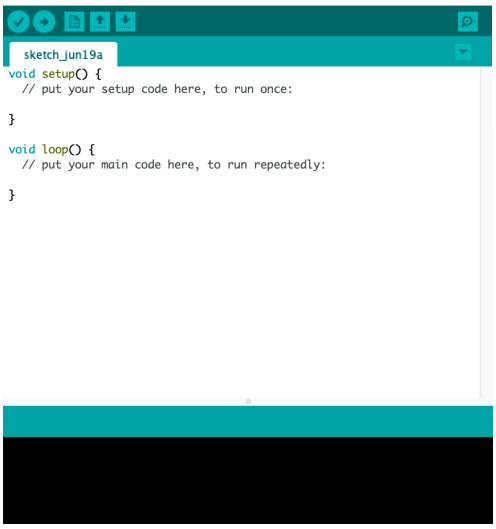
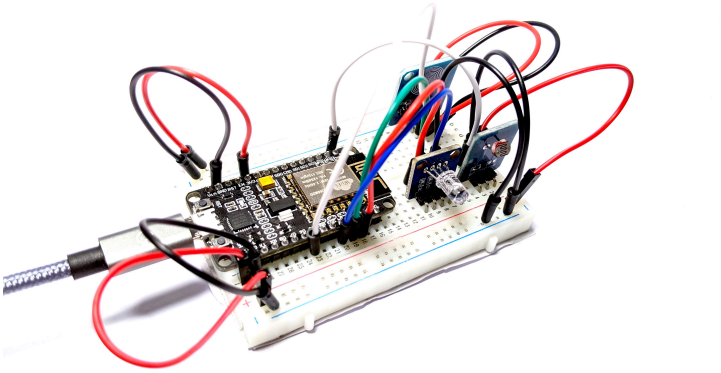
Over The Air

Loading and updating code wirelessly

Project description:

In this project, you will learn how OTA (over the air) can be used to load new code, or update existing code, to your ESP board wirelessly. This will show you an example of how you can control your ESP board codebase without the board being physically wired to your computer. You might have already explored such method in the Home Assistant exercise section of this course. There is no need for you to build a separate circuit for this exercise. You will use the smart chair circuit that you have built during the course and upload a different code sketch.



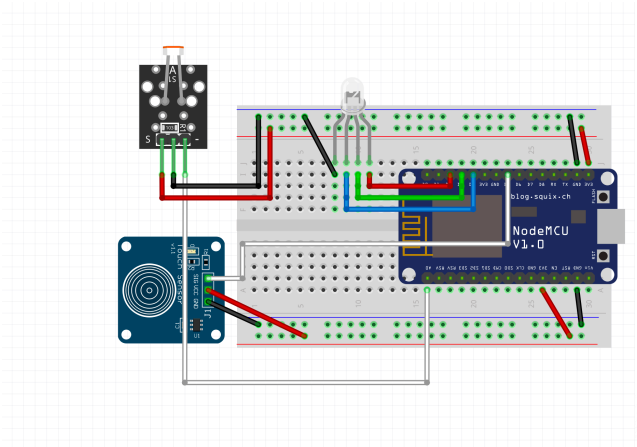


Project objectives:

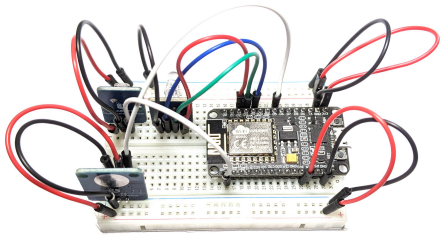
- Load the OTA library sketch on the ESP board
- Configure and select the wireless port
- Use the new wireless port to upload/update code wirelessly

Project components:

Component Reference	Component Quantity	Component Name	Component Description
A	1	Smart Chair circuit	The smart chair circuit which features the photoresistor, the RGB LED, and the touch sensor
B	1	Micro USB Cable	A USB cable to power and upload instructions to a microcontroller



A



B

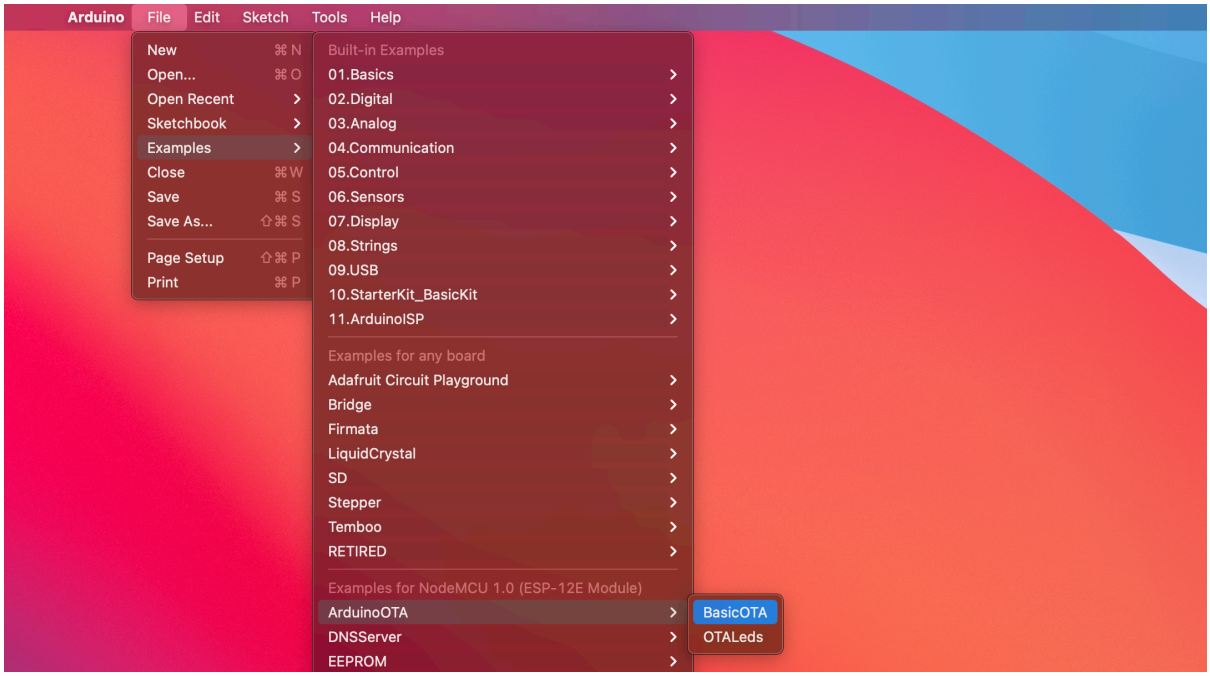
Step One
The OTA Library

So far, you have always loaded new code, or have updated existing code, via a USB cable. There is certainly nothing wrong with such method but would it be great if you could update you ESP board wirelessly?

The answer to the above question is Over The Air updating, or OTA for short. As the name implies, this technology allows you to upload new code over Wi-Fi, instead of Serial.

Before you can send new code, or update existing code wirelessly, you first need to upload the OTA sketch via USB. Let's see how you can achieve this.

Go ahead and open the **BasicOTA** sketch from the Arduino examples section as show below: File > Examples > ArduinoOTA > BasicOTA



The **BasicOTA** sketch is a built-in program that allows you to enable the OTA protocol on your ESP board using the local WiFi.

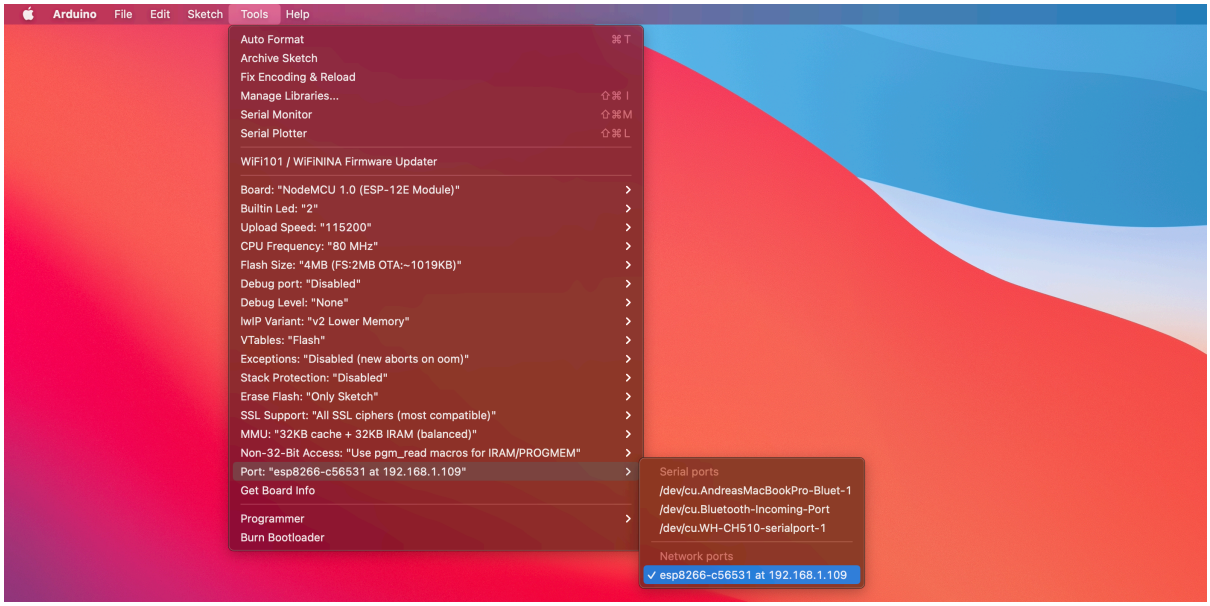
There is only one thing that you really need to change in the example sketch for the OTA protocol to work: the WiFi credentials.

Go ahead and change the **STASSID** (name of Wifi network) and the **STAPSK** (password) variable values with your WiFi details**:**

```
#ifndef STASSID
#define STASSID "[redacted]"
#define STAPSK  "[redacted]"
#endif
```

Now, simply upload the sketch to your smart chair circuit via USB as you have always done until now.

In order to see if the OTA protocol was uploaded successfully, you can first check the Serial Monitor output (make sure you set the baud rate to 115200), and then access the Port section of the Arduino IDE to see if there is a listed port under **Networks port** as shown below: Tools > Port



If you can see a port listed under **Networks ports**, the OTA upload was successful. Select it as this is the port that you can use to upload the next sketch wirelessly.

Great, you have now successfully configured the OTA protocol. Let's create a simple sketch for the smart chair circuit and upload it over WiFi.

Step Two
A simple sketch for the smart chair circuit

Now that the OTA configuration is ready for you to wirelessly upload a new sketch to the smart chair circuit ESP board, you can write the code for the new sketch.

Go ahead and create a new empty sketch from the Arduino IDE.

You should see an empty sketch like the following:

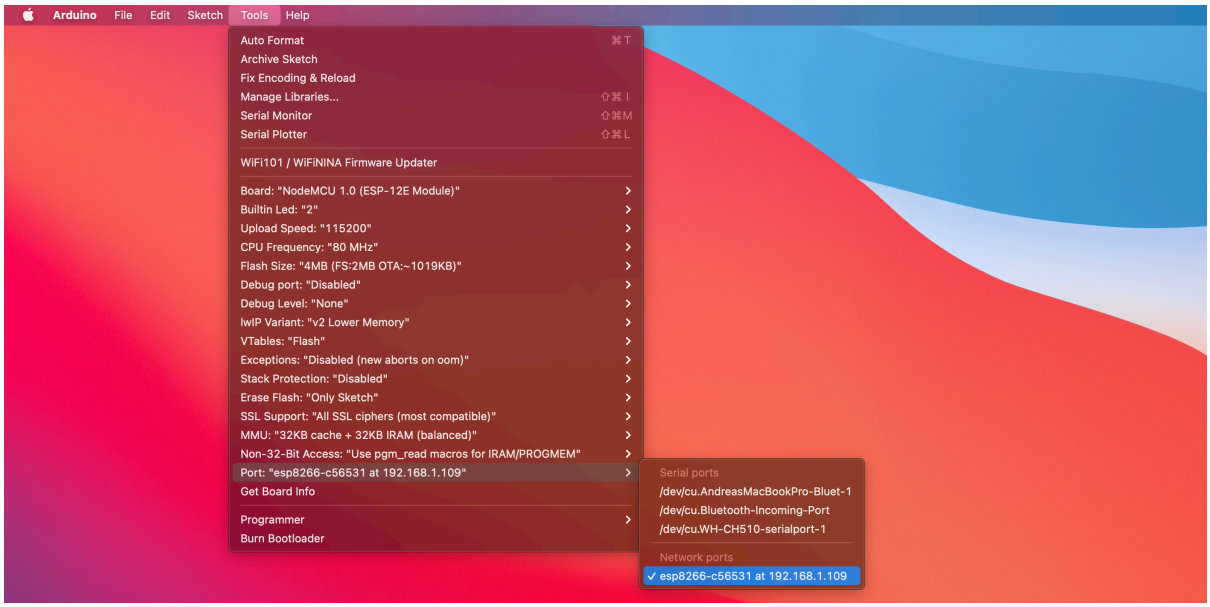
Feel free to save the sketch and rename it to something sensible:
smart_chair_ota for instance.

Here, you can be creative and code a simple behaviour for the smart chair circuit. I would suggest to reuse the code that you wrote for the touch sensor and the RGB module to simulate the chair busy/free system.

Here a reminder of the steps to create such behaviour:

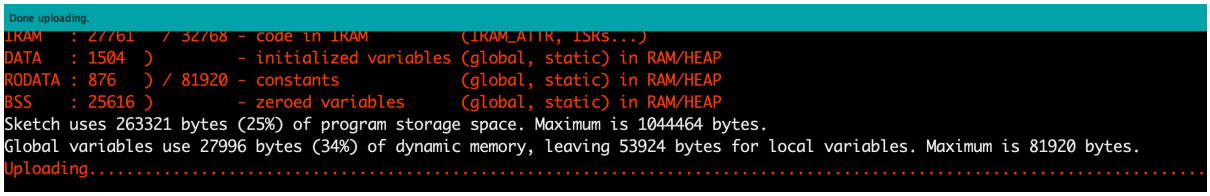
- Initialise an integer variable for the touch sensor and assign the pin value D5.
- Initialise an integer variable to store the value read from the touch sensor.
- Initialise integer variables for the RGB module and assign the pin values D2, D3, and D4.
- In the **setup()** function, set the appropriate pinMode for the touch sensor and the RGB module.
- Write a utility function called **chairSignal()**. Such function should change the RGB module to red when the chair is busy (HIGH read from the touch sensor) and to green when the chair is free (LOW read from the touch sensor).
- Call **chairSignal()** in the **loop()** function.

Now that you have you sketch ready for upload, make sure to select your ESP WiFi port if you have not done that previously:



Unplug the smart chair circuit from your computer.
You can power up the circuit with a power bank or with a spare phone charger.

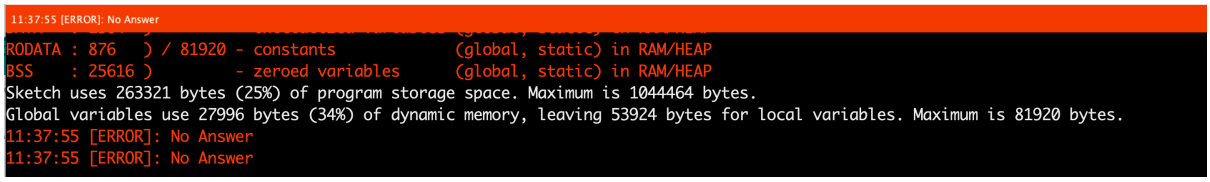
Press the upload button on the Arduino IDE sketch and watch your code magically upload to your ESP board via WiFi. You will see a slightly different output on the console, wait for it to finish the upload:



Congratulations, you have now completed your first OTA code upload.

Step Three
Additional Tasks

You may notice that the OTA upload only works on the first upload. If you try to upload any code again, you will see the following error printed to the console:



This is because the OTA code needs to be integrated with any sketch that you are planning to upload/update over the air.

Your task is to merge the OTA example sketch with the code that you just wrote for the smart chair circuit. This will allow you to upload/update the code wirelessly as many times as you want.

Hint: You will need to switch back the Port to serial in order to upload the merged code. Successively you can permanently switch to OTA Port for further upload/updates.