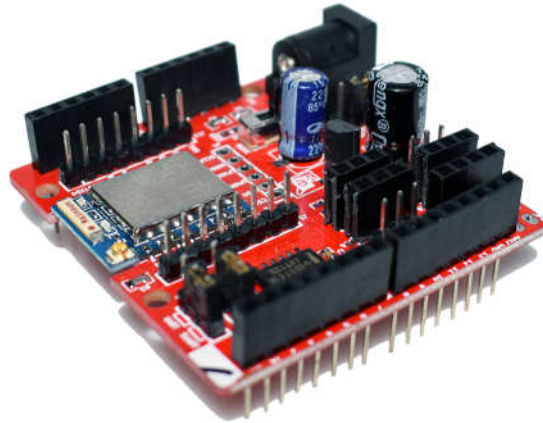


# ESP8266 (ESP-07/ESP-12E) WiFi Shield



Technical Manual Rev 1r0

---



ESP-07 and ESP-12 WiFi shield is a Wi-Fi Network solution, it can be used to host the application or connected from another application processor. It served as a Wi-Fi Adapter, wireless internet access can be added to any microcontroller based designed with simple connectivity (SPI/SDIO or I2C/UART interface). Applications are Home Automation, Monitoring Room temperature, Sending/Receiving Data though IOT etc.

## Features:

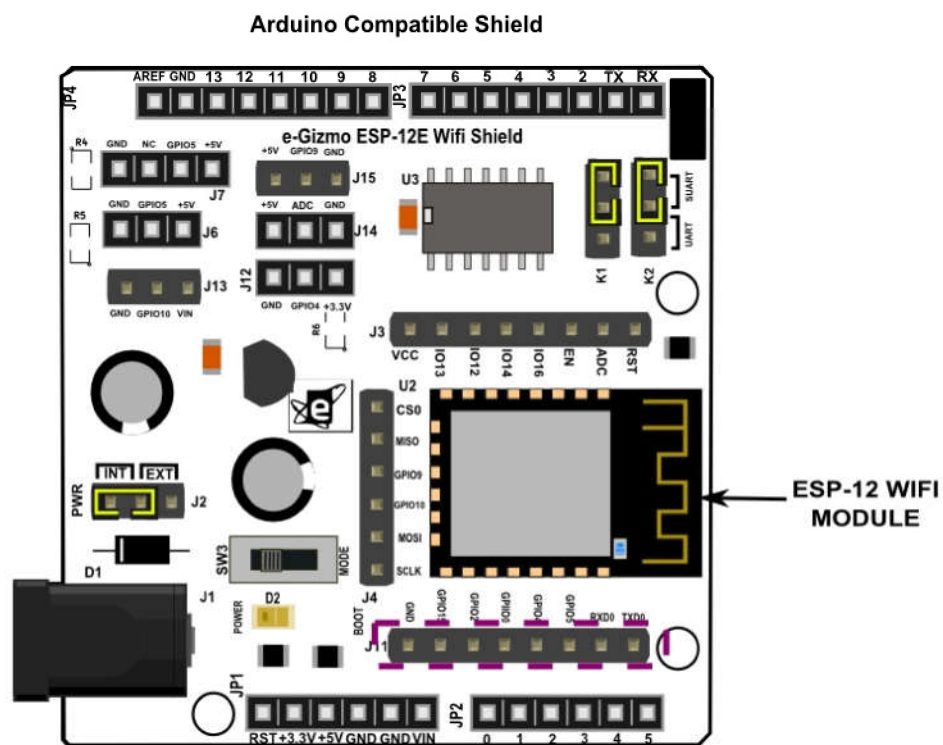
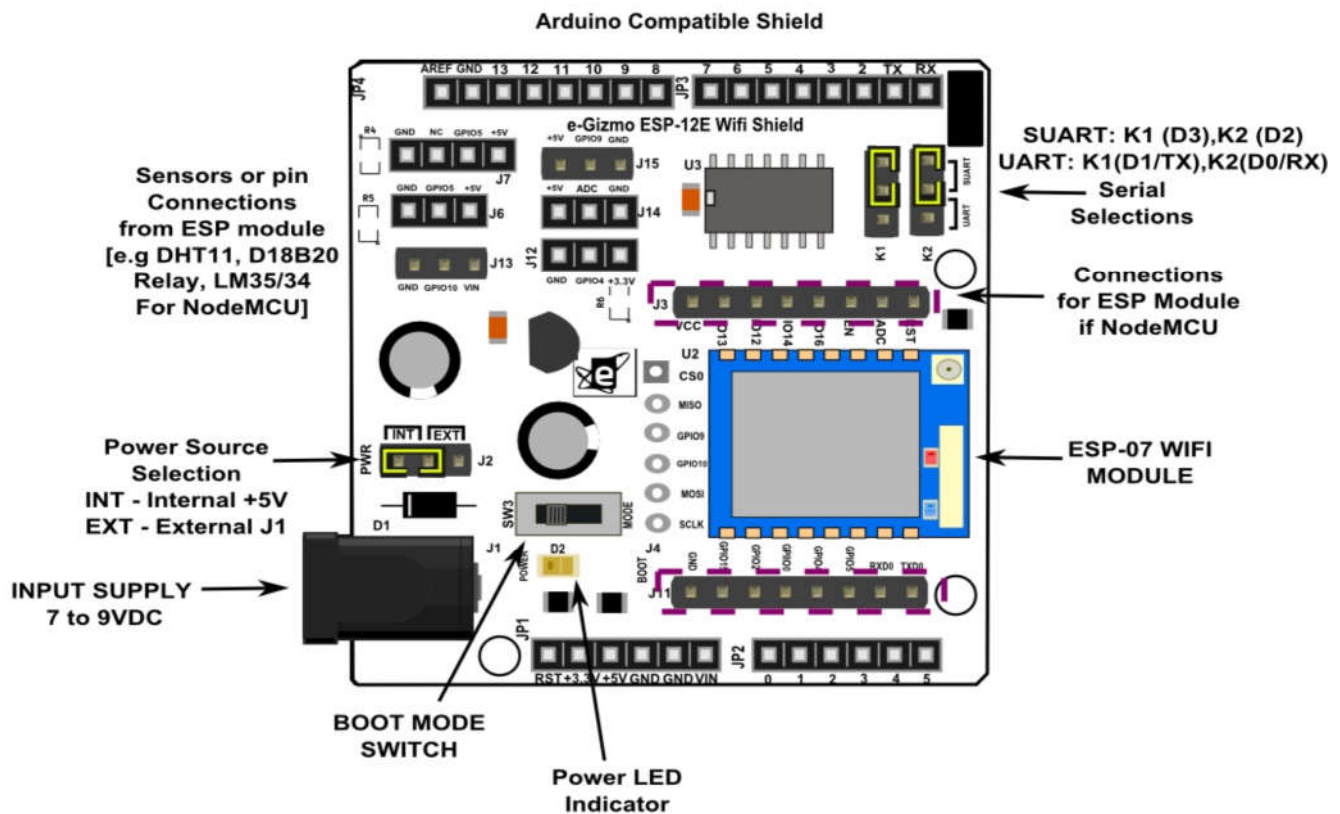
- 802.11 b/g/n
- Low power 32-bit MCU
- 10-bit ADC
- TCP/IP protocol stack
- Supports antenna diversity
- Wi-Fi 2.4GHz, support WPA/WPA2
- Supports STA/AP/STA+AP operation modes.
- Support Smart Link Function for both Android and iOS devices.
- SDIO 2.0, (H) SPI, UART, I2C, PWM, GPIO
- Arduino Compatible

## General Specifications:

**Input Voltage:** +5VDC [Internal Source]  
+7V to 9VDC [External Source]  
**ESP8266 Module Supply:** +3.3VDC  
**Type of ESP:** ESP-07 or ESP-12E (Optional)  
**Model Vendors:** AI-THINKER, DOIT, Espressif  
**Regulator Used:** AMS2950  
**On-board IC:** 74LVT125  
**Operating Temperature:** -40C~125C  
**Output Power:** +20dBm in 802.11b mode  
**PCB Dimensions:** 52.5mm x 53.5mm

For more info read the ESP-07/12 Datasheet

Note: The ESP-07 or 12 Wifi Shield has been bootloaded of firmware were you can see on this manual. It is **\*\*\*Compatible in Gizduino PLUS** (with 64KB Flash memory) recommended to use. Arduino IDE 1.8.3 with Gizduino Patch.  
Use this At Your Own Risk!.



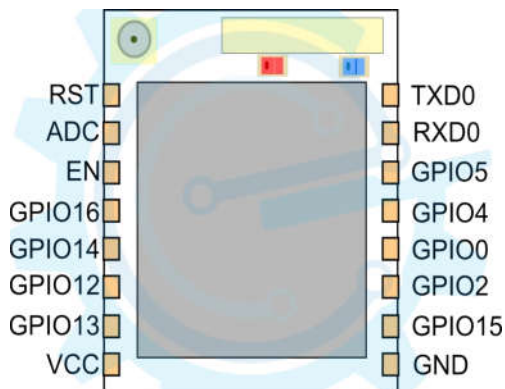


Figure 3: ESP-07 Module Pinouts

Table 1.

FOR ESP-07 ADDITIONAL PINOUTS

Pin Functions	Descriptions
RST	Reset the module [Note: If ESP-07 Module resets the firmware will erase]
ADC	A0/Analog to Digital Converter or ADC
EN	Enable pin
GPIO16	General Purpose I/O pin 16
GPIO14	General Purpose I/O pin 14
GPIO12	General Purpose I/O pin 12
GPIO13	General Purpose I/O pin 13
VCC	Input Supply 3.3VDC Required, use AMS2950 Regulator
TXD0	Transmitter, it needs voltage level translator 5VTTL to 3.3V
RXD0	Receiver
GPIO5	General Purpose I/O pin 5
GPIO4	General Purpose I/O pin 4
GPIO0	Bootload mode, Connect it to the Ground to Bootload
GPIO2	General Purpose I/O pin 2, BUILTIN_LED
GPIO15	General Purpose I/O pin 15, Connect it to the Ground to Bootload
GND	Ground 0V

FOR ESP-12 ADDITIONAL PINOUTS

SCLK	Clock
MOSI	Main Output Slave input
GPIO10	General Purpose I/O pin 10
GPIO9	General Purpose I/O pin 9
MISO	Slave output Main input
CS0	Chip selection

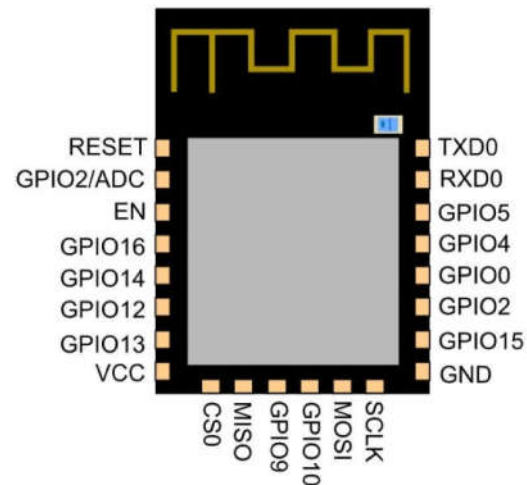


Figure 4: ESP-12 Module Pinouts

For more information:

See the ESP12E Specs/ Datasheet

You may skip this tutorial! (Default firmware when purchased

- go to Connect the ESP wifi shield to gizDuino)

Download the Following Files and Extract them:

- 1.) esp\_iot\_sdk\_v1.5.0\_15\_11\_27.zip from Espressif site  
[<http://bbs.espressif.com/download/file.php?id=989>]
- 2.) FLASH\_DOWNLOAD\_TOOLS\_v2.4\_150924.rar  
from Espressif site  
[<http://bbs.espressif.com/download/file.php?id=856>]

Materials:

1pc -USB to UART Converter Type A to B 1.8m



4pcs - 1pin Wire connector (Female-Male)



1pc -USB Cable or Printer Cable



**IMPORTANT!!!**

**SWITCH THE ESP MODULE IN FLASHING MODE!!!**



**FLASHING MODE**

Connect GPIO 0 pin to Ground.  
Switch SW3 for flashing mode.

For Correct wiring and flashing mode.

When you connect the USB-UART converter to PC.  
"Look if the **BLUE LED** indicator of ESP-07 Module  
will blink **once**"

it indicates that your connection is correct and its  
ready for downloading the firmware  
from esp flash tool.

Construct this Wiring Diagram:

USB - UART connections to ESP-07 Wifi Shield

**+5V RED** wire to +5V

**GND BLACK** wire to Ground

**RXD ORANGE** wire to Digital pin 2

**TXD BLUE** wire to Digital pin 3

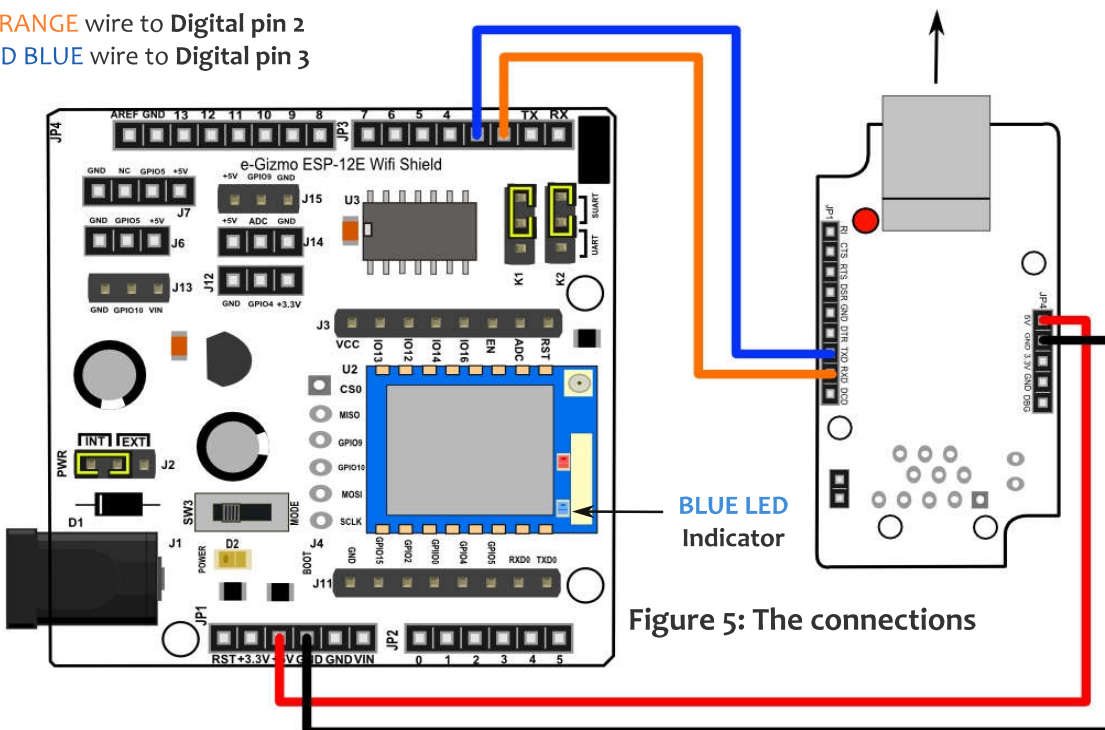
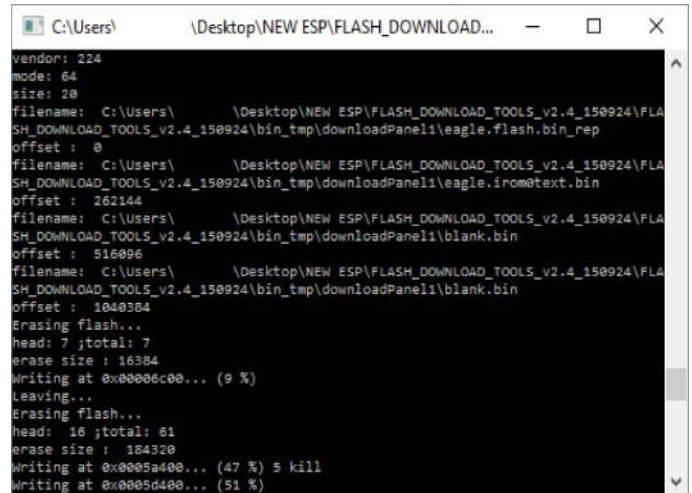


Figure 5: The connections

## Updating ESP-07's Firmware using ESP Download Tool v 2.4

Follow these steps to flash the new firmware to the ESP-07 Module.

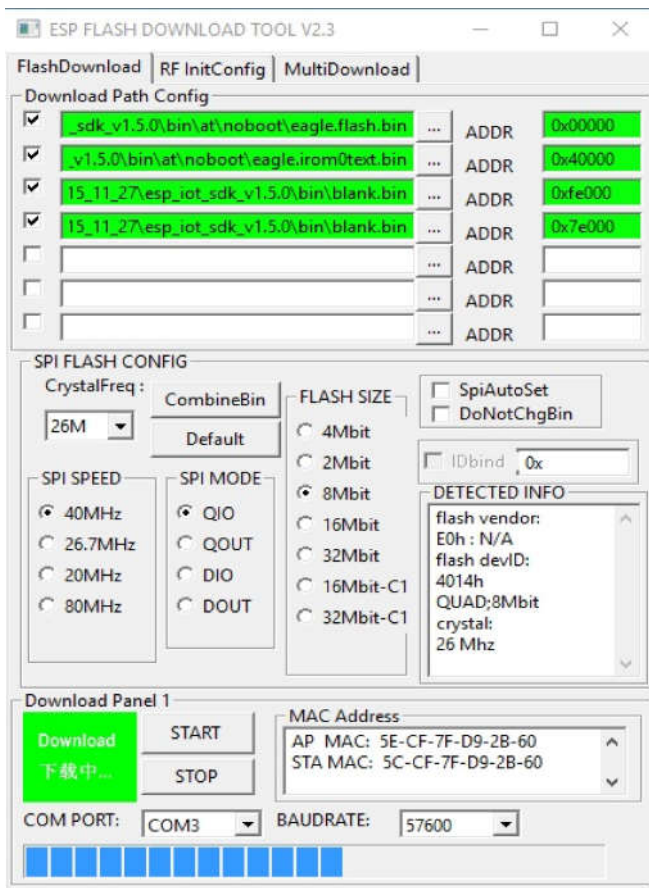
- 1.) Close the Arduino IDE if still Opened.
- 2.) Launch the ESP flash download tool  
"ESP\_DOWNLOAD\_TOOL\_2.4.exe" you have previously extracted.
- 3.) Apply the following settings
- 4.) Bin files from the "esp\_iot\_sdk extracted zip file):
  - A.) bin\at\noboot\**eagle.flash.bin** - 0x00000
  - B.) bin\at\noboot\**eagle.irom0text.bin** - 0x40000
  - C.) bin\blank.bin - 0xfe000
  - D.) bin\blank.bin - 0x7e000
- 5.) Flash size: 26MHz
- 6.) COM port: Choose your COM port number  
(note: change the COM from 2 to 4, nearest the better communication).
- 7.) Baudrate: 115200 or 345600 (this is not related to the ESP baud rate) or in my example i used 57600.



```
vendor: 224
mode: 64
size: 20
filename: C:\Users\...\Desktop\NEW ESP\FLASH_DOWNLOAD_TOOLS_v2.4_150924\FLASH_DOWNLOAD_TOOLS_v2.4_150924\bin_tmp\downloadPanel1\eagle.flash.bin_rep
offset: 0
filename: C:\Users\...\Desktop\NEW ESP\FLASH_DOWNLOAD_TOOLS_v2.4_150924\FLASH_DOWNLOAD_TOOLS_v2.4_150924\bin_tmp\downloadPanel1\eagle.irom0text.bin
offset: 262144
filename: C:\Users\...\Desktop\NEW ESP\FLASH_DOWNLOAD_TOOLS_v2.4_150924\FLASH_DOWNLOAD_TOOLS_v2.4_150924\bin_tmp\downloadPanel1\blank.bin
offset: 516096
filename: C:\Users\...\Desktop\NEW ESP\FLASH_DOWNLOAD_TOOLS_v2.4_150924\FLASH_DOWNLOAD_TOOLS_v2.4_150924\bin_tmp\downloadPanel1\blank.bin
offset: 1040384
Erasing flash...
head: 7 ;total: 7
erase size: 16384
Writing at 0x00000000... (9 %)
Leaving...
Erasing flash...
head: 16 ;total: 61
erase size: 184320
Writing at 0x0005a400... (47 %) 5 kill
Writing at 0x0005d400... (51 %)
```



In Download Panel 1, if "FINISH".  
Press the STOP button and close the Flashing tool.



Now press the "START" button and wait for the flashing process to complete.



## Test connectivity and set the default UART speed

### Verifying the firmware using Serial Monitor

Before proceeding it is better to verify that new firmware is working fine.

- 1.) Unplug the USB -UART converter from the PC.
- 2.) Switch SW3 to unflashing mode or remove GPIO 0 from the GND.
- 3.) Then Plug the UART converter to the PC..

For Correct wiring: "Look again if the **BLUE LED** indicator of ESP-07 module will blink **twice**"  
*it indicates that the connection is successful.*

Open the Arduino IDE, select the correct COM port and open the Serial Monitor [Press Ctrl+ Shift +M].

(The correct settings for the ESP firmware v1.5)

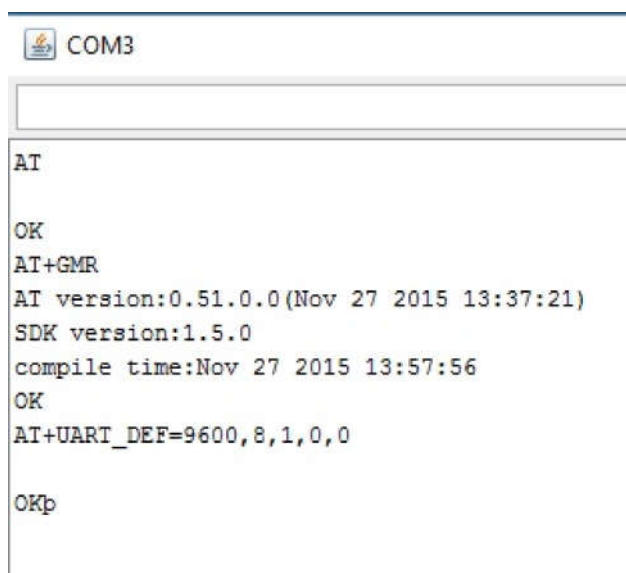
**SET Baudrate Speed: 115200**

**Line ending: Both NL & CR.**

Test connectivity with 'AT' and 'AT+GMR' commands.  
To set the correct baud rate use this command:

```
AT+UART_DEF=9600,8,1,0,0
```

Now set the Serial Monitor speed to 9600 and test again the communication.



```
COM3
AT
OK
AT+GMR
AT version:0.51.0.0(Nov 27 2015 13:37:21)
SDK version:1.5.0
compile time:Nov 27 2015 13:57:56
OK
AT+UART_DEF=9600,8,1,0,0
OKp
```

**Caution: Do not send AT+RESET or AT+RST. The firmware will be set to default bin.**

### Download the "WiFiEsp" library

Direct link here:

[<https://github.com/bportaluri/WiFiEsp/archive/master.zip>]

Compatible using Arduino IDE 1.8.x and up. I'm using the 1.8.3 for this example.

To add the library:

**Go to My Document>Arduino>libraries> (Paste it)**

[Make sure the Folder named "WiFiEsp" contains with the files inside, it must be extracted not compressed when adding a library and Restart IDE]

### Connect the ESP-07 WiFi Shield to gizDuino.

**\*\*\*GizDuino PLUS with ATMEA644P  
Recommended to use.**

### Download the Arduino IDE 1.8.3 with gizDuino Patch (Board lists).

Download link for Arduino IDE 1.8.3.

[<https://github.com/e-Gizmo/Arduino-1.8.3-IDE-Windows/archive/master.zip>]

**After downloading the IDE.**

if you are using the gizduino boards

**Install first the Prolific driver.**

[[https://github.com/e-Gizmo/Arduino-1.0.6-IDE-Windows/blob/master/drivers/Prolific%20USB%20Drivers/P\\_L2303\\_Prolific\\_DriverInstaller\\_v1.10.0.exe?raw=true](https://github.com/e-Gizmo/Arduino-1.0.6-IDE-Windows/blob/master/drivers/Prolific%20USB%20Drivers/P_L2303_Prolific_DriverInstaller_v1.10.0.exe?raw=true)]

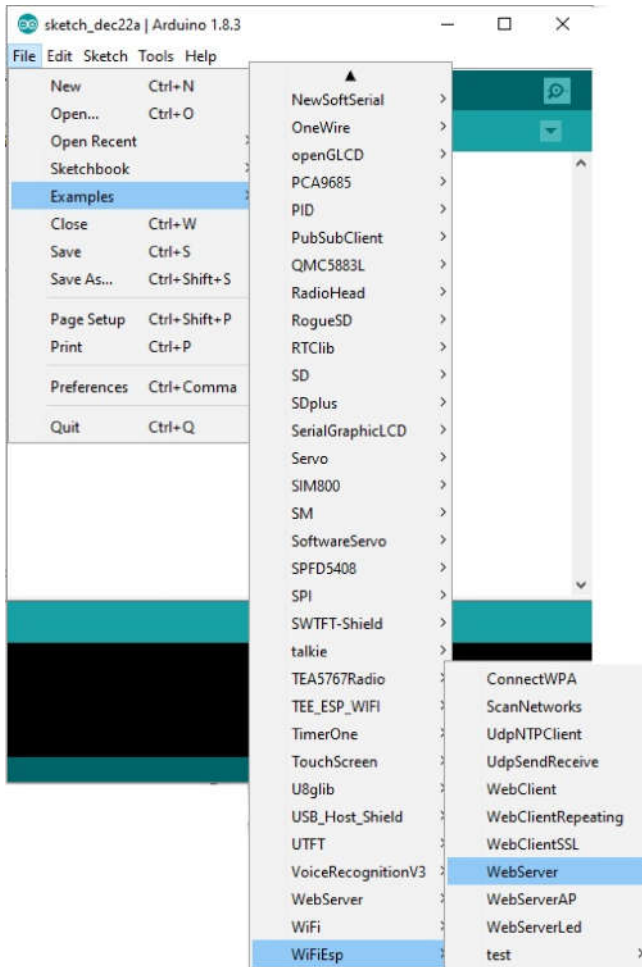
## Uploading the Sample program from WiFiEsp library

### OPEN the Arduino IDE 1.8.3

Before opening the Arduino IDE, make sure you add the WiFiEsp library.

Now Open the WiFiEsp > WebServer.

Go to File>Example> WiFiEsp> WebServer.ino



You need a WiFi Network connections  
(WLAN router, Hotspots for Wifi connectivity)

```
char ssid[] = "Twim";           // your network SSID (name)
char pass[] = "12345678";       // your network password
int status = WL_IDLE_STATUS;    // the Wifi radio's status
int reqCount = 0;               // number of requests received

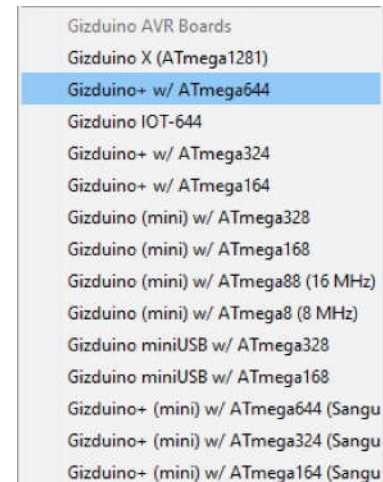
WiFiEspServer server(80);
```

Set your network here:  
char ssid[] = "YOUR\_SSID"  
char pass[] = "SSID\_PASSWORD"

Check also the port forwarding: if it is 80,8080, if necessary.  
WiFiEspServer server(80);

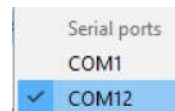
Select the board:

Tools> Board:> Gizduino+ w/ ATmega644



Select the COM Port Number:

Tools>Port> COM12



Change the Serial connection to  
Digital Pin 2 and Digital Pin 3

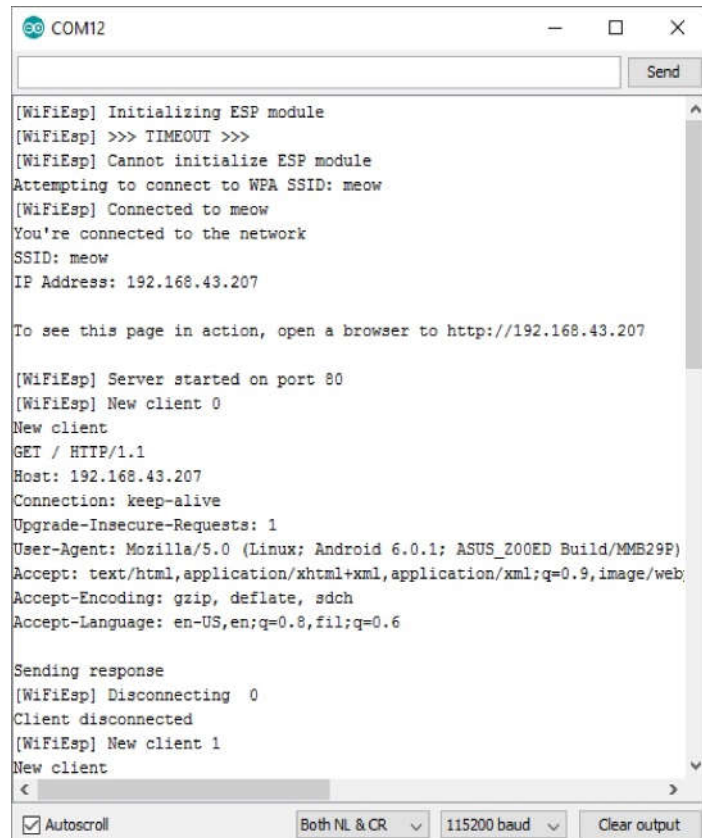
```
#include "WiFiEsp.h"

// Emulate Serial1 on pins 6/7 if not present
#ifdef HAVE_HWSERIAL1
#include "SoftwareSerial.h"
SoftwareSerial Serial1(2, 3); // RX, TX
#endif
```

Now Click Upload.

### Sample Attempting to connect to WPA SSID:

Note: This is an example Output  
from the Serial Monitor:



```
[WiFiEsp] Initializing ESP module
[WiFiEsp] >>> TIMEOUT >>>
[WiFiEsp] Cannot initialize ESP module
Attempting to connect to WPA SSID: meow
[WiFiEsp] Connected to meow
You're connected to the network
SSID: meow
IP Address: 192.168.43.207

To see this page in action, open a browser to http://192.168.43.207

[WiFiEsp] Server started on port 80
[WiFiEsp] New client 0
New client
GET / HTTP/1.1
Host: 192.168.43.207
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 6.0.1; ASUS_Z00ED Build/MMB29P)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,fil;q=0.6

Sending response
[WiFiEsp] Disconnecting 0
Client disconnected
[WiFiEsp] New client 1
New client
```

Get the IP address and  
Open browser type:  
192.168.0.112 or the given IP address

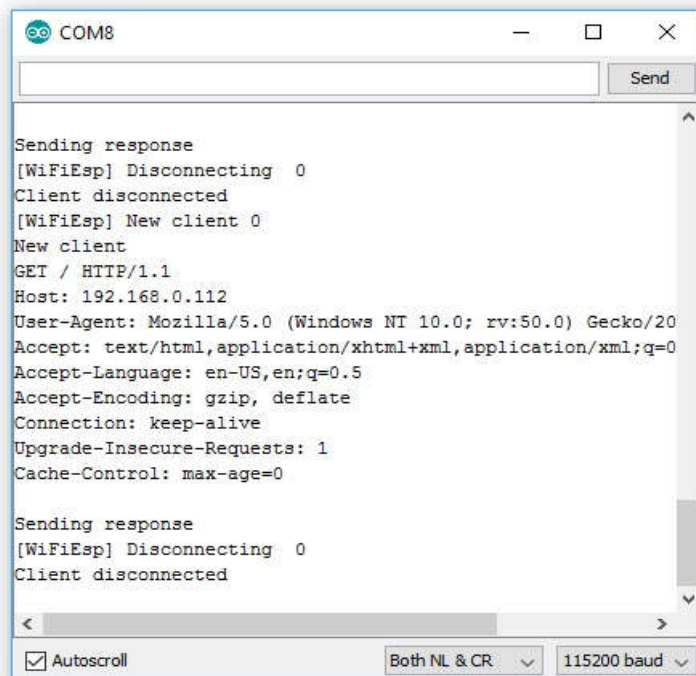
It will display the message:  
**Hello World!**  
and Requests received and Analog output

Page refresh every 20 secs.



## Hello World!

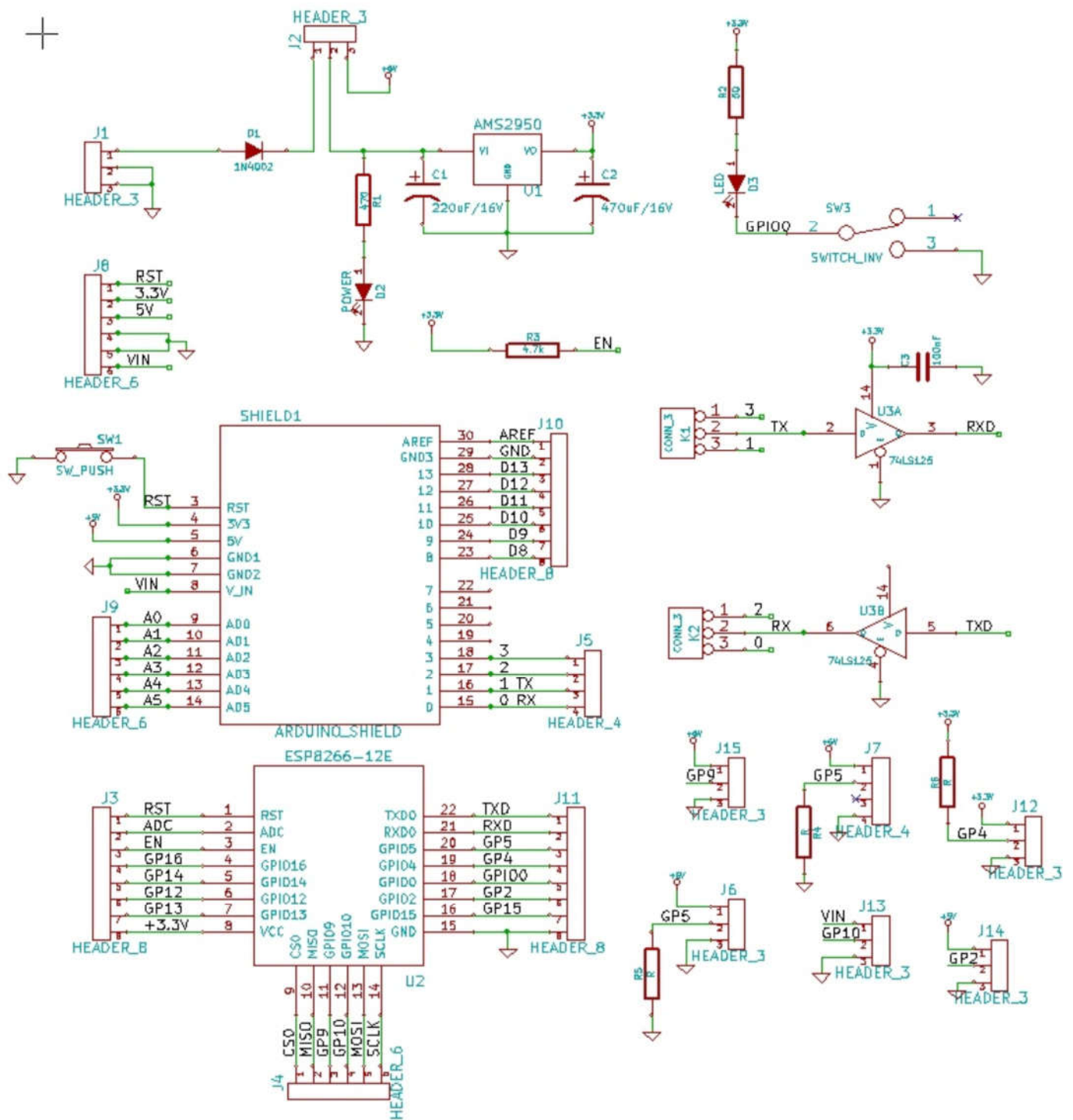
Requests received: 6  
Analog input A0: 336



```
Sending response
[WiFiEsp] Disconnecting 0
Client disconnected
[WiFiEsp] New client 0
New client
GET / HTTP/1.1
Host: 192.168.0.112
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:50.0) Gecko/20
Accept: text/html,application/xhtml+xml,application/xml;q=0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

Sending response
[WiFiEsp] Disconnecting 0
Client disconnected
```





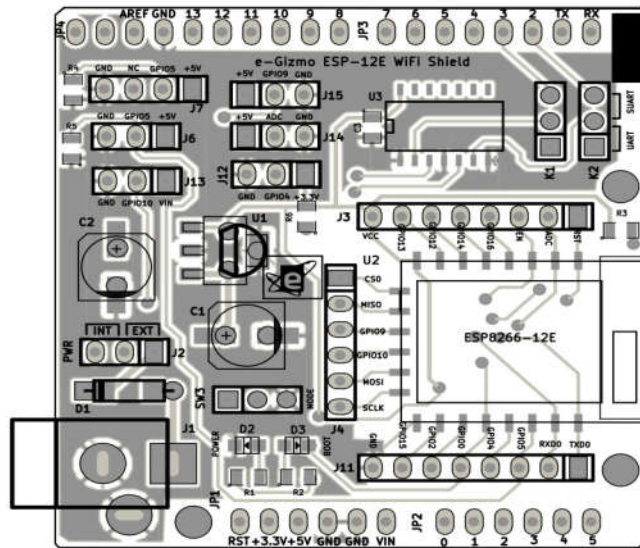


Figure : Parts Placement

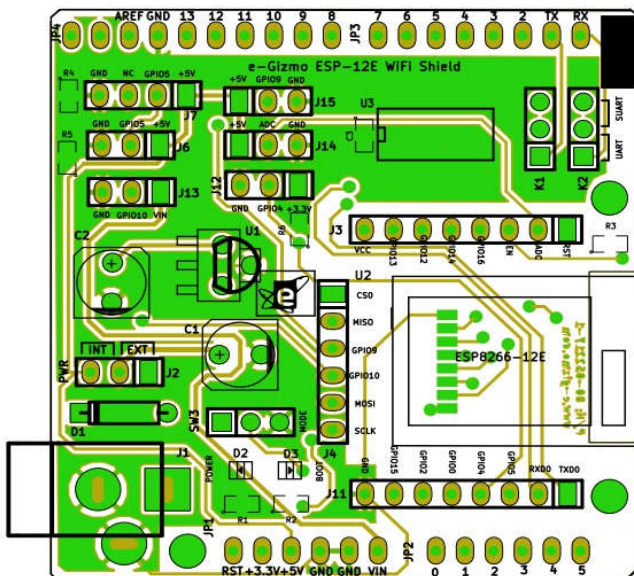


Figure : Bottom Layer Guide

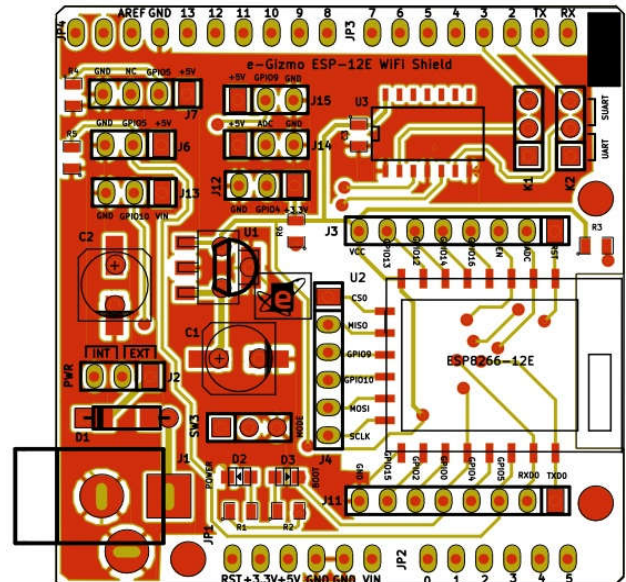


Figure : Top Layer Guide