



ESPino - Especificaciones

Resumen

Microcontrolador	ESP8266 (32-bit RISC)
Comunicación	WiFi 802.11 (station, access point, P2P)
Voltaje de operación	3.3V
Voltaje de entrada	4.4-15V
Pines de I/O Digitales	9
Pines de entrada analógica	1 (10-bit ADC)
Corriente DC por Pin I/O	12 mA
Corriente DC máxima para el Pin 3.3V	800 mA
Memoria Flash (Programa)	4 MB
Instruction RAM	64 KB
Data RAM	96 KB
Boot ROM	64 KB
Velocidad de reloj	80 Mhz

ESPino viene programado con el firmware node-mcu, el cual nos permite programarlo de forma fácil en el lenguaje de programación Lua. Puedes ver la documentación de node-mcu aquí:

https://github.com/nodemcu/nodemcu-firmware/wiki/nodemcu_api_en

De igual forma, se puede programar directamente en C o C++ por medio de herramientas como el IDE de Arduino con soporte para ESP8266.

Alimentación

ESPino se puede alimentar directamente desde la entrada USB (5V) o, desde el pin **Vin** con una fuente de entre 4.4V a 15V. El regulador LDO integrado se encarga de regular este voltaje a los 3.3V necesarios para el funcionamiento de la placa.

Los pines de alimentación son los siguientes:

- **Vin**: Entrada de alimentación DC no regulada externa. El regulador integrado en la placa acepta entradas de entre 4.4V a 15V y otorga una salida de 3.3V a máximo 800mA
- **3.3V**: Salida del regulador interno de 3.3V
- **GND**: Pines de tierra

Comunicación



El chip ESP8266 integrado en la placa otorga comunicación inalámbrica por medio del estándar WiFi (802.11 b/g/n/d/e/i/k/r), con lo que se puede comunicar de forma fácil directamente a internet. Éste puede ser configurado para funcionar como estación (cliente de un router), Access point (crea una red a la cual se pueden conectar otros dispositivos, por ejemplo, un smartphone), o modo WiFi Direct (P2P).

Además, incluye soporte para los siguientes métodos de comunicación alámbrica:

- UART (Serial)
- SPI
- I2C

Para más detalles, ver la hoja de especificaciones del ESP8266.

Dispositivos integrados

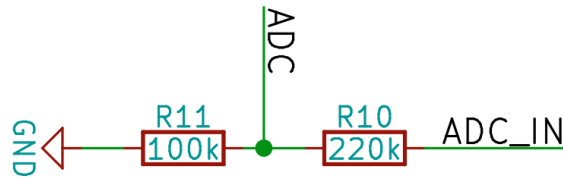
La placa cuenta con un LED RGB (rojo, verde y azul) y dos botones, de los cuales uno es RESET y el otro, USER/PROG, está conectado al pin 0 del ESP8266 y sirve para ponerlo en modo "Bootloader" al iniciar el dispositivo, o puede ser usado por el programa del usuario una vez que éste arranca.

Estos dispositivos se encuentran conectados de la siguiente forma:

- LED Rojo: Pin 2
- LED Verde: Pin 5
- LED Azul: Pin 4
- Botón USER/PROG: Pin 0
- Botón RESET: Pin RST

ADC

El chip ESP8266 incluye un ADC de 10 bits que puede leer una entrada en el rango de 0V a 1V. Para facilidad de prototipado, la placa integra un divisor de voltaje en el pin ADC que permite que la entrada sea de 0V a 3.3V.



Programación serial

ESPino integra un chip USB-Serial y un conector micro-USB que permite conectarlo a una PC, ésta lo reconocerá como un puerto serial estándar, lo que nos permitirá tanto programarlo como utilizar monitores seriales para comunicarnos con la placa.

En caso de que la placa no sea reconocida automáticamente por el sistema operativo, puede que sea necesario instalar los drivers del chip USB-Serial, los puedes descargar de la página espino.io.

Una vez conectada la placa a la PC, podemos programarla de 2 formas:

1. Con el lenguaje de programación Lua del firmware integrado node-mcu (sin entrar a modo "Bootloader")
2. Con un firmware propio escrito en C o C++, por ejemplo desde el IDE de Arduino con soporte para ESP8266. Para esto es necesario entrar en modo "Bootloader"

Modo Bootloader

El modo "Bootloader" nos sirve para subir un nuevo firmware al ESP8266, o restaurar el firmware node-mcu que viene por defecto.

Pasos para poner el ESPino en modo Bootloader:

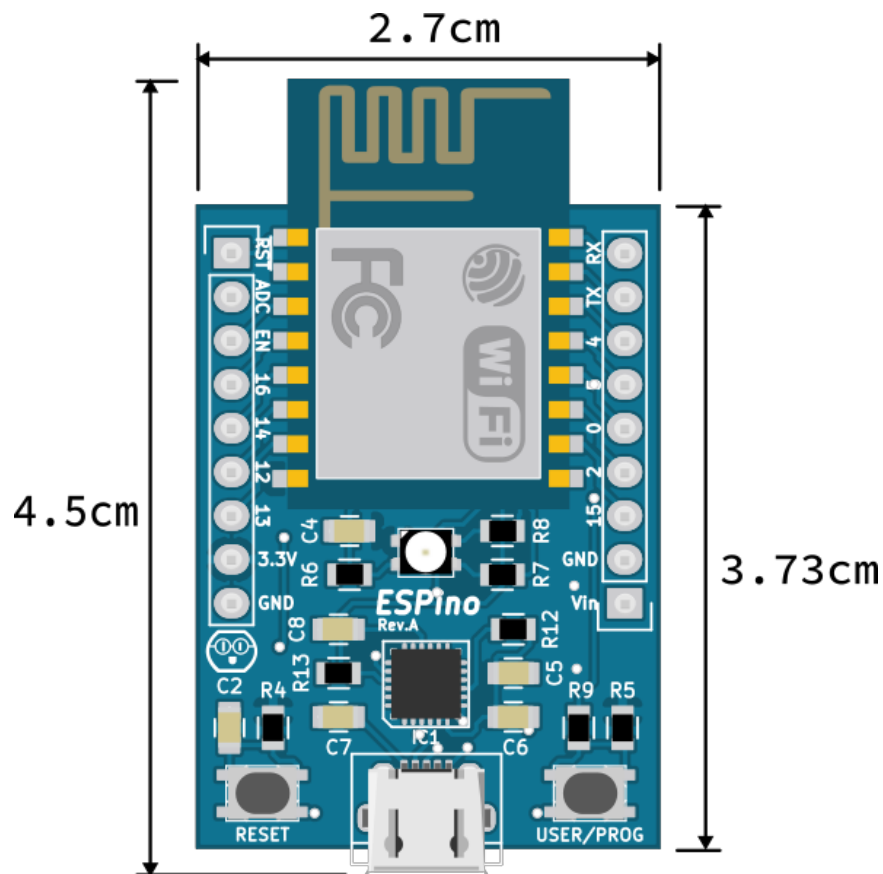
1. Presiona los botones RESET y USER/PROG al mismo tiempo sin soltarlos
2. Suelta el botón RESET, manteniendo USER/PROG presionado
3. Espera un segundo y suelta USER/PROG

Una vez en modo bootloader, podemos usar una herramienta como "esptool" o el IDE de Arduino con soporte para ESP8266 para subir el firmware.



Dimensiones físicas

Los pines de ESPino están diseñados para poder insertarse en un protoboard con espaciado estándar de 2.54mm.

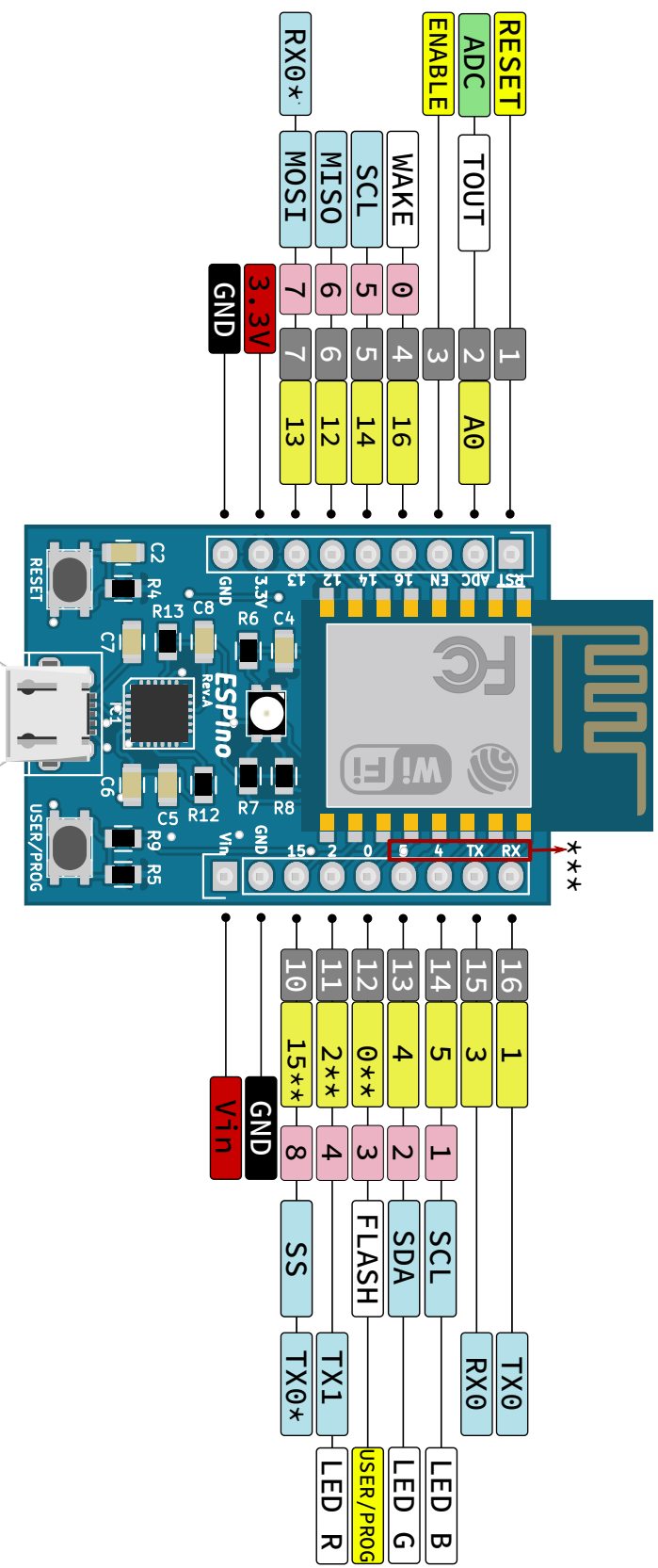
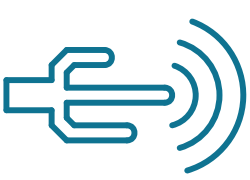


Soporte

- Para descargas, tutoriales y más información, entre a <http://espino.io>
- Para soporte y cualquier duda, entre al foro de la comunidad Aquila: <http://community.aquila.io/>
- Puede contactarnos directamente por correo electrónico a: info@makerlab.mx

ESPino

PINOUT DIAGRAM



Notas:

- Los pines 0 a 15 pueden ser PWM, ** Los pines 2 y 0 deben estar en alto y el pin 15 debe estar en bajo al inicio.

* Los pines RX0*(13) y TX0*(15) pueden ser usados como puerto serial en sustitución de TX0 y RX0.

En el IDE de Arduino esto se logra llamando Serial.swap() después de Serial.begin()

*** En los ESPino Rev. A, las etiquetas RX y TX, 4 y 5 están invertidas por error



makerlab.mx

	GND
	Alimentación
	Control
	Pin físico
	Pin en Arduino IDE
	Función Extra
	Pin Analógico
	Pin PWM
	Comunicación
	Pin en node-mcu