Embedded Systems
Appendix A2 - Tools for Clicker2 Board

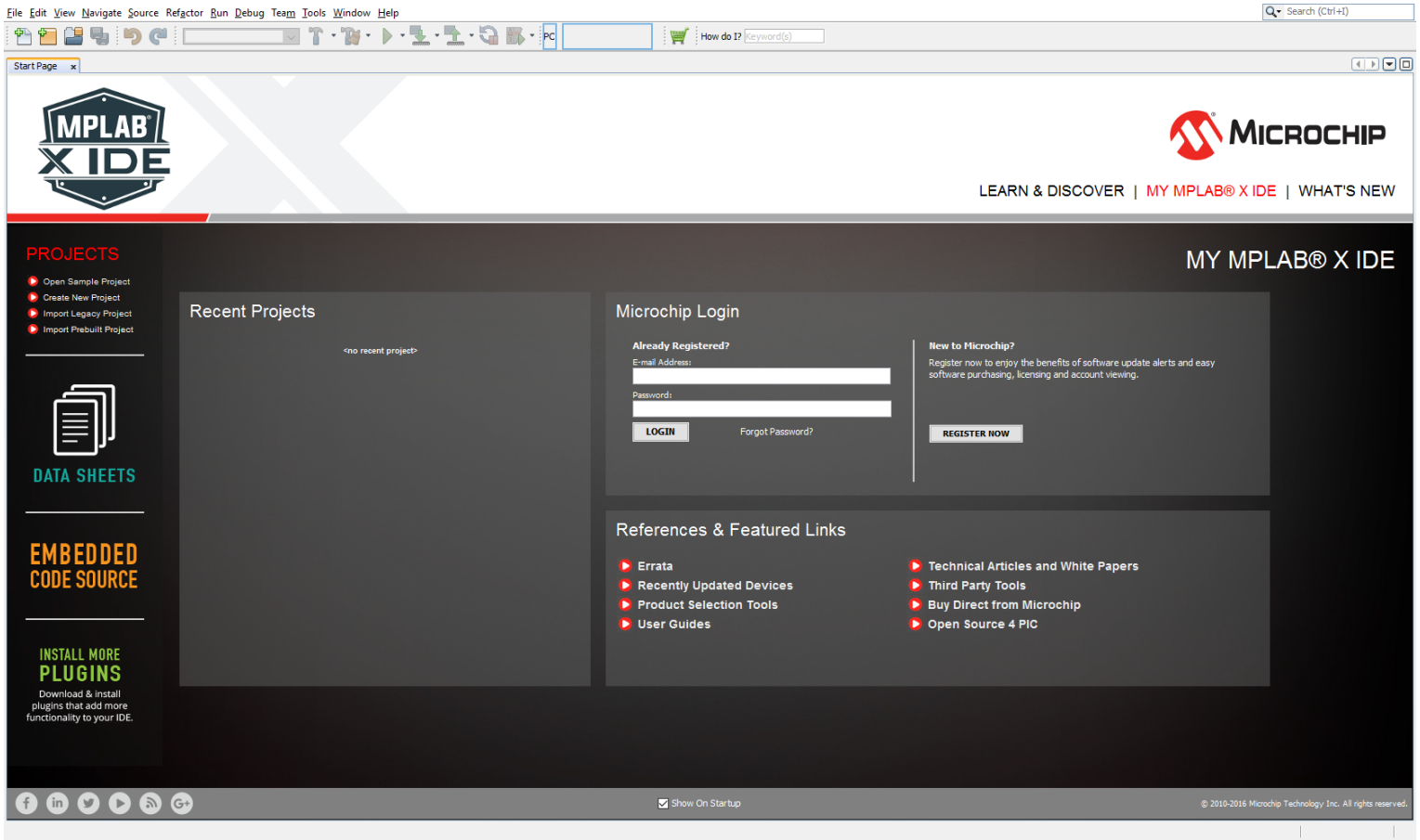
Enrico Simetti

ISME - Interuniversity Research Center on Integrated Systems for Marine Enviroment
DIBRIS - Department of Computer Science, Bioengineering, Robotics and System Engineering
University of Genova, Italy
enrico.simetti@unige.it

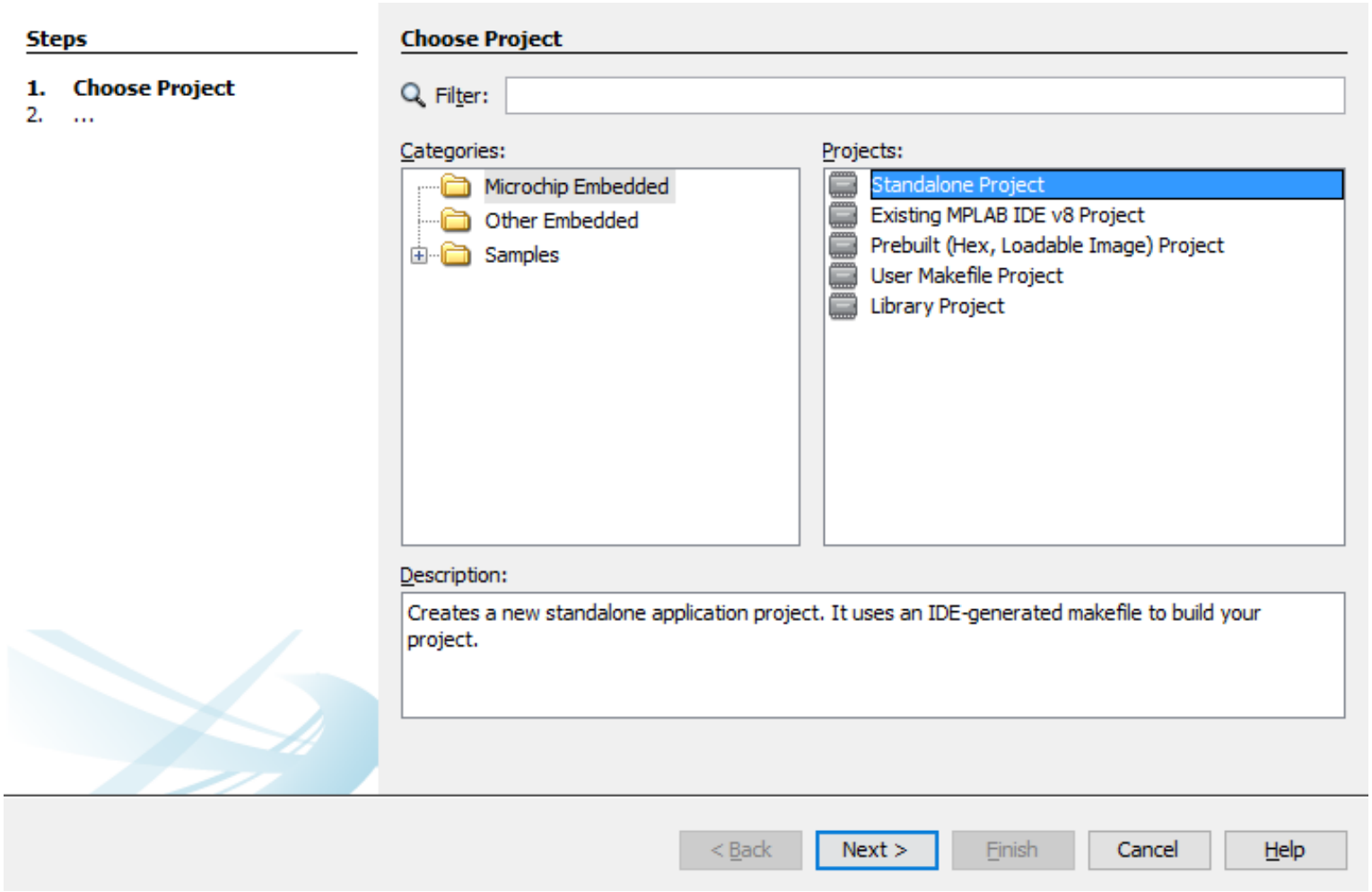
Installing the Tools

- MPLAB X IDE:
  `http://www.microchip.com/mplab/mplab-x-ide`
- XC16 Compiler:
  `http://www.microchip.com/mplab/compilers`
- MikroE mikroBootloader:
  `https://www.mikroe.com/mikrobootloader`

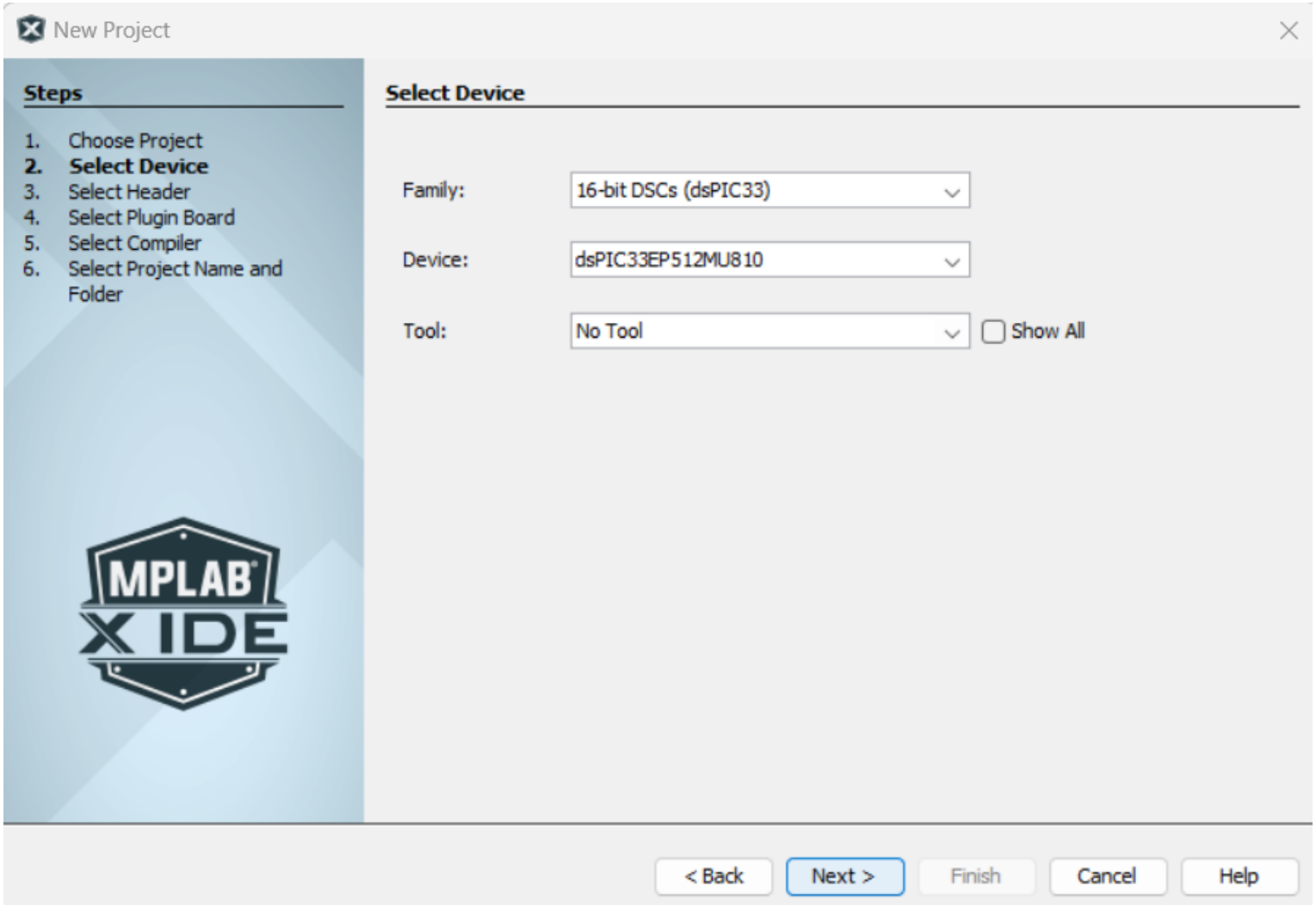
Creating a new Project I



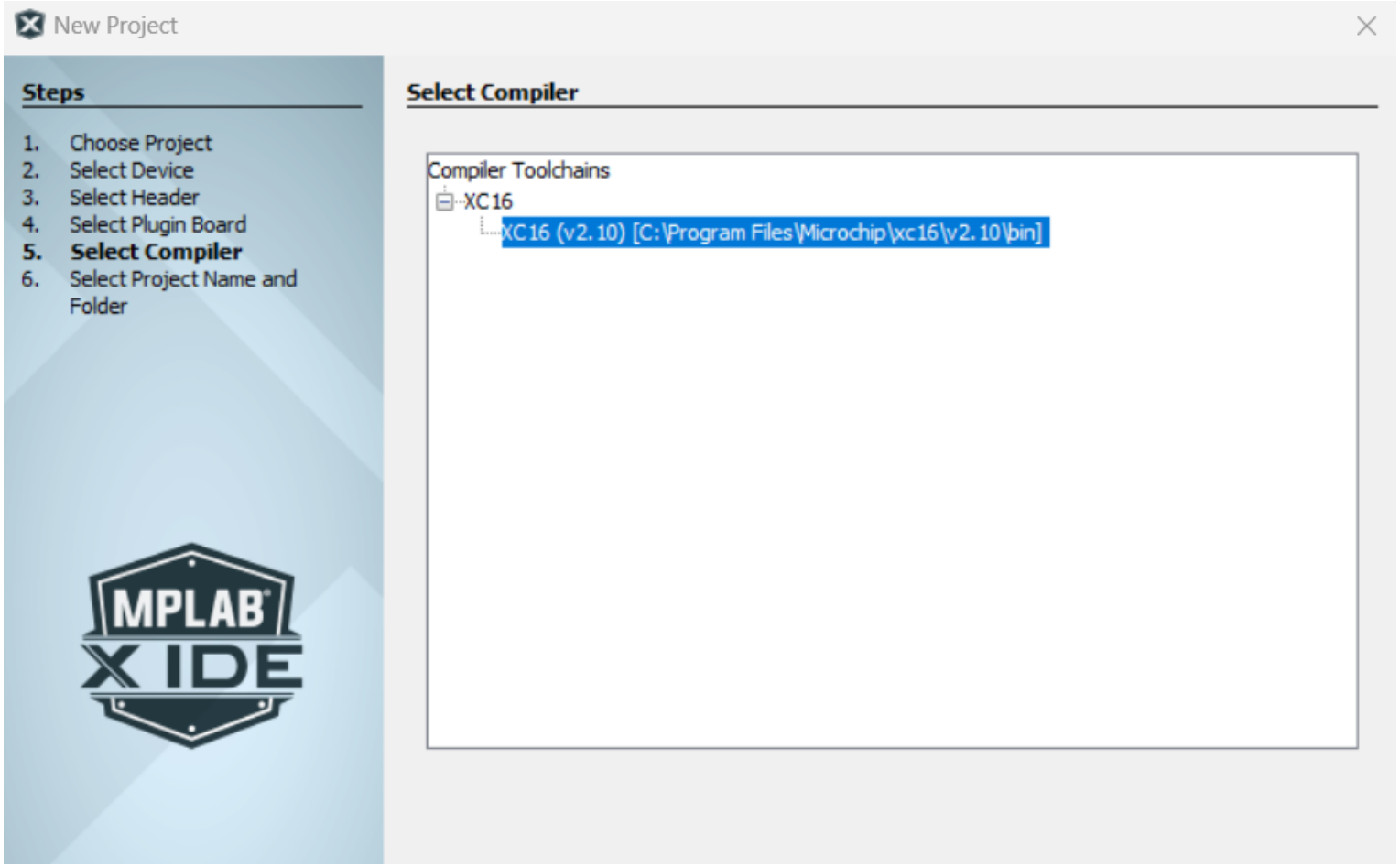
Creating a new Project II



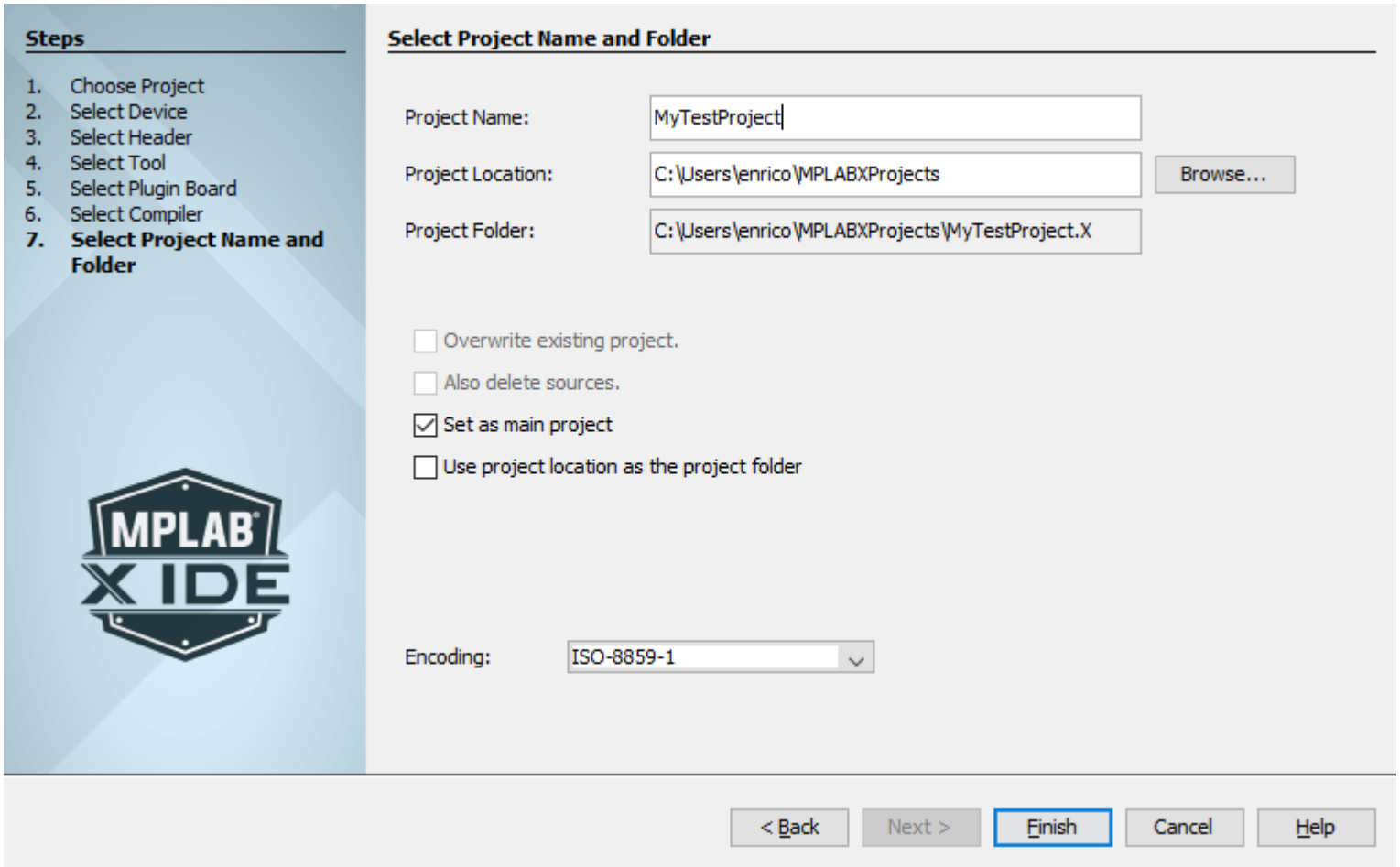
Creating a new Project III



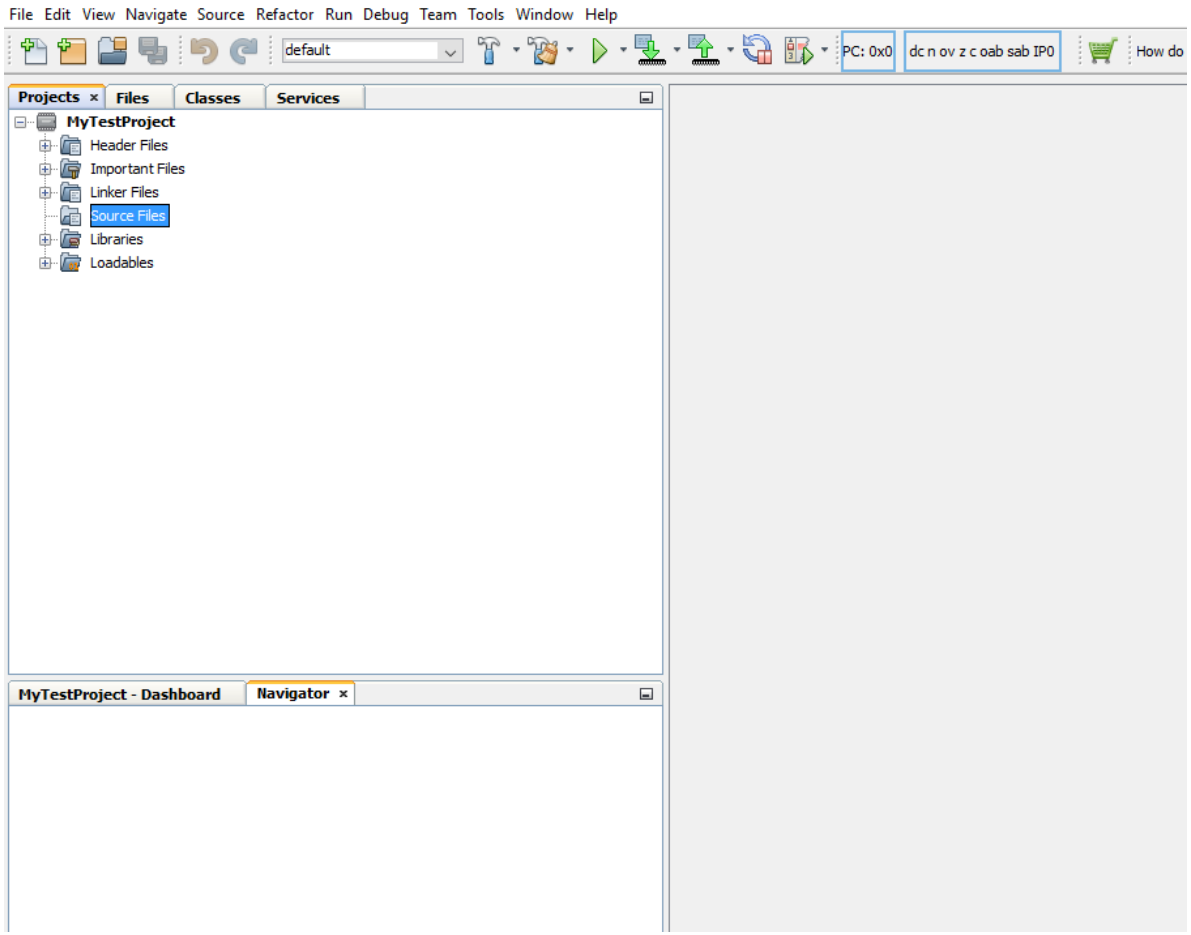
Creating a new Project IV



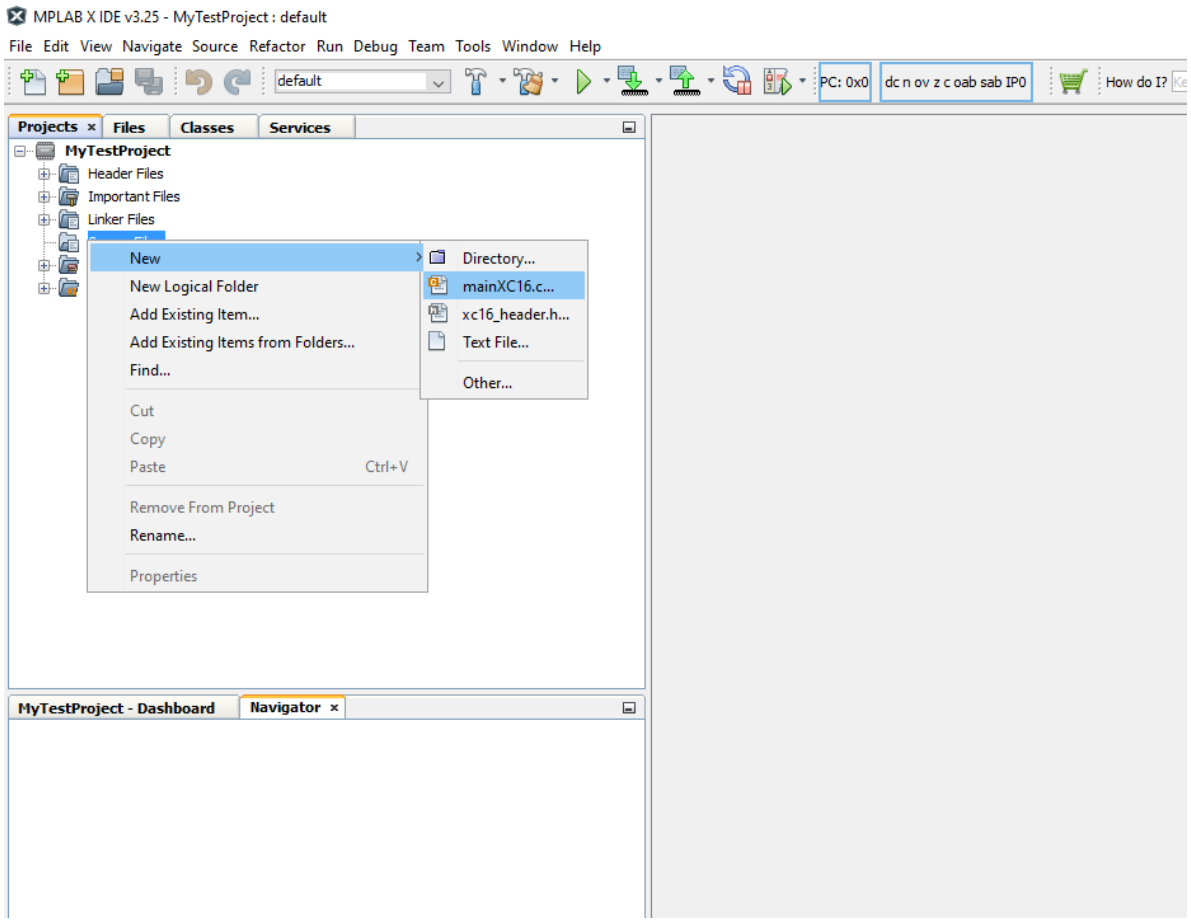
Creating a new Project V



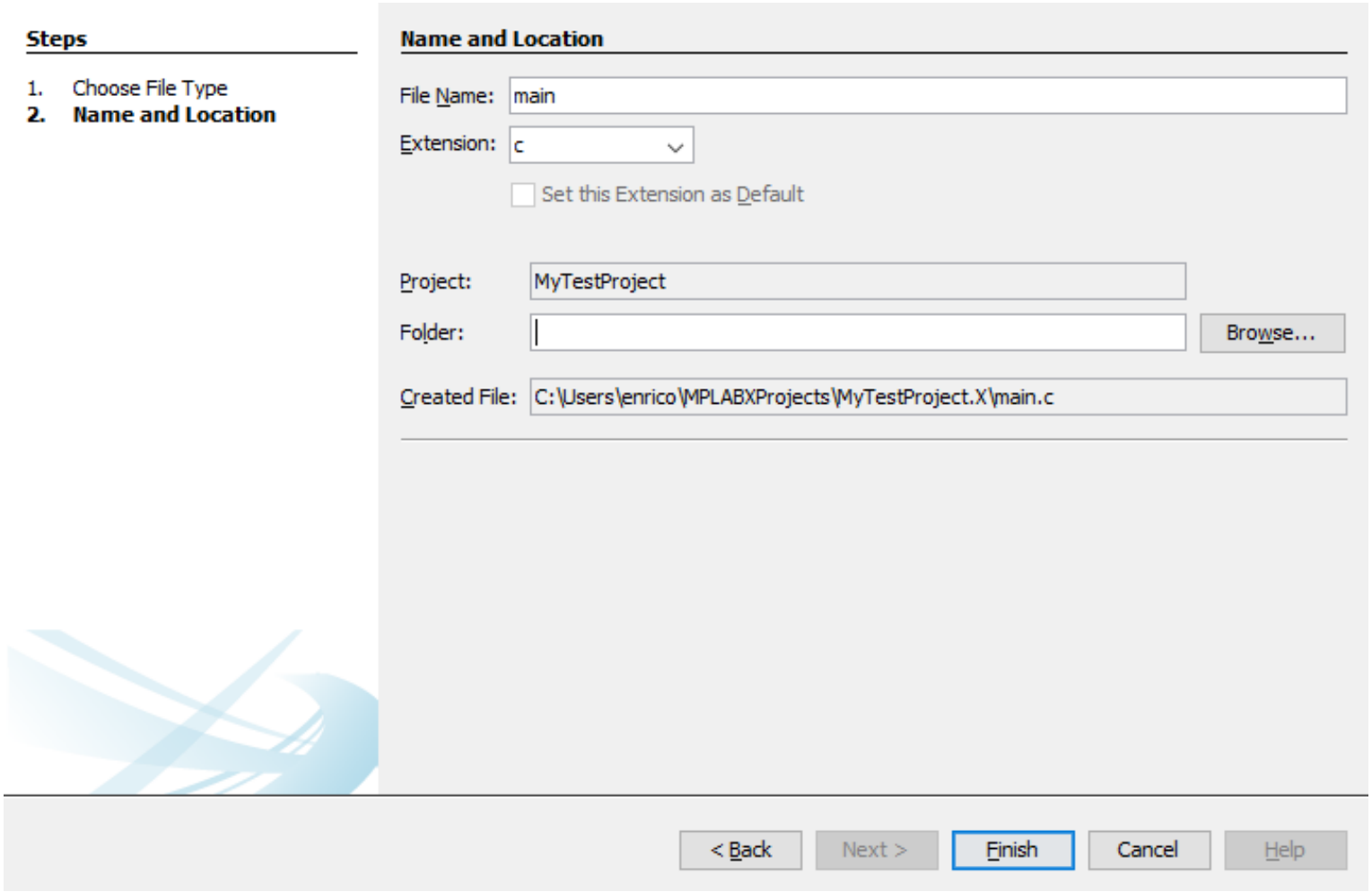
Creating a new Project VI



Creating a new Project VII

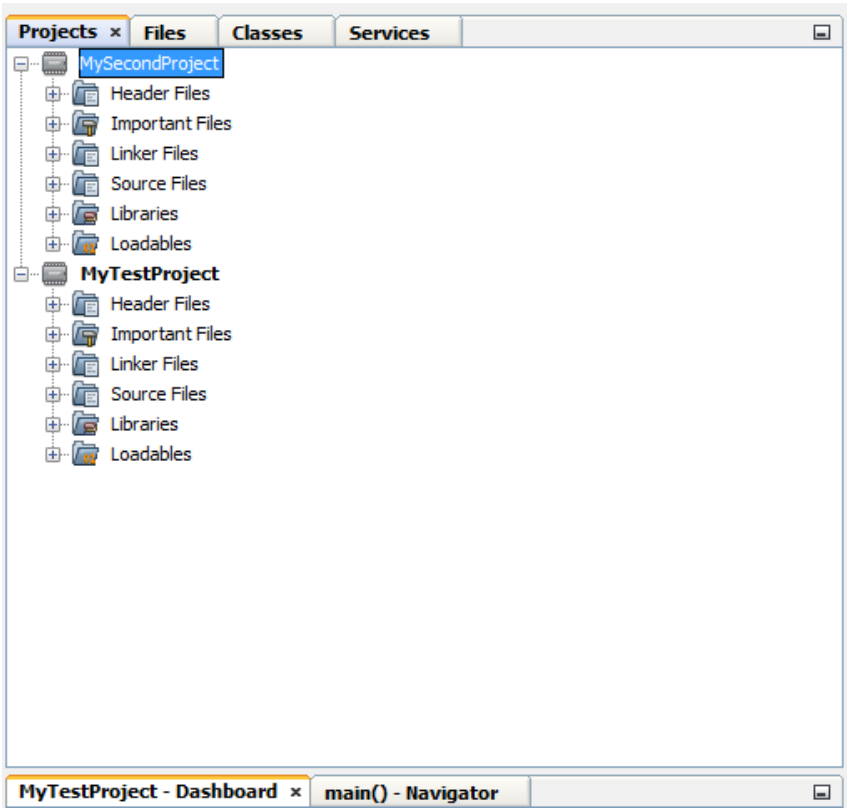


Creating a new Project VIII



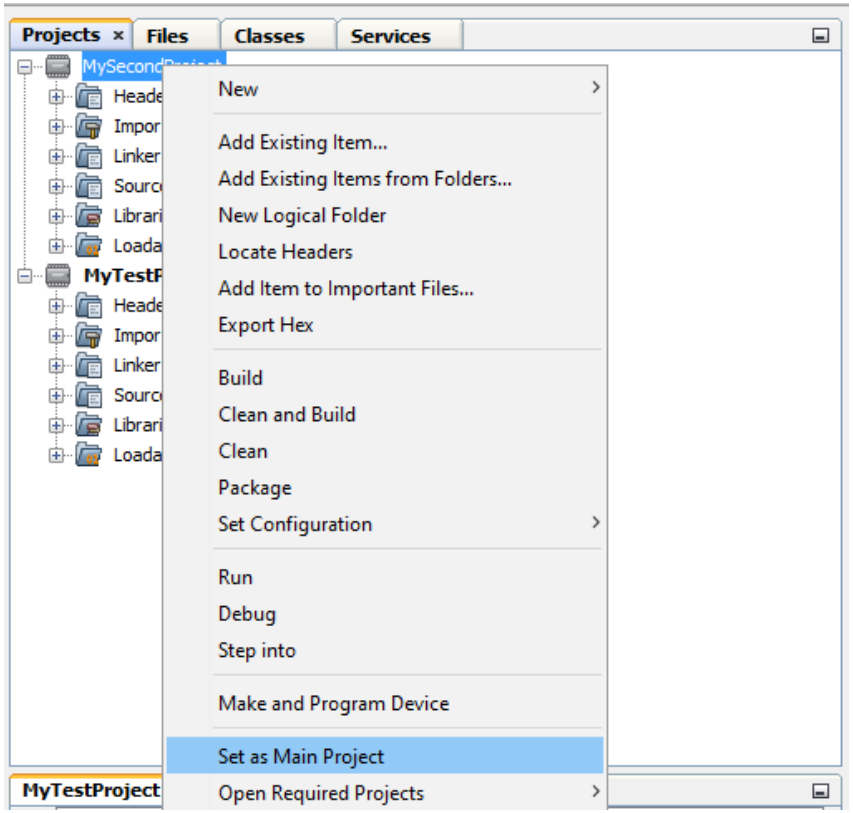
Multiple Projects I

Now the IDE is set to compile “MyTestProject”



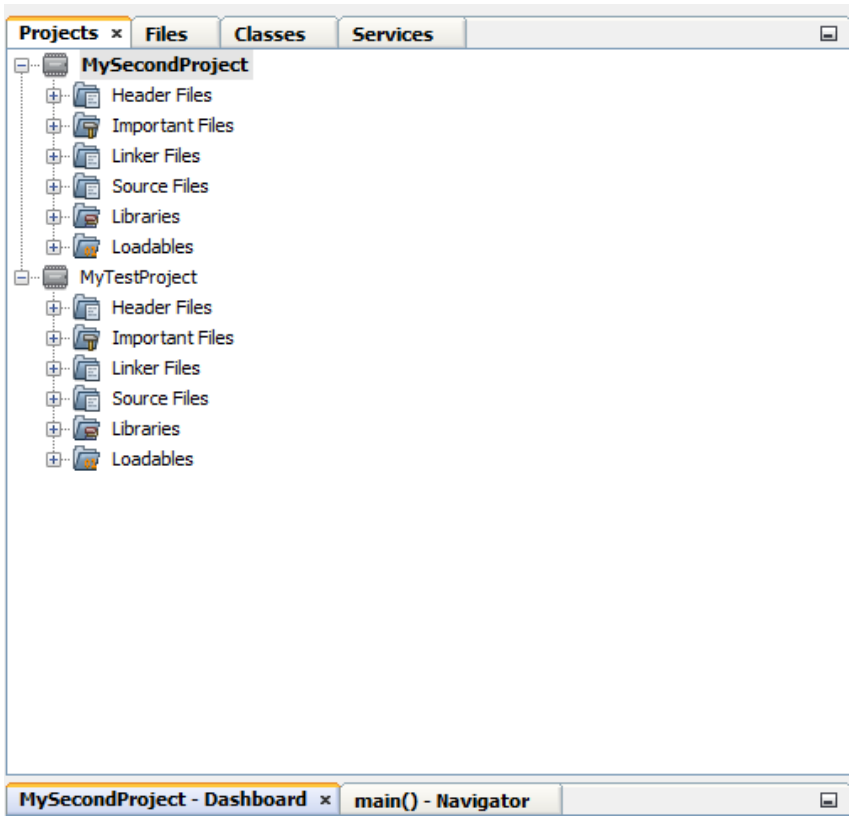
Multiple Projects II

If I want to compile “MySecondProject”, right click and select “Set as Main Project”

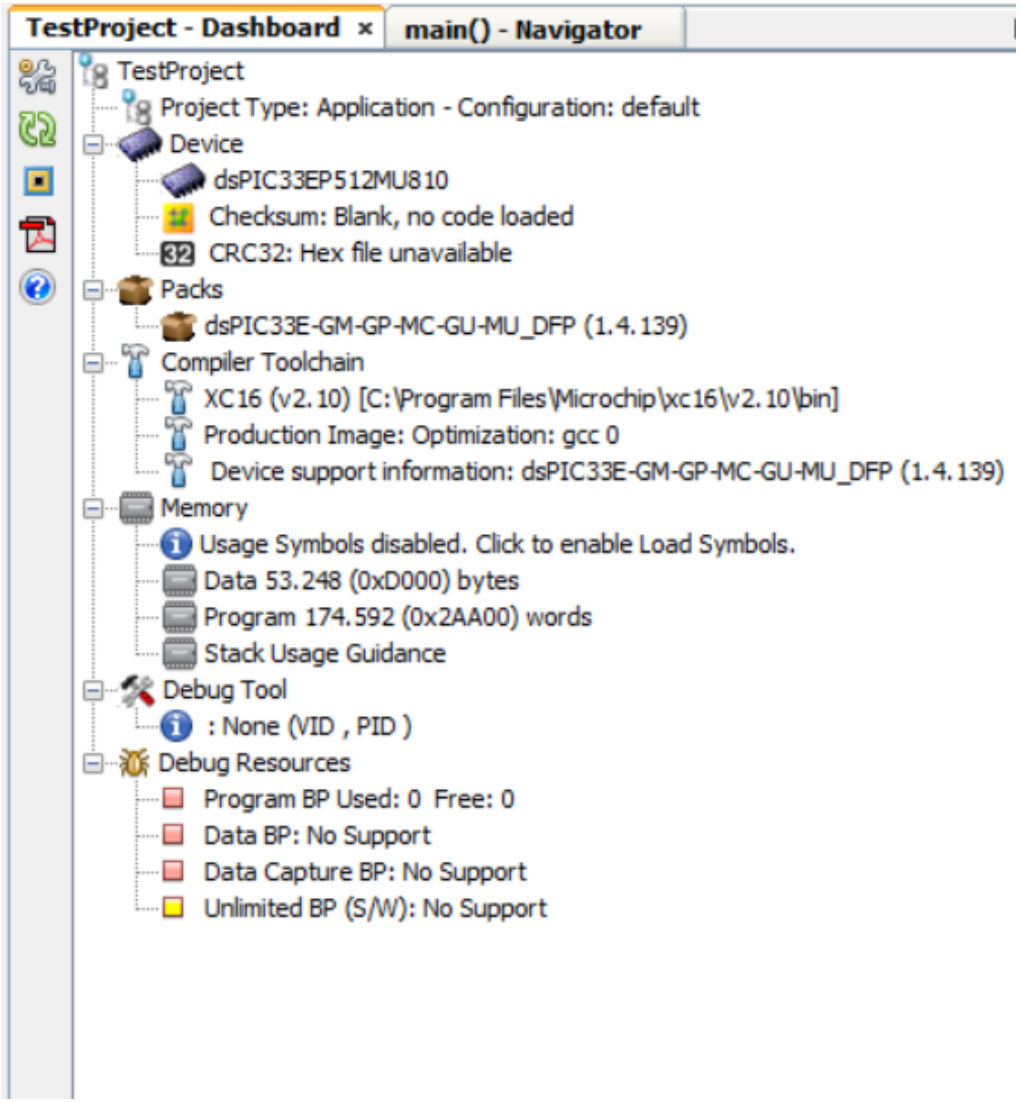


Multiple Projects III

Now “MySecondProject” is written in **bold** indicating that it is the main project



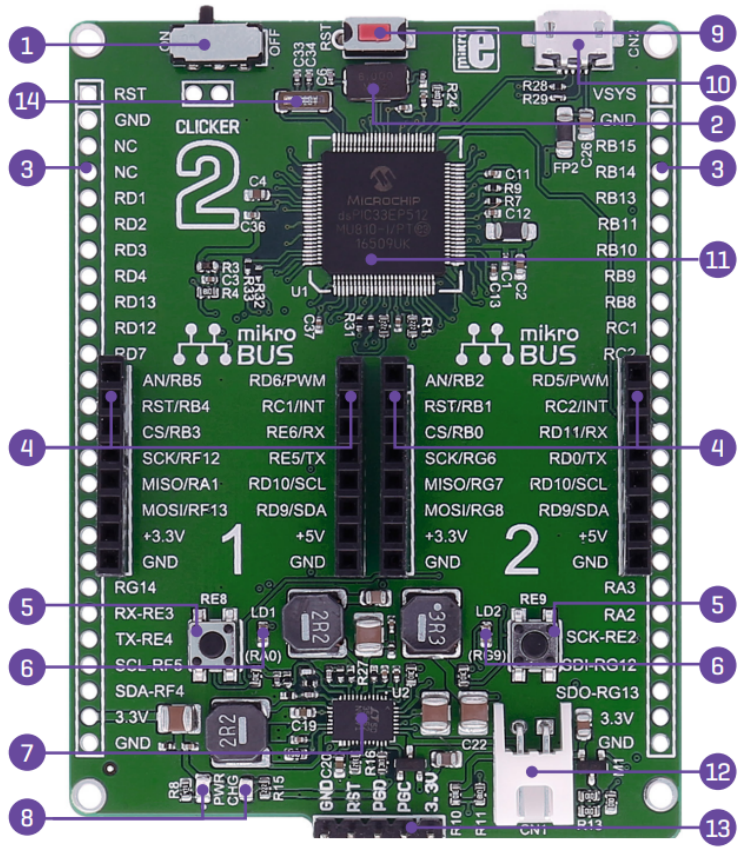
Dashboard


TestProject - Dashboard
main() - Navigator
TestProject
Project Type: Application - Configuration: default
Device
dsPIC33EP512MU810
Checksum: Blank, no code loaded
CRC32: Hex file unavailable
Packs
dsPIC33E-GM-GP-MC-GU-MU_DFP (1.4.139)
Compiler Toolchain
XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
Production Image: Optimization: gcc 0
Device support information: dsPIC33E-GM-GP-MC-GU-MU_DFP (1.4.139)
Memory
Usage Symbols disabled. Click to enable Load Symbols.
Data 53.248 (0xD000) bytes
Program 174.592 (0x2AA00) words
Stack Usage Guidance
Debug Tool
: None (VID , PID)
Debug Resources
Program BP Used: 0 Free: 0
Data BP: No Support
Data Capture BP: No Support
Unlimited BP (S/W): No Support

mikroE Clicker 2 board for dspic33

Key features

1. ON/OFF switch
2. 8 MHz crystal oscillator
3. two 1x26 connection pads
4. mikroBUS™ sockets 1 and 2
5. Pushbuttons
6. Additional LEDs
7. LTC3586 USB power manager IC
8. Power and Charge indication LEDs
9. RESET button
10. Micro USB connector
11. dsPIC33EP512MU810 MCU
12. Li-Polymer battery connector
13. mikroProg programmer connector
14. 32.768 KHz crystal oscillator



How to program? (Windows)

The Clicker2 board contains a *bootloader*, a light program stored permanently in memory that allows to flash firmware without a specific programmer. The mikroE *mikroBootloader* software is required to download the firmware on the board.
It's only available on Windows platforms.

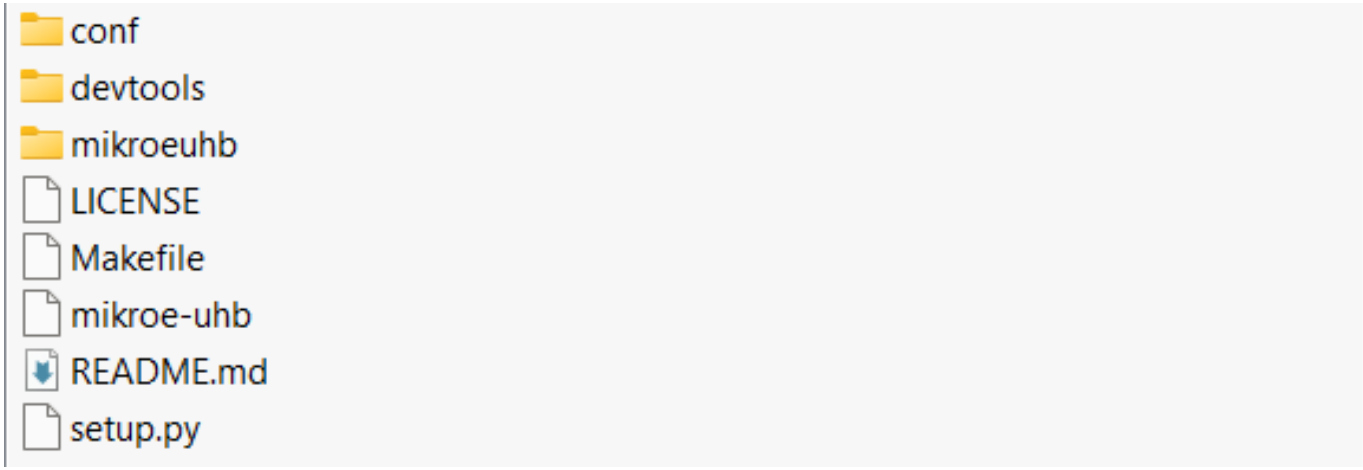
How to program? (Windows)

To write your program on the board use the following steps:

- Connect the board to your PC with the microUSB cable
- Press the RESET button (the red one) on the board
- the icon besides *Wait for the USB link* should become red. You have approx. 5 seconds to press the connect button on the interface. If you don't press it within 5 seconds the board will run the previously flashed firmware
- Once you're connected, select *Browse for HEX* and look for the following path: C:/Users/yourusername/MPLABXProjects/projectname.X/dist/default/production/projectname.X.production.hex. You generate this file every time you build your project in MPLAB.
- Click on *Begin uploading* and wait until the firmware is written on the board. After 5 seconds the board will execute your code

How to program? (Ubuntu)

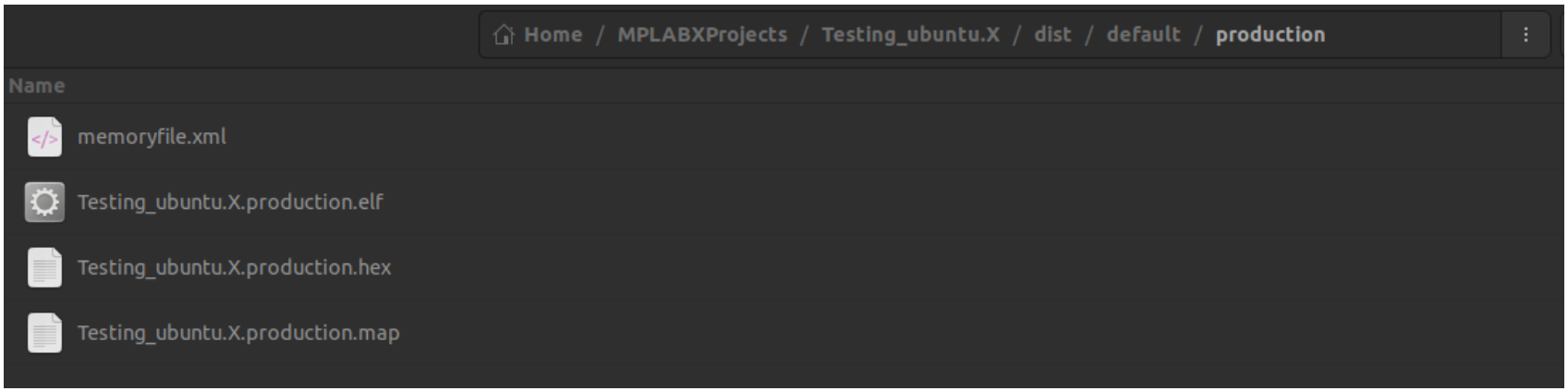
To write your program on the board from a Ubuntu distro you must install an additional tool:
`https://github.com/thotypous/mikroe-uhb`
Clone the *master* repository from the github link, the folder content will be the following:



Open a terminal in the main folder and run *make* and then *sudo make install* commands. Note that you must have a Python installation on your system. If you encounter issues in building the library, run *sudo pip install "setuptools<58.0.0"*

How to program? (Ubuntu)

To build your project on Ubuntu just follow the same steps shown in the Windows tutorial, then open a terminal in the folder containing the *.hex* file.



How to program? (Ubuntu)

Connect the board to the PC and run *sudo lsusb*: you need this step to know how the board is identified.
The output should look like follows:

```
sudo lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 0c45:6a1b Microdia Integrated_Webcam_FHD
Bus 003 Device 004: ID 046d:c03e Logitech, Inc. Premium Optical Wheel Mouse (M-BT58)
Bus 003 Device 003: ID 8087:0033 Intel Corp.
Bus 003 Device 016: ID 2dbc:0001 Mikroelektronika USB HID Bootloader
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

You can notice that the Bootloader stored on the board is recognized as *Mikroelektronika USB HID Bootloader*. If that record does not appear in the listing, press the reset button on the board and re-run the lsusb command within 5 seconds from the botton toggle.
Take note of the ID values in the listing, in this case *2dbc and 0001*.

How to program? (Ubuntu)

Now you can run the flashing command, remember that this is just an example and the *vendor id* and *product id* could be different on your machine.

```
francesco@francesco-Vostro-7620:~/MPLABXProjects/Testing_ubuntu.X/dist/default/production$ sudo mikroe-uhb -v --vend
or=0x2dbc --product=0x0001 Testing_ubuntu.X.production.hex
[sudo] password for francesco:
DEBUG:mikroeuhb.hid.linux:opening device vendor=2dbc, product=1
INFO:mikroeuhb.hid.linux:USB device 2dbc:0001 plugged
INFO:mikroeuhb.hid.linux:USB ID matches the expected one
DEBUG:mikroeuhb.device:send cmd: stx, cmd=INFO, addr=0x00000000, counter=0x0000
DEBUG:mikroeuhb.device:recv data: 3c010b0008000004080003000000c040080010500001306000004005000 7436c69636b6572203220666f
722064735049433333000000000000000000000549526cd
ERROR:mikroeuhb.bootinfo:Field 51 not recognized -- aborting parsing
McuType: 'DSPIC33'
EraseBlock: 0xc00
WriteBlock: 0x180
BootRev: 0x1300
BootStart: 0x54000
DevDsc: b'Clicker 2 for dsPIC3'
McuSize: 0x80400

DEBUG:mikroeuhb.device:send cmd: stx, cmd=BOOT, addr=0x00000000, counter=0x0000
DEBUG:mikroeuhb.device:recv cmd: stx, cmd=BOOT, addr=0x0008000b, counter=0x0400
DEBUG:mikroeuhb.device:send cmd: stx, cmd=SYNC, addr=0x00000000, counter=0x0000
DEBUG:mikroeuhb.device:recv cmd: stx, cmd=SYNC, addr=0x0008000b, counter=0x0400
DEBUG:mikroeuhb.devkit:reset code before fix: 000204000000
DEBUG:mikroeuhb.devkit:reset code after fix:  004004050000
DEBUG:mikroeuhb.devkit:transfer to device starting
```

At this point the terminal will hang in loop until the on-board bootloader is detected, showing an *opening device* message. Just press again the reset button and the flashing process will begin. The output should be similar to the following screen.