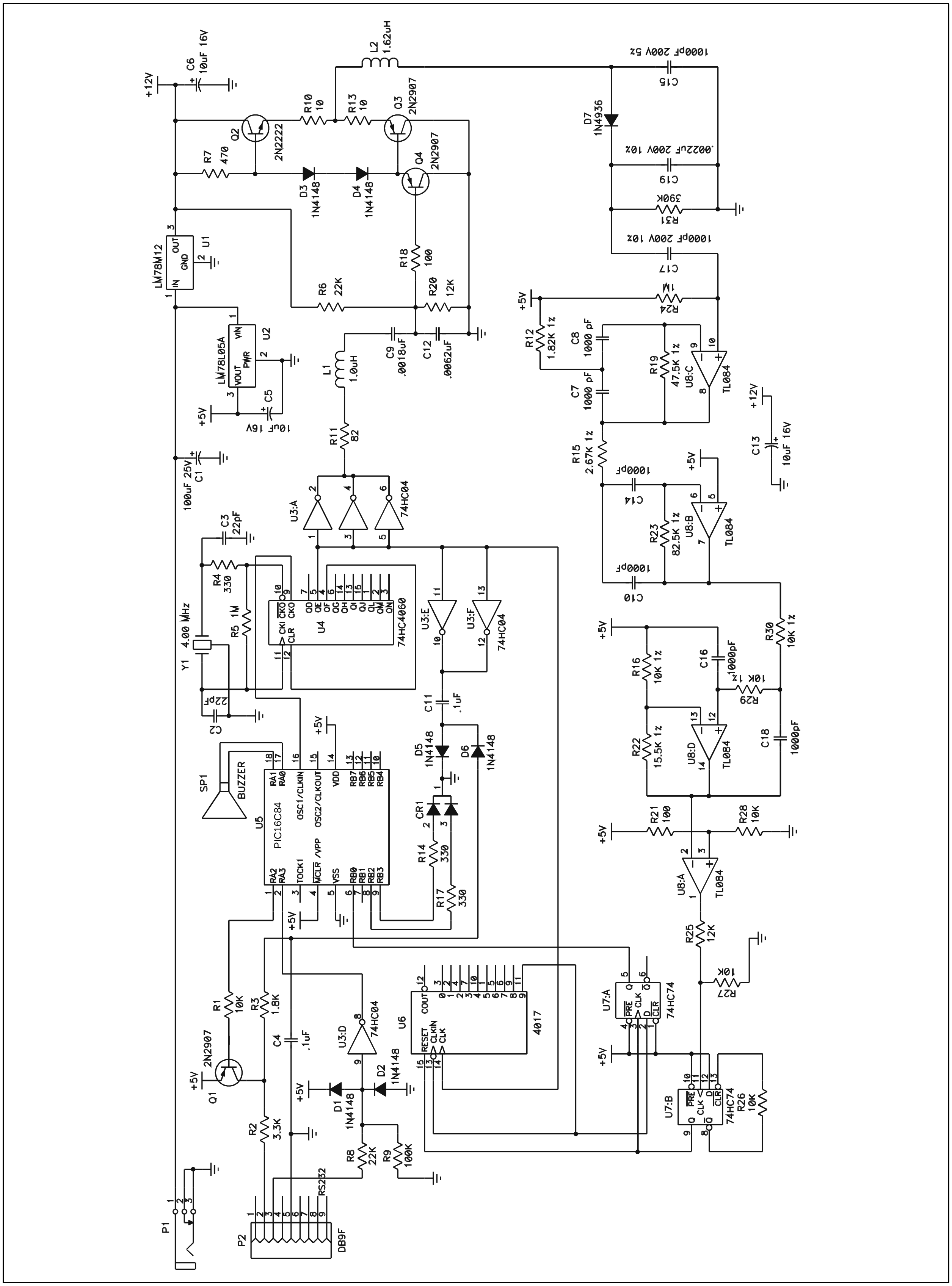
D7 and C19 form a half-wave capacitor-filtered rectifier circuit. The detected envelope signal is charged into the C19. R21 provides a discharge path for the voltage charged in the C19. This voltage passes active filters (U8) and the pulse shaping circuitry (U8) before it is fed into the PIC16C84 for data processing.

The PIC16C84 microcontroller performs data decoding and communicates with the host computer via an RS-232 serial interface.

FIGURE 3-1: SIGNAL WAVEFORM FOR FSK PROTOCOL (Fc = 125 KHZ)



4.0 FSK READER SCHEMATIC



5.0 FSK READER BILL OF MATERIALS

Item #	Qty	Part #	Reference Designator	Part Description	Manufacturer	Vendor	Vendor Part #
1	1	110-93-318-41-001	xU5	SOCKET, 18P OPEN FRAME COLLET (0.300)	MILL-MAX	DIGIKEY	ED3318-ND
2	1	DE9S-FRS	P2	CONN, D-SUB 9P RECPT RT ANGLE	SPC TECHNOLOGY		
3	1	DJ005B	P1	JACK, POWER, 2.5 mm DC PC MOUNT	LZR ELECTRONICS		
4	1	PKM22EPP-4001	SP1	BUZZER, PIEZO, 4kHz, 3-20V	MURATA		
5	2	D220J20COGHAAAC	C2, C3	CAP, 22 pF CER DISK RAD COG 100V	PHILIPS	DIGIKEY	1330PH-ND
6	6	ECQ-P6102JU	C7, C8, C10, C14, C16, C18	CAP, 0.001 uF POLYPROPYLENE 630V	PHILIPS	DIGIKEY	P3497-ND
7	2	2222 370 52102	C15, C17	CAP, 0.001 uF METAL FILM, 5%, RAD, 400V	PHILIPS	DIGIKEY	3001PH-ND
8	1	ECU-S2A182JCB	C9	CAP, 1800 pF MONOLITH CERM, 5%, RAD, 100V	PHILIPS	DIGIKEY	P4864-ND
9	1	2222 370 52222	C19	CAP, 0.0022 UF 400V 5% MF BOX	PHILIPS	DIGIKEY	3003PH-ND
10	1	ECU-S1H682JCB	C12	CAP, 6800 pF 50V CERAMIC MONO 5%	PANASONIC	DIGIKEY	P4946-ND
11	2	ECQ-E1104KF	C4, C11	CAP, 0.1UF 100VDC 10% RAD METAL POLY CAP	PANASONIC	DIGIKEY	EF1104-ND
12	3	ECS-F1CE106K	C5, C6, C13	CAP, TANT, 10uF, 16V	PANASONIC	DIGIKEY	P2038-ND
13	1	ECS-F1AE107	C1	CAP, 100 UFD @ 10VDC 20% TANTALUM CAP	PANASONIC	DIGIKEY	P2032-ND
14	6	1N4148	D1-D6	DIODE, GENERAL PURPOSE, 1N4148 (DO-35)	DIODES INC.	DIGIKEY	1N4148DITR-ND
15	1	1N4936	D7	DIODE, 1A 400V FAST-RECOVERY RECTIFIER	DIODES INC	DIGIKEY	1N4936CT-ND
16	1	-SPARE-	LED1	-SPARE- LOCATION DO NOT INSTALL			
17	1	78F102J	L1	INDUCTOR, 1000 µH, COATED	JW MILLER	DIGIKEY	M7849-ND
18	1	MCT0003-001	L2	INDUCTOR, 1.62 mH	CORNELL DUBI-LIER		
19	3	2N2907A	Q1, Q3, Q4	TRANSISTOR, PNP, 2N2907A, TO-92	MOTOROLA		
20	1	2N2222A	Q2	TRANSISTOR, NPN, 2N2222A, TO-92	MOTOROLA	ALLIED	2N2222A
21	2	5043CX10R0J	R10, R13	RES, CF 10 OHM 1/4W 5%	PHILLIPS		
22	1	82E CR-1/4W-B 5%	R11	RES, CF 82 OHM 1/4W 5%	YAGEO	DIGIKEY	82QBK-ND
23	2	5043CX100R0J	R18, R21	RES, CF 100 OHM 1/4W 5%	PHILLIPS		

Item #	Qty	Part #	Reference Designator	Part Description	Manufacturer	Vendor	Vendor Part #
24	3	5043CX330R0J	R4, R14, R17	RES, CF 330 OHM 1/4W 5%	PHILLIPS		
25	1	5043CX470R0J	R7	RES, CF 470 OHM 5% 1/4W	PHILLIPS		
26	1	1K8 CR-1/4W-B 5%	R3	RES, CF 1.8K OHM 1/4W 5%	YAGEO	DIGIKEY	1.8KQBK-ND
27	1	1K82 MF-1/4W-B 1%	R12	RES, MF 1.82K OHM 1/4W 1%	YAGEO	DIGIKEY	1.82KXBK-ND
28	1	2K67 MF-1/4W-B 1%	R15	RES, 2.67K OHM 1/4W 1% MF	YAGEO	DIGIKEY	2.67KXBK-ND
29	1	3K3 CR-1/4W-B 5%	R2	RES, CF 3.3K OHM 1/4W 5%	YAGEO	DIGIKEY	3.3KQBK-ND
30	4	10K CR-1/4W-B 5%	R1, R26, R27, R28	RES, CF 10K OHM 1/4W 5%	YAGEO	DIGIKEY	10KQBK-ND
31	3	5043ED10K00F	R16, R29, R30	RES, MF 10K 1/4W 1%	PHILLIPS		
32	2	12K CR-1/4W-B 5%	R20, R25	RES, CF 12K OHM 1/4W 5%	YAGEO	DIGIKEY	12KQBK-ND
33	1	16K5 MF-1/4W-B 1%	R22	RES, MF 16.5K OHM 1/4W 1%	YAGEO	DIGIKEY	16.5KXBK-ND
34	1	22K CR-1/4W-B 5%	R6	RES, CF 22K OHM 1/4W 5%	YAGEO	DIGIKEY	22KQBK-ND
35	1	47K5 MF-1/4W-B 1%	R19	RES, MF 47.5K OHM 1/4W 1%	YAGEO	DIGIKEY	47.5KXBK-ND
36	1	82K5 MF-1/4W-B 1%	R23	RES, 82.5K OHM 1/4W 1% MF	YAGEO	DIGIKEY	82.5KXBK-ND
37	1	5043CX100K0J	R9	RES, CF 100K 5% 1/4W	PHILLIPS		
38	2	1M0 CR-1/4W-B 5%	R5, R24	RES, CF 1.0M OHM 1/4W 5%	YAGEO	DIGIKEY	1.0MQBK-ND
39	1	390K CR-1/4W-B 5%	R31	RES, 390K OHM 1/4W 5% CF	YAGEO	DIGIKEY	390KQBK-ND
40	1	LM78M12Ct	U1	IC, REG 12V 3 TERM POS (TO-220)	NATIONAL	DIGIKEY	LM78M12CT-ND
41	1	LM78L05ACZ	U2	IC, REG, +5V 0.1A TO-92	NATIONAL	DIGIKEY	LM78L05ACZ-ND
42	1	MM74HC04N	U3	IC, HEX INVERTER 14P DIP	FAIRCHILD SEMICONDUCTOR	DIGIKEY	MM74HC04N-ND
43	1	MM74HC4060N	U4	IC, 14 STAGE BINARY COUNTER, 16P DIP	FAIRCHILD SEMICONDUCTOR	DIGIKEY	MM74HC4060N-ND
44	1	PIC16C84/P	U5	IC, PIC16C84 PLASTIC, 14P DIP	MICROCHIP		
45	1	CD4017BCN	U6	IC, DECADE COUNTER	FAIRCHILD	DIGIKEY	CD4017BCN-ND
46	1	MM74HC74AN	U7	IC, DUAL D TYPE FLIP FLOP 14P DIP	FAIRCHILD	DIGIKEY	MM74HC74AN-ND
47	1	TL084CN	U8	IC, QUAD OP AMP, 14P DIP	SGS THOMPSON	MOUSER	511-TL084CN
48	1	EFO-EC4004A4	Y1	RESONATOR, 4.00MHZ CERAMIC W/CAP	PANASONIC	DIGIKEY	PX400-ND

6.0 FSK SOURCE CODE FOR THE PICmicro® MCU

The following source code is for the PIC16C84 microcontroller used in the FSK reader electronics.

```
; #=#=#=#=#=#=#=#=#=#=#=#= PROJECT  Microchip FSK Reader =#=#=#=#=#=#=#=#=#=#=#=#
;
; PIC16C84 running at 4MHz, Ti=1us

; ////////////////////////////////////////////////////////////////////////////
; Revision history
; ////////////////////////////////////////////////////////////////////////////
;
; Ver    Date        Comment
;
; 0.01   29 Dec 97   Copied from MChip\Reader\FSK
; 0.03   28 Jan 98   TRANSMIT TAB (h’09’) REGULARLY
;        20 Aug 98   Modified to correct FSK comments
;
;       Tbit=50Tcy=400Ti
;       Ttag=96Tbit
;       Header=h’802A’
;

    processor pic16c84
    #include “p16c84.inc”
        __config b’11111111101001’
        ; Code Protect on, power-up timer on, WDT off, XT oscillator

#define _CARRY          STATUS,0
#define _ZERO           STATUS,2
#define _TO             STATUS,4
#define _RP0            STATUS,5

#define _BUZZ1          PORTA,0
#define _BUZZ2          PORTA,1
#define _RS232TX        PORTA,2
#define _RS232RX        PORTA,3
#define _T0CKI          PORTA,4
StartPORTA      = b’01100’
StartTRISA      = b’11000’
BeepPort        = PORTA
Beep0           = StartPORTA
Beep1           = StartPORTA | b’00001’
Beep2           = StartPORTA | b’00010’

#define _DATA_IN        PORTB,0
#define _UNUSED1        PORTB,1
#define _LED1           PORTB,2
#define _LED2           PORTB,3
#define _UNUSED2        PORTB,4
#define _UNUSED3        PORTB,5
#define _UNUSED4        PORTB,6
#define _UNUSED5        PORTB,7
StartPORTB      = b’00000000’
StartTRISB      = b’00000001’

StartOPTION     = b’00001111’   ; TMR0 internal, prescaler off

BO3             = h’0C’
DelayReg        = h’0C’
BitCtr          = h’0D’
BeepCtrHi       = h’0D’
TxByte          = h’0E’
BeepCtrLo       = h’0E’
```

```
Buffer0         = h'10' ; --- IMMOBILE --- IMMOBILE --- IMMOBILE --- IMMOBILE
Buffer1         = h'11' ; |
Buffer2         = h'12' ; |
Buffer3         = h'13' ; |
Buffer4         = h'14' ; |
Buffer5         = h'15' ; |
Buffer6         = h'16' ; |
Buffer7         = h'17' ; |
Buffer8         = h'18' ; |
Buffer9         = h'19' ; |
BufferA         = h'1A' ; |
BufferB         = h'1B' ; |
;BufferC         = h'1C' ; |
;BufferD         = h'1D' ; |
;BufferE         = h'1E' ; |
;BufferF         = h'1F' ; |
Old0            = h'20' ; |
Old1            = h'21' ; |
Old2            = h'22' ; |
Old3            = h'23' ; |
Old4            = h'24' ; |
Old5            = h'25' ; |
Old6            = h'26' ; |
Old7            = h'27' ; |
Old8            = h'28' ; |
Old9            = h'29' ; |
OldA            = h'2A' ; |
OldB            = h'2B' ; |
;OldC            = h'2C' ; |
;OldD            = h'2D' ; |
;OldE            = h'2E' ; |
;OldF            = h'2F' ; |

SKIP macro
        BTFSC   PCLATH,7
 endm

        org h'0000'             ; *#*#*#* RESET VECTOR *#*#*#*
        CLRF    PCLATH
        CLRF    INTCON
        CLRF    STATUS
        GOTO    RESET_A

        org h'0004'             ; *#*#*#* INTERRUPT VECTOR *#*#*#*
        CLRF    PCLATH
        CLRF    INTCON
        CLRF    STATUS
        GOTO    RESET_A

; ***** Subroutines, Page 0

Delay07                         ;[0] Delay 7Ti
        NOP                     ; |
Delay06                         ;[0] Delay 6Ti
        NOP                     ; |
Delay05                         ;[0] Delay 5Ti
        NOP                     ; |
Delay04                         ;[0] Delay 4Ti
        RETLW   0               ; |

RS232CR                         ;[1] Transmit CR on RS232
        MOVLW   d'13'           ; |
        GOTO    RS232TxW        ; |
RS232TxDigit                    ;[1] Transmit LSnybble of W on RS232
        ANDLW   h'0F'           ; |
```

```
        MOVWF   TxByte          ; |
        MOVLW   h'0A'           ; |
        SUBWF   TxByte,W        ; |
        BTFSS   _CARRY          ; |
        GOTO    DigitLT10       ; |
DigitGE10                       ; |
        MOVLW   'A'-'0'-h'0A'   ; |
        ADDWF   TxByte,f        ; |
DigitLT10                       ; |
        MOVLW   '0'             ; |
        ADDWF   TxByte,W        ; |
RS232TxW                        ;[1] Transmit W on RS232 at 9615 baud
        MOVWF   TxByte          ; | TxByte=W
RS232Tx                         ;[1] Transmit TxByte - 104us = 9615.4 baud
        BSF     _RS232TX        ; | Stop bit
        MOVLW   d'35'           ; | |
        MOVLW   DelayReg        ; | |
RS232TxD1                       ; | |
        DECFSZ  DelayReg,f      ; | |
        GOTO    RS232TxD1       ; | |
        BCF     _RS232TX        ; | Start bit
        NOP                     ; | |
        MOVLW   d'32'           ; | |
        MOVWF   DelayReg        ; | |
RS232TxD2                       ; | |
        DECFSZ  DelayReg,f      ; | |
        GOTO    RS232TxD2       ; | |
        CLRF    BitCtr          ; | BitCtr=#8
        BSF     BitCtr,3        ; | |
RS232TxL1                       ; | {% -4Ti
        BTFSC   TxByte,0        ; |   Transmit TxByte.0, RR TxByte
        GOTO    RS232TxBit1     ; |   |
        NOP                     ; |   |
RS232TxBit0                     ; |   |
        BCF     _RS232TX        ; |   |
        BCF     _CARRY          ; |   |
        GOTO    RS232TxBitDone  ; |   |
RS232TxBit1                     ; |   |
        BSF     _RS232TX        ; |   |
        BSF     _CARRY          ; |   |
        GOTO    RS232TxBitDone  ; |   |
RS232TxBitDone                  ; |   |
        RRF     TxByte,f        ; |   |% 4Ti
        MOVLW   d'30'           ; |   delay 1 bit
        MOVWF   DelayReg        ; |   |
        GOTO    RS232TxD3       ; |   |
RS232TxD3                       ; |   |
        DECFSZ  DelayReg,f      ; |   |
        GOTO    RS232TxD3       ; |   |
        DECFSZ  BitCtr,f        ; |   DEC BitCtr
        GOTO    RS232TxL1       ; | } until (BitCtr==#0)
        CALL    Delay04         ; | delay
        BSF     _RS232TX        ; | stop bit
        RETLW   0               ; end

; ***** End of subroutines, Page 0

RESET_A
        CLRWDT
                                ; Initialise registers
        CLRF    STATUS          ; | Access register page 0
        CLRF    FSR             ; | FSR=#0
        MOVLW   StartPORTA      ; | Initialise PORT and TRIS registers
        MOVWF   PORTA           ; | |
        MOVLW   StartPORTB      ; | |
        MOVWF   PORTB           ; | |
```

```
        BSF     _RP0            ;^| |
        MOVLW   StartTRISA      ;^| |
        MOVWF   TRISA           ;^| |
        MOVLW   StartTRISB      ;^| |
        MOVWF   TRISB           ;^| |
        MOVLW   StartOPTION     ;^| Initialise OPTION register
        MOVWF   OPTION_REG      ;^| |
        BCF     _RP0            ; | |
        CLRF    Old0            ; | Clear Old buffer
        CLRF    Old1            ; | |
        CLRF    Old2            ; | |
        CLRF    Old3            ; | |
        CLRF    Old4            ; | |
        CLRF    Old5            ; | |
        CLRF    Old6            ; | |
        CLRF    Old7            ; | |
        CLRF    Old8            ; | |
        CLRF    Old9            ; | |
        CLRF    OldA            ; | |
        CLRF    OldB            ; | |

BigLoop1
;303-581-1041
        BSF     _LED1           ; LEDs “reading”
        CALL    Delay07         ; |
        BCF     _LED2           ; |
        MOVLW   h’09’           ; Transmit TAB regularly
        CALL    RS232TxW        ; |
        MOVLW   d’96’           ; set BitCtr
        MOVWF   BitCtr          ; |
GetEdge                         ; Get an edge on _DATA_IN
        BTFSC   _DATA_IN        ; |
        GOTO    PreSync_H       ; |
        NOP                     ; |
PreSync_L                       ; |
        BTFSC   _DATA_IN        ; |
        GOTO    PreSync_H       ; |
        BTFSC   _DATA_IN        ; |
        GOTO    PreSync_H       ; |
DoSync_L                        ; |
        CLRWDT                  ; |
        BTFSS   _DATA_IN        ; |
        GOTO    DoSync_L        ; |
        BTFSS   _DATA_IN        ; |
        GOTO    DoSync_L        ; |
        GOTO    Sync_Done       ; |
                                ; |
PreSync_H                       ; |
        BTFSS   _DATA_IN        ; |
        GOTO    PreSync_L       ; |
        BTFSS   _DATA_IN        ; |
        GOTO    PreSync_L       ; |
DoSync_H                        ; |
        CLRWDT                  ; |
        BTFSC   _DATA_IN        ; |
        GOTO    DoSync_H        ; |
        BTFSC   _DATA_IN        ; |
        GOTO    DoSync_H        ; |
        GOTO    Sync_Done       ; |
Sync_Done                       ; |% 6 to (+4) from edge, say 8 from edge
        ;% -192Ti from sample
        MOVLW   d’62’
        MOVWF   DelayReg
        ;% -190Ti from sample
ReadBit                         ; {% -4-DelayReg*3 Ti from sample
```

```
        GOTO    ReadBitD1       ;   delay
ReadBitD1                       ;   |
        DECFSZ  DelayReg,f      ;   |
        GOTO    ReadBitD1       ;   |
        CLRF    BO3             ;   BO3.1=_DATA_IN
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |% effective sample time
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |
        BCF     _CARRY          ;   _CARRY=BO3.1
        BTFSC   BO3,1           ;   |
        BSF     _CARRY          ;   |
        RLF     Buffer0,f       ;   roll in _CARRY
        RLF     Buffer1,f       ;   |
        RLF     Buffer2,f       ;   |
        RLF     Buffer3,f       ;   |
        RLF     Buffer4,f       ;   |
        RLF     Buffer5,f       ;   |
        RLF     Buffer6,f       ;   |
        RLF     Buffer7,f       ;   |
        RLF     Buffer8,f       ;   |
        RLF     Buffer9,f       ;   |
        RLF     BufferA,f       ;   |
        RLF     BufferB,f       ;   |
                                ;  % 19Ti from sample = -381Ti from sample
        MOVLW   d'124'          ;   set bit delay
        MOVWF   DelayReg        ;   |% -379Ti from sample
        ;% -7-DelayReg*3 Ti from sample
        DECFSZ  BitCtr,f        ;   DEC BitCtr
        GOTO    ReadBit         ; } until (BitCtr==#0)

HeadSearch
        MOVLW   d'96'           ; set BitCtr
        MOVWF   BitCtr          ; |
HeadSearchL1                    ; {
        MOVLW   h'80'           ;   if (header found)
        XORWF   BufferB,W       ;   |
        BTFSS   _ZERO           ;   |
        GOTO    NotHead0        ;   |
        MOVLW   h'2A'           ;   |
        XORWF   BufferA,W       ;   |
        BTFSS   _ZERO           ;   |
        GOTO    NotHead0        ;   {
        GOTO    HeadFound       ;     goto HeadFound
NotHead0                        ;   }
        RLF     Buffer0,f       ;   ROL Buffer
        RLF     Buffer1,f       ;   |
        RLF     Buffer2,f       ;   |
        RLF     Buffer3,f       ;   |
        RLF     Buffer4,f       ;   |
        RLF     Buffer5,f       ;   |
        RLF     Buffer6,f       ;   |
        RLF     Buffer7,f       ;   |
        RLF     Buffer8,f       ;   |
        RLF     Buffer9,f       ;   |
        RLF     BufferA,f       ;   |
        RLF     BufferB,f       ;   |
        BCF     Buffer0,0       ;   |
        BTFSC   _CARRY          ;   |
        BSF     Buffer0,0       ;   |
        DECFSZ  BitCtr,f        ;   DEC BitCtr
        GOTO    HeadSearchL1    ; } until (BitCtr==#0)
        GOTO    BigLoop1        ; goto BigLoop1
```

```
HeadFound

CheckSame
        MOVF    Buffer0,W
        XORWF   Old0,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer1,W
        XORWF   Old1,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer2,W
        XORWF   Old2,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer3,W
        XORWF   Old3,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer4,W
        XORWF   Old4,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer5,W
        XORWF   Old5,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer6,W
        XORWF   Old6,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer7,W
        XORWF   Old7,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer8,W
        XORWF   Old8,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer9,W
        XORWF   Old9,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    BufferA,W
        XORWF   OldA,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    BufferB,W
        XORWF   OldB,W
        BTFSS   _ZERO
        GOTO    NotSame
        GOTO    Same
NotSame
        MOVF    Buffer0,W
        MOVWF   Old0
        MOVF    Buffer1,W
        MOVWF   Old1
        MOVF    Buffer2,W
        MOVWF   Old2
        MOVF    Buffer3,W
        MOVWF   Old3
        MOVF    Buffer4,W
        MOVWF   Old4
        MOVF    Buffer5,W
        MOVWF   Old5
```

```
        MOVF    Buffer6,W
        MOVWF   Old6
        MOVF    Buffer7,W
        MOVWF   Old7
        MOVF    Buffer8,W
        MOVWF   Old8
        MOVF    Buffer9,W
        MOVWF   Old9
        MOVF    BufferA,W
        MOVWF   OldA
        MOVF    BufferB,W
        MOVWF   OldB
        GOTO    BigLoop1
Same

TxTag                           ;- Transmit tag
        BSF     _LED2           ; LEDs “Found tag”
        CALL    Delay07         ; |
        BCF     _LED1           ; |
        MOVLW   d’4’            ; Beep at 3597Hz for 1024 cycles
        MOVWF   BeepCtrHi       ; |
        MOVLW   d’0’            ; |
        MOVWF   BeepCtrLo       ; |
BeepLoopJ1                      ; |
        GOTO    BeepLoopJ2      ; |
BeepLoopJ2                      ; |
        MOVLW   Beep1           ; |
        MOVWF   BeepPort        ; |
        MOVLW   d’34’           ; |
        MOVWF   DelayReg        ; |
BeepD1                          ; |
        CLRWDT                  ; |
        DECFSZ  DelayReg,f      ; |
        GOTO    BeepD1          ; |
        MOVLW   Beep2           ; |
        MOVWF   BeepPort        ; |
        MOVLW   d’32’           ; |
        MOVWF   DelayReg        ; |
        NOP                     ; |
        GOTO    BeepD2          ; |
BeepD2                          ; |
        CLRWDT                  ; |
        DECFSZ  DelayReg,f      ; |
        GOTO    BeepD2          ; |
        DECFSZ  BeepCtrLo,f     ; |
        GOTO    BeepLoopJ1      ; |
        DECFSZ  BeepCtrHi,f     ; |
        GOTO    BeepLoopJ2      ; |
        NOP                     ; |
        MOVLW   Beep0           ; |
        MOVWF   BeepPort        ; |

        CALL    RS232CR         ; Transmit tag info
        MOVLW   ‘F’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘S’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘K’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘ ‘             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘/’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘8’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘-’             ; |
```

```
        CALL    RS232TxW        ; |
        MOVLW   '/'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '1'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '0'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'b'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'i'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '='             ; |
        CALL    RS232TxW        ; |
        MOVLW   '5'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '0'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'c'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'y'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'C'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'o'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'n'             ; |
        CALL    RS232TxW        ; |
        MOVLW   's'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'n'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'g'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '='             ; |
        CALL    RS232TxW        ; |
        MOVLW   '9'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '6'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'b'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'i'             ; |
```

```
        CALL    RS232TxW        ; |
        MOVLW   'tʼ             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'P'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'o'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'l'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'r'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'i'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'y'             ; |
        CALL    RS232TxW        ; |
        MOVLW   ' '             ; |
        CALL    RS232TxW        ; |
        MOVLW   '0'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |

        MOVLW   BufferB         ; Transmit tag ID
        MOVWF   FSR             ; |
TxLoop1                         ; |
        SWAPF   INDF,W          ; |
        CALL    RS232TxDigit    ; |
        MOVF    INDF,W          ; |
        CALL    RS232TxDigit    ; |
        DECF    FSR,f           ; |
        BTFSC   FSR,4           ; |
        GOTO    TxLoop1         ; |
        CALL    RS232CR         ; |

        GOTO    BigLoop1        ; goto BigLoop1

        end
```


PSK Reader Reference Design

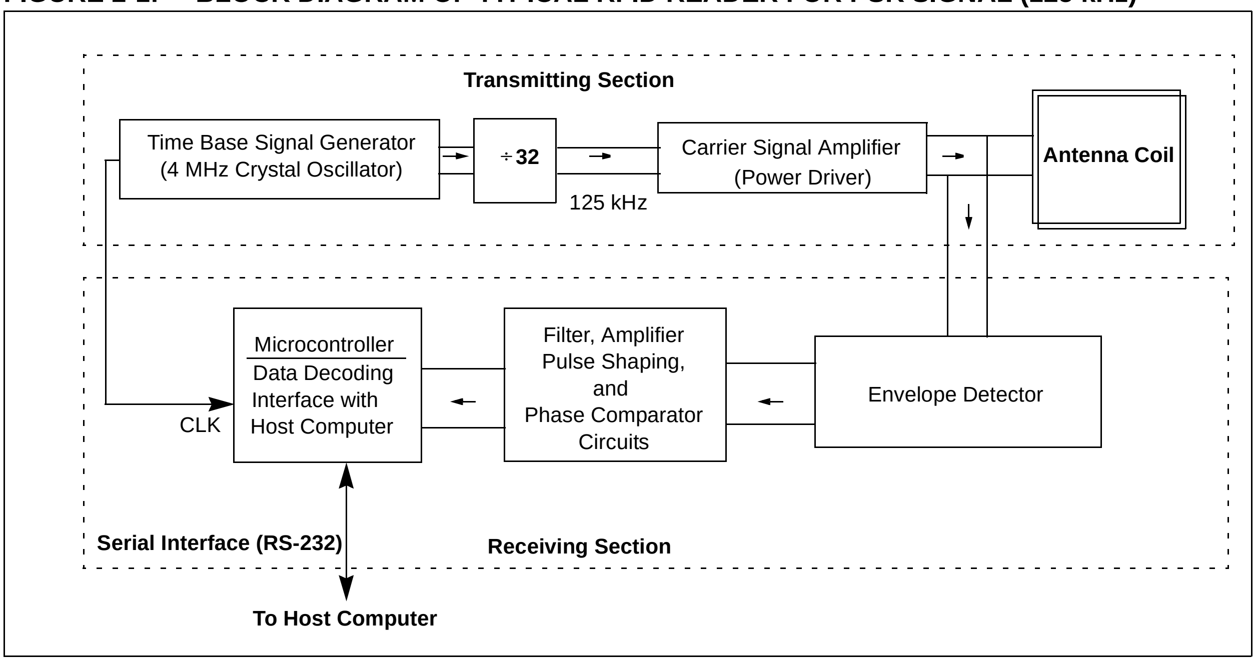
1.0 INTRODUCTION

This application note is written as a reference guide for PSK reader designers. Microchip Technology Inc. provides basic reader schematic for the MCRF200 customers as a part of this design guide. The circuit is designed for a read range of 3 ~ 5 inches with an access control card. The microID PSK Reader (demo unit), which is built based on the PSK reference design, is available in the microID Designers Kit (DV103001). The circuit can be modified for longer read range or other applications with the MCRF200. An electronic copy of the PSK microID PICmicro® source code is available upon request.

2.0 READER CIRCUITS

The RFID reader consists of transmitting and receiving sections. It transmits a carrier signal, receives the backscattering signal, and performs data processing. The reader also communicates with an external host computer. A basic block diagram of the typical RFID reader is shown in Figure 2-1.

FIGURE 2-1: BLOCK DIAGRAM OF TYPICAL RFID READER FOR PSK SIGNAL (125 kHz)



PICmicro is a registered trademark of Microchip Technology Inc.

2.1 Transmitting Section

The transmitting section contains circuitry for a carrier signal (125 kHz), power amplifiers, and a tuned antenna coil.

The 125 kHz carrier signal is typically generated by dividing a 4 MHz (4 MHz/32 = 125 kHz) crystal oscillator signal. The signal is amplified before it is fed into the antenna tuning circuit. A complementary power amplifier circuit is typically used to boost the transmitting signal level.

An antenna impedance tuning circuit consisting of capacitors is used to maximize the signal level at the carrier frequency. This tuning circuit is needed to form an exact LC resonant circuit for the carrier signal. The tuning compensates the variations in the component values and the perturbation of coil inductance due to environment effect. A design guide for the antenna coil is given in *AN710, Antenna Circuit Design for RFID Applications* (DS00710).

2.1.1 LIMITS ON TRANSMITTING SIGNAL LEVEL (FCC PART 15) IN THE USA

Each country limits the signal strength of the RF wave that is intentionally radiated by a device. In the USA, the signal strength of the carrier signal (125 kHz) radiating from the antenna coil must comply with the FCC (Federal Communications Commission) part 15 regulation. The signal level is specified by the 47 CFR Part 15.209a of the federal regulation. For a 125 kHz signal, the FCC limits the signal level to 19.2 μV per meter, or 25.66 dBμV (i.e., 20 log(19.2) = 25.66 dBμV), at 300 meters away from the antenna. For a close distance measurement, an extrapolation rule (40 dB per decade) is applied (Part 15.31.f.2). For example, the signal level at 30 meters away from the device must not exceed:

$$25.66\ dB\mu V + 40\ dB\mu V = 65.66\ dB\mu V$$

2.2 Receiving Section

The receiving section consists of an antenna coil, demodulator, filter, amplifier, pulse shaping, phase comparator, and microcontroller. In applications for proximity read-range, a single coil is often used for both transmitting and receiving. For long read range application, however, separated antennas may be used. More details on the antenna coil are given in *AN710, Antenna Circuit Design for RFID Applications* (DS00710).

In the PSK communication protocol, the phase of the modulation signal changes with the data. Two most common types of phase encoding method are: (a) change phase at any data change ('0' to '1' or '1' to '0'), and (b) change phase at '1'. A typical data rate for PSK applications is one half of the carrier frequency, and it is faster than FSK. However, it requires a wider bandwidth than FSK.

The PSK reader needs two steps for a full recovery of the data. The first step is demodulating the backscattering signal, and the second step is detecting the phase changes in the demodulation signal.

The demodulation is accomplished by detecting the envelope of the carrier signal. A full-wave capacitor-filtered rectifier circuit is used for the demodulation process. A diode detects the peak voltage of the backscattering signal. The voltage is then fed into an RC charging/discharging circuit. The RC time constant must be small enough to allow the voltage across *C* to fall fast enough to keep in step with the envelope. However, the time constant must not be so small as to introduce excessive ripple. The demodulated signal must then pass through a filter, an amplifier, signal shaping, and phase comparator circuits before it is fed to the microcontroller. The microcontroller performs data decoding and communicates with the host computer through an RS-232 or other serial interface protocols.

3.0 microID PSK READER

The MCRF200 can be configured with either PSK_1 or PSK_2 modulation. The PSK_1 changes the phase of the modulation signal on any change of the data (i.e., 0 to 1 or 1 to 0). The PSK_2 changes the phase of the modulation signal on the first clock edge of a data '1'. Figure 3-1 shows the optional PSK encoding protocols. The PSK encoded data is amplitude modulating the carrier signal. A typical PSK modulated signal is shown in Figure 3 in AN680, *Passive RFID Basics* page 6.

This reference reader was designed for use with an MCRF200 with 08Dh in its configuration register, which represents PSK_1, NRZ Direct, Fc/32, data rate, and 128 bits.

The electronic circuitry for the PSK reader is shown in Figure 3-1. The reader needs +9 to +15 VDC power supply. The 125 kHz carrier signal is generated by dividing the 4 MHz time-base signal that is generated by a crystal oscillator. A 16-stage binary ripple counter (74HC4060) is used for this purpose. The 74HC4060 also provides a clock signal for the PIC16C84 microcontroller. Signal from the U8 is also used as a phase reference for receiving signals.

The 125 kHz signal is passed to an RF choke (L1) and filter before it is fed into a power amplifier that is formed by a pair of complementary bipolar transistors (Q2 and Q3).

For long read-range applications, this power amplifier circuit can be modified. Power MOSFETs may be used instead of bipolar transistors (2N2222). These power MOSFETs can be driven by +24 VDC power supply. A push-pull predriver can be added at the front of the complementary circuit. This modification will enhance the signal level of the carrier signal.

The reader circuit uses a single coil for both transmitting and receiving signals. An antenna coil (L2: 1.62 mH) and a resonant capacitor (C21: 1000 pF) forms a series resonant circuit for 125 kHz resonance frequency. Since the C21 is grounded, the carrier signal (125 kHz) is filtered out to the ground after passing the antenna coil. The circuit provides minimum impedance at the resonance frequency. This results in maximizing the antenna current, and therefore, the magnetic field strength is maximized.

In the circuit, D7 and D8 are amplitude demodulators that are detecting the envelope of the backscattering signal. D7 provides a current path during a positive half cycle and the D8 during the negative half cycle. The detected envelope signal is charged into the C27. A discharge path for the voltage charged in the C27 is provided by R33. This voltage passes active filters (U11:C) and the pulse shaping circuitry (U11:A).

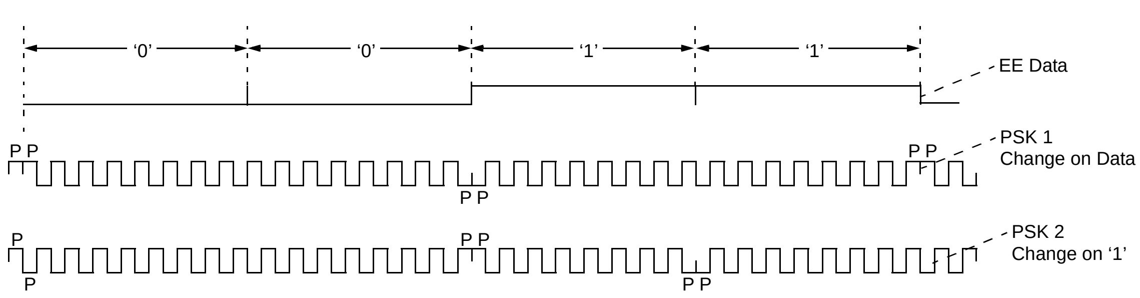
The output from the U11 is a square wave at 62.5 kHz, which exhibits 180 degree phase-shifts in accordance with changes in the data stream from the tag. This signal is used as a clock for D flip-flop (U6:A) for which the D input is a reference 62.5 kHz square wave derived from the 125 kHz transmitting signal. As the phase of the received signal changes, the output of the flip-flop changes, based on whether the clocking occurs during the high or low portions of the reference signal. The recovered data signal is fed to the input I/O pin of the PICmicro MCU (U7) for decoding.

One of the major problems encountered with the PSK reader is that the phase of the returned signal with respect to a reference signal is, for several reasons, indeterminate. If the transitions of the incoming signal and the reference are occurring at the same time, the output of the D flip-flop will be unpredictable. To guarantee that this does not happen, additional circuits have been added.

The received 62.5 kHz signal is buffered by U9:D and a pulse is generated upon every transition of the received signal by U4:C. Likewise, U4:B provides a string of pulses on every transition of the reference 62.5 kHz signal. Note that these pulse strings are at 125 kHz and are independent of the phase state of the received signal.

These pulses are fed to the set and reset lines of U5:A and result in a 125 kHz output at $\overline{Q}$ whose duty cycle is proportional to the phase difference between the two pulse signals. If the duty cycle is near 50%, then the transitions of the 62.5 kHz signals are approximately 90 degrees different which is ideal for PSK demodulation.

FIGURE 3-1: PSK DATA MODULATION



R6 and C10 filter the output of U5:A resulting in a DC level proportional to the phase shift. This level is the input to a window detector consisting of U10 and U4:A. If the DC level is near the midpoint, the output of comparator U10:B would be high and the output of comparator U10:A would be low. Therefore, the output of U4:A would be high. If the DC level is higher than the reference level set by R21, R26, and R30 then the outputs of both comparators would be high, resulting in a low output from U4:A. Similarly, if the DC level is low, both outputs would be low, which would also result in a low output at U4:A.

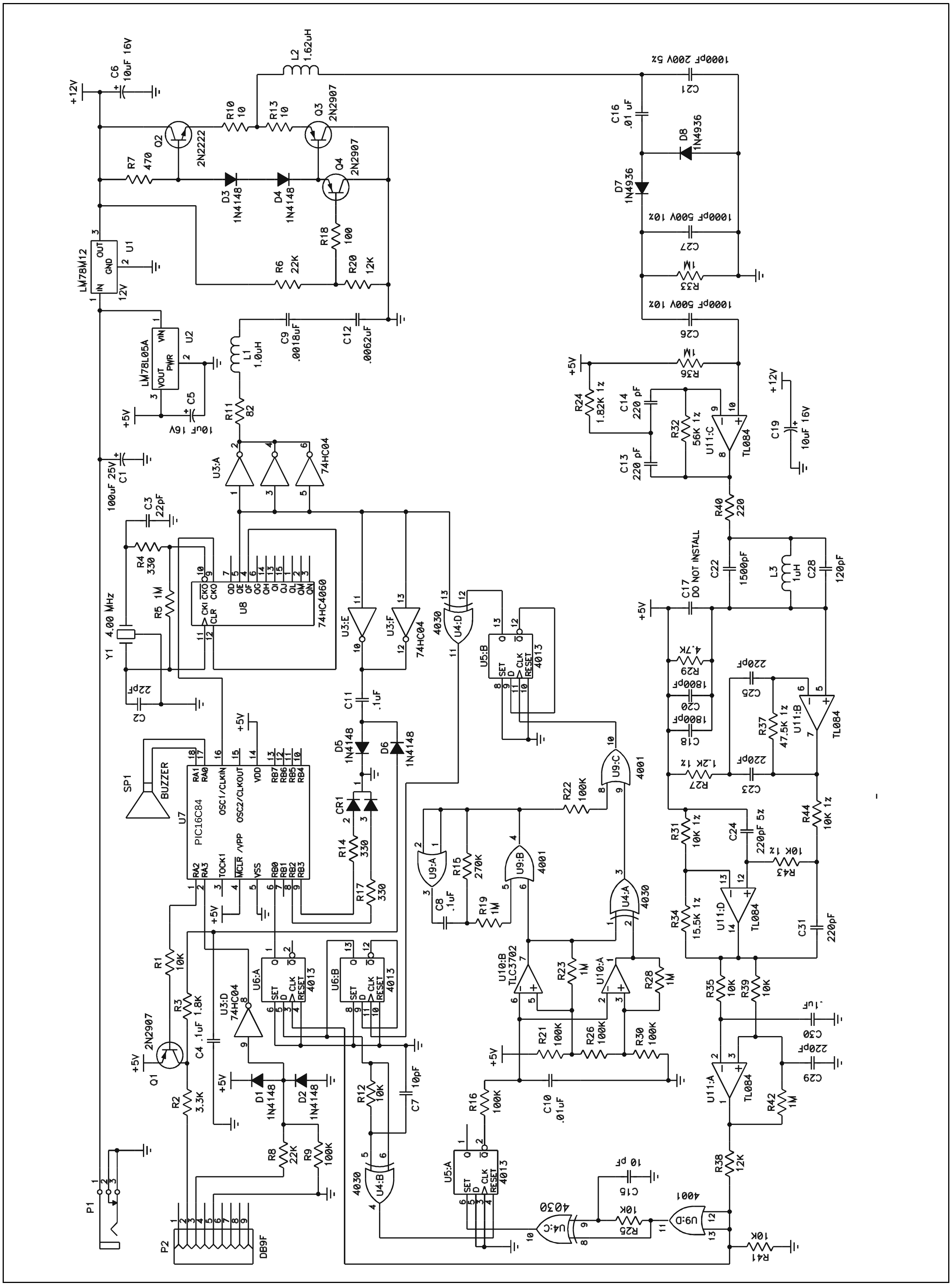
Note that the 125 kHz signal from which the 62.5 kHz reference is obtained passes through gate U4:D. A change of the state on the control output to this gate allows the 125 kHz signal to be 180 degree phase-shifted. This results in a phase-shift in the 62.5 kHz reference of 90 degrees. If the output of the U9:C is low, the flip-flop U5:B will maintain its current state.

If the output of U4:A goes low, which would signify an undesirable phase relationship between the 62.5 kHz signals, then the output of U9:C would have a transition to high, causing U5:B to change state. This would change the reference phase 90 degrees, thus bringing the phases of the 62.5 kHz signals back into a desirable relationship and return the output of U4:A to a high state.

In the event that no tag is present, $\overline{Q}$ of U5:A is always high which makes the output of U10:B low. This turns on an oscillator consisting of U9:A, U9:B, C8, R15, and R19. This oscillator toggles U5:B at about 200 Hz, allowing the reader to be looking for a tag signal with both reference signal phases. When a good tag signal appears, the circuit locks on in a good phase relationship and demodulates the incoming 62.5 kHz signal. As the tag comes closer to the reader, the phase will be shift for a number of reasons. If the shift is sufficient, the reference signal will shift as necessary to maintain good demodulation.

The PIC16C84 microcontroller performs data decoding and communicates with host computer via an RS-232 serial interface.

4.0 PSK READER SCHEMATIC



5.0 PSK READER BILL OF MATERIALS

Item #	Qty	Part #	Reference Designator	Part Description	Manufacturer	Vendor	Vendor Part #
1	1	110-93-314-41-001	xU6	SOCKET, 14P COLLET OPEN FRAME (0.300W)	MILL-MAX	DIGIKEY	ED3314-ND
2	1	DE9S-FRS	P2	CONN, D-SUB 9P RECPT RT ANGLE	SPC TECHNOLOGY		
3	1	DJ005B	P1	JACK, POWER, 2.5mm DC PC MOUNT	LZR ELECTRONICS		
4	1	PKM22EPP-4001	SP1	BUZZER, PIEZO, 4KHz, 3-20V	MURATA		
5	2	D100D20U2MHAAAC	C7, C15	CAP, 10 pF CER DISK RAD, 100V	PHILIPS	DIGIKEY	1301PH-ND
6	2	D220J20COGHAAAC	C2, C3	CAP, 22 pF CER DISK RAD COG 100V	PHILIPS	DIGIKEY	1330PH-ND
7	7	ECU-S1H221JCA	C13, C14, C23-C25, C29, C31	CAP, 220pF, CER MONO, RAD, 50V, 5%	PANASONIC	DIGIKEY	P4929-ND
8	1	ECQ-P6102JU	C21	CAP, 0.001 µF POLYPROPYLENE 630V	PHILIPS	DIGIKEY	P3497-ND
9	2	2222 370 52102	C26, C27	CAP, 0.001 µF METAL FILM, 5%, RAD, 400V	PHILIPS	DIGIKEY	3001PH-ND
10	1	ECU-S2A152JCB	C22	CAP, 1500 pF MONO-LITH CERM, 5%, RAD, 100V	PHILIPS	DIGIKEY	P4863-ND
11	3	ECU-S2A182JCB	C9, C18, C20	CAP, 1800 pF MONO-LITH CERM, 5%, RAD, 100V	PHILIPS	DIGIKEY	P4864-ND
12	1	ECU-S1H682JCB	C12	CAP, 6800 pF 50V CERAMIC MONO 5%	PANASONIC	DIGIKEY	P4946-ND
13	2	ECK-F1H103ZF	C8, C10	CAP, 0.01 µF CERM DISK, +80/-20%, RAD, 50V	PHILIPS	DIGIKEY	P4066A-ND
14	1	ECQ-V1103JM	C16	CAP, 0.01 µF 100V STACK METAL FILM	PANASONIC	DIGIKEY	P4713-ND
15	3	ECQ-E1104KF	C4, C11, C30	CAP, 0.1 µUF 100VDC 10% RAD METAL POLY CAP	PANASONIC	DIGIKEY	EF1104-ND
16	1	ECU-S1H121JCA	C28	CAP, 120 pF, CER MONO, RAD, 50V, 5%	PANASONIC	DIGIKEY	P4926-ND
17	3	ECE-A16Z10	C5, C6, C19	CAP, 10 µF, ELECTRO, RAD, 16V, 20%	PANASONIC	DIGIKEY	P6616-ND
18	1	ECE-A25Z100	C1	CAP, 100 µF, ELEC-TRO, RAD, 25V, 20%	PANASONIC	DIGIKEY	P6616-ND
19	6	1N4148	D1-D6	DIODE, GENERAL PURPOSE, 1N4148 (DO-35)	DIODES INC.	DIGIKEY	1N4148DITR-ND
20	2	1N4936	D7, D8	DIODE, 1A 400V FAST-RECOVERY RECTI-FIER	DIODES INC	DIGIKEY	1N4936CT-ND
21	1	-SPARE-	LED1, C17	-SPARE- LOCATION DO NOT INSTALL			
22	2	78F102J	L1, L3	INDUCTOR, 1000 µH, COATED	JW MILLER	DIGIKEY	M7849-ND
23	1	MCT0003-001	L2	INDUCTOR, 1.62 mH	CORNELL DUBILIER		

Item #	Qty	Part #	Reference Designator	Part Description	Manufacturer	Vendor	Vendor Part #
24	3	2N2907A	Q1, Q3, Q4	TRANSISTOR, PNP, 2N2907A, TO-92	MOTOROLA		
25	1	2N2222A	Q2	TRANSISTOR, NPN, 2N2222A, TO-92	MOTOROLA	ALLIED	2N2222A
26	2	5043CX10R0J	R10, R13	RES, CF 10 OHM 1/4W 5%	PHILLIPS		
27	1	82E CR-1/4W-B 5%	R11	RES, CF 82 OHM 1/4W 5%	YAGEO	DIGIKEY	82QBK-ND
28	1	5043CX100R0J	R18	RES, CF 100 OHM 1/4W 5%	PHILLIPS		
29	1	5043CX220R0J	R40	RES, CF 220 OHM 5% 1/4W	PHILLIPS		
30	3	5043CX330R0J	R4, R14, R17	RES, CF 330 OHM 1/4W 5%	PHILLIPS		
31	1	5043CX470R0J	R7	RES, CF 470 OHM 5% 1/4W	PHILLIPS		
32	1	1K21 MF-1/4W-B 1%	R27	RES, MF 1.21K OHM 1/4W 1%	YAGEO	DIGIKEY	1.21KXBK-ND
33	1	1K8 CR-1/4W-B 5%	R3	RES, CF 1.8K OHM 1/4W 5%	YAGEO	DIGIKEY	1.8KQBK-ND
34	1	1K82 MF-1/4W-B 1%	R24	RES, MF 1.82K OHM 1/4W 1%	YAGEO	DIGIKEY	1.82KXBK-ND
35	1	3K3 CR-1/4W-B 5%	R2	RES, CF 3.3K OHM 1/4W 5%	YAGEO	DIGIKEY	3.3KQBK-ND
36	1	5043CX4K700J	R29	RES, CF 4.7K 5% 1/4W, AXIAL	PHILLIPS		
37	6	10K CR-1/4W-B 5%	R1, R12, R25, R35, R39, R41	RES, CF 10K OHM 1/4W 5%	YAGEO	DIGIKEY	10KQBK-ND
38	3	5043ED10K00F	R31, R43, R44	RES, MF 10K 1/4W 1%	PHILLIPS		
39	2	12K CR-1/4W-B 5%	R20, R38	RES, CF 12K OHM 1/4W 5%	YAGEO	DIGIKEY	12KQBK-ND
40	1	16K5 MF-1/4W-B 1%	R34	RES, MF 16.5K OHM 1/4W 1%	YAGEO	DIGIKEY	16.5KXBK-ND
41	2	22K CR-1/4W-B 5%	R6, R8	RES, CF 22K OHM 1/4W 5%	YAGEO	DIGIKEY	22KQBK-ND
42	1	47K5 MF-1/4W-B 1%	R37	RES, MF 47.5K OHM 1/4W 1%	YAGEO	DIGIKEY	47.5KXBK-ND
43	1	56K CR-1/4W-B 5%	R32	RES, CF 56K OHM 1/4W 5%	YAGEO	DIGIKEY	56KQBK-ND
44	5	5043CX100K0J	R9, R16, R21, R22, R30	RES, CF 100K 5% 1/4W	PHILLIPS		
45	1	180K CR-1/4W-B 5%	R26	RES, CF 180K OHM 1/4W 5%	YAGEO	DIGIKEY	180KQBK-ND
46	1	270K CR-1/4W-B 5%	R15	RES, CF 270K OHM 1/4W 5%	YAGEO	DIGIKEY	270KQBK-ND
47	7	1M0 CR-1/4W-B 5%	R5, R19, R23, R28, R33, R36, R42	RES, CF 1.0M OHM 1/4W 5%	YAGEO	DIGIKEY	1.0MQBK-ND
48	1	LM78M12CT	U1	IC, REG 12V 3 TERM POS (TO-220)	NATIONAL	DIGIKEY	LM78M12CT-ND
49	1	LM78L05ACZ	U2	IC, REG, +5V 0.1 A TO-92	NATIONAL	DIGIKEY	LM78L05ACZ-ND

Item #	Qty	Part #	Reference Designator	Part Description	Manufacturer	Vendor	Vendor Part #
50	1	MM74HC04N	U3	IC, HEX INVERTER 14P DIP	FAIRCHILD SEMICONDUC-TOR	DIGIKEY	MM74HC04N-ND
51	1	CD4030CN	U4	IC, QUAD EXCLUSIVE OR GATE, 14P DIP	FAIRCHILD SEMICONDUC-TOR	DIGIKEY	CD4030CN-ND
52	2	CD4013BCN	U5, U6	IC, DUAL D FLIP FLOP, 14P DIP	FAIRCHILD SEMICONDUC-TOR	DIGIKEY	CD4013BCN-ND
53	1	PIC16C84/P	U7	IC, PIC16C84 PLAS-TIC, 14P DIP	MICROCHIP		
54	1	MM74HC4060N	U8	IC, 14 STAGE BINARY COUNTER, 16P DIP	FAIRCHILD SEMICONDUC-TOR	DIGIKEY	MM74HC4060 N-ND
55	1	CD4001BCN	U9	IC, QUAD 2-IN NOR GATE, 14P DIP	FAIRCHILD SEMICONDUC-TOR	DIGIKEY	CD4001BCN-ND
56	1	TLC3702CP	U10	IC, DUAL VOLTAGE COMPARATORS, 1000mW, 8P DIP	TEXAS INSTRUMENTS	MOUSER	TLC3702CP
57	1	TL084CN	U11	IC, QUAD OP AMP, 1 4P DIP	SGS THOMP-SON	MOUSER	511-TL084CN
58	1	EFO-EC4004A4	Y1	RESONATOR, 4.00MHZ CERAMIC W/CAP	PANASONIC	DIGIKEY	PX400-ND

6.0 PSK SOURCE CODE FOR THE PICmicro® MCU

The following source code is for the PIC16C84 microcontroller used in the PSK reader electronics.

```
; #=#=#=#=#=#=#=#=#=#=#= PROJECT Microchip PSK Reader =#=#=#=#=#=#=#=#=#=#=#
;
; PIC16C84 running at 4MHz, Ti=1us

; ///////////////////////////////////////////////////////////////////////////
; Revision history
; ///////////////////////////////////////////////////////////////////////////
;
; Ver    Date        Comment
;
; 0.01   29 Dec 97   Copied from MChip\Reader\PSK
; 0.03   28 Jan 98   TRANSMIT TAB (h'09') REGULARLY
;        20 Aug 98   Modified to correct PSK comments
;
;       Tbit=32Tcy=256Ti
;       Ttag=128Tbit
;       Header=h'802A'
;
;

    processor pic16c84
    #include "p16c84.inc"
        __config b'11111111101001'
        ; Code Protect on, power-up timer on, WDT off, XT oscillator

#define _CARRY          STATUS,0
#define _ZERO           STATUS,2
#define _TO             STATUS,4
#define _RP0            STATUS,5

#define _BUZZ1          PORTA,0
#define _BUZZ2          PORTA,1
#define _RS232TX        PORTA,2
#define _RS232RX        PORTA,3
#define _T0CKI          PORTA,4
StartPORTA      = b'01100'
StartTRISA      = b'11000'
BeepPort        = PORTA
Beep0           = StartPORTA
Beep1           = StartPORTA | b'00001'
Beep2           = StartPORTA | b'00010'

#define _DATA_IN        PORTB,0
#define _UNUSED1        PORTB,1
#define _LED1           PORTB,2
#define _LED2           PORTB,3
#define _UNUSED2        PORTB,4
#define _UNUSED3        PORTB,5
#define _UNUSED4        PORTB,6
#define _UNUSED5        PORTB,7
StartPORTB      = b'00000000'
StartTRISB      = b'00000001'

StartOPTION     = b'00001111'   ; TMR0 internal, prescaler off

BO3             = h'0C'
DelayReg        = h'0C'
BitCtr          = h'0D'
BeepCtrHi       = h'0D'
TxByte          = h'0E'
BeepCtrLo       = h'0E'
```

```
Buffer0         = h’10’ ; --- IMMOBILE --- IMMOBILE --- IMMOBILE --- IMMOBILE
Buffer1         = h’11’ ; |
Buffer2         = h’12’ ; |
Buffer3         = h’13’ ; |
Buffer4         = h’14’ ; |
Buffer5         = h’15’ ; |
Buffer6         = h’16’ ; |
Buffer7         = h’17’ ; |
Buffer8         = h’18’ ; |
Buffer9         = h’19’ ; |
BufferA         = h’1A’ ; |
BufferB         = h’1B’ ; |
BufferC         = h’1C’ ; |
BufferD         = h’1D’ ; |
BufferE         = h’1E’ ; |
BufferF         = h’1F’ ; |
Old0            = h’20’ ; |
Old1            = h’21’ ; |
Old2            = h’22’ ; |
Old3            = h’23’ ; |
Old4            = h’24’ ; |
Old5            = h’25’ ; |
Old6            = h’26’ ; |
Old7            = h’27’ ; |
Old8            = h’28’ ; |
Old9            = h’29’ ; |
OldA            = h’2A’ ; |
OldB            = h’2B’ ; |
OldC            = h’2C’ ; |
OldD            = h’2D’ ; |
OldE            = h’2E’ ; |
OldF            = h’2F’ ; |

SKIP macro
        BTFSC   PCLATH,7
 endm

        org h’0000’             ; *#*#*#* RESET VECTOR *#*#*#*
        CLRF    PCLATH
        CLRF    INTCON
        CLRF    STATUS
        GOTO    RESET_A

        org h’0004’             ; *#*#*#* INTERRUPT VECTOR *#*#*#*
        CLRF    PCLATH
        CLRF    INTCON
        CLRF    STATUS
        GOTO    RESET_A

; ***** Subroutines, Page 0

Delay07                         ;[0] Delay 7Ti
        NOP                     ; |
Delay06                         ;[0] Delay 6Ti
        NOP                     ; |
Delay05                         ;[0] Delay 5Ti
        NOP                     ; |
Delay04                         ;[0] Delay 4Ti
        RETLW   0               ; |

RS232CR                         ;[1] Transmit CR on RS232
        MOVLW   d’13’           ; |
        GOTO    RS232TxW        ; |
RS232TxDigit                    ;[1] Transmit LSnybble of W on RS232
        ANDLW   h’0F’           ; |
        MOVWF   TxByte          ; |
```



```
        MOVLW   h'0A'           ; |
        SUBWF   TxByte,W        ; |
        BTFSS   _CARRY          ; |
        GOTO    DigitLT10       ; |
DigitGE10                       ; |
        MOVLW   'A'-'0'-h'0A'   ; |
        ADDWF   TxByte,f        ; |
DigitLT10                       ; |
        MOVLW   '0'             ; |
        ADDWF   TxByte,W        ; |
RS232TxW                        ;[1] Transmit W on RS232 at 9615 baud
        MOVWF   TxByte          ; | TxByte=W
RS232Tx                         ;[1] Transmit TxByte - 104us = 9615.4 baud
        BSF     _RS232TX        ; | Stop bit
        MOVLW   d'35'           ; | |
        MOVLW   DelayReg        ; | |
RS232TxD1                       ; | |
        DECFSZ  DelayReg,f      ; | |
        GOTO    RS232TxD1       ; | |
        BCF     _RS232TX        ; | Start bit
        NOP                     ; | |
        MOVLW   d'32'           ; | |
        MOVWF   DelayReg        ; | |
RS232TxD2                       ; | |
        DECFSZ  DelayReg,f      ; | |
        GOTO    RS232TxD2       ; | |
        CLRF    BitCtr          ; | BitCtr=#8
        BSF     BitCtr,3        ; | |
RS232TxL1                       ; | {% -4Ti
        BTFSC   TxByte,0        ; |   Transmit TxByte.0, RR TxByte
        GOTO    RS232TxBit1     ; |   |
        NOP                     ; |   |
RS232TxBit0                     ; |   |
        BCF     _RS232TX        ; |   |
        BCF     _CARRY          ; |   |
        GOTO    RS232TxBitDone  ; |   |
RS232TxBit1                     ; |   |
        BSF     _RS232TX        ; |   |
        BSF     _CARRY          ; |   |
        GOTO    RS232TxBitDone  ; |   |
RS232TxBitDone                  ; |   |
        RRF     TxByte,f        ; |   |% 4Ti
        MOVLW   d'30'           ; |   delay 1 bit
        MOVWF   DelayReg        ; |   |
        GOTO    RS232TxD3       ; |   |
RS232TxD3                       ; |   |
        DECFSZ  DelayReg,f      ; |   |
        GOTO    RS232TxD3       ; |   |
        DECFSZ  BitCtr,f        ; |   DEC BitCtr
        GOTO    RS232TxL1       ; | } until (BitCtr==#0)
        CALL    Delay04         ; | delay
        BSF     _RS232TX        ; | stop bit
        RETLW   0               ; end

; ***** End of subroutines, Page 0

RESET_A
        CLRWDT
                                ; Initialise registers
        CLRF    STATUS          ; | Access register page 0
        CLRF    FSR             ; | FSR=#0
        MOVLW   StartPORTA      ; | Initialise PORT and TRIS registers
        MOVWF   PORTA           ; | |
        MOVLW   StartPORTB      ; | |
        MOVWF   PORTB           ; | |
```

```
        BSF     _RP0            ;^| |
        MOVLW   StartTRISA      ;^| |
        MOVWF   TRISA           ;^| |
        MOVLW   StartTRISB      ;^| |
        MOVWF   TRISB           ;^| |
        MOVLW   StartOPTION     ;^| Initialise OPTION register
        MOVWF   OPTION_REG      ;^| |
        BCF     _RP0            ; | |
        CLRF    Old0            ; | Clear Old buffer
        CLRF    Old1            ; | |
        CLRF    Old2            ; | |
        CLRF    Old3            ; | |
        CLRF    Old4            ; | |
        CLRF    Old5            ; | |
        CLRF    Old6            ; | |
        CLRF    Old7            ; | |
        CLRF    Old8            ; | |
        CLRF    Old9            ; | |
        CLRF    OldA            ; | |
        CLRF    OldB            ; | |
        CLRF    OldC            ; | |
        CLRF    OldD            ; | |
        CLRF    OldE            ; | |
        CLRF    OldF            ; | |

BigLoop1
        BSF     _LED1           ; LEDs “reading”
        CALL    Delay07         ; |
        BCF     _LED2           ; |
        MOVLW   h’09’           ; Transmit TAB regularly
        CALL    RS232TxW        ; |
        MOVLW   d’128’          ; set BitCtr
        MOVWF   BitCtr          ; |
GetEdge                         ; Get an edge on _DATA_IN
        BTFSC   _DATA_IN        ; |
        GOTO    PreSync_H       ; |
        NOP                     ; |
PreSync_L                       ; |
        BTFSC   _DATA_IN        ; |
        GOTO    PreSync_H       ; |
        BTFSC   _DATA_IN        ; |
        GOTO    PreSync_H       ; |
DoSync_L                        ; |
        CLRWDT                  ; |
        BTFSS   _DATA_IN        ; |
        GOTO    DoSync_L        ; |
        BTFSS   _DATA_IN        ; |
        GOTO    DoSync_L        ; |
        GOTO    Sync_Done       ; |
                                ; |
PreSync_H                       ; |
        BTFSS   _DATA_IN        ; |
        GOTO    PreSync_L       ; |
        BTFSS   _DATA_IN        ; |
        GOTO    PreSync_L       ; |
DoSync_H                        ; |
        CLRWDT                  ; |
        BTFSC   _DATA_IN        ; |
        GOTO    DoSync_H        ; |
        BTFSC   _DATA_IN        ; |
        GOTO    DoSync_H        ; |
        GOTO    Sync_Done       ; |
Sync_Done                       ; |% 6 to (+4) from edge, say 8 from edge
        ;% -120Ti from sample
        NOP
        MOVLW   d’38’
```

```
        MOVWF   DelayReg
        ;% -117Ti from sample
ReadBit                         ; {% -3-DelayReg*3 Ti from sample
        NOP                     ;   delay
ReadBitD1                       ;   |
        DECFSZ  DelayReg,f      ;   |
        GOTO    ReadBitD1       ;   |
        CLRF    BO3             ;   BO3.1=_DATA_IN
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |% effective sample time
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |
        BCF     _CARRY          ;   _CARRY=BO3.1
        BTFSC   BO3,1           ;   |
        BSF     _CARRY          ;   |
        RLF     Buffer0,f       ;   roll in _CARRY
        RLF     Buffer1,f       ;   |
        RLF     Buffer2,f       ;   |
        RLF     Buffer3,f       ;   |
        RLF     Buffer4,f       ;   |
        RLF     Buffer5,f       ;   |
        RLF     Buffer6,f       ;   |
        RLF     Buffer7,f       ;   |
        RLF     Buffer8,f       ;   |
        RLF     Buffer9,f       ;   |
        RLF     BufferA,f       ;   |
        RLF     BufferB,f       ;   |
        RLF     BufferC,f       ;   |
        RLF     BufferD,f       ;   |
        RLF     BufferE,f       ;   |
        RLF     BufferF,f       ;   |
                                ;  % 23Ti from sample = -233Ti from sample
        MOVLW   d'75'           ;   set bit delay
        MOVWF   DelayReg        ;   |% -231Ti from sample
        ;% -6-DelayReg*3 Ti from sample
        DECFSZ  BitCtr,f        ;   DEC BitCtr
        GOTO    ReadBit         ; } until (BitCtr==#0)

HeadSearch
        MOVLW   d'128'          ; set BitCtr
        MOVWF   BitCtr          ; |
HeadSearchL1                    ; {
        MOVLW   h'80'           ;   if (header found)
        XORWF   BufferF,W       ;   |
        BTFSS   _ZERO           ;   |
        GOTO    NotHead0        ;   |
        MOVLW   h'2A'           ;   |
        XORWF   BufferE,W       ;   |
        BTFSS   _ZERO           ;   |
        GOTO    NotHead0        ;   {
        GOTO    HeadPolarity0   ;     goto HeadPolarity0
NotHead0                        ;   }
        MOVLW   h'7F'           ;   if (inverse header found)
        XORWF   BufferF,W       ;   |
        BTFSS   _ZERO           ;   |
        GOTO    NotHead1        ;   |
        MOVLW   h'D5'           ;   |
        XORWF   BufferE,W       ;   |
        BTFSS   _ZERO           ;   |
        GOTO    NotHead1        ;   {
        GOTO    HeadPolarity1   ;     goto HeadPolarity1
NotHead1                        ;   }
        RLF     Buffer0,f       ;   ROL Buffer
```

```
        RLF     Buffer1,f       ;   |
        RLF     Buffer2,f       ;   |
        RLF     Buffer3,f       ;   |
        RLF     Buffer4,f       ;   |
        RLF     Buffer5,f       ;   |
        RLF     Buffer6,f       ;   |
        RLF     Buffer7,f       ;   |
        RLF     Buffer8,f       ;   |
        RLF     Buffer9,f       ;   |
        RLF     BufferA,f       ;   |
        RLF     BufferB,f       ;   |
        RLF     BufferC,f       ;   |
        RLF     BufferD,f       ;   |
        RLF     BufferE,f       ;   |
        RLF     BufferF,f       ;   |
        BCF     Buffer0,0       ;   |
        BTFSC   _CARRY          ;   |
        BSF     Buffer0,0       ;   |
        DECFSZ  BitCtr,f        ;   DEC BitCtr
        GOTO    HeadSearchL1    ; } until (BitCtr==#0)
        GOTO    BigLoop1        ; goto BigLoop1

HeadPolarity1
        COMF    Buffer0,f
        COMF    Buffer1,f
        COMF    Buffer2,f
        COMF    Buffer3,f
        COMF    Buffer4,f
        COMF    Buffer5,f
        COMF    Buffer6,f
        COMF    Buffer7,f
        COMF    Buffer8,f
        COMF    Buffer9,f
        COMF    BufferA,f
        COMF    BufferB,f
        COMF    BufferC,f
        COMF    BufferD,f
        COMF    BufferE,f
        COMF    BufferF,f
HeadPolarity0
HeadFound

CheckSame
        MOVF    Buffer0,W
        XORWF   Old0,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer1,W
        XORWF   Old1,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer2,W
        XORWF   Old2,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer3,W
        XORWF   Old3,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer4,W
        XORWF   Old4,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer5,W
        XORWF   Old5,W
        BTFSS   _ZERO
```

```
        GOTO    NotSame
        MOVF    Buffer6,W
        XORWF   Old6,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer7,W
        XORWF   Old7,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer8,W
        XORWF   Old8,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    Buffer9,W
        XORWF   Old9,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    BufferA,W
        XORWF   OldA,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    BufferB,W
        XORWF   OldB,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    BufferC,W
        XORWF   OldC,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    BufferD,W
        XORWF   OldD,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    BufferE,W
        XORWF   OldE,W
        BTFSS   _ZERO
        GOTO    NotSame
        MOVF    BufferF,W
        XORWF   OldF,W
        BTFSS   _ZERO
        GOTO    NotSame
        GOTO    Same
NotSame
        MOVF    Buffer0,W
        MOVWF   Old0
        MOVF    Buffer1,W
        MOVWF   Old1
        MOVF    Buffer2,W
        MOVWF   Old2
        MOVF    Buffer3,W
        MOVWF   Old3
        MOVF    Buffer4,W
        MOVWF   Old4
        MOVF    Buffer5,W
        MOVWF   Old5
        MOVF    Buffer6,W
        MOVWF   Old6
        MOVF    Buffer7,W
        MOVWF   Old7
        MOVF    Buffer8,W
        MOVWF   Old8
        MOVF    Buffer9,W
        MOVWF   Old9
        MOVF    BufferA,W
        MOVWF   OldA
```

```
        MOVF    BufferB,W
        MOVWF   OldB
        MOVF    BufferC,W
        MOVWF   OldC
        MOVF    BufferD,W
        MOVWF   OldD
        MOVF    BufferE,W
        MOVWF   OldE
        MOVF    BufferF,W
        MOVWF   OldF
        GOTO    BigLoop1
Same

TxTag                           ;- Transmit tag
        BSF     _LED2           ; LEDs “Found tag”
        CALL    Delay07         ; |
        BCF     _LED1           ; |
        MOVLW   d’4’            ; Beep at 3597Hz for 1024 cycles
        MOVWF   BeepCtrHi       ; |
        MOVLW   d’0’            ; |
        MOVWF   BeepCtrLo       ; |
BeepLoopJ1                      ; |
        GOTO    BeepLoopJ2      ; |
BeepLoopJ2                      ; |
        MOVLW   Beep1           ; |
        MOVWF   BeepPort        ; |
        MOVLW   d’34’           ; |
        MOVWF   DelayReg        ; |
BeepD1                          ; |
        CLRWDT                  ; |
        DECFSZ  DelayReg,f      ; |
        GOTO    BeepD1          ; |
        MOVLW   Beep2           ; |
        MOVWF   BeepPort        ; |
        MOVLW   d’32’           ; |
        MOVWF   DelayReg        ; |
        NOP                     ; |
        GOTO    BeepD2          ; |
BeepD2                          ; |
        CLRWDT                  ; |
        DECFSZ  DelayReg,f      ; |
        GOTO    BeepD2          ; |
        DECFSZ  BeepCtrLo,f     ; |
        GOTO    BeepLoopJ1      ; |
        DECFSZ  BeepCtrHi,f     ; |
        GOTO    BeepLoopJ2      ; |
        NOP                     ; |
        MOVLW   Beep0           ; |
        MOVWF   BeepPort        ; |

        CALL    RS232CR         ; Transmit tag info
        MOVLW   ‘P’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘S’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘K’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘/’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘2’             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   ‘T’             ; |
        CALL    RS232TxW        ; |
        MOVLW   ‘b’             ; |
        CALL    RS232TxW        ; |
```

```
        MOVLW   'i'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '='             ; |
        CALL    RS232TxW        ; |
        MOVLW   '3'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '2'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'c'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'y'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'C'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'o'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'n'             ; |
        CALL    RS232TxW        ; |
        MOVLW   's'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'n'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'g'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '='             ; |
        CALL    RS232TxW        ; |
        MOVLW   '1'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '2'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '8'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'b'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'i'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'P'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'o'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'l'             ; |
        CALL    RS232TxW        ; |
```

```
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'r'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'i'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'y'             ; |
        CALL    RS232TxW        ; |
        MOVLW   ' '             ; |
        CALL    RS232TxW        ; |
        MOVLW   '0'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |

        MOVLW   BufferF         ; Transmit tag ID
        MOVWF   FSR             ; |
TxLoop1                         ; |
        SWAPF   INDF,W          ; |
        CALL    RS232TxDigit    ; |
        MOVF    INDF,W          ; |
        CALL    RS232TxDigit    ; |
        DECF    FSR,f           ; |
        BTFSC   FSR,4           ; |
        GOTO    TxLoop1         ; |
        CALL    RS232CR         ; |

        GOTO    BigLoop1        ; goto BigLoop1

        end
```

ASK Reader Reference Design

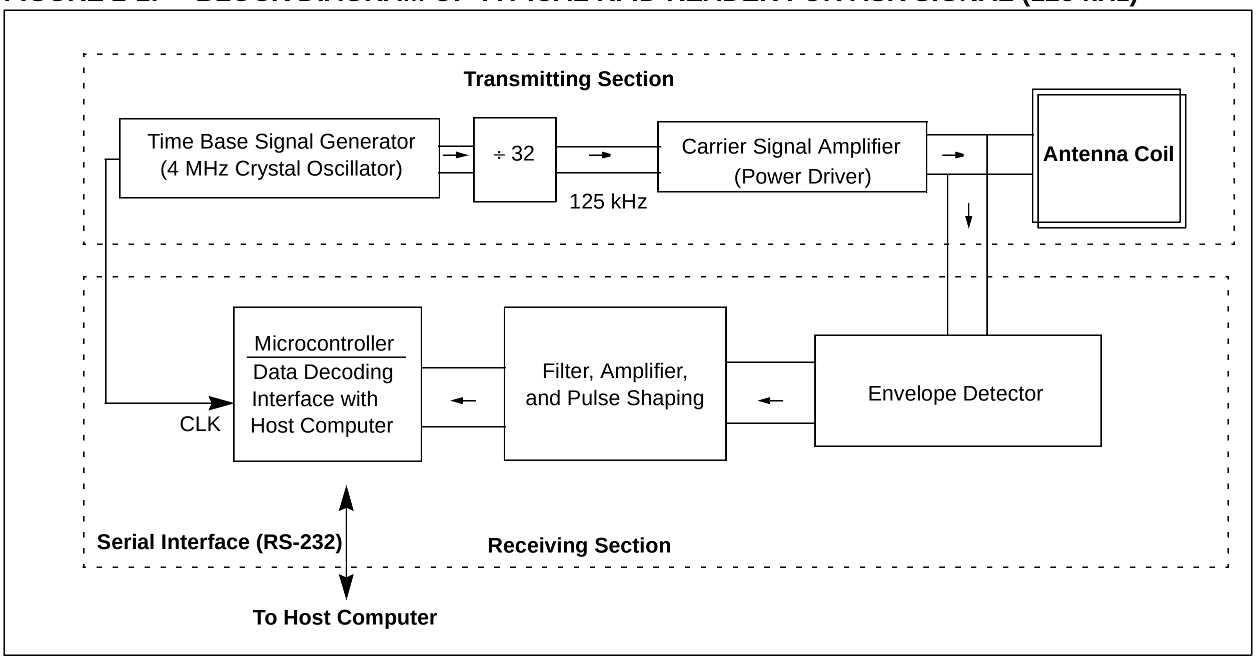
1.0 INTRODUCTION

This application note is written as a reference guide for ASK reader designers. Microchip Technology Inc. provides basic reader electronics circuitry for the MCRF200 customers as a part of this design guide. The circuit is designed for a read range of 3 ~ 5 inches with an access control card. The microID ASK Reader (demo unit), which is built based on the ASK reference design, is available in the microID Designers Kit (DV103001). The circuit can be modified for longer read range or other applications with the MCRF200. An electronic copy of the ASK microID PICmicro® source code is available upon request.

2.0 READER CIRCUITS

The RFID reader consists of transmitting and receiving sections. It transmits a carrier signal, receives the backscattering signal, and performs data processing. The reader also communicates with an external host computer. A basic block diagram of the typical ASK RFID reader is shown in Figure 2-1.

FIGURE 2-1: BLOCK DIAGRAM OF TYPICAL RFID READER FOR ASK SIGNAL (125 kHz)



PICmicro is a registered trademark of Microchip Technology Inc.

2.1 Transmitting Section

The transmitting section contains circuitry for a carrier signal (125 kHz), power amplifiers, and a tuned antenna coil.

The 125 kHz carrier signal is typically generated by dividing a 4 MHz (4 MHz/32 = 125 kHz) crystal oscillator signal. The signal is amplified before it is fed into the antenna tuning circuit. A complementary power amplifier circuit is typically used to boost the ransmitting signal level.

An antenna impedance tuning circuit consisting of capacitors is used to maximize the signal level at the carrier frequency. The tuning compensates the variations in the component values and the perturbation of coil inductance due to environment effect. A design guide for the antenna coil is given in *AN710, Antenna Circuit Design for RFID Applications* (DS00710).

2.1.1 LIMITS ON TRANSMITTING SIGNAL LEVEL (FCC PART 15) IN THE USA

Each country limits the signal strength of the RF wave that is intentionally radiated by a device. In the USA, the signal strength of the carrier signal (125 kHz) radiating from the antenna coil must comply with the FCC (Federal Communications Commission) part 15 regulation. The signal level is specified by the 47 CFR Part 15.209a of the federal regulation. For a 125 kHz signal, the FCC limits the signal level to 19.2 μv per meter, or 25.66 dBμV (i.e., 20 log(19.2) = 25.66 dBμV), at 300 meters away from the antenna. For a close distance measurement, an extrapolation rule (40 dB per decade) is applied (Part 15.31.f.2). For example, the signal level at 30 meters away from the device must not exceed:

25.66 dBμV + 40 dBμV = 65.66 dBμV

2.2 Receiving Section

The receiving section consists of an antenna coil, demodulator, filters, amplifiers, and microcontroller. In applications for close proximity read range, a single coil is often used for both transmitting and receiving. For long read-range applications, however, separated antennas may be used. More details on the antenna coil are given in *AN710, Antenna Circuit Design for RFID Applications* (DS00710).

In the ASK communication protocol, a ‘0’ and a ‘1’ are represented by an amplitude status of receiving signal. Various data coding waveforms that are available by MCRF200 are shown in Figure 1 in *AN680, Passive RFID Basics* (DS00680).

The demodulation of the ASK signal is accomplished by detecting the envelope of the carrier signal. A half-wave capacitor-filtered rectifier circuit is used for the demodulation process. The peak voltage of the back-scattering signal is detected by a diode, and this voltage is then fed into an RC charging/discharging circuit. The RC time constant must be small enough to allow the voltage across C to fall fast enough to keep in step with the envelope. However, the time constant must not be so small as to introduce excessive ripple. The charging capacitor and load R has the following relationship for a full recovery of the data signal.

$$\frac{1}{\omega_o C} > R > \frac{1}{\omega_o C}$$

where ω_s and ω_o are the angular frequencies of the modulation (data) and carrier (125 kHz), respectively. R is the load (discharging) resistor.

The demodulated signal must then pass through a filter and signal shaping circuit before it is fed to the microcontroller. The microcontroller performs data decoding and communicates with the host computer through an RS-232 or other serial interface protocols.

3.0 microID ASK READER

The electronic circuitry for an ASK reader is shown in Section 4.0. The reader needs +9 VDC power supply. The 125 kHz carrier signal is generated by dividing the 4 MHz time base signal that is generated by a crystal oscillator. A 16-stage binary ripple counter (74HC4060) is used for this purpose. The 74HC4060 also provides a clock signal for the PIC16C84 microcontroller. The 125 kHz signal is passed to an RF choke (L1) and filter before it is fed into a power amplifier that is formed by a pair of complementary bipolar transistors (Q2 and Q3).

For long read-range applications, this power amplifier circuit can be modified. Power MOSFETs may be used instead of the bipolar transistors (2N2222). These power MOSFETs can be driven by +24 VDC power supply. A push-pull predriver can be added at the front of the complementary circuit. This modification will enhance the signal level of the carrier signal and the read range of the ASK Reader.

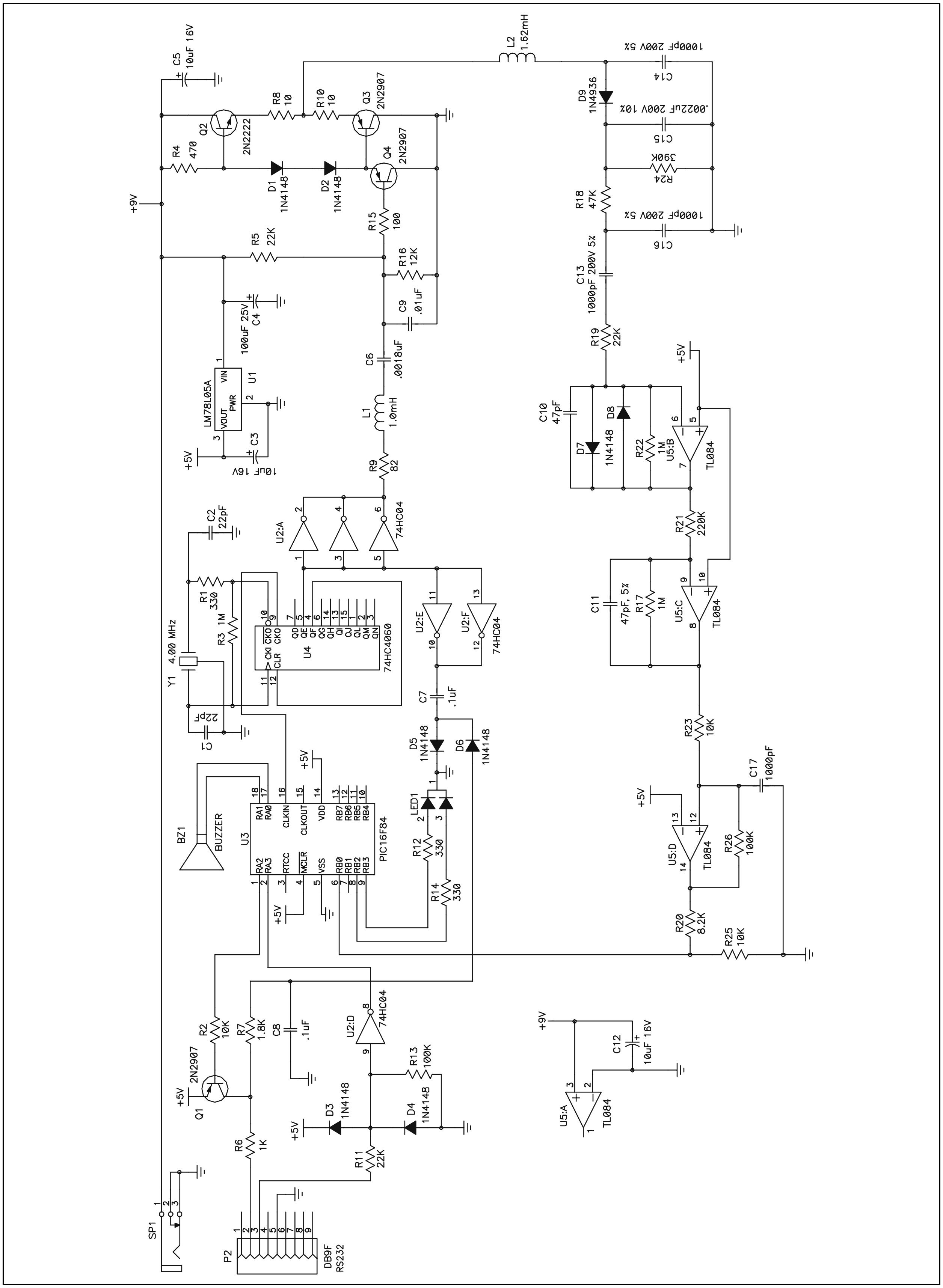
The reader circuit uses a single coil for both transmitting and receiving signals. An antenna coil (L2: 1.62 mH) and a resonant capacitor (C14: 1000 pF) forms a series resonant circuit for a 125 kHz resonance frequency. Since the C14 is grounded, the carrier signal (125 kHz) is filtered out to ground after passing the antenna coil. The circuit provides a minimum impedance at the resonance frequency. This results in maximizing the antenna current, and therefore, the magnetic field strength is maximized.

L2, C14, D7, C15, R24, and the other components in the bottom part of the circuit form a signal receiving section. D9 is a demodulator which detects the envelope of the backscattering signal.

D9 and C15 form a half-wave capacitor-filtered rectifier. The detected envelope signal is charged into C15. R24 provides a discharge path for the voltage charged in C15. This voltage passes active filters (U5:B and C) and the pulse shaping circuitry (U5:A) before it is fed into the PIC16C84 for data processing.

The PIC16C84 microcontroller performs data decoding and communicates with the host computer via an RS-232 serial interface.

4.0 ASK READER SCHEMATIC



5.0 ASK READER BILL OF MATERIALS

Quantity:	Part Number	Part Description	Reference Design
1	02-01518-D	PCB ASSEMBLY DWG, microID ASK READER	
1	03-01518	SCHEMATIC, microID ASK READER	
1	04-01518	PCB FAB, microID ASK READER	
1	08-00161	LABEL, microID ASK READER,U3,CHKS:C1AAh, v1.0, ASK1.HEX	@U3
1	110-93-318-41-001	SOCKET, 18P OPEN FRAME COLLET (0.300)	xU3
1	DE9S-FRS	CONN, D-SUB 9P RECPT RT ANGLE	P2
1	DJ005B	JACK, POWER, 2.5 mm DC	PC MOUNT SP1
1	PKM22EPP-4001	BUZZER, PIEZO, 4 kHz, 3-20V	BZ1
2	D470J25COGHAAAC CAP, 47PF 100V CERAMIC DISC C0G C10,C11 2	D220J20COGHAAAC CAP, 22 pF CER DISK RAD COG 100V	C1, C2
1	ECU-S1H221JCA	CAP, 220pF, CER MONO, RAD, 50V, 5%	C15
1	ECQ-P1102JZ	CAP, 0.001uF POLYPROPYLENE 100V	C17
3	ECQ-P6102JU	CAP, 0.001uF POLYPROPYLENE 630V	C13, C14, C16
1	ECU-S2A182JCB	CAP, 1800pF MONOLITH CERM, 5%, RAD, 100V	C6
1	ECQ-V1103JM	CAP, 0.01uF 100V STACK METAL FILM	C9
2	ECQ-E1104KF	CAP, 0.1UF 100VDC 10% RAD METAL POLY CAP	C7, C8
3	ECE-A16Z10	CAP, 10uF, ELECTRO, RAD, 16V, 20%	C3, C5, C12
1	ECE-A25Z100	CAP, 100uF, ELECTRO, RAD, 25V, 20%	C4
8	1N4148	DIODE, GENERAL PURPOSE, 1N4148 (DO-35)	D1-D8
1	1N4936	DIODE, 1A 400V FAST-RECOVERY RECTIFIER	D9
1	-SPARE- -SPARE- LOCATION DO NOT INSTALL LED1,		
1	78F102J INDUCTOR, 1000uH, COATED		L1
1	MCT0003-001	INDUCTOR, 1.62 µH,	L2
3	2N2907A-TO18	TRANSISTOR, 2N2907A PNP, GEN PURPOUS TO-18	Q1, Q3, Q4
1	2N2222A-TO18	TRANSISTOR, 2N2222A NPN, GEN PURPOUS TO-18	Q2
2	5043CX10R0J	RES, CF 10 OHM 1/4W 5%	R10,R8
1	82E CR-1/4W-B 5%	RES, CF 82 OHM 1/4W 5%	R9
1	5043CX100R0J	RES, CF 100 OHM 1/4W 5%	R15
1	5043CX1K000J	RES, CF 1K 1/4W 5%	R6
3	5043CX330R0J	RES, CF 330 OHM 1/4W 5%	R1, R12, R14
1	5043CX470R0J	RES, CF 470 OHM 5% 1/4W	R4
1	1K8 CR-1/4W-B 5%	RES, CF 1.8K OHM 1/4W 5%	R7
1	390K CR-1/4W-T 5%	RES, CF 390K-OHM,5%,1/4W	R24
1	220K CR-1/4W-T 5%	RES, CF 220K OHM 1/4W 5%	R21
1	8K2 CR-1/4W-T 5%	RES, 8.2K OHM 1/4W 5% CF	R20

Quantity:	Part Number	Part Description	Reference Design
3	10K CR-1/4W-B 5%	RES, CF 10K OHM 1/4W 5%	R2, R23, R25
1	5043CX47K00J	RES, CF 47K 5% 1/4W	R18
1	12K CR-1/4W-B 5%	RES, CF 12K OHM 1/4W 5%	R16
3	22K CR-1/4W-B 5%	RES, CF 22K OHM 1/4W 5%	R5, R11, R19
2	5043CX100K0J	RES, CF 100K 5% 1/4W	R13,R26
3	1M0 CR-1/4W-B 5%	RES, CF 1.0M OHM 1/4W 5%	R3, R17, R22
1	LM78L05ACZ	IC, REG, +5V 0.1A TO-92	U1
1	MM74HC04N	IC, HEX INVERTER 14P DIP	U2
1	PIC16F84-10/P	IC, PIC16F84 PLASTIC, 18P DIP	U3
1	MM74HC4060N	IC, 14 STAGE BINARY COUNTER, 16P DIP	U4
1	TL084CN IC, QUAD OP AMP, 14P DIP		U5
1	EFO-EC4004A4	RESONATOR, 4.00MHZ CERAMIC W/CAP	Y1
2	JS-01	SCREW, JACKSCREW, #4-40x0.416"	P2

6.0 ASK READER SOURCE CODE FOR THE PICmicro® MCU

The following source code is for the PIC16C84 microcontroller used in the ASK reader electronics.

```
; #=#=#=#=#=#=#=#=#=#=#=#= PROJECT  Microchip ASK Reader =#=#=#=#=#=#=#=#=#=#=#=#
; v002.asm
; PIC16C84 running at 4MHz, Ti=1us

; ///////////////////////////////////////////////////////////////////////////
; Revision history
; ///////////////////////////////////////////////////////////////////////////
;
; Ver    Date        Comment
;
; 0.01   01 Jul 98   Copied from MCHIP\READER\FSK
; 0.02   29 Jul 98   MICROCHIP TAG HAS 128 BITS
;
;       Tbit=64Tcy=512Ti
;       Manchester encoded
;       Microchip - Header=h'802A'  Ttag=128Tbit
;       - OR -
;       EM ASK - Header=b'111111111' trailer=b'0'  Ttag=64Tbit
;

    processor pic16c84
    #include "p16c84.inc"
        __config b'11111111101001'
        ; Code Protect on, power-up timer on, WDT off, XT oscillator

#define bit_CARRY       STATUS,0
#define bit_ZERO        STATUS,2
#define bit_RP0         STATUS,5

#define _BUZZ1          PORTA,0
#define _BUZZ2          PORTA,1
#define _RS232TX        PORTA,2
#define _RS232RX        PORTA,3
#define _T0CKI          PORTA,4
StartPORTA      = b'01100'
StartTRISA      = b'11000'
BeepPort        = PORTA
Beep0           = StartPORTA
Beep1           = StartPORTA | b'00001'
Beep2           = StartPORTA | b'00010'

#define _DATA_IN        PORTB,0
#define _UNUSED1        PORTB,1
#define _LED2           PORTB,2
#define _LED1           PORTB,3
#define _UNUSED2        PORTB,4
#define _UNUSED3        PORTB,5
#define _UNUSED4        PORTB,6
#define _UNUSED5        PORTB,7
StartPORTB      = b'00000000'
StartTRISB      = b'00000001'

StartOPTION     = b'10001111'   ; TMR0 internal, prescaler off
                                ; PORTB pullups off

BO3             = h'0C'
DelayReg1       = h'0C'
Mask            = h'0C'
BitCtr          = h'0D'
BeepCtrHi       = h'0D'
TxByte          = h'0E'
BeepCtrLo       = h'0E'
```

```
ParityReg1      = h'0E'
Period          = h'0F'
ParityReg2      = h'0F'

Buffer0         = h'10' ; --- IMMOBILE --- IMMOBILE --- IMMOBILE --- IMMOBILE
Buffer1         = h'11' ; |
Buffer2         = h'12' ; |
Buffer3         = h'13' ; |
Buffer4         = h'14' ; |
Buffer5         = h'15' ; |
Buffer6         = h'16' ; |
Buffer7         = h'17' ; |
Buffer8         = h'18' ; |
Buffer9         = h'19' ; |
BufferA         = h'1A' ; |
BufferB         = h'1B' ; |
BufferC         = h'1C' ; |
BufferD         = h'1D' ; |
BufferE         = h'1E' ; |
BufferF         = h'1F' ; |
Old0            = h'20' ; |
Old1            = h'21' ; |
Old2            = h'22' ; |
Old3            = h'23' ; |
Old4            = h'24' ; |
Old5            = h'25' ; |
Old6            = h'26' ; |
Old7            = h'27' ; |
Old8            = h'28' ; |
Old9            = h'29' ; |
OldA            = h'2A' ; |
OldB            = h'2B' ; |
OldC            = h'2C' ; |
OldD            = h'2D' ; |
OldE            = h'2E' ; |
OldF            = h'2F' ; |

SKIP macro
        BTFSC   PCLATH,7
 endm

        org h'0000'             ; *#*#*#* RESET VECTOR *#*#*#*
        CLRF    PCLATH
        CLRF    INTCON
        CLRF    STATUS
        GOTO    RESET_A

        org h'0004'             ; *#*#*#* INTERRUPT VECTOR *#*#*#*
        CLRF    PCLATH
        CLRF    INTCON
        CLRF    STATUS
        GOTO    RESET_A

; ***** Subroutines, Page 0

Delay07:                        ;[0] Delay 7Ti
        NOP                     ; |
Delay06:                        ;[0] Delay 6Ti
        NOP                     ; |
Delay05:                        ;[0] Delay 5Ti
        NOP                     ; |
Delay04:                        ;[0] Delay 4Ti
        RETLW   0               ; |

RS232CR:                        ;[1] Transmit CR on RS232
        MOVLW   d'13'           ; |
```

```
        GOTO    RS232TxW        ; |
RS232TxDigit:                   ;[1] Transmit LSnybble of W on RS232
        ANDLW   h'0F'           ; |
        MOVWF   TxByte          ; |
        MOVLW   h'0A'           ; |
        SUBWF   TxByte,W        ; |
        BTFSS   bit_CARRY       ; |
        GOTO    DigitLT10       ; |
DigitGE10:                      ; |
        MOVLW   'A'-'0'-h'0A'   ; |
        ADDWF   TxByte,f        ; |
DigitLT10:                      ; |
        MOVLW   '0'             ; |
        ADDWF   TxByte,W        ; |
RS232TxW:                       ;[1] Transmit W on RS232 at 9615 baud
        MOVWF   TxByte          ; | TxByte=W
RS232Tx:                        ;[1] Transmit TxByte - 104us = 9615.4 baud
        BSF     _RS232TX        ; | Stop bit
        MOVLW   d'35'           ; | |
        MOVLW   DelayReg1       ; | |
RS232TxD1:                      ; | |
        DECFSZ  DelayReg1,f     ; | |
        GOTO    RS232TxD1       ; | |
        BCF     _RS232TX        ; | Start bit
        NOP                     ; | |
        MOVLW   d'32'           ; | |
        MOVWF   DelayReg1       ; | |
RS232TxD2:                      ; | |
        DECFSZ  DelayReg1,f     ; | |
        GOTO    RS232TxD2       ; | |
        CLRF    BitCtr          ; | BitCtr=#8
        BSF     BitCtr,3        ; | |
RS232TxL1:                      ; | {% -4Ti
        BTFSC   TxByte,0        ; |   Transmit TxByte.0, RR TxByte
        GOTO    RS232TxBit1     ; |   |
        NOP                     ; |   |
RS232TxBit0:                    ; |   |
        BCF     _RS232TX        ; |   |
        BCF     bit_CARRY       ; |   |
        GOTO    RS232TxBitDone  ; |   |
RS232TxBit1:                    ; |   |
        BSF     _RS232TX        ; |   |
        BSF     bit_CARRY       ; |   |
        GOTO    RS232TxBitDone  ; |   |
RS232TxBitDone:                 ; |   |
        RRF     TxByte,f        ; |   |% 4Ti
        MOVLW   d'30'           ; |   delay 1 bit
        MOVWF   DelayReg1       ; |   |
        GOTO    RS232TxD3       ; |   |
RS232TxD3:                      ; |   |
        DECFSZ  DelayReg1,f     ; |   |
        GOTO    RS232TxD3       ; |   |
        DECFSZ  BitCtr,f        ; |   DEC BitCtr
        GOTO    RS232TxL1       ; | } until (BitCtr==#0)
        CALL    Delay04         ; | delay
        BSF     _RS232TX        ; | stop bit
        RETLW   0               ; end

ParityCheck:                    ;[0] Check parity
        CLRF    ParityReg1      ; | ParityReg1=0
        MOVLW   d'10'           ; | BitCtr=10
        MOVWF   BitCtr          ; | |
ParityL1:                       ; | {
        CLRF    ParityReg2      ; |   ParityReg2=0
        MOVLW   h'10'           ; |   Mask=h'10'
```

```
        MOVWF   Mask            ; |   |
ParityL2:                       ; |   {
        BCF     bit_CARRY       ; |     LSL Buffer0-7
        RLF     Buffer0,f       ; |     |
        RLF     Buffer1,f       ; |     |
        RLF     Buffer2,f       ; |     |
        RLF     Buffer3,f       ; |     |
        RLF     Buffer4,f       ; |     |
        RLF     Buffer5,f       ; |     |
        RLF     Buffer6,f       ; |     |
        RLF     Buffer7,f       ; |     |
        BTFSC   Buffer6,7       ; |     if (Buffer6.7==1)
        INCF    ParityReg2,f    ; |     { INC ParityReg2 }
        MOVF    Mask,W          ; |     W=Mask
        BTFSC   Buffer6,7       ; |     if (Buffer6.7==1)
        XORWF   ParityReg1,f    ; |     { ParityReg1=ParityReg1 XOR W }
        BCF     bit_CARRY       ; |     LSR Mask
        RRF     Mask,f          ; |     |
        BTFSS   bit_CARRY       ; |   } until (bit_CARRY==1)
        GOTO    ParityL2        ; |   |
        BTFSC   ParityReg2,0    ; |   if (ParityReg2.0==1)
        GOTO    ParityBad       ; |   { goto ParityBad }
        DECFSZ  BitCtr,f        ; |   DEC BitCtr
        GOTO    ParityL1        ; | } until (BitCtr==0)
        MOVLW   h'10'           ; | Mask=h'10'
        MOVWF   Mask            ; | |
ParityL3:                       ; | {
        BCF     bit_CARRY       ; |   LSL Buffer0-7
        RLF     Buffer0,f       ; |   |
        RLF     Buffer1,f       ; |   |
        RLF     Buffer2,f       ; |   |
        RLF     Buffer3,f       ; |   |
        RLF     Buffer4,f       ; |   |
        RLF     Buffer5,f       ; |   |
        RLF     Buffer6,f       ; |   |
        RLF     Buffer7,f       ; |   |
        MOVF    Mask,W          ; |   W=Mask
        BTFSC   Buffer6,7       ; |   if (Buffer6.7==1)
        XORWF   ParityReg1,f    ; |   { ParityReg1=ParityReg1 XOR W }
        BCF     bit_CARRY       ; |   LSR Mask
        RRF     Mask,f          ; |   |
        BTFSS   Mask,0          ; | } until (Mask.0==1)
        GOTO    ParityL3        ; | |
        MOVF    ParityReg1,W    ; | if ((ParityReg1 AND h'1E')!=0)
        ANDLW   h'1E'           ; | |
        BTFSS   bit_ZERO        ; | |
        GOTO    ParityBad       ; | { goto ParityBad }
ParityGood:                     ; |
        MOVF    BufferF,W       ; | Buffer0-7=Buffer8-F
        MOVWF   Buffer7         ; | |
        MOVF    BufferE,W       ; | |
        MOVWF   Buffer6         ; | |
        MOVF    BufferD,W       ; | |
        MOVWF   Buffer5         ; | |
        MOVF    BufferC,W       ; | |
        MOVWF   Buffer4         ; | |
        MOVF    BufferB,W       ; | |
        MOVWF   Buffer3         ; | |
        MOVF    BufferA,W       ; | |
        MOVWF   Buffer2         ; | |
        MOVF    Buffer9,W       ; | |
        MOVWF   Buffer1         ; | |
        MOVF    Buffer8,W       ; | |
        MOVWF   Buffer0         ; | |
        BCF     bit_CARRY       ; | bit_CARRY=0
        RETLW   0               ; |
```

```
ParityBad:                      ; |
        MOVF    BufferF,W       ; | Buffer0-7=Buffer8-F
        MOVWF   Buffer7         ; | |
        MOVF    BufferE,W       ; | |
        MOVWF   Buffer6         ; | |
        MOVF    BufferD,W       ; | |
        MOVWF   Buffer5         ; | |
        MOVF    BufferC,W       ; | |
        MOVWF   Buffer4         ; | |
        MOVF    BufferB,W       ; | |
        MOVWF   Buffer3         ; | |
        MOVF    BufferA,W       ; | |
        MOVWF   Buffer2         ; | |
        MOVF    Buffer9,W       ; | |
        MOVWF   Buffer1         ; | |
        MOVF    Buffer8,W       ; | |
        MOVWF   Buffer0         ; | |
        BSF     bit_CARRY       ; | bit_CARRY=1
        RETLW   0               ; |

; ***** End of subroutines, Page 0

RESET_A:
        CLRWDT
                                ; Initialise registers
        CLRF    STATUS          ; | Access register page 0
        CLRF    FSR             ; | FSR=#0
        MOVLW   StartPORTA      ; | Initialise PORT and TRIS registers
        MOVWF   PORTA           ; | |
        MOVLW   StartPORTB      ; | |
        MOVWF   PORTB           ; | |
        BSF     bit_RP0         ;^| |
        MOVLW   StartTRISA      ;^| |
        MOVWF   TRISA           ;^| |
        MOVLW   StartTRISB      ;^| |
        MOVWF   TRISB           ;^| |
        MOVLW   StartOPTION     ;^| Initialise OPTION register
        MOVWF   OPTION_REG      ;^| |
        BCF     bit_RP0         ; | |
        CLRF    Old0            ; | Clear Old buffer
        CLRF    Old1            ; | |
        CLRF    Old2            ; | |
        CLRF    Old3            ; | |
        CLRF    Old4            ; | |
        CLRF    Old5            ; | |
        CLRF    Old6            ; | |
        CLRF    Old7            ; | |

BigLoop1:
        BSF     _LED1           ; LEDs “reading”
        CALL    Delay07         ; |
        BCF     _LED2           ; |
        MOVLW   h’09’           ; Transmit TAB regularly
        CALL    RS232TxW        ; |
        MOVLW   d’128’          ; set BitCtr
        MOVWF   BitCtr          ; |

GetEdge:                        ; Get an edge on _DATA_IN
        BTFSC   _DATA_IN        ; |
        GOTO    PreSync_H0      ; |
        NOP                     ; |
PreSync_L0:                     ; |% 3 from low sample
        NOP                     ; |
        BTFSC   _DATA_IN        ; |
        GOTO    PreSync_H0      ; |
```

```
        CLRF    Period          ; | Period=0
PreSync_L1:                     ; | { % 7+Period*8 from low sample
        INCF    Period,f        ; |   INC Period
        BTFSC   Period,6        ; |   if ((Period*8Ti)>=Tbit*1.25=512Ti*1.25=640Ti)
        BTFSS   Period,4        ; |   |
        SKIP                    ; |   |
        GOTO    BigLoop1        ; |   { goto BigLoop1 }
        BTFSS   _DATA_IN        ; | } until (_DATA_IN==1)
        GOTO    PreSync_L1      ; | |
                                ; |% 6+Period*8 from low sample
                                ; |% 6 from rise
        MOVLW   d'48'           ; | if ((Period*8)>=Tbit*0.75=512Ti*0.75=384Ti)
        SUBWF   Period,W        ; | |
        BTFSC   bit_CARRY       ; | |
        GOTO    Sync_Done       ; | { goto Sync_Done }
                                ; |% 10 from rise
        CALL    Delay05         ; | delay
DoSync_H:                       ; |% 15 from rise
        MOVLW   d'2'            ; | Period=2
        MOVWF   Period          ; | |
        CALL    Delay04         ; | delay
        GOTO    DoSync_HL       ; | |
DoSync_HL:                      ; | {% 7+Period*8 from rise
        INCF    Period,f        ; |   INC Period
        BTFSC   Period,6        ; |   if ((Period*8Ti)>=Tbit*1.25=512Ti*1.25=640Ti)
        BTFSS   Period,4        ; |   |
        SKIP                    ; |   |
        GOTO    BigLoop1        ; |   { goto BigLoop1 }
        BTFSC   _DATA_IN        ; | } until (_DATA_IN==0)
        GOTO    DoSync_HL       ; | |
                                ; |% 6+Period*8 from rise
                                ; |% 6 from fall
        MOVLW   d'16'           ; | if ((Period*8Ti)<Tbit*0.25=512Ti*0.25=128Ti)
        SUBWF   Period,W        ; | |
        BTFSS   bit_CARRY       ; | |
        GOTO    BigLoop1        ; | { goto BigLoop1 }
                                ; |% 10 from fall
        MOVLW   d'48'           ; | if ((Period*8Ti)<Tbit*0.75=512Ti*0.75=384Ti)
        SUBWF   Period,W        ; | |
        BTFSS   bit_CARRY       ; | |
        GOTO    DoSync_L        ; | { goto DoSync_L }
        GOTO    Sync_Done       ; | goto Sync_Done

PreSync_H0:                     ; |% 3 from high sample
        NOP                     ; |
        BTFSS   _DATA_IN        ; |
        GOTO    PreSync_L0      ; |
        CLRF    Period          ; | Period=0
PreSync_H1:                     ; | {% 7+Period*8 from high sample
        INCF    Period,f        ; |   INC Period
        BTFSC   Period,6        ; |   if ((Period*8Ti)>=Tbit*1.25=512Ti*1.25=640Ti)
        BTFSS   Period,4        ; |   |
        SKIP                    ; |   |
        GOTO    BigLoop1        ; |   { goto BigLoop1 }
        BTFSC   _DATA_IN        ; | } until (_DATA_IN==0)
        GOTO    PreSync_H1      ; | |
                                ; |% 6+Period*8 from high sample
                                ; |% 6 from fall
        MOVLW   d'48'           ; | if ((Period*8Ti)>=Tbit*0.75=512Ti*0.75=384Ti)
        SUBWF   Period,W        ; | |
        BTFSC   bit_CARRY       ; | |
        GOTO    Sync_Done       ; | { goto Sync_Done }
                                ; |% 10 from fall
        CALL    Delay05         ; | delay
DoSync_L:                       ; |% 15 from fall
        MOVLW   d'2'            ; | Period=2
```

```
        MOVWF   Period          ; | |
        CALL    Delay04         ; | delay
        GOTO    DoSync_LL       ; | |
DoSync_LL:                      ; | {% 7+Period*8 from fall
        INCF    Period,f        ; |   INC Period
        BTFSC   Period,6        ; |   if ((Period*8Ti)>=Tbit*1.25=512Ti*1.25=640Ti)
        BTFSS   Period,4        ; |   |
        SKIP                    ; |   |
        GOTO    BigLoop1        ; |   { goto BigLoop1 }
        BTFSS   _DATA_IN        ; | } until (_DATA_IN==1)
        GOTO    DoSync_LL       ; | |
                                ; |% 6+Period*8 from fall
                                ; |% 6 from rise
        MOVLW   d’16’           ; | if ((Period*8Ti)<Tbit*0.25=512Ti*0.25=128Ti)
        SUBWF   Period,W        ; | |
        BTFSS   bit_CARRY       ; | |
        GOTO    BigLoop1        ; | { goto BigLoop1 }
                                ; |% 10 from rise
        MOVLW   d’48’           ; | if ((Period*8Ti)<Tbit*0.75=512Ti*0.75=384Ti)
        SUBWF   Period,W        ; | |
        BTFSS   bit_CARRY       ; | |
        GOTO    DoSync_H        ; | { goto DoSync_H }
        GOTO    Sync_Done       ; goto Sync_Done

Sync_Done:                      ; |% 16 from edge
                                ; |% -368 from sample
        MOVLW   d’121’          ; | DelayReg1=121
        MOVWF   DelayReg1       ; | |
        NOP                     ; | delay
ReadBit:                        ; {% -2-DelayReg1*3 Ti from sample
ReadBitD1:                      ;   delay
        DECFSZ  DelayReg1,f     ;   |
        GOTO    ReadBitD1       ;   |
        CLRF    BO3             ;   BO3.1=_DATA_IN
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |% effective sample time
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |
        BCF     bit_CARRY       ;   bit_CARRY=BO3.1
        BTFSC   BO3,1           ;   |
        BSF     bit_CARRY       ;   |
        RLF     Buffer0,f       ;   roll in bit_CARRY
        RLF     Buffer1,f       ;   |
        RLF     Buffer2,f       ;   |
        RLF     Buffer3,f       ;   |
        RLF     Buffer4,f       ;   |
        RLF     Buffer5,f       ;   |
        RLF     Buffer6,f       ;   |
        RLF     Buffer7,f       ;   |
        RLF     Buffer8,f       ;   |
        RLF     Buffer9,f       ;   |
        RLF     BufferA,f       ;   |
        RLF     BufferB,f       ;   |
        RLF     BufferC,f       ;   |
        RLF     BufferD,f       ;   |
        RLF     BufferE,f       ;   |
        RLF     BufferF,f       ;   |% 23 from sample
                                ;   |% -233 from sample
        MOVLW   d’76’           ;   delay 230Ti
        MOVWF   DelayReg1       ;   |
        NOP                     ;   |
ReadBitD2:                      ;   |
        DECFSZ  DelayReg1,f     ;   |
```

```
        GOTO    ReadBitD2       ;   |
                                ;   |% -3 from sample
        CLRF    BO3             ;   BO3.1=_DATA_IN
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |% effective sample time
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |
        BTFSC   _DATA_IN        ;   |
        INCF    BO3,f           ;   |
        BTFSC   Buffer0,0       ;   BO3.1=BO3.1 XOR Buffer0.0
        COMF    BO3,f           ;   |
        BTFSS   BO3,1           ;   if (BO3.1==0)
        GOTO    BigLoop1        ;   { goto BigLoop1 }
                                ;  % 8 from sample
                                ;  % -248 from sample
        MOVLW   d'80'           ;   DelayReg1=80
        MOVWF   DelayReg1       ;   |
        NOP                     ;   delay
                                ;  % -5-DelayReg1*3 Ti from sample
        DECFSZ  BitCtr,f        ;   DEC BitCtr
        GOTO    ReadBit         ; } until (BitCtr==#0)

HeadSearch1:
        MOVLW   d'128'          ; set BitCtr
        MOVWF   BitCtr          ; |
HeadSearch1L1:                  ; {
        MOVF    BufferF,W       ;   if (header found)
        XORLW   h'80'           ;   |
        BTFSS   bit_ZERO        ;   |
        GOTO    NotHead1A       ;   |
        MOVF    BufferE,W       ;   |
        XORLW   h'2A'           ;   |
        BTFSS   bit_ZERO        ;   |
        GOTO    NotHead1A       ;   {
        GOTO    HeadFound0      ;     goto HeadFound0
NotHead1A:                      ;   }
        MOVF    BufferF,W       ;   if (inverse header found)
        XORLW   h'7F'           ;   |
        BTFSS   bit_ZERO        ;   |
        GOTO    NotHead1B       ;   |
        MOVF    BufferE,W       ;   |
        XORLW   h'D5'           ;   |
        BTFSS   bit_ZERO        ;   |
        GOTO    NotHead1B       ;   {
        GOTO    HeadFound1      ;     goto HeadFound1
NotHead1B:                      ;   }
        RLF     Buffer0,f       ;   ROL Buffer
        RLF     Buffer1,f       ;   |
        RLF     Buffer2,f       ;   |
        RLF     Buffer3,f       ;   |
        RLF     Buffer4,f       ;   |
        RLF     Buffer5,f       ;   |
        RLF     Buffer6,f       ;   |
        RLF     Buffer7,f       ;   |
        RLF     Buffer8,f       ;   |
        RLF     Buffer9,f       ;   |
        RLF     BufferA,f       ;   |
        RLF     BufferB,f       ;   |
        RLF     BufferC,f       ;   |
        RLF     BufferD,f       ;   |
        RLF     BufferE,f       ;   |
        RLF     BufferF,f       ;   |
        BCF     Buffer0,0       ;   |
        BTFSC   bit_CARRY       ;   |
        BSF     Buffer0,0       ;   |
        DECFSZ  BitCtr,f        ;   DEC BitCtr
```

```
        GOTO    HeadSearch1L1   ; } until (BitCtr==#0)

        MOVF    Buffer0,W       ; if ((Buffer0-7)!=(Buffer8-F)) { goto BigLoop1 }
        XORWF   Buffer8,W       ; |
        BTFSS   bit_ZERO        ; |
        GOTO    BigLoop1        ; |
        MOVF    Buffer1,W       ; |
        XORWF   Buffer9,W       ; |
        BTFSS   bit_ZERO        ; |
        GOTO    BigLoop1        ; |
        MOVF    Buffer2,W       ; |
        XORWF   BufferA,W       ; |
        BTFSS   bit_ZERO        ; |
        GOTO    BigLoop1        ; |
        MOVF    Buffer3,W       ; |
        XORWF   BufferB,W       ; |
        BTFSS   bit_ZERO        ; |
        GOTO    BigLoop1        ; |
        MOVF    Buffer4,W       ; |
        XORWF   BufferC,W       ; |
        BTFSS   bit_ZERO        ; |
        GOTO    BigLoop1        ; |
        MOVF    Buffer5,W       ; |
        XORWF   BufferD,W       ; |
        BTFSS   bit_ZERO        ; |
        GOTO    BigLoop1        ; |
        MOVF    Buffer6,W       ; |
        XORWF   BufferE,W       ; |
        BTFSS   bit_ZERO        ; |
        GOTO    BigLoop1        ; |
        MOVF    Buffer7,W       ; |
        XORWF   BufferF,W       ; |
        BTFSS   bit_ZERO        ; |
        GOTO    BigLoop1        ; |

HeadSearch2:
        MOVLW   d'64'           ; set BitCtr
        MOVWF   BitCtr          ; |
HeadSearch2L1:                  ; {
        MOVF    BufferF,W       ;   if (header found)
        XORLW   h'FF'           ;   |
        BTFSS   bit_ZERO        ;   |
        GOTO    NotHead2A       ;   |
        BTFSS   BufferE,7       ;   |
        GOTO    NotHead2A       ;   |
        BTFSC   Buffer8,0       ;   |
        GOTO    NotHead2A       ;   {
        GOTO    HeadFound2      ;     goto HeadFound2
NotHead2A:                      ;   }
        RLF     Buffer0,f       ;   ROL Buffer
        RLF     Buffer1,f       ;   |
        RLF     Buffer2,f       ;   |
        RLF     Buffer3,f       ;   |
        RLF     Buffer4,f       ;   |
        RLF     Buffer5,f       ;   |
        RLF     Buffer6,f       ;   |
        RLF     Buffer7,f       ;   |
        RLF     Buffer8,f       ;   |
        RLF     Buffer9,f       ;   |
        RLF     BufferA,f       ;   |
        RLF     BufferB,f       ;   |
        RLF     BufferC,f       ;   |
        RLF     BufferD,f       ;   |
        RLF     BufferE,f       ;   |
        RLF     BufferF,f       ;   |
```

```
        BCF     Buffer0,0       ;   |
        BTFSC   bit_CARRY       ;   |
        BSF     Buffer0,0       ;   |
        DECFSZ  BitCtr,f        ;   DEC BitCtr
        GOTO    HeadSearch2L1   ; } until (BitCtr==#0)

HeadSearch3:
        MOVLW   d'64'           ; set BitCtr
        MOVWF   BitCtr          ; |
HeadSearch3L1:                  ; {
        MOVF    BufferF,W       ;   if (header found)
        XORLW   h'00'           ;   |
        BTFSS   bit_ZERO        ;   |
        GOTO    NotHead3A       ;   |
        BTFSC   BufferE,7       ;   |
        GOTO    NotHead3A       ;   |
        BTFSS   Buffer8,0       ;   |
        GOTO    NotHead3A       ;   {
        GOTO    HeadFound3      ;     goto HeadFound3
NotHead3A:                      ;   }
        RLF     Buffer0,f       ;   ROL Buffer
        RLF     Buffer1,f       ;   |
        RLF     Buffer2,f       ;   |
        RLF     Buffer3,f       ;   |
        RLF     Buffer4,f       ;   |
        RLF     Buffer5,f       ;   |
        RLF     Buffer6,f       ;   |
        RLF     Buffer7,f       ;   |
        RLF     Buffer8,f       ;   |
        RLF     Buffer9,f       ;   |
        RLF     BufferA,f       ;   |
        RLF     BufferB,f       ;   |
        RLF     BufferC,f       ;   |
        RLF     BufferD,f       ;   |
        RLF     BufferE,f       ;   |
        RLF     BufferF,f       ;   |
        BCF     Buffer0,0       ;   |
        BTFSC   bit_CARRY       ;   |
        BSF     Buffer0,0       ;   |
        DECFSZ  BitCtr,f        ;   DEC BitCtr
        GOTO    HeadSearch3L1   ; } until (BitCtr==#0)

        GOTO    BigLoop1        ; goto BigLoop1

HeadFound3:
        COMF    BufferF,f
        COMF    BufferE,f
        COMF    BufferD,f
        COMF    BufferC,f
        COMF    BufferB,f
        COMF    BufferA,f
        COMF    Buffer9,f
        COMF    Buffer8,f
        COMF    Buffer7,f
        COMF    Buffer6,f
        COMF    Buffer5,f
        COMF    Buffer4,f
        COMF    Buffer3,f
        COMF    Buffer2,f
        COMF    Buffer1,f
        COMF    Buffer0,f
        CALL    ParityCheck
        BTFSC   bit_CARRY
        GOTO    BigLoop1
        GOTO    CheckSame
```

```
HeadFound2:
        CALL    ParityCheck
        BTFSC   bit_CARRY
        GOTO    HeadSearch3
        GOTO    CheckSame

HeadFound1:
        COMF    BufferF,f
        COMF    BufferE,f
        COMF    BufferD,f
        COMF    BufferC,f
        COMF    BufferB,f
        COMF    BufferA,f
        COMF    Buffer9,f
        COMF    Buffer8,f
        COMF    Buffer7,f
        COMF    Buffer6,f
        COMF    Buffer5,f
        COMF    Buffer4,f
        COMF    Buffer3,f
        COMF    Buffer2,f
        COMF    Buffer1,f
        COMF    Buffer0,f
HeadFound0:

CheckSame:                      ; if (Buffer!=Old) { goto NotSame }
        MOVF    Buffer0,W       ; |
        XORWF   Old0,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer1,W       ; |
        XORWF   Old1,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer2,W       ; |
        XORWF   Old2,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer3,W       ; |
        XORWF   Old3,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer4,W       ; |
        XORWF   Old4,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer5,W       ; |
        XORWF   Old5,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer6,W       ; |
        XORWF   Old6,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer7,W       ; |
        XORWF   Old7,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer8,W       ; |
        XORWF   Old8,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    Buffer9,W       ; |
        XORWF   Old9,W          ; |
        BTFSS   bit_ZERO        ; |
```

```
        GOTO    NotSame         ; |
        MOVF    BufferA,W       ; |
        XORWF   OldA,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    BufferB,W       ; |
        XORWF   OldB,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    BufferC,W       ; |
        XORWF   OldC,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    BufferD,W       ; |
        XORWF   OldD,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    BufferE,W       ; |
        XORWF   OldE,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |
        MOVF    BufferF,W       ; |
        XORWF   OldF,W          ; |
        BTFSS   bit_ZERO        ; |
        GOTO    NotSame         ; |

Same:

TxTag:                          ;- Transmit tag
        BSF     _LED2           ; LEDs “Found tag”
        CALL    Delay07         ; |
        BCF     _LED1           ; |
        MOVLW   d’4’            ; Beep at 3597Hz for 1024 cycles
        MOVWF   BeepCtrHi       ; |
        MOVLW   d’0’            ; |
        MOVWF   BeepCtrLo       ; |
BeepLoopJ1:                     ; |
        GOTO    BeepLoopJ2      ; |
BeepLoopJ2:                     ; |
        MOVLW   Beep1           ; |
        MOVWF   BeepPort        ; |
        MOVLW   d’34’           ; |
        MOVWF   DelayReg1       ; |
BeepD1:                         ; |
        CLRWDT                  ; |
        DECFSZ  DelayReg1,f     ; |
        GOTO    BeepD1          ; |
        MOVLW   Beep2           ; |
        MOVWF   BeepPort        ; |
        MOVLW   d’32’           ; |
        MOVWF   DelayReg1       ; |
        NOP                     ; |
        GOTO    BeepD2          ; |
BeepD2:                         ; |
        CLRWDT                  ; |
        DECFSZ  DelayReg1,f     ; |
        GOTO    BeepD2          ; |
        DECFSZ  BeepCtrLo,f     ; |
        GOTO    BeepLoopJ1      ; |
        DECFSZ  BeepCtrHi,f     ; |
        GOTO    BeepLoopJ2      ; |
        NOP                     ; |
        MOVLW   Beep0           ; |
        MOVWF   BeepPort        ; |

        MOVF    OldF,W
```

```
MOVWF   BufferF
MOVF    OldE,W
MOVWF   BufferE
MOVF    OldD,W
MOVWF   BufferD
MOVF    OldC,W
MOVWF   BufferC
MOVF    OldB,W
MOVWF   BufferB
MOVF    OldA,W
MOVWF   BufferA
MOVF    Old9,W
MOVWF   Buffer9
MOVF    Old8,W
MOVWF   Buffer8
MOVF    Old7,W
MOVWF   Buffer7
MOVF    Old6,W
MOVWF   Buffer6
MOVF    Old5,W
MOVWF   Buffer5
MOVF    Old4,W
MOVWF   Buffer4
MOVF    Old3,W
MOVWF   Buffer3
MOVF    Old2,W
MOVWF   Buffer2
MOVF    Old1,W
MOVWF   Buffer1
MOVF    Old0,W
MOVWF   Buffer0

CALL    RS232CR         ; Transmit tag info
MOVLW   'A'             ; |
CALL    RS232TxW        ; |
MOVLW   'S'             ; |
CALL    RS232TxW        ; |
MOVLW   'K'             ; |
CALL    RS232TxW        ; |
CALL    RS232CR         ; |
MOVLW   'T'             ; |
CALL    RS232TxW        ; |
MOVLW   'b'             ; |
CALL    RS232TxW        ; |
MOVLW   'i'             ; |
CALL    RS232TxW        ; |
MOVLW   't'             ; |
CALL    RS232TxW        ; |
MOVLW   '='             ; |
CALL    RS232TxW        ; |
MOVLW   '6'             ; |
CALL    RS232TxW        ; |
MOVLW   '4'             ; |
CALL    RS232TxW        ; |
MOVLW   'T'             ; |
CALL    RS232TxW        ; |
MOVLW   'c'             ; |
CALL    RS232TxW        ; |
MOVLW   'y'             ; |
CALL    RS232TxW        ; |
CALL    RS232CR         ; |
MOVLW   'C'             ; |
CALL    RS232TxW        ; |
MOVLW   'o'             ; |
CALL    RS232TxW        ; |
```

```
        MOVLW   'n'             ; |
        CALL    RS232TxW        ; |
        MOVLW   's'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'n'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'g'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '='             ; |
        CALL    RS232TxW        ; |
        MOVF    BufferF,W       ; |
        XORLW   h'80'           ; |
        BTFSS   bit_ZERO        ; |
        GOTO    Ttag64          ; |
Ttag128:                        ; |
        MOVLW   '1'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '2'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '8'             ; |
        CALL    RS232TxW        ; |
        GOTO    TtagJ1          ; |
Ttag64:                         ; |
        MOVLW   '6'             ; |
        CALL    RS232TxW        ; |
        MOVLW   '4'             ; |
        CALL    RS232TxW        ; |
        GOTO    TtagJ1          ; |
TtagJ1:                         ; |
        MOVLW   'T'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'b'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'i'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   'P'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'o'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'l'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'a'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'r'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'i'             ; |
        CALL    RS232TxW        ; |
        MOVLW   't'             ; |
        CALL    RS232TxW        ; |
        MOVLW   'y'             ; |
```

```
        CALL    RS232TxW        ; |
        MOVLW   ' '             ; |
        CALL    RS232TxW        ; |
        MOVLW   '0'             ; |
        CALL    RS232TxW        ; |
        CALL    RS232CR         ; |
        MOVLW   BufferF         ; Transmit tag ID
        MOVWF   FSR             ; |
        MOVF    BufferF,W       ; |
        XORLW   h'80'           ; |
        BTFSC   bit_ZERO        ; |
        GOTO    TxLoop1         ; |
        MOVLW   Buffer7         ; |
        MOVWF   FSR             ; |
TxLoop1:                        ; |
        SWAPF   INDF,W          ; |
        CALL    RS232TxDigit    ; |
        MOVF    INDF,W          ; |
        CALL    RS232TxDigit    ; |
        DECF    FSR,f           ; |
        BTFSC   FSR,4           ; |
        GOTO    TxLoop1         ; |
        CALL    RS232CR         ; |

        GOTO    BigLoop1        ; goto BigLoop1

NotSame:                        ; Old=Data
        MOVF    Buffer0,W       ; |
        MOVWF   Old0            ; |
        MOVF    Buffer1,W       ; |
        MOVWF   Old1            ; |
        MOVF    Buffer2,W       ; |
        MOVWF   Old2            ; |
        MOVF    Buffer3,W       ; |
        MOVWF   Old3            ; |
        MOVF    Buffer4,W       ; |
        MOVWF   Old4            ; |
        MOVF    Buffer5,W       ; |
        MOVWF   Old5            ; |
        MOVF    Buffer6,W       ; |
        MOVWF   Old6            ; |
        MOVF    Buffer7,W       ; |
        MOVWF   Old7            ; |
        MOVF    Buffer8,W       ; |
        MOVWF   Old8            ; |
        MOVF    Buffer9,W       ; |
        MOVWF   Old9            ; |
        MOVF    BufferA,W       ; |
        MOVWF   OldA            ; |
        MOVF    BufferB,W       ; |
        MOVWF   OldB            ; |
        MOVF    BufferC,W       ; |
        MOVWF   OldC            ; |
        MOVF    BufferD,W       ; |
        MOVWF   OldD            ; |
        MOVF    BufferE,W       ; |
        MOVWF   OldE            ; |
        MOVF    BufferF,W       ; |
        MOVWF   OldF            ; |
        GOTO    BigLoop1        ; goto BigLoop1

        end
```

NOTES:

FSK Anti-Collision Reader Reference Design

1.0 INTRODUCTION

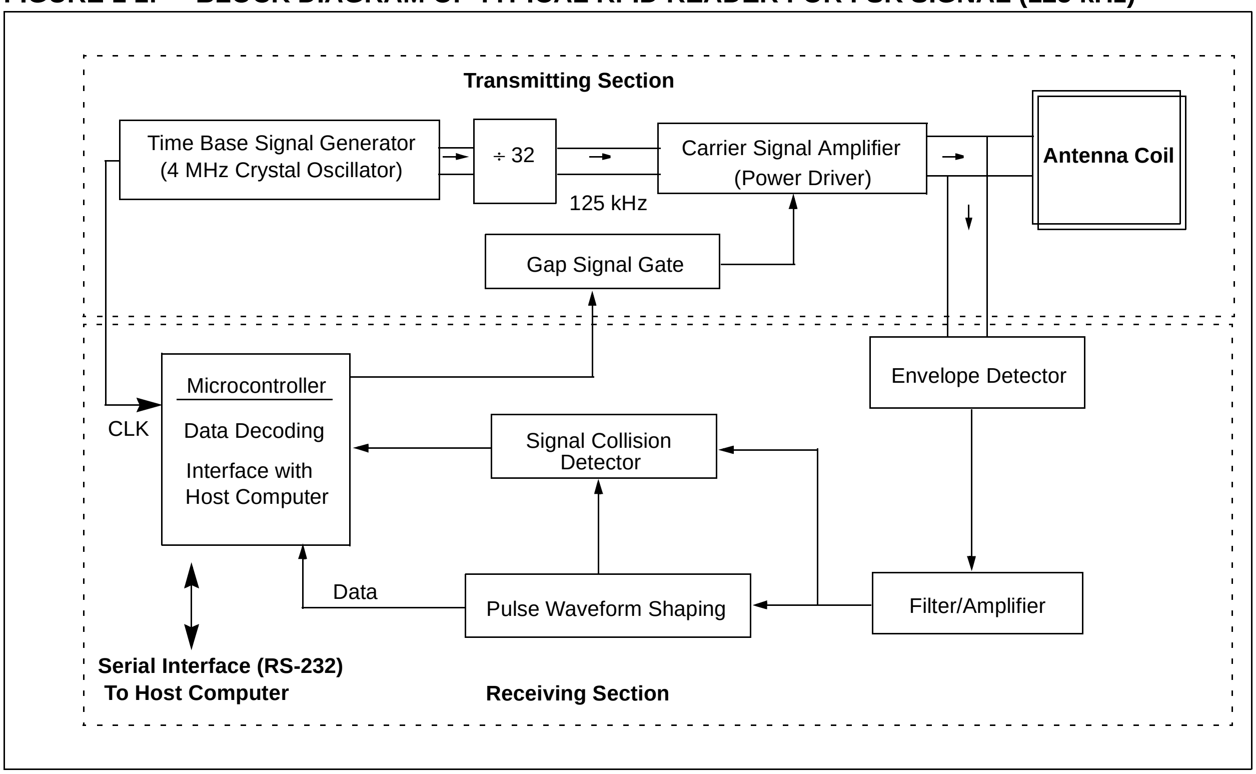
When more than one tag is in the same RF field of a reader, each tag will transmit data at the same time. This results in data collision at the receiving end of the reader. No correct decision can be made based on this data. The reader must receive data from a tag at a time for correct data processing.

The anti-collision device (MCRF250) is designed to send FSK data to reader without data collision, and it must be read by an anti-collision reader. This type of device can be effectively used in inventory and asset control applications where multiple tags are read in the same RF field. The anti-collision algorithm of the device is explained in the *MCRF250 Data Sheet* (DS21267).

This application note is written as a reference guide for anti-collision reader designers. The anti-collision reader is designed to provide correct signals to the anti-collision device (MCRF250) to perform an anti-collision action during operation.

Microchip Technology Inc. provides basic anti-collision FSK reader electronic circuitry for the MCRF250 customers as a part of this design guide. The microID Anti-collision Reader (demo unit), that can read 10 tags or more in the same RF field, is available in the microID Developers Kit (DV103002). An electronic copy of the microID PICmicro® source code is also available upon request.

FIGURE 1-1: BLOCK DIAGRAM OF TYPICAL RFID READER FOR FSK SIGNAL (125 kHz)



PICmicro is a registered tradmark of Microchip Technology Inc.

2.0 READER CIRCUITS

The anti-collision RFID reader consists of a transmitting and a receiving section. The transmitting section includes a carrier frequency generator, gap signal gate, and an antenna circuit. The receiving section includes peak detector, signal amplifier/filter, signal collision detector, and the microcontroller for data processing.

The reader also communicates with an external host computer. A basic block diagram of the typical RFID reader is shown in Figure 1-1.

The electronic circuitry for an anti-collision FSK reader is shown in Section 3.0. The reader needs a +9 VDC power supply.

The 125 kHz carrier signal is generated by dividing the 4 MHz time base signal that is generated by a crystal oscillator. A 16-stage binary ripple counter (74HC4060) is used for this purpose. The 74HC4060 also provides a clock signal for the PIC16C84 microcontroller. The 125 kHz signal from Pin no. 5 of U6 is fed into U2 (Nor gate) and two stage power amplifiers that are formed by U4, Q1, and Q2.

The 125 kHz signal from Q1 and Q2 is fed into the antenna circuit formed by L1(1.62 mH) and C22 (1000 pF). L1 and C22 form a series resonant circuit for a 125 kHz resonance frequency. Since the C22 is grounded, the carrier signal (125 kHz) is filtered out to ground after passing the antenna coil. The circuit provides a minimum impedance at the resonance frequency. This results in maximizing the antenna current, and therefore, the magnetic field strength is maximized.

The gap signal from Pin no. 7 of U7 (Microcontroller) controls the 125 KHz antenna driver circuit (Q1 and Q2). Q1 and Q2 are turned off during the gap signal “high”. There is no RF signal at the antenna coil during this gap period.

The reader circuit uses a single coil for both transmitting and receiving signals. L1, C22, D8, and the other components in the bottom parts of the circuit form a signal receiving section.

In the FSK communication protocol, a ‘0’ and a ‘1’ are represented by two different frequencies. In the MCRF250, a ‘0’ and a ‘1’ are represented by Fc/8 and Fc/10, respectively. Fc is the carrier frequency. The MCRF250 sends this FSK signal to the reader by an amplitude modulation of the carrier signal.

The demodulation is accomplished by detecting the envelope of the carrier signal. A half-wave capacitor-filtered rectifier circuit (D8, D9, and C26) is used for the demodulation process. The detected envelope signal is charged into the C26. R37 provides a discharge path for the voltage charged in the C26. This voltage passes active filters (U10:A,C,D) and the pulse shaping circuitry (U10:B) before it is fed into the PIC16C84 for data processing. U10 (A,D,C) forms a bandpass filter for 12 kHz – 16 kHz signals.

When more than one tag are transmitting data at same time, there will be wobbles in data signals in the receiver. This wobble is detected in U8. If the wobble occurs, c10 becomes fully charged. This will set CLK input of US:B, and results in a logic “LOW” in $\overline{Q}$ of the U5:B. The microcontoller (U7) detects the logic “LOW” and turns on the gap control gate (U5:A) to send a gap signla to the tags.

The PIC16C84 microcontroller performs data decoding, provides gap timing signals, and communicates with the host computer via an RS-232 serial interface.

FIGURE 2-1: RFID FSK ANTI-COLLISION WINDOW

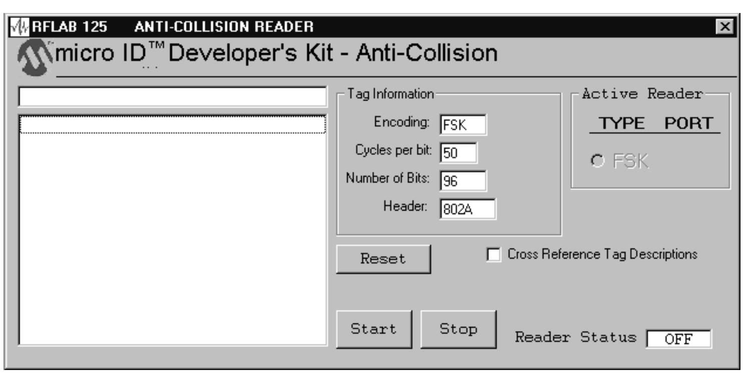
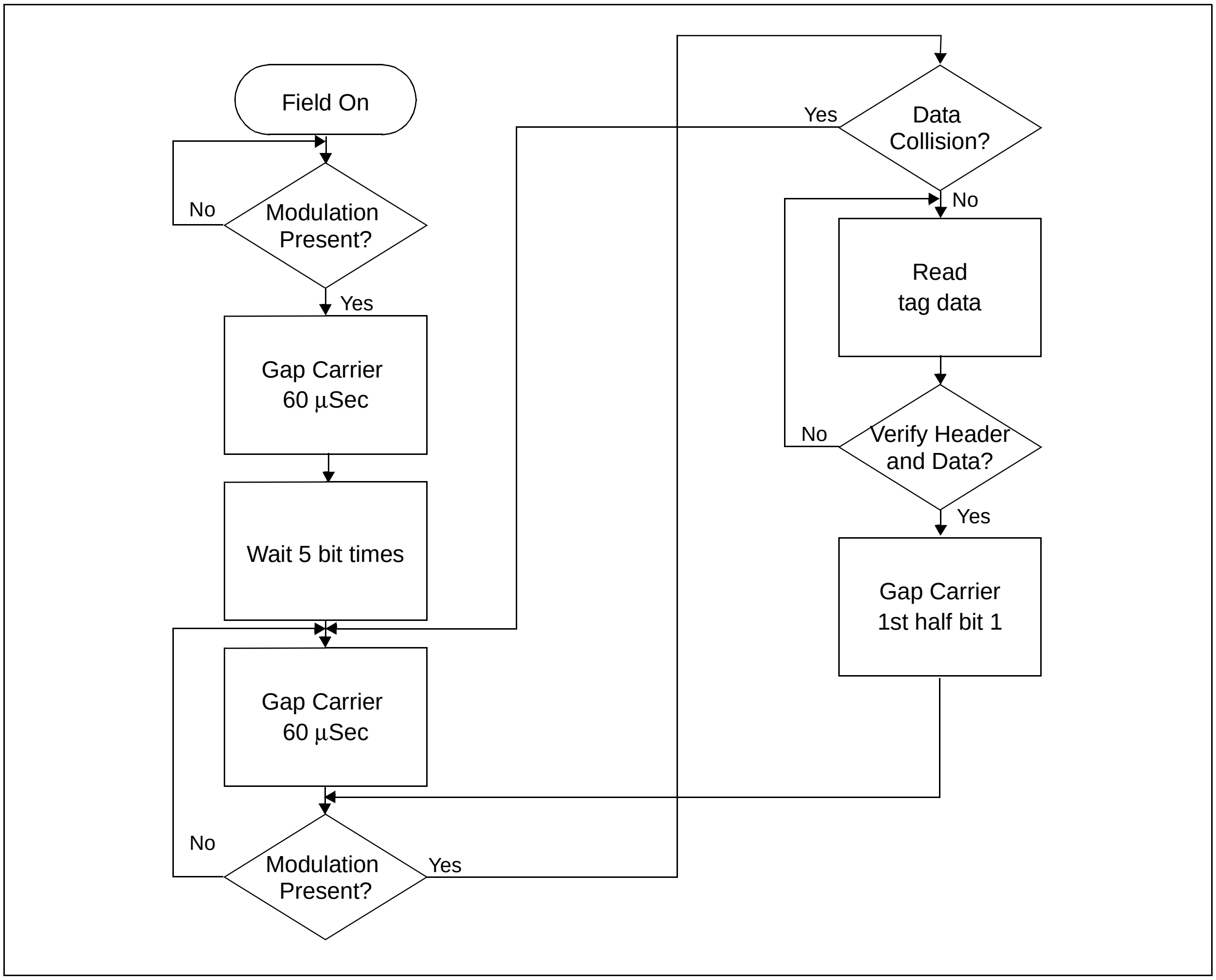
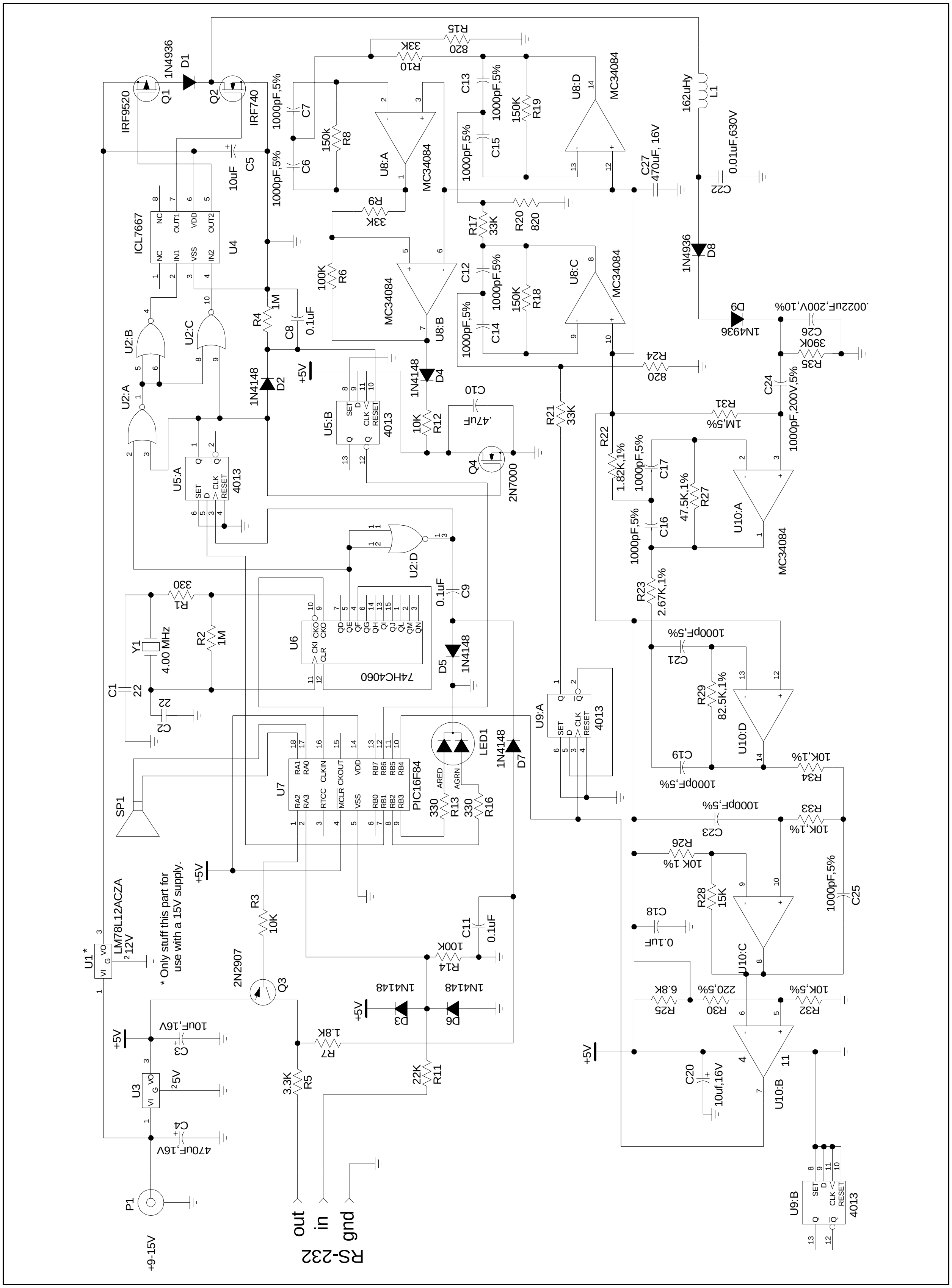


FIGURE 2-2: ANTI-COLLISION ALGORITHM FOR A MCRF250 READER



3.0 ANTI-COLLISION READER SCHEMATIC



4.0 ANTI-COLLISION READER BILL OF MATERIALS

Quantity	Type	Value	Reference Designator	Part Number
1	PIEZO Buzzer	PKM17EPP-4001	BZ1	MURATA PART #
2	Capacitor	22 pF	C1, C2	1330PH-ND
12	Capacitor	lOOO pF, 5%	C6,C7,C12,C13,C14C15,C16, C17,C19,C21,C23,C25	P4937-ND
1	Capacitor	1000 pF, 200V, 5%	C24	(P3497-ND)
1	Capacitor	.0022 µF, 200V, 10%	C26	P3S01-ND
1	Capacitor	0.01µF, 630V	c22	P3509-ND
4	Capacitor	0.1 µF	C8, C9, C11, C18	P4539-ND
1	Capacitor	0.47 µF	C10	P4967-ND
3	Capacitor	10 µF, 16V	C3, C5, C20	P6616-ND
1	Capacitor	l00 µF, 25V	C4	P10269-ND
1	Capacitor	470 µF, 16V	C27	P10247-ND
6	Diode	1N4148	D2, D3, D4, D5, D6, D7	1N4148DICT-ND
3	Diode	1N4936	D1, DB, D9	1N4936CT-ND
1	Bicolor LED	P392	LED 1	p392-ND
1	Coil Antenna	162 uBy	L1	Custom Wound
1	P-Chain MOSFET	IRF9520	Q1	IRF9520 FUTURE
1	N-Chain MOSFET	IRF740	Q2	IRF740-ND
1	PNP Transistor	2N2907	Q3	PN2907A-ND
1	N-Chain MOSFET	2N7000	Q4	2N7000DICT-ND
1	Resistor	220, 5%	R30	5043CX220ROJ
3	Resistor	330, 5%	R1, R13, R16	5043CX330ROJ
3	Resistor	820, 5%	R15, R20, R24	5043CX830ROJ
	Resistor	1.8K, 5%	R7	1K8CR-1/4W-B 5%
1	Resistor	1.82K, 1%	R22	1K82MF-1/4W-B 5%
1	Resistor	2.67K, 15	R23	2K67MF-1/4W-B 1%
1	Resistor	3.3K, 5%	R5	3K3CR-1/4W-B 5%
1	Resistor	6.8K, 5%	R25	6K8CR-1/4W-B 5%
3	Resistor	10R,1’	R26, R33, R34	5043ED10KOOF
3	Resistor	10K, 5%	R3, R12, R32	10KCR-1/4W-B 5%
1	Resistor	15K, 5%	R28	15KCR-1/4W-B 5%
1	Resistor	22K, 5%	R11	22KCR-1/4W-B 5%
4	Resistor	33K, 5%	R9,R10, R17, R21	33JCR1/4-B 5%
1	Resistor	47.5K, 1%	R27	47K5MF-1/4W-B 1%
1	Resistor	82.5K, 1%	R29	82.5KMF-1/4W-B 1%
2	Resistor	100K, 5%	R6, R14	5043CX100KOJ
3	Resistor	150K, 5%	R8, R18, R19	150KCR-1/4W-B 5%
1	Resistor	390K, 5%	R35	390KCR-1/4-B, 5%
3	Resistor	1M, 5%	R2, R4, R31	1MOCR-1/4W-B 5%
1	QUAD NOR GATE	74HC02	U2	MM74HC02N-ND
1	5V Regulator	LM78L05	U3	NJM78L05A-ND
1	MOSFET Driver	ICL7667	U4	ICL7667CPA-ND
2	DUAL FLIP-FLOP	4013	U5, U9	CD4013BCN-ND

Note: All resistors are 5% 1/4 watt carbon film resistors unless otherwise noted. DIGI-KEY part numbers follow some parts where applicable (these part numbers are only intended as a reference).

Quantity	Type	Value	Reference Designator	Part Number
1	Binary Counter	74HC4060	U6	MM74HC4060N-ND
1	Microprocessor	PIC16F84	U7	PIC16F84-04/P
2	OP-AMP	MC3407	U8, U10	FUTURE PART #
1	Crystal	4.00 MHz	Y1	X405-ND

Note: All resistors are 5% 1/4 watt carbon film resistors unless otherwise noted. DIGI-KEY part numbers follow some parts where applicable (these part numbers are only intended as a reference).

5.0 FSK ANTI-COLLISION SOURCE CODE FOR THE PICmicro® MCU

The following source code is for the PIC16C84 microcontroller used in the FSK reader electronics.

```
; #=#=#=#=#=#=#=#=# PROJECT Microchip FSK anticollision Reader #=#=#=#=#=#=#=#=#
; v008.asm
; PIC16F84 running at 4MHz, Ti=1us

; ///////////////////////////////////////////////////////////////////////////
; Revision history
; ///////////////////////////////////////////////////////////////////////////
;
; Ver    Date        Comment
;
; 0.01   29 Dec 97   Copied from MChip\Reader\FSK
; 0.02   27 Feb 98   Gap during first half of first bit
; 0.05   28 Apr 98   Change from PIC16C84 to PIC16F84
; 0.06   29 Apr 98   Count to 256 instead of to 512
; 0.07   30 Apr 98   Make PORTB.0 low output (previously demodulated data input)
; 0.07a  08 May 98   Make gaps 80us wide
; 0.08   13 Aug 98   TAKE OUT CODE INTENDED FOR LAB USE ONLY
;
;       Tbit=50Tcy=400Ti
;       Ttag=96Tbit
;       Header=h'802A'
;

    processor pic16f84
    #include "p16f84.inc"
        __config b'00000000000001'
        ; Code Protect on, power-up timer on, WDT off, XT oscillator

#define _CARRY          STATUS,0
#define _ZERO           STATUS,2
#define _TO             STATUS,4
#define _RP0            STATUS,5
#define _PAGE0          PCLATH,3

#define _BUZZ1          PORTA,0
#define _BUZZ2          PORTA,1
#define _RS232TX        PORTA,2
#define _RS232RX        PORTA,3
#define _SDA            PORTA,4
StartPORTA      = b'11100'
StartTRISA      = b'01000'
BeepPort        = PORTA
Beep0           = StartPORTA
Beep1           = StartPORTA | b'00001'
Beep2           = StartPORTA | b'00010'

#define _UNUSED1        PORTB,0
#define _COIL_PWR       PORTB,1
#define _LED1           PORTB,2
#define _LED2           PORTB,3
#define _RAW_DATA       PORTB,4
#define _UNUSED2        PORTB,5
#define _COLLISION      PORTB,6 ; <  Goes low when a collision occurs
#define _SCL            PORTB,7
StartPORTB      = b'10000010' ; Coil_Off
StartTRISB      = b'01010000'

StartOPTION     = b'10001111'
; PORTB pullups disabled, TMR0 internal, prescaler off, WDT/256

BO3             = h'0C' ; Could be doubled-up with DelayReg1
DelayReg1       = h'0D' ; Could be doubled-up with BO3
```

```
BitCtr          = h'0E' ; Could be doubled-up with BeepCtrHi
TxByte          = h'0F' ; Could be doubled-up with BeepCtrLo
TagsDetected    = h'10'
GapCountLo      = h'11'
Counter1        = h'12'
Counter2        = h'13'
Flags           = h'14'
#define _GotHeader      Flags,0
#define _FirstTime      Flags,1
Period          = h'15' ; Used to read FSK
GapCountHi      = h'16'

Buffer00        = h'18' ; --- IMMOBILE --- IMMOBILE --- IMMOBILE --- IMMOBILE
Buffer01        = h'19' ; |
Buffer02        = h'1A' ; |
Buffer03        = h'1B' ; |
Buffer04        = h'1C' ; |
Buffer05        = h'1D' ; |
Buffer06        = h'1E' ; |
Buffer07        = h'1F' ; |
Buffer08        = h'20' ; |
Buffer09        = h'21' ; |
Buffer0A        = h'22' ; |
Buffer0B        = h'23' ; |
Buffer0C        = h'24' ; |
Buffer0D        = h'25' ; |
Buffer0E        = h'26' ; |
Buffer0F        = h'27' ; |
Buffer10        = h'28' ; |
Buffer11        = h'29' ; |
Buffer12        = h'2A' ; |
Buffer13        = h'2B' ; |
Buffer14        = h'2C' ; |
Buffer15        = h'2D' ; |
Buffer16        = h'2E' ; |
Buffer17        = h'2F' ; |

BeepCtrHi       = h'30' ; Could be doubled-up with BitCtr
BeepCtrLo       = h'31' ; Could be doubled-up with TxByte

SKIP macro
        BTFSC   PCLATH,7
 endm

Coil_On macro
        BCF     _COIL_PWR
 endm

Coil_Off macro
        BSF     _COIL_PWR
 endm

        org h'0000'             ; *#*#*#* RESET VECTOR *#*#*#*
        CLRF    PCLATH
        CLRF    INTCON
        CLRF    STATUS
        GOTO    RESET_A

        org h'0004'             ; *#*#*#* INTERRUPT VECTOR *#*#*#*
        CLRF    PCLATH
        CLRF    INTCON
        CLRF    STATUS
        GOTO    RESET_A

; ***** Subroutines, Page 0
```

```
Delay10:                        ;[0] Delay 10Ti
        GOTO    Delay08         ; |
Delay08:                        ;[0] Delay 8Ti
        GOTO    Delay06         ; |
Delay06:                        ;[0] Delay 6Ti
        NOP                     ; |
Delay05:                        ;[0] Delay 5Ti
        NOP                     ; |
Delay04:                        ;[0] Delay 4Ti
        RETLW   0               ; |

;%% CALL RS232CR takes 1052Ti
;%% CALL RS232TxDigit takes 1057Ti
;%% CALL RS232TxW takes 1049Ti
RS232CR:                        ;[1] Transmit CR on RS232
        MOVLW   d'13'           ; |
        GOTO    RS232TxW        ; |
RS232TxDigit:                   ;[1] Transmit LSnybble of W on RS232
        ANDLW   h'0F'           ; |
        MOVWF   TxByte          ; |
        MOVLW   h'A'            ; |
        SUBWF   TxByte,W        ; |
        MOVLW   '0'             ; |
        BTFSC   _CARRY          ; |
        MOVLW   'A'-h'A'        ; |
        ADDWF   TxByte,W        ; |
RS232TxW:                       ;[1] Transmit W on RS232 at 9615 baud
        MOVWF   TxByte          ; | TxByte=W
RS232Tx:                        ;[1] Transmit TxByte - 104us = 9615.4 baud
        BSF     _RS232TX        ; | Stop bit
        MOVLW   d'35'           ; | | Delay 106Ti
        MOVWF   DelayReg1       ; | | |
RS232TxD1:                      ; | | |
        DECFSZ  DelayReg1,f     ; | | |
        GOTO    RS232TxD1       ; | | |
        BCF     _RS232TX        ; | Start bit
        NOP                     ; | | Delay 98Ti
        MOVLW   d'32'           ; | | |
        MOVWF   DelayReg1       ; | | |
RS232TxD2:                      ; | | |
        DECFSZ  DelayReg1,f     ; | | |
        GOTO    RS232TxD2       ; | | |
        CLRF    BitCtr          ; | BitCtr=#8
        BSF     BitCtr,3        ; | |
RS232TxL1:                      ; | {% -4Ti
        BTFSC   TxByte,0        ; |   Transmit TxByte.0, RR TxByte
        GOTO    RS232TxBit1     ; |   |
        NOP                     ; |   |
RS232TxBit0:                    ; |   |
        BCF     _RS232TX        ; |   |
        BCF     _CARRY          ; |   |
        GOTO    RS232TxBitDone  ; |   |
RS232TxBit1:                    ; |   |
        BSF     _RS232TX        ; |   |
        BSF     _CARRY          ; |   |
        GOTO    RS232TxBitDone  ; |   |
RS232TxBitDone:                 ; |   |
        RRF     TxByte,f        ; |   |% 4Ti
        MOVLW   d'30'           ; |   Delay 93Ti
        MOVWF   DelayReg1       ; |   |
        GOTO    RS232TxD3       ; |   |
RS232TxD3:                      ; |   |
        DECFSZ  DelayReg1,f     ; |   |
        GOTO    RS232TxD3       ; |   |
        DECFSZ  BitCtr,f        ; |   DEC BitCtr
```

```
        GOTO    RS232TxL1       ; | } until (BitCtr==#0)
        CALL    Delay04         ; | Delay 4Ti
        BSF     _RS232TX        ; | stop bit
        RETLW   0               ; end

DelayTtag:                      ;[?] Delay Ttag-3Ti=38400-3Ti=38397Ti
        BSF     _PAGE0
        GOTO    P1DelayTtag

; ***** End of subroutines, Page 0

RESET_A:
        CLRWDT
                                ; Initialise registers
        CLRF    STATUS          ; | Access register page 0
        CLRF    FSR             ; | FSR=#0
        MOVLW   StartPORTA      ; | Initialise PORT and TRIS registers
        MOVWF   PORTA           ; | |
        MOVLW   StartPORTB      ; | |
        MOVWF   PORTB           ; | |
        BSF     _RP0            ;^| |
        MOVLW   StartTRISA      ;^| |
        MOVWF   TRISA           ;^| |
        MOVLW   StartTRISB      ;^| |
        MOVWF   TRISB           ;^| |
        BCF     _RP0            ; | Initialise OPTION register
        MOVLW   StartOPTION     ; | |
        CLRF    TMR0            ; | |
        BSF     _RP0            ;^| |
        MOVWF   OPTION_REG      ;^| |
        BCF     _RP0            ; | |

BigLoop1:
        CALL    Delay08         ; LEDs “reading”
        BSF     _LED1           ; |
        CALL    Delay08         ; |
        BCF     _LED2           ; |
        CALL    Delay08         ; |
        Coil_Off                ; Turn coil off

        BSF     _PAGE0
        GOTO    ResetDelay
ResetDelayDone:

        CLRF    TagsDetected    ; TagsDetected=#0
        CLRF    GapCountHi      ; GapCount=#0
        CLRF    GapCountLo      ; |
GapLoop:                        ; {
        Coil_Off                ;   Turn coil off
        CALL    Delay08         ;   LEDs “reading”
        BSF     _LED1           ;   |
        CALL    Delay08         ;   |
        BCF     _LED2           ;   |
        CALL    Delay10         ;   Wait 80us
        CALL    Delay10         ;   |
        CALL    Delay10         ;   |
        CALL    Delay10         ;   |
        CALL    Delay10         ;   |
        CALL    Delay10         ;   |
        NOP                     ;   |
        Coil_On                 ;   Turn coil on
        ;% 0 Ti from 1st bit


;(Ttag=38400Ti)
; If it’s the first gap since reset, delay Ttag
```



```
        BTFSC   _FirstTime
        CALL    DelayTtag
        BCF     _FirstTime


        CLRF    DelayReg1       ;   Delay 2047Ti
GapD1:                          ;   |
        CLRWDT                  ;   |
        DECFSZ  DelayReg1,f     ;   |
        GOTO    GapD1           ;   |
GapD2:                          ;   |
        CLRWDT                  ;   |
        DECFSZ  DelayReg1,f     ;   |
        GOTO    GapD2           ;   |
        ;% 2050Ti from 1st bit
        MOVLW   d'8'            ;   DelayReg1=#8
        MOVWF   DelayReg1       ;   |
        ;% 2052Ti from 1st bit
        ;% 2076-3*DelayReg1 from 1st bit
        ;% 5*400+76-3*DelayReg1 from 1st bit
        ;% 76-3*DelayReg1 Ti from 6th bit
                                ;   Read tag, with timeouts everywhere
        MOVLW   d'2'            ;   | Counter2=#2
        MOVWF   Counter2        ;   | |
ReadBit_L1:                     ;   | {% 78-3*DelayReg1 Ti from bit
        MOVLW   d'96'           ;   |   BitCtr=#96
        MOVWF   BitCtr          ;   |   |
ReadBit_L2:                     ;   |   {% 80-3*DelayReg1 Ti from bit
ReadBit_D1:                     ;   |     delay
        DECFSZ  DelayReg1,f     ;   |     |
        GOTO    ReadBit_D1      ;   |     |
                                ;   |    % 79Ti from bit
        CLRF    Counter1        ;   |     Counter1=#0
                                ;   |    % 80Ti=10Tcy from bit, time to start frequency sample
ReadBit_Hi0:                    ;   |     {% 80+(Counter1*8)Ti from bit
        INCF    Counter1,f      ;   |       ++Counter1
                                ;   |      % 73+(Counter1*8)Ti from bit
        BTFSC   Counter1,6      ;   |       if (timeout)
        GOTO    GapX            ;   |       { goto GapX } // could be at 1st half of 1st bit!!!
        NOP                     ;   |       |
        BTFSC   _RAW_DATA       ;   |     } until (_RAW_DATA==#1)
        BTFSS   _RAW_DATA       ;   |     |
        GOTO    ReadBit_Hi0     ;   |     |
        NOP                     ;   |     |
ReadBit_Lo0:                    ;   |     {% 80+(Counter1*8)Ti from bit
        INCF    Counter1,f      ;   |       ++Counter1
                                ;   |      % 73+(Counter1*8)Ti from bit
        BTFSC   Counter1,6      ;   |       if (timeout)
        GOTO    GapX            ;   |       { goto GapX } // could be at 1st half of 1st bit!!!
        NOP                     ;   |       |
        BTFSS   _RAW_DATA       ;   |     } until (_RAW_DATA==#0)
        BTFSC   _RAW_DATA       ;   |     |
        GOTO    ReadBit_Lo0     ;   |     |
        NOP                     ;   |     |
                                ;   |    % 80+(Counter1*8)Ti from bit
        MOVF    Counter1,W      ;   |     Period=Counter1
        MOVWF   Period          ;   |     |
        INCF    Counter1,f      ;   |     |
        CALL    Delay05         ;   |     |
ReadBit_Hi1:                    ;   |     {% 80+(Counter1*8)Ti from bit
        INCF    Counter1,f      ;   |       ++Counter1;
                                ;   |      % 73+(Counter1*8)Ti from bit
        BTFSC   Counter1,6      ;   |       if (timeout)
        GOTO    GapX            ;   |       { goto GapX } // could be at 1st half of 1st bit!!!
```

```
        NOP                      ;   |         |
        BTFSC   _RAW_DATA        ;   |       } until (_RAW_DATA==#1)
        BTFSS   _RAW_DATA        ;   |       |
        GOTO    ReadBit_Hi1      ;   |       |
        NOP                      ;   |       |
ReadBit_Lo1:                     ;   |       {% 80+(Counter1*8)Ti from bit
        INCF    Counter1,f       ;   |         ++Counter1;
                                 ;   |        % 73+(Counter1*8)Ti from bit
        BTFSC   Counter1,6       ;   |         if (timeout)
        GOTO    GapX             ;   |         { goto GapX } // could be at 1st half of 1st bit!!!
        NOP                      ;   |         |
        BTFSS   _RAW_DATA        ;   |       } until (_RAW_DATA==#0)
        BTFSC   _RAW_DATA        ;   |       |
        GOTO    ReadBit_Lo1      ;   |       |
        NOP                      ;   |       |
ReadBit_Hi2:                     ;   |       {% 80+(Counter1*8)Ti from bit
        INCF    Counter1,f       ;   |         ++Counter1;
                                 ;   |      % 73+(Counter1*8)Ti from bit
        BTFSC   Counter1,6       ;   |         if (timeout)
        GOTO    GapX             ;   |         { goto GapX } // could be at 1st half of 1st bit!!!
        NOP                      ;   |         |
        BTFSC   _RAW_DATA        ;   |       } until (_RAW_DATA==#1)
        BTFSS   _RAW_DATA        ;   |       |
        GOTO    ReadBit_Hi2      ;   |       |
        NOP                      ;   |       |
ReadBit_Lo2:                     ;   |       {% 80+(Counter1*8)Ti from bit
        INCF    Counter1,f       ;   |         ++Counter1;
                                 ;   |        % 73+(Counter1*8)Ti from bit
        BTFSC   Counter1,6       ;   |         if (timeout)
        GOTO    GapX             ;   |         { goto GapX } // could be at 1st half of 1st bit!!!
        NOP                      ;   |         |
        BTFSS   _RAW_DATA        ;   |       } until (_RAW_DATA==#0)
        BTFSC   _RAW_DATA        ;   |       |
        GOTO    ReadBit_Lo2      ;   |       |
        NOP                      ;   |       |
                                 ;   |      % 80+(Counter1*8)Ti from bit
        MOVF    Period,W         ;   |       Period=Counter1-Period
        SUBWF   Counter1,W       ;   |       |
        MOVWF   Period           ;   |       |
                                 ;   |      % 83+(Counter1*8)Ti from bit
        COMF    Counter1,W       ;   |       W=32-Counter1
        ADDLW   d'1'             ;   |       |
        ADDLW   d'32'            ;   |       |
                                 ;   |      % 86+(32-W)*8Ti from bit
                                 ;   |      % 86+(Counter1*8)Ti from bit
        INCF    Counter1,f       ;   |       ++Counter1
        INCF    Counter1,f       ;   |       ++Counter1
        NOP                      ;   |       |
                                 ;   |      % 89+(32-W)*8Ti from bit
                                 ;   |      % 73+(Counter1*8)Ti from bit
        BTFSS   _CARRY           ;   |       if (W<0)
        GOTO    GapX             ;   |       { goto GapX } // could occur in 1st half of 1st bit!!!
                                 ;   |      % 91+(32-W)*8Ti from bit
        MOVWF   Counter1         ;   |       Counter1=W
                                 ;   |      % 92+(32-Counter1)*8 Ti from bit
ReadBit_D2:                      ;   |       Delay 4+Counter1*8 Ti
        MOVF    Counter1,f       ;   |       |
        BTFSC   _ZERO            ;   |       |
        GOTO    ReadBit_D2_done  ;   |       |
        NOP                      ;   |       |
        NOP                      ;   |       |
        DECF    Counter1,f       ;   |       |
        GOTO    ReadBit_D2       ;   |       |
ReadBit_D2_done:                 ;   |       |
                                 ;   |      % 92+32*8-(oldCounter1)*8+4+(oldCounter1)*8 Ti from bit
                                 ;   |      % 352Ti from bit
```

```
        BTFSS   _COLLISION      ;   |      if (collision occurred)
        GOTO    Gap1            ;   |      { goto Gap1 } // after 1st half of bit
        MOVF    Period,W        ;   |      if (Period<#14)
        ADDLW   low(0-d'14')    ;   |      |
        BTFSS   _CARRY          ;   |      |
        GOTO    Gap0            ;   |      { goto Gap0 } // after 1st half of bit
        ADDLW   low(d'14'-d'18');   |      if (Period<#18)
        BTFSS   _CARRY          ;   |      |
        GOTO    ReadBit_Got0    ;   |      { goto ReadBit_Got0 }
        ADDLW   low(d'18'-d'22');   |      if (Period>=#22)
        BTFSC   _CARRY          ;   |      |
        GOTO    Gap0            ;   |      { goto Gap0 } // after 1st half of bit

ReadBit_Got1:                   ;   |     % 364Ti from bit
        BSF     _CARRY          ;   |      _CARRY=#1
        GOTO    ReadBit_GotBit  ;   |      goto ReadBit_GotBit

ReadBit_Got0:                   ;   |     % 362Ti from bit
        NOP                     ;   |      |
        NOP                     ;   |      |
        BCF     _CARRY          ;   |      |
        GOTO    ReadBit_GotBit  ;   |      |

ReadBit_GotBit:                 ;   |     % 367Ti from bit
        RLF     Buffer00,f      ;   |      roll in _CARRY
        RLF     Buffer01,f      ;   |      |
        RLF     Buffer02,f      ;   |      |
        RLF     Buffer03,f      ;   |      |
        RLF     Buffer04,f      ;   |      |
        RLF     Buffer05,f      ;   |      |
        RLF     Buffer06,f      ;   |      |
        RLF     Buffer07,f      ;   |      |
        RLF     Buffer08,f      ;   |      |
        RLF     Buffer09,f      ;   |      |
        RLF     Buffer0A,f      ;   |      |
        RLF     Buffer0B,f      ;   |      |
        RLF     Buffer0C,f      ;   |      |
        RLF     Buffer0D,f      ;   |      |
        RLF     Buffer0E,f      ;   |      |
        RLF     Buffer0F,f      ;   |      |
        RLF     Buffer10,f      ;   |      |
        RLF     Buffer11,f      ;   |      |
        RLF     Buffer12,f      ;   |      |
        RLF     Buffer13,f      ;   |      |
        RLF     Buffer14,f      ;   |      |
        RLF     Buffer15,f      ;   |      |
        RLF     Buffer16,f      ;   |      |
        RLF     Buffer17,f      ;   |      |
                                ;   |     % 391Ti from bit
                                ;   |     % -9Ti from bit (Tbit=400Ti)
        MOVLW   d'28'           ;   |      DelayReg1=#28
        MOVWF   DelayReg1       ;   |      |
                                ;   |     % -7Ti from bit
                                ;   |     % 77-3*DelayReg1 Ti from bit
        DECFSZ  BitCtr,f        ;   |      DEC BitCtr
        GOTO    ReadBit_L2      ;   |    } until (BitCtr==#0)
                                ;   |  % -5Ti from bit
        MOVLW   d'26'           ;   |    DelayReg=#26
        MOVWF   DelayReg1       ;   |    |
                                ;   |  % -3Ti from bit
                                ;   |  % 75-3*DelayReg1 Ti from bit
        DECFSZ  Counter2,f      ;   |    DEC Counter2
        GOTO    ReadBit_L1      ;   | } until (Counter2==#0)

                                ;  % -1Ti from first bit
```

```
        BSF     _PAGE0          ;   Delay 1568Ti
        GOTO    BigDelay        ;   |
BigDelayDone:                   ;   |% 1567Ti from first bit

CheckTtag:                      ;   if (tag is not 96 bits long) { goto Gap2 }
        MOVLW   Buffer00        ;   | FSR=#Buffer00
        MOVWF   FSR             ;   | |
        MOVLW   h'0C'           ;   | Counter1=h'0C'
        MOVWF   Counter1        ;   | |
CheckTTagLoop:                  ;   | {% 1571+(12-Counter1)*15Ti from first bit
        BTFSS   _COLLISION      ;   |   if (collision occurred)
        GOTO    Gap1            ;   |   { goto Gap1 } // never happens during first bit
        MOVF    INDF,W          ;   |   Counter2=INDF
        MOVWF   Counter2        ;   |   |
        MOVLW   h'0C'           ;   |   FSR=FSR+h'0C'
        ADDWF   FSR,f           ;   |   |
        MOVF    INDF,W          ;   |   if (Couter2!=INDF)
        XORWF   Counter2,W      ;   |   |
        BTFSS   _ZERO           ;   |   |
        GOTO    Gap2            ;   |   { goto Gap2 } // never happens during first bit
        MOVLW   low(0-h'0C'+1)  ;   |   FSR=FSR-h'0C'+1
        ADDWF   FSR,f           ;   |   |
        DECFSZ  Counter1,f      ;   |   DEC Counter1
        GOTO    CheckTTagLoop   ;   | } until (Counter1==#0)
                                ;  % 1570+12*15Ti = 1752Ti from first bit
HeadSearch:                     ;   if (no header in Buffer) { goto Gap2 }
        MOVLW   d'96'           ;   | set BitCtr
        MOVWF   BitCtr          ;   | |
HeadSearchL1:                   ;   | {% 1752+(96-BitCtr)*31 Ti from first bit
        BTFSS   _COLLISION      ;   |   if (collision occurred)
        GOTO    Gap1            ;   |   { goto Gap1 } // never happens during 1st bit
        BSF     _GotHeader      ;   |   if (header found) { goto HeadFound }
        MOVF    Buffer0B,W      ;   |   |
        XORLW   h'80'           ;   |   |
        BTFSS   _ZERO           ;   |   |
        BCF     _GotHeader      ;   |   |
        MOVF    Buffer0A,W      ;   |   |
        XORLW   h'2A'           ;   |   |
        BTFSS   _ZERO           ;   |   |
        BCF     _GotHeader      ;   |   |
        BTFSC   _GotHeader      ;   |   |
        GOTO    HeadFound       ;   |   |
        RLF     Buffer00,f      ;   |   ROL Buffer
        RLF     Buffer01,f      ;   |   |
        RLF     Buffer02,f      ;   |   |
        RLF     Buffer03,f      ;   |   |
        RLF     Buffer04,f      ;   |   |
        RLF     Buffer05,f      ;   |   |
        RLF     Buffer06,f      ;   |   |
        RLF     Buffer07,f      ;   |   |
        RLF     Buffer08,f      ;   |   |
        RLF     Buffer09,f      ;   |   |
        RLF     Buffer0A,f      ;   |   |
        RLF     Buffer0B,f      ;   |   |
        BCF     Buffer00,0      ;   |   |
        BTFSC   _CARRY          ;   |   |
        BSF     Buffer00,0      ;   |   |
        DECFSZ  BitCtr,f        ;   |   DEC BitCtr
        GOTO    HeadSearchL1    ;   | } until (BitCtr==#0)
                                ;   |% 1751+96*31 Ti = 4727Ti from first bit
        GOTO    Gap2            ;   | goto Gap2 // never happens during first bit
HeadFound:                      ;  % 1766+(96-BitCtr)*29 Ti from first bit
                                ;   Delay to fixed time
HeadDelay:                      ;   | {% 1766+(96-BitCtr)*31 Ti from first bit
        BTFSS   _COLLISION      ;   |   if (collision occurred)
        GOTO    Gap1            ;   |   { goto Gap1 } // never happens during 1st bit
```



```
        CALL    Delay08         ;   |   Delay 26Ti
        CALL    Delay08         ;   |   |
        CALL    Delay06         ;   |   |
        CALL    Delay04         ;   |   |
        DECFSZ  BitCtr,f        ;   |   DEC BitCtr
        GOTO    HeadDelay       ;   | } until (BitCtr==#0)
                                ;  % 1765+96*31 = 4741Ti from first bit

        BTFSS   _COLLISION      ;   if (collision occurred)
        GOTO    Gap1            ;   { goto Gap1 } // never happens during 1st bit

                                ;  % 4743Ti from first bit

        BSF     _LED2           ;   LEDs "Found tag"
        CALL    Delay08         ;   |
        BCF     _LED1           ;   |
                                ;  % 4753Ti from first bit

        SWAPF   Buffer0B,W      ;   Transmit tag ID
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer0B,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer0A,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer0A,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer09,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer09,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer08,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer08,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer07,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer07,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer06,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer06,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer05,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer05,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer04,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer04,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer03,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer03,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer02,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer02,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer01,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer01,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        SWAPF   Buffer00,W      ;   |
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
        MOVF    Buffer00,W      ;   |
```

```
        CALL    RS232TxDigit    ;   |%% CALL RS232TxDigit takes 1057Ti
;% 30145Ti from first bit

        CALL    RS232CR         ;%% CALL RS232CR takes 1052Ti
;% 31197Ti from first bit

        MOVLW   d'255'          ;   Delay 7396Ti
        MOVWF   DelayReg1       ;   |
WaitingL1:                      ;   |
        CLRWDT                  ;   |
        CALL    Delay10         ;   |
        CALL    Delay10         ;   |
        CALL    Delay05         ;   |
        DECFSZ  DelayReg1,f     ;   |
        GOTO    WaitingL1       ;   |

;% 38593Ti from first bit
;% 38400+193 = 193Ti from first bit, -7Ti from gap
        INCFSZ  GapCountLo,f    ;   INC GapCount
        SKIP                    ;   |
        INCF    GapCountHi,f    ;   |
        BTFSC   GapCountHi,0    ; } until (GapCount>#257)
        BTFSS   GapCountLo,1    ; |
        GOTO    GapLoop         ; |
        GOTO    BigLoop1

Gap1: ; !!!!! goto here after collision

                                ;  % -4Ti from gap
        CLRF    GapCountHi
        CLRF    GapCountLo
        GOTO    GapLoop

GapX:                           ;% 76+(Counter1*8)Ti from bit
GapXDelay:                      ; Delay 3+(128-Counter1)*8Ti
        BTFSC   Counter1,7      ; |
        GOTO    GapXDelayDone   ; |
        INCF    Counter1,f      ; |
        NOP                     ; |
        GOTO    GapXDelayJ1     ; |
GapXDelayJ1:                    ; |
        GOTO    GapXDelay       ; |
GapXDelayDone:                  ; |
                                ;% 76+(oldCounter1)*8+3+128*8-(oldCounter1)*8Ti from bit
                                ;% 1103Ti from bit = (400*2)+303Ti from bit
                                ;// Not in first half of bit
Gap0: ; !!!!! goto here for gap which does NOT occur in first half of first bit
                                ;  % -7Ti from gap
        INCFSZ  GapCountLo,f    ;   INC GapCount
        SKIP                    ;   |
        INCF    GapCountHi,f    ;   |

        BTFSC   GapCountHi,0    ; } until (GapCount>#257)
        BTFSS   GapCountLo,1    ; |
        GOTO    GapLoop         ; |
        GOTO    BigLoop1

Gap2: ; !!!!! goto here for valid FSK but invalid code
        INCFSZ  GapCountLo,f    ;   INC GapCount
        SKIP                    ;   |
        INCF    GapCountHi,f    ;   |
        BTFSC   GapCountHi,0    ; } until (GapCount>#257)
        BTFSS   GapCountLo,1    ; |
        GOTO    GapLoop         ; |
        GOTO    BigLoop1
```

```
        org h’0200’
P1Delay20:
        GOTO    P1Delay18
P1Delay18:
        NOP
P1Delay17:
        NOP
P1Delay16:
        GOTO    P1Delay14
P1Delay14:
        NOP
P1Delay13:
        NOP
P1Delay12:
        GOTO    P1Delay10
P1Delay10:
        GOTO    P1Delay08
P1Delay08:
        GOTO    P1Delay06
P1Delay06:
        GOTO    P1Delay04
P1Delay04:
        RETLW   0

BigDelay:
;!!!!! delay (1568-6)Ti = 1562Ti

        MOVLW   d’15’           ; Delay 1501Ti
        MOVWF   DelayReg1       ; |
BigDelayL1:                     ; |
        CALL    P1Delay20       ; |
        CALL    P1Delay20       ; |
        CALL    P1Delay20       ; |
        CALL    P1Delay20       ; |
        CALL    P1Delay17       ; |
        DECFSZ  DelayReg1,f     ; |
        GOTO    BigDelayL1      ; |
        CALL    P1Delay20       ; Delay 61Ti
        CALL    P1Delay20       ; |
        CALL    P1Delay20       ; |
        NOP                     ; |

        BCF     _PAGE0
        GOTO    BigDelayDone


P1DelayTtag:                     ; Delay 38393Ti
        CLRF    DelayReg1       ; | Delay 38144Ti
P1DelayTtagL1:                  ; | |
        CALL    P1Delay20       ; | |
        CALL    P1Delay20       ; | |
        CALL    P1Delay20       ; | |
        CALL    P1Delay20       ; | |
        CALL    P1Delay20       ; | |
        CALL    P1Delay20       ; | |
        CALL    P1Delay20       ; | |
        CALL    P1Delay06       ; | |
        DECFSZ  DelayReg1,f     ; | |
        GOTO    P1DelayTtagL1   ; | |
        MOVLW   d’19’           ; | Delay 248Ti
        MOVLW   DelayReg1       ; | |
P1DelayTtagL2:                  ; | |
        CALL    P1Delay10       ; | |
        DECFSZ  DelayReg1,f     ; | |
        GOTO    P1DelayTtagL2   ; | |
```

```
        NOP                     ; | Delay 1Ti
        BCF     _PAGE0
        RETLW   0

ResetDelay:
        CALL    RS232CR         ; Transmit CR regularly
        MOVLW   d'4'            ; Beep at 3597Hz for 1024 cycles
        MOVWF   BeepCtrHi       ; |
        MOVLW   d'0'            ; |
        MOVWF   BeepCtrLo       ; |% 27277Ti from first bit
BeepLoopJ1:                     ; |
        GOTO    BeepLoopJ2      ; |
BeepLoopJ2:                     ; |
        MOVLW   Beep1           ; |
        MOVWF   BeepPort        ; |
        MOVLW   d'34'           ; | Delay 137Ti
        MOVWF   DelayReg1       ; | |
BeepD1:                         ; | |
        CLRWDT                  ; | |
        DECFSZ  DelayReg1,f     ; | |
        GOTO    BeepD1          ; | |
        MOVLW   Beep2           ; |
        MOVWF   BeepPort        ; |
        MOVLW   d'32'           ; | Delay 132Ti
        MOVWF   DelayReg1       ; | |
        NOP                     ; | |
        GOTO    BeepD2          ; | |
BeepD2:                         ; | |
        CLRWDT                  ; | |
        DECFSZ  DelayReg1,f     ; | |
        GOTO    BeepD2          ; | |
        DECFSZ  BeepCtrLo,f     ; |
        GOTO    BeepLoopJ1      ; |
        DECFSZ  BeepCtrHi,f     ; |
        GOTO    BeepLoopJ2      ; |
        NOP                     ; |
        MOVLW   Beep0           ; |
        MOVWF   BeepPort        ; |

        MOVLW   d'20'           ; Wait ~10ms (reset gap)
        MOVWF   Counter1        ; |
ResetGapL1:                     ; |
        MOVLW   d'124'          ; | Wait ~500us
        MOVWF   DelayReg1       ; | |
ResetGapL2:                     ; | |
        CLRWDT                  ; | |
        DECFSZ  DelayReg1,f     ; | |
        GOTO    ResetGapL2      ; | |
        DECFSZ  Counter1,f      ; |
        GOTO    ResetGapL1      ; |
        BSF     _FirstTime

        Coil_On                 ; Turn coil on
        MOVLW   d'6'            ; Wait ~6ms
        MOVWF   Counter1        ; |
ResetDelayL1:                   ; |
        MOVLW   d'250'          ; |
        MOVWF   DelayReg1       ; |
ResetDelayL2:                   ; |
        CLRWDT                  ; |
        DECFSZ  DelayReg1,f     ; |
        GOTO    ResetDelayL2    ; |
        DECFSZ  Counter1,f      ; |
        GOTO    ResetDelayL1    ; |
        BCF     _PAGE0
        GOTO    ResetDelayDone

        end
```

Using the microID® Programmer

1.0 INTRODUCTION

Microchip's MCRF2XX family of RFID products is normally programmed at the factory (SQTP™ – see Technical Brief TB023 (DS91023), but can be contactlessly programmed by hand during system development using a microID development kit or programmer. A contactless programmer (PG103001), user interface software (rfLAB™), and a host computer are needed to program the MCRF2XX devices. The programmer can also be controlled by a standard terminal (i.e., c:\windows\terminal.exe) in place of rfLAB software, but rfLAB software is recommended. See Figure 5-1 for the programming sequence.

The microID programmer requires an external power supply (+9 VDC, >750 mA). The rfLAB software runs under Microsoft® (MS) Windows® 95, 98, 2000 and XP. The programmer communicates with a host computer via an RS-232 serial interface at 9600 baud, 8 data bits, 1 Stop bit and no parity.

The PG103001 programmer (also included in DV103001 and DV103002 kits) is optimized for programming ISO card and clamshell cards, such as those provided in the DV103001 and DV103002 kits. Other tag sizes and shapes may be programmed as well, but programming yield in that case is unknown.

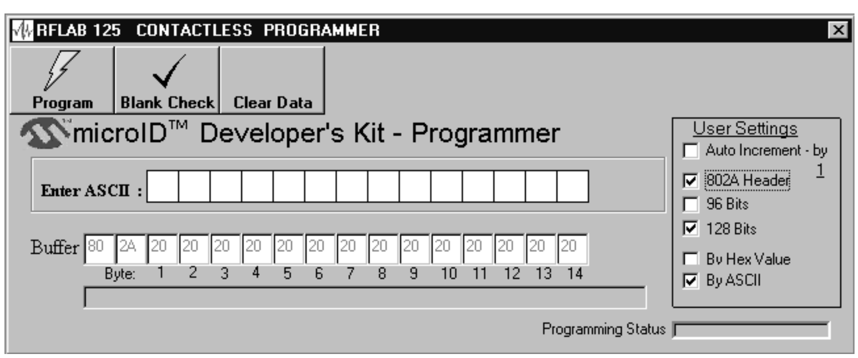
Since the MCRF2XX is a One-Time-Programmable (OTP) device, only a blank (unlocked) device can be programmed by the programmer. Therefore, the programmer first checks the status of the memory in the device before initiating programming. A blank device contains an array of all '1's.

The device can be programmed with 16 bytes (128 bits) or 12 bytes (96 bits) of data length. Once the MCRF2XX enters its Programming mode, it sets a lock bit at the same time. If the programming is interrupted for any reason during the programming period, the programming will be stopped and the device may be left partially programmed. The device will still be locked even though it has not been programmed completely. In this case, the programmer will return a fail code to the host computer.

Any device that has been programmed, either fully or partially, will remain in a locked status; therefore, it cannot to be reprogrammed. If programming has been successfully completed, the programmer will return a verification code to the host computer.

In order to program the MCRF2XX device, it is necessary to provide a proper programming signal level to the device. The device requires specific peak-to-peak voltages for programming. Since the voltage induced in the tag coil varies depending on the coil parameters, the output signal level of the programmer must be calibrated to provide a proper programming signal level at the tag coil. A detailed calibration procedure is described in **Section 3.0 "Calibration of Programming Voltage"**.

FIGURE 1-1: rfLAB™ SOFTWARE RUNNING UNDER MS WINDOWS 95



2.0 PROGRAMMING SIGNAL WAVEFORM

Figure 2-1 shows the waveform of the programming signal. Once the programmer sends a power-up and gap signal to the device, the device transmits back a verification bit stream in FSK. The verification signal represents the contents of the memory in the device. The blank device has all '1's in its memory. A bit '1' in FSK is represented by a low signal level for five cycles and a high signal level for an additional five cycles (Figure 2-1).

The device will respond with a nonmodulated (no data) signal if the device has not recognized the power-up signal. In this case, the power-up signal level should be calibrated to provide a proper signal level to the device. The calibration procedure is explained in **Section 3.0 "Calibration of Programming Voltage"**.

After the device is verified as blank, the programmer sends a programming signal to the device. The programming data is represented by an amplitude modulation signal. Therefore, bit '1' and '0' are represented by a low-power (level) signal and a high-power (level) signal, respectively, as shown in Figure 2-1. Each data bit is represented by 128 cycles of the carrier signal. An microID® 125 kHz Design Guide configured for 128 bits uses all bits in the transfer; an microID® 125 kHz Design Guide configured for 96 bits ignores bits 33 through 64, although they are present in the programming sequence. Therefore, for a 125 kHz carrier signal, it takes 1.024 ms for one data bit (128 cycles x 8 μs/cycles) and 131.072 ms for 128 data bits (128 cycles/bit x 8 μs/cycle x 128 bits).

A guardband of Δt = 10 cycles (80 μs) should be kept at each end of a high power (0) bit as shown in Figure 2-1. This is to prevent accidental programming or disturbing of adjacent bits in the array.

The memory array is locked at the start of the programming cycle. Therefore, when the device leaves the programming field, it locks the memory permanently, regardless of the programming status. The device should not be interrupted during the programming cycle.

The device transmits the programmed (data contents) circuits back to the programmer for verification. If the verification bit stream is correct, the programmer sends a verified signal ('v') to the host computer; otherwise, it sends an error message ('n', see Figure 5-1).

The programming signal level must be within a limit of the programming voltage window for successful programming. The calibration of the signal level is explained in **Section 3.0 "Calibration of Programming Voltage"**.

2.1 Power-up, Gap and Verification Signals

The programming signal starts with a power-up signal for 80 ~ 180 μS, followed by a gap signal (0 volt) for 50 ~ 100 μS. The purpose of these signals is to check whether the device is blank and establish a Programming mode in the device. Once the device recognizes the power-up signal, it transmits back the contents of its memory. If the device transmits back with the blank bit stream (FSK with all '1's), it is ready to be programmed. If the device is not blank, the programmer informs the host computer that it is nonprogrammable.

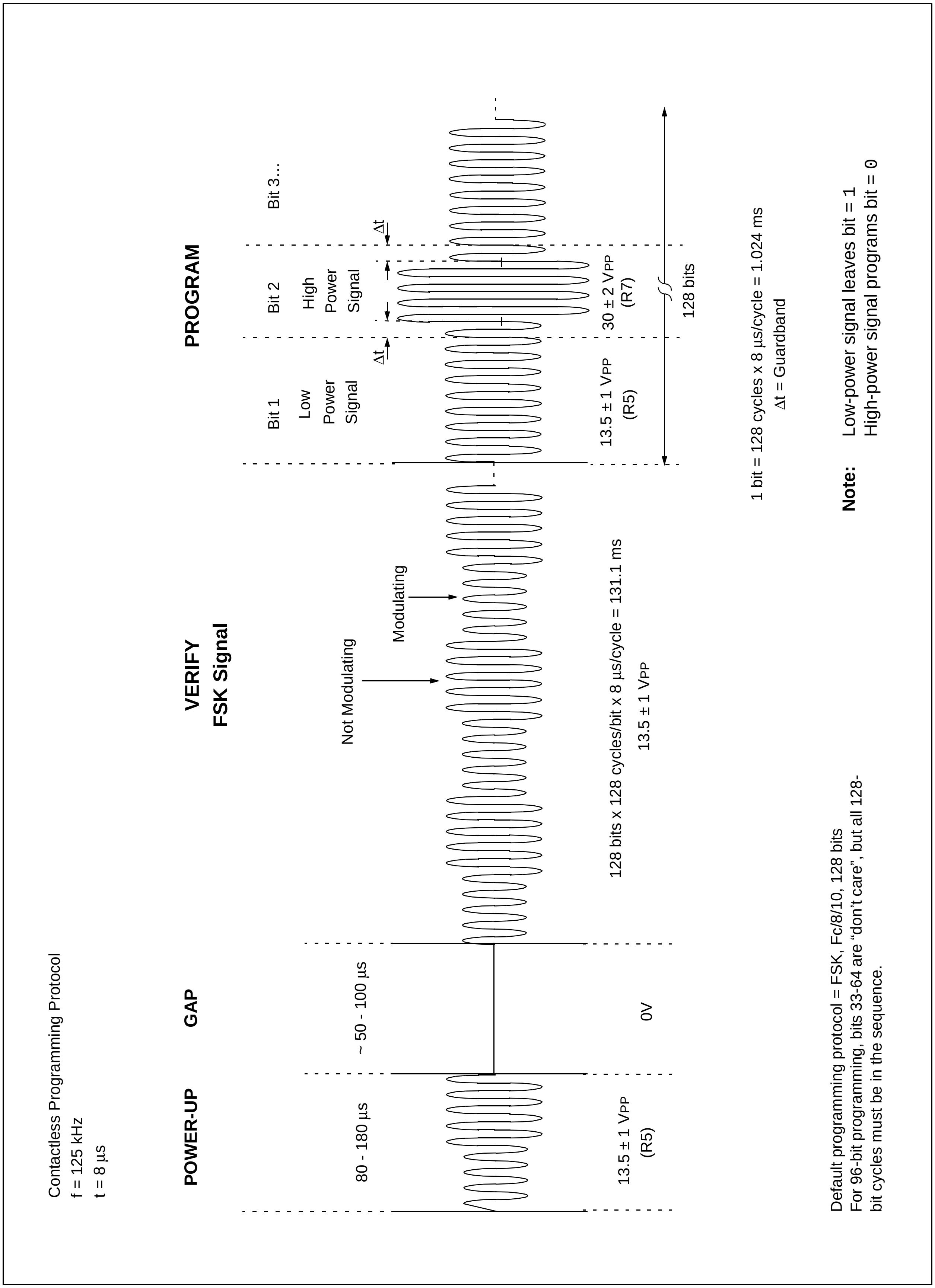
If the power-up signal level is out of the programming voltage range, the device will transmit back a nonmodulated signal (no data). The nonmodulated signal has no variation in the amplitude (constant voltage signal). A variable resistor, R5 in the microID programmer, should be adjusted to provide a proper power-up signal level. A typical signal level is about 22 ± 3 V_{PP} across the tag coil. This calibration procedure is described in **Section 3.0 "Calibration of Programming Voltage"**.

2.2 Programming Sequence

Once the device has been verified blank for programming, the programmer sends a programming sequence to the device. The programming data entered in the rfLAB software is sent to the device via the programmer. The programming signal waveforms are shown in Figure 2-1. One bit of data is represented by 128 cycles of the carrier signal. It takes 131.072 ms to complete one programming cycle for the total of 128 data bits. An microID® 125 kHz Design Guide configured for 128 bits uses all bits in the transfer; an microID® 125 kHz Design Guide configured for 96 bits ignores bits 33 through 64, although they are present in the programming sequence. After the programming sequence, the device transmits back a verification bit stream. The programmer reports to the host computer the status of the programming.

The data is programmed only if the programming signal level is within the limit in the programming voltage requirement of the device. It takes several Programming/Verify cycles to completely program each bit of the microID® 125 kHz Design Guide. The microID programmer uses ten (10) blind Program/Verify cycles before checking the final verify sequence for correct programming. Faster programmers can be designed by checking each Program/Verify cycle; after approximately 3 ~ 5 cycles, the device will verify correctly. Once a correct verify sequence is received, one additional program cycle should be run to ensure proper programming margin.

FIGURE 2-1: CONTACTLESS PROGRAMMING WAVEFORM



3.0 CALIBRATION OF PROGRAMMING VOLTAGE

If you are using your own tag coil (with resonant capacitor) with the microID® 125 kHz Design Guide or MCRF250, you may need to calibrate the programmer for your circuit. Follow this procedure if you are unable to program your tag.

a) Open the programmer, turn R5 and R6 full counter-clockwise. Remove the four screws at the back of the programmer.

b) Set up the programmer and calibration tag as shown in Figure 3-1.

Set Up:

- Connect the +9 VDC power supply to the programmer.
- Connect the RS-232 cable from the external serial port in the programmer box to a COM port in the host.
- Open up the rfLAB software on the host computer.
- Place the calibration tag in the center of the tag area on the programmer. A calibration tag is any tag using microID® 125 kHz Design Guide or MCRF250 silicon and your own coil and capacitor.

c) Run the programming software (rfLAB).

Power-up Signal Level:

d) Click the **Blank Check** button in the rfLAB software.

If the device is blank, a green bar appears in the window with a message indicating that it is blank. If the device is not blank or the power-up signal is out of range, a red bar appears in the window with an error message indicating that it is not blank. The variable resistor (R5) in the programmer should be adjusted to provide a proper “low-power” voltage level to the tag coil. A typical signal level is about 13.5 ± 1 VPP at the tag coil, but it can vary outside of this range.

R5: Turn clockwise in 1/16-inch increments

Repeat step (d) while adjusting R5. Once the device has been verified as a blank, turn it clockwise one more increment. Then move to the next step.

Programming Signal Level:

e) Click on the buttons in rfLAB software for the appropriate data type and protocol for your tag.

f) Enter the programming data in the text box.

g) Click the **Program** button. This will send the programming data to the device. A typical signal level for programming is 32 ± 2 VPP at the tag coil, but can vary outside of this range.

h) After the device has been programmed, it transmits back the programmed data for verification.

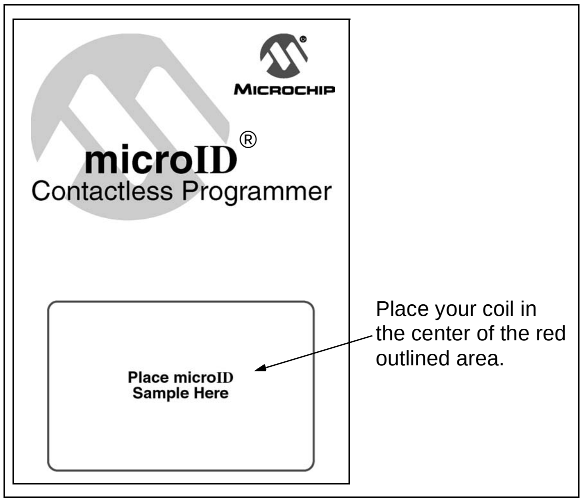
i) If the data has been programmed correctly, a green bar will appear for a few seconds with a message indicating *Programming successful*.

j) If the programming has been unsuccessful due to insufficient programming signal levels, a message indicating *Programming unsuccessful* will appear in the rfLAB software (see Figure 1-1). In this case, R7 (“High Power”) must be adjusted to provide a proper programming signal level to the tag coil. Turn R7 clockwise in 1/16-inch increments, repeating steps (f) through (h) until programming is successful. Then turn R7 clockwise one more increment.

Note:	The microID® 125 kHz Design Guide or MCRF250 lock may be locked even if the programming cycle was unsuccessful; therefore, a new microID® 125 kHz Design Guide sample may be required for each pass through steps (f) through (h).

k) After programming is completed successfully, keep these R5 and R7 settings for future programming of your tags. Once this calibration has been done, remove the calibration tag from the programmer and reinstall the four screws.

FIGURE 3-1: MCRF2XX microID PROGRAMMER AND CALIBRATION TAG COIL ARRANGEMENT FOR PROGRAMMING SIGNAL LEVEL MEASUREMENT



4.0 PROGRAMMING PROCEDURE

a) Set up the programmer and open up the rfLAB software on the host computer.

Set Up:

- Connect the +9 VDC power supply to the programmer.
- Connect from the external serial port in the programmer box to a COM port in the host computer using the RS-232 cable.

b) Place the RFID device at the center of the programmer.

c) Click **Blank Check** button if you want to check whether the device is blank. This button can also be used to verify that the device is assembled properly.

Note:	The device can't be programmed unless it is blank.

d) Enter the programming data in the rfLAB software and select appropriate data type.

e) If several devices are going to be programmed sequentially by any number, click the **Auto Increment** button and specify the increment number.

f) Click the **Program** button. This will send the data to the device.

g) If the data has been programmed correctly, there will be a green bar with a message indicating *Programming successful*.

If the programming has been unsuccessful due to out-of-range in the programming signal level, a message and red bar will show up indicating *Programming unsuccessful.* In this case, the programming signal voltage may need to be calibrated for your tag. See the calibration procedure for the programming signal level in the previous section.

h) Repeat step (a) through (g) for other tags.

4.1 Error Conditions

If the host computer does not send programming data to the programmer for more than 3 seconds, the programmer will time out and reset. If the programmer does not respond to the host computer, there will be an error message indicating *Programmer time out*. If invalid programming data is sent to the programmer during the loading of the program buffer, the programmer will return a message indicating *Invalid.*

4.2 Programming Precautions

Taking these steps will help to ensure proper programming and minimize programming fallout.

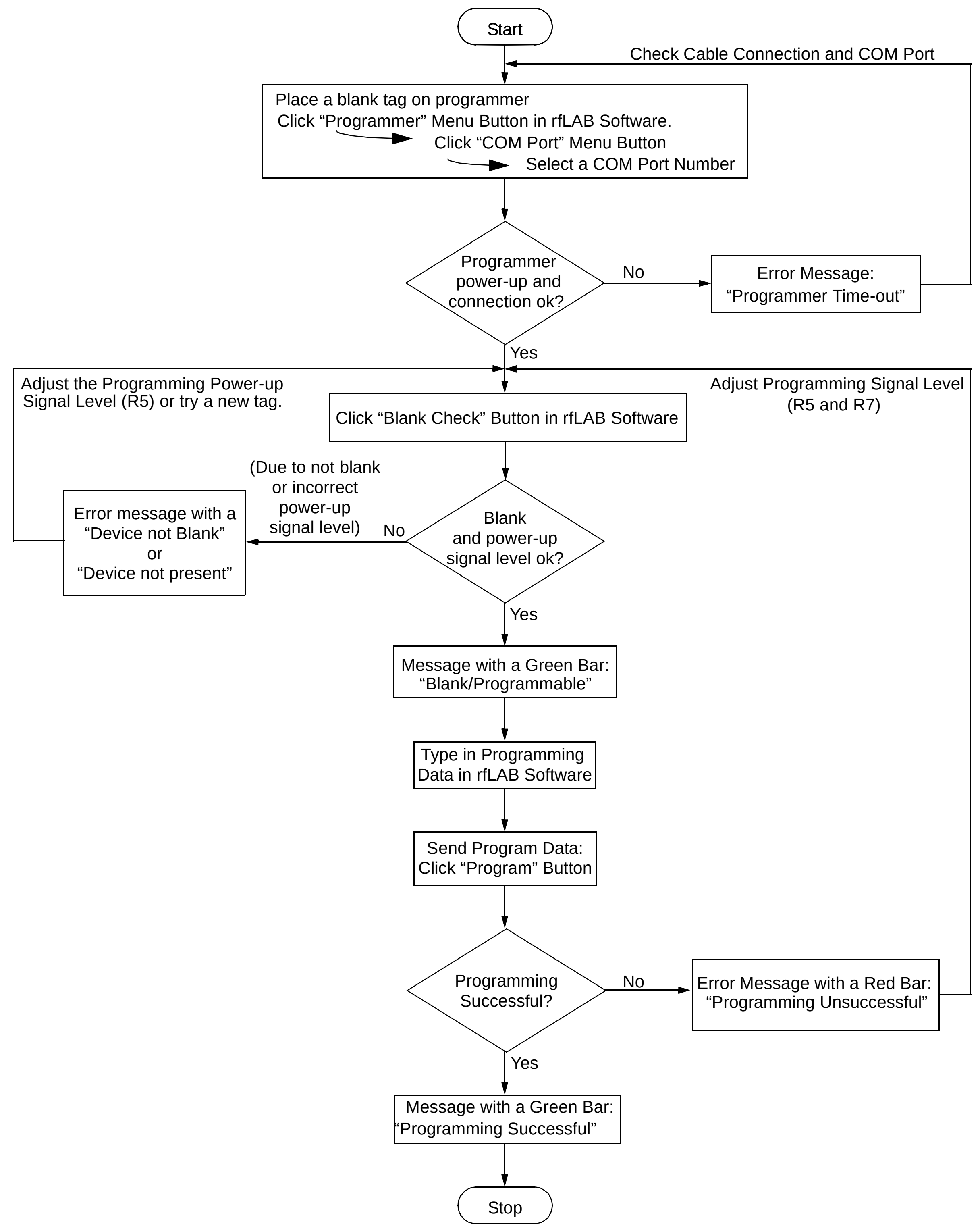
a) If your workbench is metal, set the programmer on top of a cardboard box, stack of books or other nonconductive material to elevate the programmer at least 6" from the metal surface.

b) Because most computer monitors contain an oscillator in the pass band of the programmer, make sure the programmer is at least 18" from any PC monitor.

c) Keep the 9V power supply at least 18" away from the programmer.

d) Do not disturb a tag during blank check or programming cycle; the programmer must be idle when moving a tag onto the programmer or when removing a tag."

FIGURE 4-1: PROGRAMMING FLOW CHART USING rfLAB™ SOFTWARE



5.0 PROGRAMMING IN A STANDARD TERMINAL MODE

In special cases, the device can also be programmed in a standard Terminal mode by executing the `terminal.exe` program (c:\windows\terminal.exe) or by any customer production software. The programmer setup, signal waveforms and calibration procedure are the same as programming with the rfLAB software.

The following is a description of how to interface a host computer to Microchip's contactless programmer without the use of rfLAB software. The programmer will check for a blank, unlocked MCRF2XX tag before initiating programming. Once programming has been completed, the programmer will return a pass or fail code. The programmer communicates at 9600 baud, 8 data bits, 1 Stop bit, and no parity.

Figure 5-1 shows the programming flow and communication handshakes between host and programmer.

5.1 Programmer Wake-up

Sending an ASCII 'W' (57h) to the programmer on the RS-232 interface will tell the programmer to wake-up and be prepared to receive commands. The programmer will reply with ASCII 'R' (52h) when it is ready.

5.2 Blank Check

Sending an ASCII 'T' (54h) will signal the programmer to read the part about being contactlessly programmed and check to see if it is blank (all 1's) and unlocked. If the part is blank and unlocked, the programmer will reply with an ASCII 'Y' (59h) to signify programming should continue. If the part is not blank or not unlocked, the programmer will reply with an ASCII 'N' (4Eh) to indicate an error. It is always necessary to perform a blank check before programming MCRF2XX devices.

5.2.1 SENDING DATA TO THE PROGRAMMER

If the programmer responds with an ASCII 'Y', indicating that the part is blank, the PC can begin passing the 16 bytes of required data to the programmer data buffer. An microID® 125 kHz Design Guide configured for 128 bits uses all 16 bytes of data in the transfer; when programming a 96-bit device, however, bits 33 through 64 are "don't care" and are ignored by the microID® 125 kHz Design Guide. The data should be passed in ASCII equivalent hex bytes and the programmer will acknowledge the receipt of each byte by echoing back what it has received. For example, to program 05 hex data into the first byte, the PC would send ASCII '0' (30h), the programmer would echo '0' back. Next, the programmer would send ASCII '5' (35h), and the programmer will echo back '5'. All of the data must be sent in UPPERCASE ASCII equivalent only. See Figure 5-1 for a typical programming sequence.

5.3 Program and Verify the Device

After 16 bytes of data have been received by the programmer, it is ready to begin programming the data buffer into the MCRF2XX. Sending an ASCII 'V' (56h) will tell the programmer to program the 16 bytes it has received and verify that the device has programmed properly. When the device programs properly, the programmer replies with ASCII 'y' (79h). If the programming was not successful, the programmer replies with ASCII 'n' (6Eh). A successful programming operation should take about 3 to 4 seconds per device.

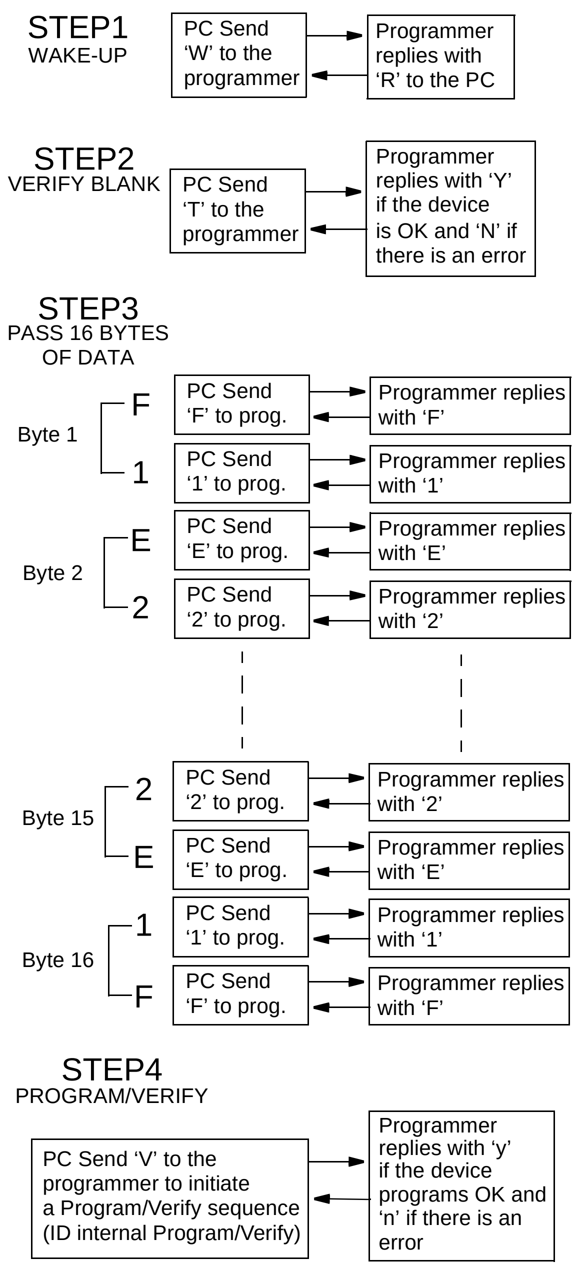
5.4 Error Conditions

If the PC does not send a byte to the programmer for more than 3 seconds, the programmer will time out and reset. The entire programming sequence will need to be repeated, beginning with the programmer wake-up byte ASCII 'W'.

If invalid bytes are sent to the programmer during the loading of the program buffer, the programmer will return an ASCII 'I' (49h). In this case, the entire programming sequence must be repeated, beginning with the programmer wake-up byte ASCII 'W'.

FIGURE 5-1: TYPICAL SEQUENCE

The following is the programming sequence necessary to wake-up the programmer, check if a MCRF2XX part is blank, unlocked and ready to be programmed, send F1E2D3C4B5A6978888796A5B4C3D2E1F ASCII data to the programmer, and instruct the programmer to program and verify the device.

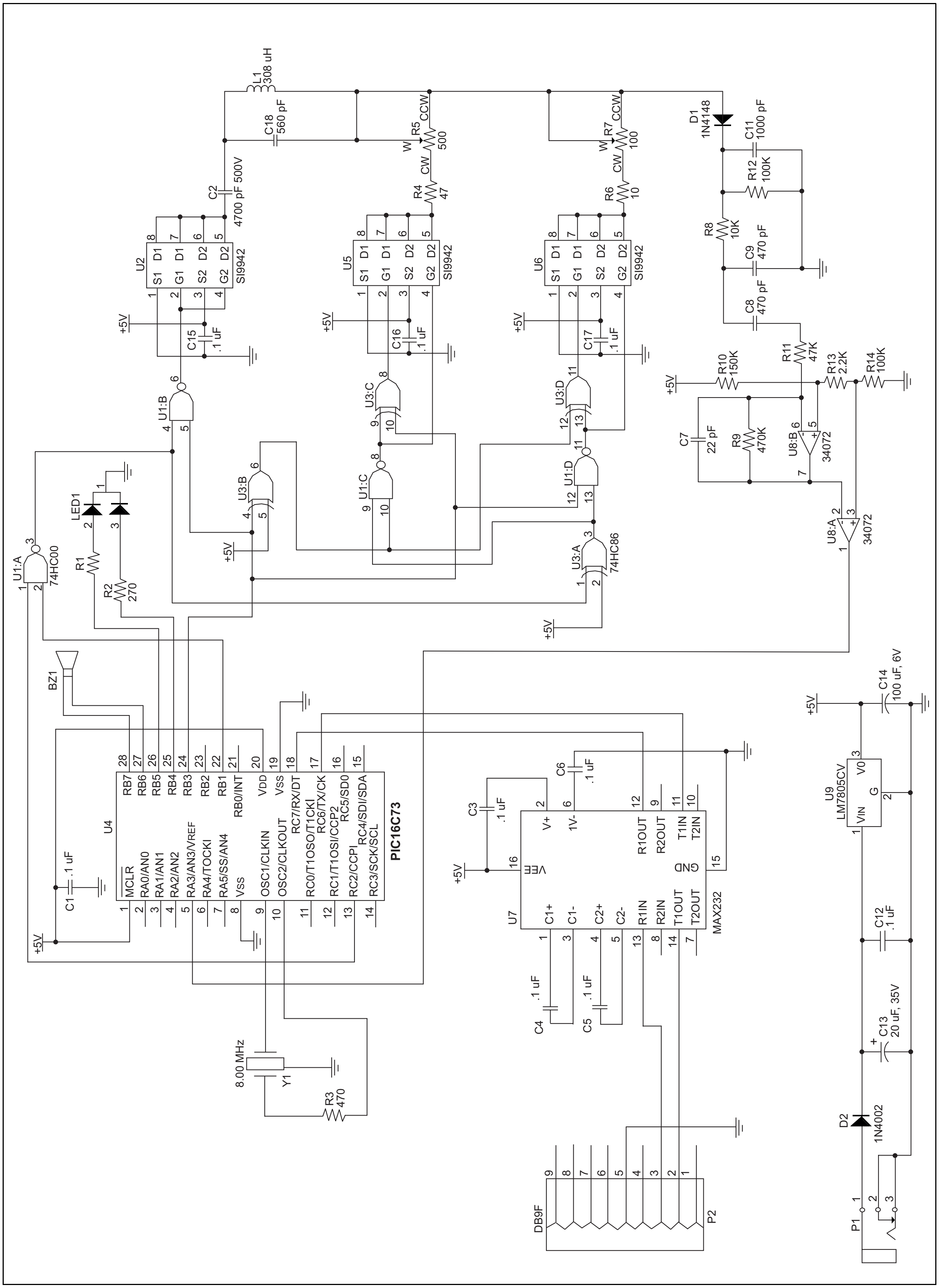


Note: See the signal waveforms and calibration procedure in **Section 2.0 "Programming Signal Waveform"** and **Section 3.0 "Calibration of Programming Voltage"**.

TABLE 5-1: ASCII CHARACTER SET

		Most Significant Characters							
	Hex	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Least Significant Characters	0	NUL	DLE	Space	0	@	P	`	p
	1	SOH	DC1	!	1	A	Q	a	q
	2	STX	DC2	"	2	B	R	b	r
	3	ETX	DC3	#	3	C	S	c	s
	4	EOT	DC4	$	4	D	T	d	t
	5	ENQ	NAK	%	5	E	U	e	u
	6	ACK	SYN	&	6	F	V	f	v
	7	Bell	ETB	'	7	G	W	g	w
	8	BS	CAN	(	8	H	X	h	x
	9	HT	EM	)	9	I	Y	i	y
	A	LF	SUB	*	:	J	Z	j	z
	B	VT	ESC	+	;	K	[	k	{
	C	FF	FS	,	<	L	\	l	|
	D	CR	GS	-	=	M	]	m	}
	E	SO	RS	.	>	N	^	n	~
	F	SI	US	/	?	O	_	o	DEL

6.0 microID® PROGRAMMER SCHEMATIC



7.0 microID® PROGRAMMER BILL OF MATERIALS

Item #	Qty	Part #	Reference Designator	Part Description	Manufacturer	Vendor	Vendor Part #
1	1	ICA-328-S-ST	U4	SOCKET, 28 PIN,.300, MACHINED COLLET	SAMTEC		
2	1	-SPARE-	SP1, LED1, R1, R2	-SPARE- LOCATION DO NOT INSTALL			
3	1	PCC220CNCT-ND	C7	CAP SMT, 22 pF NPO 0805	PANASONIC		
4	2	0805N471J101NT	C8, C9	CAP SMT, 470 pF 5% 100V 0805	MALLORY		
5	1	CD15FC561JO3	C18	CAP, 560 pF, MICA, DIPPED, 300V, AX (0.234LS)	CORNELL DUBILIER	MOUSER	5982-15-300V560
6	1	ECU-V1H102JCX	C11	CAP SMT, 1000 pF 50V NPO CER, 0805	PANASONIC		
7	1	CD19FD472JO3	C2	CAP, 4700 pF, MICA, DIPPED, 500V, AX (0.344LS)	CORNELL DUBILIER	MOUSER	5982-19-500V4700
8	9	250R18Z104MV4E-6	C1, C3-C6, C12, C15-C17	CAP SMT, 0.1 µF 20% 50V 0805	JOHANSON	NEWARK	50F3674
9	1	ECS-H1ED106R	C13	CAP SMT, 10 µF, TANT ELEC, 25V, 7343	PANASONIC	DIGIKEY	PCT5106CT-ND
10	1	ECE-V0JA101SP	C14	CAP SMT, 100 µF, TANT ELEC, 6.3V, (VS-D)	PANASONIC	DIGIKEY	PCE3058CT-ND
11	1	LL4148	D1	DIODE SMT, 5uA, 100V, 500 mW, FAST SWITCHING, DL-35	DIODES INC	DIGIKEY	LL4148DITR-ND
12	1	DL4002	D2	DIODE SMT, RECTIFIER, 1N4002, 1A, 100V, DL-41	DIODES INC.	DIGIKEY	DL4002DITR-ND
13	1	3345P-1-101	R7	RES, POT, 100 OHM 1/2 RD WW ST SL	BOURNS	DIGIKEY	3345P-101-ND
14	1	3345P-1-501	R5	RES, POT, 500 OHM 1/2 RD WW ST SL	BOURNS	DIGIKEY	3345P-501-ND
15	1	ERJ-6GEYJ100	R6	RES SMT, 10 OHM 1/10W 5% TYPE 0805	PANASONIC		P10ACT-ND
16	1	ERJ-6GEYJ470V	R4	RES SMT, 47 OHM 1/10W 5% TYPE 0805	PANASONIC	DIGIKEY	P470ATR-ND
17	1	ERJ-6GEYJ471V	R3	RES SMT, 470 OHM 1/10W 5% TYPE 0805	PANASONIC		P470ATR-ND
18	1	ERJ-6GEYJ222V	R13	RES SMT, 2.2K OHM 1/10W 5% TYPE 0805	PANASONIC		P2.2KATR-ND

Item #	Qty	Part #	Reference Designator	Part Description	Manufacturer	Vendor	Vendor Part #
19	1	ERJ-6GEYJ103V	R8	RES SMT, 10K 1/8W 5% TYPE 0805	PANASONIC	DIGIKEY	P10KATR-ND
20	1	ERJ-6GEYJ473V	R11	RES SMT, 47K OHM 1/10W 5% TYPE 0805	PANASONIC	DIGIKEY	P473ATR-ND
21	2	ERJ-6GEYJ104V	R12, R14	RES SMT, 100K OHM 1/10W 5% TYPE 0805	PANASONIC	DIGIKEY	P100KATR-ND
22	1	ERJ-6GEYJ154V	R10	RES SMT, 150K OHM 1/8W 5% 0805	PANASONIC	DIGIKEY	P150KATR-ND
23	1	ERJ-6GEYJ474V	R9	RES SMT, 470K OHM 1/8W 5% 0805	PANASONIC	DIGIKEY	P470KATR-ND
24	1	MM74HC00M	U1	IC, SMT, 74HC00 QUAD 2 IN NAND (SO-14)	FAIRCHILD SEMICONDUCTOR	DIGIKEY	MM74HC00M-ND
25	3	NDS9942	U2, U5, U6	IC, SMT, 9942 MOSFET N-CH & P-CH 20V (SO-8)	FAIRCHILD SEMICONDUCTOR	DIGIKEY	NDS9942TR-ND
26	1	MM74HC86MX	U3	IC, SMT, 74HC86, QUAD XOR GATE (SO-14)	FAIRCHILD SEMICONDUCTOR	DIGIKEY	
27	1	PIC16C73A /P	U4	IC, PIC16C73A /P, PLASTIC DIP, 28P, 0.300	MICROCHIP		
28	1	MAX232ACSE	U7	IC, MAX232ACSE DUAL RS-232 TRANSMITTER/ RCVR, (SO-16)	MAXIM	DIGIKEY	MAX232ACSE-ND
29	1	MC34072D	U8	IC, DUAL OP AMP, (SO-8)	MOTOROLA		
30	1	L7805CV	U9	IC, REG, +5V, 1.5A, 10%, TO-220	SGS THOMSON	MOUSER	511-L7805CV
31	1	EFO-EC8004A4	Y1	OSC, 8.00 MHz CER RESONATOR W/CAP 3 PIN	PANASONIC	DIGIKEY	PX800-ND
32	1	MCT0003-000	L1	INDUCTOR, 162 µH	CORNEL DUBILIER		
33	1	DE9S-FRS	P2	CONN, D-SUB 9P RECPT RT ANGLE	SPC TECHNOLOGY		
34	1	DJ005B	P1	JACK, POWER, 2.5mm DC PC MOUNT	LZR ELECTRONICS		

8.0 PROGRAMMER SOURCE CODE FOR PIC16C73


```
; #=#=#=#=#=#=#=#=#=#=# PROJECT Microchip Programmer Reader #=#=#=#=#=#=#=#=#=#=#
; #=#=#=#=#=#=#=#=#=#=#          16C73A module               #=#=#=#=#=#=#=#=#=#=#
; rfgopr5.asm
; PIC16C73A running at 8.5MHz, Ti = 0.47us
; Tcy = 16 Ti

; ////////////////////////////////////////////////////////////////////////////
; Revision history
; ////////////////////////////////////////////////////////////////////////////
;
; Ver    Date        Comment
;
; 1.00   10/24/97    Shannon/Hugh first pass
; 1.04   13 Feb 98   ADDED TIMEOUT TO TESTMOD
;
        LISTP=PIC16C73A
    INCLUDE "P16C73A.INC"
    __config b'11111111110010'
    ; Code Protect off, Brown-out detect on, Power-up timer on, WDT off,
    ;   HS oscillator

 constant StartPORTA    = b'000000'
 constant StartTRISA    = b'010111'

 #define _LED1          PORTB,4
 #define _LED2          PORTB,5
 #define _BUZZ1         PORTB,6
 constant StartPORTB    = b'00010010'
 constant StartTRISB    = b'00000100'
 constant StartOPTION   = b'10001000'
        ; Pullups disabled, TMR0 internal, WDT*1

COUNT1EQU0x20  ; COUNT REGISTER
DATA0EQU0x21
DATA1EQU0x22
DATA2EQU0x23
DATA3EQU0x24
DATA4EQU0x25
DATA5EQU0x26
DATA6EQU0x27
DATA7EQU0x28
DATA8EQU0x29
DATA9EQU0x2A
DATAAEQU0x2B
DATABEQU0x2C
DATACEQU0x2D
DATADEQU0x2E
DATAEEQU0x2F
DATAFEQU0x30
BIT EQU 0x31
OVERPROEQU0x32
DELAY1EQU0x33
```



```
DELAY2EQU0x34
DelayReg?H   = h'35'
DelayReg?L   = h'36'
CycleCtr?H   = h'37'
CycleCtr?L   = h'38'
TimerHi      = h'39'
TimerMid     = h'3A'
TimerLo      = h'3B'
BitCtr       = h'3C'
BO3          = h'3D'
RxByte       = h'3E'
TxByte       = h'3F'
ByteCtr      = h'40'
NoiseTimeout = h'41'
SampTimeout  = h'42'
CycleCtr2?L  = h'43'
CycleCtr2?H  = h'44'
 #define _RAW_DATA      PORTA,4
 #define _RS232OUT      PORTC,6
 #define _CARRY         STATUS,0
 #define _TMR2ON        T2CON,2
 #define _RS232IN       PORTC,7
 #define _ZERO          STATUS,2
 #define _COIL_PWR_0    PORTB,3       ; cycle at 30ms period (1=low power)
 #define _COIL_EN       PORTB,1

SKIP macro
        BTFSC   PORTA,7
 endm

; ***** Reset Vector

        org h'000'
        CLRF    STATUS
        CLRF    PCLATH
        CLRF    INTCON
        GOTO    RESET_A

; ***** Interrupt Vector - no interrupts yet

        org h'004'
        CLRF    STATUS
        CLRF    PCLATH
        GOTO    RESET_A

RS232StopBit                    ;[0] Delay >=208 cycles with _RS232OUT high
    BSF     _RS232OUT       ; |
    MOVLW   d'208'-d'12'+d'40'; |
DelayW12                        ;[0] Delay 12+W cycles
    MOVWF   DelayReg?L      ; |
Delay1                          ;[0] Delay 11+Delay cycles
    MOVLW   d'4'            ; |
Delay1L                         ; |
    SUBWF   DelayReg?L,f    ; |
    BTFSC   _CARRY          ; |
    GOTO    Delay1L         ; |
    COMF    DelayReg?L,W    ; |
    ADDWF   PCL,f           ; |
Delay07                         ;[0] Delay 7 cycles
    NOP                     ; |
Delay06                         ;[0] Delay 6 cycles
    NOP                     ; |
Delay05                         ;[0] Delay 5 cycles
    NOP                     ; |
Delay04                         ;[0] Delay 4 cycles
    RETLW   h'00'           ; |
```

```
RESET_A
        CLRWDT                  ; Initialise registers, clear watchdog timer
        CLRF    STATUS          ; | Access register page 0
        CLRF    FSR             ; | FSR=#0
        MOVLW   StartPORTA      ; | Initialise PORT registers
        MOVWF   PORTA           ; | |
        MOVLW   StartPORTB      ; | |
        MOVWF   PORTB           ; | |
        CLRF    INTCON          ; | Interrupts off
        MOVLW   b'110001'       ; | TMR1 prescale *8, on
        MOVWF   T1CON           ; | |
        MOVLW   b'0000000'      ; | TMR2 postscale *1, off, prescale *1
        MOVWF   T2CON           ; | |
        MOVLW   d'8'            ; | Duty on period = 8 Ti @@@
        MOVWF   CCPR1L          ; | |
        MOVLW   b'001100'       ; | CCP1 to PWM, 0,0 extra duty time @@@
        MOVWF   CCP1CON         ; | |
        MOVLW   b'00000000'     ; | A/D convertor OFF
        MOVWF   ADCON0          ; | |
        BSF     STATUS,RP0      ;^| Initialise TRIS registers
        MOVLW   StartTRISA      ;^| |
        MOVWF   TRISA           ;^| |
        MOVLW   StartTRISB      ;^| |
        MOVWF   TRISB           ;^| |
        MOVLW   0x82
        MOVWF   TRISC
        MOVLW   StartOPTION     ;^| Initialise OPTION register
        MOVWF   OPTION_REG      ;^| |
        MOVLW   d'15'           ;^| PR2=7 (period of TMR2=16) @@@
        MOVWF   PR2             ;^| |
        MOVLW   h'03'           ;^| (It says so on page 2-584)
        MOVWF   PCON            ;^| |
        MOVLW   b'110'          ;^| No analog inputs
        MOVWF   ADCON1          ;^| |
        BCF     STATUS,RP0      ; | |


        ; !!!!! set TRIS registers, and other hardware registers.
        BCF     T2CON,2; turn coil off
        CLRF    TMR2
        BCF     PORTB,3

        CALL    RS232On

BigLoop1
    CALL    RS232WaitForever
CheckRxByte
    MOVF    RxByte,W
    XORLW   'W'
    BTFSC   _ZERO
    GOTO    INTERRUPT
    CALL    RS232On
    MOVLW   'Q'
    CALL    RS232TxW
    GOTO    BigLoop1

INTERRUPT
    CALL    Delay07         ; LED1 on, LED2 on (orange/yellow)
    BSF     _LED1           ; |
    CALL    Delay07         ; |
    BSF     _LED2           ; |
    CALL    Delay07         ; |
INT_WAKEUP
        MOVLW   'R'
```



```
        MOVWF   RxByte
    CALL    RS232On         ; delay
    MOVF    RxByte,W        ; Transmit RxByte
    CALL    RS232TxW        ; |
    CALL    RS232Rx         ; Read byte from RS-232
    BTFSC   _CARRY          ; | (if timeout, goto INT_END)
    GOTO    INT_END         ; |
    MOVF    RxByte,W        ; if (RxByte<>#'T')
    XORLW   'T'             ; |
    BTFSS   _ZERO           ; |
    GOTO    CheckRxByte     ; { goto CheckRxByte }


        MOVLW   d'10'
        MOVWF   CycleCtr?H
        CLRF    CycleCtr?L

Top1    BCFPORTB,3; SET FOR LOW VOLTAGE
    CALLDELAY  ; CALL A SMALL DELAY

GAP1; THIS IS THE ROUTINE THAT SETS THE GAP

        BCF     PORTB,3
        CALL    DELAY

    BSF T2CON,2 ; TURN ON THE COIL

    MOVLW0x32  ; MOVE 32 HEX TO W, NUMBER CYCLES BEFORE A GAP
    MOVWFCOUNT1; MOVW W INTO COUNT1
LOOP11DECFSZCOUNT1,1; DECREMENT COUNT 1 UNTIL IT IS ZERO
    GOTOLOOP11

    BCF T2CON,2 ; TURN OFF THE COIL

    MOVLW0x40  ; MOVE 10 HEX TO W, DURATION OF GAP
    MOVWFCOUNT1; MOVW W INTO COUNT1
LOOP21DECFSZCOUNT1,1; DECREMENT COUNT 1 UNTIL IT IS ZERO
    GOTOLOOP21

    BSF T2CON,2 ; TURN THE COIL BACK ON

        CALL    TWC             ; CALL A DELAY FOR AMP TO SETTLE
        CALL    TWC
        CALL    TWC
        CALL    TWC
        CALL    TWC


WaitFall1                       ; Wait for falling edge
WaitFall1A                      ; | Wait for high
        MOVLW   d'200'          ; | | Set timeout
        MOVWF   DelayReg?H      ; | | |
        CLRF    DelayReg?L      ; | | |
WaitFall1AL                     ; | | { {
        DECFSZ  DelayReg?L,f    ; | |     if (timeout)
        SKIP                    ; | |     |
        DECFSZ  DelayReg?H,f    ; | |     |
        SKIP                    ; | |     |
        GOTO    INT_ErrorN      ; | |     { goto INT_ErrorN }
        BTFSS   _RAW_DATA       ; | |   } until (_RAW_DATA==#1)
        GOTO    WaitFall1AL     ; | |   |
        NOP                     ; | |   |
        DECFSZ  DelayReg?L,f    ; | |   if (timeout)
        SKIP                    ; | |   |
        DECFSZ  DelayReg?H,f    ; | |   |
```

```
        SKIP                    ; | |   |
        GOTO    INT_ErrorN      ; | |   { goto INT_ErrorN }
        BTFSS   _RAW_DATA       ; | | } until (_RAW_DATA==#1)
        GOTO    WaitFall1AL     ; | |
WaitFall1B                      ; | Wait for low
        MOVLW   d'200'          ; | | Set timeout
        MOVWF   DelayReg?H      ; | | |
        CLRF    DelayReg?L      ; | | |
WaitFall1BL                     ; | | { {
        DECFSZ  DelayReg?L,f    ; | |     if (timeout)
        SKIP                    ; | |     |
        DECFSZ  DelayReg?H,f    ; | |     |
        SKIP                    ; | |     |
        GOTO    INT_ErrorN      ; | |     { goto INT_ErrorN }
        BTFSC   _RAW_DATA       ; | |   } until (_RAW_DATA==#0)
        GOTO    WaitFall1BL     ; | |   |
        NOP                     ; | |   |
        DECFSZ  DelayReg?L,f    ; | |   if (timeout)
        SKIP                    ; | |   |
        DECFSZ  DelayReg?H,f    ; | |   |
        SKIP                    ; | |   |
        GOTO    INT_ErrorN      ; | |   { goto INT_ErrorN }
        BTFSC   _RAW_DATA       ; | | } until (_RAW_DATA==#0)
        GOTO    WaitFall1BL     ; | | |

        CLRF    DelayReg?L      ; Clear timer
WaitFall2                       ; Time falling edge
WaitFall2A                      ; | Wait for high
WaitFall2AL                     ; | | { {
        NOP                     ;
        NOP                     ;
        INCF    DelayReg?L,f    ; | |     Increment timer
        BTFSC   DelayReg?L,7    ; | |     if timeout,
        GOTO    INT_ErrorN      ; | |     { goto INT_ErrorN }
        BTFSS   _RAW_DATA       ; | |   } until (_RAW_DATA==#1)
        GOTO    WaitFall2AL     ; | |   |
        NOP                     ; | |   |
        NOP                     ;
        NOP                     ;
        INCF    DelayReg?L,f    ; | |   Increment timer
        BTFSC   DelayReg?L,7    ; | |   if timeout,
        GOTO    INT_ErrorN      ; | |   { goto INT_ErrorN }
        BTFSS   _RAW_DATA       ; | | } until (_RAW_DATA==#1)
        GOTO    WaitFall2AL     ; | |
        NOP                     ;
WaitFall2B                      ; | Wait for low
WaitFall2BL                     ; | | { {
        NOP                     ;
        NOP                     ;
        INCF    DelayReg?L,f    ; | |     Increment timer
        BTFSC   DelayReg?L,7    ; | |     if timeout,
        GOTO    INT_ErrorN      ; | |     { goto INT_ErrorN }
        BTFSC   _RAW_DATA       ; | |   } until (_RAW_DATA==#0)
        GOTO    WaitFall2BL     ; | |   |
        NOP                     ; | |   |
        NOP                     ;
        NOP                     ;
        INCF    DelayReg?L,f    ; | |   Increment timer
        BTFSC   DelayReg?L,7    ; | |   if timeout,
        GOTO    INT_ErrorN      ; | |   { goto INT_ErrorN }
        BTFSC   _RAW_DATA       ; | | } until (_RAW_DATA==#0)
        GOTO    WaitFall2BL     ; | | |
        ; DelayReg?L*8Ti = period of signal
        ; period of _RAW_DATA on FSK = Tcy*10 = Ti*160
        ; DelayReg?L = 20 if FSK present
```



```
                                ; if period does not match FSK, goto INT_ErrorN
        MOVF    DelayReg?L,W    ; | if (DelayReg?L<14)
        ADDLW   low(0-d'14')    ; | |
        BTFSS   _CARRY          ; | |
        GOTO    INT_ErrorN      ; | { goto INT_ErrorN }
        ADDLW   low(d'14'-d'22'); | if (DelayReg?L>=22)
        BTFSC   _CARRY          ; | |
        GOTO    INT_ErrorN      ; | { goto INT_ErrorN }

        MOVLW   d'7'            ; CycleCtr > 13*128=1664
        MOVWF   CycleCtr?H      ; |
        MOVLW   d'164'          ; |
        MOVWF   CycleCtr?L      ; |
TestGotLo
        DECFSZ  CycleCtr?L,f
        SKIP
        DECFSZ  CycleCtr?H,f
        SKIP
        GOTO    INT_ErrorN
        MOVLW   0x20
        MOVWF   COUNT1
        BTFSS   _RAW_DATA
        GOTO    TestGotHi
TestGotLoLoop
        BTFSS   _RAW_DATA
        GOTO    TestGotHi
        DECFSZ  COUNT1,1
        GOTO    TestGotLoLoop
        GOTO    MChip_Prog

TestGotHi
        MOVLW   0x20
        MOVWF   COUNT1
        BTFSC   _RAW_DATA
        GOTO    TestGotLo
TestGotHiLoop
        BTFSC   _RAW_DATA
        GOTO    TestGotLo
        DECFSZ  COUNT1,1
        GOTO    TestGotHiLoop
;END TEST FOR NO MODULATION

MChip_Prog
        BCF     _TMR2ON
        CALL    TWC

    CLRF    DATA0
    CLRF    DATA1
    CLRF    DATA2
    CLRF    DATA3
    CLRF    DATA4
    CLRF    DATA5
    CLRF    DATA6
    CLRF    DATA7
    CLRF    DATA8
    CLRF    DATA9
    CLRF    DATAA
    CLRF    DATAB
    CLRF    DATAC
    CLRF    DATAD
    CLRF    DATAE
    CLRF    DATAF

    MOVLW   'Y'             ; RxByte='Y'
    MOVWF   RxByte          ; |
```

```
    MOVLW   DATAF           ; FSR=#DATAF
    MOVWF   FSR             ; |
    MOVLW   h'20'           ; ByteCtr=#h'20'
    MOVWF   ByteCtr         ; |
RS_ByteLoop                   ; {
    CALL    RS232On         ;   delay
    MOVF    RxByte,W        ;   Transmit RxByte on RS-232
    CALL    RS232TxW        ;   |
    CALL    RS232Rx         ;   Read RS-232 byte into RxByte
    BTFSC   _CARRY          ;   |(if timeout, goto INT_END)
    GOTO    INT_END         ;   |
    MOVF    RxByte,W        ;   BO3=RxByte
    MOVWF   BO3             ;   |
    MOVLW   h'30'           ;   if (BO3<#h'30')
    SUBWF   BO3,W           ;   |
    BTFSS   _CARRY          ;   |
    GOTO    CheckRxByte     ;   { goto CheckRxByte }
    MOVWF   BO3             ;   BO3=BO3-#h'30'
    MOVLW   h'3A'-h'30'     ;   if (BO3>=#h'3A'-#h'30')
    SUBWF   BO3,W           ;   |
    BTFSS   _CARRY          ;   |
    GOTO    RSDataJ1        ;   {
    MOVWF   BO3             ;     BO3=BO3-#h'3A'+#h'30'
    MOVLW   h'41'-h'3A'     ;     if (BO3<#h'41'-#h'3A')
    SUBWF   BO3,W           ;     |
    BTFSS   _CARRY          ;     |
    GOTO    CheckRxByte     ;     { goto CheckRxByte }
    MOVWF   BO3             ;     BO3=BO3-#h'41'+#h'3A'
    MOVLW   h'47'-h'41'     ;     if (BO3>=#h'47'-#h'41')
    SUBWF   BO3,W           ;     |
    BTFSC   _CARRY          ;     |
    GOTO    CheckRxByte     ;     { goto CheckRxByte }
    MOVLW   h'0A'           ;     BO3=BO3+#h'0A'
    ADDWF   BO3,f           ;     |
RSDataJ1                      ;   }
    SWAPF   BO3,W           ;   W = { BO3 swapped  if ByteCtr,0==#0
    BTFSC   ByteCtr,0       ;   |   { BO3          if ByteCtr,0==#1
    MOVF    BO3,W           ;   |
    IORWF   INDF,f          ;   INDF=INDF OR W
    BTFSC   ByteCtr,0       ;   if (ByteCtr,0==#1)
    DECF    FSR,f           ;   { FSR=FSR-#1 }
    DECFSZ  ByteCtr,f       ;   DEC ByteCtr
    GOTO    RS_ByteLoop     ; } until (ByteCtr==#0)

    CALL    RS232On         ; delay
    MOVF    RxByte,W        ; Transmit RxByte on RS-232
    CALL    RS232TxW        ; |
    CALL    RS232Rx         ; Read RS-232 byte into RxByte
    BTFSC   _CARRY          ; |( if timeout, goto INT_END)
    GOTO    INT_END         ; |
    MOVF    RxByte,W        ; if (RxByte!=#'V')
    XORLW   'V'             ; |
    BTFSS   _ZERO           ; |
    GOTO    CheckRxByte     ; { goto CheckRxByte }


; ********

Top BCF PORTB,3 ; SET FOR LOW VOLTAGE
    CALLDELAY  ; CALL A SMALL DELAY

GAP ; THIS IS THE ROUTINE THAT SETS THE GAP

        BCF     PORTB,3
```



```
        CALL    DELAY

    BSF T2CON,2 ; TURN ON THE COIL

    MOVLW0x32  ; MOVE 32 HEX TO W, NUMBER CYCLES BEFORE A GAP
    MOVWFCOUNT1 ; MOVW W INTO COUNT1
LOOP1DECFSZCOUNT1,1; DECREMENT COUNT 1 UNTIL IT IS ZERO
    GOTOLOOP1

    BCF T2CON,2 ; TURN OFF THE COIL

    MOVLW0x40  ; MOVE 10 HEX TO W, DURATION OF GAP
    MOVWFCOUNT1 ; MOVW W INTO COUNT1
LOOP2DECFSZCOUNT1,1; DECREMENT COUNT 1 UNTIL IT IS ZERO
    GOTOLOOP2

    BSF T2CON,2 ; TURN THE COIL BACK ON

        MOVLWd'8'; MOVE 5 INTO THE W REGISTER
    MOVWFOVERPRO; THIS IS THE NUMBER OF OVERPROGRAMMING

        CALL    TWC             ; CALL A DELAY FOR AMP TO SETTLE
        CALL    TWC
        CALL    TWC
        CALL    TWC
        CALL    TWC

MODING  CALL    TESTMOD
        CALL    PROGRAM

        MOVLW   0x60
        MOVWF   COUNT1
BIGDLY  CALL    TWC             ; CALL A DELAY TO ALLOW THE AMP TO SETTLE
        DECFSZ  COUNT1,f
        GOTO    BIGDLY

        DECFSZ  OVERPRO,1       ; DECREMENT THE OVERPROGRAMMING NUMBER
    GOTOMODING ; GOTO LOOK FOR THE MODULATION TO STOP
        GOTOVERIFY

;***************************************

VERIFY
        CALL    TESTMOD         ; Wait for modulation to stop
                                ;% 167Ti of constant _RAW_DATA

StartWatch                      ; Wait >~Ttag (for mod to start again)

        MOVLW   h'00'           ; Delay >~262144Ti
        MOVWF   DelayReg?H      ; |
VerifyD1a                       ; |
        MOVLW   h'FF'           ; | delay 1021Ti
        MOVWF   DelayReg?L      ; | |
VerifyD1b                       ; | |
        CLRWDT                  ; | |
        DECFSZ  DelayReg?L,f    ; | |
        GOTO    VerifyD1b       ; | |
        DECFSZ  DelayReg?H,f    ; |
        GOTO    VerifyD1a       ; |
StopWatch                       ; |

        CLRF    BitCtr          ; BitCtr=#128
        BSF     BitCtr,7        ; |

VerifyL1                        ; {
```

```
;% reftime-1345
        CLRF    CycleCtr?L
;% reftime-1344
;% reftime-3-10*6-183*7
        MOVLW   d'10'           ;   set NoiseTimeout
        MOVWF   NoiseTimeout    ;   |
;% reftime-1-10*6-183*7
;% reftime-1-NTO*6-183*7
        MOVLW   d'183'          ;   set SampTimeout to 80Tcy
        MOVWF   SampTimeout     ;   |
;% reftime+1-NTO*6-183*7
;% reftime+1-NTO*6-STO*7

        BTFSC   _RAW_DATA
        GOTO    VerS1
        NOP

VerS0
;% reftime+4-NTO*6-STO*7
        DECFSZ  NoiseTimeout,f
        SKIP
        GOTO    VerFail
        BTFSC   _RAW_DATA
        GOTO    VerS1
VerGot0
;% reftime+3-NTO*6-STO*7
VerGot0a
;% reftime+3-NTO*6-STO*7
        CLRWDT
        DECFSZ  SampTimeout,f
        SKIP
        GOTO    SampleDone
        BTFSS   _RAW_DATA
        GOTO    VerGot0
        NOP
VerGot0b
;% reftime+3-NTO*6-STO*7
        CLRWDT
        DECFSZ  SampTimeout,f
        SKIP
        GOTO    SampleDone
        BTFSS   _RAW_DATA
        GOTO    VerGot0
        NOP
VerGotRise
;% reftime+3-NTO*6-STO*7
        CLRWDT
        DECFSZ  SampTimeout,f
        SKIP
        GOTO    SampleDone
        INCF    CycleCtr?L,f
        GOTO    VerGot1

VerS1
;% reftime+4-NTO*6-STO*7
        DECFSZ  NoiseTimeout,f
        SKIP
        GOTO    VerFail
        BTFSS   _RAW_DATA
        GOTO    VerS0
VerGot1
;% reftime+3-NTO*6-STO*7
VerGot1a
;% reftime+3-NTO*6-STO*7
        CLRWDT
```

```
        DECFSZ  SampTimeout,f
        SKIP
        GOTO    SampleDone
        BTFSC   _RAW_DATA
        GOTO    VerGot1
        NOP
VerGot1b
;% reftime+3-NTO*6-STO*7
        CLRWDT
        DECFSZ  SampTimeout,f
        SKIP
        GOTO    SampleDone
        BTFSC   _RAW_DATA
        GOTO    VerGot1
        NOP
VerGotFall
;% reftime+3-NTO*6-STO*7
        CLRWDT
        DECFSZ  SampTimeout,f
        SKIP
        GOTO    SampleDone
        INCF    CycleCtr?L,f
        GOTO    VerGot0

SampleDone
;% reftime+1-NTO*6-STO*7
;& STO=0
;% reftime+1-NTO*6
NoiseMargin
;% reftime+1-NTO*6
        NOP
        NOP
        NOP
        DECFSZ  NoiseTimeout,f
        GOTO    NoiseMargin
;% reftime+0-NTO*6
;% NTO=0
;% reftime+0

        BTFSC   DATAF,7
        GOTO    Verify1
        NOP

Verify0
;% 3 from ref time
; if ‘0’ bit, _DATA_IN cycles 10 times in 80 Tcy
; CycleCtr?L should be 20
        MOVF    CycleCtr?L,W
        ADDLW   low(0-d’18’)
        BTFSS   _CARRY
        GOTO    INT_Failure
        ADDLW   low(d’18’-d’22’)
        BTFSS   _CARRY
        GOTO    Bit_Verified
        GOTO    INT_Failure

Verify1
;% 3 from ref time
; if ‘1’ bit, _DATA_IN cycles 8 times in 80Tcy
; CycleCtr?L should be 16
        MOVF    CycleCtr?L,W
        ADDLW   low(0-d’14’)
        BTFSS   _CARRY
        GOTO    INT_Failure
        ADDLW   low(d’14’-d’18’)
```

```
        BTFSS   _CARRY
        GOTO    Bit_Verified
        GOTO    INT_Failure

Bit_Verified
;% 11 from ref time
        BCF     _CARRY
        BTFSC   DATAF,7
        BSF     _CARRY
        RLF     DATA0,f
        RLF     DATA1,f
        RLF     DATA2,f
        RLF     DATA3,f
        RLF     DATA4,f
        RLF     DATA5,f
        RLF     DATA6,f
        RLF     DATA7,f
        RLF     DATA8,f
        RLF     DATA9,f
        RLF     DATAA,f
        RLF     DATAB,f
        RLF     DATAC,f
        RLF     DATAD,f
        RLF     DATAE,f
        RLF     DATAF,f

;% 30 from ref time

        MOVLW   d'167'          ; Delay 670Ti
        MOVWF   DelayReg?L      ; |
        NOP                     ; |
VerDelay                        ; |
        CLRWDT                  ; |
        DECFSZ  DelayReg?L,f    ; |
        GOTO    VerDelay        ; |

;% 700 from ref time
;% (ref times 128*16Ti apart = 2048Ti apart)
;% -1348 from ref time
        DECFSZ  BitCtr,f        ;   DEC BitCtr
        GOTO    VerifyL1        ; } until (BitCtr==#0)

;****************************************

INT_Success
        CALL    RS232On
    MOVLW   'y'
    CALL    RS232TxW
    GOTO    BigLoop1

VerFail
INT_Failure

        CALL    RS232On
        MOVLW   'n'
        CALL    RS232TxW
        GOTO    BigLoop1

INT_END ; RS-232 TIMEOUT
    NOP
    GOTO    BigLoop1

INT_ErrorN
    CALL    RS232On
    MOVLW   'N'
```

```
    CALL    RS232TxW
    GOTO    BigLoop1

DELAYMOVLW0x05
    MOVWFDELAY1
HOLD4DECFSZDELAY1,1
    GOTOHOLD4
    RETLW0

; TWC lasts
TWC MOVLW0xB0  ; WRITE CYCLE TIMER SUBROUTINE
    MOVWFDELAY1
HOLD1MOVLW0x02
    MOVWFDELAY2
HOLD2DECFSZDELAY2,1
    GOTOHOLD2
    DECFSZDELAY1,1
    GOTOHOLD1
        RETLW0

BUFFERMOVLW0x58
    MOVWFDELAY1
HOLD3DECFSZDELAY1,1
    GOTOHOLD3
        NOP
        NOP
        RETLW0

TESTMOD; THIS ROUTINE TESTS THE RAW DATA LINE TO SEE IF THE
    ; PART IS MODULATING OR NOT

; This routine returns when _RAW_DATA stays constant for some time
; some time = 7Ti+32*5Ti = 167Ti = 10.4375Tcy

        MOVLW   d'7'            ; CycleCtr2 > 13*128=1664
        MOVWF   CycleCtr2?H     ; |
        MOVLW   d'164'          ; |
        MOVWF   CycleCtr2?L     ; |

TestModLo
        DECFSZ  CycleCtr2?L,f
        SKIP
        DECFSZ  CycleCtr2?H,f
        SKIP
        GOTO    INT_Failure
        MOVLW   0x20
        MOVWF   COUNT1
        BTFSS   _RAW_DATA
        GOTO    TestModHi
TestModLoLoop
        BTFSS   _RAW_DATA
        GOTO    TestModHi
        DECFSZ  COUNT1,1
        GOTO    TestModLoLoop
        RETLW   0
TestModHi
        MOVLW   0x20
        MOVWF   COUNT1
        BTFSC   _RAW_DATA
        GOTO    TestModLo
TestModHiLoop
        BTFSC   _RAW_DATA
        GOTO    TestModLo
        DECFSZ  COUNT1,1
        GOTO    TestModHiLoop
```

```
;END TEST FOR NO MODULATION
        RETLW   0

PROGRAM BCFPORTB,3; CLEAR THE HIGH VOLTAGE

        MOVLW0x07; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITEFBTFSSDATAF,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATAF,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATAF,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITEF; GOTO WRITEF IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

        MOVLW0x07; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITEEBTFSSDATAE,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATAE,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATAE,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITEE; GOTO WRITEE IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITEDBTFSSDATAD,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATAD,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATAD,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITED; GOTO WRITEF IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
```

```
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITECBTFSSDATAC,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATAC,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATAC,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITEC; GOTO WRITEC IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITEBBTFSSDATAB,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATAB,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATAB,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITEB; GOTO WRITEB IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITEABTFSSDATAA,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATAA,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATAA,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITEA; GOTO WRITEA IF BIT IS NOT EQUAL TO ZERO
        NOP
```

```
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE9BTFSSDATA9,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA9,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA9,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE9; GOTO WRITE9 IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE8BTFSSDATA8,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA8,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA8,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE8; GOTO WRITE8 IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

        MOVLW0x07; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE7BTFSSDATA7,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA7,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA7,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE7; GOTO WRITE7 IF BIT IS NOT EQUAL TO ZERO
```

```
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE6BTFSSDATA6,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA6,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA6,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE6; GOTO WRITE6 IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE5BTFSSDATA5,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA5,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA5,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE5; GOTO WRITE5 IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE4BTFSSDATA4,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA4,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA4,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
```

```
        GOTOWRITE4; GOTO WRITE4 IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE3BTFSSDATA3,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA3,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA3,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE3; GOTO WRITE3 IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE2BTFSSDATA2,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA2,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA2,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE2; GOTO WRITE2 IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE1BTFSSDATA1,7        ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA1,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA1,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
```

```
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE1; GOTO WRITEF IF BIT IS NOT EQUAL TO ZERO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

    MOVLW0x07  ; MOVW 7 HEX INTO W
    MOVWFBIT   ; MOVE THIS INTO THE BIT COUNTER
WRITE0BTFSSDATA0,7 ; TEST MOST BYTE
    BSF PORTB,3 ; SET THE HIGH VOLTAGE
    CALLTWC    ; CALL THE WRITE CYCLE TIMER
    BCF     STATUS,C        ; CLEAR THE CARRY BIT
        BTFSC   DATA0,7         ; TEST THE MSB
        BSF     STATUS,C        ; SET THE CARRY BIT
        RLF     DATA0,1         ; ROTATE DATAF
        BCFPORTB,3; CLEAR THE HIGH VOLTAGE
        CALL BUFFER; CALL THE BUFFER TIMER
    DECF    BIT,1; DECREMENT BIT, SKIP IF ZERO
    BTFSS   BIT,7           ; SKIP IF SET
        GOTOWRITE0; GOTO WRITE0 IF BIT IS NOT EQUAL TO ZERO
        RETLW   0

Delay12
        NOP
Delay11
        GOTO    Delay09
Delay09
        GOTO    Delay07

RS232On                         ;[1] Initialise RS-232
    BCF     _TMR2ON         ; | Turn coil off
    CALL    RS232StopBit    ; | Transmit stop bits
    CALL    RS232StopBit    ; | |
    CALL    RS232StopBit    ; | |
    CALL    RS232StopBit    ; | |
    CALL    RS232StopBit    ; | |
    CALL    RS232StopBit    ; | |
    CALL    RS232StopBit    ; | |
    CALL    RS232StopBit    ; | |
    CALL    RS232StopBit    ; | |
    CALL    RS232StopBit    ; | |
    RETLW   h’00’           ; | return

RS232WaitForever                ;[1] ~9600 baud
BigWaitL1                       ; | {
    CLRWDT                  ;!|
    BTFSS   _RS232IN        ; |   if (_RS232IN==#0)
    GOTO    RS232RxL1Done   ; |   { goto RS232RxL1Done }
    NOP                     ;!|
    GOTO    BigWaitL1       ; | } until (0)

RS232Rx                         ;[1] ~9600 baud
    MOVLW   d’16’           ; | Set timeout of ~2.9s
    MOVWF   TimerHi         ; | |
    CLRF    TimerMid        ; | |
    CLRF    TimerLo         ; | |
RS232RxL1                       ; | {
    CLRWDT                  ;!|
    BTFSS   _RS232IN        ; |   if (_RS232IN==#0)
    GOTO    RS232RxL1Done   ; |   { goto RS232RxL1Done }
    DECFSZ  TimerLo,f       ; | }
```

```
    GOTO    RS232RxL1       ; | |
    DECFSZ  TimerMid,f      ; | |
    GOTO    RS232RxL1       ; | |
    DECFSZ  TimerHi,f       ; | |
    GOTO    RS232RxL1       ; | |
    BSF     _CARRY          ; | return with error
    RETLW   h’00’           ; | |
               ; |
RS232RxL1Done                   ; |% 3 to (+6, +8, +10) - say 10us
               ; |% 10-104=-94
    MOVLW   d’90’           ;!|
    CALL    DelayW12        ;!|% 9
    CLRF    BitCtr          ; | BitCtr=#8
    BSF     BitCtr,3        ; | |
RS232RxLoop                     ; | {% 11
    MOVLW   d’181’          ;!|
    CALL    DelayW12        ;!|  % 205
    CLRF    BO3             ; |   BO3,1=_RS232IN
    BTFSC   _RS232IN        ; |   |
    INCF    BO3,f           ; |   |% 208
    BTFSC   _RS232IN        ; |   |% 1
    INCF    BO3,f           ; |   |
    BTFSC   _RS232IN        ; |   |
    INCF    BO3,f           ; |   |% 4
    RRF     RxByte,f        ; |   RR RxByte
    BCF     RxByte,7        ; |   RxByte,7=BO3,1
    BTFSC   BO3,1           ; |   |
    BSF     RxByte,7        ; |   |% 8
    DECFSZ  BitCtr,f        ; |   DEC BitCtr
    GOTO    RS232RxLoop     ; | } until (BitCtr==#0)
    BCF     _CARRY          ; | return with no error
    RETLW   h’00’           ; | |

RS232TxW                        ;[1] Transmit W on RS232 at ~9600 baud
    MOVWF   TxByte          ; | TxByte=W
    CALL    RS232StopBit    ; | stop bit
    CLRF    BitCtr          ; | BitCtr=#8
    BSF     BitCtr,3        ; | |
    BCF     _RS232OUT       ; | Start bit
    MOVLW   d’191’          ; | |
    CALL    DelayW12        ; | |
RS232TxLoop                     ; | {% 205
    BTFSS   TxByte,0        ; |   _RS232OUT=TxByte,0
    BCF     _RS232OUT       ; |   |% 207
    BTFSC   TxByte,0        ; |   |% 208
    BSF     _RS232OUT       ; |   |% 1
    RRF     TxByte,f        ; |   RR TxByte
    MOVLW   d’187’          ;!|
    CALL    DelayW12        ;!|  % 202
    DECFSZ  BitCtr,f        ; |   DEC BitCtr
    GOTO    RS232TxLoop     ; | } until (BitCtr==#0)
    GOTO    RS232TxJ1       ; |
RS232TxJ1                       ; |
    NOP                     ; |% 207
    BSF     _RS232OUT       ; | Stop bit
    RETLW   h’00’           ; | return

        end
```

NOTES:

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

Boston
Westford, MA
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

San Jose
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Qingdao
Tel: 86-532-502-7355
Fax: 86-532-502-7205

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

Japan - Kanagawa
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Taiwan - Hsinchu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

EUROPE

Austria - Weis
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark - Ballerup
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Massy
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Ismaning
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

England - Berkshire
Tel: 44-118-921-5869
Fax: 44-118-921-5820

10/20/04