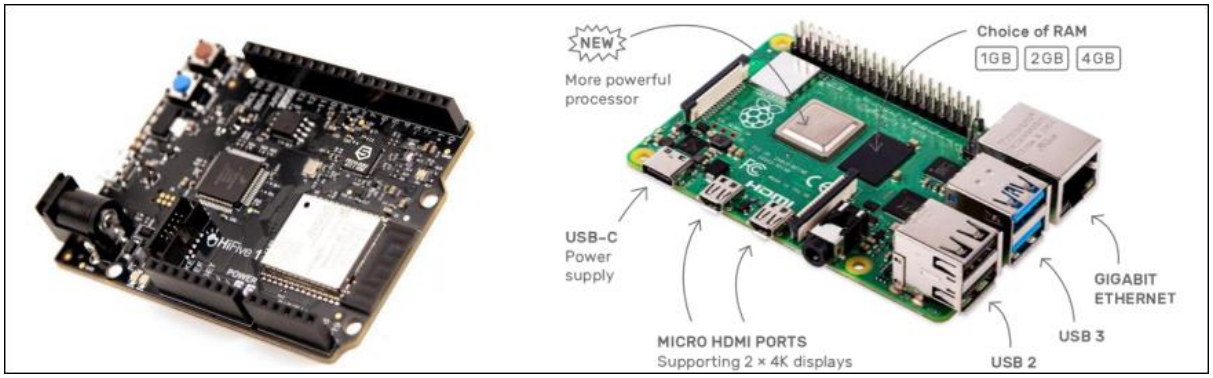
EECS 388 Lab #2

Introduction to Hardware and Development Environment

In this semester, you will develop parts of a self-driving car using small, embedded computers, sensors, and actuators. In the process, you will learn the fundamental concepts and practical skills to design and implement an embedded system.

Hardware Platforms

We will use two embedded single-board computer (SBC) platforms, shown below.



(a)HiFive1 Microcontroller (b) Raspberry Pi 4

The first is an Arduino-compatible SBC featuring a RISC-V architecture microcontroller called HiFive1, which will be responsible for basic control and safety of the car, and the second is a Raspberry Pi 4, which will be responsible for vision-based steering using deep learning. In the first half of the semester, we will use HiFive1 while in the second half of the semester, we will use both platforms. More detailed technical specs can be found in the following links. We will provide additional details of the hardware platforms when necessary for the labs in the future.

SiFive HiFive 1 rev b: https://www.sifive.com/boards/hifive1-rev-b
Raspberry Pi 4: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

Task 1 : Setup development environment – 20 points

For software development on the microcontroller, we will use Visual Studio Code (VSCode) and PlatformIO IDE combination. VSCode is already installed on your computer but you will need to install the PlatformIO IDE and other extensions.

(Note that the following installation instructions are based on the PlatformIO IDE for VSCode documentation at: https://docs.platformio.org/en/latest/ide/vscode.html#installation)

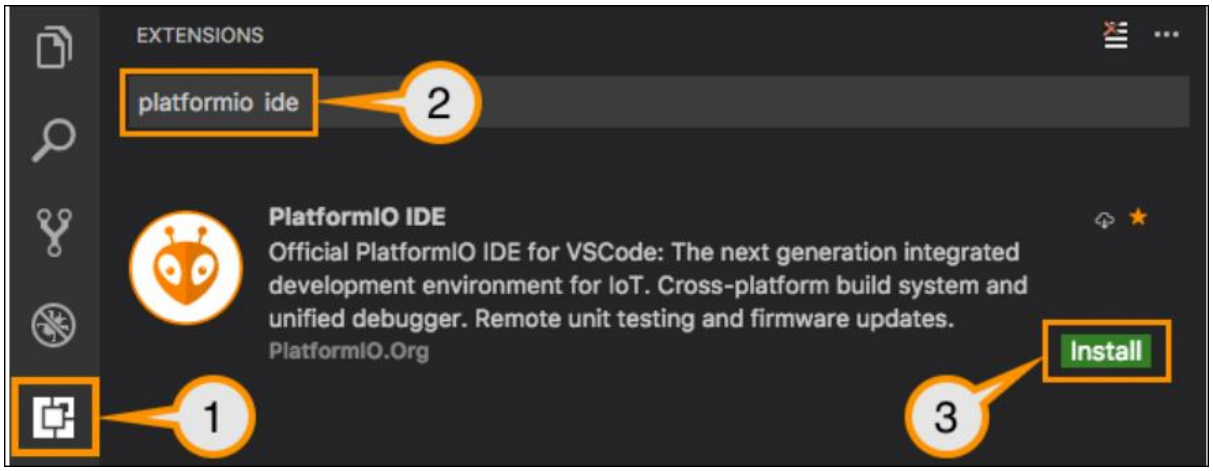
Task 1.1: Take a look at Visual Studio Code

Launch the Visual Studio Code program from the command line as follows.

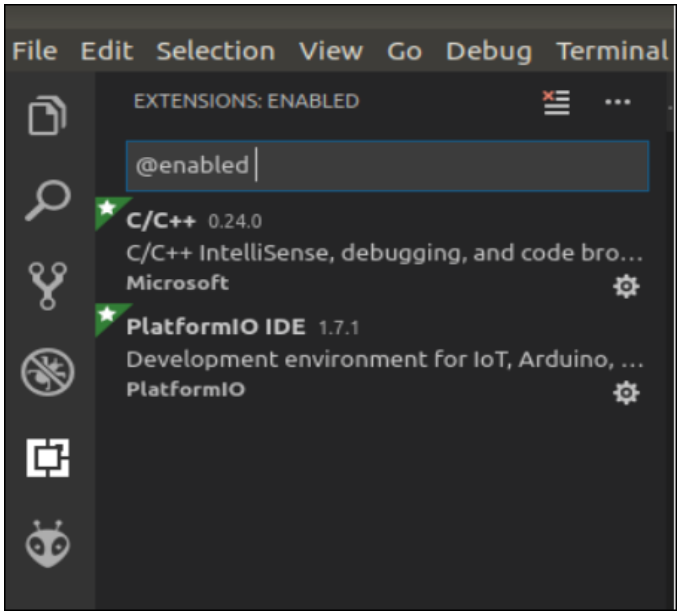
$ code

Task 1.2: Install PlatformIO extensions Next, Install the PlatformIO IDE extension for VSCode as follows.

1. Open VSCode Package Manager
2. Search for official “platformio ide” extension
3. Install PlatformIO IDE

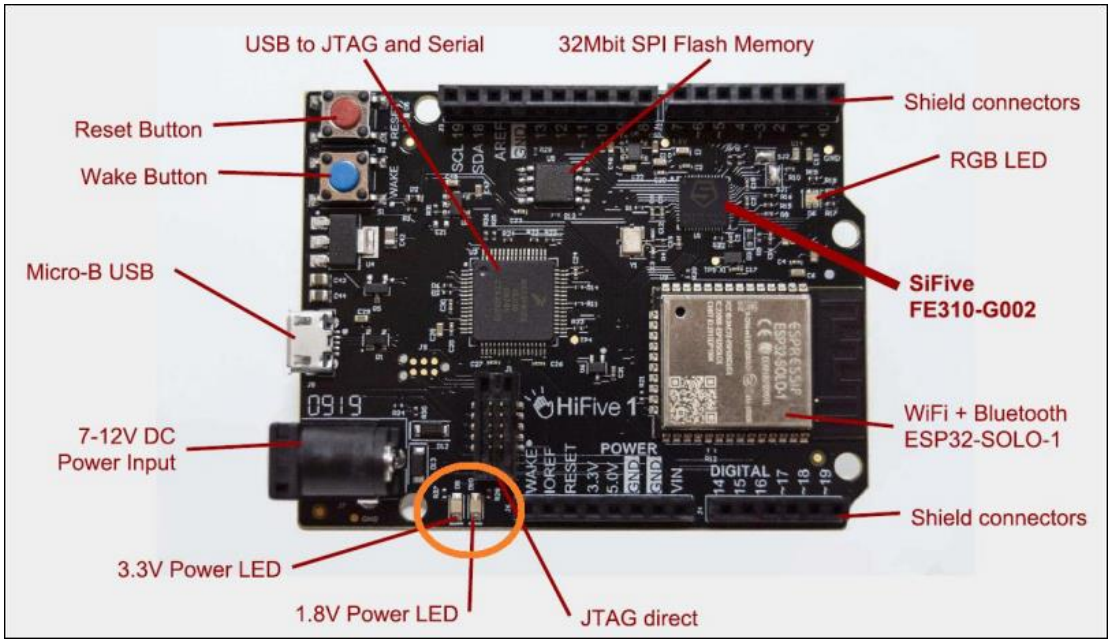


Install PlatformIO IDE After the installation is completed, check if you have both ‘PlatformIO IDE’ and ‘C/C++’ extensions installed as follows. Optionally, installing ‘vscode-icons’ and ‘vscode-pdf’ extensions are also recommended.



Task 1.3. Connect the board to your PC

Next, connect your board to one of your PC's USB ports. Once the board is connected to the PC, two yellow power LEDs (3.3V and 1.8V) should be turned on. (See the figure below).



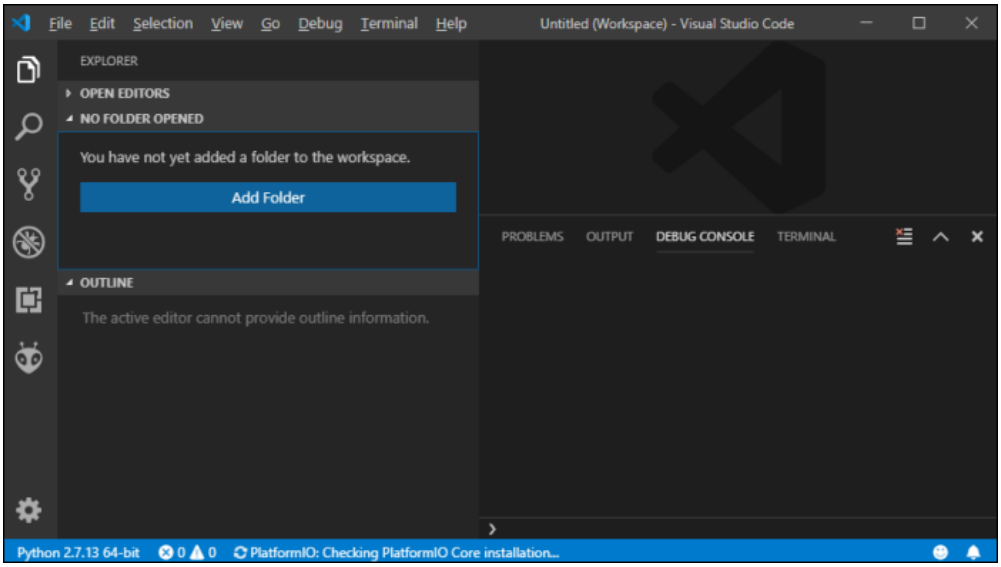
Task 1.4. Run your first program on the HiFive1 board

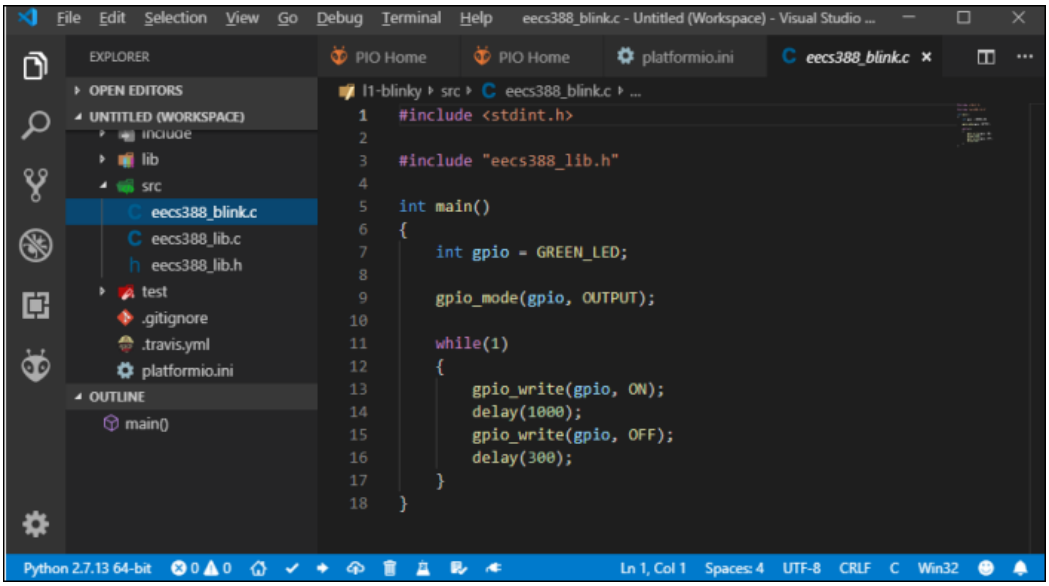
Setup a project and download the sample project as follows.

1) $ mkdir -p ~/Documents/PlatformIO
2) $ cd ~/Documents/PlatformIO
3) Download zip from canvas (l**ab02.tar.gz**) as follows
    a. Sign into the Canvas account and go to the EEC 388 Embedded Systems section.
    b. Inside the "Files" section, go to "Lab Material" -> "Source Codes" folder.

c. Click on the l**ab02.tar.gz** file to download the source code for lab 2.
d. Extract the file inside a folder on your PC.
    i. You can extract using the File explorer by right clicking, OR
    ii. Using the terminal, use this command: **`tar -xzvf lab02.tar.gz`**

4) **Add project to the folder:** Add the lab02 folder in VSCode.



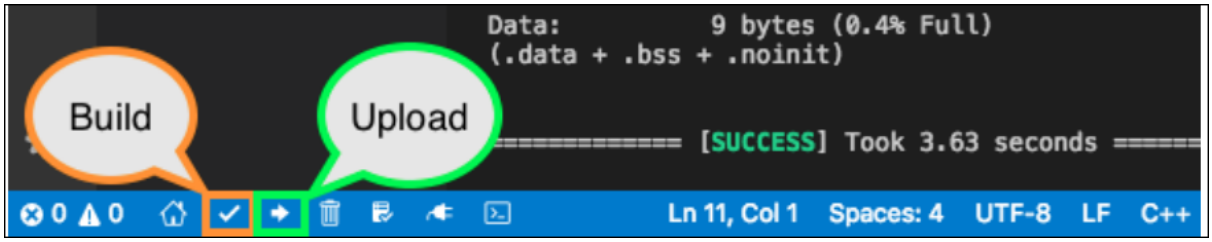


You should be able to see the screen above.

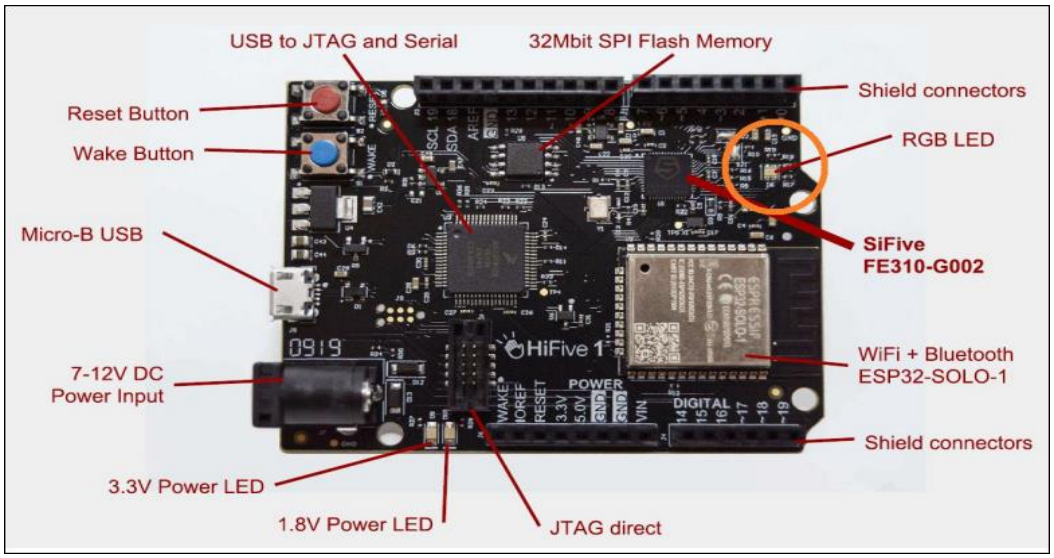
5) **Build and Deploy :** You are now ready to build the code and deploy it on the target board.

First build the program by clicking the build button in the toolbar as shown below or with ctrl+alt+b hotkey.

If it is successful, you can now upload the compiled program binary to the board by clicking the upload button or with ctrl+alt+u hotkey.

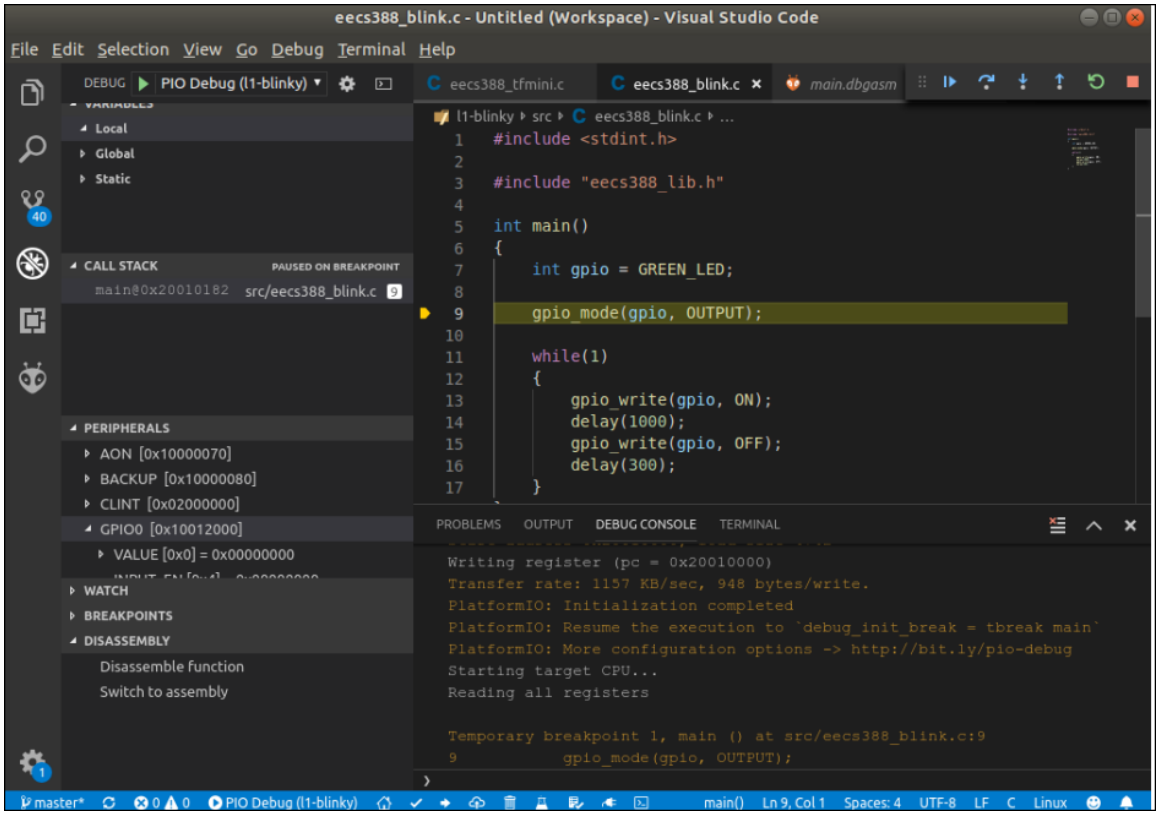


If it was successful, you should see the green led on the board is blinking. (orange circle in the figure below)



Task 1.5. Debugging (Optional)

The HiFive1 board is already equipped with a hardware debugger support. Thus, you can utilize PlatformIO + VSCode's debugging capability to debug your code. To use debugging, find the 'Debug' menu from the pull-down menu or hit 'F5' (or 'Ctrl+F5'). You should be able to see something like the following screen. Then, you can use the debugger toolbar (right top corner of the screenshot) to navigate the code.



Task 2: RGB Blinky – 80 points

The code is the same as the work in **Task1**, but includes additional information on what you need to do to blink RGB LEDs instead of just the green led.

Follow the instructions in the code base to implement Task 2.

Task 2.1. Review the EECS388 library

Review the EECS388 library header (eecs388_lib.h) and the implementation (eecs388_lib.c) as per instruction.

Task 2.2. Implement RGB blinky

Follow the instructions in the code and modify the code (eecs388_blink.c) to blink RGB LEDs.

Task 2.3 (OPTIONAL) - 10 additional points

Modify your code so that it also shows white light in the loop. Red -> Blue -> Green -> white

Submission

Go to the "Assignments" section of the Canvas and submit your modified **eecs388_blink.c** file to the Lab02 submission link for your specific lab section.
Also, make sure that you have shown your completed work to your respective GTA for a demo.

Appendix

PlatformIO Key bindings

- ctrl+alt+b / cmd-shift-b / ctrl-shift-b Build Project
- cmd-shift-d / ctrl-shift-d Debug project
- ctrl+alt+u Upload Firmware
- ctrl+alt+s Open Serial Port Monitor

PlatformIO Toolbar

1) PlatformIO Home
2) PlatformIO: Build
3) PlatformIO: Upload
4) PIO Remote
5) PlatformIO: Clean
6) PIO Unit Testing
7) Run a task… (See “Task Runner” below)
8) Serial Port Monitor
9) PIO Terminal

PlatformIO documentation

https://docs.platformio.org/en/latest/integration/ide/vscode.html#installation