

# Universidad de la Sabana

## Automatización y Control de Procesos

Workshop 6  
Juan David Baldion 0000191147  
Hadden Carolina Diaz 0000156276

May 6, 2022

### 1 Introducción

Teniendo en cuenta las indicaciones brindadas en el taller número 6, se dispone a la solución de una práctica con tecnología 4.0 IoT (Internet de las cosas), haciendo uso del proveedor de servicios en la nube como Microsoft Azure y un simulador Raspberry Pi. Asimismo, se llevaron a cabo una serie de pasos para la creación de una cuenta estudiantil en Azure. Lo anterior, con el objetivo de entender la implementación básica de un sistema IoT conectado a un servicio en la nube.

### 2 Modelo Basico de IoT

Apartir de la imagen dada se identificaron los diferentes componentes de la solución IoT propuesta los cuales serán explicados detalladamente a continuación.

#### 2.1 Sensor

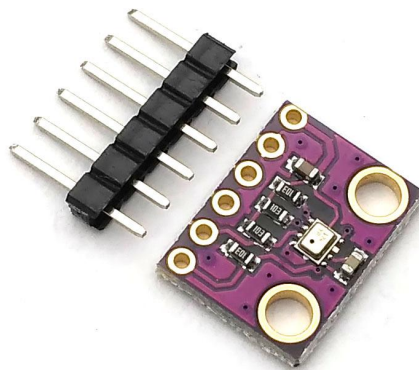


Figure 1: Sensor BM280.

El sensor BM280 se utiliza para la medición de la presión atmosférica, temperatura y humedad, en el sistema propuesto la comunica con el sistema integrado se hace por medio de la interfaz I2C. A continuación se muestran las especificaciones del sensor BM280[1]

### 2.1.1 Especificaciones

- Voltaje de Operación: 1.8V – 3.3V DC
- Interfaz de comunicación: I2C o SPI (3.3V)
- Rango de Presión: 300 a 1100 hPa
- Resolución: 0.16 Pa
- Precisión absoluta: 1 hPa
- Rango de Temperatura: -40°C a 85°C
- Resolución de temperatura: 0.01°C
- Precisión Temperatura: 1°C
- Rango de Humedad Relativa: 0-100% RH
- Precisión de HR: +-3%
- Ultra-bajo consumo de energía
- Completamente calibrado
- Frecuencia de Muestreo: 157 Hz (máx.)

## 2.2 Sistema Integrado



Figure 2: Raspberry PI 3 Model B v1.2.

Para el sistema integrado de la solución IoT propuesta se hizo uso de un Raspberry PI 3 Model B v1.2 el cual es una computadora compacta y de bajo costo la cual es ideal para el prototipado de soluciones como la propuesta ya que permite el uso de sistemas operativos como Linux además de la conexión sencilla de componentes como sensores. A continuación se muestra las especificaciones del Raspberry PI 3 Model B v1.2[2]

### 2.2.1 Especificaciones

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO

- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

## 2.3 Conectividad

En el caso de la conectividad el sistema propuesto hace uso del protocolo de comunicacion MQTT (MQ Telemetry Transport) el cual esta basado en TCP/IP. Este protocolo funciona por medio del patron publicador/suscriptor donde el publicador es el sistema IoT (Sensores) y el suscriptor es el servicio que se utilice para el analisis de datos (Servicio en la Nube).

## 2.4 Análisis de datos

Para el analisis de datos se hizo uso del proveedor de servicios en la nube Microsoft Azure mas especificamente Iot HUB el cual nos permite el despliegue sencillo de servicios para la supervision de sistemas IoT.

# 3 Implementación del Simulador en Raspberry Pi

En primer lugar, se creó una cuenta estudiantil en Microsoft Azure, se creó un nuevo recurso llamado "Internet de las cosas" y se escogió el apartado "Centro de IoT".

En segundo lugar, se desplegó una ventana de configuración del hub y se creó un grupo de recursos "Auto". Así pues, se dio inicio al proceso de nombrar el hub como "AutoWorkshop6". Por último, se dio lugar al hub en East US.

The screenshot shows the Microsoft Azure portal interface for configuring an IoT Hub. The top navigation bar includes the Microsoft Azure logo and a search bar. The breadcrumb trail indicates the path: Inicio > Centro de IoT >. The main heading is 'Centro de IoT' with a Microsoft logo. Below this, there are tabs for 'Aspectos básicos', 'Redes', 'Administración', 'Etiquetas', and 'Revisar y crear'. The 'Aspectos básicos' tab is active. A description states: 'Cree un centro de IoT que le ayude a conectar, supervisar y administrar miles de millones de recursos de IoT. Más información'. Under 'Detalles del proyecto', there are two dropdown menus: 'Suscripción' (set to 'Azure para estudiantes') and 'Grupo de recursos' (set to 'Auto'). A 'Crear nuevo' link is present. Under 'Detalles de instancia', there are two more dropdown menus: 'Nombre de IoT Hub' (set to 'AutoWorkshop6') and 'Región' (set to 'East US').

Figure 3: Centro de IoT.

En tercer lugar, se seleccionó la pestaña de administración de redes, la cual permite la separación de precios y dispositivos conectados al hub. Lo anterior con finalidad de mitigación de riesgos en cobros de Azure.

Al crear el hub, conectamos Raspberry Pi al mismo. Pues al realizar esta conexión, se buscó un apartado llamado "dispositivos IoT" en el hub, el cual crea un nuevo dispositivo "RaspbWorkshop6" y se generaron las llaves automáticamente.

Microsoft Azure

Inicio > workshop6Auto >

### Crear un dispositivo

Buscar dispositivos Azure Certified for IoT en el catálogo de dispositivos

Id. de dispositivo \* RaspbWorkshop6

Tipo de autenticación  
☒ Clave simétrica ☐ X.509 autofirmado ☐ X.509 firmado por CA

Generar claves automáticamente ☒

Conectar este dispositivo a un centro de IoT  
☒ Habilitar ☐ Deshabilitar

Dispositivo principal  
**No hay ningún dispositivo primario**  
[Establecer un dispositivo primario](#)

Figure 4: Creación de un nuevo dispositivo.

En cuarto lugar, se selecciona el dispositivo de la lista y se generan las llaves de conexión. De estas llaves, se escogió la llave de cadena de conexión principal y esta se adiciona exactamente en la línea 15 del código del simulador.

Microsoft Azure

Inicio > workshop6Auto >

### raspb1

workshop6Auto

Guardar Mensaje al dispositivo Método directo Agregar identidad de módulo Dispositivo gemelo Administrar claves Actualizar

Id. de dispositivo raspb1

Clave principal

Clave secundaria

Cadena de conexión principal

Cadena de conexión secundaria

Habilitar la conexión a IoT Hub ☒ Habilitar ☐ Deshabilitar

Dispositivo principal  
**No hay ningún dispositivo primario**  
[Establecer un dispositivo primario](#)

Identidades de módulo Configuraciones

Identificador del módulo	Estado de conexión	Última actualización del estado	Hora de la última actividad (UTC)
--------------------------	--------------------	---------------------------------	-----------------------------------

Figure 5: Dispositivo.

```

1 /*
2  * IoT Hub Raspberry Pi NodeJS - Microsoft Sample Code - Copyright (c) 2017 - Licensed MIT
3  */
4 const wpi = require('wiring-pi');
5 const client = require('azure-iot-device').Client;
6 const Message = require('azure-iot-device').Message;
7 const Buzzer = require('azure-iot-device-mqtt').Mqtt;
8 const BME280 = require('i2c-bus-sensor');
9
10 const BME280_OPTION = {
11   i2cBusNo: 1, // defaults to 1
12   i2cAddress: BME280.BME280_DEFAULT_I2C_ADDRESS() // defaults to 0x77
13 };
14
15 const connectionString = "HostName=workshop8454.azure-devices.net;DeviceId=raspbpi;SharedAccessKey=lsk7mwhdTFR0dIBogI7W0Yc99460";
16 const LEDPin = 4;
17
18 var sendingMessage = false;
19 var messageId = 0;
20 var client;
21 var blinkLEDTimeout = null;
22
23 function getMessage(cb) {
24   messageId++;
25   sensor.readSensorData()
26     .then(function (data) {
27       console.log('Received data:');
28       console.log('messageId: ' + messageId);
29       console.log('deviceId: ' + 'Raspberry Pi web Client');
30       console.log('temperature: ' + data.temperature_C);
31       console.log('humidity: ' + data.humidity);
32       console.log('data.temperature_C: ' + data.temperature_C);
33     })
34     .catch(function (err) {
35       console.error('Failed to read out sensor data: ' + err);
36     });
37 }
38
39 function sendMessage() {
40   if (!sendingMessage) { return; }
41
42   getMessage(function (content, temperatureAlert) {
43     var message = new Message(content);
44     message.properties.add('temperatureAlert', temperatureAlert.toString());
45     console.log('Sending message: ' + content);
46     client.sendEvent(message, function (err) {
47       if (err) {
48         console.error('Failed to send message to Azure IoT Hub');
49       } else {
50         console.log('Message sent to Azure IoT Hub');
51       }
52     });
53   });
54 }
55
56 function onStart(request, response) {
57   console.log('Try to invoke method start(' + request.payload + ')');
58   sendingMessage = true;
59   response.send(200, 'Successfully start sending message to cloud', function (err) {
60
61
62

```

Figure 6: Código del simulador.

## 4 Referencias

- [1] Sensor de Presión, Temperatura y Humedad – BME280. (2018). Electronilab. [Electronilab.co](http://Electronilab.co).
- [2] Raspberry Pi Ltd. (2016). Raspberry pi 3 model B. Raspberry Pi. [raspberrypi.com](http://raspberrypi.com)