

ESP8266: Un microcontrolador para el Internet de las Cosas

Jorge Valderrama¹ y Ebert Brea^{2*}

¹ Universidad Central de Venezuela

² Universidad Central de Venezuela

*ebertbrea@gmail.com

RESUMEN

En el mundo de los microcontroladores la interconexión entre los sensores, los sistemas de control y los usuarios, es fundamental para llevar a cabo funciones de control, monitoreo y automatización. Las nuevas tecnologías desarrolladas apuntan a lo que hoy en día se conoce como el Internet de las Cosas (IoT), la cual busca la comunicación entre los equipos de las industrias, de los hogares y de los usuarios de modo automático. El ESP8266 fabricado por Espressif es un microcontrolador que incluye su propia antena, además de un conjunto de puertos de comunicación y un amplio conjunto de instrucciones para la comunicación vía TCP/IP, lo que permite de una forma sencilla conectarse a internet por medio de WIFI. Adicional a esto, el ESP8266 cuenta con software de desarrollo para el diseño de programación de aplicaciones que permite conectarse con plataformas del IoT. Se demostrará entonces mediante dos métodos con ejemplos, el uso de este dispositivo para comunicarse con plataformas del IoT. En el primero de los ejemplos se utilizó las librerías suministradas por una página WEB que ofrece el servicio gratuito de publicación y se enviaron datos con el protocolo MQTT, este método resulta sencillo de utilizar pero muy limitado para configurarlo en detalle pues las librerías no están comentadas y resulta complicado determinar cuáles herramientas están en uso y cuales parámetros fueron los elegidos para la configuración. En el segundo ejemplo se realizó el código en ANSI-C para enviar datos con el protocolo TCP/IP a una dirección de IP virtual genérica, este método es mucho más complejo pues es necesario realizar todas las configuraciones para los protocolos, sin embargo nos permite tener un control total sobre todas las herramientas y rutinas que está ejecutando el microcontrolador.

Palabras Clave: ESP8266, Internet de las Cosa, microcontrolador, WIFI, MQTT, TCP/IP, Control, Monitoreo, Automatización, Arduino, ANSI-C.

INTRODUCCIÓN

El ESP8266 es un microcontrolador perteneciente a una nueva generación de computadores embebidos que ya no sólo limita sus comunicaciones a medios alámbricos y radio sino que también dispone de conexión inalámbrica a la Internet por WIFI.

El uso de los microcontroladores principalmente cumplen funciones de control, monitoreo y automatización en tiempo real. Para el ESP8266 estas funciones siguen siendo las mismas pues dispone de las herramientas de hardware y software necesarias que le permite ejercer dichas funciones, además de esto al disponer de la comunicación por WIFI en conjunto con una antena integrada hace que dichas funciones las pueda realizar de una forma mucho más completa.

La capacidad de conectar el microcontrolador por WIFI implica tener la posibilidad de enviar o recibir datos por la Internet mediante algún protocolo, lo que se traduce en interactuar de forma directa con el usuario o con un bróker automatizado.

Basado en lo descrito anteriormente, el presente trabajo consta de dos ejemplos de interconexión a la Internet por WIFI. Se utilizó el microcontrolador ESP8266 con la versión *ESP-E12* embebido en la tarjeta de desarrollo NodeMcu V1.0.

METODOLOGÍA

El diagrama esquemático de los puertos del microcontrolador que utilizaremos se observa en la Figura 1.

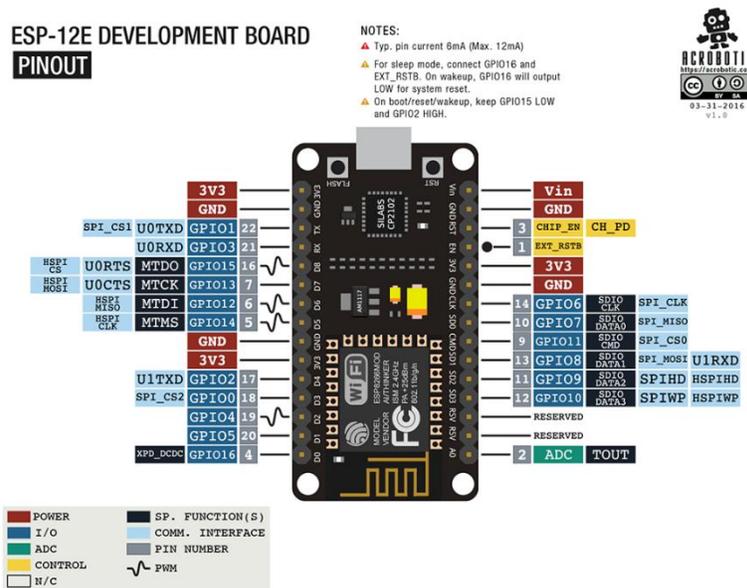
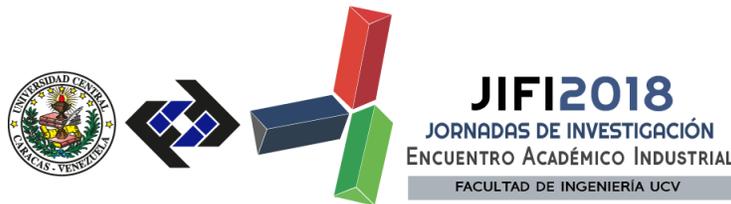


Figura 1. Diagrama esquemático de los puertos en la tarjeta de desarrollo ESP-12E

Del esquemático mostrado en la Figura 1 se puede observar que la tarjeta incluye: 17 puertos multifuncionales con capacidad de funcionamiento de propósito general para entrada o salida (*GPIO*), 1 puerto ADC, 3 puertos de entradas/salidas de 3.3V, 1 puerto de entrada/salida de 5V, 4 puertos de conexiones de tierra, 2 diodos led de iluminación, 1 transformador 5V/3.3V, 1 botón de *Reset*, 1 botón de *Flash*, 1 conector TTL a microUSB.

Algunas de las características de hardware del microcontrolador con la versión ESP-E12 serán las siguientes:

- Procesador: *Tensilica L106 32bit* de bajo consumo que alcanza una velocidad de hasta 160MHz.
- Memoria: 4Mb, expandible por el puerto SPI hasta 16 Mb.
- Interfaces: UART, Puertos GPIO, I2C, I2S, SDIO, PWM, ADC, SPI.



WIFI: Certificado por *WI-FI Alliance*, con protocolo 802.11 b/g/n, rango de frecuencia entre 2.4 y 2.5 G.

Voltaje de operación: 2.5-3.6V directamente en el microcontrolador.

Corriente de operación: 80mA.

Clock: Típicamente 40MHz.

Algunas de las características de software del microcontrolador ESP8266 serán las siguientes:

WIFI: En los modos de Station/SoftAP/SoftAP+Station.

Seguridad: WPA/WPA2/WEP.

Firmware: UART/OTA.

Protocolos de red: IPv4, TCP/UDP/HTTP/FTP.

Timers por software y timers por hardware: Dispone de un timer por hardware.

La programación del microcontrolador se puede realizar en lenguaje ANSI-C utilizando alguno de los dos kit de desarrollo de software oficiales del ESP8266 que son: el Non-OS SDK y el RTOS SDK. También se puede utilizar otros lenguajes de programación como el Lúa, Python, Arduino, etc. con los kit de desarrollo de software no oficiales de cada uno de los lenguajes. Estos kit de desarrollo se manejan a través de comandos API y son las herramientas necesarias y suficientes para realizar la programación y configurar el dispositivo.

Para objeto de este trabajo se utilizó el programa Platformio con su ambiente de desarrollo integrado Atom.

Para configurar las conexiones al Internet de las Cosas se muestran dos grandes corrientes, a continuación se mostrará un ejemplo para cada una de ellas:

El primero de los ejemplos se realizó por un método más automatizado, este tipo de configuración se realiza con librerías ya escritas y suministradas por las páginas WEB que ofrecen servicios de nube, para este ejemplo publicaremos la información en la nube de la página WEB <https://io.adafruit.com/>.

Como para la gran mayoría de estas páginas WEB, las librerías que utiliza están escritas en lenguajes de programación Arduino. Para este lenguaje los comandos utilizados para iniciar la conexión a la red WIFI se resumen en escribir el nombre y contraseña de dicha red y llamar una rutina ya escrita en las librerías del kit de desarrollo de software.

Para conectarse con la nube de Adafruit se descargaron las librerías *Adafruit_MQTT.h* y *Adafruit_MQTT_Client.h* que están escritas en lenguaje Arduino y se pueden encontrar y descargar directamente desde la página WEB de Adafruit o desde el ambiente de desarrollo Atom en la pestaña de librerías. En la Figura 2 se muestra en la opción “Libraries” donde se pueden descargar dichas librerías.

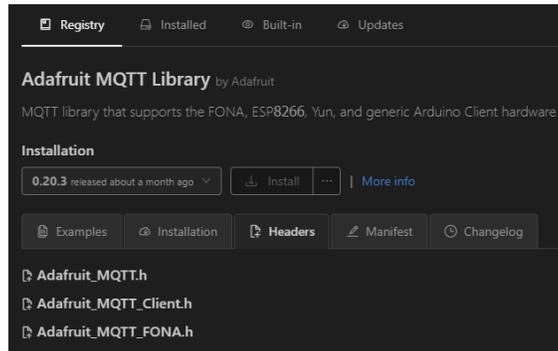


Figura 2. Librerías desde el IDE Atom.

Para poder configurar el microcontrolador de forma tal que pudiese enviar información a una cuenta en específica de la nube fue necesario registrarse en la página WEB de Adafruit y así obtener un usuario y una llave llamada “AIO KEY” que permite al servidor identificar que la información que se enviaba con esos datos pertenecía a esa cuenta creada. En la Figura 3 se muestra un ejemplo de dicha llave.

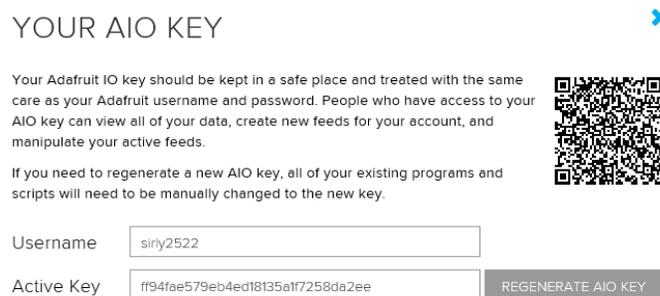


Figura 3. Ejemplo de usuario y llave identificación en la página WEB de Adafruit.

Teniendo ya la cuenta registrada en la página de Adafruit se procedió a crear el Feed por cada dato que se envió y lo configuramos para luego poder presentarlos en la WEB de adafruit según nuestra conveniencia. En la Figura 4 se muestra las ventanas para la creación y configuración de un feed.

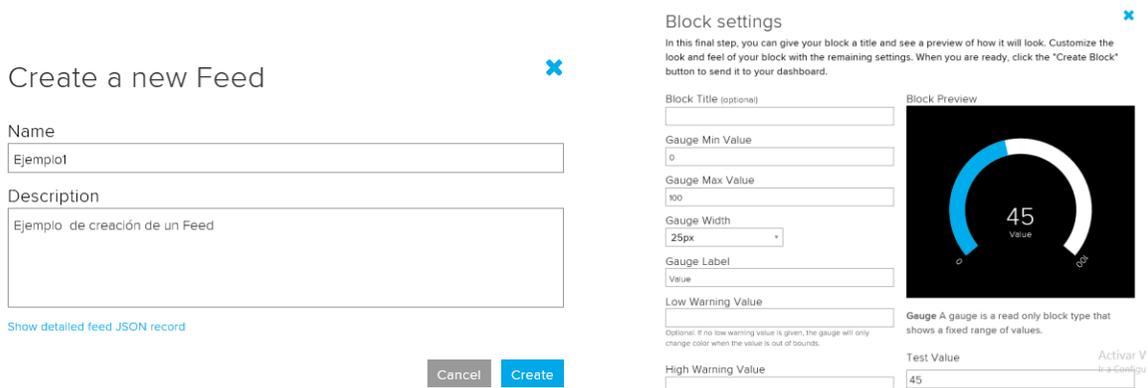


Figura 4. Creación de un Feed (izquierda), configuración de un Feed (derecha).

Una vez que fue creado y configurado cada Feed para ser presentados en la página de WEB se procedió a escribir el código para el microcontrolador. Mediante la utilización de algunas líneas de código básicas se pudo realizar la publicación de información, para el ejemplo que se realizó se envían tres valores. El primero de ellos es el feed del valor medido por el convertidor analógico digital que se envía cada segundo, el segundo de ellos es el feed de alerta que envía los valores del convertidor analógico digital cuando el valor medido es superior a -6 y el tercer valor es el feed de promedio y envía el valor promedio de las mediciones que se toman cada minuto. El código escrito se puede conseguir en un repositorio de Github (*JIFI-Ejemplo-1-ADAFRUIT* [1]).

El segundo de los ejemplos se realizó en lenguaje de programación ANSI-C con el kit de desarrollo de software oficial Non-OS del microcontrolador ESP8266.

Ya que no se utilizó ninguna librería externa fue necesario escribir el código para su total funcionamiento, sin embargo ya que el microcontrolador dispone de una pila de instrucciones para realizar la conexión a una red WIFI y para comunicaciones por TCP/IP se simplifica la configuración utilizando únicamente los comandos API que incluye el SDK. En vista de la cantidad y complejidad de estos comandos se utilizó la guía de prácticas para el microprocesador ESP8266 de *Valderrama* [2] para seleccionar los comandos que permitan configurar de forma básica el WIFI y el protocolo TCP/IP.

El código que se programó incluye la conexión a una red WIFI en modo estación sin embargo, se debe mencionar que con algunos ajustes en el código se pudo haber configurado en modo estación y punto de acceso simultáneamente. En esta otra configuración el microcontrolador hubiese creado su propia red WIFI como punto de acceso, tomado los datos que otros dispositivos le transmitieran y posiblemente enviarlos a algún servidor en modo estación.

Una vez conectado al WIFI se configuró el modo de cliente para un servidor TCP/IP con los protocolos de respuesta ante la reconexión y conexión con el servidor.

Ya obtenida la conexión con el servidor se estableció las rutinas ante el envío de información, recepción de información, carga de información, en el búfer de transmisión y desconexión con el servidor al que está conectado.

El código escrito se puede conseguir en un repositorio de Github (*JIFI-Ejemplo-2-ANSI-C* [1]).

Ya que el código con el que se configuró el dispositivo le permite conectarse a una dirección de IP genérica entonces fue necesario utilizar el programa “*TCP&UDP Debug*” para crear un servidor virtual y poder interactuar con el microcontrolador.

RESULTADOS Y DISCUSIÓN

Para el primero de los ejemplos se logró exitosamente realizar la publicación de cada feed en la página WEB de Adafruit. En la Figura 5, se muestra la información visible en el link de la pizarra creada en la nube de Adafruit.

sirly2522 / Dashboards / Monitoreo de Temperatura ESP-8266

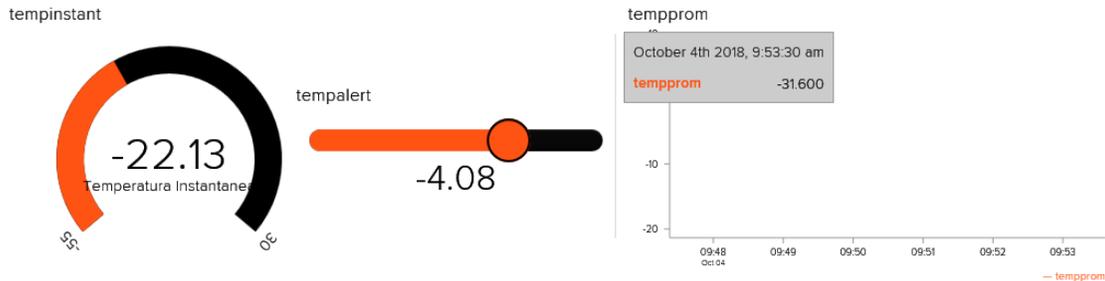


Figura 5. Datos publicados en la nube de Adafruit.

El segundo de los ejemplos logró exitosamente establecer la conexión con el servidor TCP/IP y comunicarse con el protocolo creado. En la Figura 5, se muestra la información mostrada por el microcontrolador y por el servidor cuando realizan el protocolo de comunicación.

```
connected with ESP8266, channel 6
dhcp client start...
ip:192.168.43.199,mask:255.255.255.0,gw:192.168.43.1

Conexión realizada al wifi en modo estación

Cliente TCP conectado

Conectando a servidor TCP/IP

Conexión TCP exitosa!!!

Envío exitoso de información!!!
pm open,type:2 0
tcp received !!! _Hola ESP8266 desde el servidor!

Envío exitoso de información!!!
```

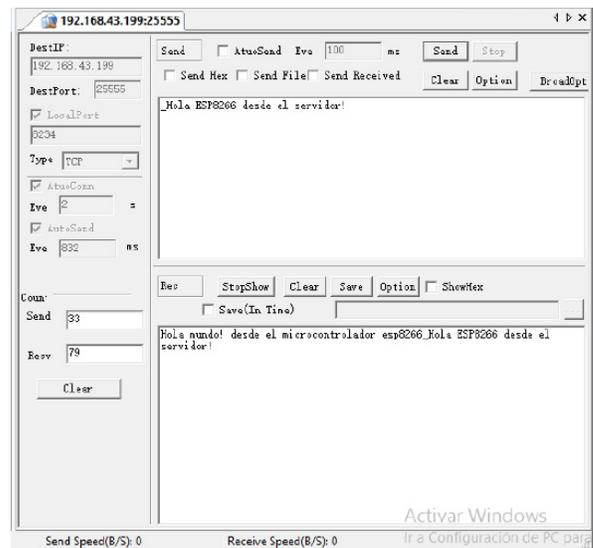


Figura 6. Información mostrada por el ESP8266 (izquierda), Información mostrada por el servidor (derecha).

Se puede observar que para el primero de los ejemplos los resultados resultan estar mejor presentados que en segundo de los ejemplos. Esto ocurre porque para el primer ejemplo la página WEB donde se están presentado los valores enviados ofrece dichas características visuales, sin embargo al ser una aplicación web que no es propia se está limitado a las características que la aplicación puede ofrecer y no se puede modificar la interface y protocolos. El segundo de los ejemplos solo envía la trama de transmisión y la presentación de los resultados quedara de parte del bróker o remota que reciba la transmisión.

En el primer ejemplo el código de programación está limitado a realizarlo en lenguaje Arduino pues las librerías que utiliza fueron escritas en dicho lenguaje de programación. Este es un factor fundamental que tener en cuenta cuando se realiza un trabajo o investigación con este microprocesador. El lenguaje Arduino está sujeto al uso de múltiples librerías incluso para funciones básicas, modificando registros y utilizando recursos del microprocesador sin que el desarrollador del código que se está escribiendo esté necesariamente consciente de ello, esto resulta en errores de funcionamiento para programas un poco más complejos. El segundo ejemplo realiza la programación en lenguaje ANSI-C y utilizando el kit de desarrollo de software oficial, esto permite tener un control total sobre las herramientas y funciones del microcontrolador sin importar la longitud del código que se está escribiendo, sin embargo será imprescindible tener un dominio sobre los comandos API del SDK para poder escribir los protocolos y rutinas en general.

En base a los ejemplos presentados anteriormente quedan entonces las siguientes preguntas para discusión:

1. ¿Es el microprocesador ESP8266 un microprocesador para el internet de las cosas?
2. ¿En qué punto deja de ser viable el uso de librerías externas para realizar códigos de aplicaciones complejas?
3. ¿Siguiendo siendo el lenguaje de programación ANSI-C una herramienta viable para programar las nuevas funciones que ofrecen los microcontroladores?
4. ¿Es el internet de las cosas la siguiente generación en la interconexión para los procesos de automatización?

CONCLUSIONES

El microcontrolador esp8266 es la herramienta ideal para el internet de las cosas pues dispone de las herramientas de hardware y software necesarias para realizar la conexión a internet por WIFI y comunicarse mediante distintos protocolos con otras remotas, sensores, páginas web, etc. Además de esto es compatible con múltiples protocolos de comunicación lo que le permite el uso de las herramientas que cada ambiente de desarrollo incluya.

Al tener la capacidad de ser programado utilizando distintos lenguajes de programación para las librerías que los servidores ofrecen hace que el dispositivo se convierte en la herramienta ideal para conectarse al Internet de las Cosas. Se debe destacar que para realizar programas que incluyan protocolos más elaborados será prudente realizar la programación sin el uso de librerías de configuración externas, de tal forma que se pueda controlar en la totalidad los recursos del dispositivo. Para programas simples es viable el uso de librerías de configuración pues es poco probable que se generen errores.

La gran ventaja de utilizar el lenguaje de programación ANSI-C es que permite realizar la configuración con total control de los registros, teniendo entonces la completa noción de lo que el microcontrolador está procesando.

El cambio en el mundo del control y la automatización mediante el Internet de las Cosas radica en la posibilidad que se tiene de poder conectar cualquier sensor, equipo electrónico, computador, etc. a la Internet, lo que se traduce es poder actualizar o monitorear los parámetros de funcionamiento de los equipos en tiempo real ya sea por parte de un usuario o de un bróker automatizado.

REFERENCIAS

- [1] Valderrama, J. (2018) Trabajo Especial de Grado: Diseño de un conjunto de prácticas para la configuración y uso básico del microcontrolador ESP8266. Universidad Central de Venezuela
- [2] JIFI-Ejemplo-1-ADAFRUIT. (Octubre 2018) <https://github.com/Sirly2522/JIFI-Ejemplo-1-ADAFRUIT>
- [3] JIFI-Ejemplo-2-ANSI-C. (Octubre 2018) <https://github.com/Sirly2522/JIFI-Ejemplo-2-ANSI-C>
- [4] Vongsingthong, S.; Smachat, S. (2014). «[Internet of Things: A review of applications & technologies](#)». Suranaree Journal of Science and Technology.
- [5] Huang, Y., & Li, G. (2010). A Semantic Analysis for Internet of Things. IEEE (2010 International Conference on Intelligent Computation Technology and Automation), 336-339.
- [6] Espressif. (2015). Esp8266 datasheet. (Versión 4.3).
- [7] Espressif (2018a). Esp8266 non-os sdk api reference.(Versión 2.2).
- [8] Espressif (2018b). Esp8266ex datasheet (Versión 5.8).
- [9] Espressif. (2018c). Esp8266 sdk getting started guide. (Versión 3.0)