

CAN Bootloader

1. Introduction	2
2. Features.....	2
3. List of Code Modules	3
4. Using the Library Module in a Project.....	3
5. Shared Parameters.....	4
<i>Shared Data Bytes</i>	<i>4</i>
<i>Shared Functions</i>	<i>4</i>
<i>Shared Macros.....</i>	<i>4</i>
6. Assembly Functions Details	5
7. Assembly Macros Details	6
8. Error and Status Flags	7

1. Introduction

This is the CAN bootloader module for PIC18FXX8 family of microcontrollers with integrated CAN module. The included code provides low-level In-Application Programming via CAN. This particular module is not your typical Microchip Application Maestro Module. The CAN bootloader is designed to be a separate piece of firmware that is mutually exclusive of any application firmware; thus this module will typically be the only module in a project. Refer to AN247 for details about design, implementation, and usage.

2. Module Features

The CAN bootloader has the following features:

- Small code size, fits within the PIC18FXX8 512 byte boot block.
- Configurable for source or self program verification.
- Flexible compile-time options.
- Allows fast In-Application Programming.

3. List of Component Modules

<code>CANBoot.lkr.txt</code>	This is the linker script file used to build the bootloader. Rename this file to <code>CANBoot.lkr</code> , and include it into your project if you want to use this linker script.
<code>CANBoot.asm</code>	This is the core code implementing the bootloader.
<code>P18Cxxx.inc</code>	General-purpose processor definition file for PIC18 family provided. It selects processor dependant '.inc' file for library module according to processor selection in development mode.

4. Using the Library Module in a Project

Although you can include this with application code, it is designed to be completely separate from application code. Thus you build this and load a PIC18F CAN device with this firmware. Then you create your application in a separate project with the appropriate changes to the applications linker script file to accommodate the CAN bootloader.

Please follow below steps to use this library module in your project.

1. Use the Application Maestro to configure your code as required.
2. At the Generate Files step, save the output to the directory where your code project resides.
3. Launch MPLAB, and open the project's workspace.
4. Verify that the Microchip language tool suite is selected (*Project>Select Language Toolsuite*).
5. In the Workspace view, right-click on the "Source Files" node. Select the "Add Files" option. Select `CANBoot.asm` and click **OK**.
6. Now right-click on the "Linker Scripts" node and select "Add Files". Add the appropriate linker file (`.lkr`) for the project's target CAN microcontroller.

5. List of Shared Parameters

There are no shared parameters. The CAN bootloader is a resident piece of firmware that is entirely independent of any application.

Shared Data Bytes

NA.

Shared Functions

NA.

Shared Macros

NA.

6. Functions

There are no functions to call. The CAN bootloader is a resident piece of firmware that is entirely independent of any application.

7. Macros

There are no macros for this module.

8. Error and Status Flags

There are no public flags.