

Proyecto

En esta entrada quiero hacer una pequeña relación de los materiales que hacen falta para montarlo, aunque no está completo. De momento estoy terminando de ver lo necesario para darle una larga vida al asunto. Con el controlador Anakin_aquarium_V2 podrás controlar el acuario con una Placa Arduino Mega + ESP8266.

Funciones

- Encendido y apagado de luces les noche (azul) y blanca con efecto amanecer.
- Control más exacto de la temperatura del agua.
- Relleno automático de agua
- Control de temporizadores como el de CO2, lampara UV o Aireador.
- Sensor de pH de agua y corte de CO2 si está fuera de parámetros.
- Supervisión de valores y control mediante aplicación móvil [Blynk](#).
- Aviso de alarmas en móvil cuando haya algo fuera de parámetros.
- Control de bombas peristálticas.
- Control de comedero.
- Control de ventilador de refrigeración.

Estoy desarrollando una placa base PCB donde aparecerán todos los componentes integrados y sea solamente conectar y listo. De todos modos todavía queda un poco. Si alguien se anima a ayudar todo es bienvenido, ya que es todo código abierto para que el que quiera montarlo pueda hacerlo. Es un proyecto colaborativo.

Software

Preparando el IDE

Está realizado con el IDE de Arduino. En concreto estoy usando actualmente la 1.8.3 para OSX. Para poder usar este tipo de placa con el IDE necesitamos hacer previamente unos pasos.

Código

El código lo tengo alojado en GitHub, donde voy subiendo las diferentes versiones. Suelen ser ya estables, aunque voy añadiendo y mejorando cada vez que puedo.

Esta es la web de GitHub: [github](#)

El código está en este ZIP: [ZIP FILE](#)

Para poder subirlo al IDE debemos descomprimir el archivo master.zip, luego debemos renombrar la carpeta "Anakino_Aquarium_V2-master" a "Anakino_Aquarium_V2" es decir, quitarle el -master al nombre.

Una vez dentro, solamente debemos ejecutar el archivo llamado: Anakino_Aquarium_V2.ino o bien abrirlo desde el IDE de Arduino.

Automáticamente se generarán las pestañas diferentes que tiene el código. Está así para tratar de organizarme mejor.

Blynk

Es la app tanto para Iphone como para Android. Nos mostrará la información en la pantalla del móvil y podremos configurar varios parámetros como por ejemplo, la temperatura del agua o la hora de la comida, así como los temporizadores. El número total de Energia necesario es: Y aqui teneis el código de la app:

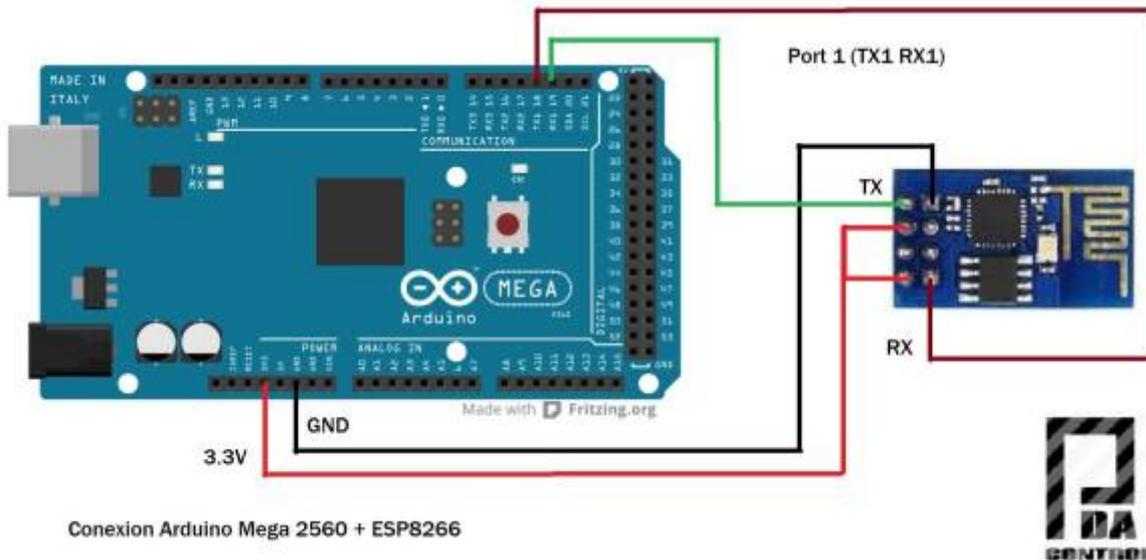


Componentes

- Placa Arduino Mega
- ESP8266
- L7805 (regulador de voltaje 12v → 5v)
- AMS1117 (Regulador de voltaje 12v → 3.3V)
- Sensor de pH
- Sensor TDS
- IRF540 (para dimeado de luces y ventiladores)
- Reles
- Pantalla oled
- Sensores de temperatura ds18b20
- Condensadores 10uF
- Resistencias 4k7

Placa Arduino MEGA con ESP8266

Para eso cuento con la placa Arduino Mega a la que se le ha conectado un ESP8266 según este esquema:

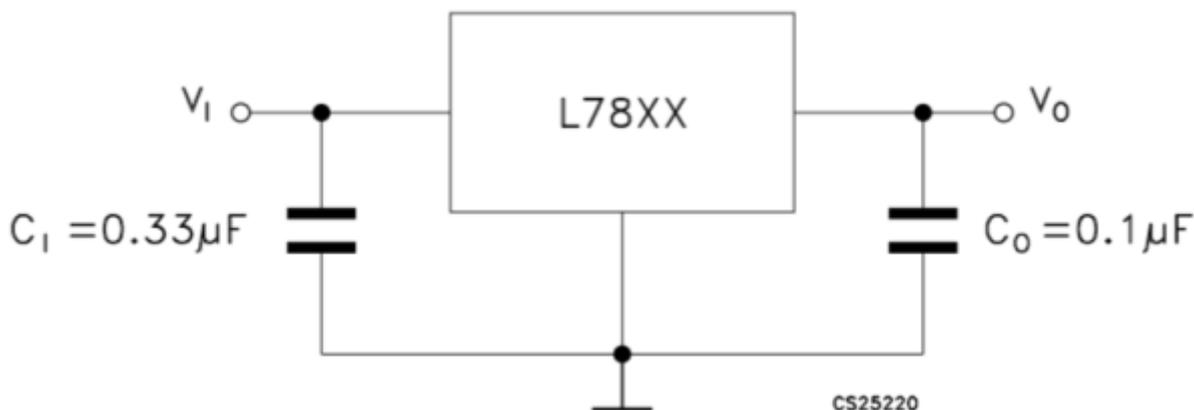


ESP8266

Módulo que utilizaremos para la conexión wifi, Aunque se alimenta a 3.3v la salida del MEGA proporciona la suficiente potencia para su funcionamiento.

L7805

Regulador de voltaje de 12v a los 5v que necesita la placa MEGA y los módulos que vamos a instalar.



IRF540

Tengo pensado usarlo para poder dimmear tanto la luz, como quizás los ventiladores que refrigerará

el agua.

Reles

Modulo de 4 Reles de estado sólido para Arduino. Se utilizan de 5v para controlar los diferentes componentes a 220v.

Pantalla oled

“1.3 inch IIC I2C Serial 128×64 OLED LCD Display White Screen” esta es la descripción de la pantalla oled que he comprado por Ebay.

Al ser conectada por I2C podemos ahorrar pines aunque con esta placa tenemos de sobra.



Esta nos mostrará información básica, como temperatura o el valor del pH. La pantalla no es muy grande.

DTH22

Sensor de humedad y de temperatura, utilizado para conocer los parámetros de la habitación del acuario.

Sensores de temperatura

Utilizaremos dos sensores de temperatura sumergibles ([DS18B22 como estos](#)) pensados, uno para controlar la temperatura del agua y otro para la temperatura ambiente.

Reles

Los reles de estado solido van a controlar los 4 enchufes que servirán para manejar el calentador, el aireado, la lampara uv y el CO2

La conexión con la placa se hará por cable de red que permite poner los enchufes es un sitio cómodo y seguro.

| In | LAN | POE | Out |
|----|------|-------|-----|
| 1 | Data | Data | 1 |
| 2 | Data | Data | 2 |
| 3 | Data | Data | 3 |
| 4 | N.C. | +Volt | 4 |
| 5 | N.C. | +Volt | 5 |
| 6 | Data | Data | 6 |
| 7 | N.C. | GND | 7 |
| 8 | N.C. | GND | 8 |

INPUT: 12V-48V DC 



Montaje

Parte A

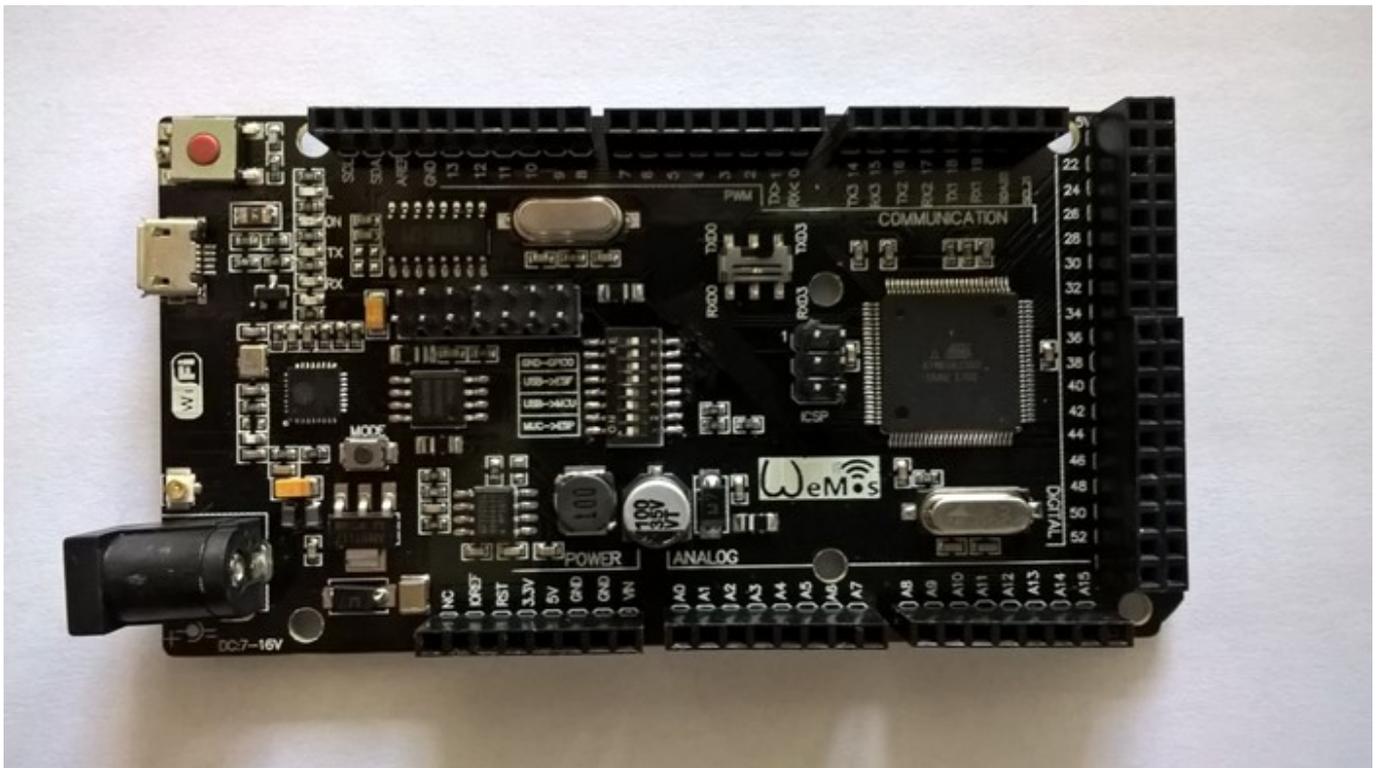
Aqui estará la placa mega con la pantalla y las conexiones de los diferentes sensores y activaciones

de ventiladores y sensores.

Parte B

En esta parte estará la caja con los enchufes a 220v, el transformador para alimentar la Parte A. De esta manera minimizamos componentes.

Placa Mega Wifi de Robotdyn (alternativa)

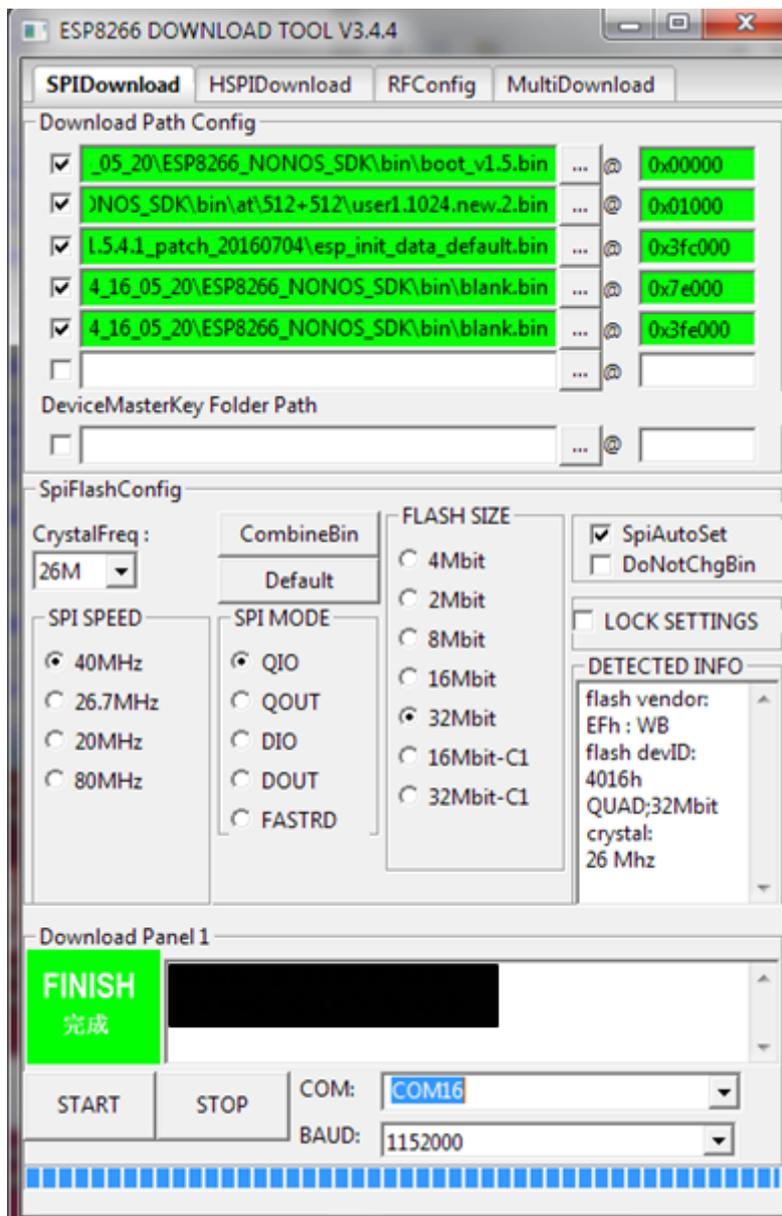


In the bin folder coming from the .zip file are the files needed to flash the ESP8266.

Please note that in the folder from the patch is a newer esp_init_data_default.bin file to be used. In the AT folder under the bin file are folders and files.

In the README.txt file are instruction on what file to store where in the ESP8266 memory location. As the Robotdynam UNO + WiFi has ESP8266 which is stated to be 8Mb I used:

Which means files from the 512+512 were used.



After flashing the ESP it finally started to reply to AT commands.

Flash size 32Mbit: 512KB+512KB

| | |
|---------------------------|---------------------|
| boot_v1.2+.bin | 0x00000 |
| user1.1024.new.2.bin | 0x01000 |
| esp_init_data_default.bin | 0x3fc000 (optional) |
| blank.bin | 0x7e000 & 0x3fe000 |

Upload code

Para subir el código es importante fijarse en los interruptores:

```
1- OFF
2- OFF
3- ON
4- ON
5- OFF
6- OFF
7- OFF
RXD0 x TXD0
```

¡Ten cuidado con la posición de los interruptores!

1. Para esto deben estar el 3.ON 4.ON los otros en OFF.
2. I have downloaded an example 5 and made some changes.
3. You need to change Serial from 1 to 3 like this: #define EspSerial Serial3.

(exactly this i was solving four months) Code should look like this:

```
#define BLYNK_PRINT Serial

#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "...";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "...";
char pass[] = "...";

// Hardware Serial on Mega, Leonardo, Micro...
#define EspSerial Serial3

// or Software Serial on Uno, Nano...
// #include <SoftwareSerial.h>
// SoftwareSerial EspSerial(2, 3); // RX, TX

// Your ESP8266 baud rate:
#define ESP8266_BAUD 115200

ESP8266 wifi(&EspSerial);

void setup()
{
```

```
// Debug console
Serial.begin(9600);
delay(10);

// Set ESP8266 baud rate;
EspSerial.begin(ESP8266_BAUD);
delay(10);

Blynk.begin(auth, wifi, ssid, pass);
// You can also specify server:
//Blynk.begin(auth, wifi, ssid, pass, "blynk-cloud.com", 80);
//Blynk.begin(auth, wifi, ssid, pass, IPAddress(192,168,1,100), 8080);
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```

Esto es el código mínimo para conectar la placa y probarlo. Cuando la conexión sea satisfactoria podrás ver como el status de tu aplicación Blynk pasa a estar ONLINE.

Ajuste de los interruptores

Para que funcione el código deben estar los interruptores así.

```
1- ON
2- ON
3- ON
4- ON
5- OFF
6- OFF
7- OFF
RXD3 x TXD3
```

3.1. esto permite la comunicación entre el MEGA y el ESP8266 y el puerto serie en tu PC. This setting let you see serial print and at the same time it will communicate with ESP. 3.2. switch near ICSP set to After moving all necessary switches go to the next step.

Conecta la placa, abre tu aplicación y tomate una cerveza!

That's all folks...

Problemática.

Wrong Dip-setting Weak WIFI signal Wrong SSID, PASS, AUTH Use external power supply Possibly it could be high baud of ESP (never happen to me). You can reduce it to 9600 (to the same level as

Serial.print) For reducing baud use this

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

or

```
AT+CI0BAUD=9600
```

it should reply

```
OK
```

From:

<https://www.atorcha.es/> - **Atorcha**

Permanent link:

https://www.atorcha.es/anakino/anakino_aquarium_v2

Last update: **20:29 19/05/2025**

