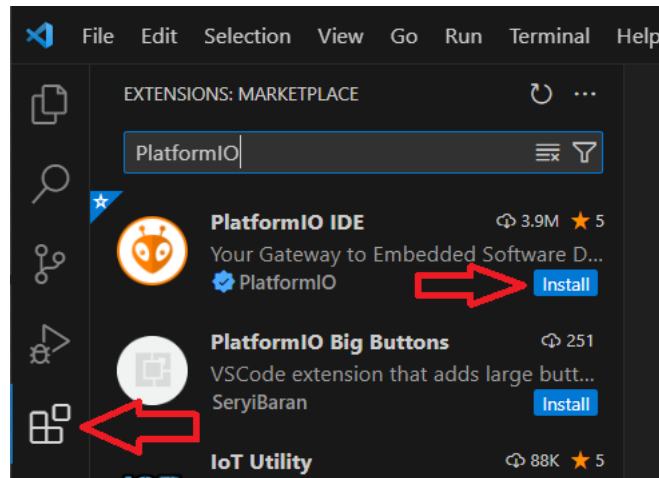



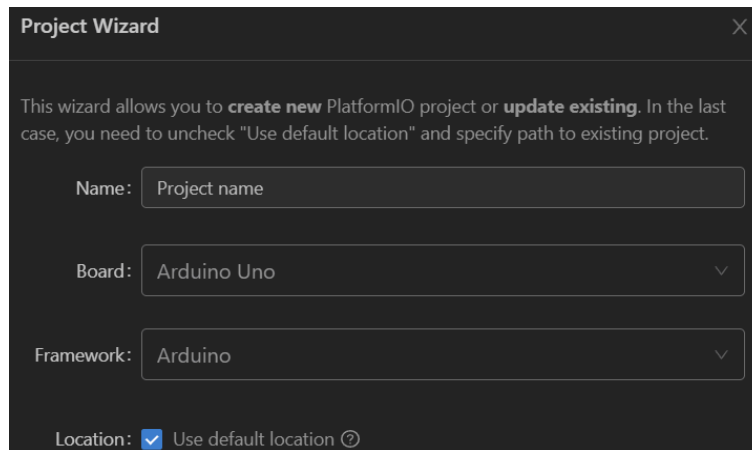
PlatformIO IDE Setup Guide for ECE 484

This guide has been proven to work for Windows 11, [ADD MAC OS HERE]

1. If you don't have it already, install the latest version of Visual Studio Code [here](#).
2. Open Visual Studio Code, Click the extensions menu, search for **PlatformIO** and click install.
 - i. Documentation on PlatformIO can be found on their website [here](#).
 - ii. It should take about minute to finish installing "PlatformIO Core"



3. Open the PlatformIO homepage by clicking “” and selecting **Open** found on the side dropdown menu under Quick Access > PIO Home > Open.
 - a. Click the “+ New Project” button to start a new project.
 - b. Enter in a **Project Name**, **Board**, and **Framework**
 - i. Unless you are using a different device than the syllabus states (Arduino UNO R3), it should look like this.



4. It should automatically generate a file structure that looks like this.

- i. PlatformIO supports both C and C++ development for Atmel AVR devices (Arduino UNO R3's ATmega328P)
- ii. *To continue programming in C++ skip the project setup here and begin programming in the **main.cpp** file.*

b. To begin programming in C, open the **platformio.ini** and

remove the line: `framework = arduino`

- i. We do this because the arduino framework defaults to using C++ and the `#include <Arduino.h>` APIs.

c. Next, rename the **main.cpp** to **[filename].c**

d. From here you can `#include <avr/io.h>`

e. Useful Links:

1. <https://community.platformio.org/t/use-c-instead-of-c/26768>
2. <https://community.platformio.org/t/does-it-let-us-select-a-c-compiler-for-avr/15583/3>

5. To test our setup, we can use the following **blink_led.c** code from [here](#).

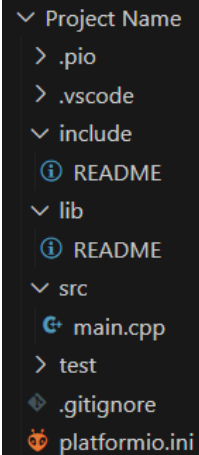
```
#include <avr/io.h>
#include <util/delay.h>
#define MS_DELAY 3000

int main(void)
{
    /*Set to one the fifth bit of DDRB to one
    **Set digital pin 13 to output mode */
    DDRB |= _BV(DDB5);
    while (1)
    {
        /*Set to one the fifth bit of PORTB to one
        **Set to HIGH the pin 13 */
        PORTB |= _BV(PORTB5);

        /*Wait 3000 ms */
        _delay_ms(MS_DELAY);

        /*Set to zero the fifth bit of PORTB
        **Set to LOW the pin 13 */
        PORTB &= ~_BV(PORTB5);

        /*Wait 3000 ms */
        _delay_ms(MS_DELAY);
    }
}
```

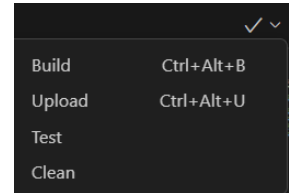


6. Finally, to upload to the program to the device:

- a. Make sure the Arduino is connected to the computer via USB cable.
- b. Click the checkmark at the top right of the Visual Studio Code window to **Build**.

i. This will compile all the files inside the src directory.

- c. Click the dropdown open then select **Upload**.



i. For parts b & c if it is the first time building the project it may take a minute or two as PlatformIO installs the necessary components. (avr-gcc, avrdude, etc.)

- ii. If it cannot find the device go back to the PIO Home, under **Devices** there is a refresh option for serial devices.

- d. For quick links on the bottom toolbar:

