## Introduction

Before assembling your firmware, you need to make a number of changes to the *.asm file so that it configures your target microcontroller correctly.

## Device Support

The first thing you need to do is see if your target microcontroller is supported. To do this, open "devices.inc". You will see a number of entries that look something like this:

```
#ifdef      __18F452
   #include                        p18F452.inc
   #define DEVICE_ID               0x0420
   #define DEVICE_SERIES           DEVICE_IS_18F
   #define DEVICE_ROM              0x08000
   #define DEVICE_BLOCK_ERASE      64
   #define DEVICE_BLOCK_WRITE      8
   #define DEVICE_EEPROM           256
#endif
```

If your device is not included, you will need to add it. For this example, we will add support for the 18F25K20, like this:

```
#ifdef      __18F25K20
   #include                        p18F25K20.inc
   .....
#endif
```

notice (a) the definition name of the device, with a double underscore preceding it and (b) the assembler include file for the device. In the above example, it's 'p18F25K20.inc'. Next, we need to add a number of additional definitions. These are used to (a) guide the assembly process and (b) pass important information to the loader application that resides on the PC.

- **#define** DEVICE_ID - This is not actually used by the firmware, it's just passed to the PC host application and is used to lookup a textual description of the device when a user requests information about the target microcontroller. You can set this value to zero if you like, but it's really best to give it the correct ID. You can usually find this information in the MPLAB 'device' subfolder. For example, "C:\Program Files\Microchip\MPLAB IDE\Device\PIC18F25K20.dev".

- **#define** DEVICE_SERIES - can be either DEVICE_IS_16F or DEVICE_IS_18F. This tells the host PC application the programming algorithm to use to program the target device.

- **#define** DEVICE_ROM - the size of device ROM. For 16 series, the value should be given in WORDS. For 18 series, the value is in BYTES. You can obtain this information from the device datasheet.

- **#define** DEVICE_BLOCK_ERASE - the size of the device erase block. This value should always be given in BYTES, even for 16 series devices. You can obtain this information from the device datasheet.

- **#define** DEVICE_AUTO_ERASE - this value is only used for 16 series only. Many 16 series devices automatically erase ROM before performing a write operation. If this is the case, you need to set to "1" (without the quotes!). If the device requires an explicit erase (for example, 16F88) then set this value to "0". You can obtain this information from the device datasheet.

- **#define** DEVICE_BLOCK_WRITE - the size of the device write block. This value should always be given in BYTES, even for 16 series devices. You can obtain this information from the device datasheet.

- **#define** DEVICE_EEPROM - the size of onboard EEPROM. This values is given in BYTES. If the device does not have EEPROM, then a value of zero should be used. You can obtain this information from the device datasheet.

The final entry will now look like this:

```
#ifdef      __18F25K20
   #include                        p18F25K20.inc
   #define DEVICE_ID               0x2060
   #define DEVICE_SERIES           DEVICE_IS_18F
   #define DEVICE_ROM              0x08000
   #define DEVICE_BLOCK_ERASE      64
   #define DEVICE_BLOCK_WRITE      32
   #define DEVICE_EEPROM           256
#endif
```

**Configuring the Main Loader File**

After you have added you device to devices.inc (if required) you now need to configure the main loader *.asm file. The area you need to edit is at the top of the file, and has the start and end of the block marked like this:

```
;=================================================================
; USER CONFIGURATION SECTION
;=================================================================
......
;=================================================================
; END USER CONFIGURATION SECTION
;=================================================================
```

The most important value are

- **Processor** - give the name of the processor you want to target
- **#define** DEVICE_CLOCK - the clock speed of the target device
- **#define** BAUDRATE - the communications baudrate

Next, there is an area you can place you optional startup code. For example, some devices require their TX and RX pins be explicitly set to DIGITAL. You can do that in

the 'UserConfig' macro:

```
UserConfig macro
   ; place code in here
   endm
```

Finally, you need to set you device configuration fuses. Setting device fuses is beyond the scope of this document as there is so much variation between devices. For more information on configuration fuses, you should refer to your device datasheet.