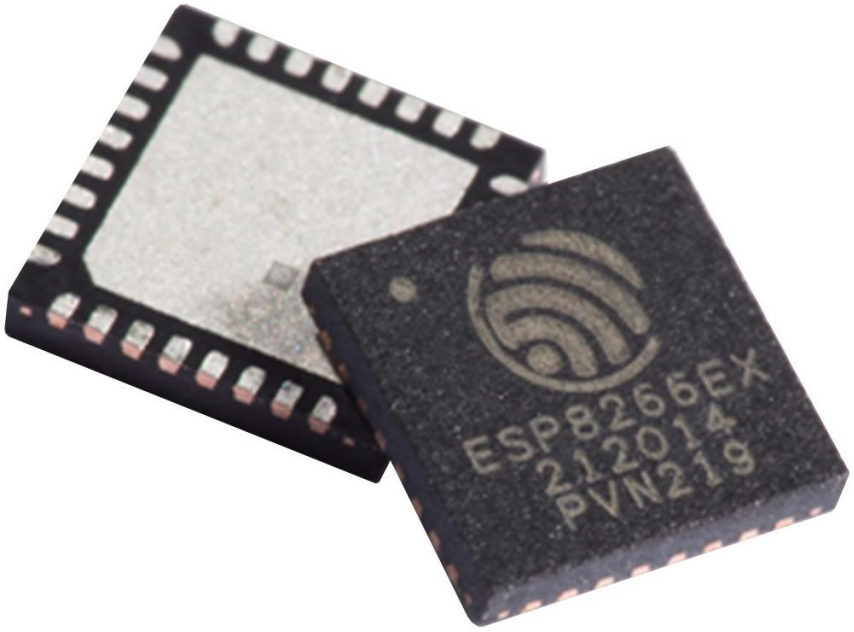
ESP8266



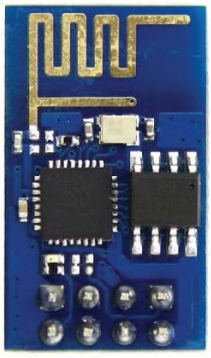
ESP8266 – The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack, radio and microcontroller produced by Shanghai-based Chinese manufacturer, Espressif.

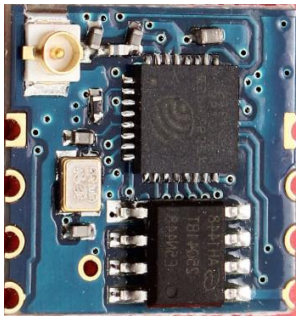
Features:

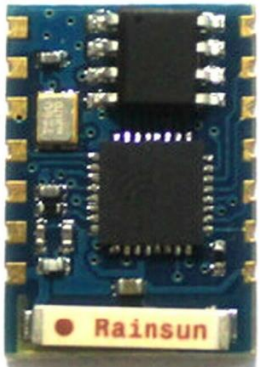
- SOC (System on a Chip)
- 32-bit RISC CPU, 80 MHz
- 64K Instruction RAM
- 96K Data RAM
- External Flash up to 16MiB
- WiFi 802.11 b/g/n
    1. WEP
    2. WPA/WPA2
- 16 GPIO Pins
    1. SPI
    2. I2C
    3. UART
    4. 16 I/O
    5. 1 10-bit ADC

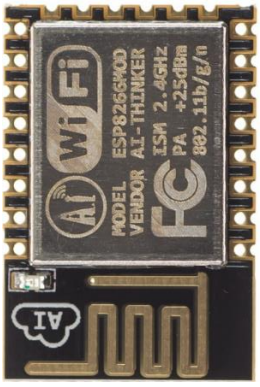
Boards / Modules

- ESP-01
- ESP-02
- ESP-03
- ESP-12
- ESP-12E
- ESP-12 Development Board
- D1 – ESPduino









ESP-01

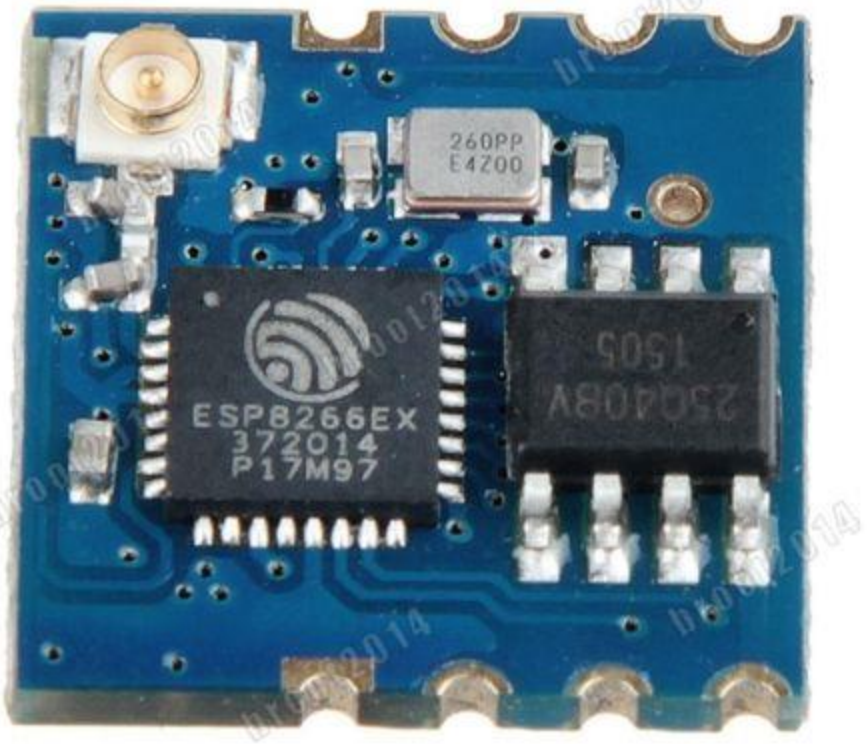
- 8 Pin Module
- On Board Antenna
- 2 (3) usable I/O

1. GND
2. TXD
3. GPIO2
4. Enable
5. Reset
6. GPIO0
7. VCC
8. RXD

ESP-02

ii. 8 Pin Module
iii. External Antenna
iv. 2 (3) usable I/O

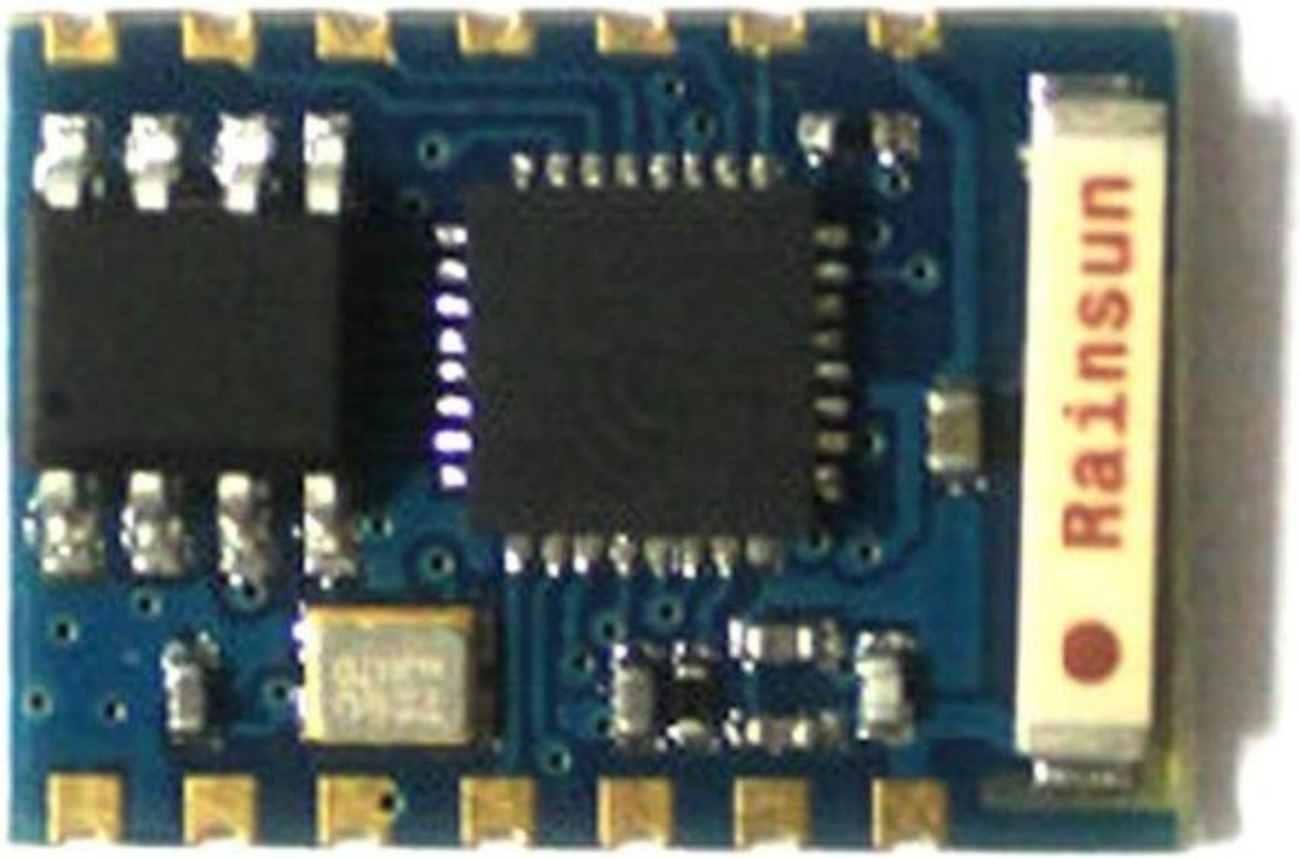
1. GND
2. TXD
3. RXD
4. VCC
5. Reset
6. GPIO2
7. GPIO0
8. Enable



ESP-03

- 14 Pin Module
- On Board Chip Antenna
- 7 usable I/O

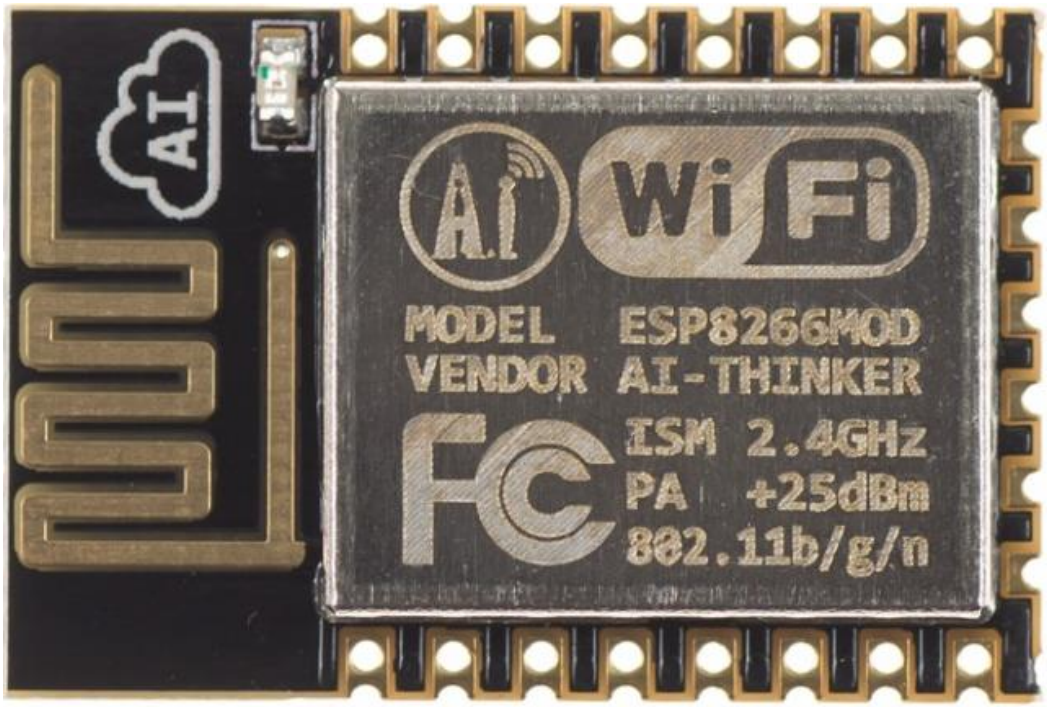
1. VCC
2. GPIO14
3. GPIO12
4. GPIO13
5. GPIO15
6. GPIO2
7. GPIO0
8. GND
9. N/C
10. TXD
11. RXD
12. GPIO18
13. Enable
14. Antenna



ESP-12(E)

- 22 Pin Module
- On Board Antenna
- 10 usable I/O
- FCC Approved

1. Reset
2. ACD
3. Enable
4. GPIO16
5. GPIO14
6. GPIO12
7. GPIO13
8. VCC
15. GND
16. GPIO18
17. GPIO2
18. GPIO0
19. GPIO4
20. GPIO5
21. RXD
22. TXD



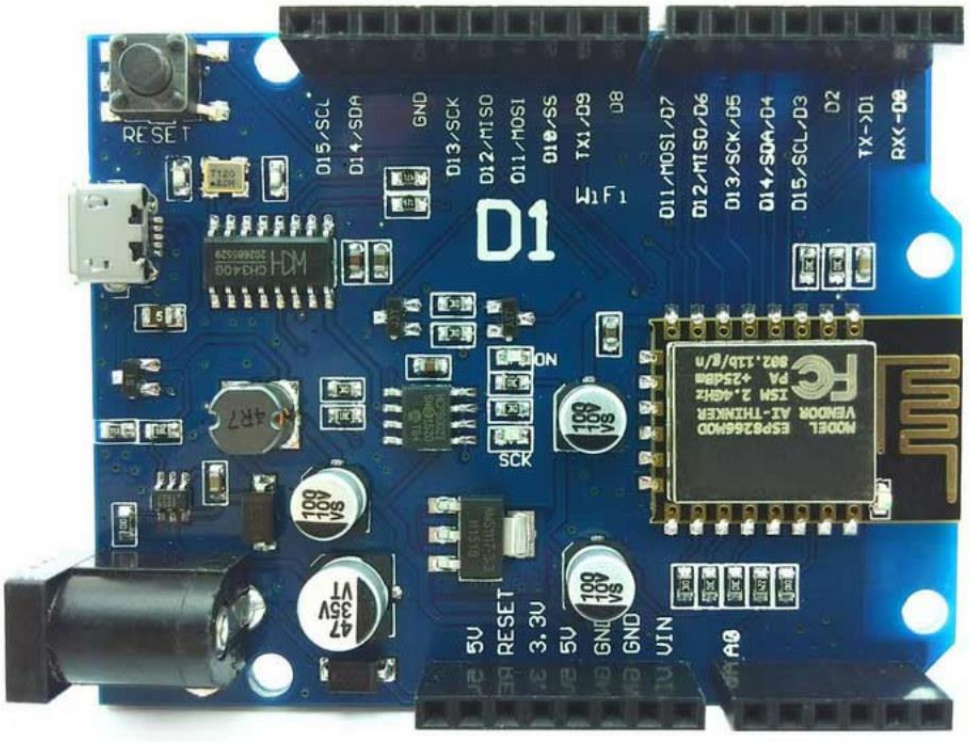
ESP-12(E) Development Board

- 30 Pin Module
- On Board Antenna
- 14 usable I/O
- USB (Serial)
- Reset Button
- Flash Button

1. ACD
2. Reserved
3. Reserved
4. GPIO10
5. GPIO9
6. MOSI
7. CS
8. MISO
9. SCLK
10. GND
11. 3.3V
12. EN
13. Reset
14. GND
15. VIN
16. GPIO16
17. GPIO5
18. GPIO4
19. GPIO0
20. GPIO2
21. 3.3V
22. GND
23. GPIO14
24. GPIO12
25. GPIO13
26. GPIO15
27. GPIO3
28. GPIO1
29. GND
30. 3.3V

D1 ESPduino

- UNO Footprint
- On Board Antenna
- 10 usable I/O
    1. RXD
    2. TXD
    3. D2
    4. D3
    5. D4
    6. D5
    7. D6
    8. D7
    9. D8
    10. D9
    11. D10
    12. D11
    13. D12
    14. D13
    15. GND
    16. N/C
- USB (Serial)
- Reset Button
- UNO Pinout
    17. D14
    18. D15
    19. N/C
    20. 5V
    21. Reset
    22. 3.3V
    23. 5V
    24. GND
    25. GND
    26. VIN
    27. A0
    28. N/C
    29. N/C
    30. N/C
    31. N/C
    32. N/C



Software

1) SDK

The SDK was originally provided by the Shanghai-based Chinese chip manufacturer, Espressif and is fully open source. There are many versions out there and the easiest one to use is the VMWare version that comes completely configured and ready to go.

2) NodeMCU

NodeMCU is an Open Sourced lua based firmware for the ESP8266. It is well supported and reasonably stable with ongoing development. It was originally targeted for IOT but is well suited for Robotics.

a. Firmware

The NodeMCU firmware is available as standard build with integer or floating point math. You can also compile your own custom version with the SDK or use the website http://nodemcu-build.com/ to build your own custom version from the latest source and any options you want.

NodeMCU – lua, is an interpretive language. You can enter commands directly via a serial port or edit and upload scripts as a file and execute them. Many scripts can operate seemingly at the same time. Typically timers, interrupts and the watchdog control the flow of your programs.

Scripts look a lot like C but:

Variables do not have to be defined
No ";" required at the end of every line
No {} for functions, if-then-else

```
lighton=0
tmr.alarm(0,1000,1,function()
  if lighton==0 then
    lighton=1
    led(512,512,512)
  else
    lighton=0
    led(0,0,0)
  end
end)
```

b. Tools

There are two categories in tools, Loaders and IDEs. Loaders allow you to just load a binary image onto your ESP8266 while the IDEs typically have an editor and allow you to manage your programming.

- Flasher/Uploader

  If you want to use NodeMCU-lua with your ESP8266 then you will need to select a version and upload it to the flash memory on your device. There are several out there but the one I like and use is:

  ESP8266 Flasher

- IDE

  In order to write programs and run them under NodeMCU-lua on your ESP8266 you will need to edit and upload the files to your device. For this you will use an IDE. There are several available:

  - ESPlorer
  - LDT (lua dev tools, Eclipse)
  - ESP Manager
  - NodeMCU Studio

I personally like and use ESPlorer. Some of the others are more complicated and the learning curve seemed too steep. ESPlorer allows you to manage your files on your computer and within NodeMCU on your device.

3) ESPduino

This board is the same footprint as an Arduino UNO. You can use it with most Arduino Shields and it runs many of the UNO programs. The Arduino IDE (1.6.3 and above) support the ESPduino with the supported boards files.

a. Board File

Launch the Arduino IDE and go to File→Preferences and enter http://arduino.esp8266.com/stable/package_esp8266com_index.json additional boards Manager URLs. Then click on the Tools pulldown, then select Board. At the top of the list there is the Board Manager. Select Type of all and then enter ESP and the esp8266 Boards package will appear. Install it and restart the IDE. Now go to the Tools pulldown and select Board, there will be several ESP8266 boards in the list. Select the proper one and that's it.

b. SPIFFS Filesystem

The ESPduino firmware support a flash SPIFFS filesystem. You can allocate a portion of the Flash memory for storage of data and other files.
There is a module that you can add to the Arduino tools directory that allows you to manipulate the files stored on the SPIFFS file system. This tool will upload files stored in the current Sketch's data directory to your D1 (or any attached ESP8266 board).

c. Examples

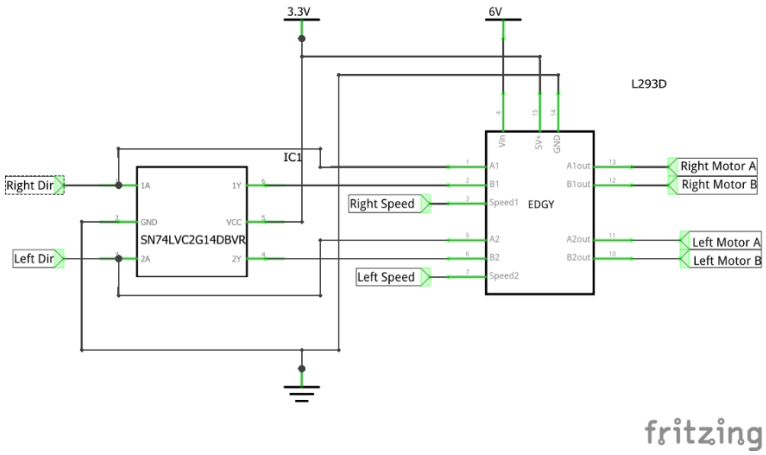
With the Boards Package are several examples for the ESP8266. To start there is the Blink, Blink without Delay and other familiar sketches. Also many for different variants of the board and many examples of WiFi, Access Point and Client mode along with a Mesh Network.

ESP Motor Controller Shield

The ESP8266 Motor Controller Shield has a L293 Dual H Bridge, breakout pins for all I/O, Screw terminals for power and motor connections, A Power Switch and Power LED. It is designed for the NodeMCU v2 board, which is a slightly narrower and a little more costly than the newer NodeMCU v3.

There is an Enable and Direction pin for each motor, unlike other motor controllers that require 3 pins for each H-Bridge. This board has the added logic and reduced the pin count. There are also resistors to help prevent the motors from running if the I/O pins are not initially configured on power up.

The board has separate motor and processor power inputs, but also has a jumper to allow a single power supply to be used. It is highly recommended that you use two power supplies to prevent the current surges of the motors from interfering with the processor and WiFi radio.

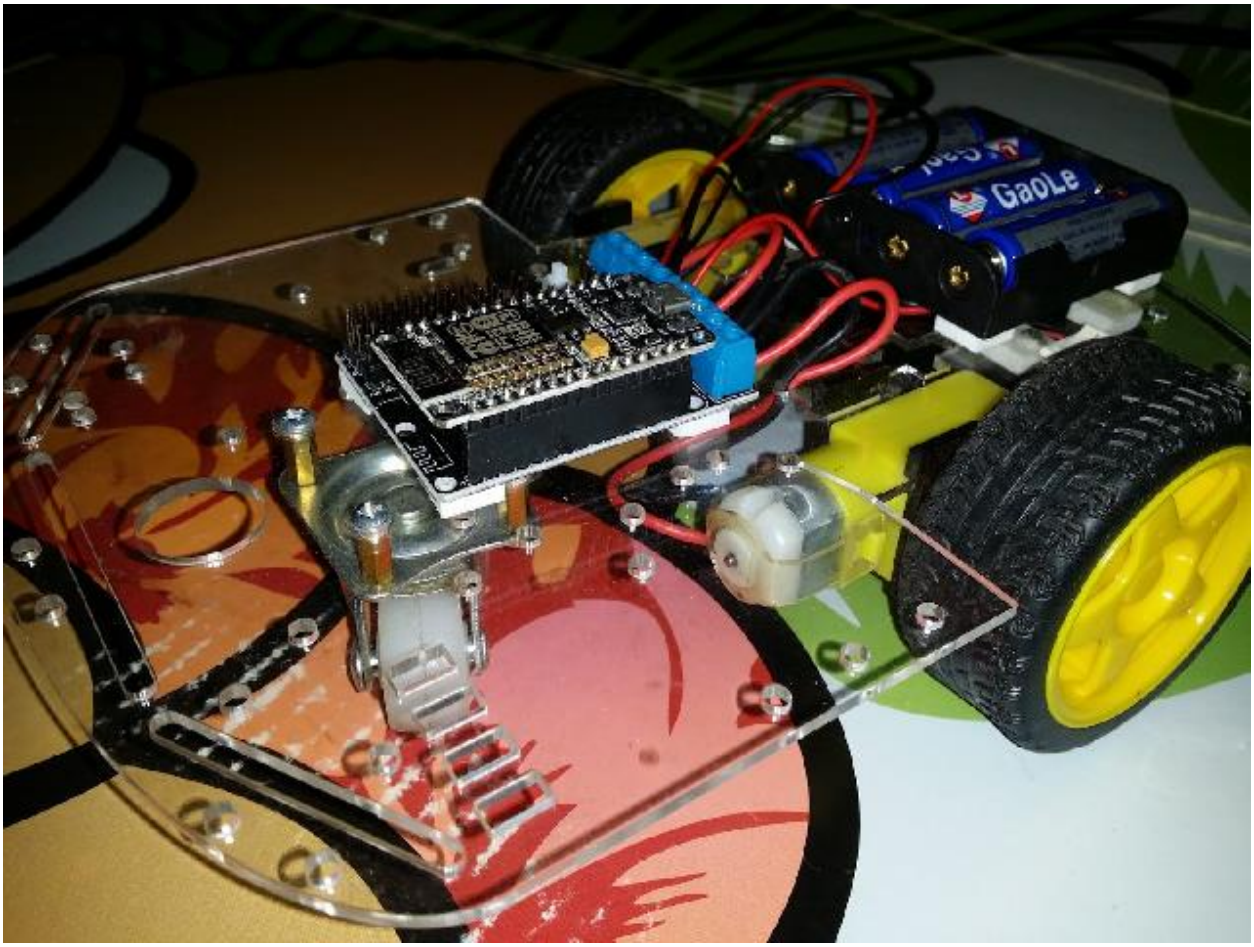


LandShark WiFi

The LandShark WiFi is a 3 wheel robot base that I have incorporated the ESP8266 and Motor Controller Shield to make a simple, low cost, WiFi controlled vehicle. I have collected some sample code and modified some of it to my own taste.

In its current state if is just a simple remote controlled car but has lots of I/O and processing power to add many features. I am currently working on adding an ultrasonic ranger to prevent collisions and send distance measurements back to the controller.

Currently the LandShark WiFi can be controlled by any web browser that supports JavaScript and is connected via WiFi to the ESP8266. If you are using a cell phone or tablet the software utilizes the accelerometer in the device to allow controlling the LandShark WiFi by simply tilting the phone/tablet.



References, URLs and Downloads

This presentation:

http://hacker.instanet.net/ESP8266/ESP8266-Presentation.pdf

ESP8266 Spec:

http://hacker.instanet.net/ESP8266/ESP8266_Specifications_English.pdf

WeMos D1 files:

http://hacker.instanet.net/ESP8266/D1/

NodeMCU Studio:

http://hacker.instanet.net/ESP8266/NodeMCU_Studio/

Custom NodeMCU Images:

http://hacker.instanet.net/ESP8266/NodeMCU_Custom_Images/

ESPlorer:

http://hacker.instanet.net/ESP8266/ESPlorer-master.zip

http://hacker.instanet.net/ESP8266/ESPlorer.zip

NodeMCU Documentation:

http://nodemcu.readthedocs.io/en/dev/en/

http://nodemcu.com/index_en.html