How to program the Fast Amplifiers

Ivo Carvalho,

2010

introduction

The Fast Amplifiers were programmed in Assembly using the MPLAB program supplied freely by Microchip^{tm} . the micro-controller used is the DsPIC30F2020 which belongs to the Switched Mode Power Supply (SMPS) dspic30F product family. The datasheet can be found in http://ww1.microchip.com/downloads/en/DeviceDoc/70178C.pdf

In order to program the Fast Amplifiers without damaging the system, you should verify that noting else is connected besides the corresponding top 5V switch of the FA that you're attempting to program. All the other voltages in the circuit should be 0V, because when programming the micro-controller, the outputs are in a unstable state and that could lead to destruction of the power circuit if there energy stored in it.

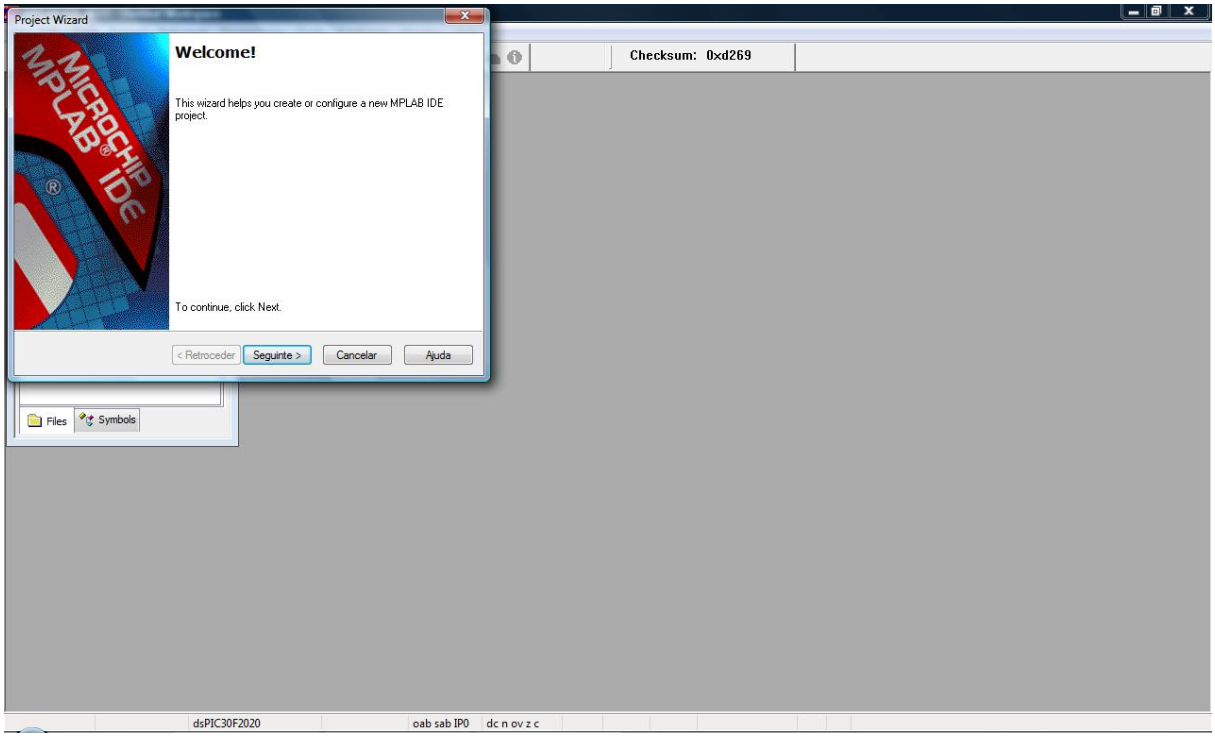
This tutorial is organized in simple steps with several key images to help in the interpretation of the steps.

1st Step: setting up the system

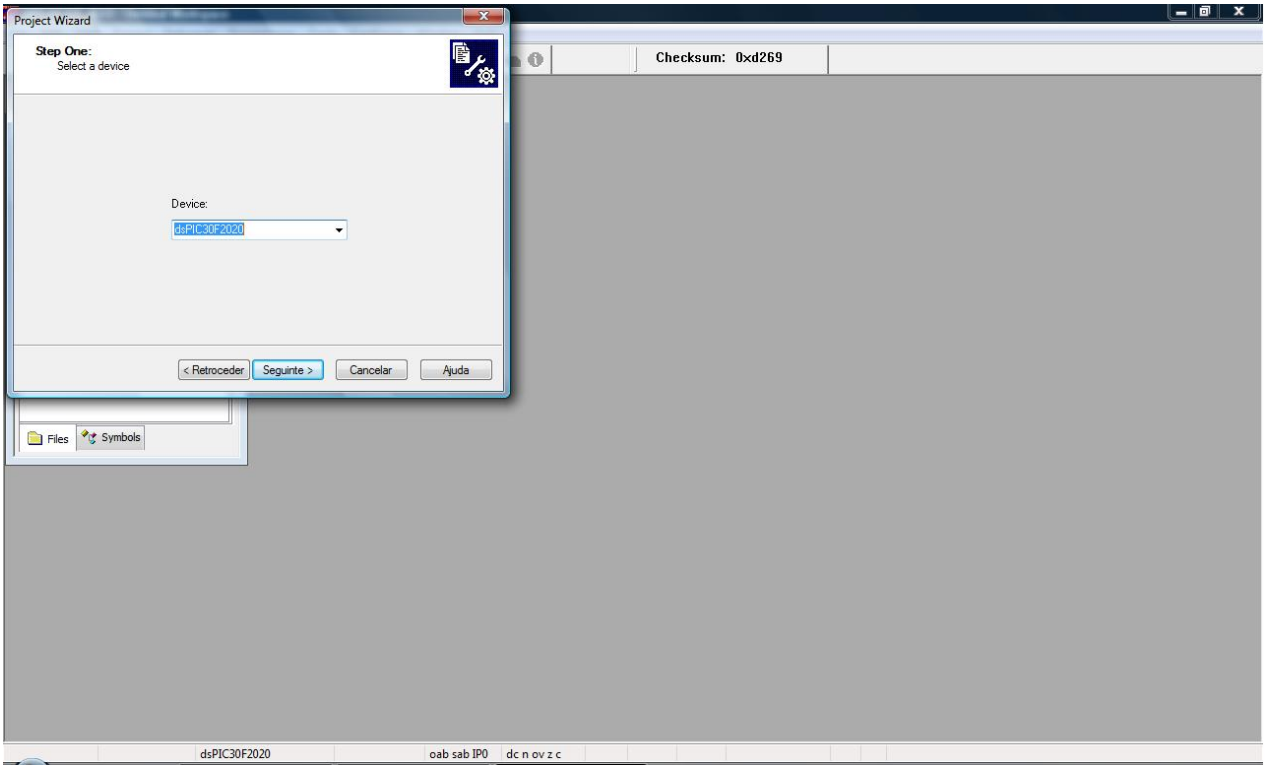
Install the most recent version of MPLAB (currently 8.53). It can be found in the following link: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002

2nd Step: project wizard

Create a new project or use an existing one with assembly and DsPIC30F2020

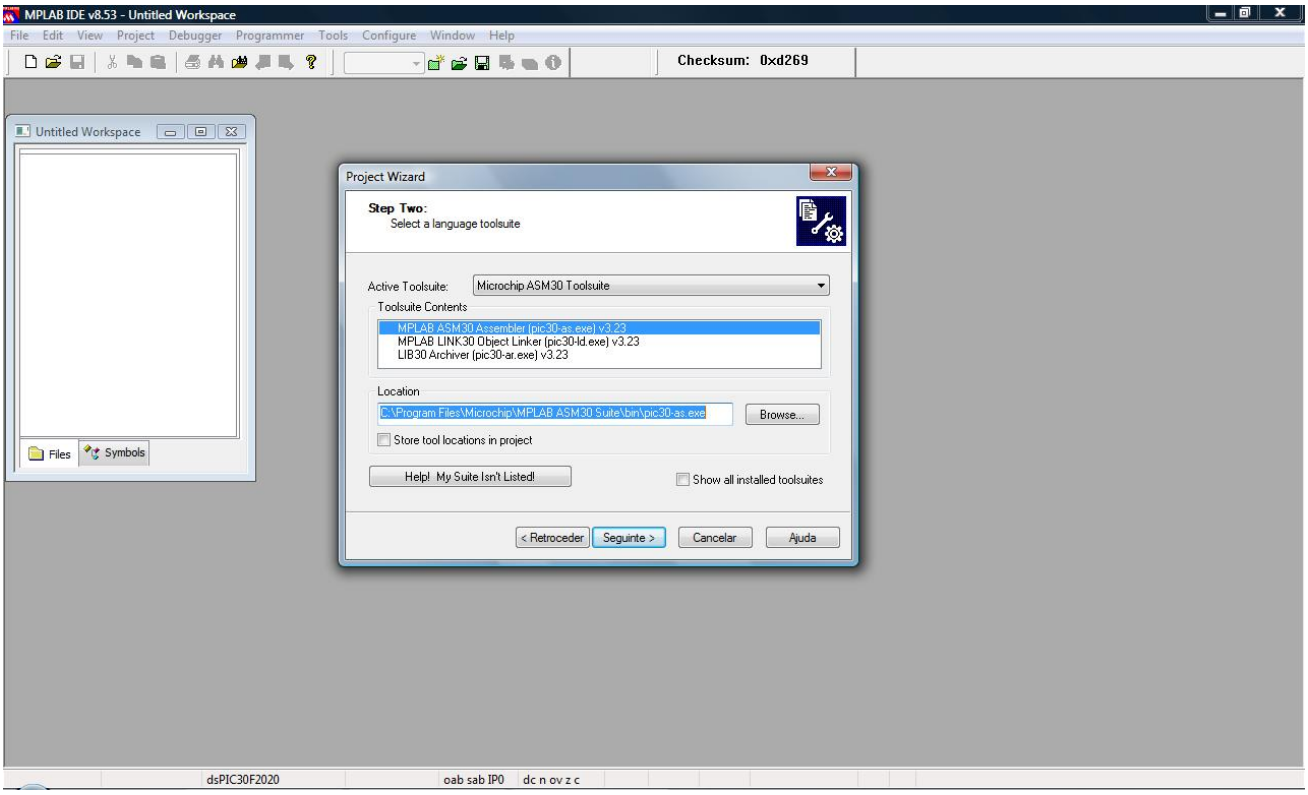


Then select the DsPIC30F2020 (this is the dspic used in the FA controllers and also on the thyristors control board.



After selecting the dspic for the project MPLAB will prompt for the tool-suit for programming. It should be selected the Microchip ASM30 Toolsuite, then in the same menu the Assembler,

linker and archiver programs should be indicated to the wizard (these .exe are found in the MPLAB installation dir, find the files as in the following figure). This files are named: pic30-as.exe , pic30-ld.exe and pic30-ar.exe .



After this you will be prompted to add files, download the source code file on CDAQ (.s extension) and then locate it on your hard drive.

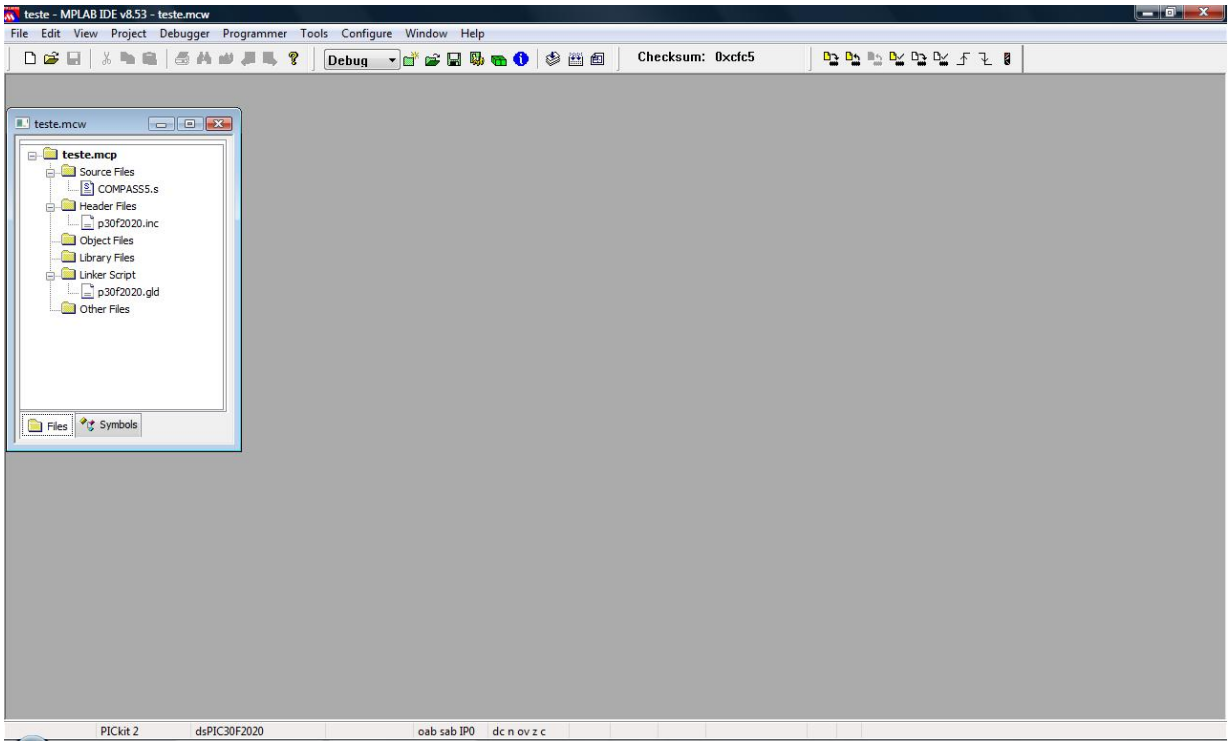
3rd Step: adding header files and linker script

After the project wizard, you should visualize your current project. Then you should the following:

- second click on the header files directory then select add files. search for the file p30f2020.inc that is inside the MPLAB installation directory, then add this file.

- second click on the Linker script directory then select add files. search for the file p30f2020.gld that is inside the MPLAB installation directory, then add this file.

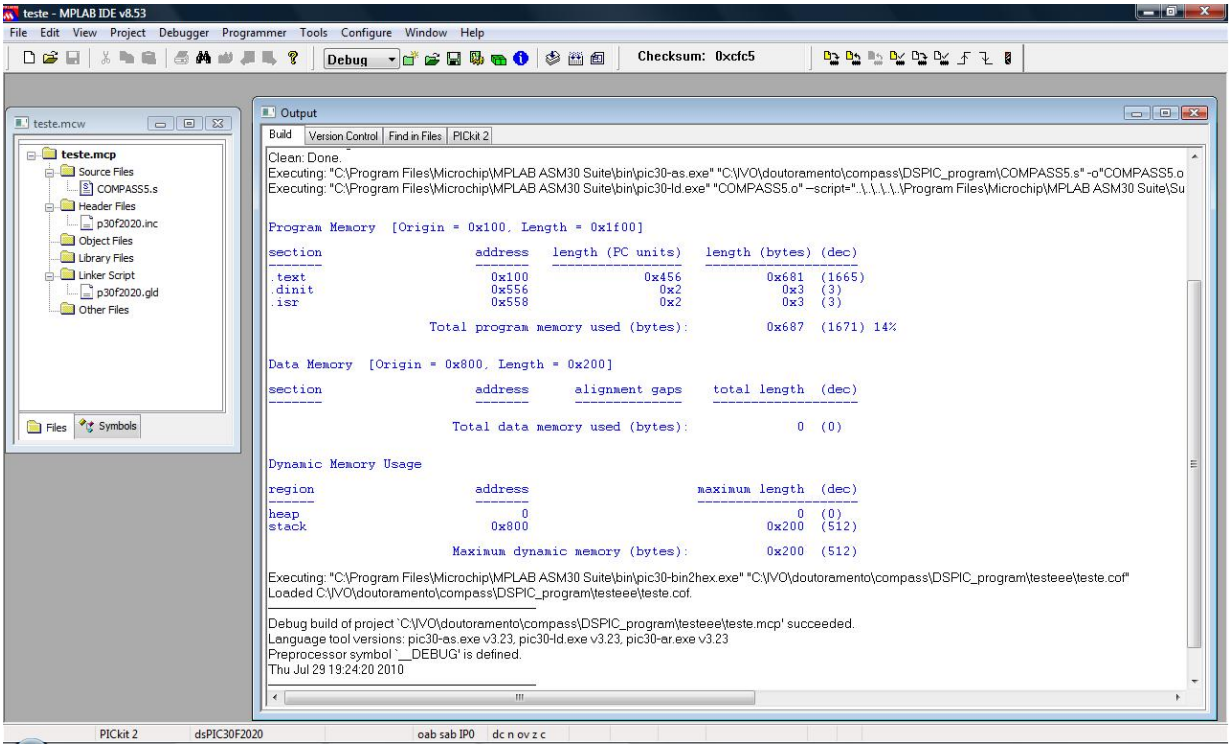
After adding this files your project should look like the following image



4th Step: compiling

After adding all of the relevant files you can start compiling the code, the “Build all” button is on the top center of the shortcut buttons or you can use the Project menu and then “Build all” or use the short cut keys “Ctrl + F10”

The compiling should look like this:



5th Step: programming the target device

First connect your programmer to the USB port and to the 6-pin RJ11 plug on the Fast amplifier that you what to program.

Then go to the Programmer menu on MPLAB and select you programmer. After doing that some information about the programmer connection should pop out in the output window, verify that everything is ok and then use the programmer menu and select “Program”. If everything is ok, your micro-controller is now programmed.

Extra chapter: How to change the operation time of the fast amplifiers

Normally the fast amplifiers are programmed for 1 second of operation after a start command, however it can be changed easily in the source code. **This is a crucial parameter of circuit protection, so please be aware of the dangers and consequences of over-passing the circuit limits.**

This reset after 1 second of operation is controlled by the DsPIC30F2020 Watch Dog Timer (WDT), when in operation this WDT is not cleared thus provoking a reset after the programmed time.

The WDT configuration is made in the beginning of the source code in the following line:

```
config __FWDT, 0b0000000000000000011011000 ;watch dog timer -> 1024 ms
```

To change the WDT you only need to modify this line of code and to reconfigure the bits of the register according to your needs.

The FWDT register was 16 bit, the first bit on the right is the Least Significant Bit (LSB) and and it is the bit number 0, so the bits are ordered from bit-15 to bit-0, left to right (the “0b” prefix is to use binary numbers afterwards)

The WDT configuration is described in the following table:

TABLE 18-7: FWDT AND FPOR BIT DESCRIPTIONS FOR dsPIC30F1010/202X

Bit Field	Register	Description
FWDTEN	FWDT	Watchdog Timer Enable bit 1 = Watchdog Timer always enabled. (LPRC oscillator cannot be disabled. Clearing the SWDTEN bit in the RCON register will have no effect.) 0 = Watchdog Timer enabled/disabled by user software (LPRC can be disabled by clearing the SWDTEN bit in the RCON register)
WWDTEN	FWDT	Watchdog Timer Window Enable bit 1 = Watchdog Timer in Non-Window mode 0 = Watchdog Timer in Window mode
WDTPRE	FWDT	Watchdog Timer Prescaler bit 1 = 1:128 0 = 1:32
WDTPOST<3:0>	FWDT	Watchdog Timer Postscaler bits 1111 = 1:32, 768 1110 = 1:16, 384 . . . 0001 = 1:2 0000 = 1:1
FPWRT<2:0>	FPOR	Power-on Reset Timer Value Select bits 111 = PWRT = 128 ms 110 = PWRT = 64 ms 101 = PWRT = 32 ms 100 = PWRT = 16 ms 011 = PWRT = 8 ms 010 = PWRT = 4 ms 001 = PWRT = 2 ms 000 = PWRT = Disabled

bit 7 -> WDT enable bit (**without 1 on this bit the micro-controller will never reset!!!!!**)
bit 6 -> window mode (please put always 1 or it might malfunction)
bit 4 -> prescaller bit (0==32; 1==128)
bit 3 to bit 0 -> postscaller bits (2^(bitvalue) example: 0100 -> 2^4 = 16)

To know the exact timing of the reset you can use the following formula:

WDT period = (1/512kHz) x pre-scale value x post-scale value x 16

Table with all combinations:

postscale	prescale = 0 (x32)	prescale = 1 (x128)
0000 (x1)	0.001	0.004
0001 (x2)	0.002	0.008
0010 (x4)	0.004	0.016
0011 (x8)	0.008	0.032
0100 (x16)	0.016	0.064
0101 (x32)	0.032	0.128
0110 (x64)	0.064	0.256
0111 (x128)	0.128	0.512
1000 (x256)	0.256	1.024
1001 (x512)	0.512	2.048
1010 (x1024)	1.024	4.096
1011 (x2048)	2.048	8.192
1100 (x4096)	4.096	16.384
1101 (x8192)	8.192	32.768
1110 (x16384)	16.384	65.536
1111 (x32768)	32.768	131.072

(time in seconds)

example:

config __FWDT, 0b00000000000000000011010110 ;~256ms (no window mode)

pre-scale -> 1== 128, post-scale -> 0110 = 2^6 = 64 --> WDT period = 1/(512 kHz) x 128 x 64 x 16 = 256ms