

GESTIÓN DE LOS HUMOS DE UN VEHÍCULO A MOTOR A TRAVÉS DE SIGFOX

DISPOSITIVOS Y REDES INALÁMBRICAS

ÁNGEL ORTEGA ALFARO
ÁLVARO LEÓN CEBRIÁN

Contenido

INTRODUCCIÓN	2
DESARROLLO DEL PROTOTIPO	2
CONCLUSIONES	6
BIBLIOGRAFÍA	6
Ilustración 1: Creación del centro de IoT	2
Ilustración 2: Escala del centro IoT.....	3
Ilustración 3: Centro IoT.....	3
Ilustración 4: Centro IoT.....	3
Ilustración 5: Callback de sigfox	4
Ilustración 6: Azure function	4
Ilustración 7: Resultado función.....	5
Ilustración 8: Código arduino	5

INTRODUCCIÓN

Debido a los niveles de contaminación actuales en el mundo, se ha propuesto medir en directo los niveles de contaminación que emiten los vehículos, por tanto, se ha implantado una placa de arduino que se encarga de controlar el estado del vehículo. En concreto, se encarga de medir el nivel de humos del motor, y en el caso de que se exceda del nivel establecido por el organismo competente, mandaría un mensaje a la empresa fabricante del vehículo haciendo conocer el problema, de tal forma que la empresa pueda avisar al cliente del estado de su vehículo.

La placa de arduino mide distintos parámetros del coche, que los envía a la empresa y al cliente simplemente le avisa de que hay un problema. El sensor estaría situado en la zona del catalizador, cerca del escape. Si detecta un error, mandaría información acerca del proceso de combustión del motor.

Toda esta información sería enviada a través de la red de SigFox, donde se tiene cobertura en todo el territorio español, y, por lo tanto, no tendremos problemas de conectividad.

Por otro lado, los datos serán tratados en la plataforma Azure de Microsoft, en donde se avisará al responsable desde el primer momento en el que haya problemas, de tal forma que se reducirá el tiempo en el que el coche emita más gases contaminantes de los que debería.

DESARROLLO DEL PROTOTIPO

El primer paso para desarrollar nuestro prototipo será registrarse en la plataforma Azure de Microsoft y seleccionar la plataforma Azure IoT, que será el sitio al que mandemos la información obtenida del coche.

Dentro de la plataforma IoT, deberemos de crear un centro de IoT, que nos servirá para recoger la información enviada por el Callback de SigFox.

Centro de IoT
Microsoft

[Aspectos básicos](#) [Escala y tamaño](#) [Revisar y crear](#)

Cree una instancia de IoT Hub que le ayude a conectar, supervisar y administrar miles de millones de recursos de IoT. [Obtener más información](#)

DETALLES DEL PROYECTO

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

* Suscripción ⓘ

* Grupo de recursos ⓘ
[Crear nuevo](#)

* Región ⓘ

* Nombre de la instancia de IoT Hub ⓘ ✓

Ilustración 1: Creación del centro de IoT

Puesto que es un prototipo, usaremos la escala mínima para la creación del centro.

Centro de IoT

Microsoft

[Aspectos básicos](#) [Escala y tamaño](#) [Revisar y crear](#)

Cada instancia de IoT Hub se aprovisiona con un determinado número de unidades en un nivel específico. El nivel y el número de unidades determinan la cuota máxima diaria de mensajes que puede enviar. [Más información](#)

UNIDADES Y NIVEL DE ESCALA

* Nivel de precios y de escala ⓘ S1: Nivel estándar ▼
[Más información sobre cómo elegir el nivel de IoT Hub adecuado para su solución](#)

Número de unidades de IoT Hub de S1 ⓘ 1
Esto determina la funcionalidad de escalabilidad de IoT Hub y se puede cambiar a medida que aumente la necesidad.

Ilustración 2: Escala del centro IoT

Así se quedaría el centro IoT implementado.

✓ Se completó la implementación

[Ir al recurso](#)



Nombre de implementación: Microsoft.IoTHub-620182850
Suscripción: [Evaluación gratuita](#)
Grupo de recursos: [recursos-sigfox](#)

DETALLES DE IMPLEMENTACIÓN [\(Descargar\)](#)

Hora de inicio: 20/6/2019 18:28:53
Duración: 3 minutos 6 segundos
Id. de correlación: a36f1482-9429-4b5f-8641-3c6fb6a4a211

RECURSO	TIPO	ESTADO	DETALLES DE LA OPERACI...
✓ instancia-sigfox	Microsoft.Devices/IoT...	OK	Detalles de la operación

Ilustración 4: Centro IoT

instancia-sigfox - Directivas de acceso compartido

Buscar (Ctrl+/)

+ Agregar

IoT Hub usa permisos para conceder acceso a cada punto de conexión de IoT Hub. Los permisos limitan el acceso.

Buscar para filtrar elementos...

DIRECTIVA	PERMISOS
iothubowner	escribir en el registro, conexión d
service	conexión del servicio
device	conexión del dispositivo
registryRead	leer registro
registryReadWrite	escribir en el registro

Permisos

- ☒ Leer Registro ⓘ
- ☒ Escribir en el Registro ⓘ
- ☒ Conexión del servicio ⓘ
- ☒ Conexión del dispositivo ⓘ

Claves de acceso compartidas

Clave principal ⓘ

Clave secundaria ⓘ

Cadena de conexión (clave principal) ⓘ

Cadena de conexión (clave secundaria) ⓘ

Ilustración 3: Centro IoT

Una vez dentro del centro IoT, son muy importantes los datos que aparecen en el apartado “Claves de acceso compartidas”, ya que son las que permiten la conexión con los dispositivos IoT. Por ello, copiaremos la cadena de conexión (Clave principal) para usarla más tarde en el callback de sigfox.

The screenshot shows the 'Callbacks' configuration page. It has several input fields with dropdown menus and a text area. The 'Type' field has 'DATA' and 'UPLINK' options. The 'Custom payload config' field contains 'humos::char:2'. The 'Connection string' field contains 'Endpoint=instancia-sigfox.azure-devices.net;SharedAccessKeyName=iothubowne'. The 'Url' field contains 'https://www.instancia-sigfox.azure-devices.net'. The 'JSON body' field contains a JSON object with the following structure: { "device" : "{device}", "data" : "{data}", "humos" : "{customData#humos}", "time" : {time}, "duplicate" : {duplicate}, "seqNumber" : {seqNumber} }.

Ilustración 5: Callback de sigfox

Dentro del callback, tenemos que indicar la cadena de conexión anteriormente nombrada, y tenemos que indicar el JSON que mandaremos a Azure. Como se puede comprobar en la ilustración 5, enviaremos un dato de tipo char que indicará el nivel de humos en el coche.

A continuación, deberemos de crear una azure function app la cual será la encargada de mostrar la información en una pagina web:

```
1 #r "Newtonsoft.Json"
2
3 using System.Net;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Extensions.Primitives;
6 using Newtonsoft.Json;
7
8 public static async Task<IActionResult> Run(HttpRequest req, ILogger log)
9 {
10     log.LogInformation("C# HTTP trigger function processed a request.");
11
12     string name = req.Query["name"];
13
14     string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
15     dynamic data = JsonConvert.DeserializeObject(requestBody);
16     name = name ?? data?.name;
17
18     //name recibiría el callback del backend a través de IoT hub y mostraría la
19     //información obtenida
20     return name != null
21         ? (ActionResult)new OkObjectResult($"Hello, {name}")
22         : new BadRequestObjectResult("Se ha superado el nivel de humos permitido");
23 }
24
```

Ilustración 6: Azure function

El resultado de la función sería el siguiente:

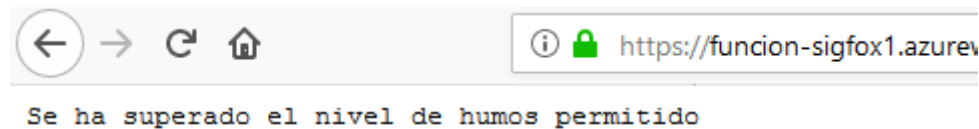


Ilustración 7: Resultado función

Por último, deberemos de escribir el código que será cargado en la placa de Arduino:

```
#include <SigFox.h>

void setup() {
  Serial.begin(9600);
  SigFox.begin();
  SigFox.debug();

  //El bucle iniciará mientras el coche esté funcionando

  float cocheArrancado = 1;

  //Declaramos una variable que servirá para comprobar
  //en la estructura de control el nivel de humo, ya
  //que en el sistema real este valor vendría dado por
  //un detector de humo que actualiza la variable
  //de forma constante.

  //Dicho detector mide el porcentaje de humos, suponemos
  //que mas de un 40 tendremos problemas
  float humos = 66.6;

  while(cocheArrancado == 1){
    if(humos>=40){
      Serial.println("Se ha superado el nivel de humos permitido");
    } else {
      //No manda aviso ya que no se ha excedido el nivel de humos
      Serial.println(" ");
    }
  }

  SigFox.beginPacket();
  SigFox.write(humos);
  SigFox.endPacket(true);
}

void loop() {
  while (SigFox.available()) {
    Serial.print("0x");
    Serial.println(SigFox.read(), HEX);
  }
}
```

Ilustración 8: Código arduino

CONCLUSIONES

El hecho de monitorizar constantemente un vehículo nos puede permitir conocer fallos que estén ocurriendo y arreglarlos antes de que puedan causar un accidente. Y no solo la cantidad de contaminación de este caso, si monitorizamos más partes de los vehículos, gracias a SigFox podremos obtener una conducción más eficiente.

Respecto a Microsoft Azure, queda claro que es una herramienta que facilita mucho la implementación de aplicaciones IoT, de tal forma que se nos permita a los usuarios automatizar procesos que sin estas tecnologías no sería posible.

BIBLIOGRAFÍA

[1] Azure IoT – Última vez accedido el 24/06/2019

<https://docs.microsoft.com/es-es/azure/iot-central/>

[2] Azure functions – Última vez accedido el 24/06/2019

<https://docs.microsoft.com/es-es/azure/azure-functions/>