



instructables

Arduino MEGA 2560 With WiFi Built-in - ESP8266

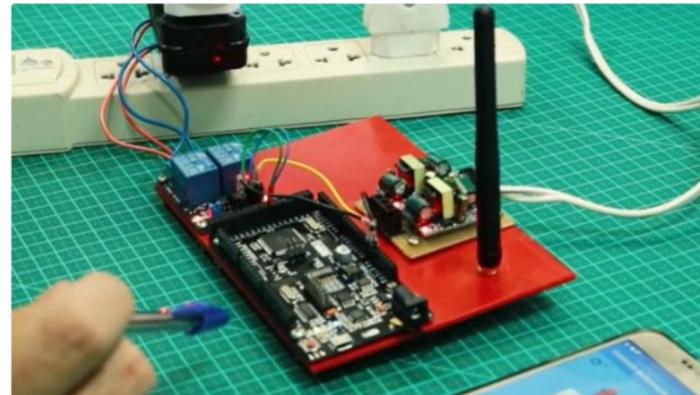
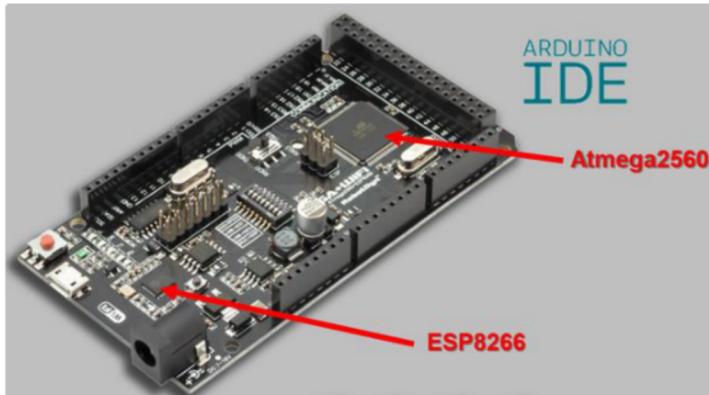


by Fernando Koyanagi

In today's text, we discuss an Arduino that I consider extremely special, as it has an ESP8266 embedded in its board. It doesn't have the ESP12 soldered onto the board. Instead, it has the Espressif chip. So, on the board you have the built-in Tensilica chip with 4MB of memory, along with the ATmega2560, which is the traditional Arduino Mega.

Let's move onto how this Arduino works, and let's do an assembly that shows when you should select ESP or Mega to perform a home automation. With this, we can turn lamps on and off, which is a mechanism that can be very useful for you to make improvements in your house.

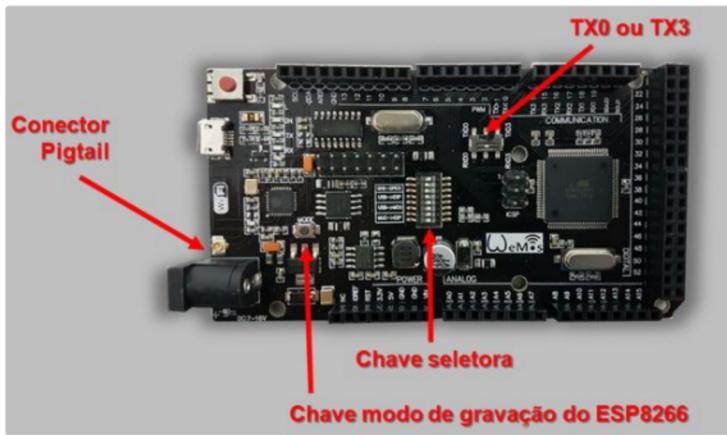
<https://youtu.be/Yj5apiv4wcE>



Step 1: Physical Characteristics of the Board

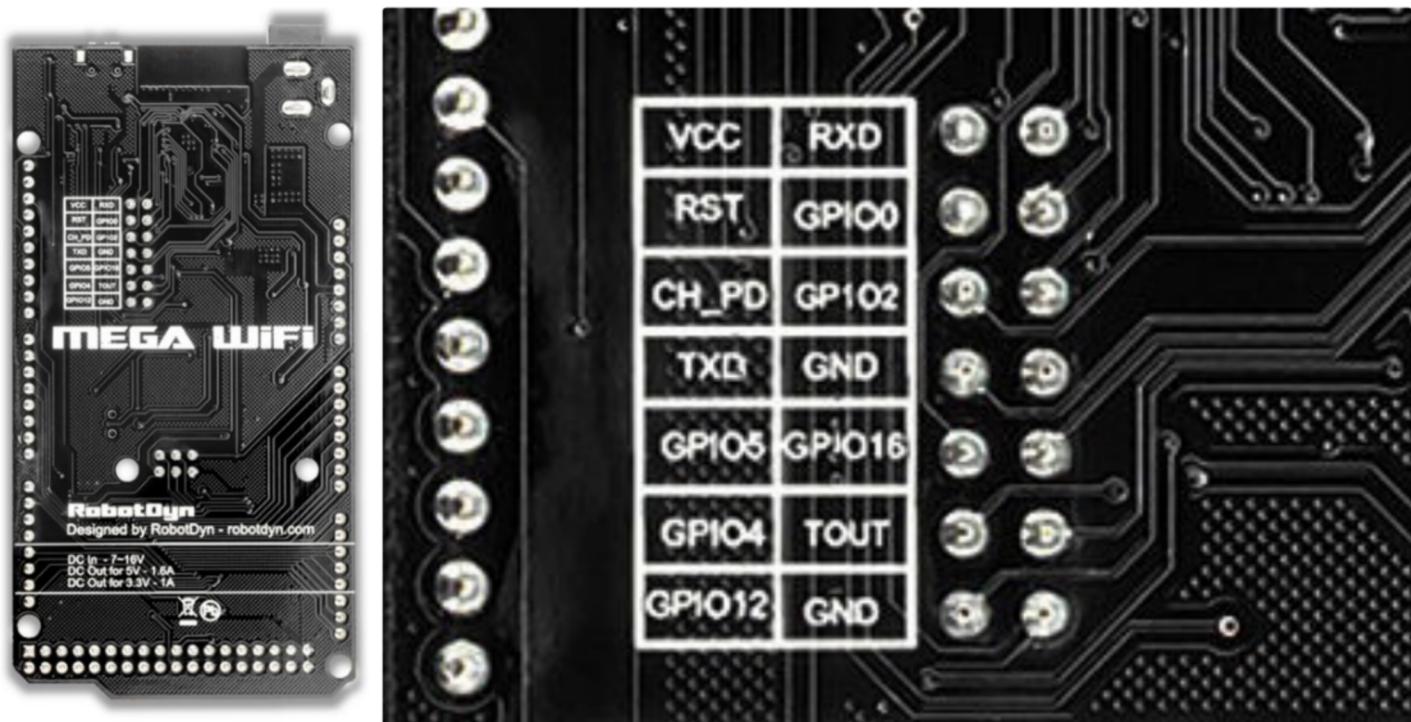
I really like that this Arduino has a Pigtail connector for an antenna. Why is this good? If you connect an antenna on this device, you will have a huge benefit, as it will increase your reach, straight from 90 meters to 240 meters away. I came to this conclusion after a test I did, so I did not have to rely solely on the datasheet manual.

This board also has a selector switch that allows the ESP to interleave the connection between TX0 and TX3, remembering that the ATmega has four serials. A second selector switch is the DIP Switch, and we also have a key recording mode of the ESP8266. All the pinning is completely compatible with the ATmega pinout.



Step 2: Access to ESP8266 Pins

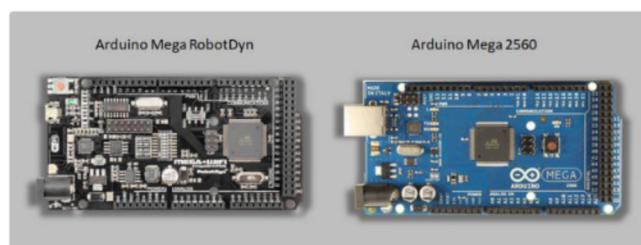
Here, I show the back of the board, where there is a table that displays access to the ESP pins.



Step 3: Comparing the Two Arduinos

Here, we have a comparison between the Mega Arduino with the integrated ESP (Arduino Mega RobotDyn) and the traditional Mega Arduino (Arduino Mega 2560). We can see that they are similar, but in the 2560, we have the USB printer, which is a large connector. However, in the RobotDyn, we have the mini-USB. I particularly like the more compact option, but the power is the same in both.

7 to 12 volts, and the ESP8266 is already powered, and already has a voltage reducer. So, feeding the Arduino is already feeding the ESP, which is already down to 3v3, and internally that microcontroller is already 3v3.



We can see, then, that the intention of the RobotDyn creators was to maintain the architecture of the ATmega.

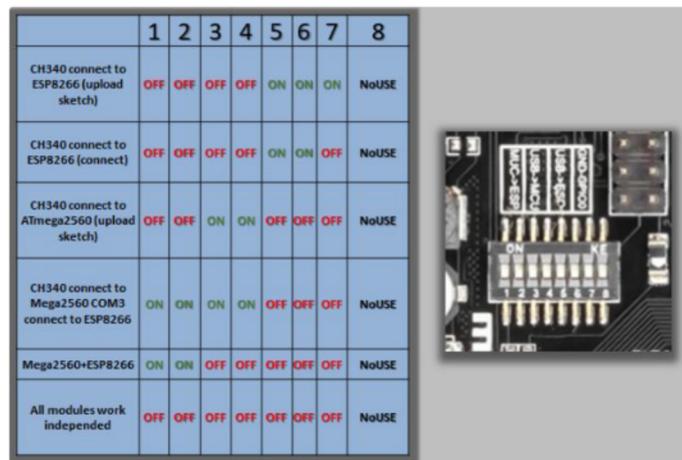
As we see in the table above, the ATmega has 32MB of memory, not counting the ESP memory. This is wonderful, since the traditional Mega Arduino has only 256kb of memory. The power in the RobotDyn is

The processors are the same, 16MHz, and one great advantage of these models is the high amount of IOs.

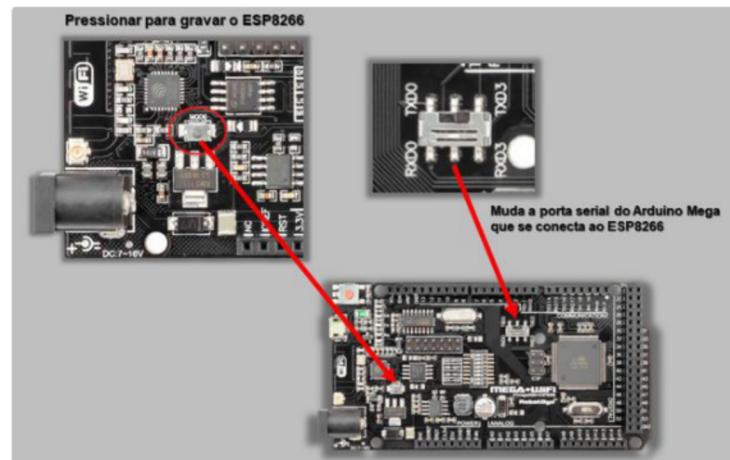
Arduino	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM	Flash [kB]	USB	UART
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1kb	2kb	32kb	Regular	1
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4kb	8kb	256kb	Regular	4
RobotDyn	ATmega2560 Esp8266	5 V / 7-12 V 3v3	16 MHz 80 MHz	16/0 1/0	54/15	4kb	8kb 64kb	32Mb 8Mb	CH340G 1/wifi	4

Step 4: Switch Status and Mode Selection

We have here the DIP Switch and a table with several positions. These help in the connections, depending on your objective. One essential detail is that if you are writing Flash in ESP, you have to be aware of addresses that are even slightly different.



In the image below, we zoomed in on the key that changes the serial port of the Arduino Mega. This connects to the ESP, and also in the key Mode, requiring that we must press the ESP8266 to record.



Step 5: AT Firmware Installation

If you want to use ESP8266 in AT mode, download the PDF file. You should now configure the card so that the ESP8266 is connected to the USB and in recording mode. To do this, set switches 5, 6, and 7 to ON (left) and all other switches to OFF (right).

If you want to use ESP8266 in AT mode, you should configure the Flash Download Tool as follows:

File\bin\esp_init_data_default.bin at address 0x3fc000

File\bin\blank.bin at address 0x37e000

File\bin\boot_v1.4(b1).bin at address 0x00000

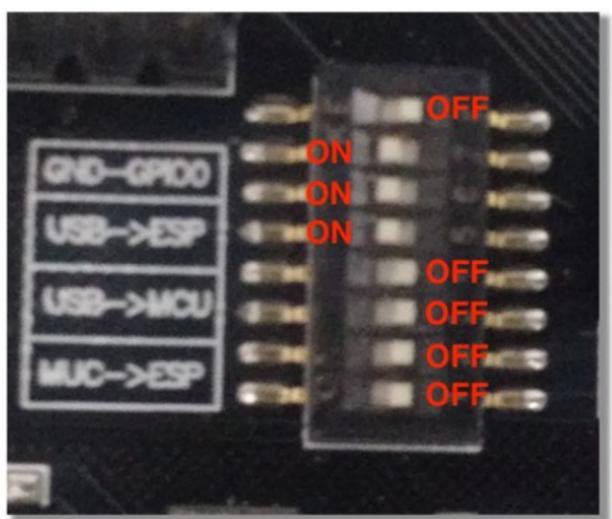
SPI Speed = 80MHz

SPI Mode = DIO

Flash Size = 32Mbit 4mb bytes x 8 bits = 32m bits

Crystal Freq = 26M

File\bin\at\512+512\user1.1024.new.2.bin at address 0x1000



Step 6: Verifying AT Firmware

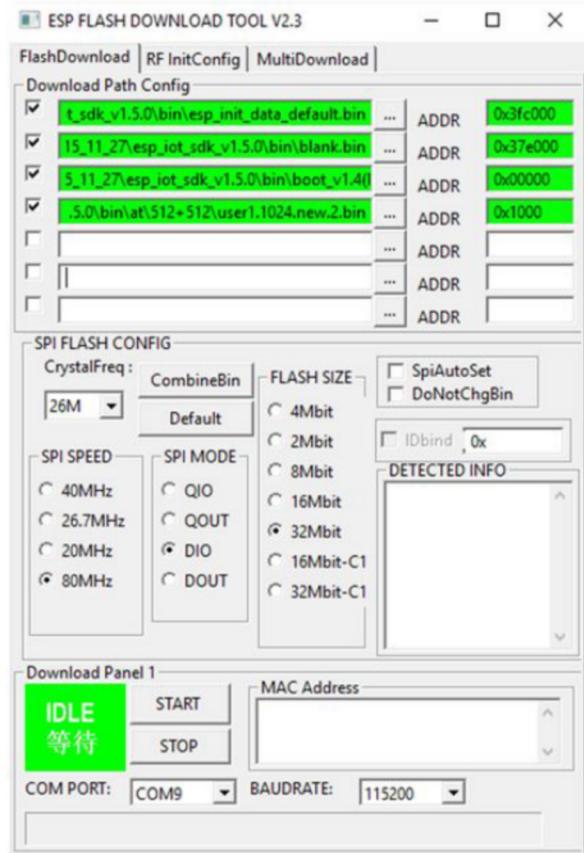
In this part, I used esptool.exe, which is a command tool that accesses the ESP8266 Flash and checks for some settings, such as the chip type and memory size.

```
C:\Python27\Scripts>esptool.exe --port COM9 flash_id
esptool.py v2.1
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266
Uploading stub...
Running stub...
Stub running...
Manufacturer: c8
Device: 4016
Detected flash size: 4MB
Hard resetting...
```

Step 7: Example

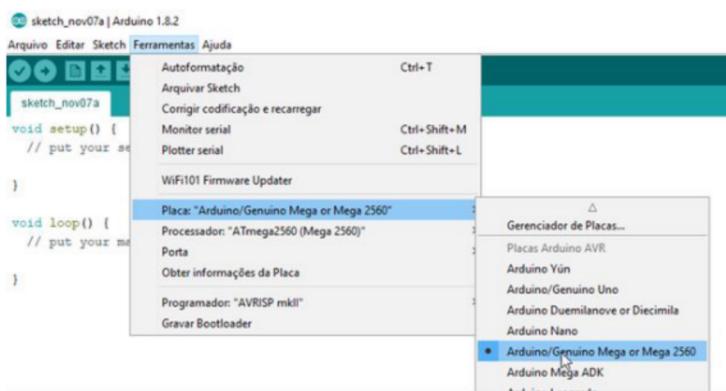
In this example we show the hexadecimal addresses we use to write with the Flash Download Tool.

Also, for those who do not have much experience with the ESP8266, I suggest two of my previous videos: [Recording in ESP01](#) and [Introduction to ESP8266](#).



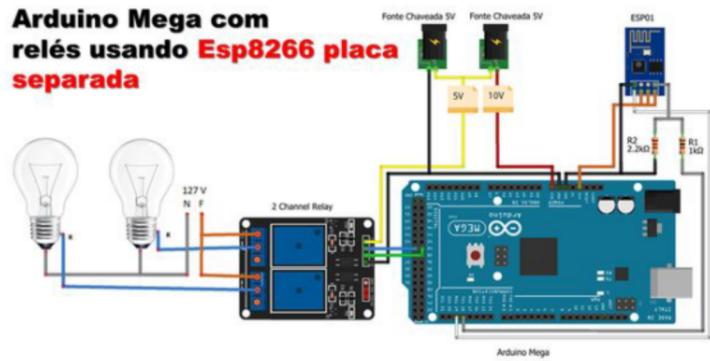
Step 8: Configure Arduino IDE Environment

To record the Arduino, there is no mystery at all. You only need to set up the Mega Arduino 2560 board as if it were a traditional Arduino.



Step 9: Arduino Mega With Relays Using Esp8266 Separate Board

We have here the assembly scheme that I perform in the video. We connected the Arduino Mega to the ESP01 and controlled two relays for one application.



Step 10: Arduino Mega With Built-in Esp8266

Here, we do the same thing mentioned above, but while using the Arduino Mega with integrated ESP. One tip is that you watch the video entitled Residential Automation with Arduino Mega and ESP8266 for more details about this subject.

