

OTA (over the air) firmware upload for ESP8266

<http://docs.platformio.org/en/stable/projectconf.html>

https://esp8266.github.io/Arduino/versions/2.0.0/doc/ota_updates/ota_updates.html

<https://www.youtube.com/watch?v=GoQXOLB50HA>

<https://ptarmiganlabs.com/blog/2016/02/23/esp8266-over-the-air-updating-what-are-the-options/>

<https://harizanov.com/2015/06/firmware-over-the-air-fota-for-esp8266-soc/>

Ejemplo de arduino

El ejemplo de github está desactualizado y no funciona, tomado desde el IDE de arduino si funciona.

<https://github.com/esp8266/Arduino/blob/master/libraries/ArduinoOTA/examples/BasicOTA/BasicOTA.ino>

Se fusiona el código de OTA con la librería [WifiManager](#) para administrar la configuración a la red.



Con este programa y la modificación que se explica abajo en el archivo **platformio.ini** la actualizaciones siguientes del firmware del esp se harán a través de wifi

```
/*
 * El código de los ejemplos de github es diferente al de el IDE de arduino.
 * El de github no funcionó
 */

#include <ESP8266WiFi.h>
#include <ESP8266mDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include <WiFiManager.h>           // https://github.com/tzapu/WiFiManager

void setup() {
  Serial.begin(115200);
  Serial.println("Booting");

  //WiFiManager
  //Local intialization. Once its business is done, there is no need to keep
  it around
  WiFiManager wifiManager;
  //reset saved settings
  //wifiManager.resetSettings();
```

```
//set custom ip for portal
//wifiManager.setAPConfig(IPAddress(10,0,1,1), IPAddress(10,0,1,1),
IPAddress(255,255,255,0));

//fetches ssid and pass from eeprom and tries to connect
//if it does not connect it starts an access point with the specified name
//here "AutoConnectAP"
//and goes into a blocking loop awaiting configuration
wifiManager.autoConnect("AutoConnectAP");
//or use this for auto generated name ESP + ChipID
//wifiManager.autoConnect();

//if you get here you have connected to the WiFi
Serial.println("connected...yeey :)");

// Port defaults to 8266
// ArduinoOTA.setPort(8266);

// Hostname defaults to esp8266-[ChipID]
// ArduinoOTA.setHostname("myesp8266");

// No authentication by default
// ArduinoOTA.setPassword((const char *)"123");

ArduinoOTA.onStart([]() {
  Serial.println("Start");
});
ArduinoOTA.onEnd([]() {
  Serial.println("\nEnd");
});
ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
  Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
});
ArduinoOTA.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 after OTA desconectado del pc");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
}
```

```
void loop() {  
  ArduinoOTA.handle();  
}
```

También hay que editar el archivo **platformio.ini** donde se agrega la línea `upload_board`

```
; PlatformIO Project Configuration File  
;  
; Build options: build flags, source filter, extra scripting  
; Upload options: custom port, speed and extra flags  
; Library options: dependencies, extra library storages  
;  
; Please visit documentation for the other options and examples  
; http://docs.platformio.org/en/stable/projectconf.html  
  
[env:esp12e]  
platform = espressif8266  
board = esp12e  
framework = arduino  
lib_install = 567  
lib_deps = WiFiManager  
upload_port = 192.168.1.71 ; la ip que me entregó el router de la casa
```

Funciona con platformio no se con arduino cómo sería pero se que también se puede `upload_port = 192.168.1.71` ## La ip que le entregó el router de mi casa

From:
<https://wiki.unloquer.org/> -

Permanent link:
https://wiki.unloquer.org/proyectos/jardin_delicias/tecnologicos/esp8266_ota?rev=1475786901

Last update: **2016/10/06 20:48**

