Imperial College
London

ESP32:
Vscode installation
&
connection to SQL database

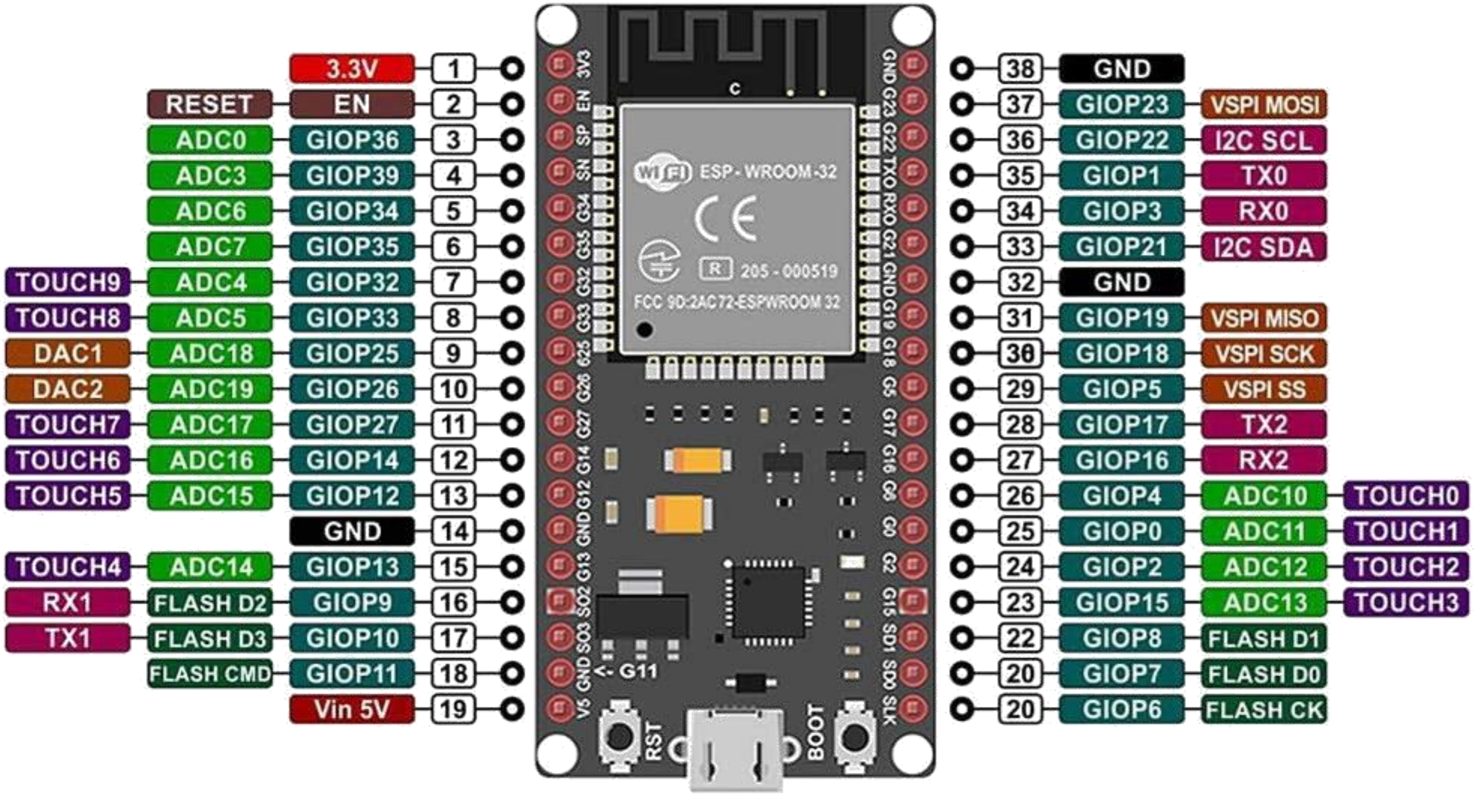
Mars Rover 2022

Adam Bouchaala
Teaching Fellow
Imperial College London

Imperial College London

VS code installation guide

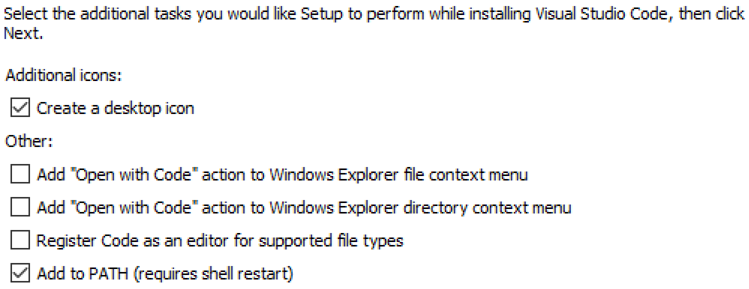
ESP32 Architecture



Imperial College London

VS Code and PlatformIO IDE for ESP32

- The Arduino IDE works great for small applications.
- However, for advanced projects with more than 200 lines of code, multiple files, and other advanced features like auto completion and error checking, VS Code with the PlatformIO IDE extension is the best alternative.
- Go to https://code.visualstudio.com/ and download the stable build for your operating system (Windows, Mac OS X, Linux Ubuntu).



Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☑ Create a desktop icon

Other:

□ Add "Open with Code" action to Windows Explorer file context menu

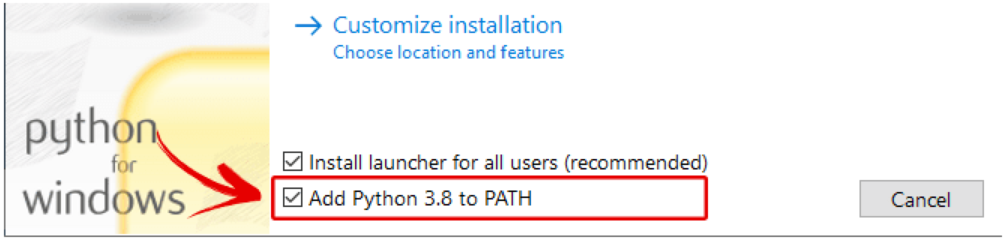
□ Add "Open with Code" action to Windows Explorer directory context menu

□ Register Code as an editor for supported file types

☑ Add to PATH (requires shell restart)

VS Code and PlatformIO IDE for ESP32

- **Installing Python on Windows**
- **To program the ESP32 and ESP8266 boards with PlatformIO IDE you need Python 3.5 or higher installed on your computer. We're using Python 3.8.5.**
- **Go to python.org/download and download Python 3.8.5 or the newest version.**

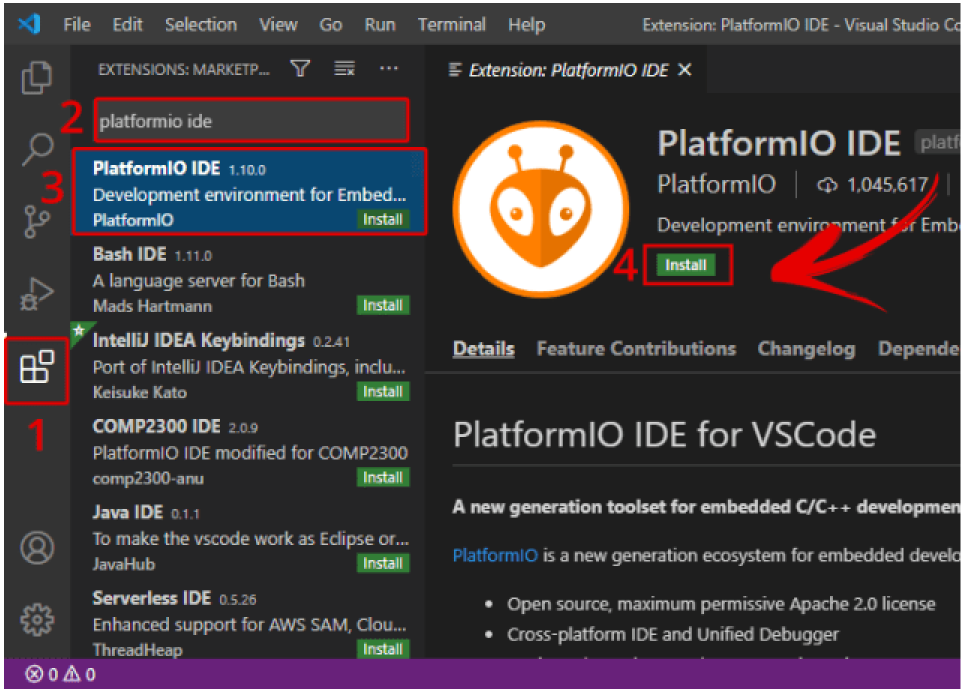


→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.8 to PATH

Installing PlatformIO IDE Extension

1. **Click on the Extensions icon or press Ctrl+Shift+X to open the Extensions tab**
2. **Search for “PlatformIO IDE”**
3. **Select the first option**
4. **Finally, click the Install button (Note: the installation may take a few minutes)**



platformio ide

PlatformIO IDE 1.10.0
Development environment for Embed...
PlatformIO

Bash IDE 1.11.0
A language server for Bash
Mads Hartmann

IntelliJ IDEA Keybindings 0.2.41
Port of IntelliJ IDEA Keybindings, inclu...
Keisuke Kato

COMP2300 IDE 2.0.9
PlatformIO IDE modified for COMP2300
comp2300-anu

Java IDE 0.1.1
To make the vscode work as Eclipse or...
JavaHub

Serverless IDE 0.5.26
Enhanced support for AWS SAM, Clou...
ThreadHeap

PlatformIO IDE
PlatformIO | 1,045,617
Development environment for Emb

PlatformIO IDE for VSCode

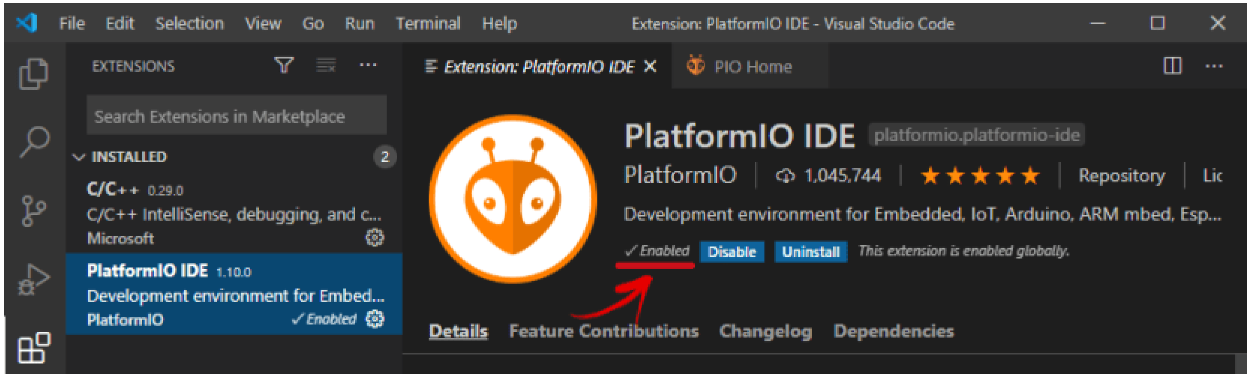
A new generation toolset for embedded C/C++ developmen

PlatformIO is a new generation ecosystem for embedded develo

- Open source, maximum permissive Apache 2.0 license
- Cross-platform IDE and Unified Debugger

Installing PlatformIO IDE Extension

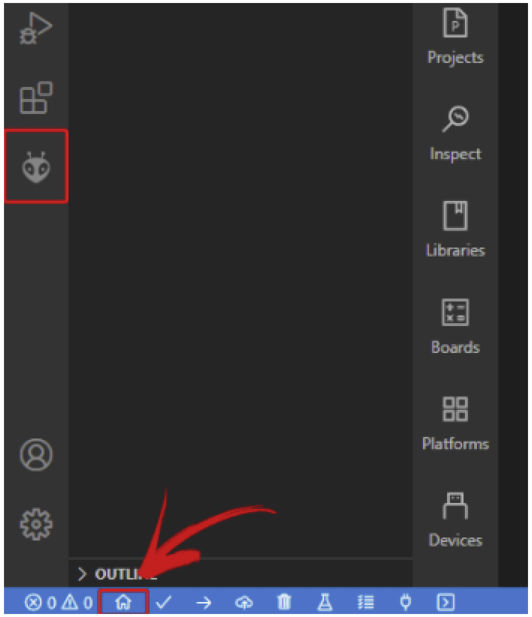
- After installing, make sure that PlatformIO IDE extension is enabled as shown below.

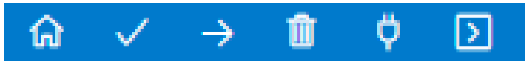


File
Edit
Selection
View
Go
Run
Terminal
Help
Extension: PlatformIO IDE - Visual Studio Code
EXTENSIONS
Search Extensions in Marketplace
INSTALLED
2
C/C++ 0.29.0
C/C++ IntelliSense, debugging, and c...
Microsoft
PlatformIO IDE 1.10.0
Development environment for Embed...
PlatformIO
✓ Enabled
Extension: PlatformIO IDE
PIO Home
PlatformIO IDE
platformio.platformio-ide
PlatformIO
1,045,744
★★★★★
Repository
Lic
Development environment for Embedded, IoT, Arduino, ARM mbed, Esp...
✓ Enabled
Disable
Uninstall
This extension is enabled globally.
Details
Feature Contributions
Changelog
Dependencies

Installing PlatformIO IDE Extension

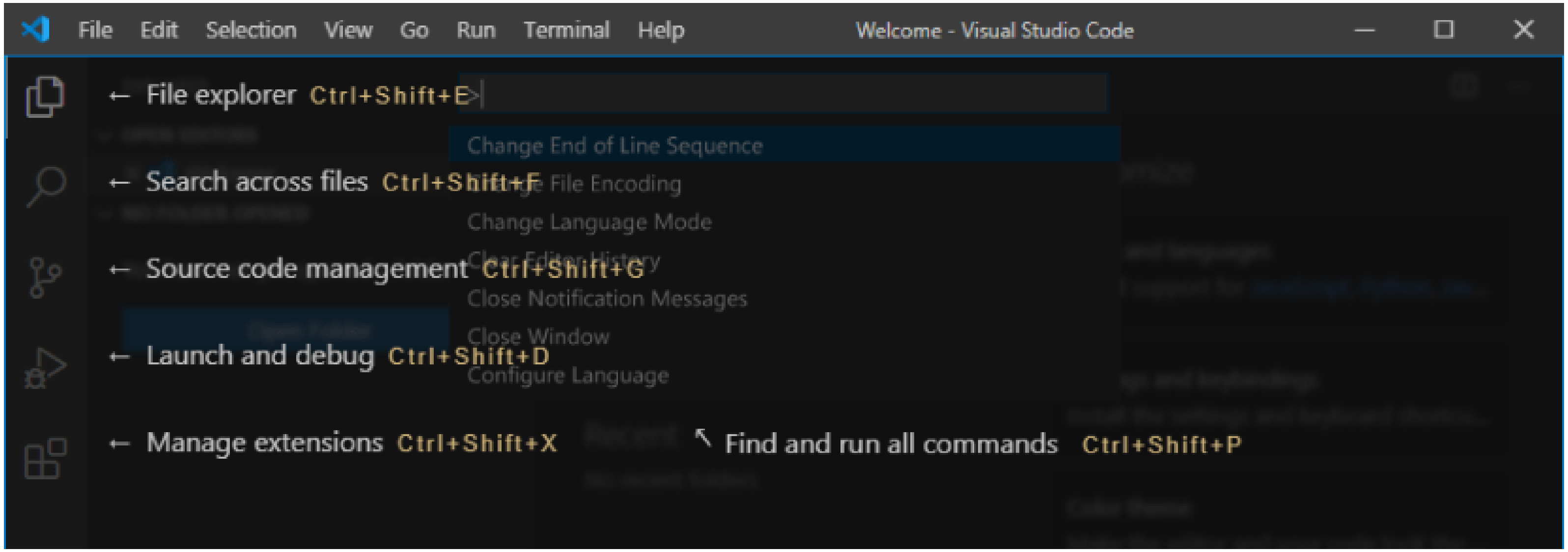
- After that, the PlatformIO icon should show up on the left sidebar as well as a Home icon that redirects you to PlatformIO home.
- If you don't see the PIO icon and the quick tools at the bottom, you may need to restart VS code for the changes to take effect.
- At the bottom, there's a blue bar with PlatformIO commands.





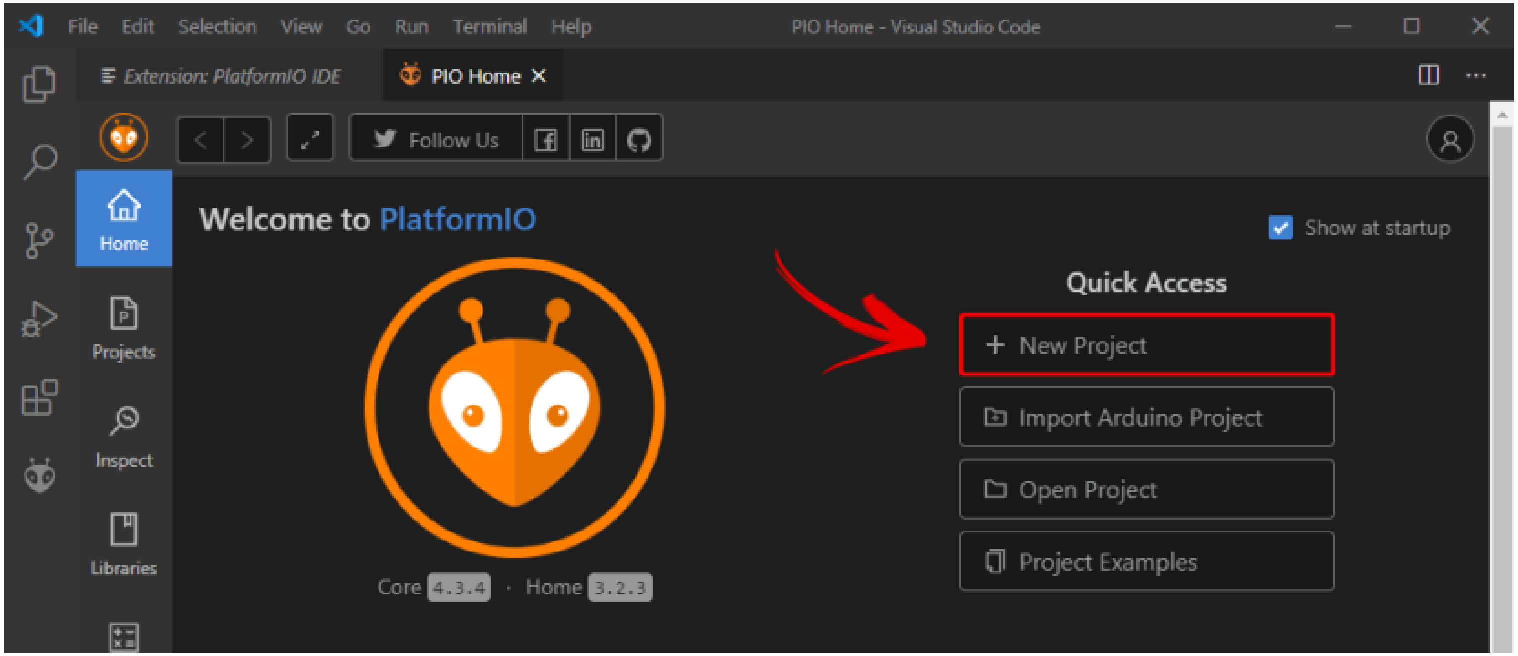
Installing PlatformIO IDE Extension

You can press Ctrl+Shift+X or go to View > Command Palette… to show all the available commands. If you're searching for command and you don't know where it is or its shortcut, you just need to go to the Command Palette and search for it.

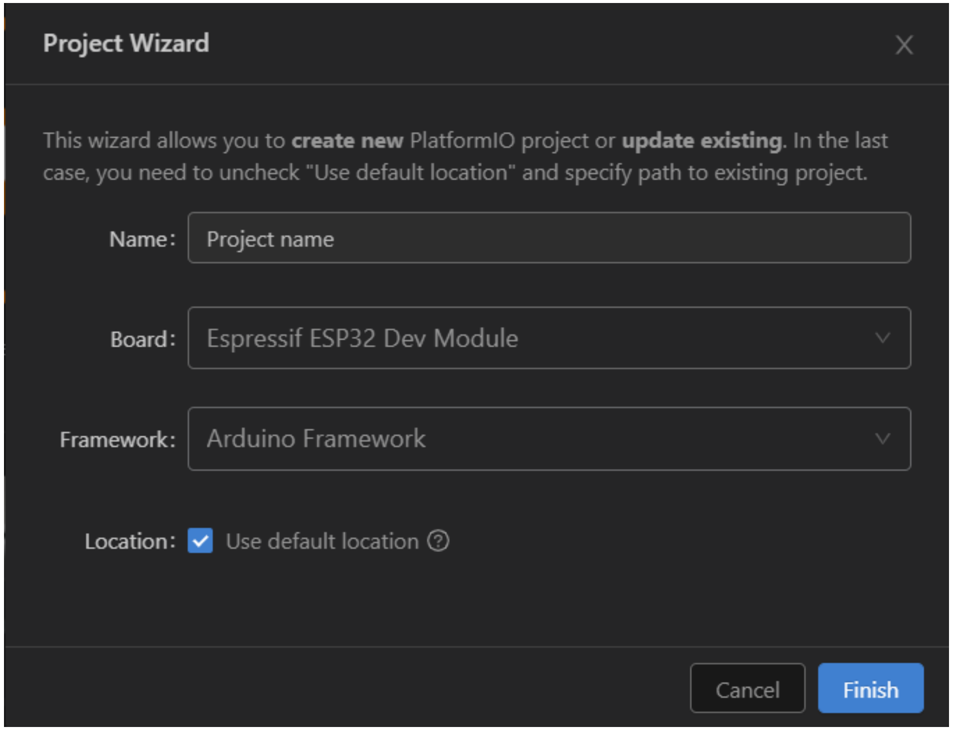

File Edit Selection View Go Run Terminal Help
Welcome - Visual Studio Code
← File explorer Ctrl+Shift+E
>
Change End of Line Sequence
← Search across files Ctrl+Shift+F
File Encoding
Change Language Mode
← Source code management Ctrl+Shift+G
Clear Editor History
Close Notification Messages
Close Window
← Launch and debug Ctrl+Shift+D
Configure Language
← Manage extensions Ctrl+Shift+X
↖ Find and run all commands Ctrl+Shift+P

Imperial College
London

Create a New Project (blink Led)



Create a New Project (blink Led)



Project Wizard

This wizard allows you to **create new** PlatformIO project or **update existing**. In the last case, you need to uncheck "Use default location" and specify path to existing project.

Name: Project name

Board: Espressif ESP32 Dev Module

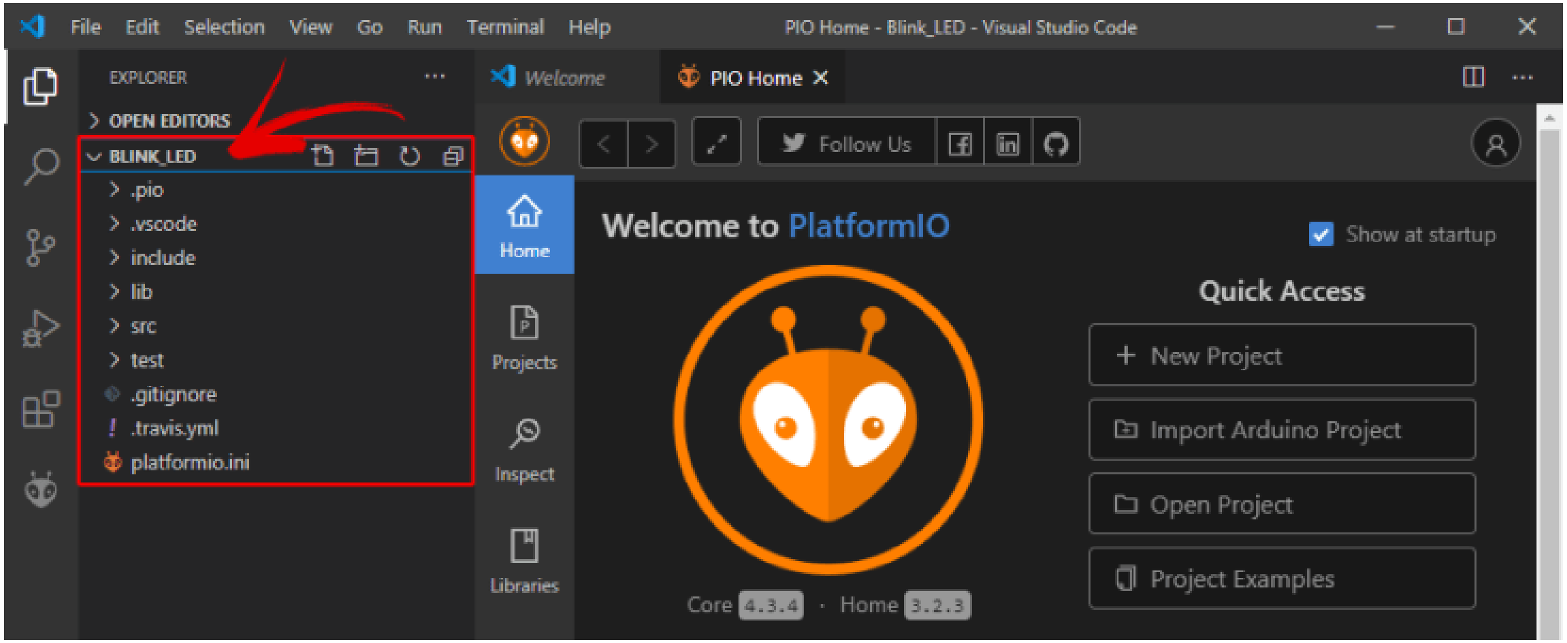
Framework: Arduino Framework

Location: ☑ Use default location

Imperial College London

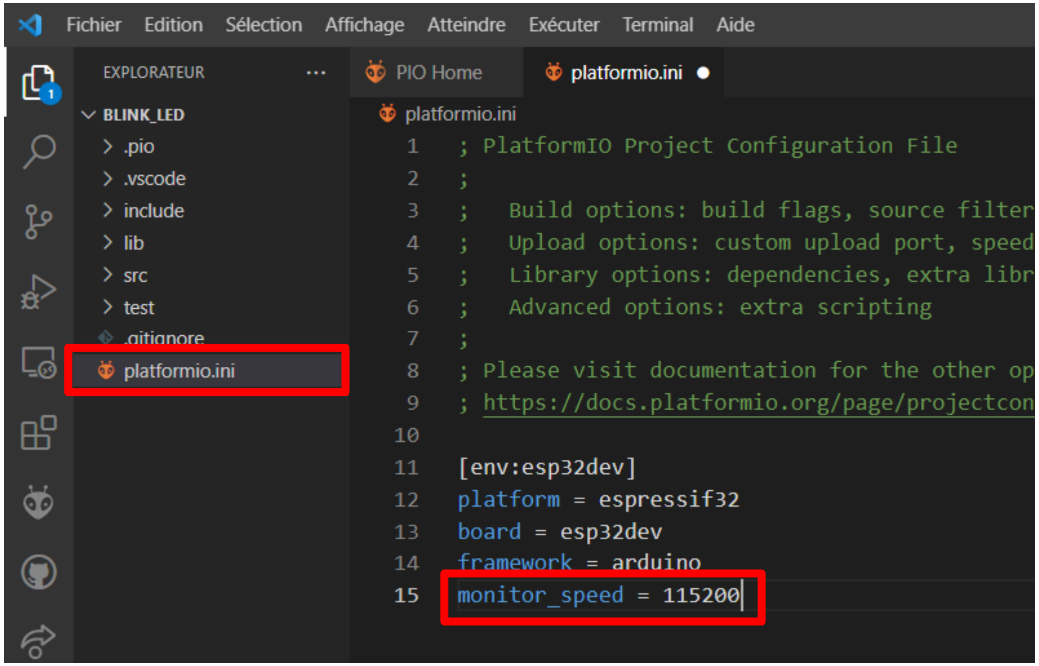
Create a New Project (blink Led)

The Blink_LED project should be accessible from the Explorer tab.



Create a New Project (blink Led)

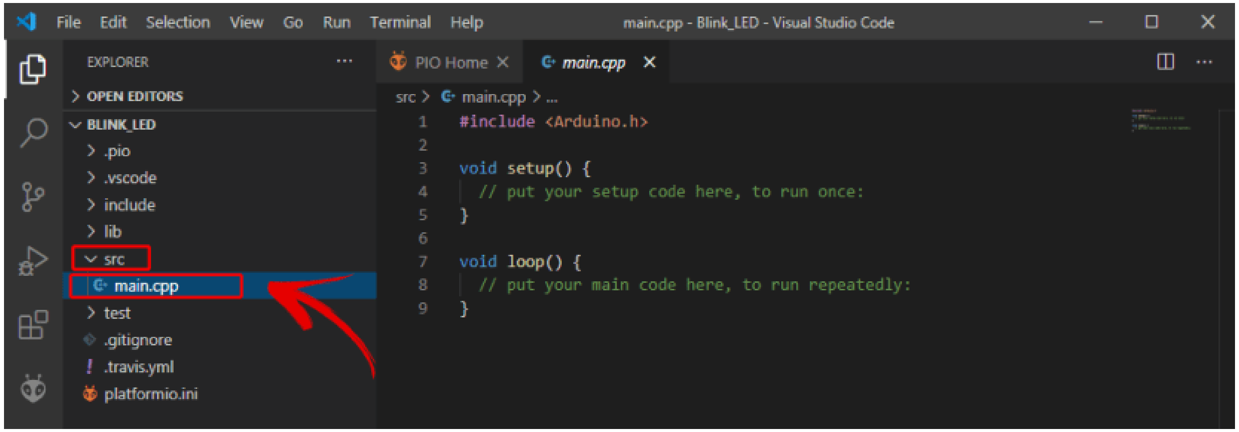
- **The platformio.ini file is the PlatformIO Configuration File for your project. It shows the platform, board, and framework for your project.**
- **You can also add other configurations like libraries to be included, and upload options.**
- **With the ESP32, if you want to use a baud rate of 115200 in your Serial Monitor, you just need to add** monitor_speed = 115200**.**



```
; PlatformIO Project Configuration File
;
;   Build options: build flags, source filter
;   Upload options: custom upload port, speed
;   Library options: dependencies, extra libr
;   Advanced options: extra scripting
;
; Please visit documentation for the other op
; https://docs.platformio.org/page/projectcon

[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
monitor_speed = 115200
```

Create a New Project (blink Led)

- The src folder is your working folder. Under the src folder, there's a main.cpp file. That's where you write your code. Click on that file. The structure of an Arduino program should open with the setup() and loop() functions.
- In PlatformIO, all your Arduino sketches should start with the #include <Arduino.h>.



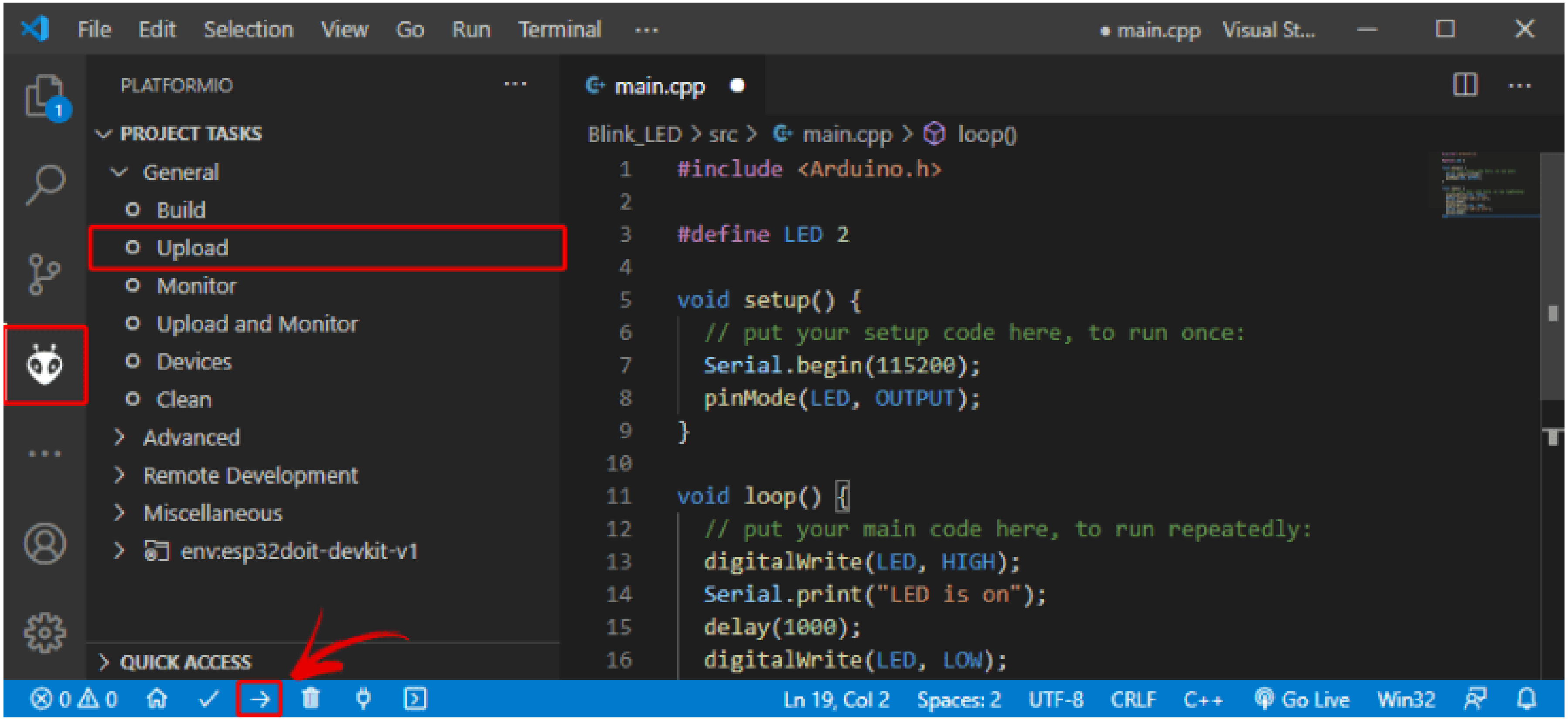
```
#include <Arduino.h>

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Create a New Project (blink Led)

```
#include <Arduino.h>
#define LED 2
void setup() {
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
}
```

```
void loop() {
    digitalWrite(LED, HIGH);
    Serial.println("LED is on");
    delay(1000);
    digitalWrite(LED, LOW);
    Serial.println("LED is off");
    delay(1000);
}
```

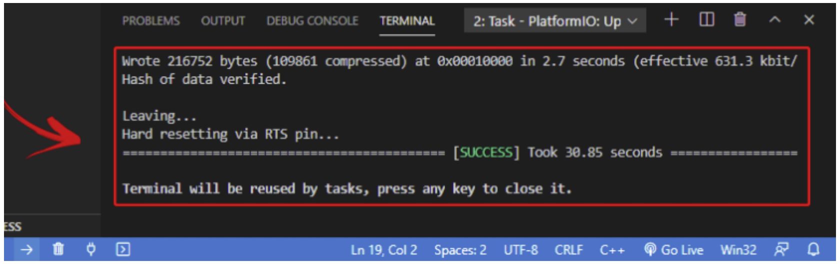
Create a New Project (blink Led)



```cpp
#include <Arduino.h>

#define LED 2

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(LED, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED, HIGH);
  Serial.print("LED is on");
  delay(1000);
  digitalWrite(LED, LOW);
```

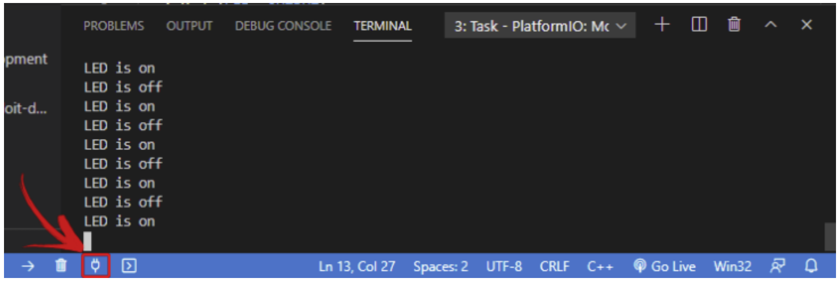
Create a New Project (blink Led)

Now, click on the Serial Monitor icon and you should see it printing the current LED state.



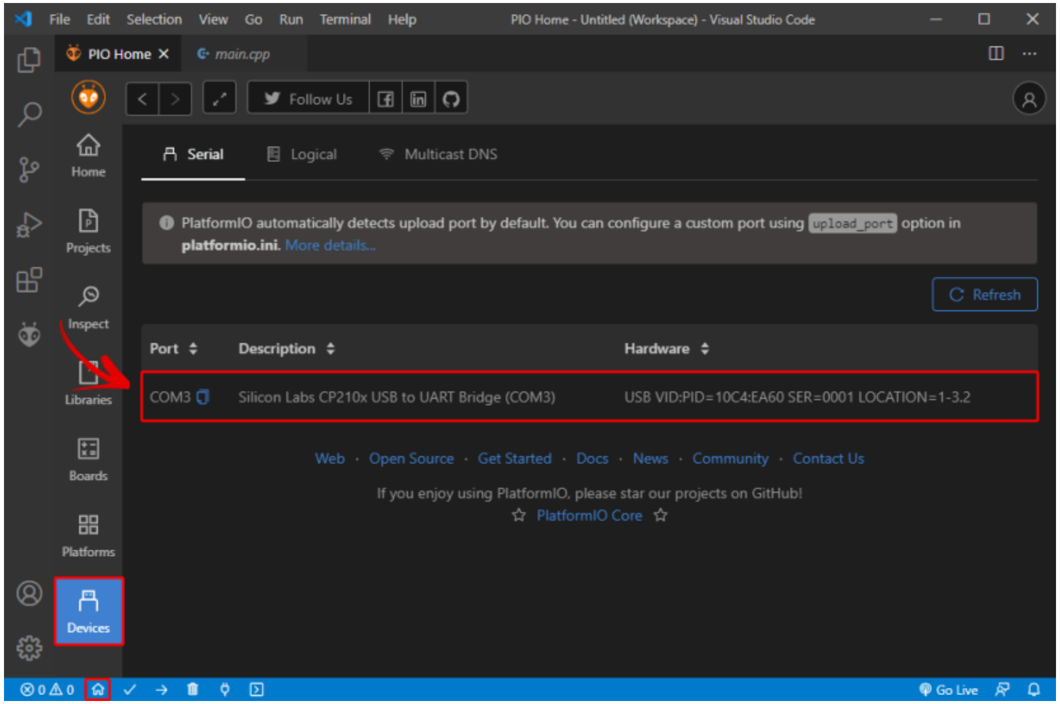
```
Wrote 216752 bytes (109861 compressed) at 0x00010000 in 2.7 seconds (effective 631.3 kbit/
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
========================================= [SUCCESS] Took 30.85 seconds ================

Terminal will be reused by tasks, press any key to close it.
```

If the code is successfully uploaded, you should get the following message.



```
LED is on
LED is off
LED is on
LED is off
LED is on
LED is off
LED is on
LED is off
LED is on
```

Detect COM Port



PlatformIO automatically detects upload port by default. You can configure a custom port using `upload_port` option in **platformio.ini.** More details...

Port	Description	Hardware
COM3	Silicon Labs CP210x USB to UART Bridge (COM3)	USB VID:PID=10C4:EA60 SER=0001 LOCATION=1-3.2

If you enjoy using PlatformIO, please star our projects on GitHub!

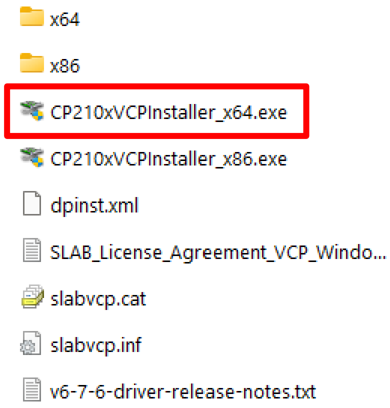
☆ PlatformIO Core ☆

PlatformIO will automatically detect the port your board is connected to. To check the connected devices, you can go to the PIO Home and click the Devices icon.

if no device is detected, install the esp32 driver in the shared folder.

Imperial College London

Detect COM Port


x64
x86
CP210xVCPInstaller_x64.exe
CP210xVCPInstaller_x86.exe
dpinst.xml
SLAB_License_Agreement_VCP_Windo...
slabvcp.cat
slabvcp.inf
v6-7-6-driver-release-notes.txt

PlatformIO will automatically detect the port your board is connected to. To check the connected devices, you can go to the PIO Home and click the Devices icon.

if no device is detected, install the esp32 driver in the shared folder.

If PlatformIO cannot detect the port, try restarting VS Code.

ESP32 Read Digital Inputs

read digital inputs like a button switch :

- **Set the GPIO you want to control as an** INPUT **by using the** pinMode() **function as follows:**

  ```
  pinMode(GPIO, INPUT);
  ```
- **To read a digital input, like a button, use the** digitalRead() **function, that accepts as argument, the GPIO** (int number) **you are referring to.**

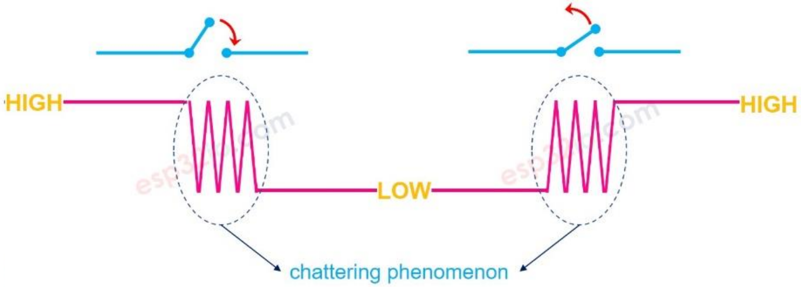
  ```
  digitalRead(GPIO);
  ```

All ESP32 GPIOs can be used as inputs, except GPIOs 6 to 11 (connected to the integrated SPI flash).

Imperial College
London

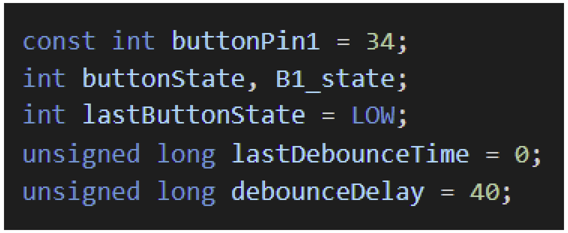
Debounce on a Pushbutton

When a button is pressed/released or when a switch is toggled between ON and OFF, its state is changed from LOW to HGH (or HIGH to LOW) once. Is this correct?

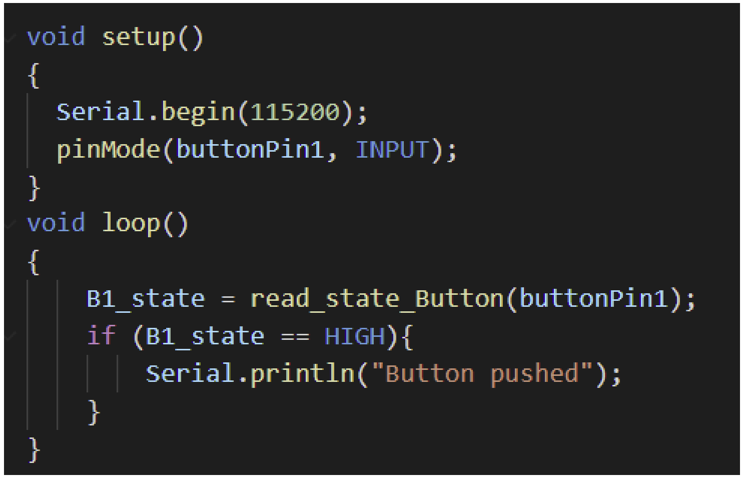


Pushbuttons often generate spurious open/close transitions when pressed, due to mechanical and physical issues: these transitions may be read as multiple presses in a very short time fooling the program.

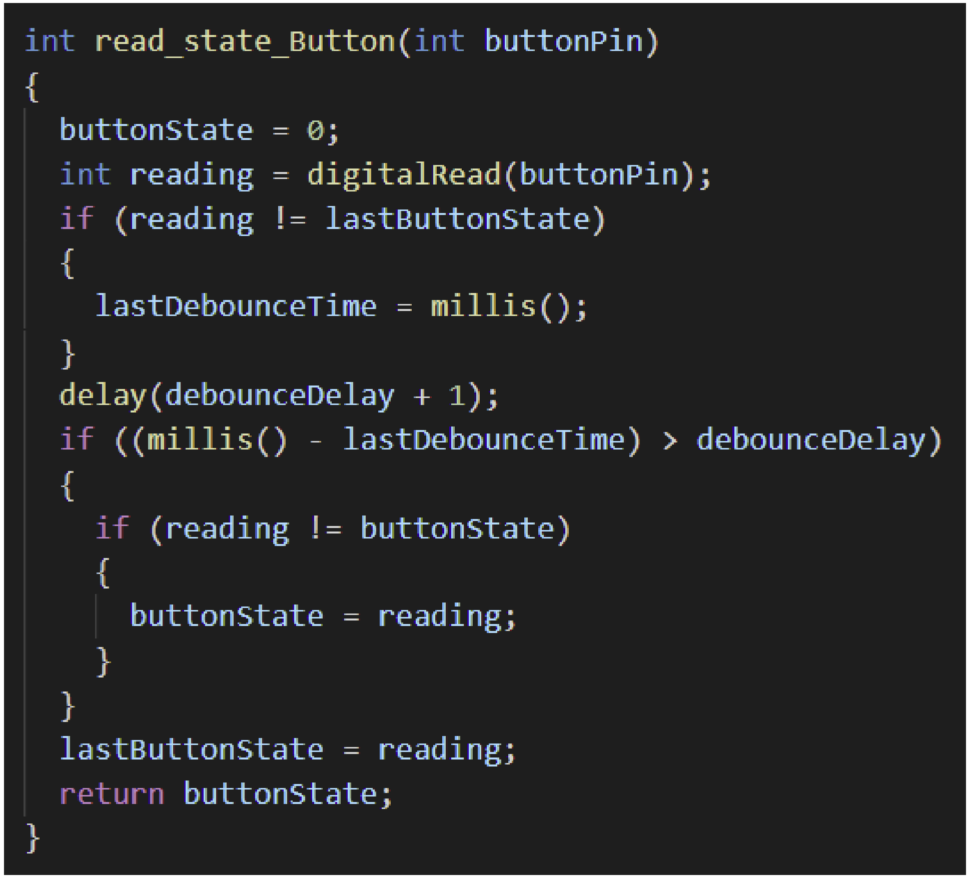
Debounce on a Pushbutton



```
const int buttonPin1 = 34;
int buttonState, B1_state;
int lastButtonState = LOW;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 40;
```



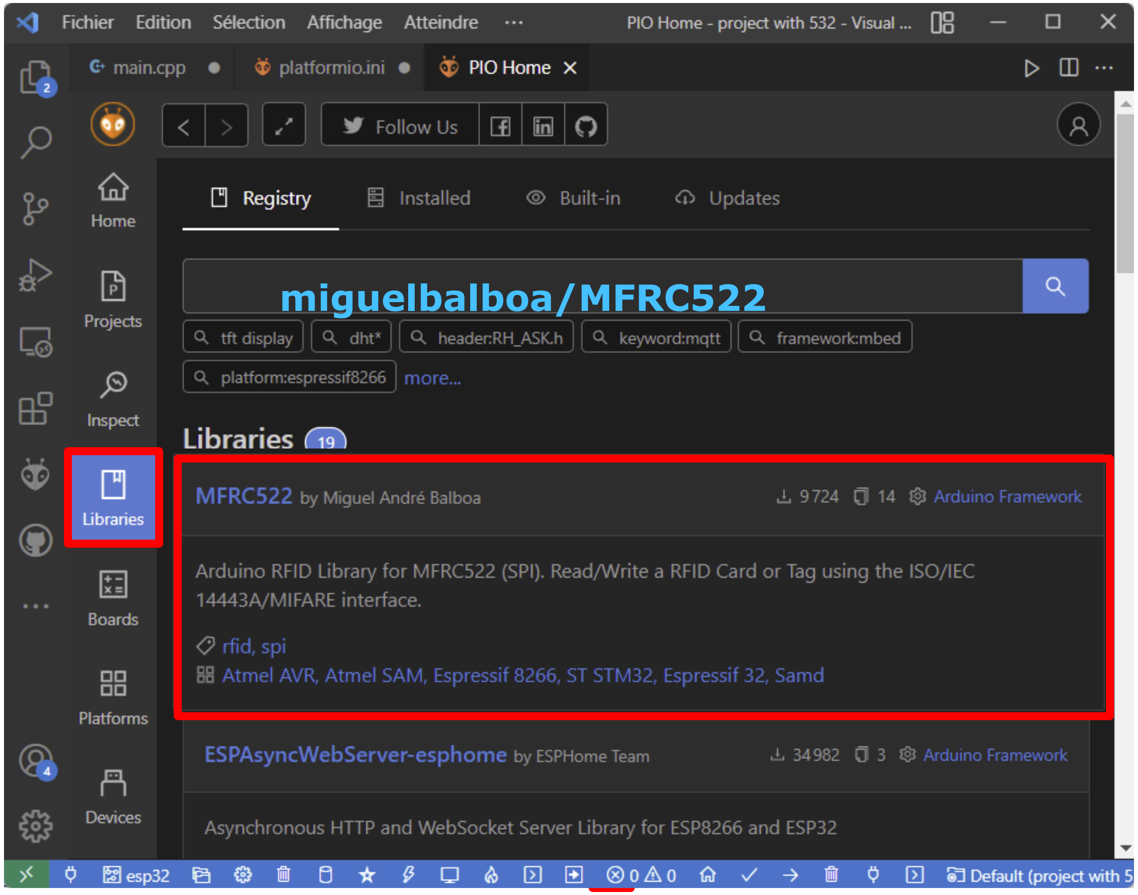
```
void setup()
{
  Serial.begin(115200);
  pinMode(buttonPin1, INPUT);
}
void loop()
{
    B1_state = read_state_Button(buttonPin1);
    if (B1_state == HIGH){
        Serial.println("Button pushed");
    }
}
```



```
int read_state_Button(int buttonPin)
{
  buttonState = 0;
  int reading = digitalRead(buttonPin);
  if (reading != lastButtonState)
  {
    lastDebounceTime = millis();
  }
  delay(debounceDelay + 1);
  if ((millis() - lastDebounceTime) > debounceDelay)
  {
    if (reading != buttonState)
    {
      buttonState = reading;
    }
  }
  lastButtonState = reading;
  return buttonState;
}
```

Installing ESP32 Libraries

- Click the Home icon to go to PlatformIO Home.
- Click on the Libraries icon on the left sidebar.
- Search for the library you want to install.
- For example, RFID 522 library:
- miguelbalboa/MFRC522



miguelbalboa/MFRC522

Libraries 19

MFRC522 by Miguel André Balboa ⤓ 9724 14 Arduino Framework

Arduino RFID Library for MFRC522 (SPI). Read/Write a RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface.

rfid, spi

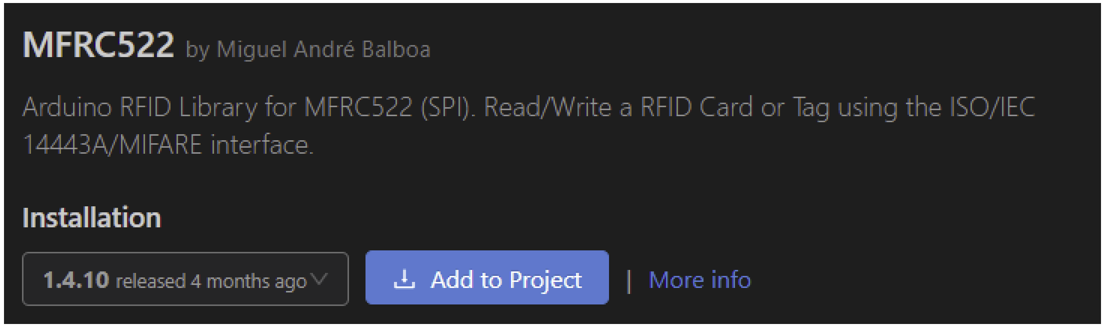
Atmel AVR, Atmel SAM, Espressif 8266, ST STM32, Espressif 32, Samd

ESPAsyncWebServer-esphome by ESPHome Team ⤓ 34982 3 Arduino Framework

Asynchronous HTTP and WebSocket Server Library for ESP8266 and ESP32

Installing ESP32 Libraries

- Click on the library you want to include in your project. Then, click Add to Project.
- Select the project where you want to use the library.



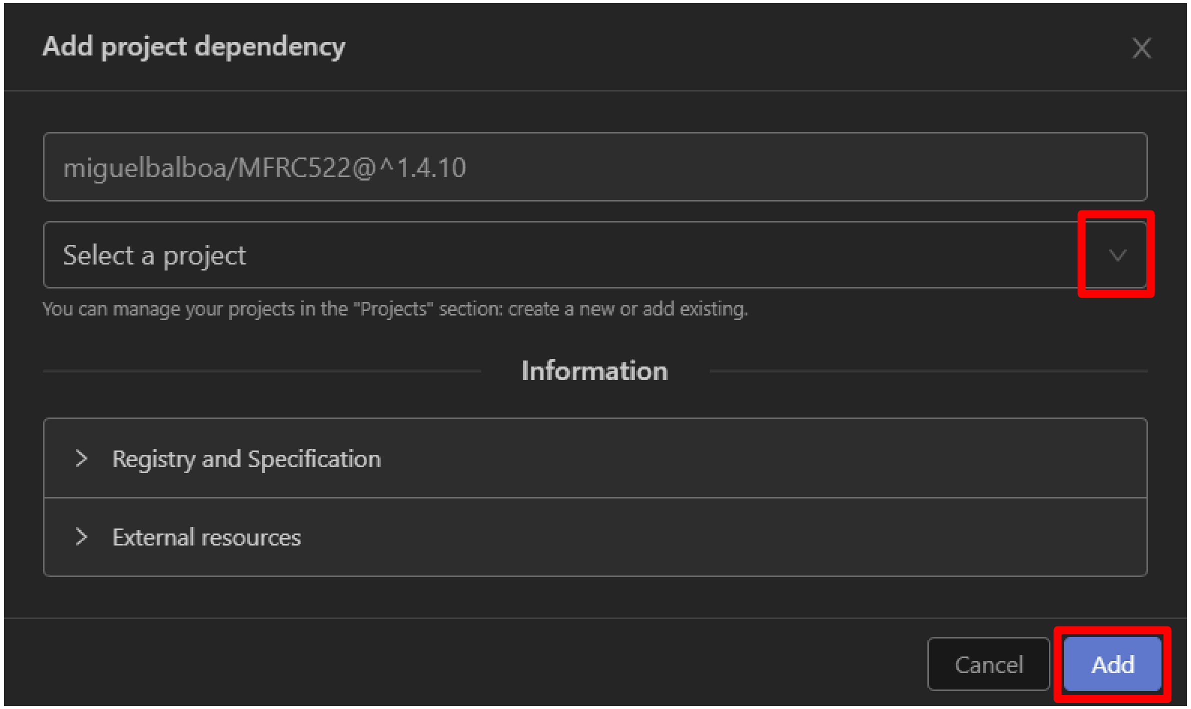
MFRC522 by Miguel André Balboa

Arduino RFID Library for MFRC522 (SPI). Read/Write a RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface.

Installation

Installing ESP32 Libraries

- **Click on the library you want to include in your project. Then, click Add to Project.**
- **Select the project where you want to use the library.**



Add project dependency

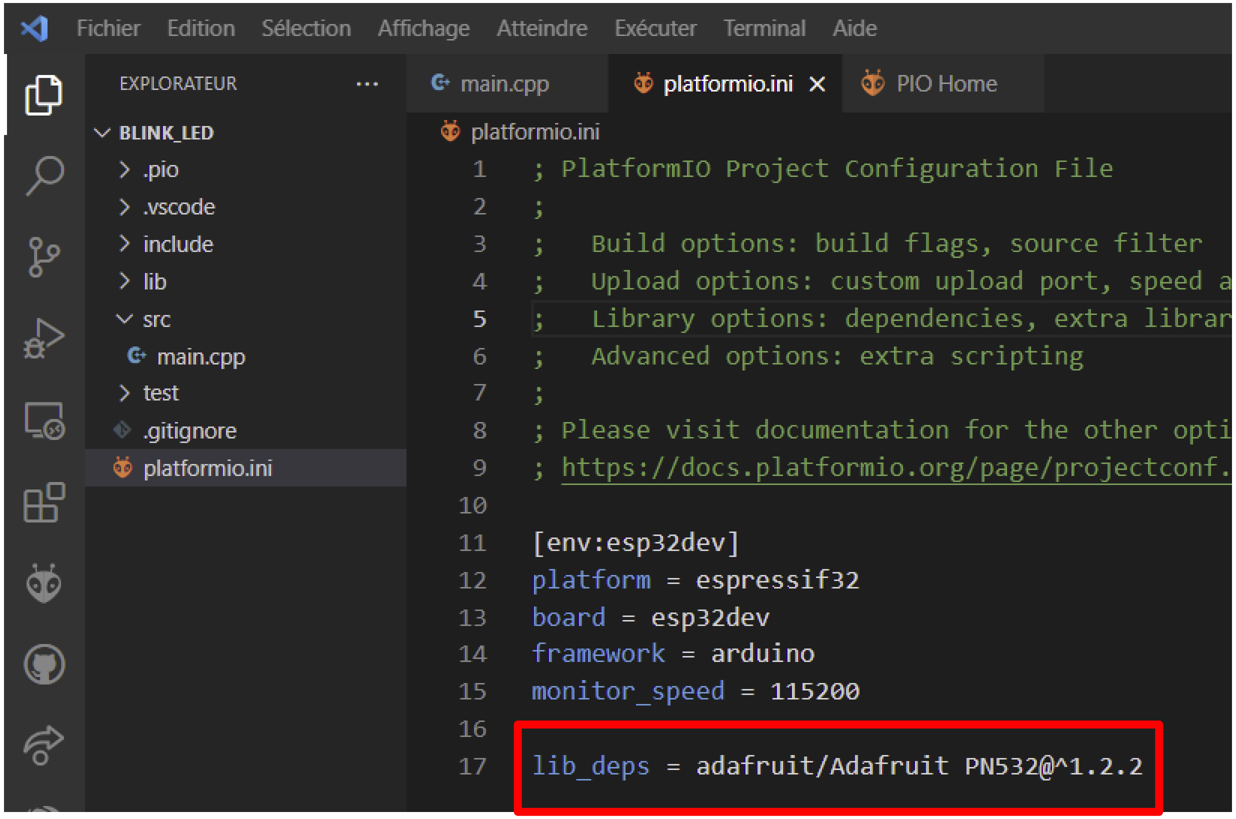
miguelbalboa/MFRC522@^1.4.10

Select a project

You can manage your projects in the "Projects" section: create a new or add existing.

Installing ESP32 Libraries

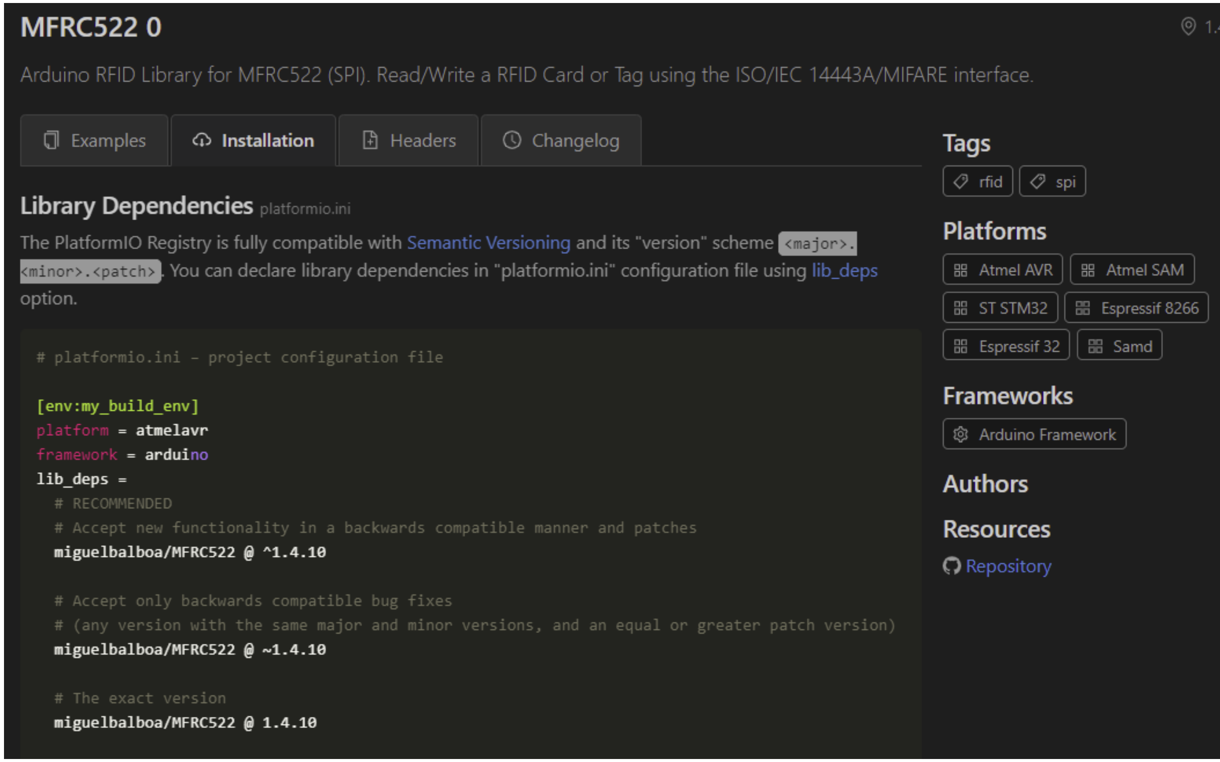
This will add the library identifier using the lid_deps directive on the platformio.ini file.



```
; PlatformIO Project Configuration File
;
;   Build options: build flags, source filter
;   Upload options: custom upload port, speed a
;   Library options: dependencies, extra librar
;   Advanced options: extra scripting
;
; Please visit documentation for the other opti
; https://docs.platformio.org/page/projectconf.

[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
monitor_speed = 115200

lib_deps = adafruit/Adafruit PN532@^1.2.2
```

Installing ESP32 Libraries

Alternatively, on the library window, if you select the Installation tab and scroll a bit, you'll see the identifier for the library. You can choose any of those identifiers depending on the options you want to use.



MFRC522 0

Arduino RFID Library for MFRC522 (SPI). Read/Write a RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface.

Library Dependencies platformio.ini

The PlatformIO Registry is fully compatible with Semantic Versioning and its "version" scheme `<major>.<minor>.<patch>`. You can declare library dependencies in "platformio.ini" configuration file using lib_deps option.

```
# platformio.ini – project configuration file

[env:my_build_env]
platform = atmelavr
framework = arduino
lib_deps =
  # RECOMMENDED
  # Accept new functionality in a backwards compatible manner and patches
  miguelbalboa/MFRC522 @ ^1.4.10

  # Accept only backwards compatible bug fixes
  # (any version with the same major and minor versions, and an equal or greater patch version)
  miguelbalboa/MFRC522 @ ~1.4.10

  # The exact version
  miguelbalboa/MFRC522 @ 1.4.10
```

Tags

rfid spi

Platforms

Atmel AVR, Atmel SAM, ST STM32, Espressif 8266, Espressif 32, Samd

Frameworks

Arduino Framework

Authors

Resources

Repository

Installing ESP32 Libraries

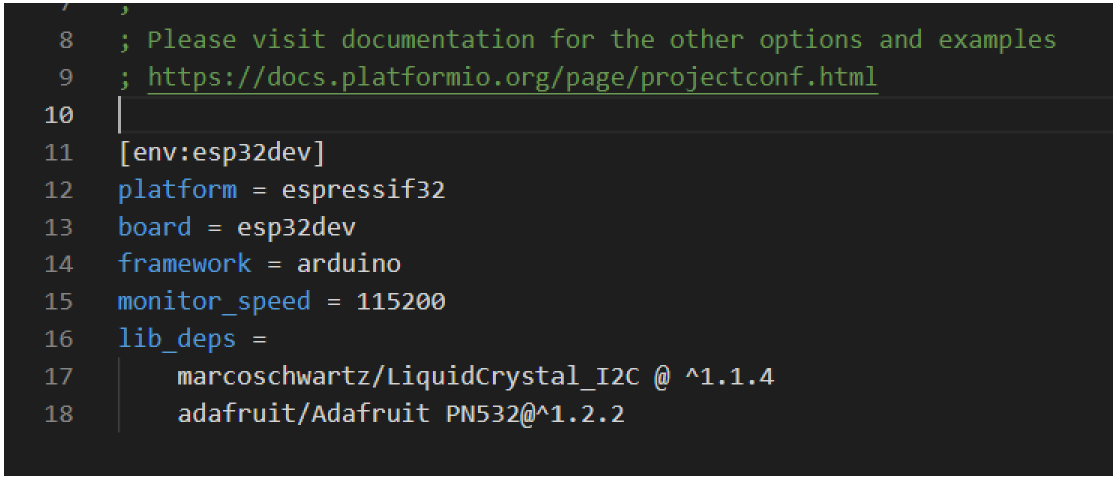
Alternatively, on the library window, if you select the Installation tab and scroll a bit, you'll see the identifier for the library. You can choose any of those identifiers depending on the options you want to use.

https://registry.platformio.org/

https://registry.platformio.org/libraries/miguelbalboa/MFRC522

Installing ESP32 Libraries

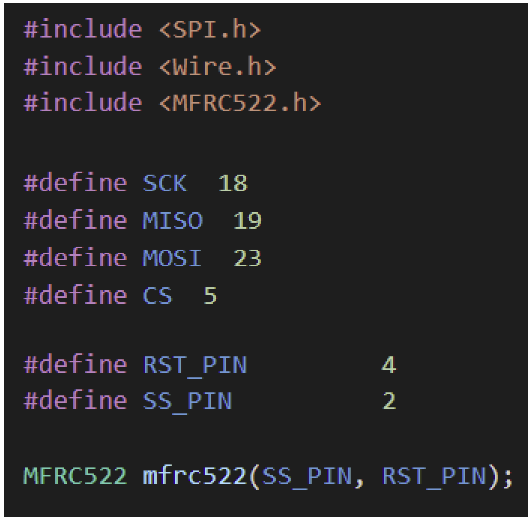
- If you need multiple libraries, you can separate their name by a coma or put them on different lines.
- After installing the MFRC532 library, go to Examples ;



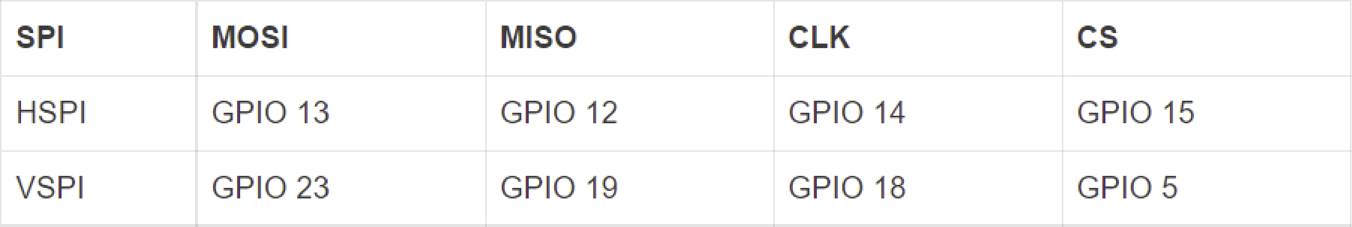
```
;
; Please visit documentation for the other options and examples
; https://docs.platformio.org/page/projectconf.html

[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
monitor_speed = 115200
lib_deps =
    marcoschwartz/LiquidCrystal_I2C @ ^1.1.4
    adafruit/Adafruit PN532@^1.2.2
```

Libraries / SPI communication

- The code starts by including the needed libraries
- As we’re going to use SPI communication you need to change the pin definition to use the ESP32 GPIOs.



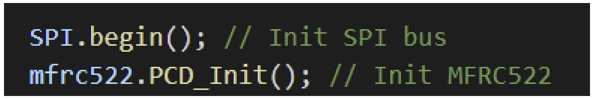
```
#include <SPI.h>
#include <Wire.h>
#include <MFRC522.h>

#define SCK  18
#define MISO  19
#define MOSI  23
#define CS  5

#define RST_PIN         4
#define SS_PIN          2

MFRC522 mfrc522(SS_PIN, RST_PIN);
```



SPI	MOSI	MISO	CLK	CS
HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15
VSPI	GPIO 23	GPIO 19	GPIO 18	GPIO 5

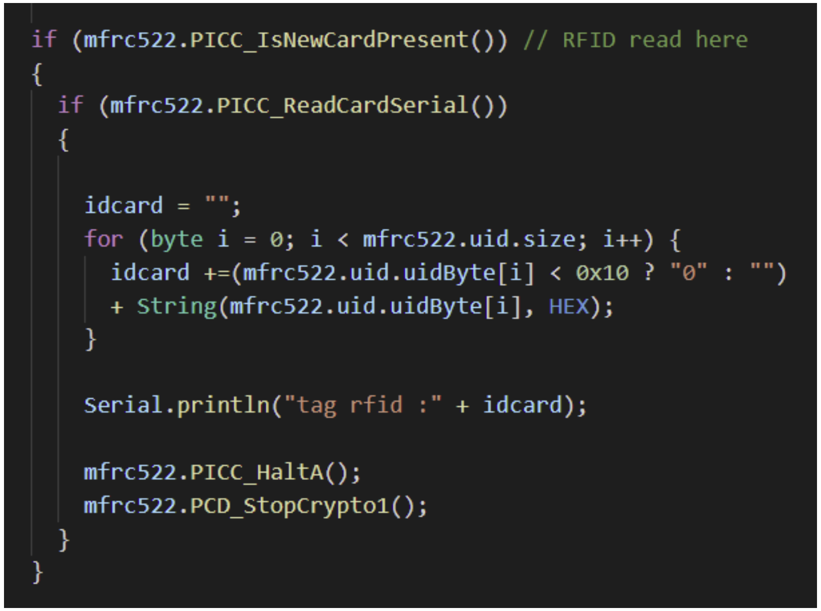
ESP 32 Wi-Fi connection

- **In the setup() you add the sensor initialization:**



```
SPI.begin(); // Init SPI bus
mfrc522.PCD_Init(); // Init MFRC522
```

- **In the loop() you add the following code:**



```
if (mfrc522.PICC_IsNewCardPresent()) // RFID read here
{
  if (mfrc522.PICC_ReadCardSerial())
  {

    idcard = "";
    for (byte i = 0; i < mfrc522.uid.size; i++) {
      idcard +=(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "")
      + String(mfrc522.uid.uidByte[i], HEX);
    }

    Serial.println("tag rfid :" + idcard);

    mfrc522.PICC_HaltA();
    mfrc522.PCD_StopCrypto1();
  }
}
```

ESP 32 Wi-Fi connection

- **The first thing you need to do to use the ESP32 Wi-Fi functionalities is to include the WiFi.h library in your code, as follows:**

```
#include <WiFi.h>
```

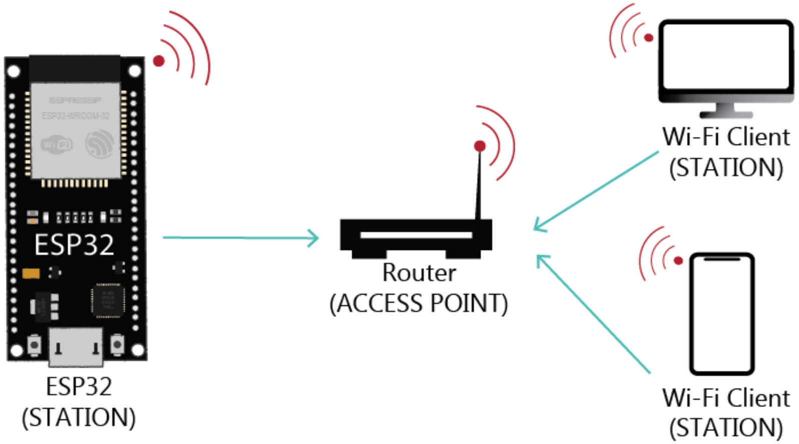
- **This library is automatically "installed" when you start a new project with an ESP32 board in VS Code + PlatformIO.**

- **The ESP32 board can act as Wi-Fi Station, Access Point or both. To set the Wi-Fi mode, use WiFi.mode() and set the desired mode as an argument:**

WiFi.mode(WIFI_STA)	**station mode: the ESP32 connects to an access point**
WiFi.mode(WIFI_AP)	**access point mode: stations can connect to the ESP32**
WiFi.mode(WIFI_STA_AP)	**access point and a station connected to another access point**

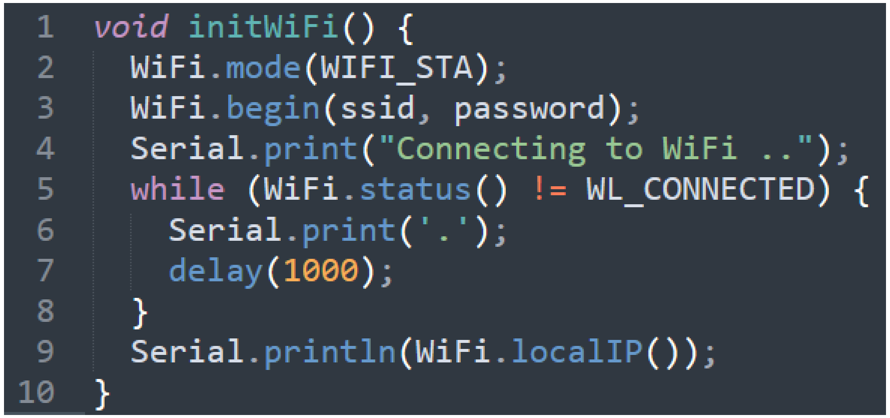
ESP 32 Wi-Fi connection

- When the ESP32 is set as a Wi-Fi station, it can connect to other networks (like your router).
- In this scenario, the router assigns a unique IP address to your ESP board.
- You can communicate with the ESP using other devices (stations) that are also connected to the same network by referring to the ESP unique IP address.



ESP 32 Wi-Fi connection

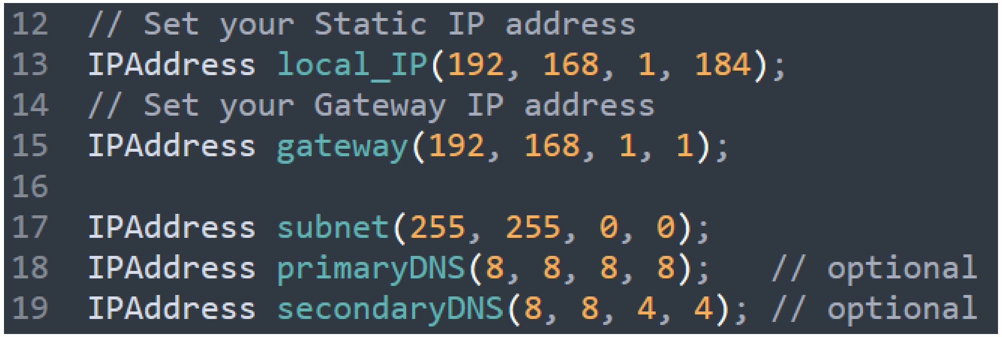
- To connect the ESP32 to a specific Wi-Fi network, you must know its SSID and password.
- Additionally, that network must be within the ESP32 Wi-Fi range.
- You can use the following function to connect the ESP32 to a Wi-Fi network.



```
void initWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
  }
  Serial.println(WiFi.localIP());
}
```

Set a Static ESP32 IP Address

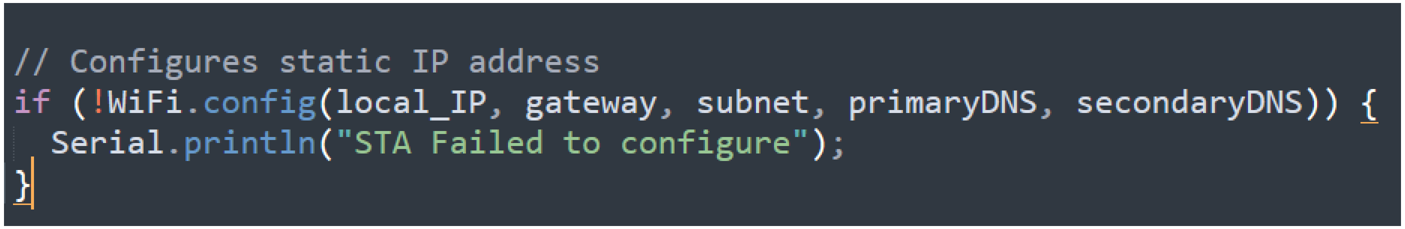
- Instead of getting a randomly assigned IP address, you can set an available IP address of your preference to the ESP32 using WiFi.config().
- Outside the setup() and loop() functions, define the following variables with your own static IP address and corresponding gateway IP address.
- By default, the following code assigns the IP address 192.168.1.184 that works in the gateway 192.168.1.1.



```
// Set your Static IP address
IPAddress local_IP(192, 168, 1, 184);
// Set your Gateway IP address
IPAddress gateway(192, 168, 1, 1);

IPAddress subnet(255, 255, 0, 0);
IPAddress primaryDNS(8, 8, 8, 8);   // optional
IPAddress secondaryDNS(8, 8, 4, 4); // optional
```

Set a Static ESP32 IP Address

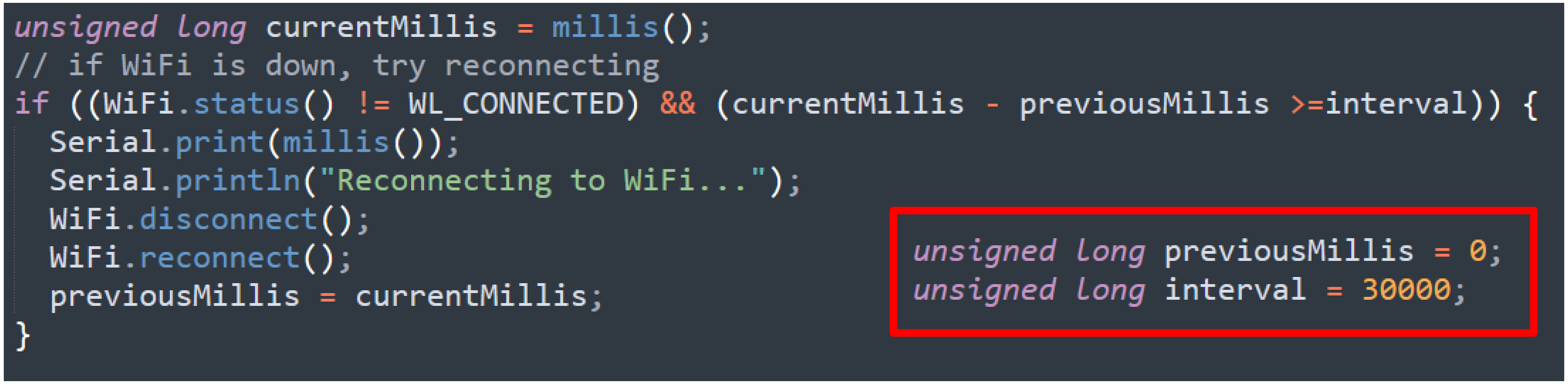
- Then, in the setup() you need to call the WiFi.config() method to assign the configurations to your ESP32.
- The primaryDNS and secondaryDNS parameters are optional and you can remove them.



```
// Configures static IP address
if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {
  Serial.println("STA Failed to configure");
}
```

Reconnect to Wi-Fi Network: After Losing Connection

- **To reconnect to Wi-Fi after a connection is lost, we can use WiFi.reconnect() to try to reconnect to the previously connected access point:**
- **Or, we can call WiFi.disconnect() followed by WiFi.begin(ssid,password).**
- **In the loop() we can check once in a while if the board is connected.**



```
unsigned long currentMillis = millis();
// if WiFi is down, try reconnecting
if ((WiFi.status() != WL_CONNECTED) && (currentMillis - previousMillis >=interval)) {
  Serial.print(millis());
  Serial.println("Reconnecting to WiFi...");
  WiFi.disconnect();
  WiFi.reconnect();
  previousMillis = currentMillis;
}
```

```
unsigned long previousMillis = 0;
unsigned long interval = 30000;
```

Connection to database

Imperial College London

ESP 32 connection to database

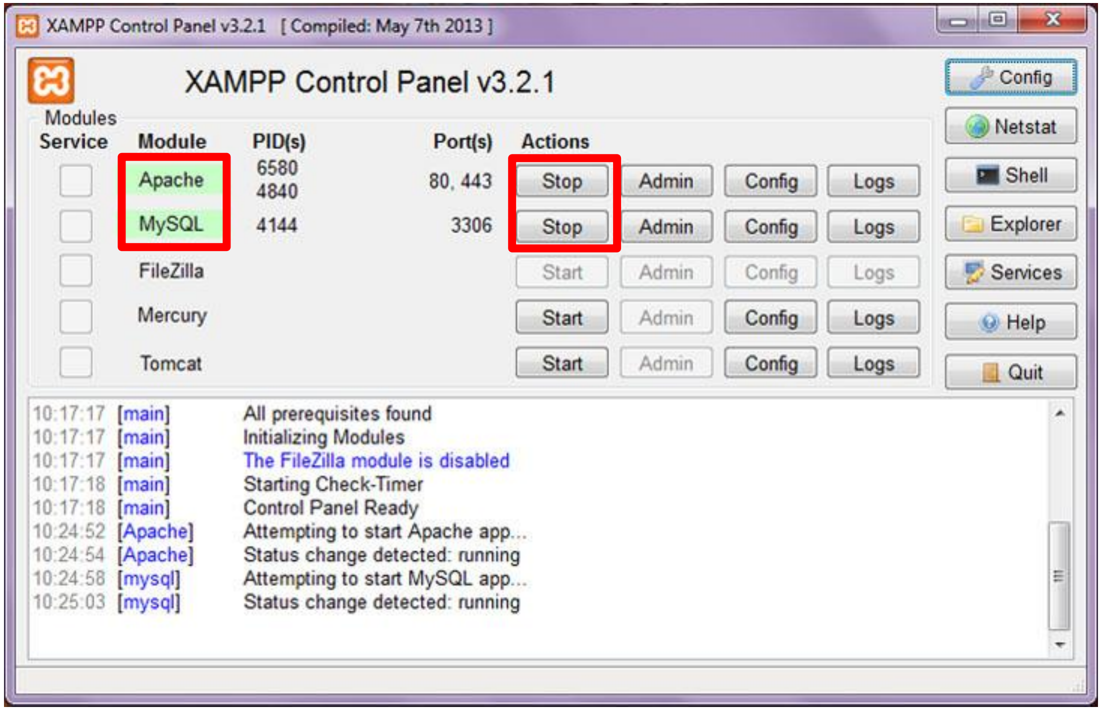
- In this tutorial, we will build an ESP32 client that sends an HTTP POST request to a PHP script to insert data (sensor readings) into a MySQL database.
- You'll also have a web page that displays the sensor readings, timestamps and other information from the database. You can visualize your data from anywhere in the world by accessing your own server.
- As an example, we will use an RFID reader connected to an ESP card. You can change the code provided to send readings from another sensor or use multiple cards.
- In order to create and build this project, you'll use these technologies:
  - ✓ ESP32 programmed with PlatformIO / Arduino IDE / etc ...
  - ✓ Hosting server and domain name
  - ✓ PHP script to insert data into MySQL and display it on a web page
  - ✓ MySQL database to store readings

ESP 32 connection to database

XAMPP server installation

- Uncheck the Learn more about bitnami option and click Next button.
- Choose the root directory path to set up the htdocs folder for our applications. For example, ‘C:\xampp’.
- Click the Allow access button to allow the XAMPP modules from the Windows firewall.
- After the installation process, click the Finish button of the XAMPP Setup wizard.
- Now the XAMPP icon is clearly visible on the right side of start menu. Show or Hide can be set by using the control panel by clicking on the icon.
- To start Apache and MySql, just click on the Start button on the control panel.

ESP 32 connection to database



XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

XAMPP Control Panel v3.2.1

Modules

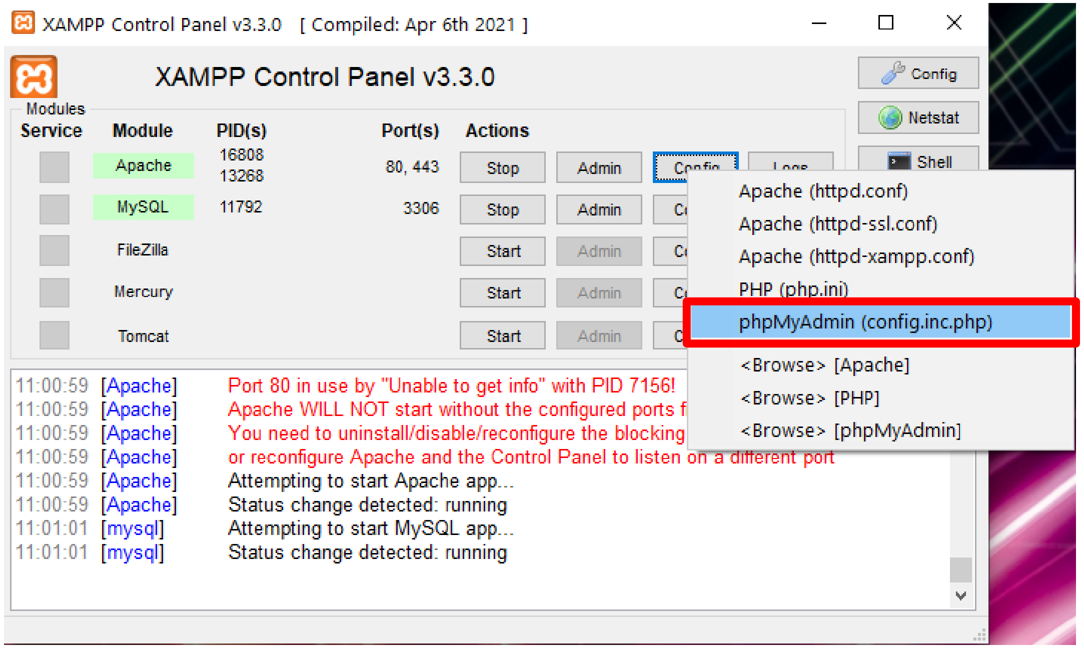
Service	Module	PID(s)	Port(s)	Actions			
□	Apache	6580 4840	80, 443	Stop	Admin	Config	Logs
□	MySQL	4144	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
10:17:17  [main]      All prerequisites found
10:17:17  [main]      Initializing Modules
10:17:17  [main]      The FileZilla module is disabled
10:17:18  [main]      Starting Check-Timer
10:17:18  [main]      Control Panel Ready
10:24:52  [Apache]    Attempting to start Apache app...
10:24:54  [Apache]    Status change detected: running
10:24:58  [mysql]     Attempting to start MySQL app...
10:25:03  [mysql]     Status change detected: running
```

Imperial College London

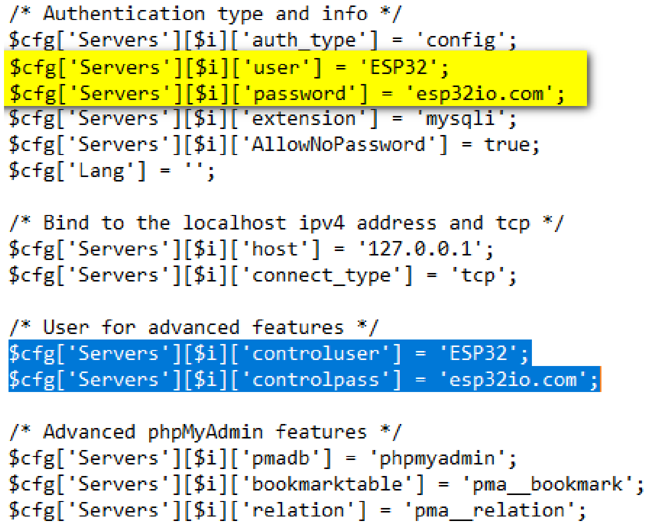
ESP 32 connection to database

change password



Service	Module	PID(s)	Port(s)	Actions			
□	Apache	16808 13268	80, 443	Stop	Admin	Config	Logs
□	MySQL	11792	3306	Stop	Admin	C	
□	FileZilla			Start	Admin	C	
□	Mercury			Start	Admin	C	
□	Tomcat			Start	Admin	C	

```
11:00:59  [Apache]   Port 80 in use by "Unable to get info" with PID 7156!
11:00:59  [Apache]   Apache WILL NOT start without the configured ports f
11:00:59  [Apache]   You need to uninstall/disable/reconfigure the blocking
11:00:59  [Apache]   or reconfigure Apache and the Control Panel to listen on a different port
11:00:59  [Apache]   Attempting to start Apache app...
11:00:59  [Apache]   Status change detected: running
11:01:01  [mysql]    Attempting to start MySQL app...
11:01:01  [mysql]    Status change detected: running
```

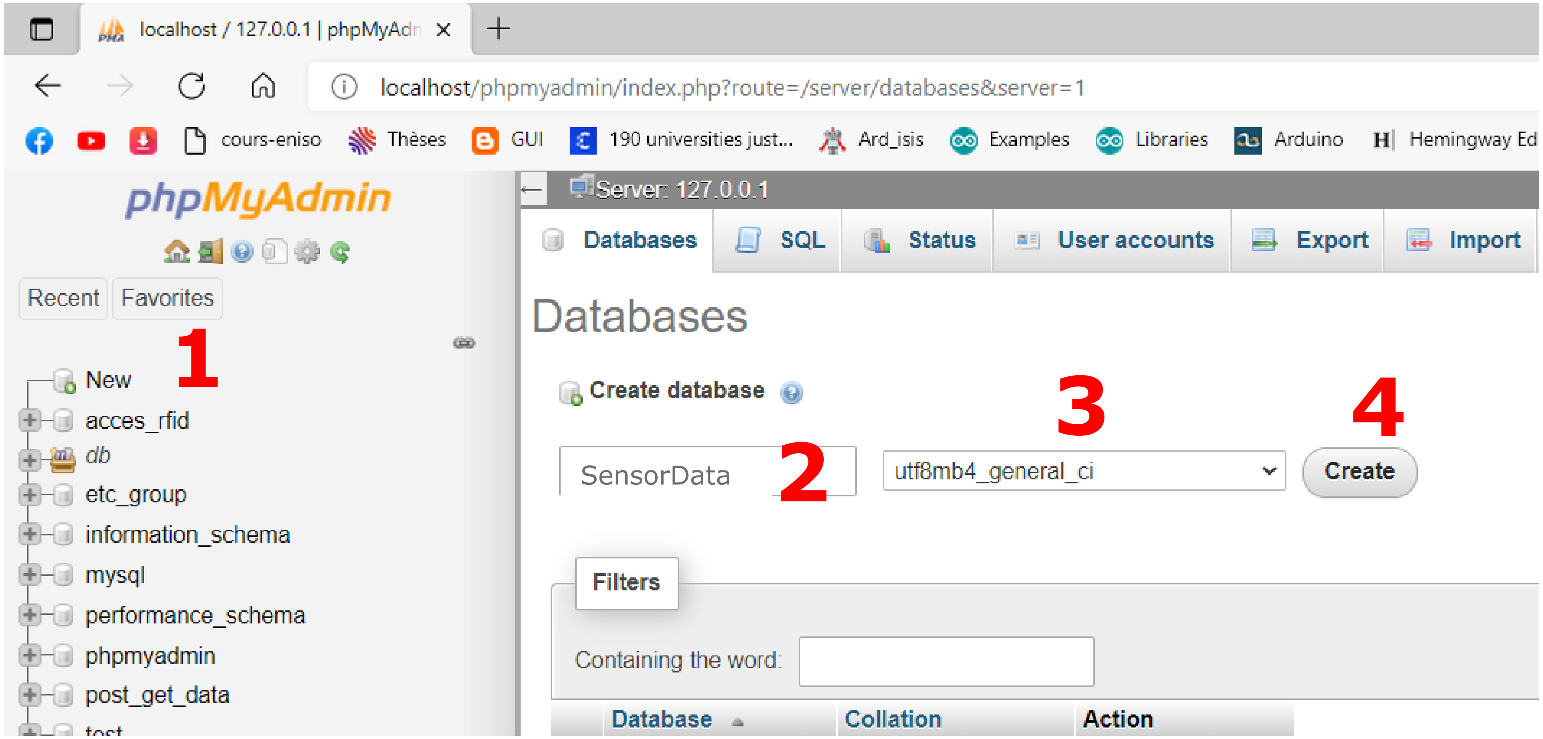


```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'ESP32';
$cfg['Servers'][$i]['password'] = 'esp32io.com';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
$cfg['Lang'] = '';

/* Bind to the localhost ipv4 address and tcp */
$cfg['Servers'][$i]['host'] = '127.0.0.1';
$cfg['Servers'][$i]['connect_type'] = 'tcp';

/* User for advanced features */
$cfg['Servers'][$i]['controluser'] = 'ESP32';
$cfg['Servers'][$i]['controlpass'] = 'esp32io.com';

/* Advanced phpMyAdmin features */
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma__bookmark';
$cfg['Servers'][$i]['relation'] = 'pma__relation';
```

Imperial College London

ESP 32 connection to database

Create a new data base

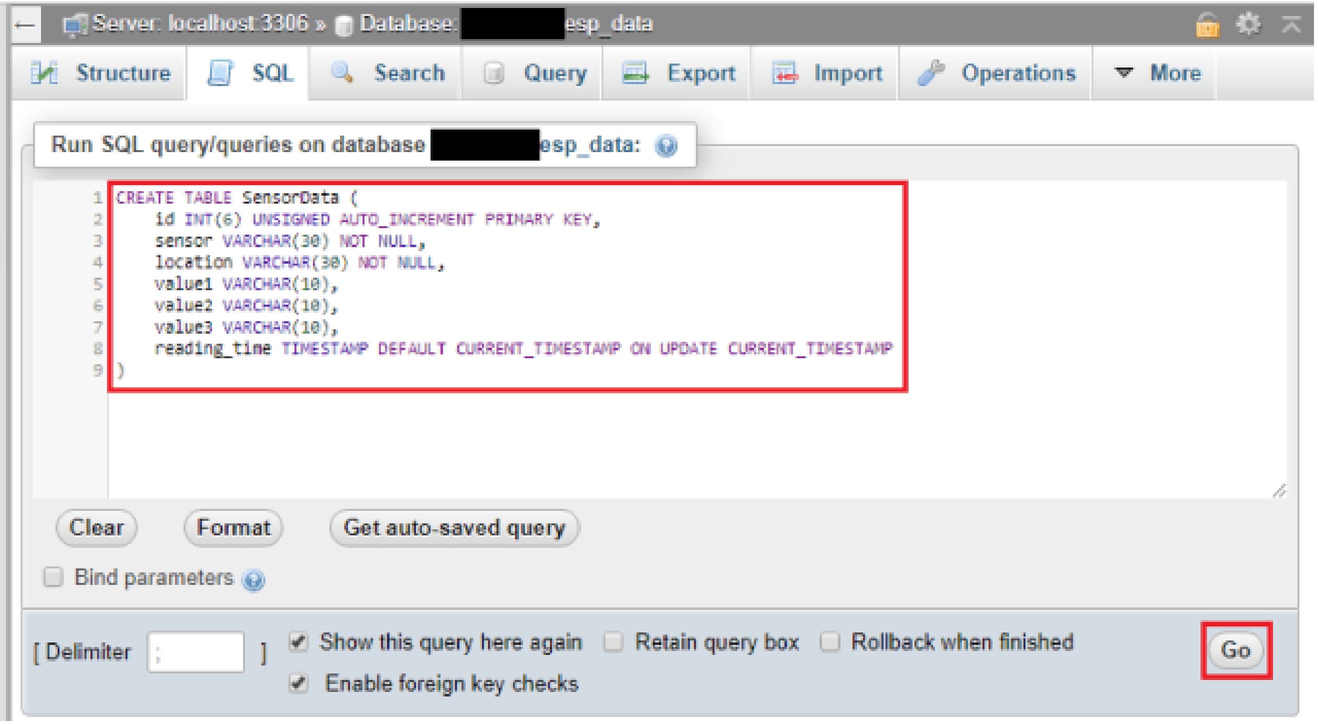


SensorData

utf8mb4_general_ci

ESP 32 connection to database

Create Table Procedure 1 :



Server: localhost:3306 » Database: [redacted]esp_data

Run SQL query/queries on database [redacted]esp_data:

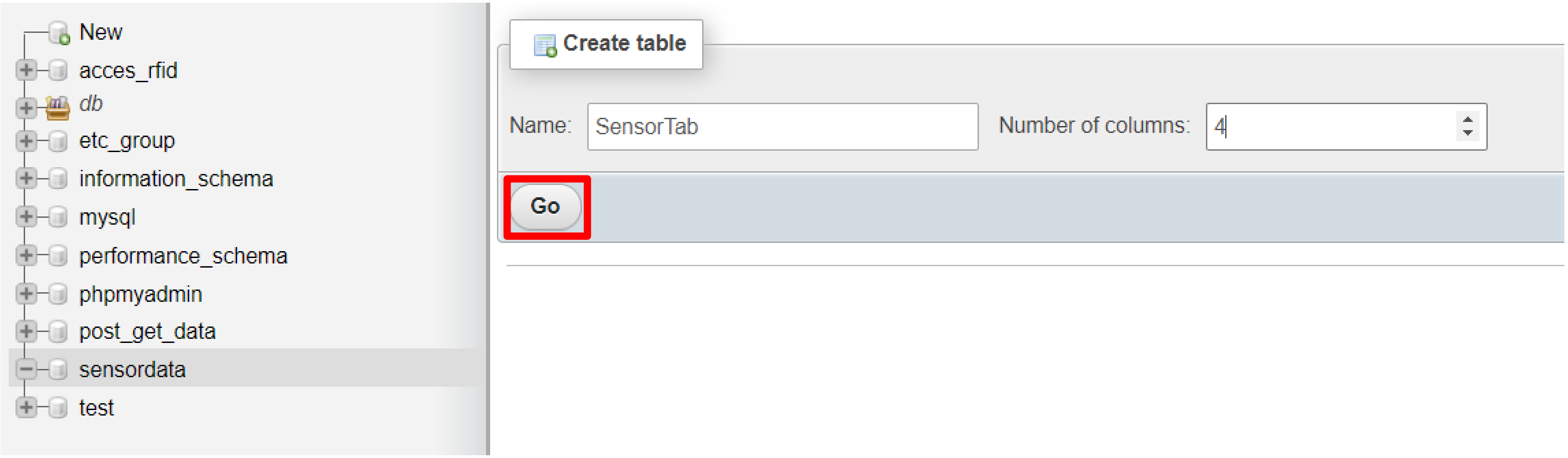
```
CREATE TABLE SensorData (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    sensor VARCHAR(30) NOT NULL,
    location VARCHAR(30) NOT NULL,
    value1 VARCHAR(10),
    value2 VARCHAR(10),
    value3 VARCHAR(10),
    reading_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)
```

□ Bind parameters

[Delimiter ;]
☑ Show this query here again
□ Retain query box
□ Rollback when finished
☑ Enable foreign key checks

ESP 32 connection to database

Create Table Procedure 2 :



Name: SensorTab

Number of columns: 4

ESP 32 connection to database

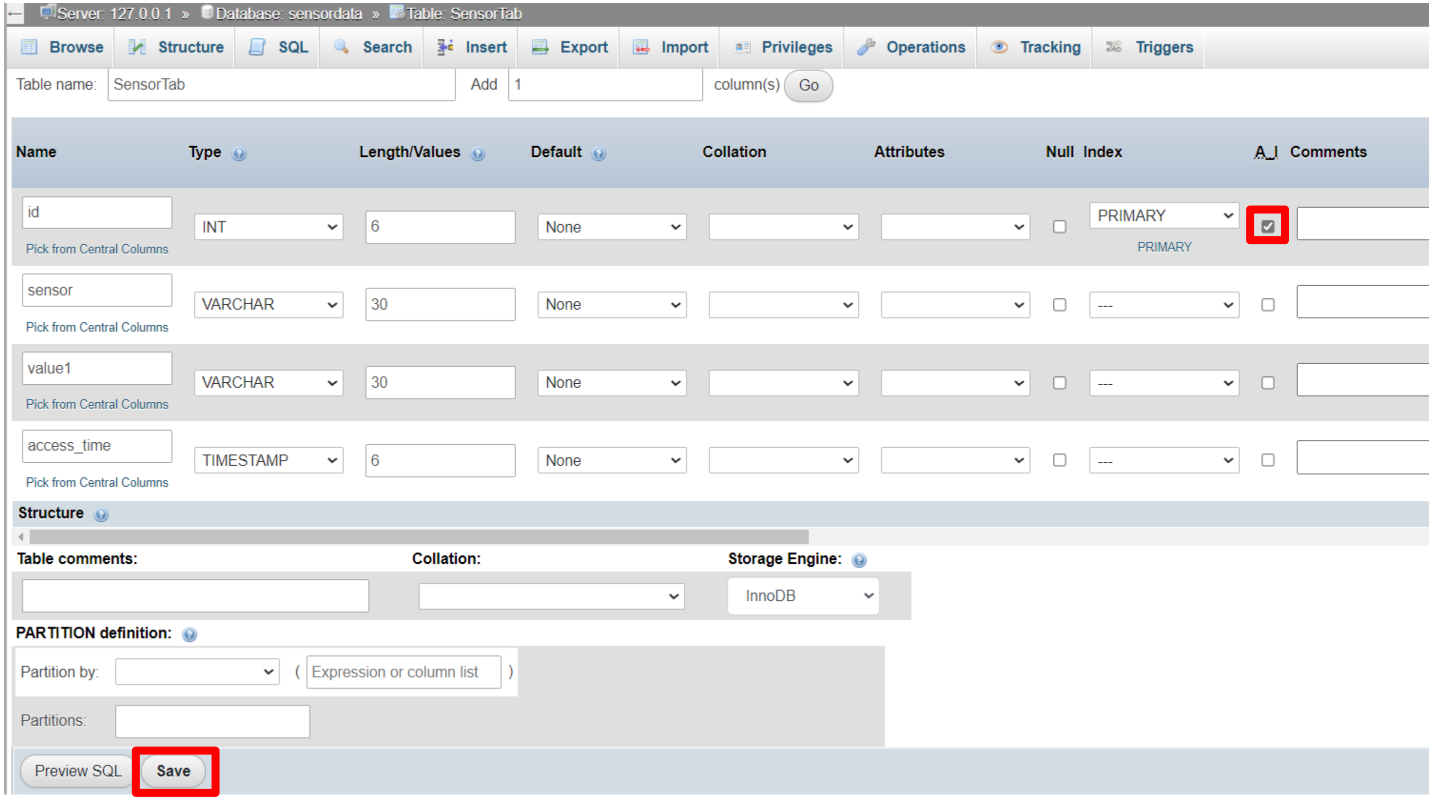


Table name: SensorTab Add 1 column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
id	INT	6	None			□	PRIMARY PRIMARY	☑	
sensor	VARCHAR	30	None			□	---	□	
value1	VARCHAR	30	None			□	---	□	
access_time	TIMESTAMP	6	None			□	---	□	

Structure

Table comments:

Collation:

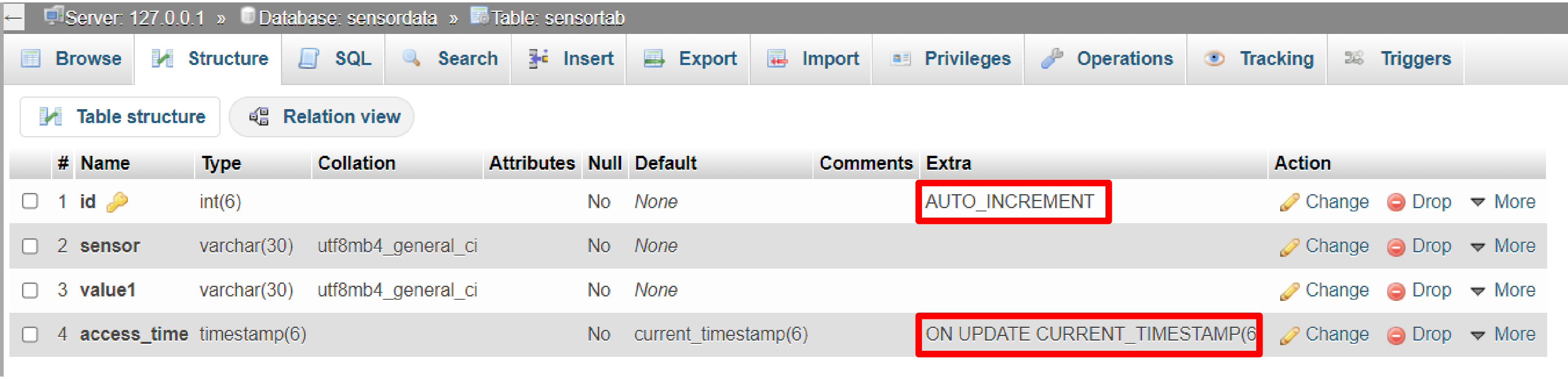
Storage Engine: InnoDB

PARTITION definition:

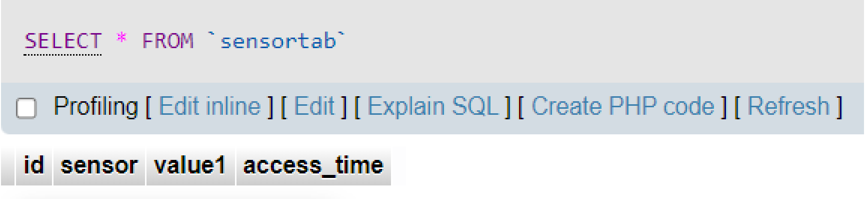
Partition by: (Expression or column list)

Partitions:

ESP 32 connection to database



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(6)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**sensor**	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**value1**	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**access_time**	timestamp(6)			No	current_timestamp(6)		ON UPDATE CURRENT_TIMESTAMP(6	Change Drop More



```
SELECT * FROM `sensortab`
```

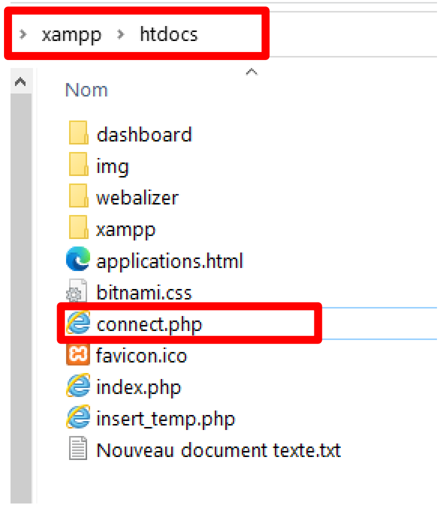
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id	sensor	value1	access_time

Imperial College London

ESP 32 connection to database

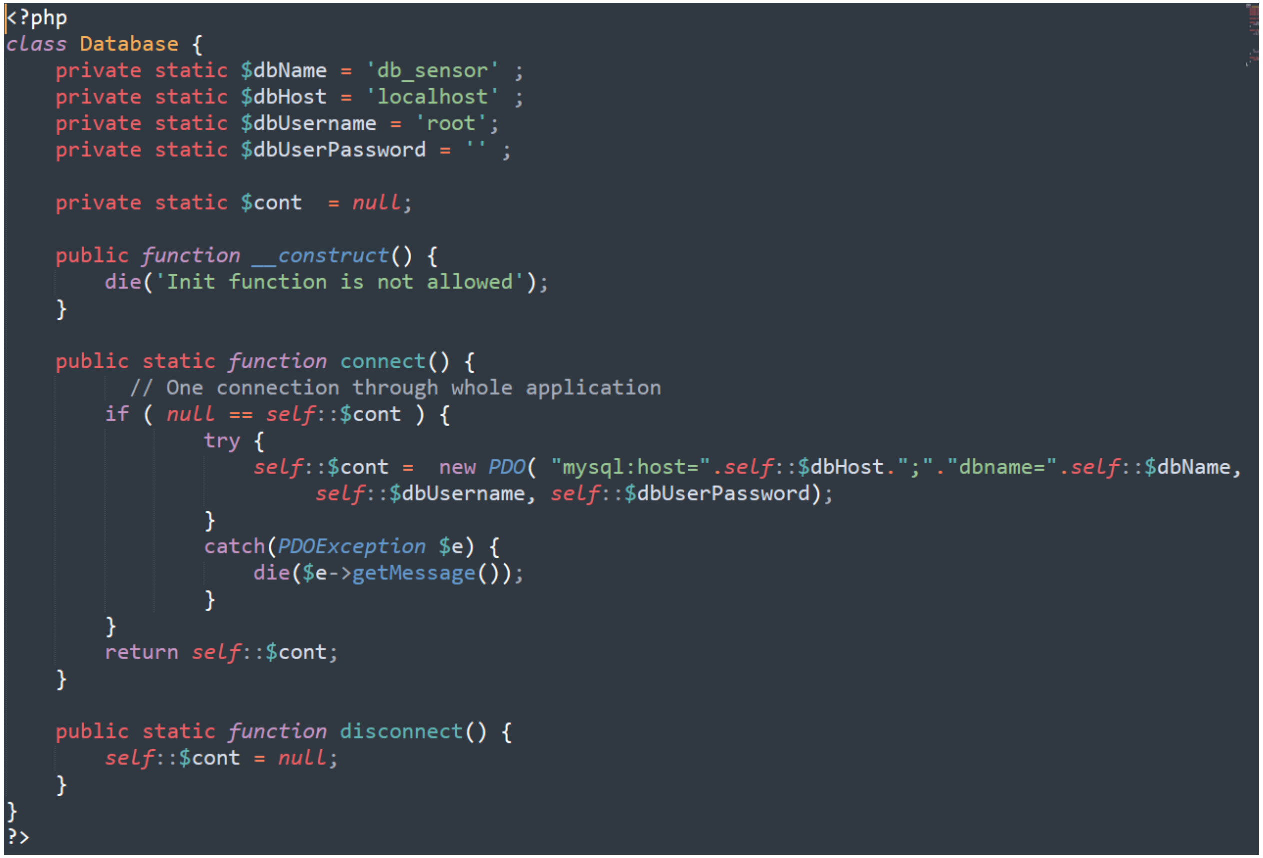
Save php file in htdoc Folder



› xampp › htdocs

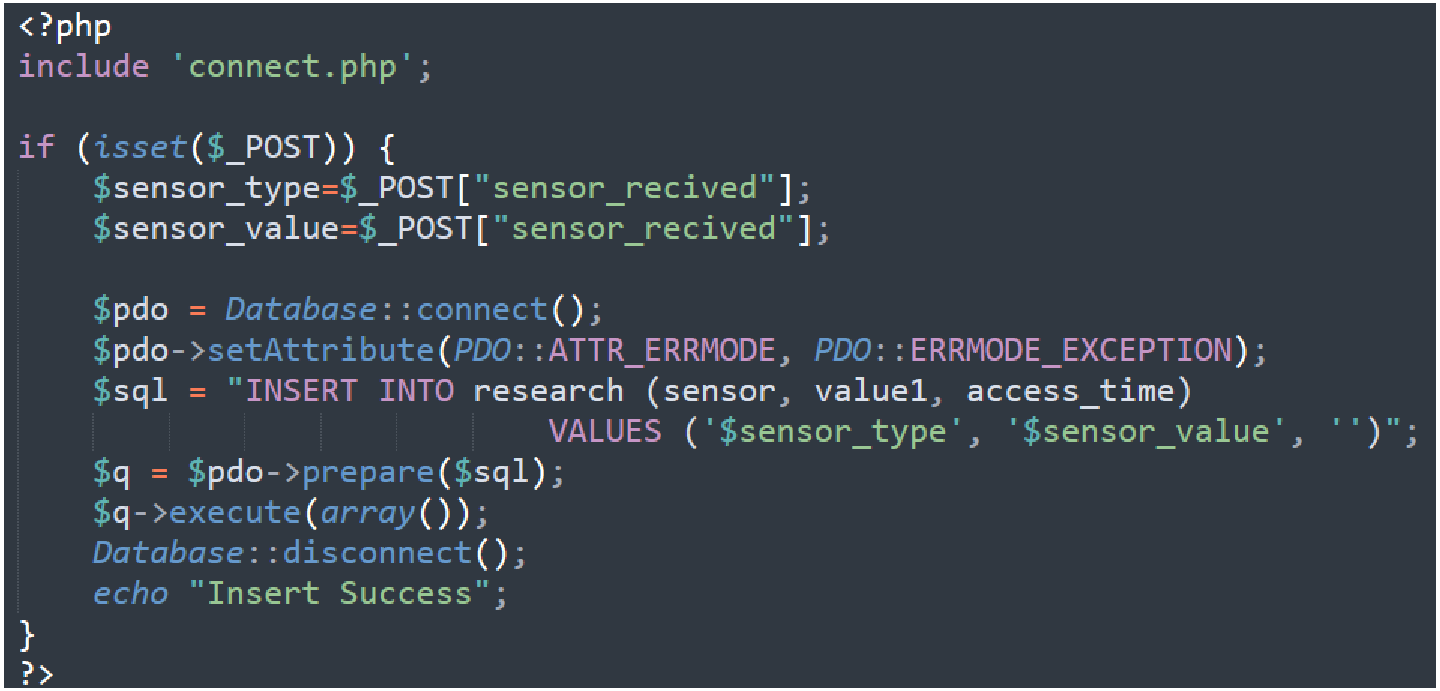
Nom

- dashboard
- img
- webalizer
- xampp
- applications.html
- bitnami.css
- connect.php
- favicon.ico
- index.php
- insert_temp.php
- Nouveau document texte.txt



```
<?php
class Database {
    private static $dbName = 'db_sensor' ;
    private static $dbHost = 'localhost' ;
    private static $dbUsername = 'root';
    private static $dbUserPassword = '' ;

    private static $cont  = null;

    public function __construct() {
        die('Init function is not allowed');
    }

    public static function connect() {
          // One connection through whole application
        if ( null == self::$cont ) {
                try {
                    self::$cont =  new PDO( "mysql:host=".self::$dbHost.";"."dbname=".self::$dbName,
                         self::$dbUsername, self::$dbUserPassword);
                }
                catch(PDOException $e) {
                    die($e->getMessage());
                }
        }
        return self::$cont;
    }

    public static function disconnect() {
        self::$cont = null;
    }
}
?>
```

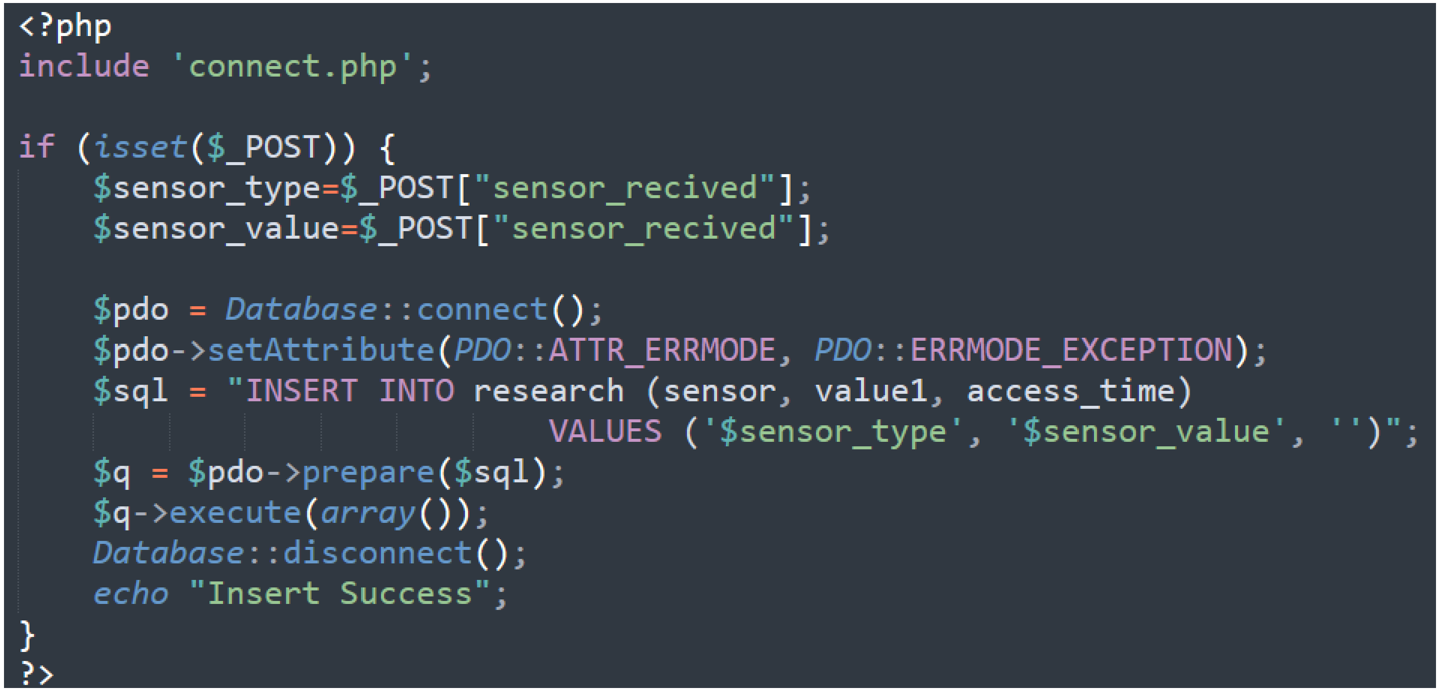
ESP 32 connection to database



```php
<?php
include 'connect.php';

if (isset($_POST)) {
    $sensor_type=$_POST["sensor_recived"];
    $sensor_value=$_POST["sensor_recived"];

    $pdo = Database::connect();
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO research (sensor, value1, access_time)
                            VALUES ('$sensor_type', '$sensor_value', '')";
    $q = $pdo->prepare($sql);
    $q->execute(array());
    Database::disconnect();
    echo "Insert Success";
}
?>
```

ESP 32 connection to database

- **Create a new file in xampp/htdoc with this exact name and extension: post_sensor_data.php**



```php
<?php
include 'connect.php';

if (isset($_POST)) {
    $sensor_type=$_POST["sensor_recived"];
    $sensor_value=$_POST["sensor_recived"];

    $pdo = Database::connect();
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO research (sensor, value1, access_time)
                            VALUES ('$sensor_type', '$sensor_value', '')";
    $q = $pdo->prepare($sql);
    $q->execute(array());
    Database::disconnect();
    echo "Insert Success";
}
?>
```

ESP 32 connection to database

- Now we return to the Vs code.
- we need to add some libraries to connect to the wifi and make http requests to access the database. we can use separate files to lighten the main code.
  - ➢ Add `#include "main_function.h"`
- In the main_fuction file, we find these libraries

```
#include <WiFiClient.h>
#include <HTTPClient.h>
#include <WiFiUdp.h>
#include <NTPClient.h> //https://github.com/taranais/NTPClient
```

- Download the **NTPClient** library from the given link.
- Extract it and put it in **your_project/lib** folder.
- Make sure that the library that you download contain the .h file in first folder which corresponds to `#include <NTPClient.h>` (VsCode rule).

Imperial College
London

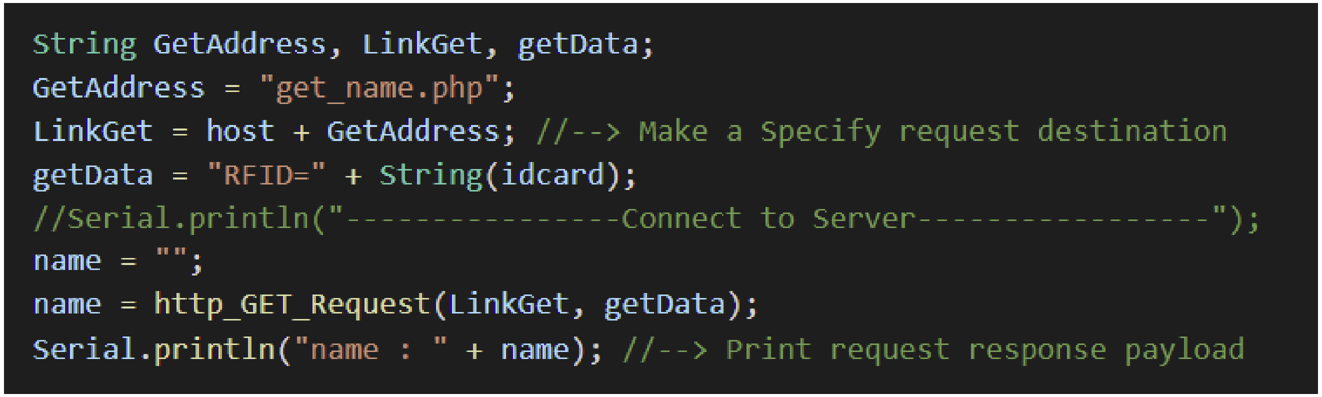
ESP 32 connection to database

- **In the main.cpp file, make sure that all these libraries are included.**

```
#include <Arduino.h>
#include <stdint.h>
#include <SPI.h>
#include <MFRC522.h>
#include <string.h>
#include <WiFi.h>
#include <Wire.h>
#include "main_function.h“
```

- **Define the host variable in which you put your ip address**

```
const char  *host = "http://your-ip-adress/" ;
```



```
String GetAddress, LinkGet, getData;
GetAddress = "get_name.php";
LinkGet = host + GetAddress; //--> Make a Specify request destination
getData = "RFID=" + String(idcard);
//Serial.println("----------------Connect to Server-----------------");
name = "";
name = http_GET_Request(LinkGet, getData);
Serial.println("name : " + name); //--> Print request response payload
```

Sockets problem

If there is a socket problem, follow the steps in this link

https://windowsreport.com/windows-sockets-registry-entries-required-for-network-connectivity-are-missing/#2

ESP32 OTA (Over-the-Air) Updates

- OTA (Over-the-Air) update using the AsyncElegantOTA library is the process of loading new firmware to the ESP32 board using a Wi-Fi connection rather than a serial communication.
- This functionality is extremely useful in case of no physical access to the ESP32 board.
- The Async Elegant OTA library creates a web server that allows the update of new firmware (a new sketch) to the ESP32 board without the need to make a serial connection between the ESP32 and the computer.
- Additionally, with this library, we can also upload new files to the ESP32 filesystem (SPIFFS).
- The files you upload should be in .bin format.

ESP32 OTA (Over-the-Air) Updates

To add OTA capabilities to your projects using the AsyncElegantOTA library, follow these steps:

1. **Iclude the AsyncElegantOTA, AsyncTCP and ESPAsyncWebServer libraries in the platformio.ini file of your project;**

```
lib_deps = ESP Async WebServer
           ayushsharma82/AsyncElegantOTA @ ^2.2.5
```

ESP32 OTA (Over-the-Air) Updates

2. **Include AsyncElegantOTA library at the top of the code:**

```
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <AsyncElegantOTA.h>
AsyncWebServer server(80);
```

3. **Add this line AsyncElegantOTA.begin(&server); before server.begin();**

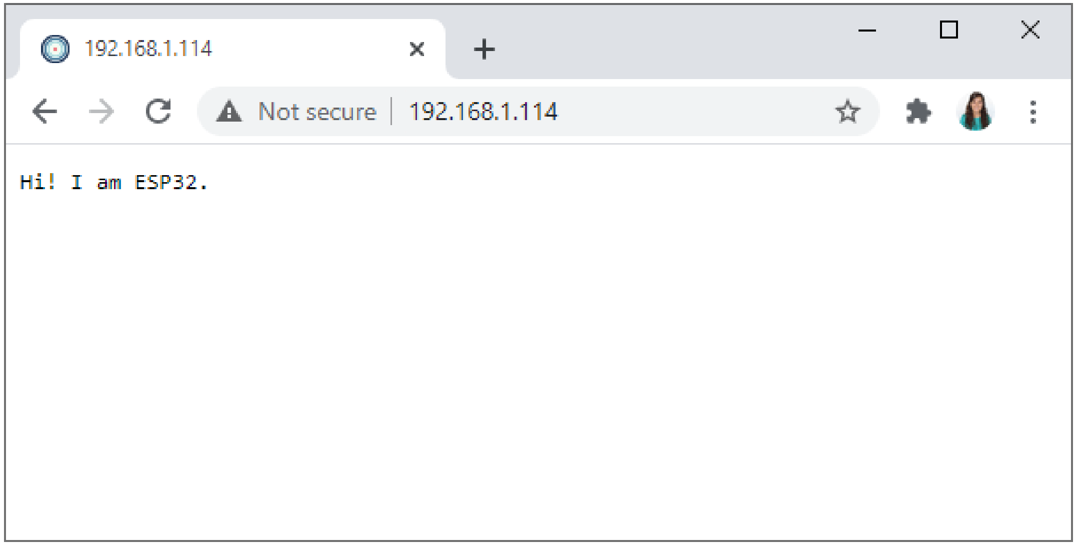
```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {          request-
>send(200, "text/plain", "Hi! I am ESP32.");

});

AsyncElegantOTA.begin(&server); // Start ElegantOTA

server.begin(); //initialize the server:
```

ESP32 OTA (Over-the-Air) Updates

- **The first sketch should be uploaded via the serial port.**
- **This sketch should contain the code to create the OTA Web Updater so that you are able to upload the code later using your browser.**
- **The OTA Web Updater sketch creates a web server you can access to upload a new sketch via a web browser.**
- **Then, you need to implement OTA routines in every sketch you upload, so that you're able to do the next updates/uploads over-the-air.**
- **If you upload a code without an OTA routine, you'll no longer be able to access the web server and upload a new sketch over-the-air.**

ESP32 OTA (Over-the-Air) Updates

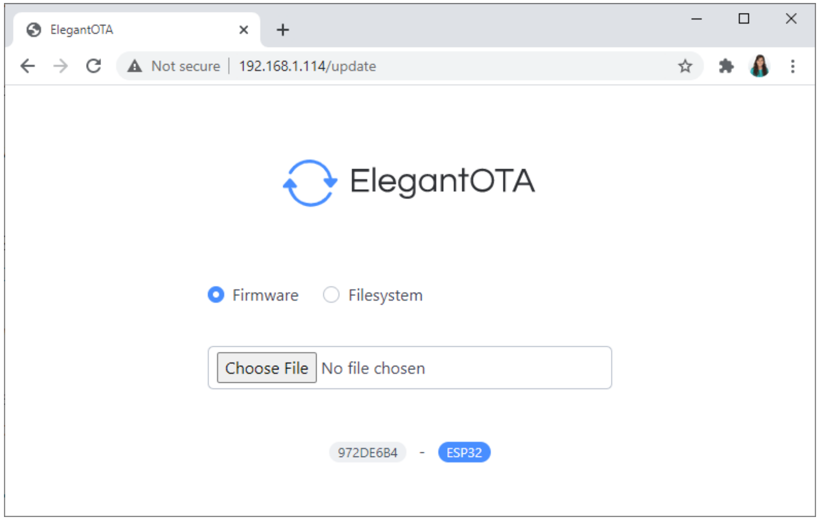
- In the local network, open the browser and type the ESP32 IP address.
- We should get access the root (/) web page with some text displayed.



```
Hi! I am ESP32.
```

ESP32 OTA (Over-the-Air) Updates

4. Open the browser and go to http://<IPAddress>/update, where <IPAddress> is the ESP32 IP address.
   - The following web page should load.



ElegantOTA

Not secure | 192.168.1.114/update

● Firmware ○ Filesystem

Choose File No file chosen

972DE6B4 - ESP32

- VS Code automatically generates the .bin file for the project every code compilation.
- The file is called firmware.bin and it is saved in this root.
- .pio/build/esp32doit-devkit-v1/firmware.bin