Sistemas Embebidos

Práctica 3: Conectividad en dispositivos embebidos

Objetivos

- Prueba de tecnologías de comunicación del dispositivo embebido.
- Instalación y configuración de dispositivo en conexión con una plataforma en la Nube.
- Adquirir conocimientos y habilidades de envío y recepción de mensajes con el protocolo MQTT.

Descripción

El MCU **Arduino IoT** es un dispositivo sencillo para construir aplicaciones de internet de las cosas. Con este MCU ya hemos realizado nuestro primer dispositivo embebido con el que obtenemos los datos de un sensor. El objetivo de esta práctica es utilizar la capacidad de conectividad de esté MCU para añadir conectividad a nuestro dispositivo.

Para ello haremos uso de dos formas de conectividad distintas: BLE y WiFi.

Para la conectividad WiFi haremos uso de **Ubidots stem.** Se trata de una plataforma de aplicación para el despliegue de aplicaciones IoT en la nube: https://ubidots.com/stem

Para la conectividad BLE haremos uso de una aplicación móvil con la que se pueden visualizar y enviar mensajes Bluetooth.

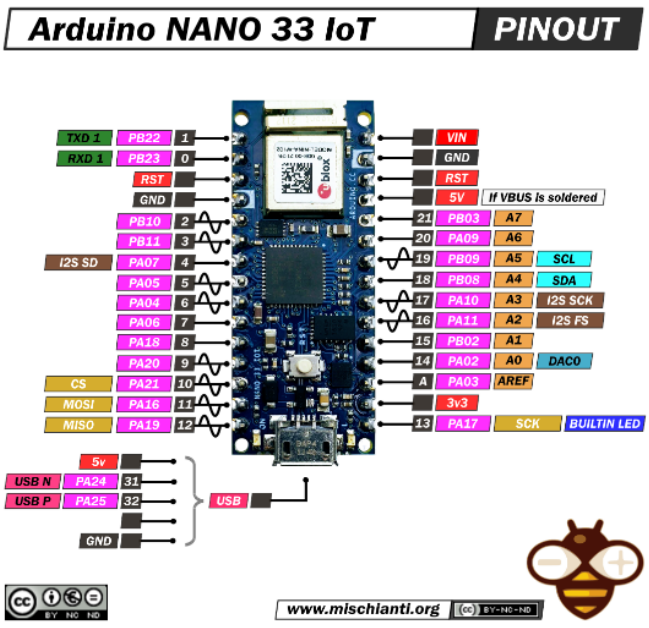
Trabajo a realizar:

1. Configurar Arduino IoT cloud
    a. Crea una cuenta en el portal de Ubidots stem
    b. Conecta tu dispositivo Arduino 33 IoT a la plataforma Cloud.
        - Crea un dispositivo.
        - Crea una variable denominada “pulsaciones” de tipo entero, con permiso “read & write” y Actualización “On charge”.
        - Crea un dispositivo y guarda las credenciales que te ofrece la plataforma.
        - Establece la conexión “Network” a la Wifi de tu dispositivo móvil o Wifi de casa. En la Universidad es probable que necesites generar una red wifi para poder conectar el dispositivo.
        - Crea un cuadro de mando “Dashboard” que muestra en un panel el valor discreto de la variable “pulsaciones”.
        - Crea un código para el dispositivo que se conecte a la red wifi, utiliza la librería arduino-libraries/WiFiNINA, y usando las credenciales facilitadas por el portal envíe un entero aleatorio entre 60 y 120 que representa la media de las pulsaciones. Haz que esté mensaje se envíe cada 5 segundos con un valor distinto. Para esté ejemplo utiliza la documentación de API REST facilitada por la plataforma. https://docs.ubidots.com/v1.6/reference/welcome
    c. Observa los cambios en el valor de la variable en el “Dashboard” a lo largo del tiempo. Añade un control de tipo “chart” en el “Dashboard” para observar de forma gráfica esta evolución.

2. Enviar y recibir datos por MQTT.
    a. Crea un botón en el portal de la plataforma y añade el comportamiento para cambiar el valor numérico cuando se pulsa el botón, esto hará que se envíe un mensaje al topic de MQTT correspondiente.
    b. Crea un código para el dispositivo que una vez conectada a la red wifi se conecte al servidor de MQTT de ubidots y haz que se suscriba al topic correspondiente para esperar los mensajes del servidor.

3. Prueba de la función BLE
    a. Carga a tu entorno la biblioteca “ArduinoBLE”.
    b. **Modo emisor**: Carga en el entorno el ejemplo “ArduinoBLE/Peripheral/BatteryMonitor”. Conecta el teléfono móvil con el dispositivo mediante Bluetooth y observa la información suministrada. Recuerda cambiar el valor que se obtiene por uno fijo establecido por ti, ya que no tienes ninguna batería conectada.
    c. **Modo monitor**: Carga en el entorno el ejemplo “ArduinoBLE/Central/Scan”. Observa en la consola los dispositivos que se detectan y la información obtenida de ellos. Observa la frecuencia con la que se detectan.

4. Reemplaza en tu prototipo el Arduino UNO usado en la **práctica 2** por el dispositivo Arduino nano IoT y verifica que las conexiones son correctas teniendo en cuenta el pinout del nuevo dispositivo.
    a. Modifica el código utilizado para la práctica 2 para que se conecte a una red wifi. Realiza también las modificaciones necesarias para que el dispositivo envíe cada 5 segundos los datos del sensor a la plataforma mediante MQTT.
    b. Modifica el código y añade la conectividad BLE para que una vez se conecte por BLE un dispositivo al arduino se le envie el valor del sensor cada 5 segundos.
    c. Modifica el código para que cuando llegue un mensaje por MQTT el dispositivo quede suspendido, esto quiere decir que no realice ninguna medición y que muestre por pantalla la palabra suspendido. Hasta que se reinicie o llegue otro mensaje.



Ponte en contacto con el profesor de la asignatura si no dispones de dispositivo Arduino IoT.

Aclaraciones para la práctica:

A la hora de integrar todos los componentes, en la anterior práctica utilizamos el pin de 5v de Arduino para alimentar los sensores y pantalla. Cuando tengas que hacer esté paso para el Arduino IoT utiliza el pin de 3.3v, ya que en la placa el pin de 5v está desactivado por defecto y hay que activarlo con una soldadura que no hemos realizado.

Si en algún momento no llegas a leer los mensajes del serial que ocurren durante el setup() añade el siguiente código después de Serial.begin(9600); while(!Serial); de esta forma el programa esperará a que Serial esté disponible para ser usado antes de continuar.

Normas de entrega:

- La realización del trabajo es individual.
- El documento debe seguir el formato definido para las publicaciones de *Lecture Notes in Computer Science* de *Springer* más una portada e índice en la primera y segunda páginas: https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines
- La fecha límite de entrega es el **6 de abril de 2023**.
- La entrega se realizará a través de la herramienta de entrega de trabajos de Campus Virtual.
- Los formatos válidos del documento son *MS Word* (.doc, .docx), *OpenDocument* (.odt) o *Portable Document Format* (.pdf).