Evo Laser Firmware Developer’s Manual

Table of Content

1. Introduction

Evo Laser is the next generation laser pointer released by Wickedlasers. Its also the first fully open sourced and smartphone enabled intelligent laser pointer in the world.

Due to the open source nature of the laser pointer, this manual is produced to facilitate development of alternate firmware for the Evo Laser pointer.

Who should read this document

This manual is produced for hardware and firmware developers who are interested in developing for the Evo Laser series.

License applicable to Evo Laser firmware source files and hardware project files

All firmware source and project files for the Evo Laser series released by WickedLasers Inc are licensed under ***Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)*** license. Under this license you are allowed to

- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work

However, you can not derive commercial work upon these resources without written permission from Wickedlasers Inc.

The details of the license can be found here http://creativecommons.org/licenses/by-nc-sa/3.0/.

Wickedlasers Inc encourages development of commercial products based on our Evo Laser products but we would like to be informed of such developments.

2. Hardware Overview and Its subsystems

2.1 Overview

The core Evo Laser pointer hardware is consisted of the following components

- Laser diode
- PWM controlled laser driver
- Microcontroller IC from Microchip PIC12F683
- and an extension header which can be used to connect to the Smartport^{TM} adapter.

The Smartport^{TM} is an adapter for the Evo Laser pointer that can be used to connect to smartphones with a common 3.5mm audio jack. Currently iOS and Android apps are available which use this technology.

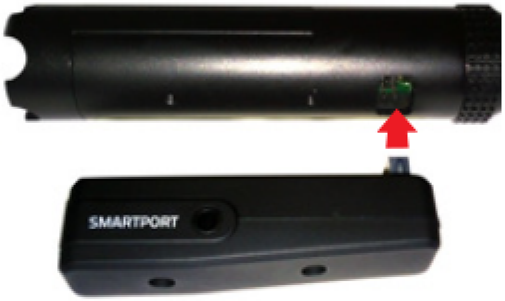


Figure 1. Smartport^{TM} Extension

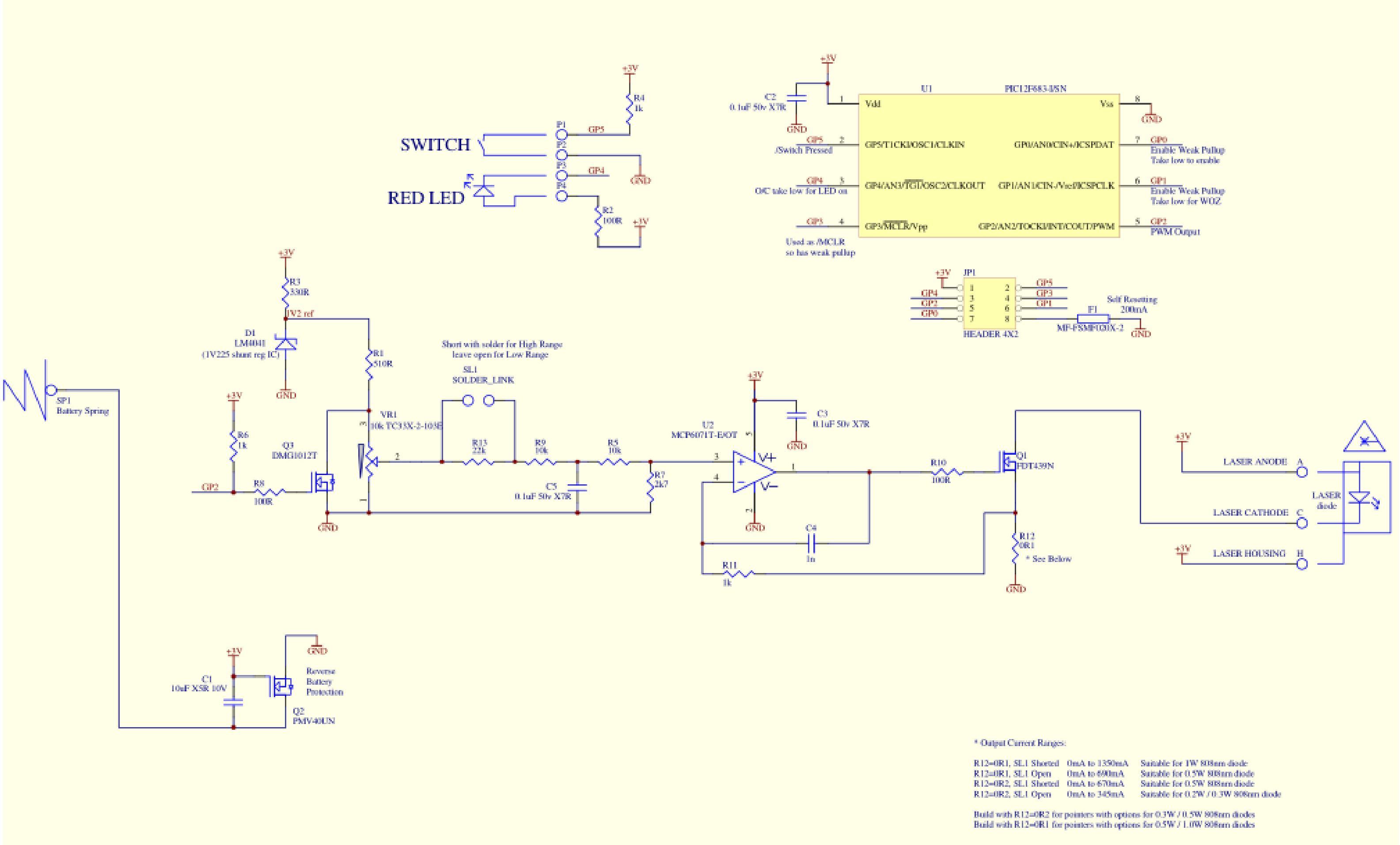
2.2 Evo Laser Hardware Core System

This section details the internal working of Evo lasers. The goal is to provide sufficient background knowledge to assist development of alternative firmware.

Figure 2 is the complete schematic of the Evo Laser pointer. Following are the key points in understanding the schematic.

- D1, R3 provides a supply independent voltage reference.
- R6, R3 and Q3 inverts the PWM signal.

- R1 and VR1 provides an adjustable voltage reference with varying source impedance (which is an unwanted but negilible).
- R13, SL1, R9, R5, R7 and C5 form a low pass filter which averages out the PWM signal.
- When SLI is shorted, the 3DB point of the RC low pass filter is 175Hz and when SLI openned, the bandwidth is 280Hz. Since PWM is running at 31.250kHz at GP2, it is way higher than the 3DB point.
- R5 and R7 is a voltage scaler which is used to size the non-inverting voltage of U2 correctly. Without them, C5 needs to be a lot larger.
- U2 and Q1 is a voltage follower (buffer). Voltage across R12 will be the same as voltage at the non-inverting input (V+) of U2.
- R10, R11 and C4 provides dominant pole compensation
- R12 converts voltage to current.



* Output Current Ranges:

R12=0R1, SL1 Shorted	0mA to 1350mA	Suitable for 1W 808nm diode
R12=0R1, SL1 Open	0mA to 690mA	Suitable for 0.5W 808nm diode
R12=0R2, SL1 Shorted	0mA to 670mA	Suitable for 0.5W 808nm diode
R12=0R2, SL1 Open	0mA to 345mA	Suitable for 0.2W / 0.3W 808nm diode

Build with R12=0R2 for pointers with options for 0.3W / 0.5W 808nm diodes
Build with R12=0R1 for pointers with options for 0.5W / 1.0W 808nm diodes

Figure 2 Evo Laser Schematic

The microcontroller used in Evo laser is PIC12F683 from Microchip. It has 8 pins which are used as follows in this project.

- Pin 1, Vdd, voltage supply
- Pin 2, GP5 (General Purpose 5), SmartSwitch input, digital, active low.
- Pin 3, GP4, red LED control, output, digital, active low.
- Pin 4, Master Clear, weak pullup enabled.
- Pin 5, GP2, PWM output, digital, this controls the laser driver
- Pin 6 and Pin 7, GP1 and GP0, input, week pullup enabled, active low, these pins are used as mode select. See firmware chapter for more details.
- Pin 8, Ground.

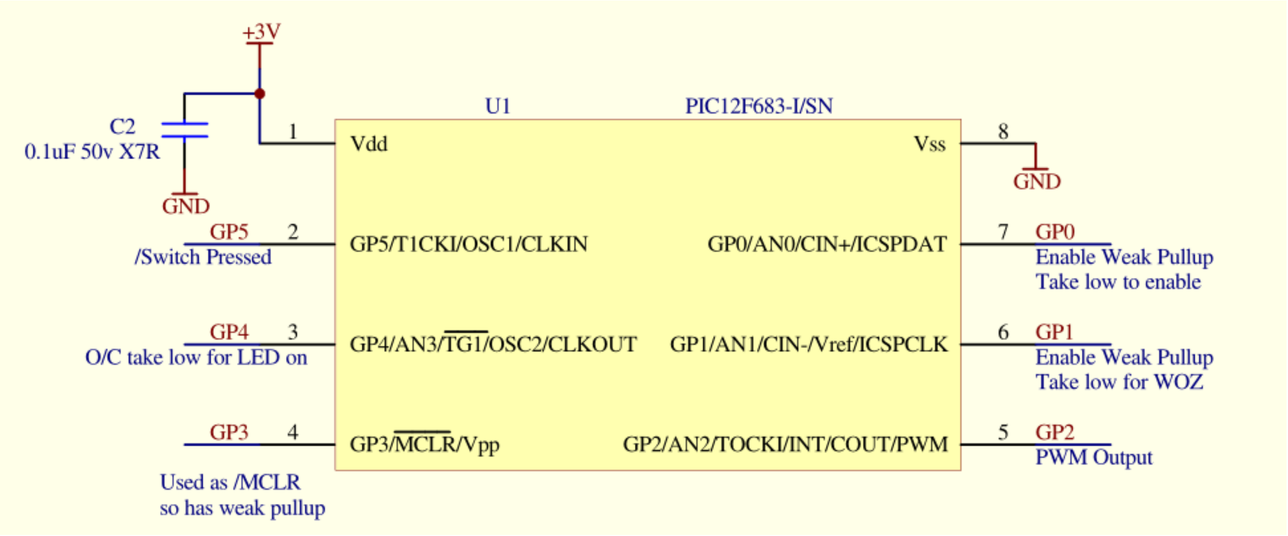


Figure 3, PIC12F683

The last component of Evo schematic is the extension header. Image is attached and it is self explanatory.

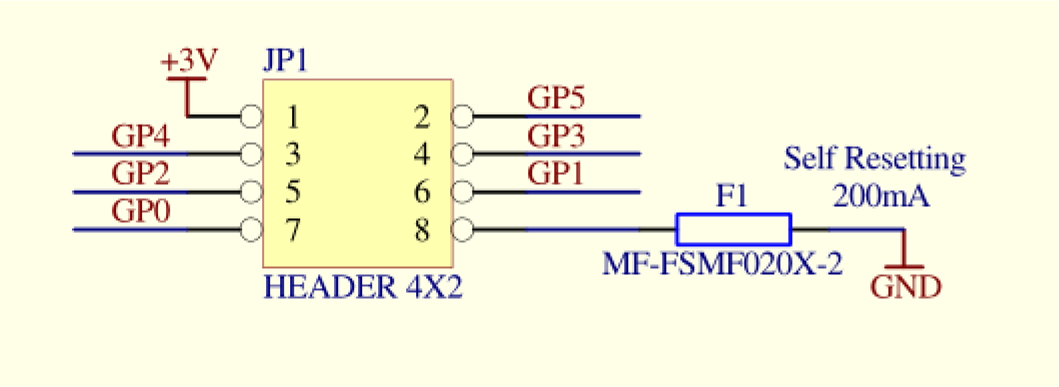


Figure 4, Extension Header

2.3 *Evo Laser Smartport™*

Smartport™ adapter is the proprietary adapter that can be used to connect Evo laser to smartphones with 3.5mm audio port. There is no microcontroller used Smartport™ so no firmware development is possible with it. If you are interested in developing smartphone apps based on Smartport™, please refer to our mobile app development guide.

The hardware of Smartport™ is very simple. It consists of a 3.5mm Audio Jack and two signal paths from L and R of audio jack to GP0 and GP1. GP0's path is AC coupled while GP1 is DC coupled. Both path have voltage clamping diode protection.

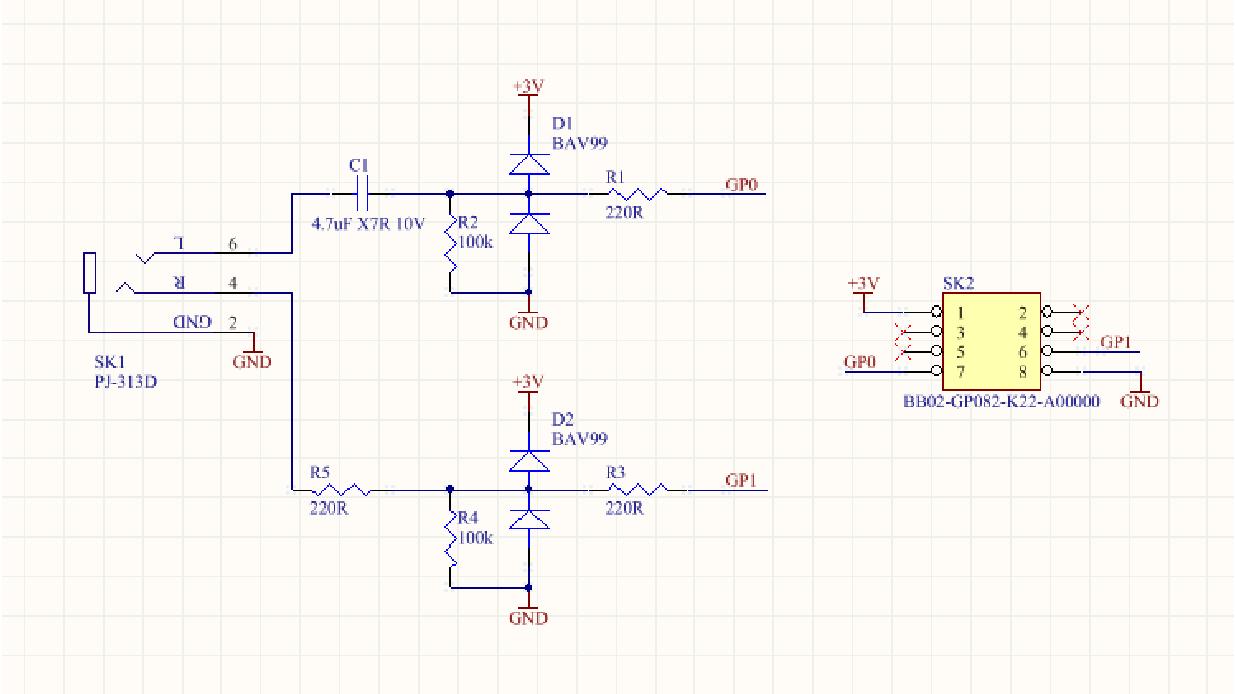


Figure 5, Smartport Schematic

3.Firmware Development

3.1 Introduction

The original and official firmware for Evo laser has been developed using WizC IDE and PIC compiler. However, we discourage future development on Evo to be based on this toolset as we discovered a few bugs associated with the WizC toolset.

We have plans to port the original source code of Evo laser firmware to Microchip's offical IDE, MPLAB X.

Due to these reasons, the source code released for Evo laser will be based on WizC but this document will only details the setup necessary for development with MPLAB X.

MPLAB X based source code will be released as soon as they are available.

3.2 Evo Laser Development Requirements

There are three requirements to Evo firmware development besides having a Evo laser pointer.

- Evo Laser programmer
- MPLAB X IDE
- Free MPLAB XC8 C Compiler

The only extra piece of hardware you will need for developing Evo firmware is the Wickedlasers Evo Programmer (also called Programming Plug in the Evo user manual). This programmer can be used to flash custom firmware into the Evo Laser as well as used as a programmer/debugger with MPLab X IDE.

This programmer can be purchased directly on our website.

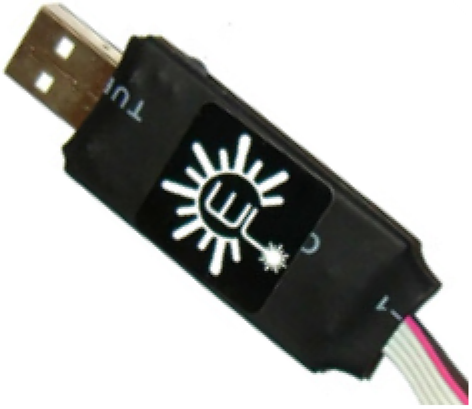
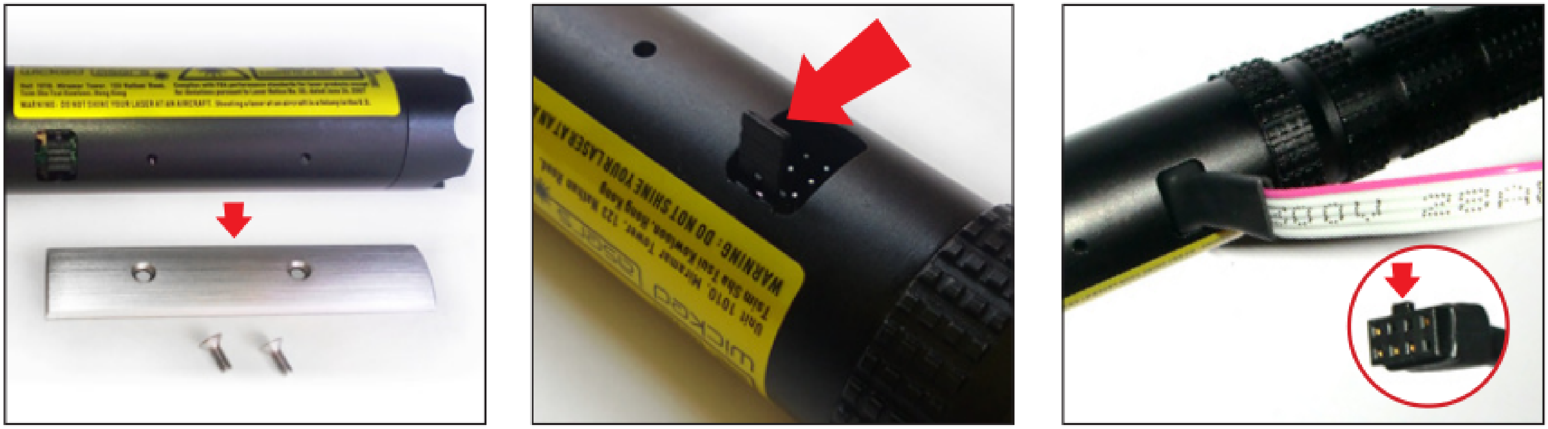


Figure 6, WickedLasers Evo Programmer

The Evo programmer has a custom plug that can be directly plugged into the Evo extension socket. Make sure you that you have the notch on the plug facing front of the laser. The other end of the Evo Programmer is an standard USB plug. Plug this into your computer.



With the laser powered off, remove the bottom metal panel.

You will see the port where to connect the Programming Plug. Remove the jumper first.

Connect the Programming Plug. Make sure that the notch on the plug is facing the front of the laser.

If the laser is facing left, the red wire should be on top when properly connected.

Figure 7, how to plug in the Evo Programmer

MPLAB X IDE can be download at http://www.microchip.com/pagehandler/en-us/family/mplabx/. MPLAB X supports Windows, Linux and OS X. Choose the package appropriate for your platform.

Lastly, you will need to download the free MPLAB XC8 C Compiler, which can be located here http://www.microchip.com/pagehandler/en_us/devtools/mplabxc/

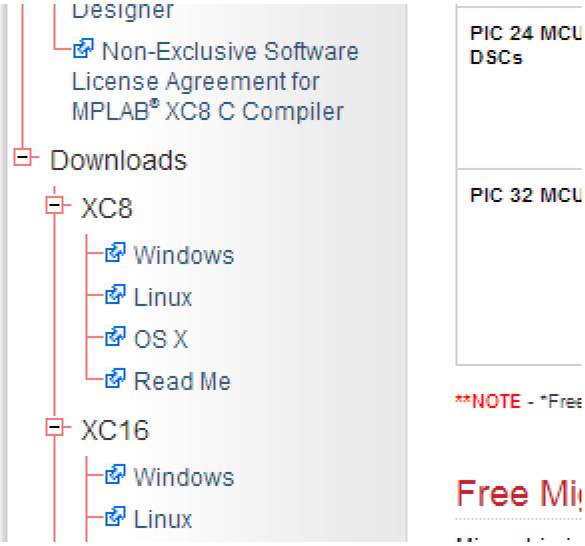
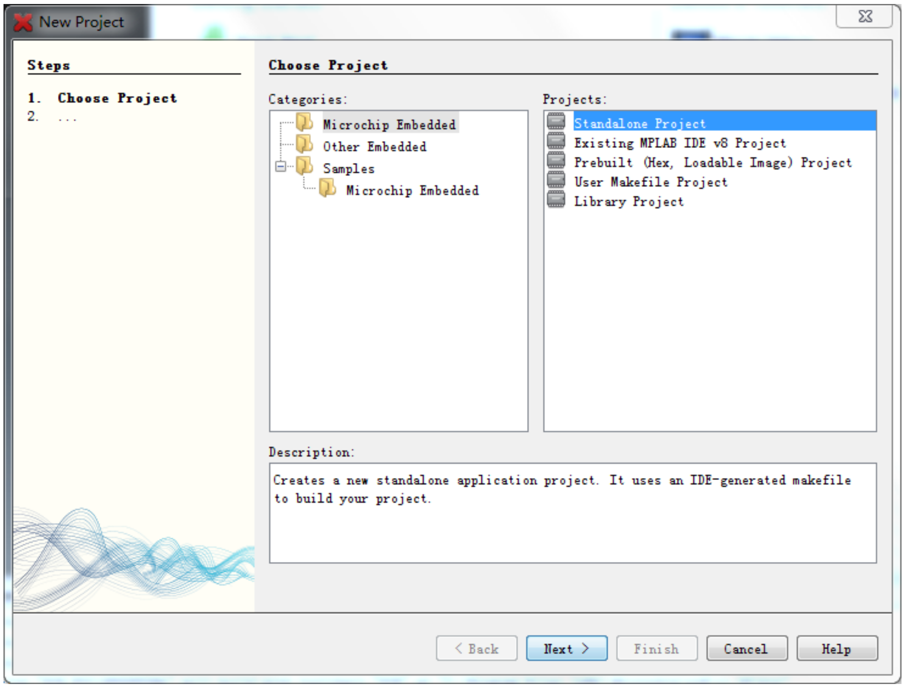


Figure 8, Locate XC8 on the page http://www.microchip.com/pagehandler/en_us/devtools/mplabxc/

3.3 MPLAB X Step By Step Guide

While Evo programmer plugged in, you are ready to start develop for Evo!
Follow the guide below to kick start your fantastic project!



New Project

Steps

1. **Choose Project**
2. ...

Choose Project

Categories:

- Microchip Embedded
- Other Embedded
- Samples
  - Microchip Embedded

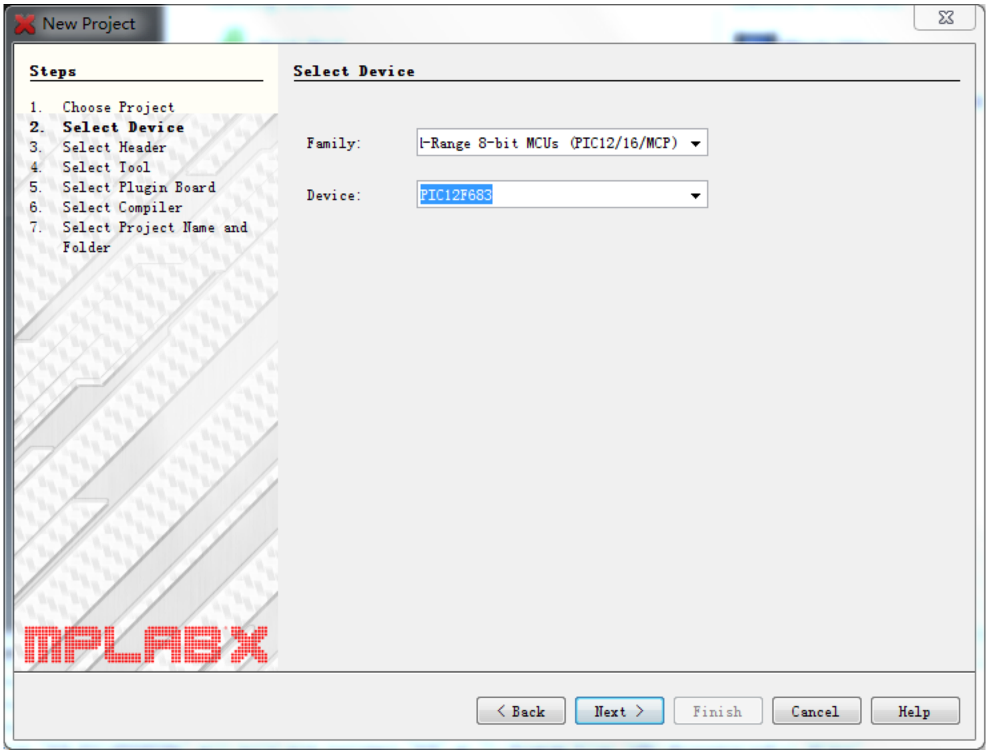
Projects:

- Standalone Project
- Existing MPLAB IDE v8 Project
- Prebuilt (Hex, Loadable Image) Project
- User Makefile Project
- Library Project

Description:

Creates a new standalone application project. It uses an IDE-generated makefile to build your project.

Step 1. Create Standalone Project with menu command File->New Project

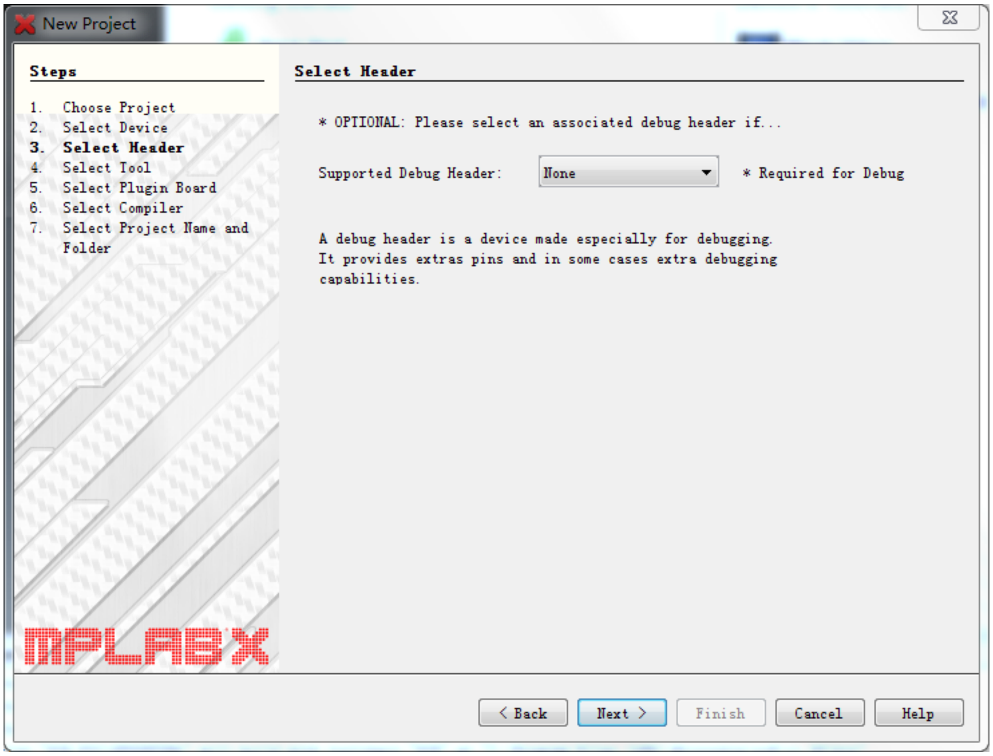


Select Device

Family: l-Range 8-bit MCUs (PIC12/16/MCP)

Device: PIC12F683

Step 2, Select Device Mid Range 8-bit, PIC12F683



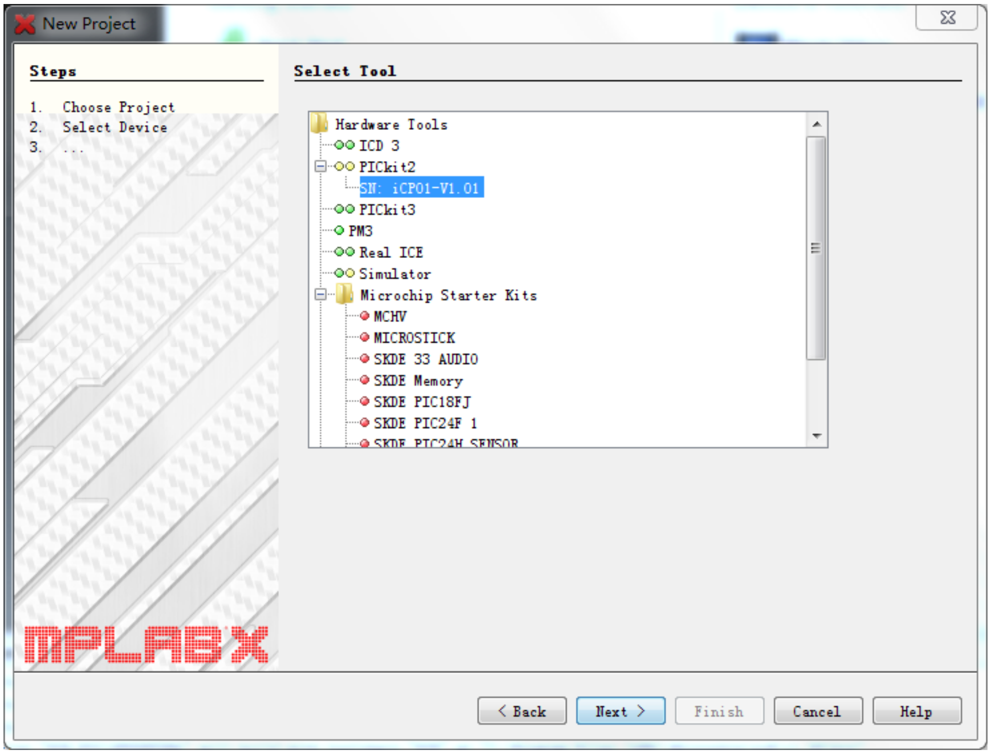
Select Header

* OPTIONAL: Please select an associated debug header if...

Supported Debug Header: None * Required for Debug

A debug header is a device made especially for debugging. It provides extras pins and in some cases extra debugging capabilities.

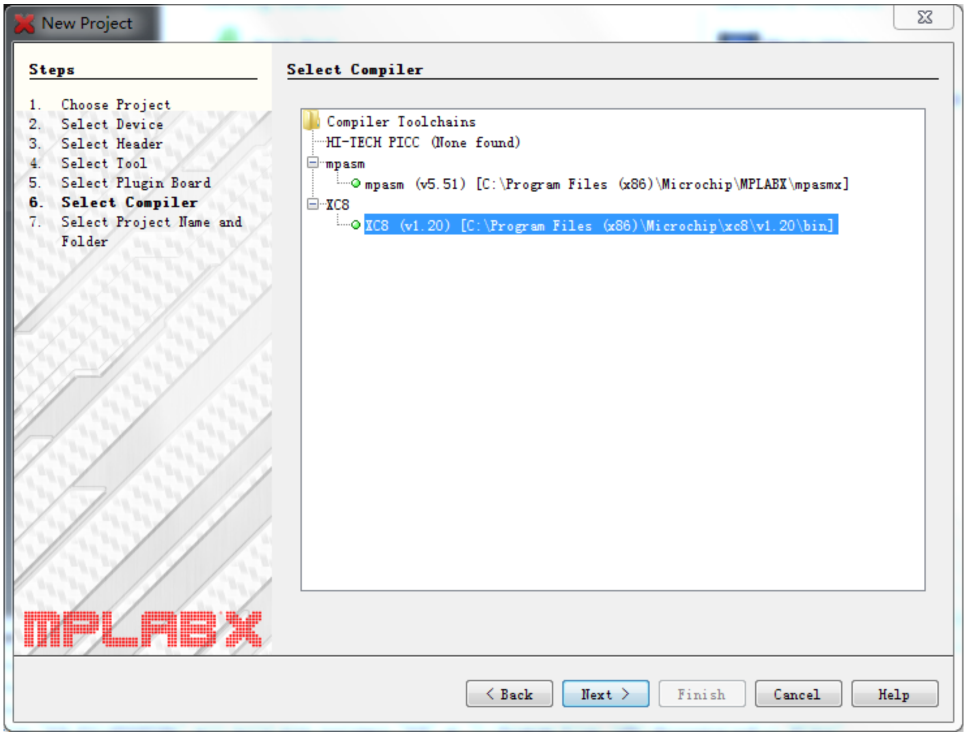
Step 3, Choose None for Debug Header



Select Tool

- Hardware Tools
  - ICD 3
  - PICkit2
    - SN: iCP01-V1.01
  - PICkit3
  - PM3
  - Real ICE
  - Simulator
  - Microchip Starter Kits
    - MCHV
    - MICROSTICK
    - SKDE 33 AUDIO
    - SKDE Memory
    - SKDE PIC18FJ
    - SKDE PIC24F 1
    - SKDE PIC24H SENSOR

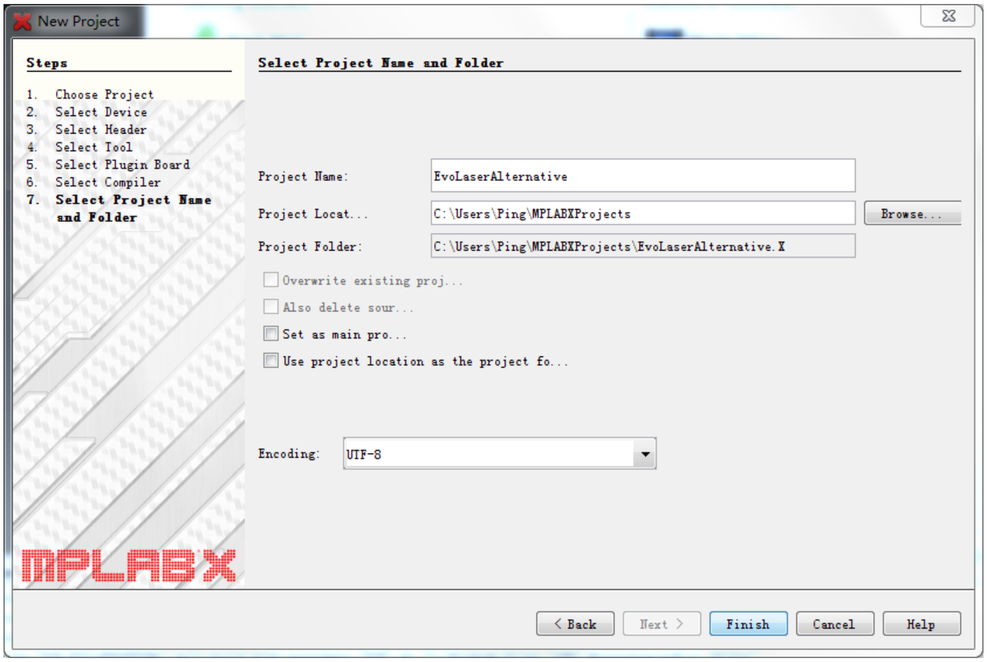
Step 4, Choose PICkit2 as tool, your SN may be different.



Select Compiler

- Compiler Toolchains
  - HI-TECH PICC (None found)
  - mpasm
    - mpasm (v5.51) [C:\Program Files (x86)\Microchip\MPLABX\mpasmx]
  - XC8
    - XC8 (v1.20) [C:\Program Files (x86)\Microchip\xc8\v1.20\bin]

Step 5, Choose XC8 as compiler, it’s only available if you have downloaded and installed the XC8 compiler



Steps

1. Choose Project
2. Select Device
3. Select Header
4. Select Tool
5. Select Plugin Board
6. Select Compiler
7. **Select Project Name and Folder**

Select Project Name and Folder

Project Name: EvoLaserAlternative

Project Locat... C:\Users\Ping\MPLABXProjects

Project Folder: C:\Users\Ping\MPLABXProjects\EvoLaserAlternative.X

□ Overwrite existing proj...

□ Also delete sour...

□ Set as main pro...

□ Use project location as the project fo...

Encoding: UTF-8

Step 6, last step, name your project and click finish!

You are now ready to get creative!

Appendix

Revisions

8/27/2013 Added Chapter 3 and Appendix
8/26/2013 Initial Work