Práctica: Dispositivo IoT (Telémetro)

Contenido

1 Objetivo de la práctica

El objetivo principal de esta práctica consiste en la realización de un dispositivo IoT comunicado con MQTT y otras API`s específicas de algunas plataformas IoT, así como conocer un firmware abierto, Tasmota, para la programación de dispositivos basados en ESP8266 y ESP32.

2 Material necesario para la realización de la práctica

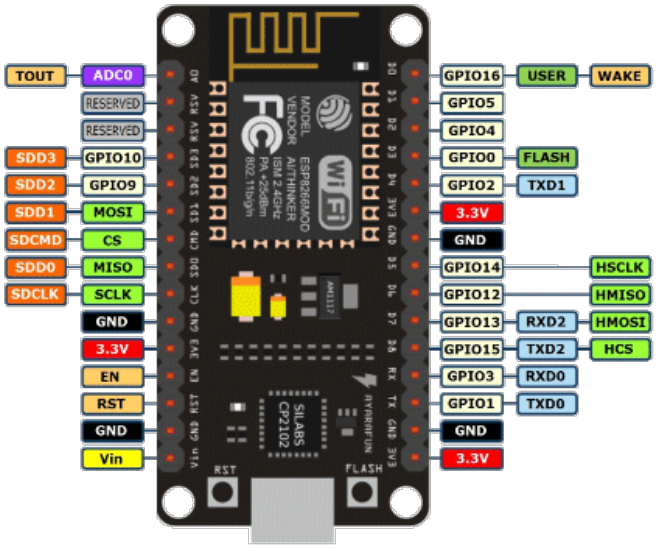
Es necesario disponer de un NodeMCU basado en ESP8266 y/o ESP32, y de algún sensor (telémetro de ultrasonidos, detector de presencia, sensor de temperatura, de humedad, presión, etc.), un cable USB-uUSB para conectarlo al PC, placa de pruebas (protoboard) y cables de conexión.

En este guion se describe la conexión de un telémetro ultrasónico, pero puede usar otros sensores que tenga disponibles así como conectar algún led de salida a través de una resistencia. Busque la información que necesite para ello.

3 La tarjeta ESP

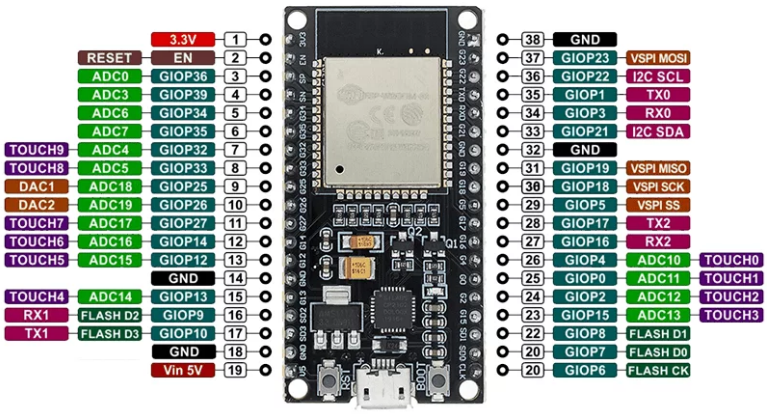
La tarjeta NodeMCU está basada en ESP8266 de Espressif, con chip ESP12E. En la figura puede ver un esquema de los pines de un NodeMCU ESP8266. Hay distintos modelos de tarjeta, el diagrama mostrado puede no coincidir con su tarjeta.

Puede encontrar más información en internet, por ejemplo en https://www.prometec.net/nodemcu-arduino-ide/



Como alternativa podemos usar un módulo basado en ESP32. En la figura se muestra un módulo de 38, pero al igual que para ESP8266, existen numerosos modelos de tarjeta de evaluación.

Hay mucha información en internet, por ejemplo en https://www.prometec.net/esp32/.



Se pueden programar con el IDE de Arduino añadiendo las URL de tarjetas correspondientes.

4 Instalación del IDE de Arduino para NodeMCU

1. Instale el IDE de Arduino desde la página oficial: https://www.arduino.cc/en/Main/Software

   Más información en: https://aprendiendoarduino.wordpress.com/2016/03/30/instalacion-del-ide-arduino/

2. Instalar *pluggin*. En *Archivo>Preferencias*, añadir http://arduino.esp8266.com/stable/package_esp8266com_index.json, https://dl.espressif.com/dl/package_esp32_index.json en "*Gestor de URLs Adicionales de Tarjetas*".

3. En *Herramientas>Placa:…>Gestor de tarjetas* buscar *esp8266* y ESP32 e instalar.

   Más información en https://www.prometec.net/esp8266-pluggin-arduino-ide/

4. En *Herramientas>Placa* seleccione la placa *NodeMCU 1.0 (ESP-12 Module)* o ESP32.

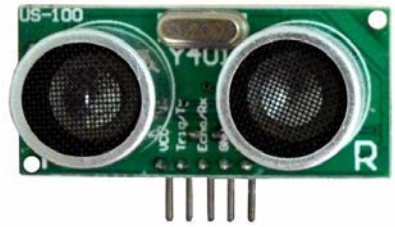
5 El telémetro ultrasónico US-100

El US-100 es similar al más conocido HC-SR04, pero con alguna ventaja:

a. Es más preciso (1mm frente a 3mm del HC-SR04).

b. Admite tensión de alimentación más baja, se puede alimentar entre 2.4V y 5V.

c. Además del método de medida por duración de pulso similar al del HC-SR04, tiene un modo de comunicación serie asíncrona que permite medir también la temperatura. Para usarlo como un HC-SR04 hay que quitar el jumper trasero.



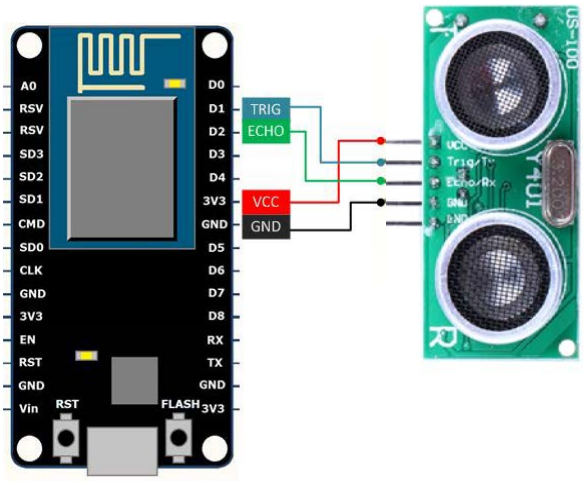
Más información en:

https://minibots.wordpress.com/2013/10/14/comparacion-de-los-sensores-de-ultrasonidos-us-100-y-hc-sr04/

https://minibots.wordpress.com/2014/10/12/control-del-sensor-de-ultrasonidos-us-100-en-modo-serie/

5.1 Conexión entre telémetro y NodeMCU

En la figura se representa la conexión que debe realizarse entre el telémetro el NodeMCU mediante 4 hilos. El telémetro se alimenta con los 3,3V que provee el NodeMCU. Las dos señales, TRIG es la salida del NodeMCU a través de la cual enviamos un pulso de 10uS para iniciar la medida, y ECHO es la entrada por la que se recibe el pulso de respuesta del telémetro cuya duración se corresponde con el tiempo de ida y vuelta de la señal de ultrasonido. No olvide quitar el jumper para que funcione en modo *echo-triger*.



Si utiliza un HC-SR04, aunque algunos funcionan también a 3.3V, posiblemente tenga que

alimentarlo a 5V, En ese caso debe poner una resistencia en serie en la conexión de la señal "echo" vara evitar roturas por la diferencia de tensión con el ESP.

La conexión a un ESP32 es similar, pero debe elegir los pines de entrada/salida adecuados.

5.2 Funcionamiento del telémetro

Puede encontrar información del método de medida en diversos enlaces:

https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/

https://create.arduino.cc/projecthub/josemanu/medir-distancias-con-hc-sr04-63f81e

https://www.prometec.net/sensor-distancia/

6 Ejemplo de programa para medir distancia enviando la medida por el puerto serie

```
// Pines del sensor de ultrasonidos
  #define trigPin 5
  #define echoPin 4
void setup() {
  Serial.begin(9600);
  pinMode ( LED_BUILTIN, OUTPUT );
  pinMode ( trigPin, OUTPUT );
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);
}
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);
//  Medida de la distancia
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  float distancia = 0.0172*pulseIn(echoPin, HIGH);
// Visualización por el puerto serie
  Serial.print("Distancia: ");
  Serial.println(distancia);
}
```

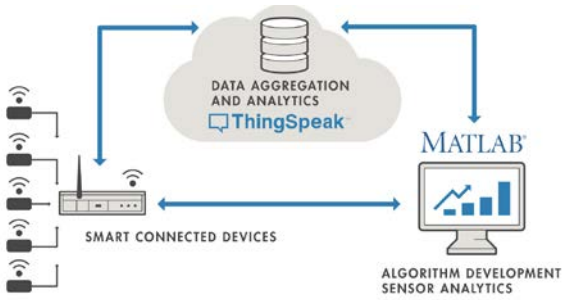
6.1 Para probarlo:

1. Copie el código en un nuevo programa para arduino.
2. Conectce el NodeMCU con el cable USB-uUSB
3. En *Herramientas* seleccione el *Puerto* de comunicación COMn correspondiente
4. Haga click en *Subir*.
5. Desde *Herramientas* abra una ventana de *Monitor Serie*. Ajuste la velocidad de comunicación.

7 Conexión a una plataforma de IoT

Como primer ejemplo vamos a conectar el telémetro a ThingSpeak.

ThingSpeak es una plataforma que permite recoger, almacenar, analizar, visualizar y actuar sobre los datos de sensores o actuadores basados en módulos microcontroladores tales como Arduino ®, Frambuesa Pi ™, BEAGLEBONE Black y otro hardware.



7.1 Registro en ThingSpeak

Regístrese en https://thingspeak.com. Si ya está registrado en https://es.mathworks.com (Matlab) identifíquese en *thingspeak* con su "usuario" y "contraseña" de *mathworks*.

7.2 Crear un canal

El elemento principal de la actividad de *ThingSpeak* es el *canal (channel)*, que contiene campos de datos (*data fields*) y otra información.

Cree un canal con dos campos de datos para el valor de RSSI de la WiFi y para la distancia medida por el telémetro siguiendo las instrucciones de la ayuda:
https://es.mathworks.com/help/thingspeak/collect-data-in-a-new-channel.html

7.3 Recopilar datos

En la pestaña "API Key" del canal creado puede ver las claves que necesitará usar para poder escribir (publicar) y leer (suscribir) los datos con la API de *ThingSpeak*. En "Account > My Profile" puede generar claves para MQTT.

7.4 Ejemplo del programa telémetro con la API de ThingSpeak

1. Con el gestor de librerías del IDE de Arduino instale la librería de ThingSpeak: *Programa > Incluir Librería > Gestionar Librerías*
2. Copie el código en un nuevo programa de arduino
3. Cambie los valores del nombre y clave de la red WiFi por los correspondientes a su punto de acceso WiFi, y los valores del número de canal y clave de escritura de ThingSpeak por los del canal que haya creado.
4. Compile y suba el programa al NodeMCU
5. Visualice los datos desde la web de ThingSpeak

```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
 // Pines del sensor de ultrasonidos
   #define trigPin 5
   #define echoPin 4
```

```
// Punto de acceso wifi al que se conecta el dispositivo
   char ssid[] = "YOUR_SSID";   // your network SSID Name
   char pass[] = "YOUR_PASS";  // your network password
// Parámetros del cloud
   int status = WL_IDLE_STATUS;
   WiFiClient cliente;
   unsigned long myChannelNumber = 123456;   // your ThingSpeak Channel Number
   const char * myWriteAPIKey = "0123456789ABCDEF";  // your ThingSpeak Write API Key
void setup() {
   Serial.begin(9600);
   pinMode ( LED_BUILTIN, OUTPUT );
   pinMode ( trigPin, OUTPUT );
   pinMode(echoPin, INPUT);
   digitalWrite(trigPin, LOW);
//Conexión a la WiFi y al cloud
 WiFi.begin(ssid, pass);
 ThingSpeak.begin(cliente);
 delay(500);
}
void loop() {
   digitalWrite(LED_BUILTIN, HIGH);
   delay(500);
   digitalWrite(LED_BUILTIN, LOW);
//  Medida de la distancia
   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(trigPin, LOW);
   float distancia = 0.0172*pulseIn(echoPin, HIGH);
 // Visualización por el puerto serie
   Serial.print("Distancia: ");
   Serial.println(distancia);
//Medida del rssi de la wifi
   float rssi = WiFi.RSSI();
  //Contrucción del mensaje y envío al cloud
   ThingSpeak.setField(1,rssi);
   ThingSpeak.setField(2,distancia);
   ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
   delay(19500);  //cada 20 segundos
}
```

8 Visualización de datos desde una App móvil

Instale en su *smartphone* la aplicación "*IoT ThingSpeak Monitor Widget*". Coloque el correspondiente *widget*, introduzca número de canal y clave de lectura y pruébelo.

9 Comunicación con MQTT

1. Conecte el telémetro a *ThinSpeak* usando MQTT (buscar información en *ThingSpeak*). Un ejemplo de uplink a ThingSpeak con MQTT puede encontrarlo en: https://www.instructables.com/id/Uploading-Data-to-ThingSpeak-With-MQTT/
2. Conecte el telémetro a otras plataformas IoT mediante MQTT, por ejemplo a Hivemq: https://www.hivemq.com/public-mqtt-broker/
3. Use aplicaciones cliente MQTT para Smartphone y PC para visualizar y controlar los datos del telémetro:
   - Android: por ejemplo “IoT MQTT Panel”, “IoT MQTT Dashboard”, “MQTT Dash”, etc.
   - PC: MQTTBox (http://workswithweb.com/mqttbox.html)

10 Conexión a otras plataformas IoT

Explore la conexión a otras plataformas IoT. Ejemplos:

- Carriot, Adafruit.IO, Cayenne, Thinger.io, Node-RED, … (http://pdacontroles.com)
- Cayenne (https://www.hackster.io/suhail_jr/glo-iot-smart-light-b3fb5d)
- Thinger.io (https://www.hackster.io/detox/mqtt-with-the-esp8266-on-the-thethingsio-1ba119)
- IFTTT (https://ifttt.com/applets/388514p-send-email-with-the-esp8266, https://programarfacil.com/blog/arduino-blog/como-ifttt-arduino-nodemcu/, https://www.hackster.io/matlab-iot/smart-humidity-sensor-thingspeak-matlab-andifttt-1a8495)
- AWS IoT (https://aws.amazon.com/es/iot/)
- IBM Watson IoT (https://www.ibm.com/es-es/marketplace/internet-of-things-cloud)
- Microsoft Azure IoT (https://docs.microsoft.com/es-es/azure/iot-hub/, https://github.com/Azure/azure-iot-arduino#simple-sample-instructions)
- Kaa (https://www.kaaproject.org/overview/)
- Blynk (https://blynk.io/)

11 Firmware Tasmota

Programe el mismo u otro dispositivo con un firmware *Tasmota*: https://github.com/arendst/Tasmota.

11.1 Instalación de Tasmota

11.1.1 Instalación de Tasmota en el NodeMCU ESP8266.

- Tasmota-PyFlasher-1.0: https://github.com/tasmota/tasmota-pyflasher/releases
- tasmota-ES.bin https://github.com/arendst/Tasmota/releases
- Cargar el firmware “tasmota-ES.bin” con el programa Tasmota-PyFlasher.
- Conectarse a la wifi del Nodemcu (tasmota).
- Configurar el punto de acceso wifi al que se conectará para acceder a la red. Al terminar el tasmota se conecta a esa red.

11.1.2 Instalación de Tasmota para ESP32.

Los programas mencionados para el *flasheo* de Tasmota en el ESP8266 aún no son compatibles con el ESP32. El procedimiento para ESP32 está descrito en https://www.youtube.com/watch?v=Dz2dc-_HR5M:

- Es necesario descargar la aplicación oficial desde la página web de Espressif: https://www.espressif.com/en/support/download/other-tools
- Es necesario instalar cuatro archivos que puede descargar desde https://github.com/techiesms/TASMOTA-on-ESP32. Puede sustituir el binario de Tasmota por su versión en español u otro idioma.
- Además, con esto no se genera un punto de acceso WiFi, sino que debemos proporciónale a través de una terminal serie, por ejemplo Termite, los datos de acceso a nuestra red WiFi. Tras esto, a nuestro dispositivo se le asignará una IP que debemos introducir en el navegador para acceder a la interfaz web de configuración del dispositivo tasmota.

11.2 Configuración de tasmota

- Acceda a la web del tasmota a través del navegador introduciendo su dirección IP.
- Para el NodeMCU ESP8266, en “Configuración” se selecciona el tipo de módulo “Generic(18)” y se guarda.
- En “Configuración del Módulo” se asocian los pines de los sensores y actuadores que hayamos conectado.
- Configure MQTT para conectarlo a su broker.
- Si tiene salidas puede configurar temporizadores.
- Puede realizar otras configuraciones introduciendo comando desde la consola.