**Course Project – Module 2**

**College of Engineering and Information Sciences**

**Course Number: CEIS114**

Background

## The Internet of Things is growing exponentially. New technologies and applications are being developed on a regular basis and this creates an abundance of new job opportunities each with its requisite skill sets. It is important to stay current with the skills required for this evolving job market and to take advantage of the many available learning opportunities currently available. This course project covers the fundamentals topics of IoT and is divided into six parts. Each week builds upon the project resulting in an IoT device that will simulate a traffic controller. The design and development process of the device will include planning, hardware setup, programming, and networking. It will encompass the many aspects of the IoT and prepare you for your future career in technology.

Scenario

## Design and develop a two-way traffic light controller with a pedestrian crossing and an emergency signal. The pedestrian crossing will have an LCD display messages whether the pedestrian can or cannot cross the street. When they can cross, the LCD will count down from ten (10) to zero (0) indicating the amount of time they have to cross. The emergency signal will force red lights to turn on in all directions and the blue light will be flashing. The final step is to secure the system so that it could be controlled remotely via a web browser. The alternative final step is a non-internet connection using a motion sensor to allow traffic to continuously flow on major street and only switches to slow traffic street when motion is detected.

# **Project Plan for IoT Traffic Controller**

# Objectives

* To introduce the course project
* To prepare materials for course project
* To Install the **ESP32S\_NodeMCU** Board in Arduino IDE

Key Performance Indicators:

* Problem solving methodology utilized with no errors
* Communication skills were demonstrated through a report with no errors
* Deliverables were submitted on time with no errors

**Overall project PROCEDURE**

Design and implement a four-way traffic light controller system with an option for pedestrian crossing utilizing the IoT kit and ESP32 Microcontroller. Develop a project flow diagram with the deliverables listed below:

1. Setup ESP32with one traffic light
2. Setup ESP32with two traffic lights
3. Add a pedestrian crossing option for the system
4. Display alert signs for the pedestrian to walk or not to walk
5. Add an emergency button to emulate police or ambulance control of the traffic lights
6. Secure and control the lights via Web application.

**Alternatively, the final step in the project flow diagram could be:**

1. Develop a non-internet connection option to design a motion detector to allow traffic to continuously flow on major street and only switches to slow traffic street when motion is detected.

**Module 2 project PROCEDURE**

1. Gather the project inventory listed in the Project Template PowerPoint.
2. Install and Test the **ESP32S\_NodeMCU** Board in Arduino IDE. See **Appendix A** for the detailed steps.

# **Module 2 Project Deliverables**

* Complete the Course Project PowerPoint Deliverable
* Include a picture of your inventory
* Include a picture of the ESP32 mounted on the breadboard and powered ON
* Include a screenshot of the scanned WiFi networks

**Appendix A**

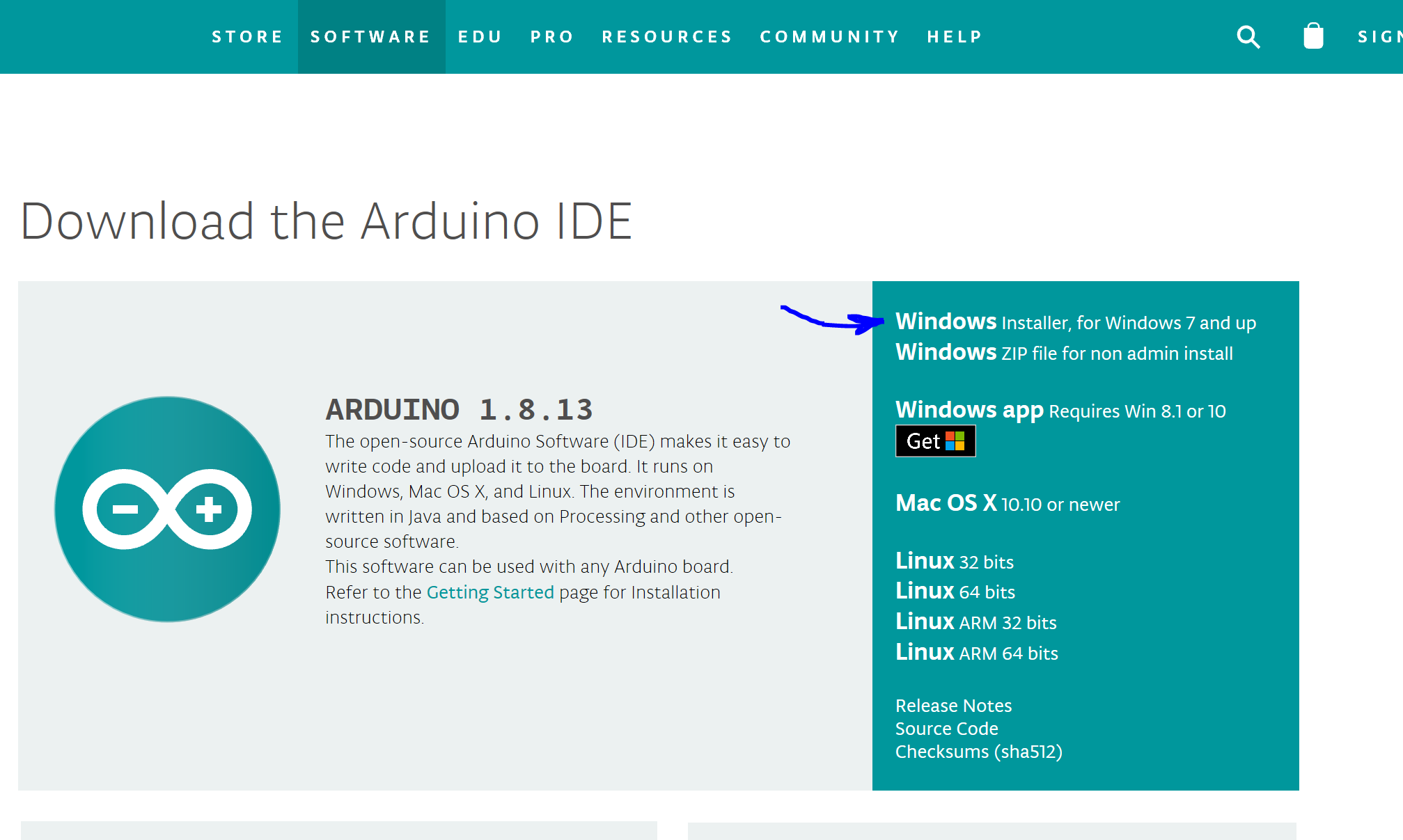
# **Installing the ESP32 Board in Arduino IDE**

# **(Windows instructions)**

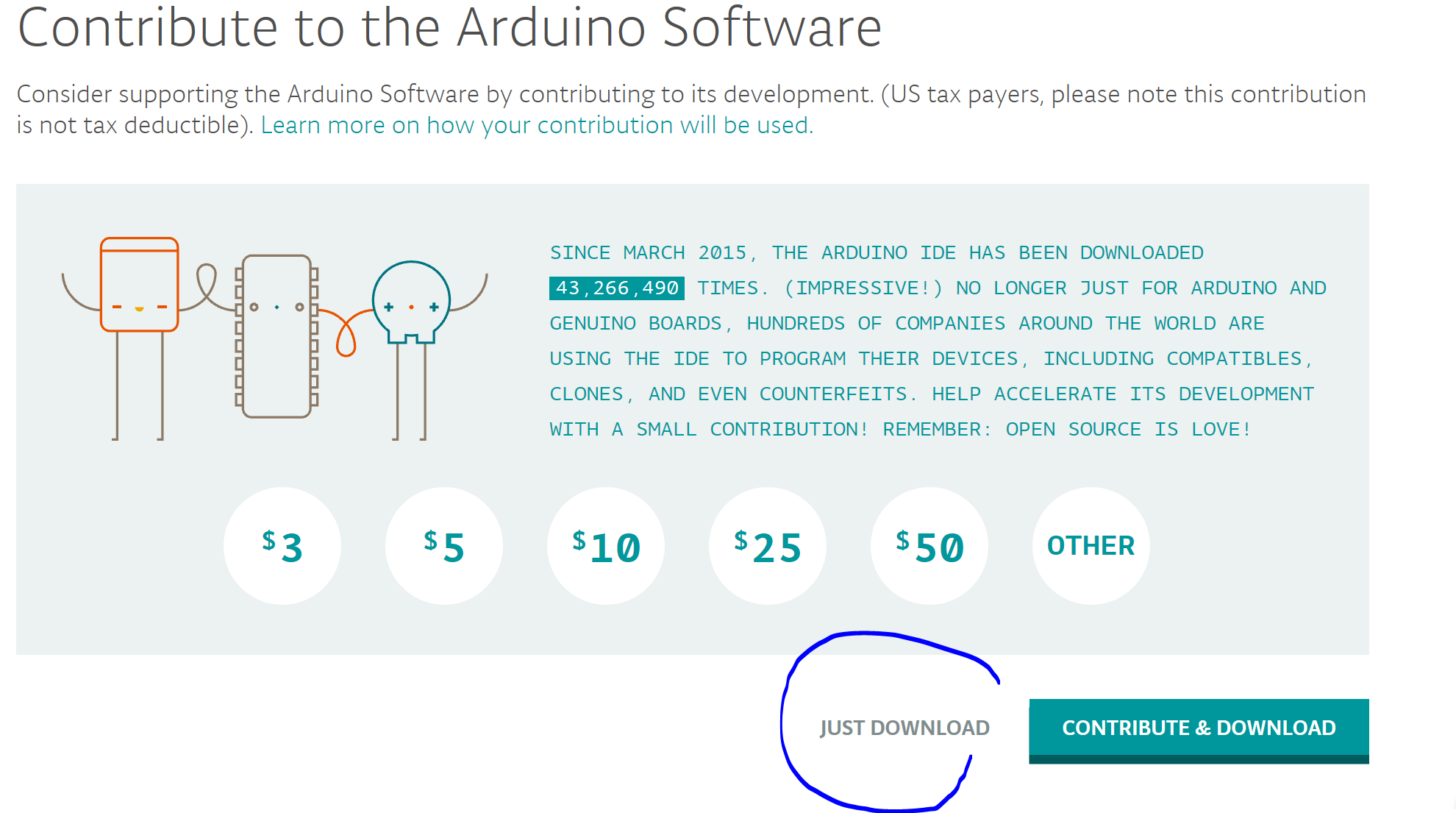
**Important:** before starting this installation procedure, make sure that you have the latest version of the Arduino IDE installed in your computer. If you don’t, uninstall it and install it again. Otherwise, it may not work. Having the latest Arduino IDE software installed from [**arduino.cc/en/Main/Software**](https://www.arduino.cc/en/Main/Software), continue with this tutorial.

To download and Install Arduino. Go to : [**arduino.cc/en/Main/Software**](https://www.arduino.cc/en/Main/Software)

**Click on Windows installer as shown below**

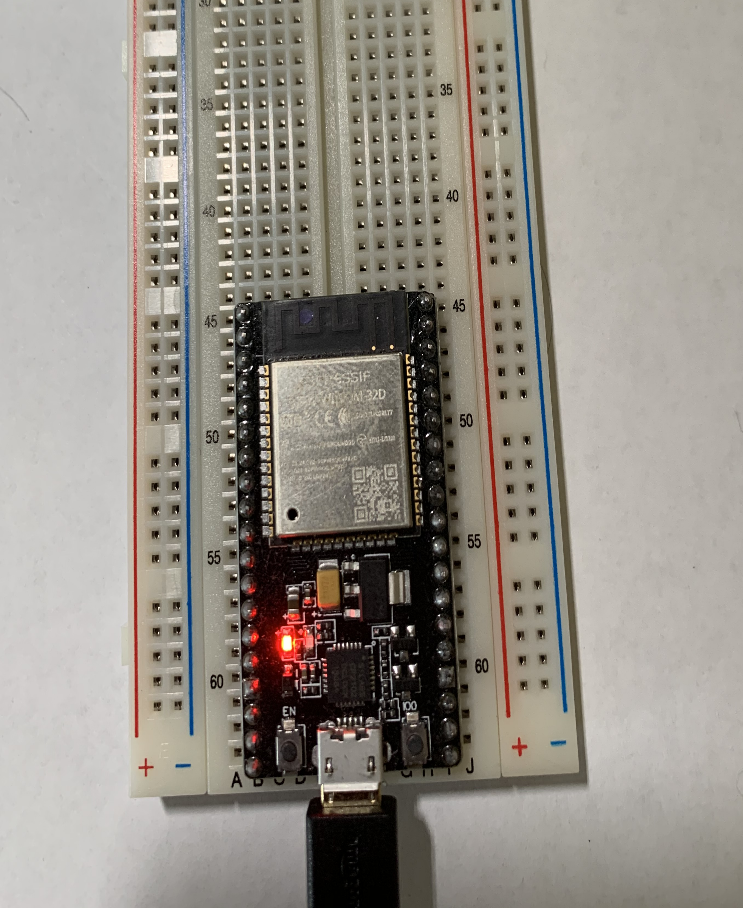


**A new window will open (see below) click on Just download as shown below:**



**IMPORTANT NOTE**:

1. If this is your first time installing the ESP32 on the Arduino IDE, simply connect your ESP32 to your computer and follow the installation procedure described below:

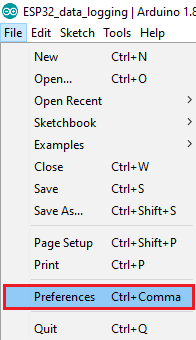


1. If you’ve already installed the ESP32 add-on using the old method, you should remove the *espressif*folder first. Go to the end of this post to learn how to remove the *espressif*folder.

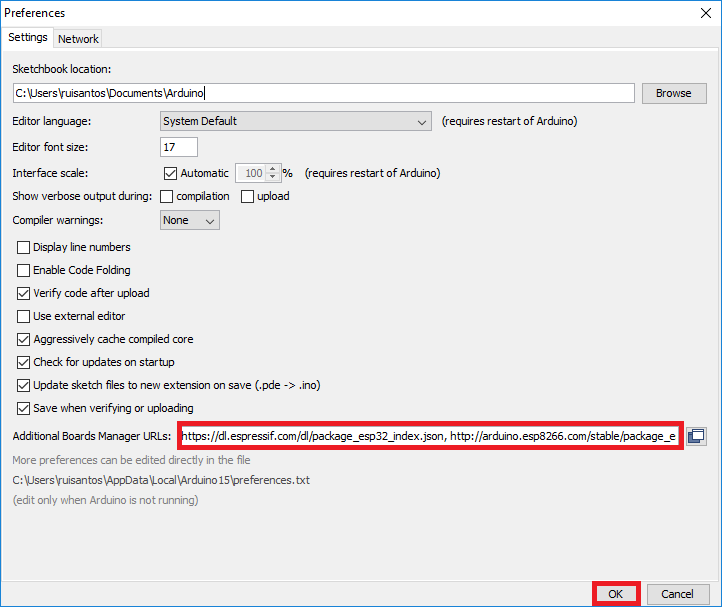
**1. Installing the ESP32 Board**

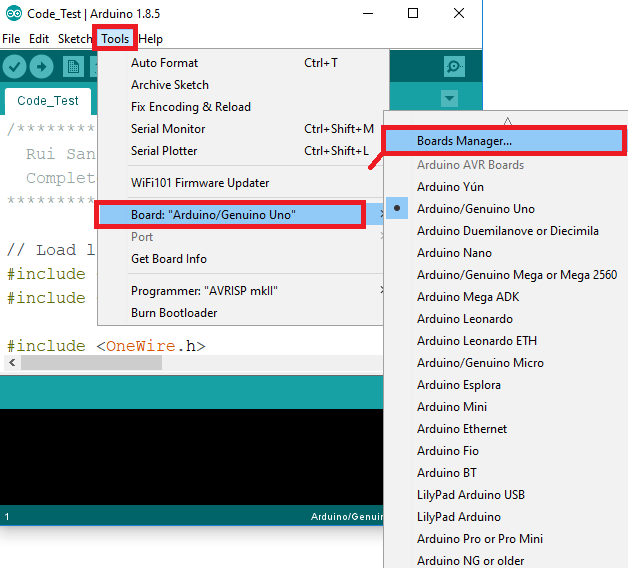
To install the ESP32 board in your Arduino IDE, follow these next instructions:

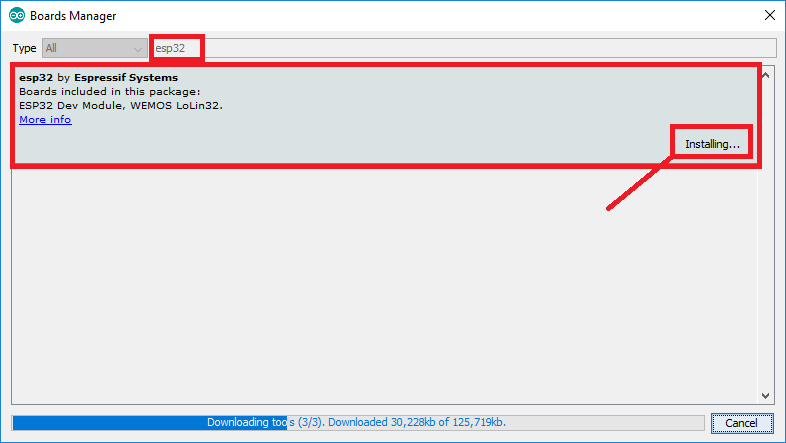
1) Open the preferences window from the Arduino IDE. Go to **File**> **Preferences**



2) Enter**https://dl.espressif.com/dl/package\_esp32\_index.json** into the “Additional Board Manager URLs” field as shown in the figure below. Then, click the “OK” button:



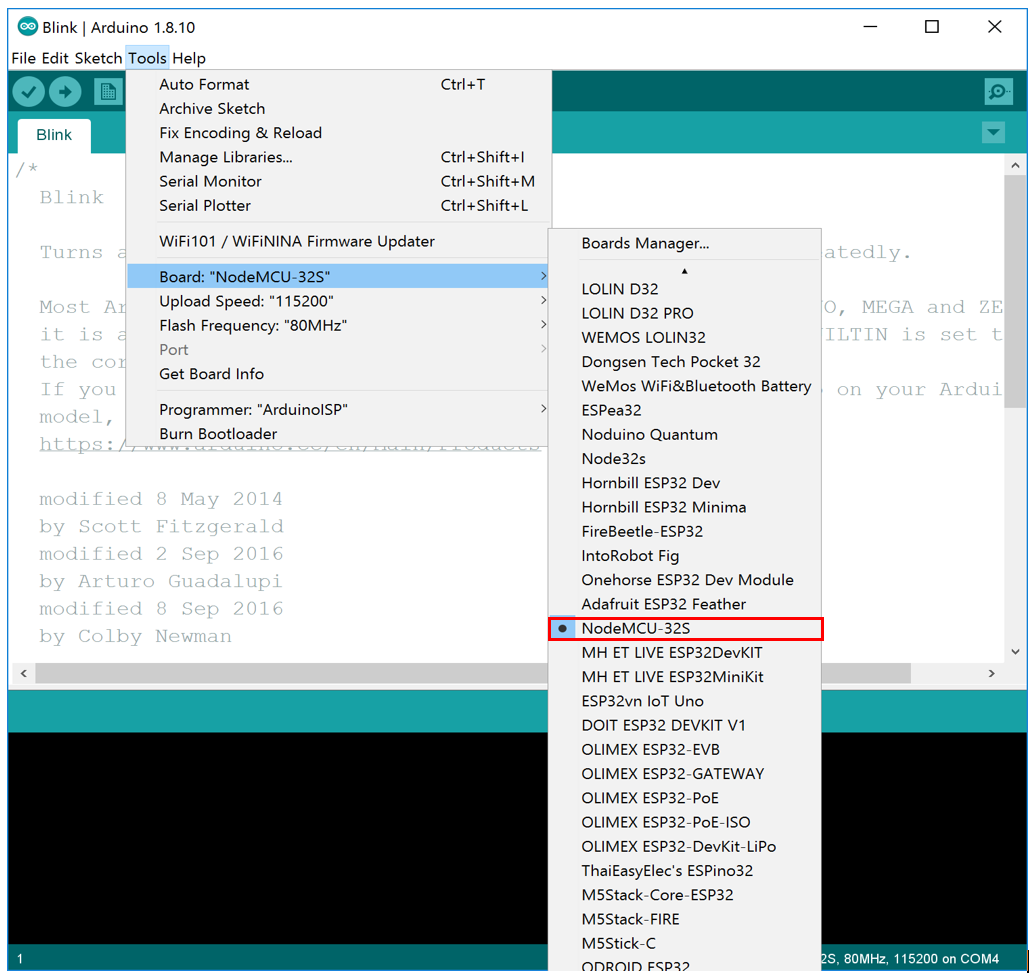
3) Open boards manager. Go to **Tools** > **Board** > **Boards Manager…**

4) Search for ESP32 and press install button for the “**ESP32 by Espressif Systems**“:

5) That’s it. It should be installed after a few seconds.

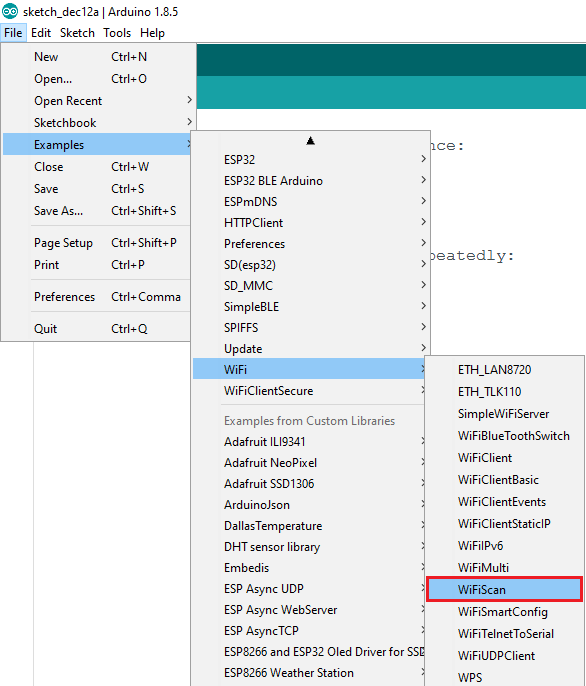
**Appendix B**

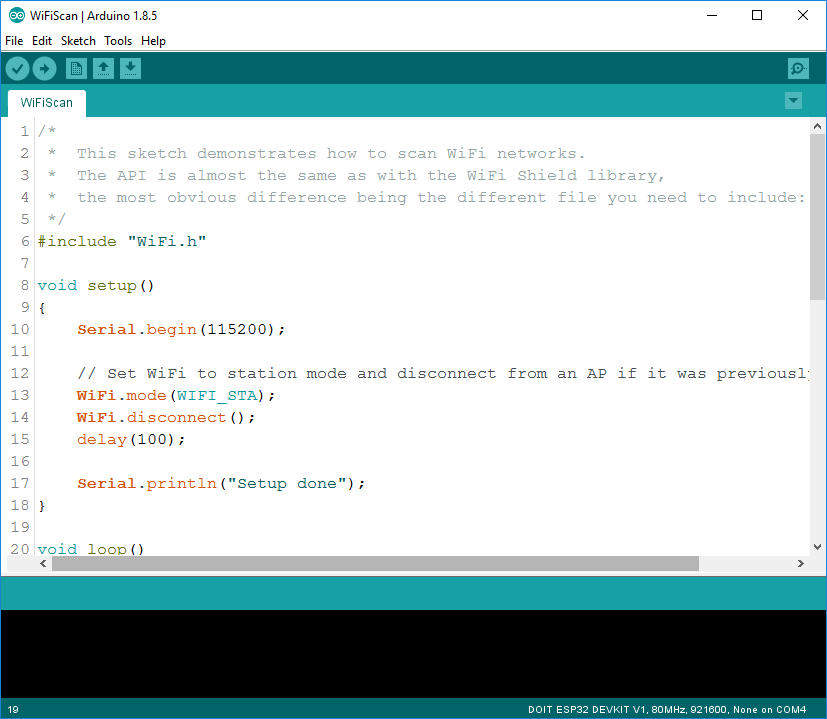
**Testing the Installation**

Plug the ESP32 board to your computer. Then, follow these steps:  
  
1) Open the Arduino IDE  
  
2) Select your Board in **Tools** > **Board** menu (in my case it’s the **NodeMCU-32S**

3) Select the Port (if you don’t see the COM Port in your Arduino IDE, you need to install the [ESP32 CP210x USB to UART Bridge VCP Drivers](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers)):

4) Open the following example under **File** > **Examples** > **WiFi (ESP32)** > **WiFi Scan**

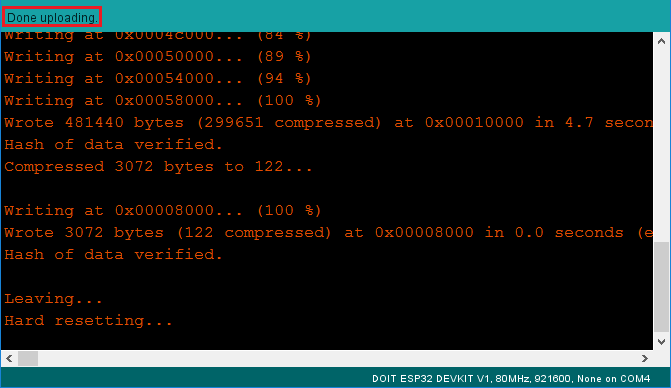


5) A new sketch opens:  


6) Press the **Upload** button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.

https://i2.wp.com/randomnerdtutorials.com/wp-content/uploads/2016/12/arduino-ide-upload-button.png?resize=34%2C29&ssl=1

7) If everything went as expected, you should see a “**Done uploading.**” message.



8) Open the Arduino IDE Serial Monitor at a baud rate of 115200:

https://i0.wp.com/randomnerdtutorials.com/wp-content/uploads/2016/12/mac-open-arduino-ide-serial-monitor.png?resize=38%2C29&ssl=1

9) Press the ESP32 on-board **Enable** button and you should see the networks available near your ESP32:

