



Over-the-Air (OTA) Programming on ESP32 Using Arduino IDE and Web Updater OTA



by Asis Gotam

In this article, we will be discussing about Over-The-Air programming on our famous ESP32 Development board. We have previously built many [ESP32 Projects](#) and having an option to program these boards without having to connect them to computer every single time is really interesting. So in this article, we will also be looking into the two ways which can be

used to program ESP32 wirelessly. One is the basic OTA method which requires an Arduino IDE and the other is a Web Updater OTA method which only requires a browser to update your program. We will look into both the methods and understand how to use them, so tune in till the end to learn everything.

<https://youtu.be/IMKqQtkIRXI>

Step 1: What Is OTA?

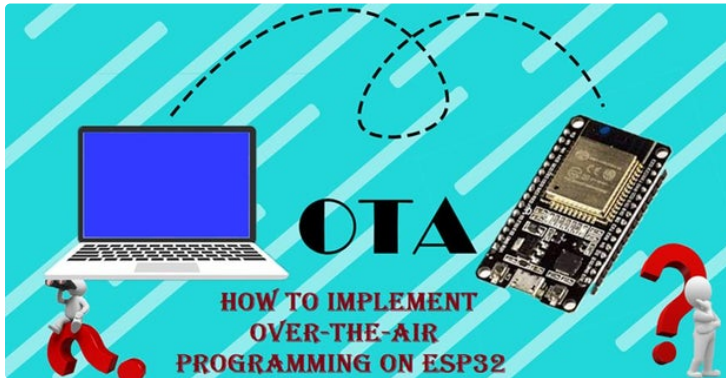
OTA stands for Over The Air. It allows uploading a new program to ESP32 using Wi-Fi instead of connecting the ESP32 to a computer via USB to do the update. It is extremely useful in situations where you don't have physical access to the module. This feature can be used to reprogram the devices that are fixed on a roof or any other place where connecting cable is difficult.

Step 2: What Is the Need of OTA?

Let us say, you have created a phenomenal project using ESP32. You fixed it in your home and powered it up. Everything is working fine and you have got comfortable in your smart home. But one day, you need to make slight changes to the code or maybe change the code with an updated version of libraries and kinds of stuff or maybe add some more features to it. Will you like the mess of plugging out the whole

module, removing the connections, plugging it to your PC, uploading the code, re-doing the connections, and then setting it all up again?

Well no! No one will. That is where OTA kicks in. For establishing OTA with ESP32, you need to include few extra lines to your code and after that, you can upload the codes wirelessly over the air.



Step 3: How to Implement OTA With ESP32?

There are two ways to perform an OTA update on the ESP32 board. They are Basic OTA, in which updates are sent via Arduino IDE, and Web updater OTA, in which updates are sent via a webpage/web browser.

Step 4: How to Use Basic OTA on ESP32

For using the Basic OTA feature with your ESP32 board, you just need to follow these three steps. This method uses Arduino IDE to program our ESP32 board, so if you are completely new to ESP32 and Arduino, you can check out [how to get started with ESP32 using Arduino IDE](#) to understand better.

Step 1:- Installing Python in Your PC

In order to use the OTA functionality, you need to have [Python 2.7.xx](#) installed on your PC. You can [download python](#) from the official website.

The installation process is pretty straight forward. Just open the installer after downloading and follow the installation wizard. After installing Python on your PC, you can continue with the second step.

Step 2:- Uploading the Basic OTA Sketch

The ESP32 board does not support OTA updates by default, hence you need to upload the new firmware in order to perform the OTA updates. This is a mandatory step as this will later allow you to push the new codes wirelessly via Wi-Fi. Firstly, you need to

have the ESP32 board package installed in your Arduino IDE. After installing the boards, go to

File > Examples > ArduinoOTA > BasicOTA

This is the sketch that you need to install in order to give your ESP32 the power of OTA.

The changes that you need to make in the code are SSID and password. You need to give these credentials so that the ESP can connect to your router in order to receive the updates. Once you are done with those, go ahead and upload the sketch.

Once you have uploaded the Basic OTA sketch. Open your Serial Monitor on a baud rate of 115200. If everything went alright, then you will see the IP Address printed at the end of the monitor. Note the IP address.

Now, that you have completed all the steps. It's time to upload the new sketches over the air. Excited? So am I.

Step 3:- Uploading the Codes Over The Air

Now let us try uploading a new sketch wirelessly to ESP32. Remember that you need to include the Basic OTA sketch with every sketch that you upload else you'll lose the OTA functionality after uploading the sketch. For example, if you want to upload a basic LED blinking sketch too, then you need to include the Basic OTA sketch in it also. An example of such sketch is given below.

```
#include
```

```
#include
```

```
#include
```

```
#include
```

```
const char* ssid = ".....";
```

```
const char* password = ".....";
```

```
//This is the part of the LED blinking code
```

```
const int led = 2; //ESP32 Pin to which onboard LED is connected
```

```
unsigned long previousMillis = 0; //will store last time LED was updated
```

```
const long interval = 1000; //interval at which to blink (milliseconds)
```

```
int ledState = LOW; //ledState used to set the LED
```

```
void setup() {
```

```
pinMode(led, OUTPUT);
```

```
Serial.begin(115200);
```

```
Serial.println("Booting");
```

```
WiFi.mode(WIFI_STA);
```

```
WiFi.begin(ssid, password);
```

```
while (WiFi.waitForConnectResult() != WL_CONNECTED) { }
```

```
Serial.println("Connection Failed! Rebooting...");
```

```
delay(5000);
```

```
ESP.restart();
```

```
}
```

```
//Port defaults to 3232
```

```
//ArduinoOTA.setPort(3232);
```

```
//Hostname defaults to esp3232-[MAC]
```

```
//ArduinoOTA.setHostname("myesp32");
```

```
//No authentication by default
```

```
//ArduinoOTA.setPassword("admin");
```

```
//Password can be set with it's md5 value as well
```

```
//MD5(admin) = 21232f297a57a5a743894a0e4a801fc3
```

```
//
```

```
ArduinoOTA.setPasswordHash("21232f297a57a5a743894a0e4a801fc3");
```

```
ArduinoOTA
```

```
.onStart([]() {
```

```
String type;
```

```
if (ArduinoOTA.getCommand() == U_FLASH)
```

```
type = "sketch";
```

```
else //U_SPIFFS
```

```
type = "filesystem";
```

```
// NOTE: if updating SPIFFS this would be the place to unmount SPIFFS using SPIFFS.end()
```

```
Serial.println("Start updating " + type);
```

```

.onEnd([]) {
    Serial.println("\nEnd");
}

.onProgress([(unsigned int progress, unsigned int total) {
    Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
}

.onError([(ota_error_t error) {
    Serial.printf("Error[%u]: ", error);

    if(error == OTA_AUTH_ERROR) Serial.println("Auth Failed");

    else if(error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");

    else if(error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");

    else if(error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");

    else if(error == OTA_END_ERROR) Serial.println("End Failed");

}]);

ArduinoOTA.begin();

Serial.println("Ready");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());
}

void loop() {
    ArduinoOTA.handle();

    //Here is the LED blinking section
    unsigned long currentMillis = millis();

    if(currentMillis - previousMillis >= interval) {

        //save the last time you blinked the LED

        previousMillis = currentMillis;

        // if the LED is off turn it on and vice-versa:

        ledState = not(ledState);

        //set the LED with the ledState of the variable:

        digitalWrite(led, ledState);
    }
}

}

Once you have written your program in such a manner that it includes the Basic OTA sketch as well, then just click on Tools and go to the ports section. You will see something like this "esp32-xxxxxx at your_esp_ip_address", Choose that board and hit on the upload button.

And Wallah! Just wait for few seconds and your code is sent to ESP32 Over The Air.

```

Looking for a specific release?

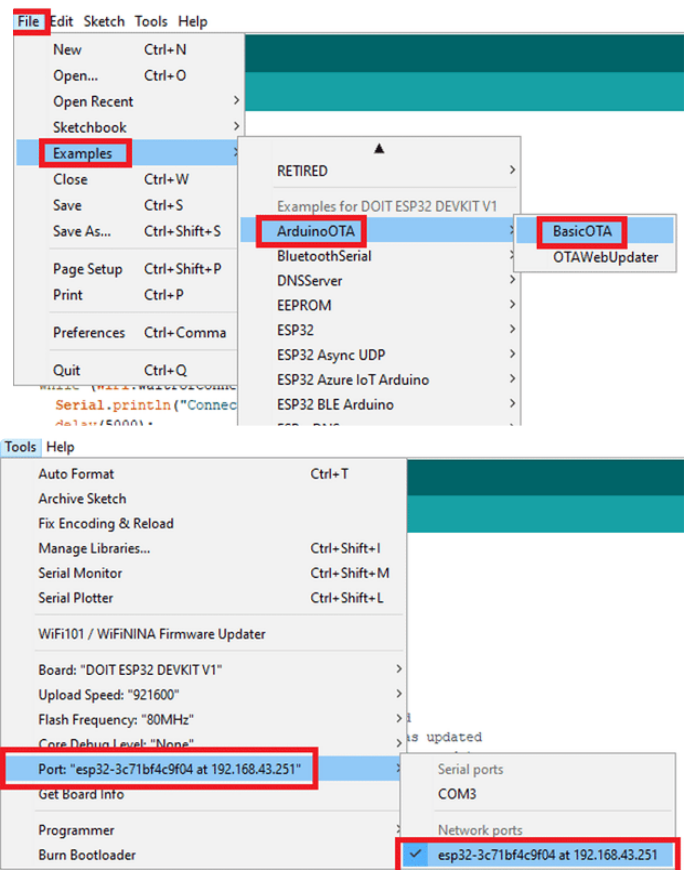
Python releases by version number:

Release version	Release date	
Python 3.6.11	June 27, 2020	Download
Python 3.8.3	May 13, 2020	Download
Python 2.7.18	April 20, 2020	Download
Python 3.7.7	March 10, 2020	Download
Python 3.8.2	Feb. 24, 2020	Download
Python 3.8.1	Dec. 18, 2019	Download
Python 3.7.6	Dec. 18, 2019	Download

COM3

ets Jun 8 2016 00:22:57

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:8896
load:0x40080400,len:5816
entry 0x400806ac
Booting
Ready
IP address: 192.168.43.251
```



Step 5: How to Use Web Updater OTA on ESP32

For using the Web-updater OTA feature with your ESP32 board. You just need to follow these three steps.

Step 1:- Uploading the WebUpdater Sketch

The ESP32 board does not support OTA updates by default, hence you need to upload the new firmware in order to perform the OTA updates. This is a mandatory step as this will later allow you to push the new codes wirelessly via Wi-Fi. Firstly, you need to have the ESP32 board package installed in your Arduino IDE. After installing the boards, Go to-

File > Examples > ArduinoOTA > BasicOTA

This is the sketch that you need to install in order to give your ESP32 the power of OTA. The changes that you need to make in the code are SSID and password. You need to give these credentials so that the ESP can connect to your router in order to receive the updates. Once you are done with those, go ahead and upload

the sketch.

Step 2:- Access the Web-Server Created by Your ESP32

Once you have uploaded the Basic OTA sketch. Open your Serial Monitor on a baud rate of 115200. If everything went alright, then you will see the IP Address printed at the end of the monitor. Note the IP address.

Visit this IP address with any browser, make sure you are connected with the same WiFi network as your ESP board.

You'll be greeted with a page like this. Both the default username and password are "admin", You may change them according to your wish in the code itself. Just change this line of the code.

```
"if(form.userid.value=='admin' &&
form.pwd.value=='admin')"
```

Step 4:- Done! Upload the Code Over-the-Air through Web

Now that you have generated the binary file of your code, it's time to upload the code to the board through the web page. Open the /serverIndex page in your browser. Click on Choose File... Select the generated .bin file, and then click Update.

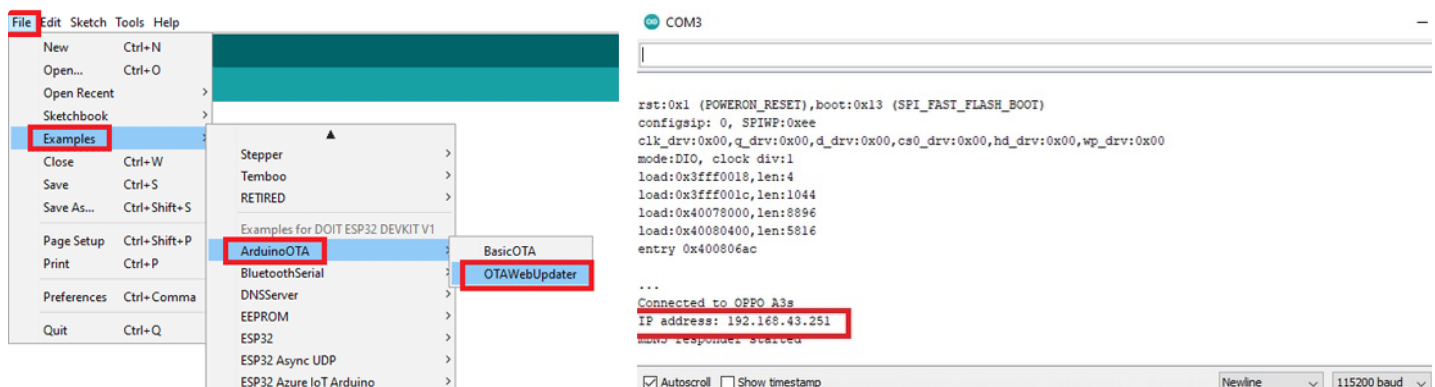
Wait for a few seconds..... And Wallah! The code will be uploaded to the board over the air without the Arduino IDE.

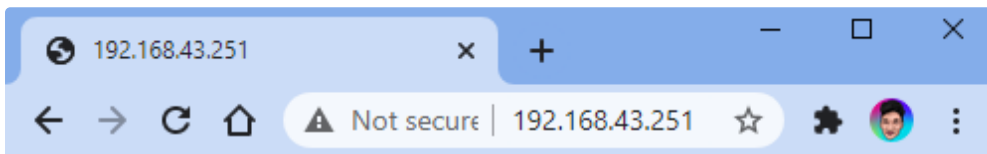
How cool is that? Share your thoughts in the comment section below.

Step 3:- Get the Binary File of Your Code

After you log in, you'll see the page from which you can upload sketch but for that, we first need to generate the bin file of our sketch. The code is for blinking an LED while also keeping the OTA feature intact.

Now we need to generate the bin file of this code. For this, just go into the sketch tab in Arduino IDE and choose the export compiled binary.





ESP32 Login Page

Username:

Password:

Login

