ECE3411

Lecture#1c

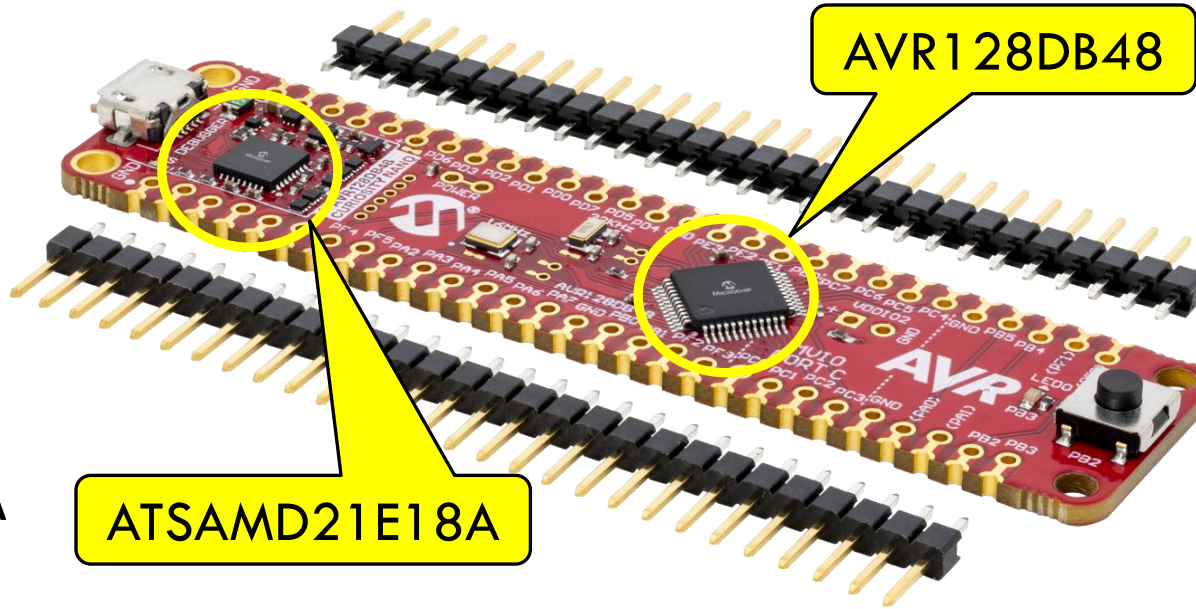
AVRDX Curiosity Nano Board Setup

Sung-Yeul Park
Department of Electrical & Computer Engineering
University of Connecticut
Email: sung_yeul.park@uconn.edu

Slides adopted from Marten van Dijk, Syed Kamran Haider, and John Chandy

AVR128DB48 Curiosity Nano

- The AVR128DB48 Curiosity Nano Mini evaluation board provides a development platform for the Microchip AVRDB Microcontroller.
- Target Microcontroller: AVR128DB48
- On-board Programming & Debugging capability using Atmel Studio
  - Debugger Microcontroller: ATSAMD21E18A
- USB connectivity
- Headers & Connectors for accessing target microcontroller's I/O pins



Development Board Setup has two steps

1. Attaching connectors for AVR128DB48 Curiosity Nano board
2. Putting everything together on the breadboard

Initial board setup

- Setup Microchip Studio
  - Microchip Studio is available for download at: https://www.microchip.com/en-us/tools-resources/develop/microchip-studio
  - Equivalent to Atmel Studio
- Before you start wiring up the breadboard make sure the Curiosity board by itself is working fine.
  - Get the test code provided on the next slide working for your Curiosity board.

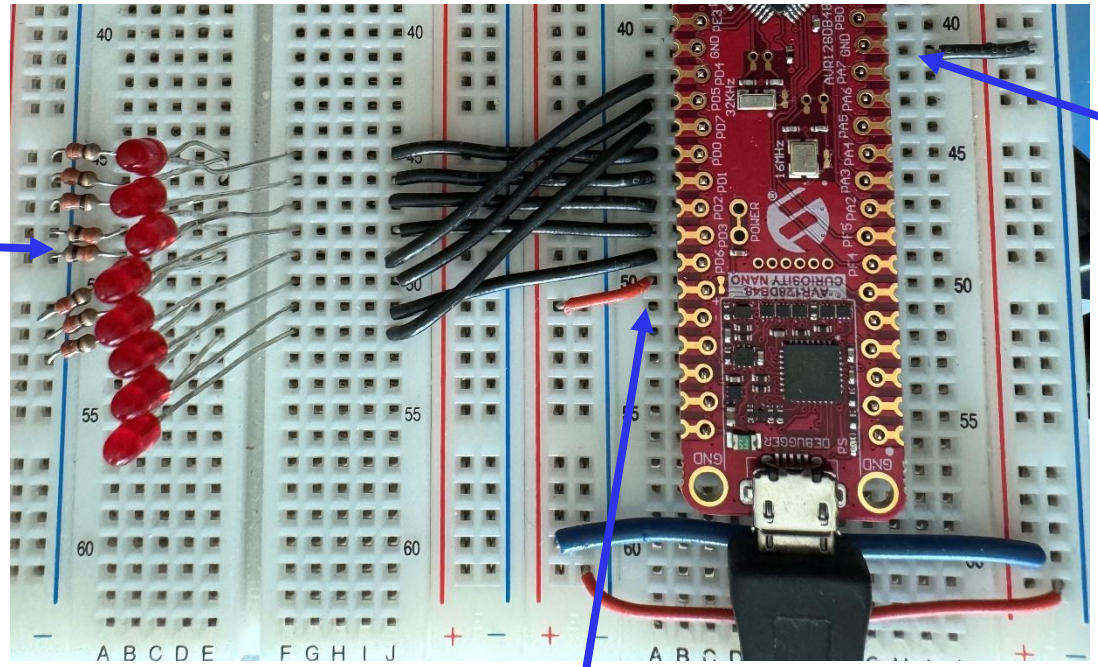
Connections for general digital output lab practice

- Make the connection as depicted on next slide.
- Connection requires 8 LEDs and 8 resistors
- Port D will be connected to the LED arrays. Port D has pins from PD0 to PD7
- Resistors are with the value of 330Ω
- Be cautious about the polarity of LEDs and value of the resistors.
- Also connect ground to the common point of the resistors

Connections

Note that the
PORTD pins are
not in order

330Ω resistors
to GND



GND

3.3V Target voltage

Test code

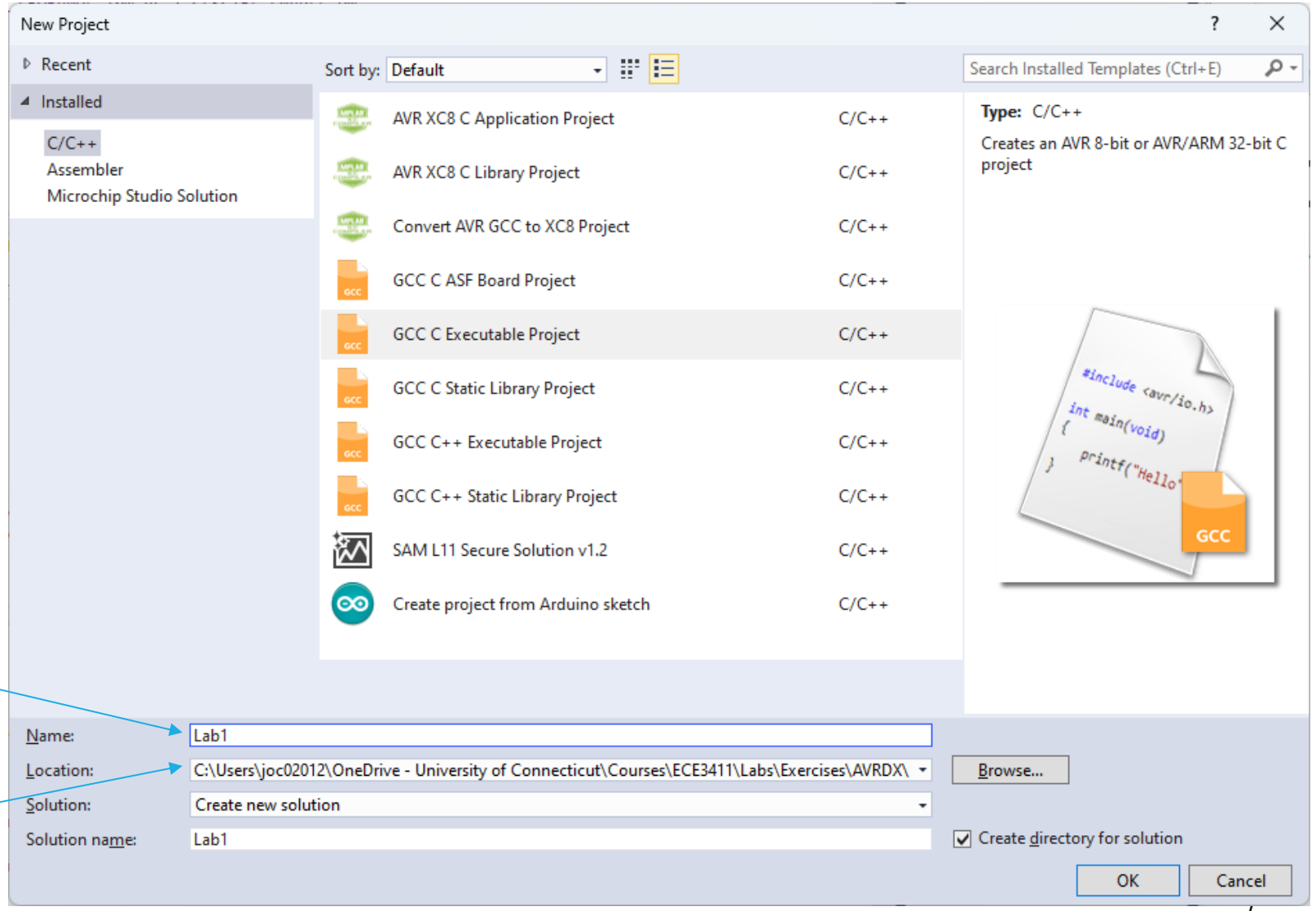
```
// ------- Preamble -------- //
#define F_CPU 4000000UL    /* Tells the Clock Freq to the Compiler. */
#include <avr/io.h>        /* Defines pins, ports etc. */
#include <util/delay.h>    /* Functions to waste time */

int main(void) {
    // -------- Inits --------- //
    /* Data Direction Register B: writing a one to the bit enables output. */
    PORTB.DIRSET = PIN3_bm;

    // ------ Event loop ------ //
    while (1) {
        PORTB.OUTSET = PIN3_bm;   /* Turn on the LED bit/pin in PORTB */
        _delay_ms(1000);          /* wait for 1 second */
        PORTB.OUTCLR = PIN3_bm;   /* Turn off the LED bit/pin */
        _delay_ms(1000);          /* wait for 1 second */
    } /* End event loop */
    return (0); /* This line is never reached */
}
```

Microchip Studio Tutorial

- File->New Project ..

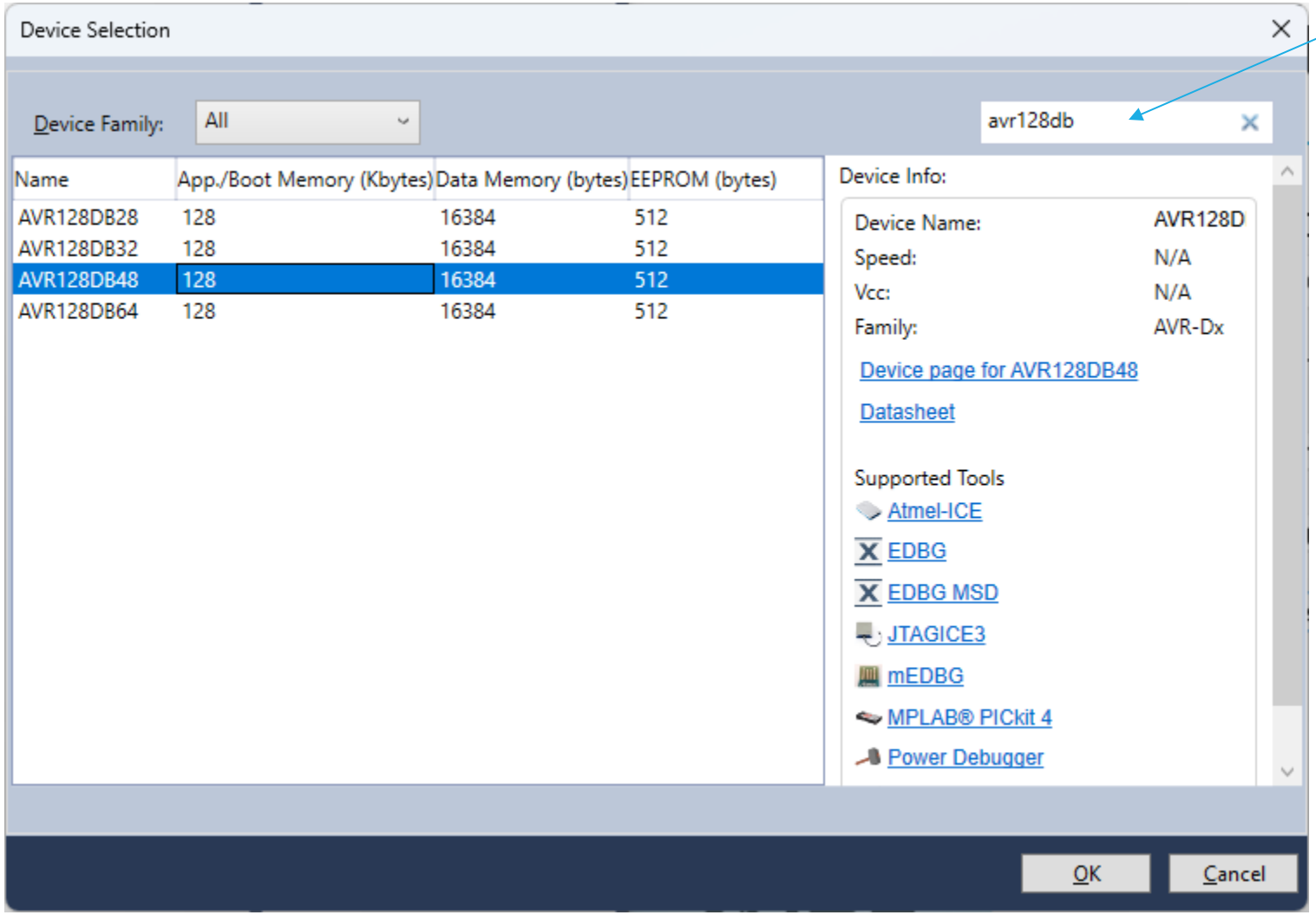


Choose a reasonable name – don't use the default GccApplication1

Change the location to somewhere on OneDrive

Microchip Studio Tutorial

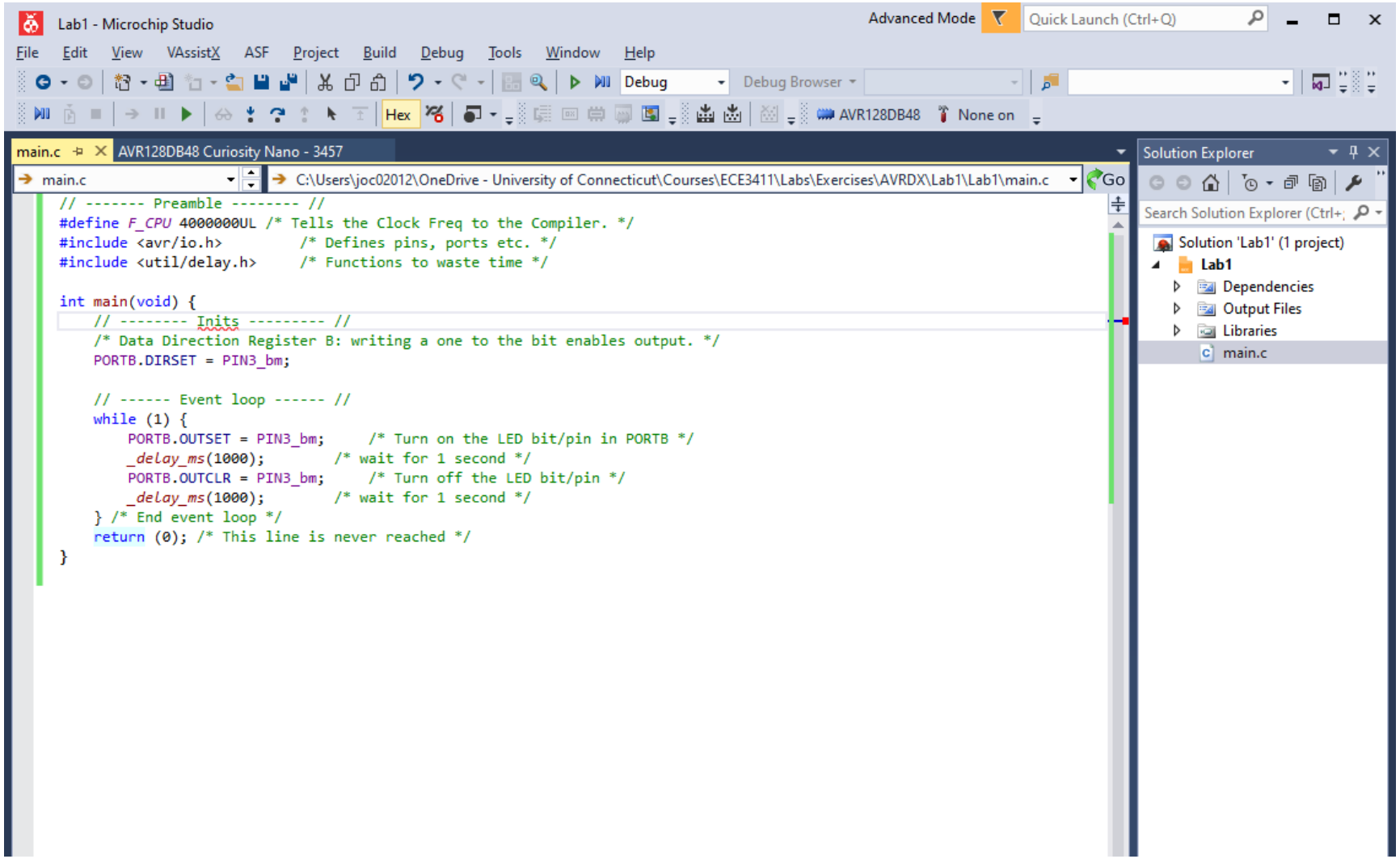
- Select the **AVR128DB48** device



Use the search box to find it faster

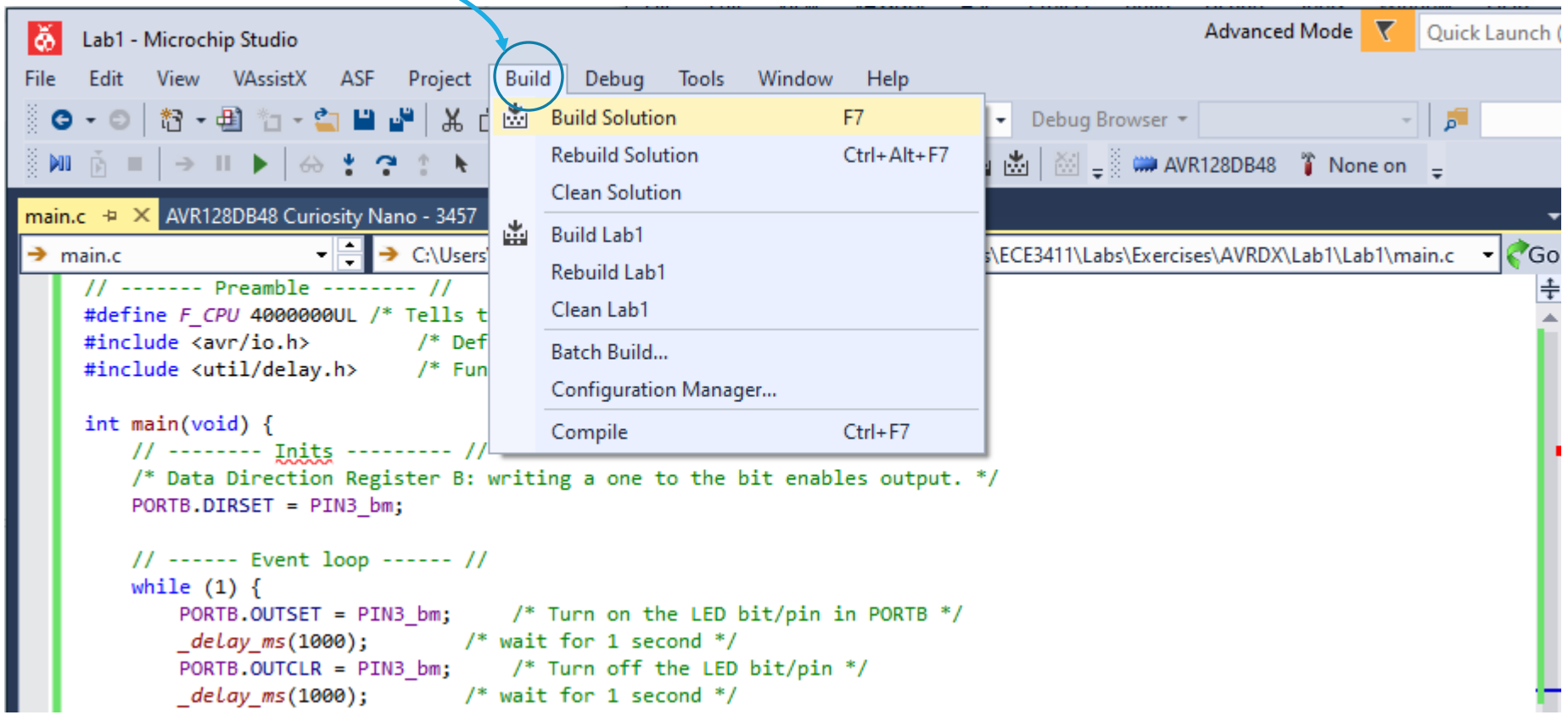
Microchip Studio Tutorial

- Copy test code into `main.c`



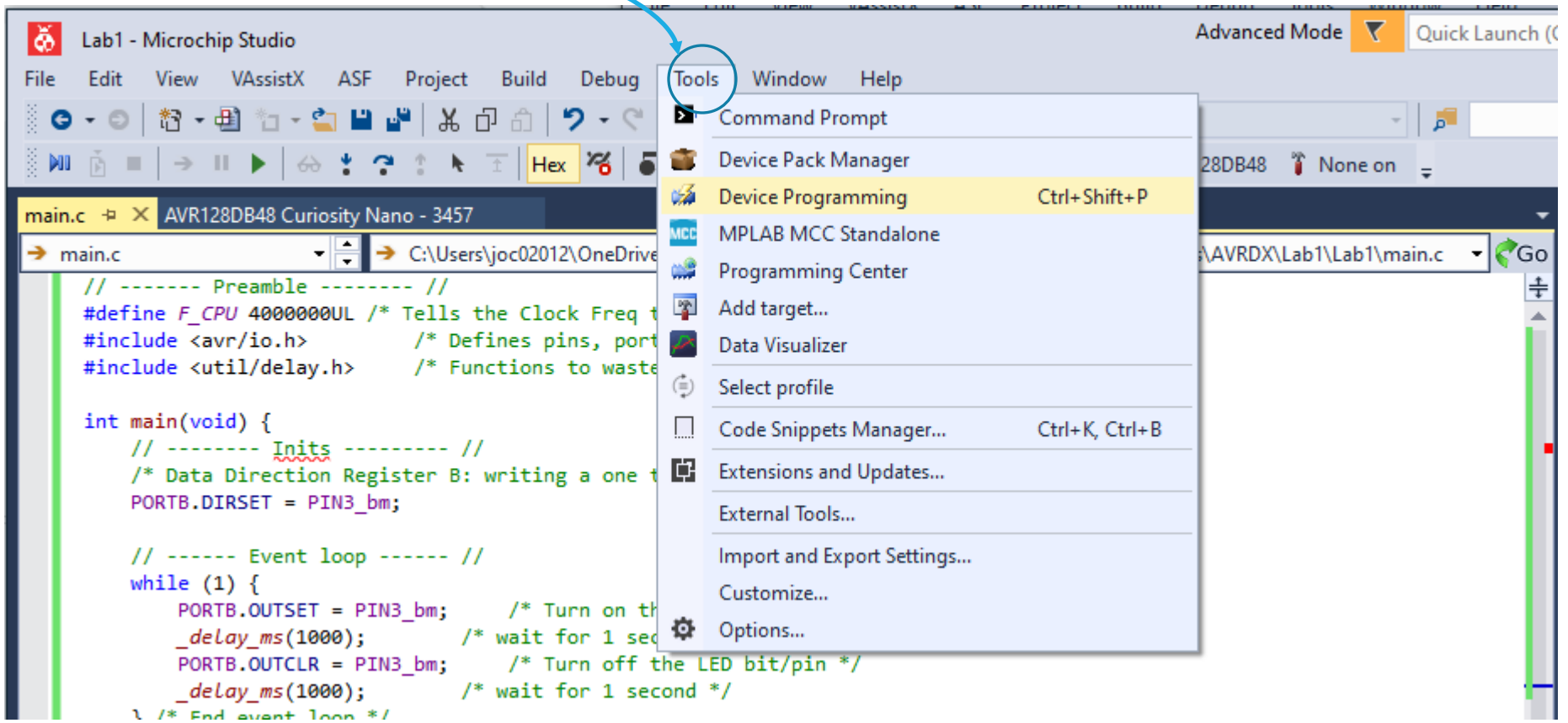
Microchip Studio Tutorial

- Build project



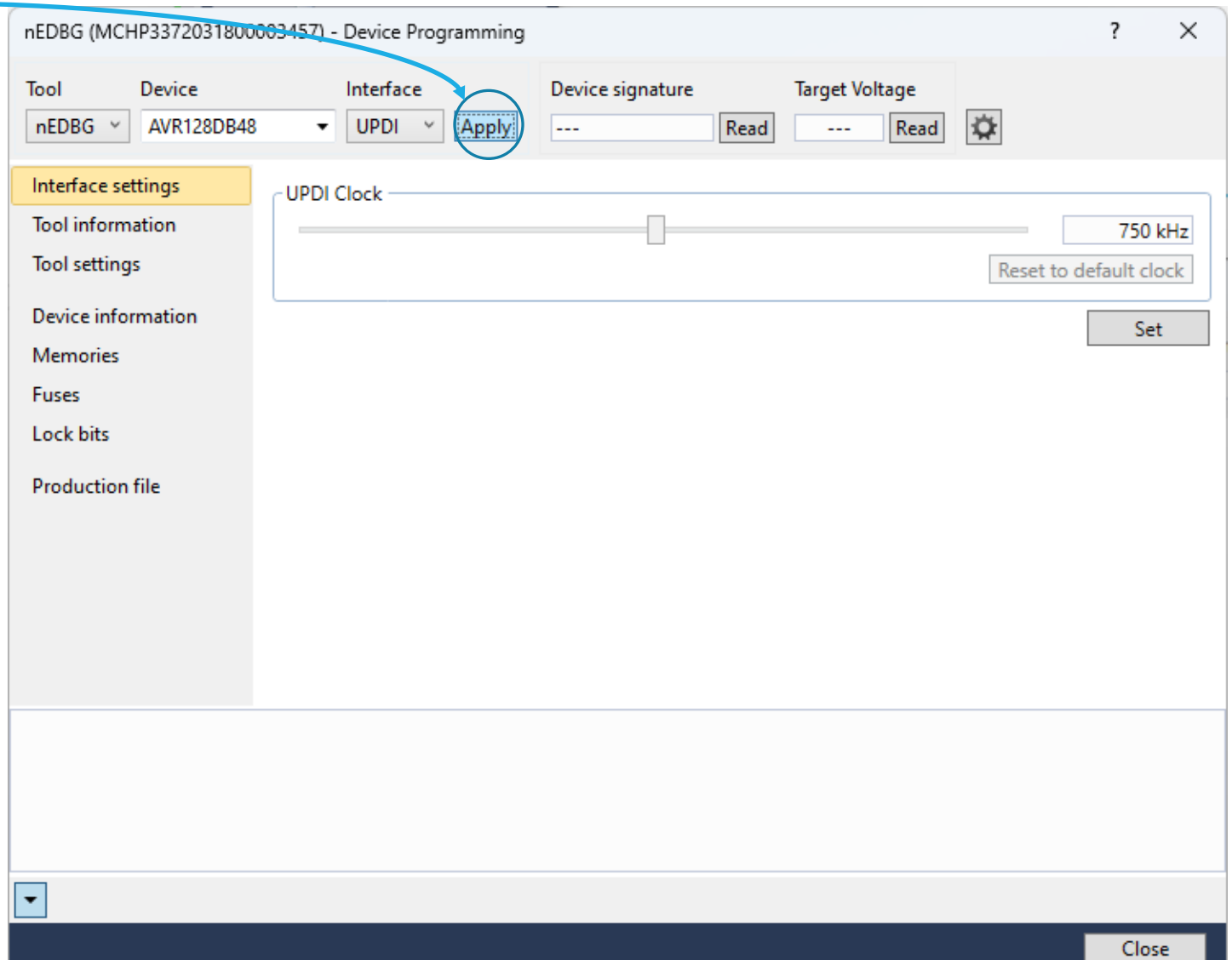
Microchip Studio Tutorial

- Program Board



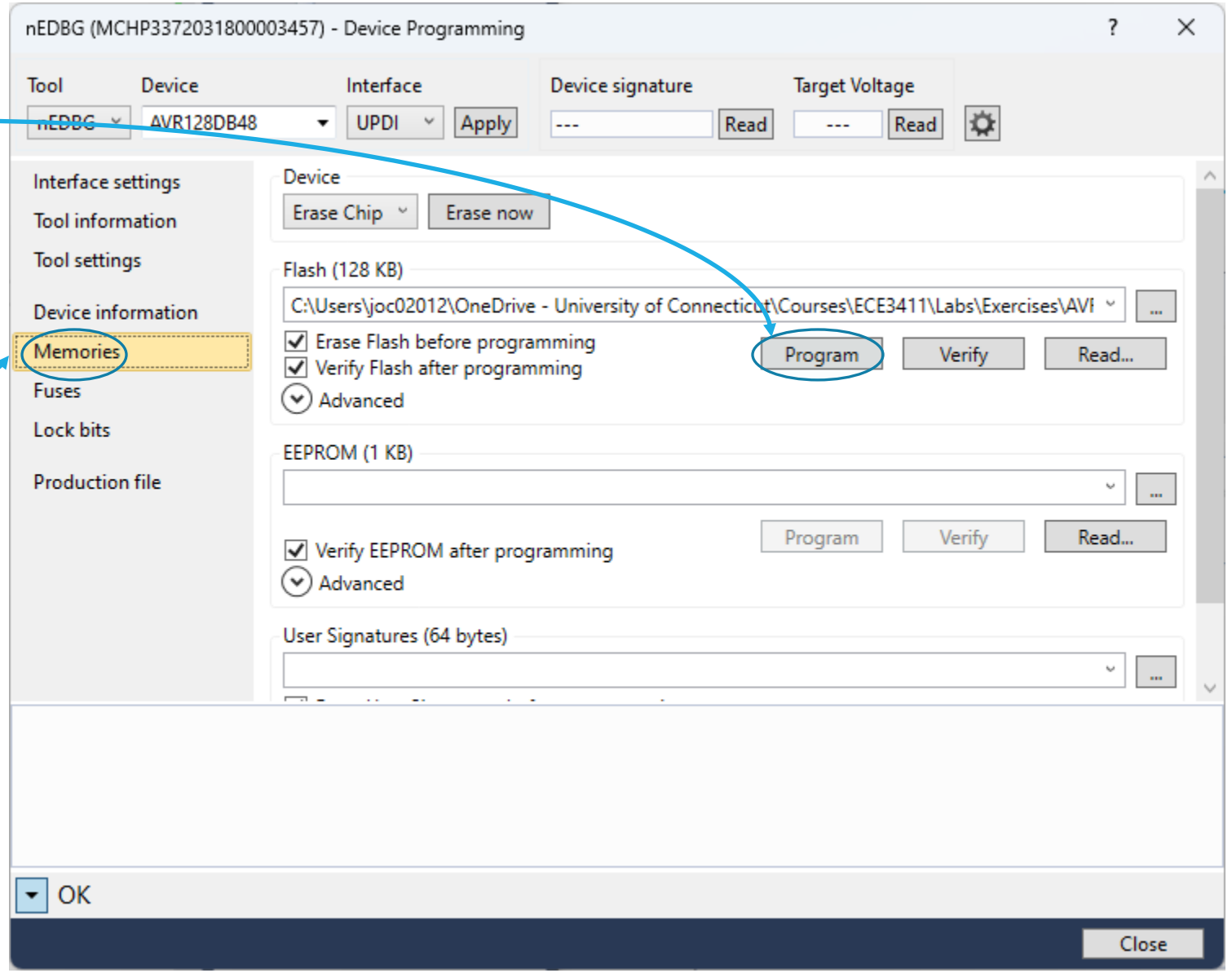
Microchip Studio Tutorial

- Click Apply



Microchip Studio Tutorial

- Click Memories and then Program

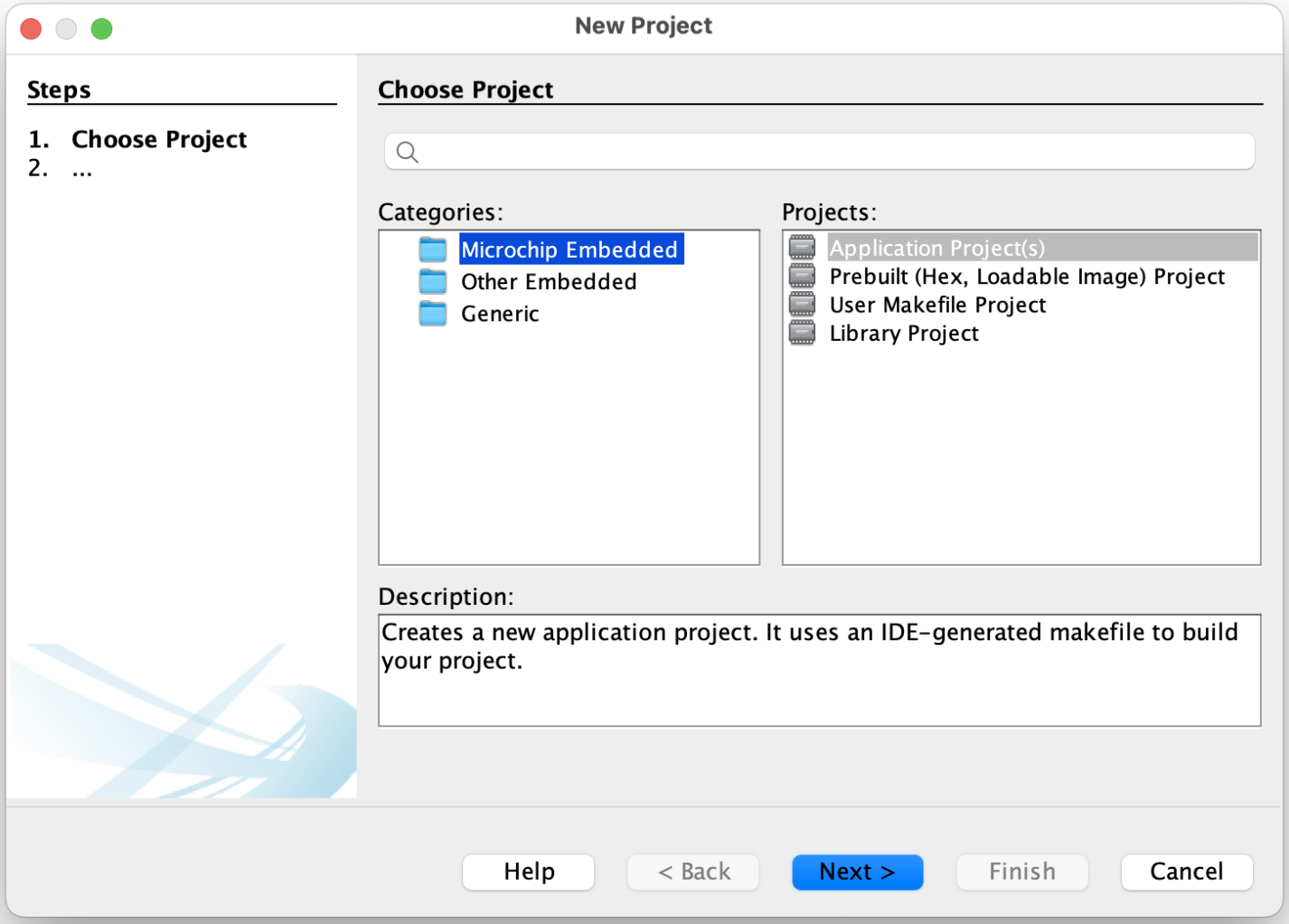


Initial board setup

- Setup MPLAB X IDE (if you are using a Mac)
  - MPLAB is available for download at: https://www.microchip.com/en-us/tools-resources/develop/mplab-x-ide
  - You will also need to download the XC8 compiler from: https://www.microchip.com/en-us/tools-resources/develop/mplab-xc-compilers/downloads-documentation#XC8

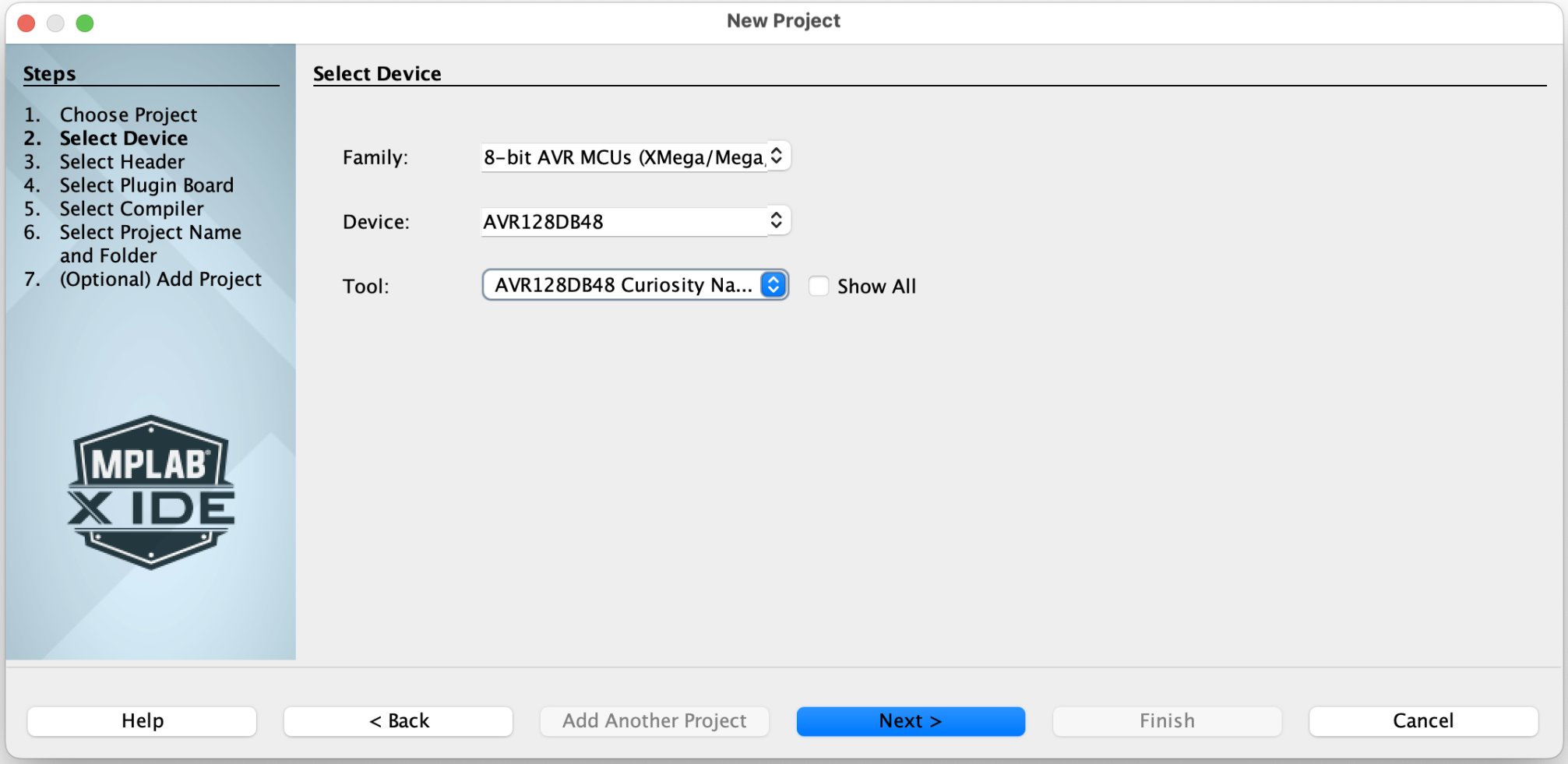
MPLAB Tutorial

- File->New Project ..



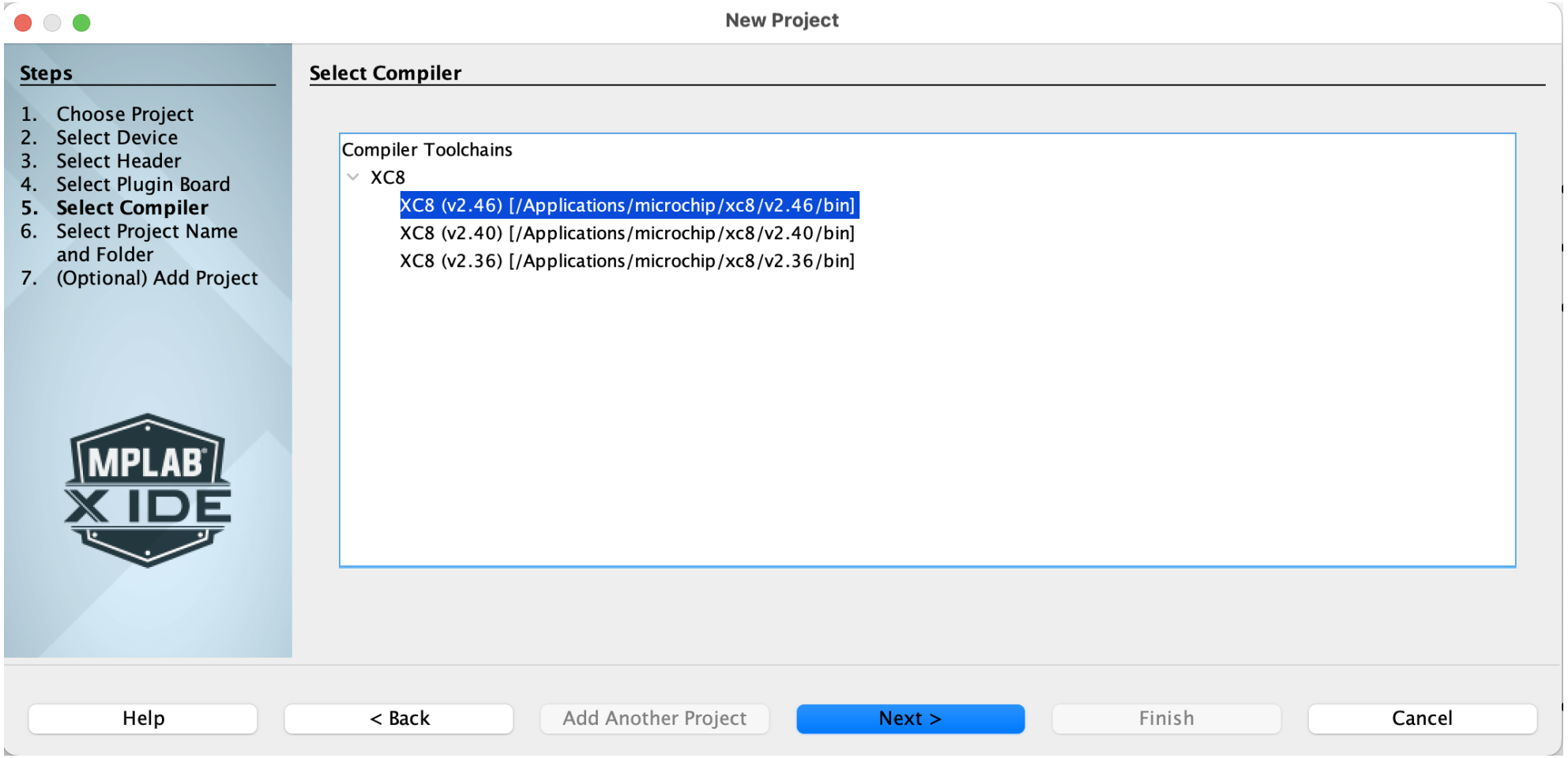
MPLAB Tutorial

- Select the **AVR128DB48 Curiosity Nano** tool. Make sure the board is plugged in.



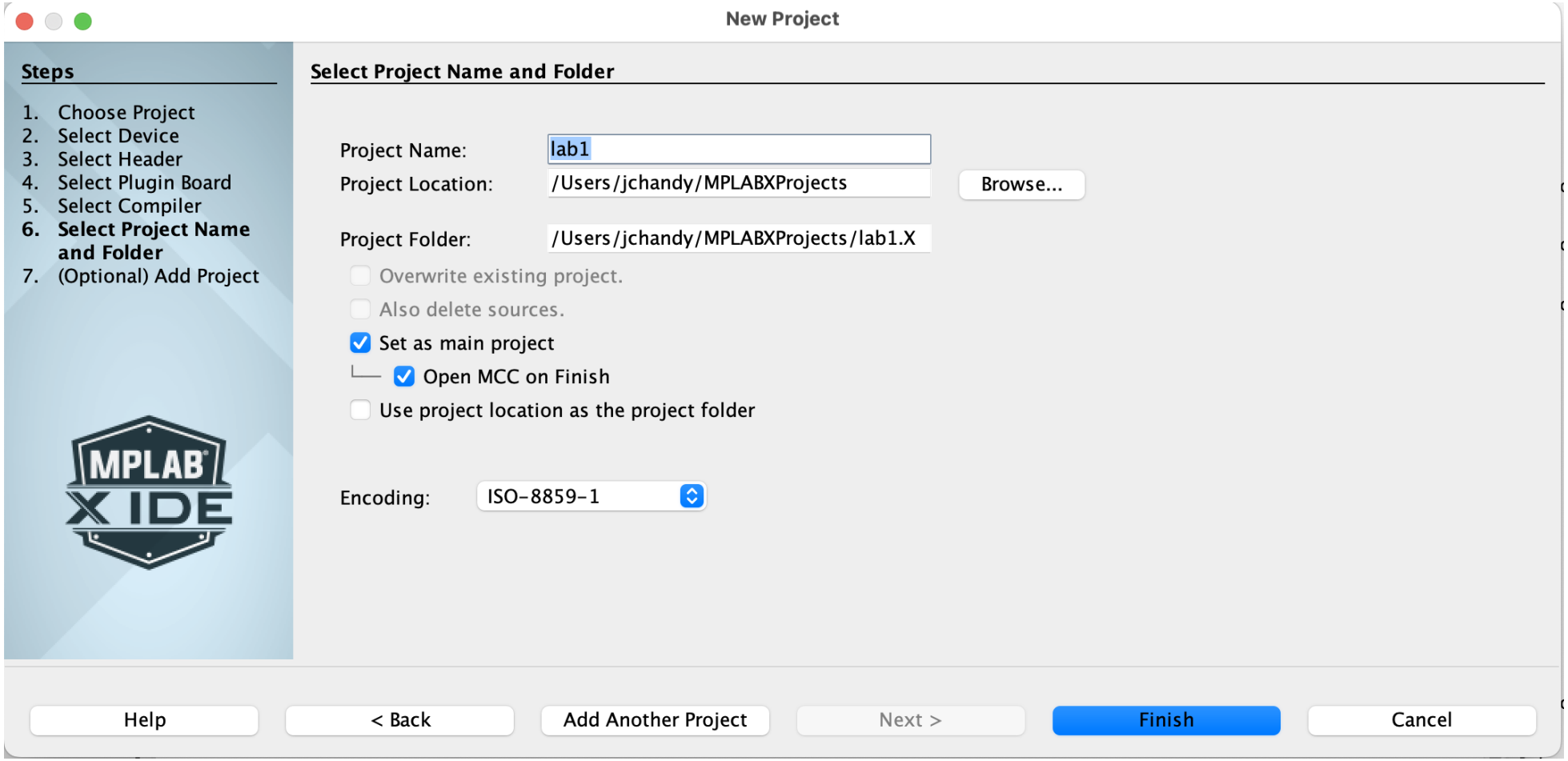
MPLAB Tutorial

- The XC8 compiler needs to be installed before MPLAB



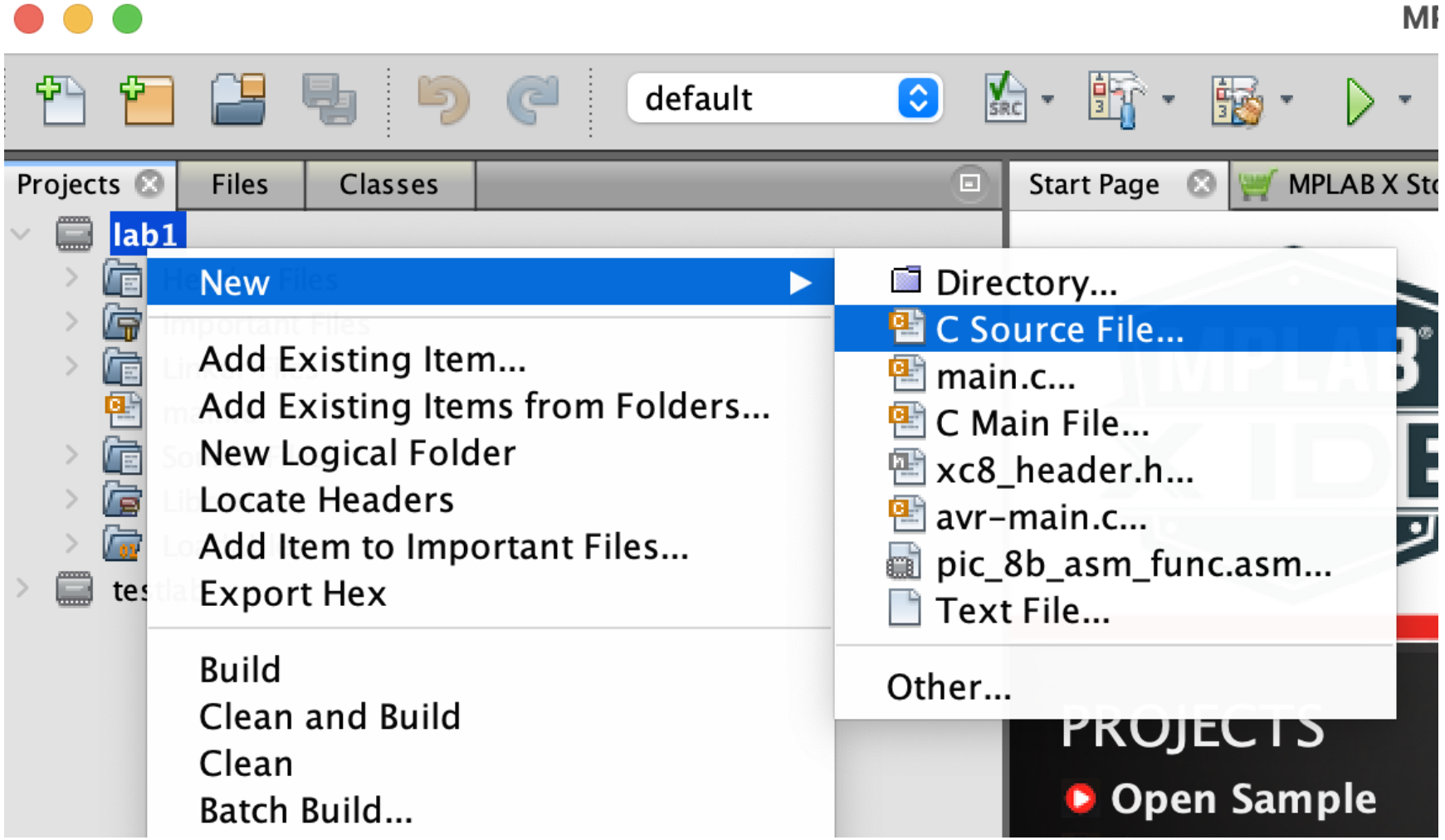
MPLAB Tutorial

- Set the Project Location to somewhere on your OneDrive

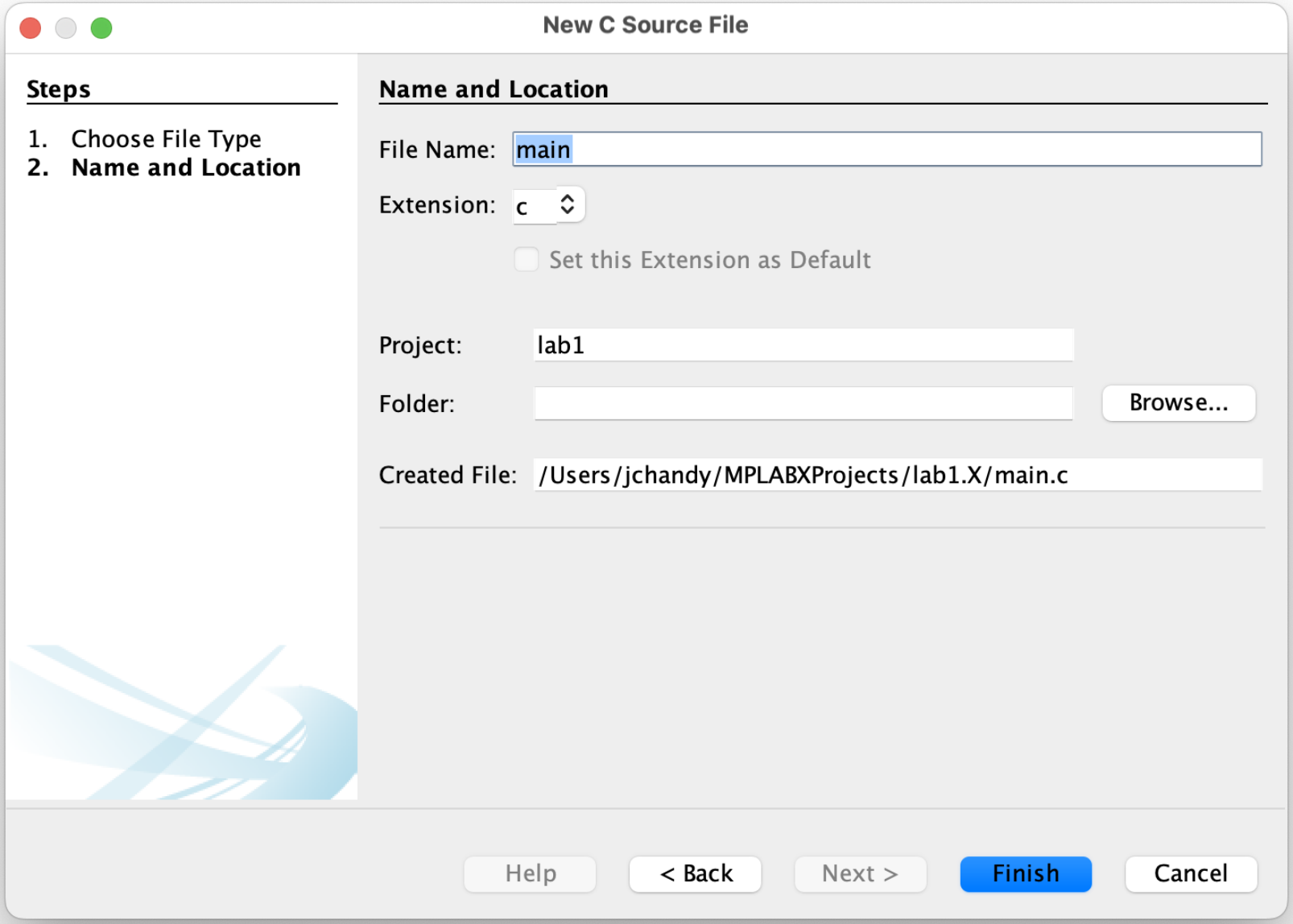


MPLAB Tutorial

- Right click on project name and choose New→C Source File…

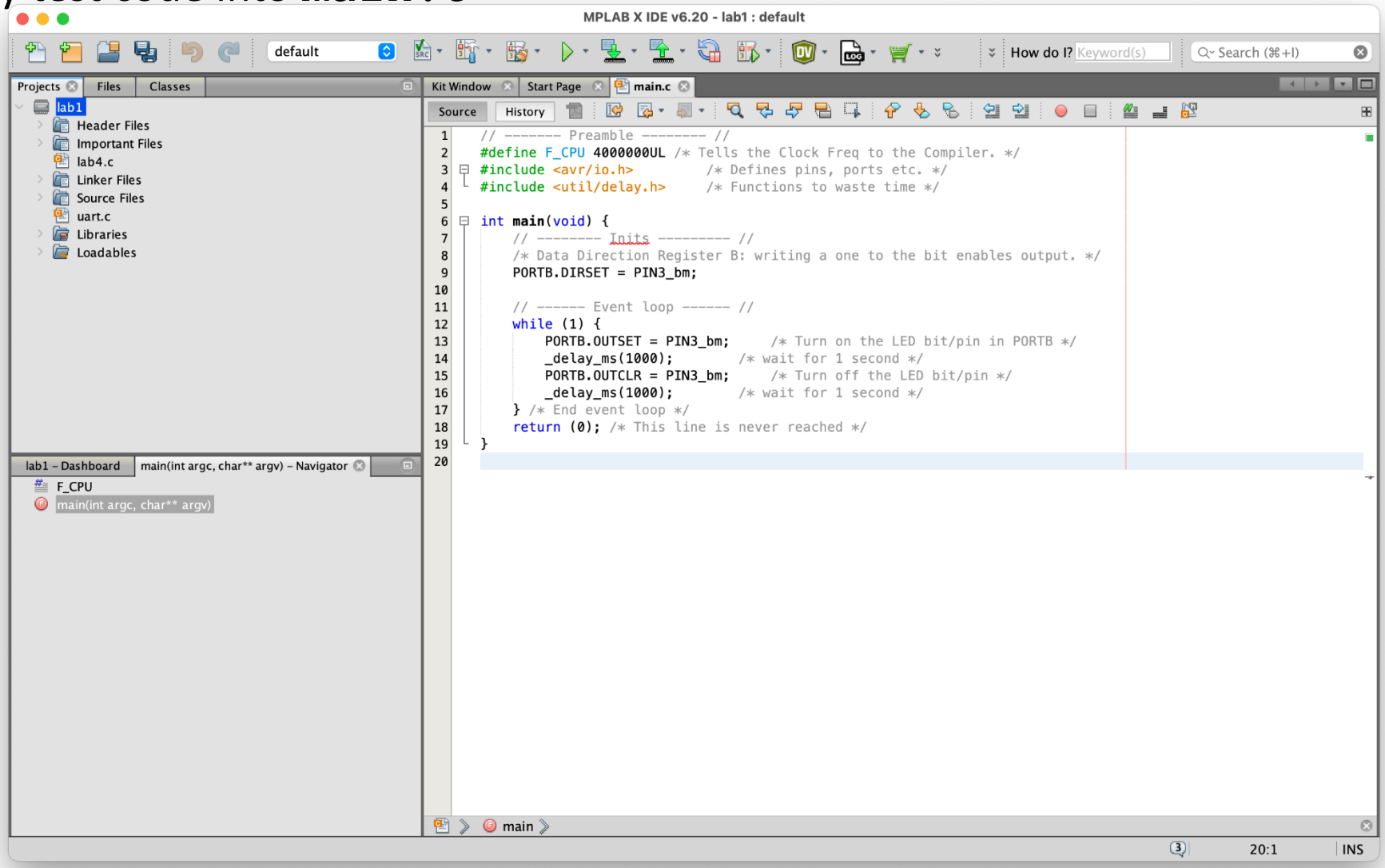


MPLAB Tutorial



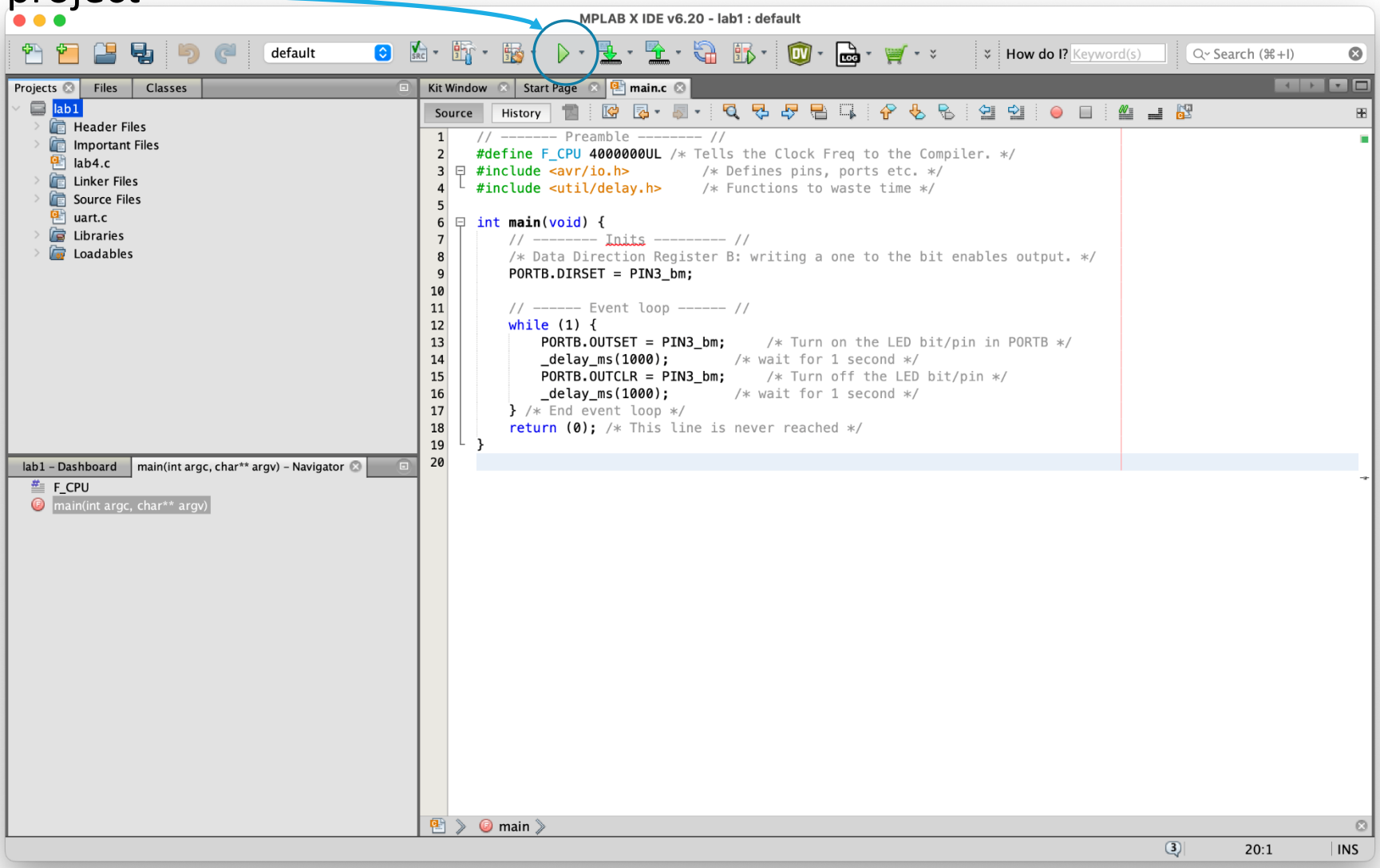
MPLAB Tutorial

- Copy test code into `main.c`



MPLAB Tutorial

- Run project



Comparison between Microchip Studio and MPLAB X IDE

Feature	Microchip Studio	MPLAB X IDE
Target MCU families	AVR, SAM	AVR, PIC, dsPIC, SAM
Toolchain	AVR-GCC, ARM-GCC	XC Compilers, AVR-GCC
Platform	Windows only	Cross-platform
UI Base	Visual Studio	NetBeans
Debug Tools	Atmel ICE, Dragon	PICkit, Snap, ICD
Active Development	❌ Deprecated	✅ Actively maintained

Comparison between MPLAB X IDE and XC8 Compiler

Feature	MPLAB X IDE	XC8 Compiler
Type	Integrated Development Environment (IDE)	C Compiler for 8-bit MCUs
Purpose	Code editing, project management, debugging	Code compilation
Supports	Multiple compilers (XC8, XC16, XC32, AVR-GCC)	Only 8-bit PIC and AVR MCUs
Platform	Windows, macOS, Linux	Integrated into MPLAB X IDE