ECE177: Programming I: From C...

Lab #1 — Software Setup

Week of 26 January 2026

1 Introduction

In this lab we will get your laptop set up with the coding and compiling infrastructure that you will be using for the rest of the semester.

We encourage you to work in groups on these labs, however each individual person will submit their own final result for grading.

2 Getting Compiler Setup

1. Install the software specific for your OS. Directions for this are found in a separate pdf file on the website. It's most likely what you want to use are the Windows instructions unless you are using a Mac or if you're a Linux hacker.

   (a) Windows

   https://web.eece.maine.edu/~vweaver/classes/ece177_2026s/labs/windows_instructions.pdf

   (b) MacOS

   https://web.eece.maine.edu/~vweaver/classes/ece177_2026s/labs/macos_instructions.pdf

   (c) Linux

   https://web.eece.maine.edu/~vweaver/classes/ece177_2026s/labs/linux_instructions.pdf

2. **Once you have this working, take a screenshot of the terminal! This will be uploaded as Result 1/6 of the Lab Assignment**

3 Getting VS Code Installed

1. Install the Raspberry Pi Pico VS Code Extension

   (a) Note: these instructions are based on those from Chapter 3.2 from the official "Getting Started with Raspberry Pi Pico" documentation that you can find here:
   `https://datasheets.raspberrypi.com/pico/getting-started-with-pico.pdf`

   (b) You can find the extension in the VS Code Extensions Marketplace.

      i. After starting VS Code, click on the "Extensions" Icon on the sidebar

      ii. Search for the Raspberry Pi Pico extension, published by Raspberry Pi.

      iii. Click the Install button to add it to VS Code.

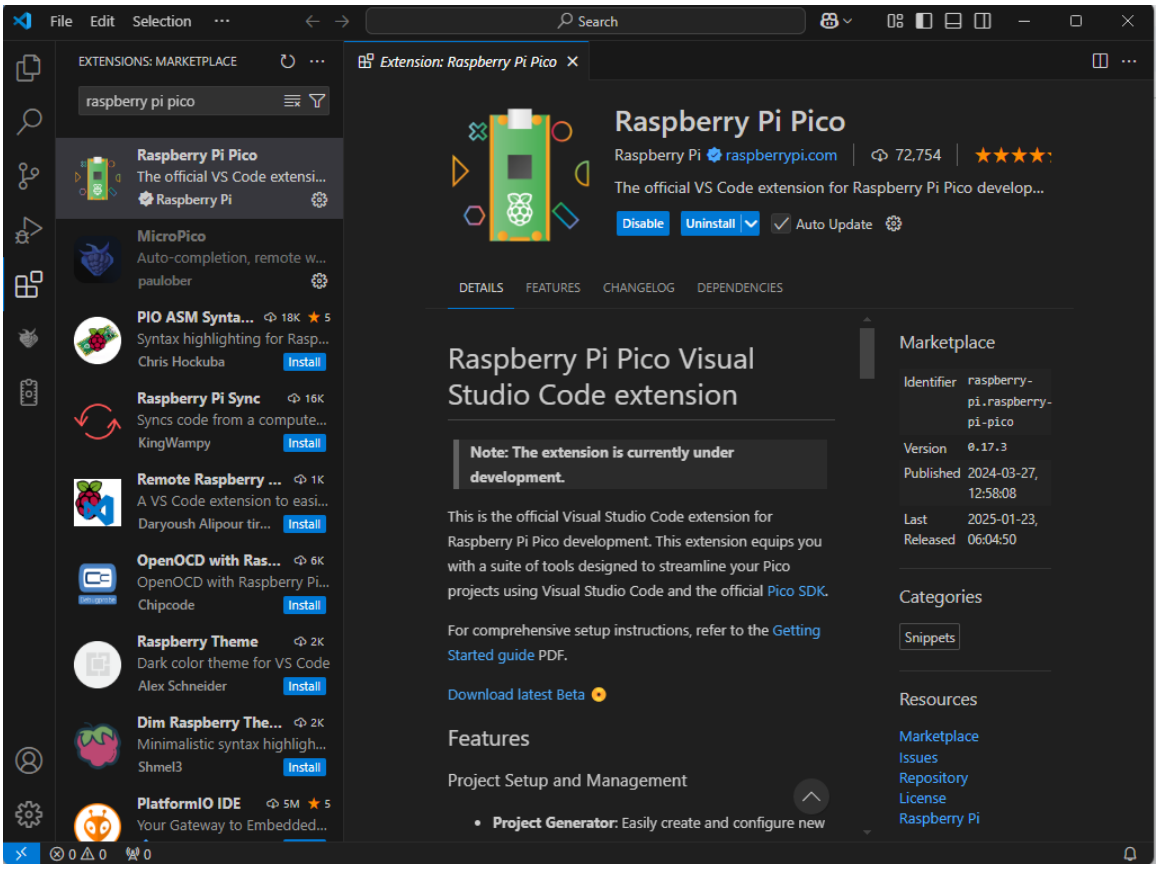

Figure 1: Example VS extension install.

(c) When installation completes the Activity sidebar should now have a Pi Pico icon with the label "Raspberry Pi Pico Project"

(d) **When done, take a screenshot of VS Code with the Pi Extension Visible! This is Result 2/6 for your Lab Grade**

2. Download the Lab1 Code from the website

(a) Create a new folder in `Documents` called `ece177`

(b) Create a new folder under `ece177` called `labs`

(c) Download `lab01_code.zip` file from the ECE177 labs website
`https://web.eece.maine.edu/~vweaver/classes/ece177_2026s/labs/lab01_code.zip`

(d) Unzip `lab01_code.zip` in `Documents/ece177/labs/`
(this will create a `lab01_code` folder)

(e) Import `lab01_code` in VS Code using the Pico Extension (see Figure 2 for what this might look like). *NOTE* import (select) the "lab01_code" directory, not the files inside.

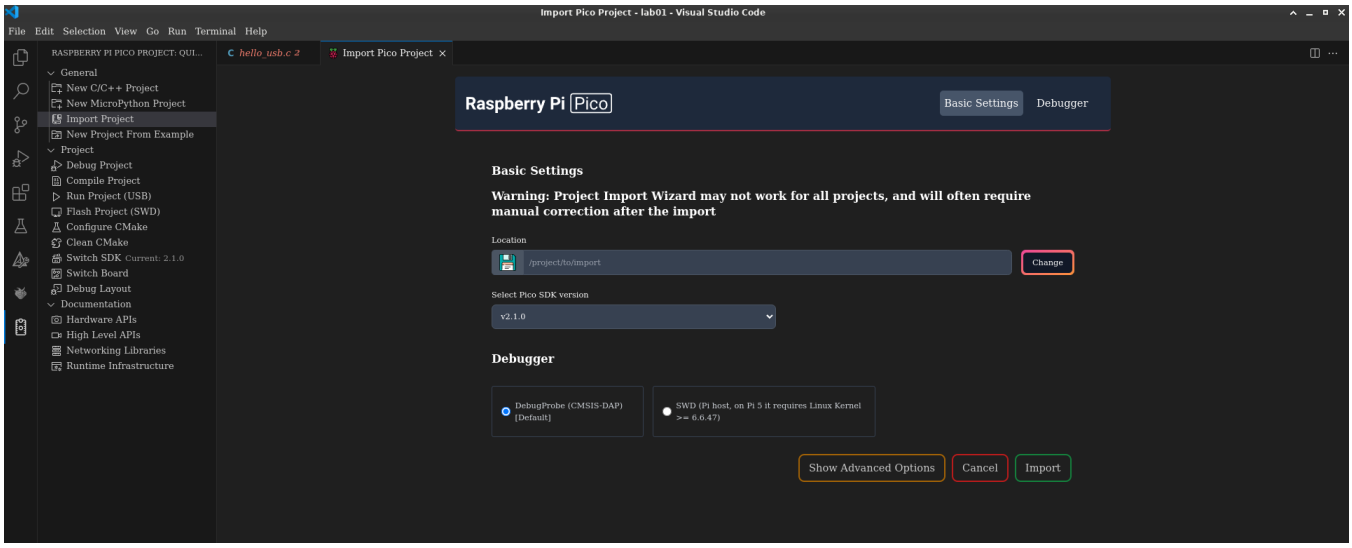

Figure 2: Importing example.

(f) **It may take a bit to install everything the first time as it downloads the toolchain**

(g) If prompted, accept and trust the authors of all files in the ‘labs’ folder

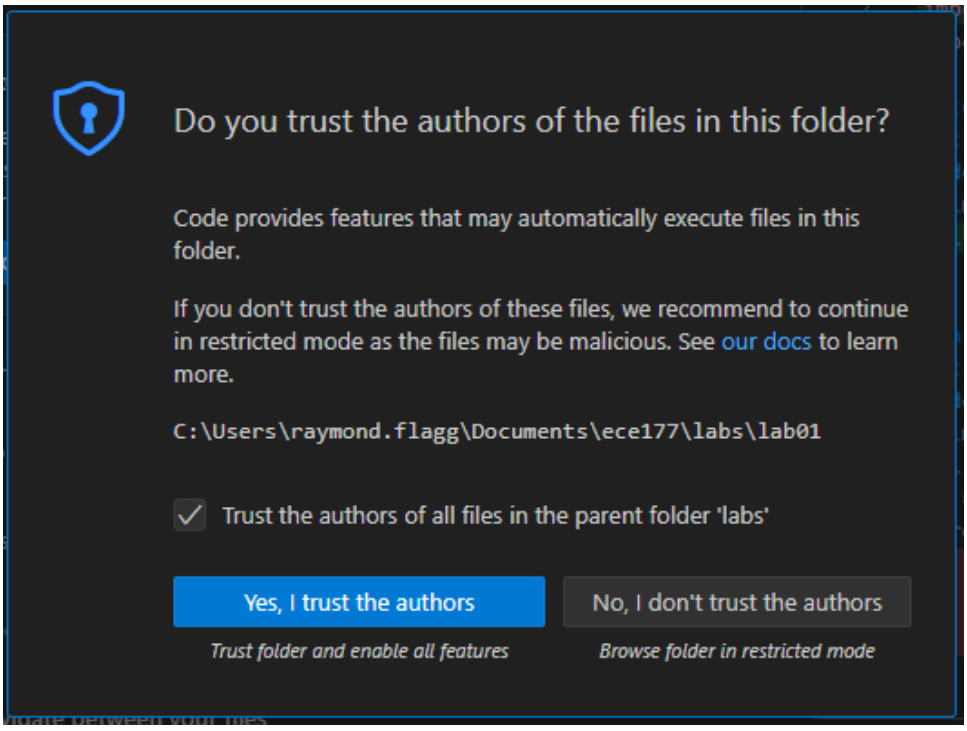

Figure 3: Dialog asking you to trust it.

4 Test the Pico by compiling and running lab01

1. Use VS Code to build and upload the sample code
2. If it's working correctly, the LED should blink and a lot of lines of `-1` will be written to the serial port.
3. If running on Windows, Verify this serial port output by running PuTTY or `screen` (on Mac or Linux use `screen`, the platform specific documents from earlier might have better instructions on getting that going)
4. Directions for installing and configuring PuTTY are separate and can be found here:
   `https://web.eece.maine.edu/~vweaver/classes/ece177/labs/configure_putty.pdf`
5. Running `screen` involves something like:

   ```
   sudo screen DEVICE 115200
   ```

   (Note: `DEVICE` is likely either `/dev/ttyACM0` or `/dev/ttyS0`
6. TAs will verify programming a Pico if your device hasn't been handed out yet.
7. **Take a screenshot of PuTTY or screen showing the output! This is part 3/6 of your Lab Grade**

5 Write a local C Program

1. This describes testing out the C compiler on your laptop using `nano` and `gcc`.
   (a) If on Windows, first open Ubuntu on Windows
   (b) To create a C file called "hello.c" use the following command:

   ```
   nano hello.c
   ```

   (c) The result should look like this:

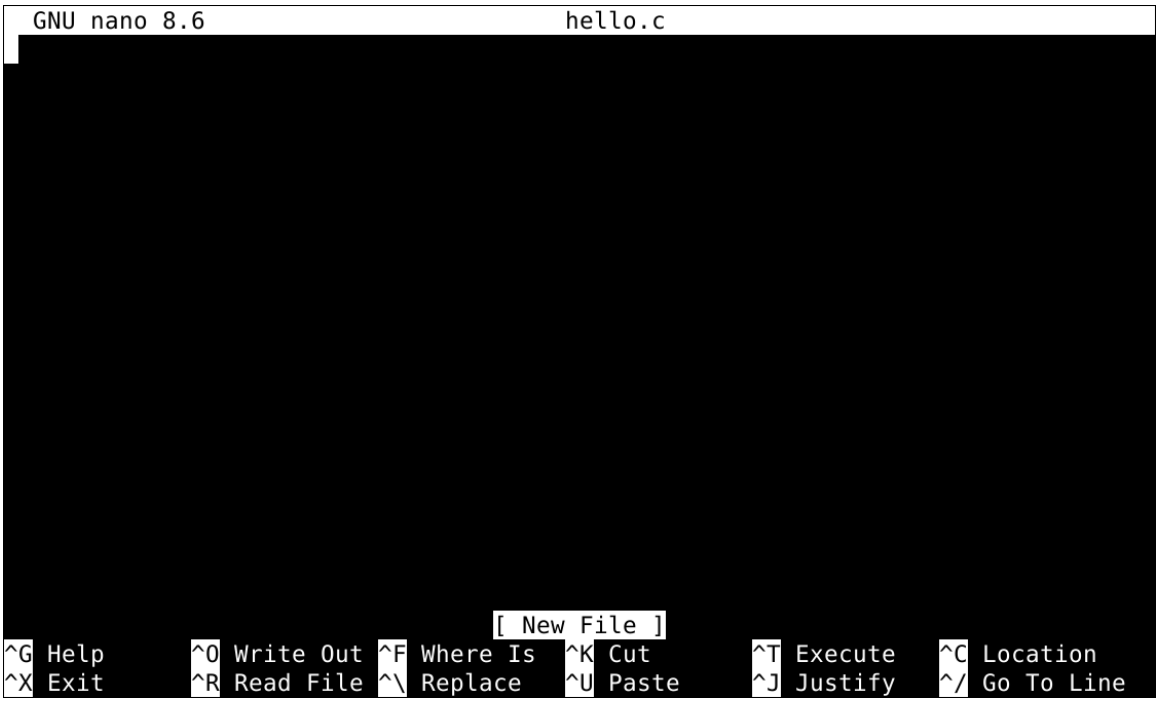

Figure 4: Nano text editor.

(d) Type the following into the editor:

```
/* Sample Hello World Program */

#include <stdio.h>

int main(int argc, char **argv) {

        printf("Hello World!\n");

        return 0;

}
```

(e) It is always important to comment your code! In this example the text between the /* ... */ is a comment and ignored by the compiler. You can use comments to describe your code.

(f) Update the comments in this example to include your name, the date, and a description of the program.

(g) Take a screenshot of your finished program inside of `nano`! **This is Part 4/6 of your Lab Assignment**

(h) Once you have entered the code you need to save it, which you do by pressing Control-O (the bar at the bottom of the screen has a list of common things you might do, save is ^O which means press Control-O)

(i) Now you will want to exit back to the command prompt by pressing Control-X

(j) Now you will want to compile the code. Type the following command on the command line (not in `nano`):

```
gcc -Wall -o hello hello.c
```

(k) NOTE: this command compiles the `hello.c` file using the GCC compiler that we previously installed. The `-Wall` says to print all warnings. The `-o hello` tells gcc to give the resulting executable program the name "hello".

(l) If the compiler gives you any errors you will want to re-run nano on the file and double check that you typed everything in correctly and try again.

(m) **Take a Screenshot of the window with your code compilation in it. This is part 5/6 of your Lab Assignment**

(n) In order to run the executable file use the following command:

```
./hello
```

The output should look like this:

```
$ ./hello
Hello World!
```

(o) **Take a Screenshot of the window with your code running. This is part 6/6 of your Lab Assignment**

6 Submitting the Assignment

- Be sure to upload the following items to BrightSpace:
  (you might have to navigate to assessments/asssignments to find the lab assignments)

  1. A screenshot of WSL (terminal/shell for OSX or Linux) running on your laptop.
  2. A screenshot of VS Code with the Raspberry Pi Pico Extension installed.
  3. A screenshot of `screen` or PuTTY with the output of the programmed Pico.
  4. A screenshot of the "Hello World" code you wrote (in Section 5) on your laptop
  5. A screenshot of compiling (to completion) the "Hello World" code you wrote on your laptop.
  6. A screenshot of the program output after executing.

- TAs will ask you to demonstrate the lab requirements and will enter grades after the screenshots have been uploaded.

- TAs: Enter grades only after all six screenshots have been uploaded and the assignment has been submitted.