



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

Escuela Politécnica Superior de Alcoi – Universidad Politécnica de Valencia

Sistema portátil autónomo para la adquisición de datos conectado a la nube

Grado en Ingeniería Informática

Convocatoria: Septiembre 2017

Tutor: D. Rubén García Llorens
Cotutor Externo: D. Jorge Mayor Gisbert

Alumno: Víctor Sánchez Ribes

Resumen

Este proyecto consiste en la elaboración de un sistema autónomo de adquisición de datos, en este caso el pH de un medio líquido para controlar, cuantificar y crear un histórico de dicha variable.

Para la realización de este trabajo, se ha utilizado un microcontrolador y una sonda para la medición de la variable en cuestión.

La información que recibe el microcontrolador se manipula y se envía a la nube.

Como microcontrolador se utilizará una Raspberry Pi 2 modelo B y como sonda para la medición del pH, cualquier sonda que sea compatible con Arduino.

Abstract

This project consists of the development of an autonomous data acquisition system, in this case the pH of a liquid medium to control, quantify and create a history table of those variable.

For the development of this work, a microcontroller and a sensor have been used for the measurement of the variable in question.

The information received by the microcontroller is manipulated and sent to the cloud.

As microcontroller will use a Raspberry Pi 2 model B and as sensor for pH measurement, any sensor that is compatible with Arduino.

Índice

Introducción.....	5
Fundamentos Jurídicos.....	5
pH.....	6
pH en piscinas	6
Objetivos	7
A corto plazo.....	7
A largo plazo.....	7
Material necesario.....	9
Microcontrolador	9
Especificaciones	9
Características	10
Sonda pH.....	11
Especificaciones	11
MCP3008.....	12
Adaptador Wi-Fi TP-Link TL-WN725N.....	12
Diseño del proyecto.....	14
Diagrama de diseño del proyecto	14
Diagrama de flujo del proyecto	15
Ensamblaje del proyecto	16
Bus SPI.....	16
Ventajas	16
Inconvenientes.....	16
Bus I2C.....	17
Ensamblado del hardware	18
Cableado Raspberry – MCP3008	18
Cableado MCP3008 – Sonda.....	20
Programación del sistema.....	22
Raspberry Pi 2.....	22
Obtención de pH.....	22
Transmisión a Azure.....	26
Configuración de Azure	27
Creación de un IOT HUB	27
Creación de Stream Analytics Job	28

Configuración de Power Bi	30
Visualización de los datos	30
Creación de Alertas.....	32
Configuración de Microsoft Flow.....	34
Integrar “como servicio” el proyecto.....	35
Conclusiones.....	36
Informe Power Bi	36
Posibles Mejoras	39
Bibliografía	40
Anexo	42
Raspberry Pi 2 Model B (schematics).....	42
pH Meter v1.0	43
MCP3008.....	44
Adaptador Wifi TP-Link TL-WN725N	45
Ensamblado Completo	46
Diagrama de funcionamiento	47
Código fuente	48
Azure IOT HUB.....	51
Azure Stream Analytics job.....	51
Stream Analytics Job	52
Input.....	52
Consulta (Query).....	52
Output	53
Alerta Microsoft Flow en un entorno real	54
Tipo de alerta desencadenada.....	54
Destinatario alertado e información enviada	55

Introducción

Fundamentos Jurídicos

Según el Decreto 255/1994 de 7 de diciembre de la Generalitat Valenciana, por el que se regulan las “normas higiénico-sanitarias de las piscinas de uso colectivo y de los parques acuáticos” y el Decreto 143/2015, de 11 de septiembre, que se referencia las “normas de seguridad de las piscinas de uso colectivo y parques acuáticos”, se dispone que:

“CAPITULO II

Características, tratamiento y vigilancia del agua

Artículo octavo

- 1. La calidad del agua de los vasos se refiere a unas condiciones y cualidades analíticas mínimas que la hagan adecuada para la inmersión de los usuarios.*

Artículo noveno

El agua de los vasos deberá ser filtrada y desinfectada; no será irritante para la piel, ojos o mucosas y deberá cumplir en todo momento los parámetros especificados en el anexo I.”

También se añade en un capítulo posterior que:

“El agua recirculada será sometida a un adecuado tratamiento fisicoquímico, utilizando al efecto un sistema de depuración que deberá encontrarse en funcionamiento continuo durante el tiempo suficiente para mantener las condiciones de idoneidad del agua exigidas por esta legislación, y, en cualquier caso, durante el tiempo en que la actividad permanezca abierta a los usuarios.”

Por lo que los parámetros fisicoquímicos autorizados son los detallados a continuación:

“ANEXO I

REQUISITOS DE CALIDAD DEL AGUA DEL VASO.

Parámetros Valor límite

PARAMETROS FISICOQUIMICOS

pH 7,0 - 8,2”

Debido a lo dispuesto en estos documentos y la necesidad de llevar a cabo estas mediciones de pH de una forma manual. Añadiendo la posibilidad de que, por un precio razonable y las tecnologías existentes actualmente para la automatización de esta tarea y el almacenamiento, a modo de histórico, de dichos valores se vislumbra una necesidad de investigación y para ello ha nacido este proyecto.

pH

“El pH es una medida de acidez o alcalinidad de una disolución. El pH indica la concentración de iones hidronio [H₃O⁺] presentes en determinadas sustancias.”

Esta medida tiene un rango de valores de 0 - 14, por cuyo resultado se determina que si el pH es menor a 7 se denomina pH ácido, igual a 7 es un pH llamado neutro y si el pH es mayor a 7 se denomina pH básico.

ACIDO	0	HCl
	0,5	Ácido de batería
	1	Ácido sulfúrico
	2	Jugo gástrico
	2,4	Jugo de limón
	2,9	Vinagre
	3	Jugo de naranja
	5	Café
	5,6	Lluvia ácida
	5,5 - 6,5	Orina
	6,5	Leche
NEUTRO	7	Agua pura
	6,5 - 7,4	Saliva humana
	7,35 - 7,45	Sangre
	8	Agua de mar
	9 - 10	Jabón de manos
	11,5	Amoniaco
	12,5	Lejía
	13,5	Limpia tuberías
BASICO	14	NaOH

pH en piscinas

El pH en las piscinas está directamente relacionado con la calidad del agua de una piscina ya que el cloro (Cl) que se añade para la desinfección, tanto del agua que contiene como de las paredes de la misma, solamente es efectivo si el pH de la piscina se encuentra entre 6,5 y 8. Si el pH es menor a 6,5 o mayor de 8, la adición de cloro a la piscina no servirá para nada. Debido a este problema es interesante tener una monitorización del pH constante puesto que si este aumenta de 8 o es inferior a 6,5 puede producir agua turbia, desaprovechamiento de productos químicos, irritación en la dermis, así como de picor en los ojos o problemas respiratorios.

Objetivos

A corto plazo

Debido a lo expuesto anteriormente en la Introducción, la necesidad de la automatización de la tarea de medición de pH se llevará a cabo mediante un microcontrolador, una sonda para la medición de dicha variable y una conexión internet vía Wifi para la subida de los datos a la nube.

Con una mayor concreción, este proyecto utiliza un microcontrolador, que debido a su razonable precio y su capacidad de soportar múltiples sistemas operativos lo hace idóneo para este fin, llamado Raspberry Pi 2 modelo B con el que tenemos como objetivos:

- Conexión y puesta en funcionamiento de la Raspberry Pi 2 mediante un sistema operativo.
- Puesto que la Raspberry no dispone de entradas analógicas y se tiene una sonda que utiliza este tipo de salida de datos, se conectará un dispositivo que permita adquirir un valor digital y se configurará el entorno de desarrollo, así como la Raspberry para que sea posible la utilización de librerías que hagan la tarea más fácil.
- Transferencia del dato que interesa a un entorno Cloud.
- Utilización de diferentes herramientas para el tratamiento de la variable interesante.

A largo plazo

Vista a un futuro, este proyecto tiene como finalidad la simplificación de la tarea de medición del pH en lugares allí donde se necesite tanto un histórico de valores y una medición con una escala de horas o días, así como una medición con un intervalo de tiempo más reducido como un intervalo de tiempo de segundos o minutos. Por este motivo, en lugares como:

- Piscinas (naturales o artificiales)
- Parques
- Parques acuáticos
- Embalses
- Charcas
- Pozos
- Acuíferos
- Depuradoras
- Desalinizadoras
- Depósitos de agua
- Fabricas
- Industria Química
- Agricultura y ganadería (sector primario)

Sería interesante el estudio de la implantación de este proyecto para la monitorización del estado del agua que contiene garantizando así: la aptitud para el consumo humano, estado de la instalación que contiene el agua, mejora de la

calidad de vida de las personas, así como de la ausencia de algún elemento tóxico o dañino.

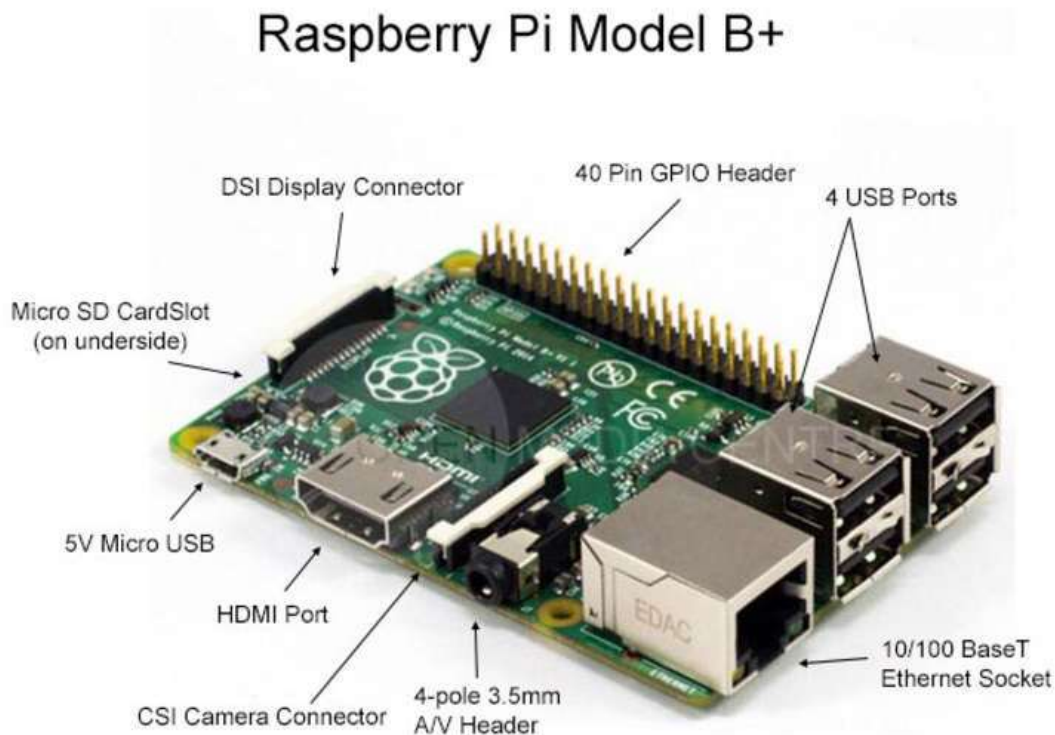
Con este proyecto instalado se lograría también, la eliminación de registros en soportes como el papel o directamente en el ordenador de un operario, dejando constancia así, de forma persistente de los resultados obtenidos con una mayor precisión del que se puede proveer de cualquier dispositivo comercial común.

En definitiva, este proyecto nace con la esperanza de automatizar y facilitar una tarea repetitiva y para mejorar la calidad de vida de las personas que se aprovechen de las ventajas que aporta este proyecto.

Material necesario

Microcontrolador

Para la realización de este sistema autónomo de recogida de datos se ha decantado por la opción de uno de los microcontroladores más conocidos como es la Raspberry Pi 2 modelo B.



Especificaciones

- *Procesador Broadcom BCM2836 de 900 MHz ARM Cortex-A7 de cuatro núcleos con GPU VideoCore IV de doble núcleo*
- *GPU proporciona una tecnología Open GL ES 2.0, hardware acelerado OpenVG y admite imágenes de alta resolución 1080p30 H.264*
- *GPU tiene una capacidad de 1 Gpixel/s, 1,5 Gtexel/s o 24 GFLOPs con filtrado e infraestructura DMA*
- *SDRAM LPDDR2 de 1 GB*
- *Salida de vídeo: HD 1080p*
- *Salida de vídeo compuesto (PAL/NTSC)*
- *Salida de audio estéreo*
- *Conector hembra Ethernet RJ45 10/100 BaseT*
- *Conector hembra de vídeo/audio HDMI 1.3 y 1.4*

- *Conector hembra de salida de vídeo compuesto/audio de 3,5 mm 4 polos*
- *4 conectores hembra USB 2.0*
- *Conector MPI CSI-2 de 15 vías para cámara de vídeo HD Raspberry Pi (775-7731)*
- *Conector de interfaz serie de display de 15 vías*
- *Conector para tarjeta MicroSD*
- *Arranca desde tarjeta MicroSD, ejecutando una nueva versión del sistema operativo Windows 10 IOT*
- *Conector macho de 40 pines para buses serie y GPIO (compatible con el conector macho de 26 pines Raspberry Pi 1)*
- *Fuente de alimentación: +5V a 2A a través de conector hembra microUSB*
- *Dimensiones: 86 x 56 x 20 mm*

Características

Debido a las especificaciones mostradas anteriormente, se ha elegido esta versión por las características respecto a los anteriores modelos de Raspberry como son:

- Un procesador Quad Core con una frecuencia de reloj de 900MHz.
- 1 GB de memoria RAM, lo que nos permite aplicaciones más potentes.
- 40 pins GPIO totalmente compatibles con los anteriores modelos aun teniendo los modelos anteriores solamente 26 pins.
- Ranura Micro SD para el almacenamiento de datos o la carga de sistemas operativos.
- Jack de audio con capacidad de reproducción en estéreo.
- Se puede proveer hasta 1.2 AMP al puerto USB para habilitar la conexión de dispositivos USB que consuman más electricidad.

Por estas razones se ha elegido este microcontrolador ya que también puede ser exportado y adaptado sin cambios notables a microcontroladores superiores a éste.

Sonda pH

La sonda utilizada para la medición de pH será esta:



Esta sonda nos permite detectar el pH del líquido en el que se sumerja de una forma analógica.

Especificaciones

- *Alimentación: 5.00V*
- *Dimensiones: 43x32mm (controlador)*
- *Rango de medición: 0-14 pH*
- *Temperatura de medición: 0-60 °C*
- *Precisión: $\pm 0.1\text{pH}$ (25 °C)*
- *Tiempo de respuesta: $\leq 1\text{min}$*
- *Sonda de pH con conector BNC*
- *Controlador pH 2.0 (3 pines)*
- *Ajuste de ganancia*
- *Indicador LED*

MCP3008

Debido a que la Raspberry Pi no dispone de ninguna entrada analógica, y la sonda tiene una interfaz de salida analógica es necesario la utilización del MCP 3008 que es un convertor de datos analógicos a datos digitales.



Con este convertor se puede disponer de 8 dispositivos analógicos utilizando los pines de la Raspberry que se corresponden con el protocolo SPI.

Adaptador Wi-Fi TP-Link TL-WN725N

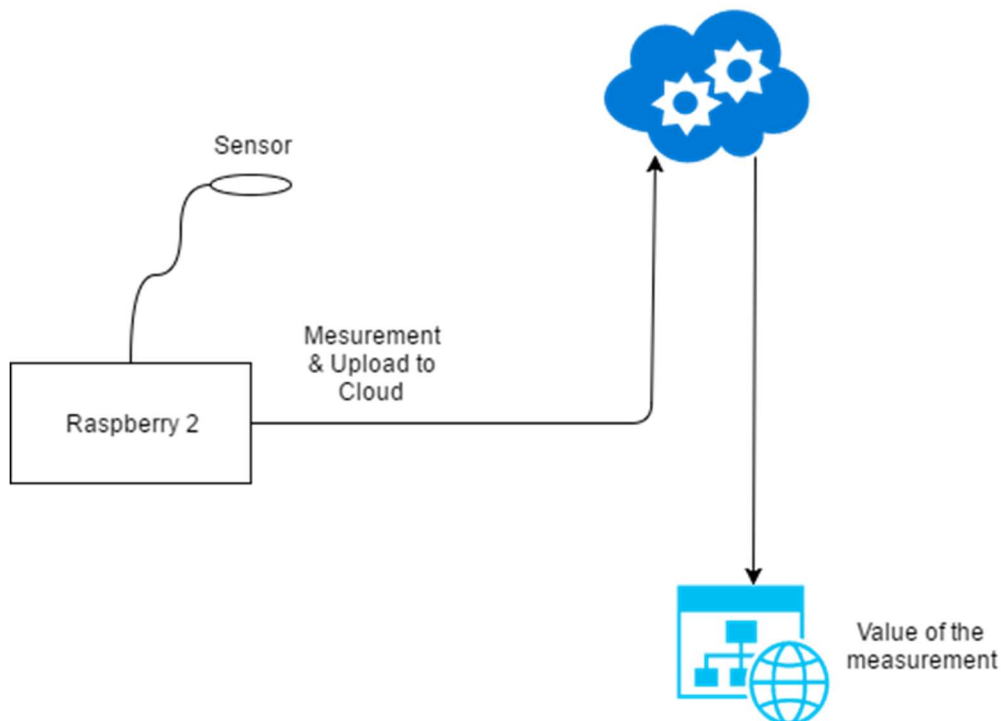
Puesto que la Raspberry Pi no dispone de comunicación inalámbrica, solamente de un adaptador Ethernet 10/100, y con la finalidad de hacer de este proyecto un sistema autónomo se conectará este dispositivo.



Debido a que es un dispositivo Plug&Play solamente será necesario la conexión a la Raspberry Pi y la autenticación mediante la función dada, para la conexión a una red, por el sistema operativo, *Windows 10 IOT*.

Diseño del proyecto

Diagrama de diseño del proyecto

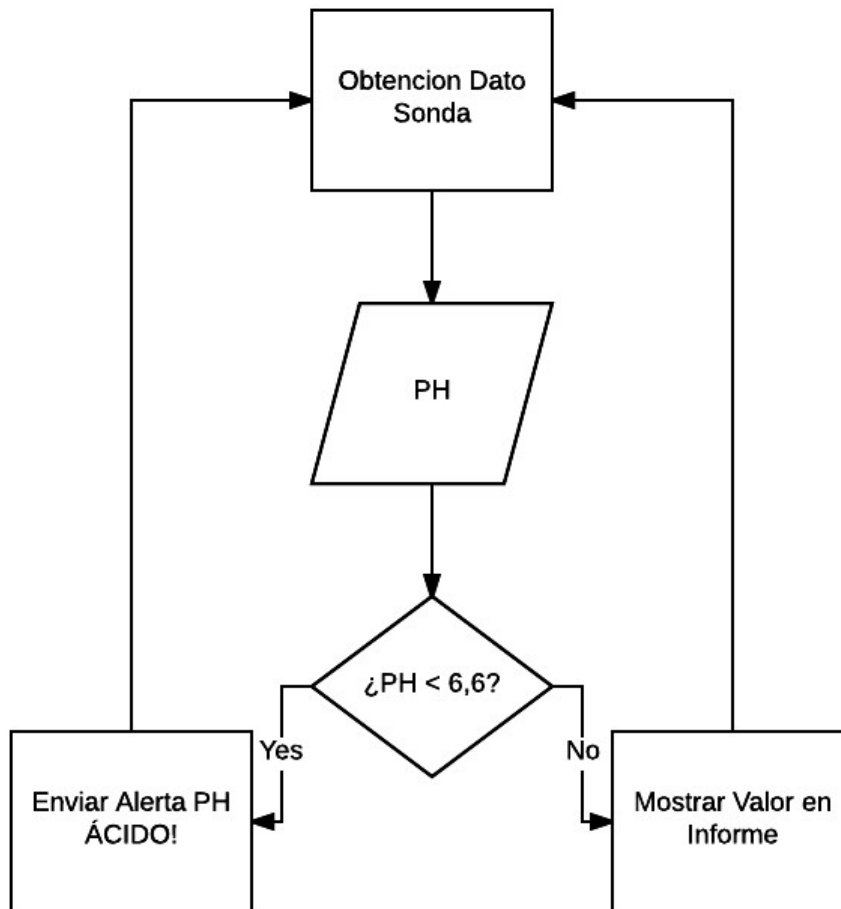


Este proyecto, a grandes rasgos funcionara tal y como se puede ver en el gráfico colocado debajo de estas líneas.

Por tanto, el proyecto utilizará la variable obtenida por la sonda y transformada mediante un programa, realizado en C#, que se ocupará de dicha transformación y de la posterior subida al Cloud (representada en la primera parte del gráfico). También, al mismo tiempo, mediante una función característica de Azure llamada "*Stream Analytics*" se procesa el dato, en tiempo real (*streaming*), y se envía a una aplicación de Microsoft llamada "*Power Bi*" cuyo funcionamiento nos permitirá mostrar al usuario, la variable interesada mediante la creación de informes, con diferentes opciones, en tiempo real y a su vez en cualquier dispositivo que tenga una conexión a Internet (smartphone, Tablet, ordenador...). Finalmente, como añadidura al proyecto, y que es interesante, la creación de un evento, en dicho visor de variables, que nos alertará de si en un momento determinado aquel valor a medir, en nuestro caso el pH, se encuentra por debajo de un límite con el fin de que se corrija el desperfecto ocasionado y este valor se vuelva a encontrar dentro del intervalo de confianza.

Diagrama de flujo del proyecto

El funcionamiento básico del proyecto es la siguiente:



En este diagrama se pretende mostrar solamente aquella funcionalidad más básica sin entrar detalladamente en cada uno de los procesos que se ven en el diagrama.

Por lo que se puede observar, el proyecto se encarga de leer el dato transmitido por la sonda el cual hay que tratarlo y transformarlo a pH puesto el dato que envía la sonda en un potencial eléctrico dado por el líquido en el que hemos sumergido este sensor.

A continuación, se obtiene el pH y se evalúa mediante una condición y se dan dos opciones para la posible respuesta de la evaluación. Si se cumple dicha condición, además de mostrarse en un informe a tiempo real, se emitirá una alerta al usuario final resumiendo el motivo de la alerta para que posteriormente ésta sea corregida. Si el resultado de la evaluación es negativo este valor se mostrará en un informe en tiempo real al usuario final.

Ensamblaje del proyecto

En este apartado veremos las conexiones y el cableado necesario para que el microcontrolador reciba datos de la sonda.

Para que se reciban los datos en el microcontrolador es necesaria la conexión de los pines correspondientes del bus de datos a utilizar. En caso de la Raspberry Pi 2 modelo B, tiene 2 configuraciones posibles más utilizadas.

Bus SPI

SPI es el acrónimo de “*Serial Peripheral Interface*”, protocolo de comunicación síncrona (existencia de una línea *Clock* que se encargará de leer o escribir un bit) de 4 hilos que permite el alcance de velocidades elevadas con la posibilidad de transmisión *full-duplex*, o comunicación bidireccional, por el que se puede transmitir datos y recibir datos al mismo tiempo.

Estos 4 hilos son:

- SCLK (*Clock*): Pin del reloj. Encargado de comunicar los pulsos de reloj.
- MOSI (*Master Output Slave Input*): Conectará la salida del dispositivo *master* con la entrada del dispositivo *slave*.
- MISO (*Master Input Slave Output*): Conectará la entrada del dispositivo *master* con la salida del dispositivo *slave*.
- SS: Se podrá seleccionar un *slave* o para activar un *slave* desde el *master*.

Ventajas

- Comunicación *Full Duplex*.
- Más rápido que el I2C y que el puerto Serie asíncrono normal.
- El tamaño de los mensajes puede ser arbitrariamente grande.
- Se requiere un hardware sencillo.
- Requiere un menor consumo y menor electrónica de conexión que el I2C.
- Como el *Clock* lo proporciona el *master*, los *slave* no necesitan osciladores.

Inconvenientes

- Es necesario una mayor cantidad de pines que el bus I2C o el Puerto serie normal.
- Las comunicaciones tienen que estar perfectamente establecidas de antemano.
- No hay señal de conformidad del *slave*.
- *Master* único.
- Funciona solo en distancias muy cortas
- Normalmente necesita un pin adicional por cada *slave* siendo un problema si el número de estos se eleva.

Bus I2C

I2C es el acrónimo de *Inter-Integrated Circuit*, es un bus serie de datos usado principalmente para la comunicación entre diferentes partes de un circuito de una forma síncrona.

Está diseñado como un bus maestro-esclavo, por la que el maestro siempre inicializa la transferencia de datos y el esclavo reacciona permitiendo también el modo multi-maestro por el que se pueden comunicar dos maestros entre sí.

Su principal característica es la utilización de dos líneas para la transmisión de información:

- SDA: Canal de transmisión de datos.
- SCL: Canal Reloj por el que se regirá la comunicación síncrona.
- GND: Pin de conexión de toma tierra.

Las dos primeras líneas necesitarán de resistencias *pull-up* para evitar causar conflicto en las transmisiones de bits al mismo tiempo con diferente valor. Así los dispositivos estarán en un nivel bajo mientras que el bus en un nivel alto forzando, con esto, a que si se envía algo se tenga que comunicar al bus.

Debido a lo expuesto anteriormente, para nuestro proyecto se ha optado por la elección del bus SPI ya que nos permite:

- Comunicación en *full-duplex*.
- Mayor velocidad.
- Sencillez en el hardware

Ensamblado del hardware

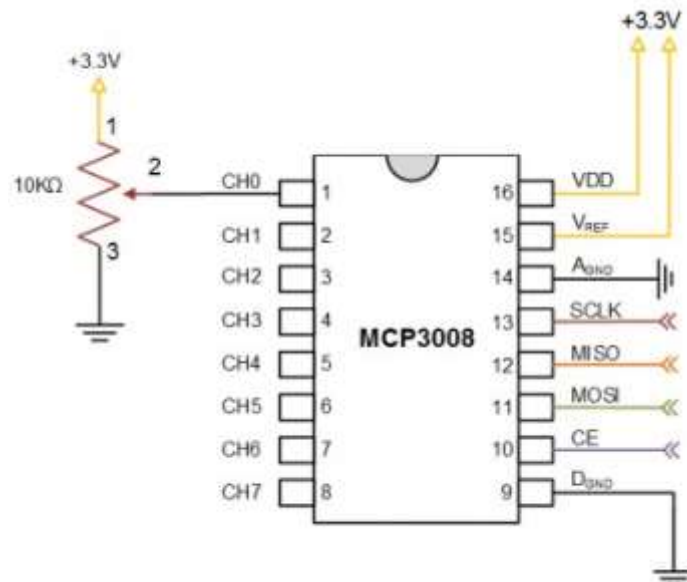
Una vez elegido el bus, fundamental tanto para la comunicación con la sonda como para el tratamiento de los datos, será necesario el cableado para dicho fin.

Para esto definiremos dos fases, la primera que abarcará el cableado de la Raspberry con el convertor MCP3008 y la segunda que tendrá como objetivo añadir la sonda al montaje anterior.

Cableado Raspberry – MCP3008

Para llevar a cabo esta tarea, es necesario consultar la ficha de especificaciones que proporciona el fabricante del MCP3008 y de Raspberry con el fin de conectar cada borne correctamente para que la comunicación sea efectiva.

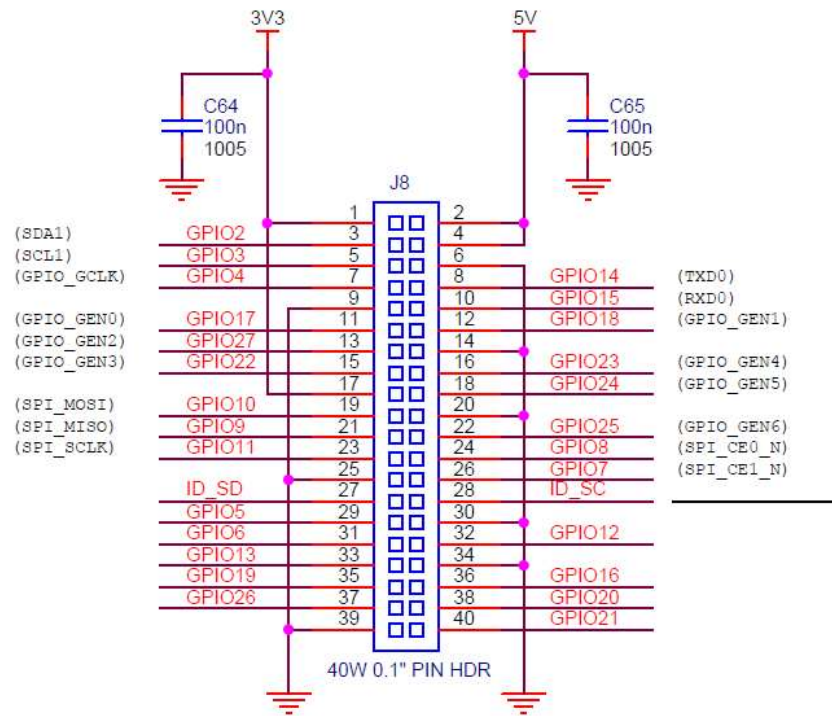
MCP3008 Schematics



En este convertor se indica que los bornes 10, 11, 12, 13 son aquellos encargados de llevar la comunicación mediante el protocolo SPI, así que son los que se conectarán en los pines correspondientes de la Raspberry.

A la izquierda de la imagen se puede ver que hay 8 canales disponibles por los que se puede enviar información. En este proyecto solo se utilizará el canal 0 (CH0, borne 1) puesto que solamente se dispone de un periférico.

Raspberry Pi Schematics

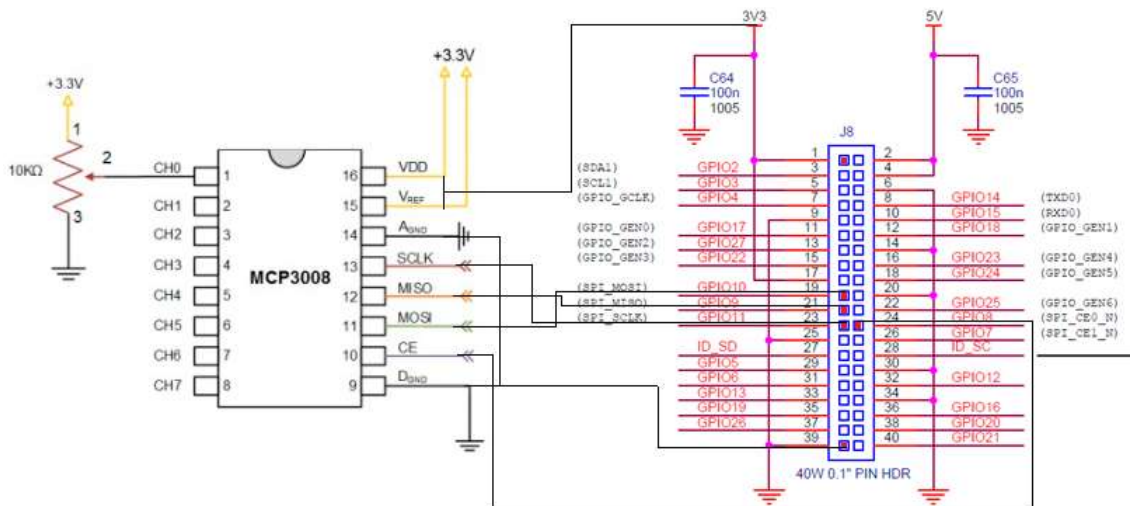


Aquí se observa la especificación de la Raspberry Pi 2 modelo B para la conexión de periféricos llamado GPIO.

GPIO (*General Purpose Input/Output*) es un pin en un chip que se utiliza, como su nombre indica, para la *Entrada/Salida de propósito genérico*.

Para este proyecto se utilizarán los pines GPIO correspondientes con el protocolo cuyo bus de transmisión se utilice. En este caso, se procederá al conexionado de la Raspberry y el conversor MCP3008 mediante el bus SPI así que serán utilizados los pines GPIO 8, GPIO 9, GPIO 10, GPIO 11 y, por supuesto, que también el pin 4, pin 6 para la alimentación tanto del MCP 3008 como de la sonda.

A continuación, se muestra el diagrama de conexión de la Raspberry con el conversor MCP3008.



Cableado MCP3008 – Sonda

Para la correcta recepción del valor de pH y la puesta en funcionamiento de la sonda que medirá dicho valor, es necesario su conexión al conversor para la transformación de la entrada analógica a entrada digital ya que la Raspberry Pi, como se ha comentado anteriormente, no dispone de entradas analógicas.

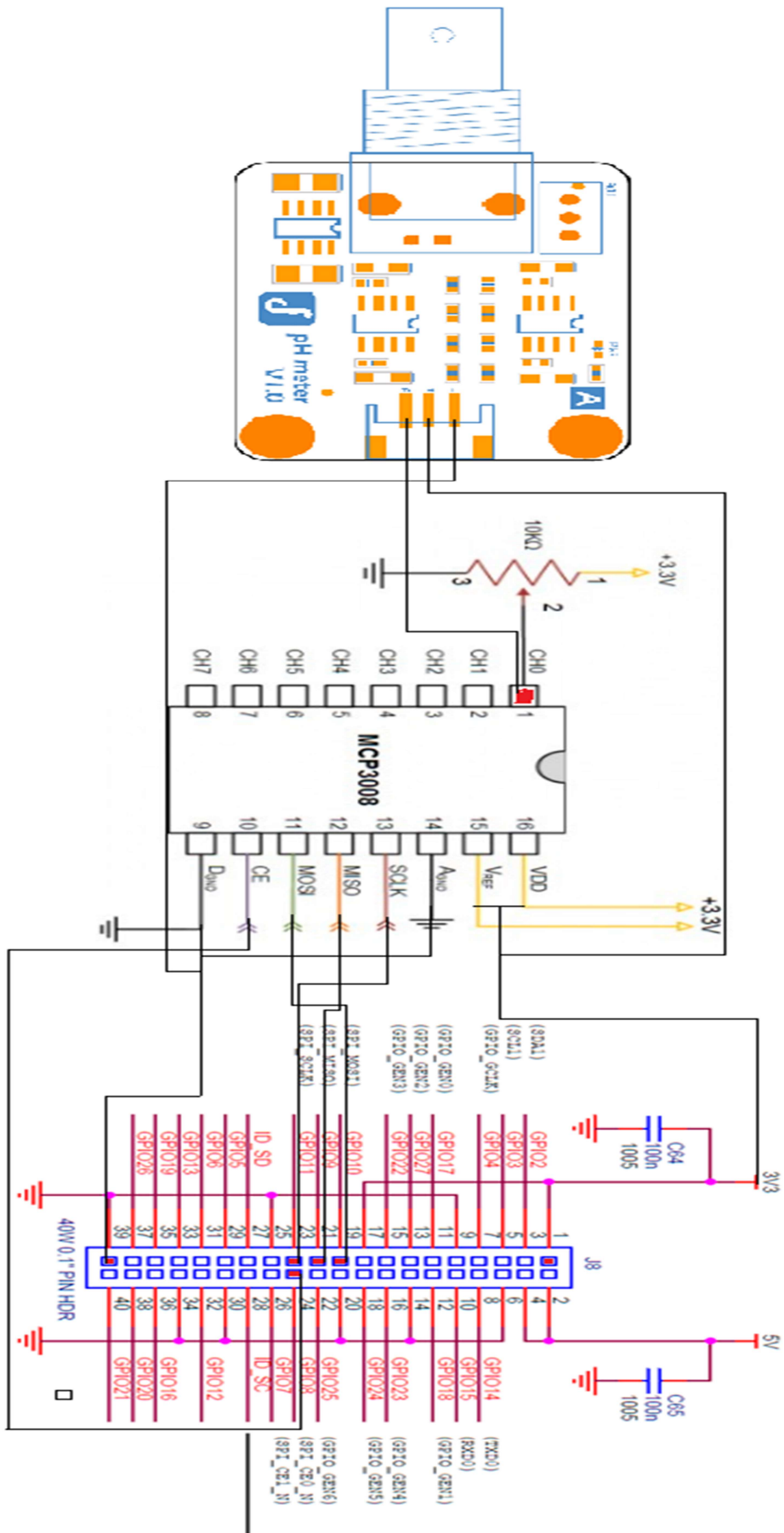
En este caso, la conexión es relativamente sencilla puesto que solamente se debe identificar el borne correspondiente con la salida de datos de la sonda y conectándola al canal del MCP3008 correspondiente. Para esto, se conectará la salida de datos con el canal 0 del conversor. Además de tener que añadir dos cables más al conexionado para dar suministro de electricidad y una toma GND.

A continuación, se muestra el diagrama de conexión de todo el proyecto al completo puesto que es necesario ver la conexión con los puertos Vcc y GND de la Raspberry Pi y del canal 0.

Una vez realizado todo el cableado solamente habrá que comprobar que todo el conjunto recibe electricidad para que esté listo para funcionar. En este caso solamente habrá que encender la Raspberry y comprobar que en la sonda existe un led de comprobación que si está todo correctamente conectado se encenderá.

En el caso que alguna conexión sea errónea es posible comprobar donde ocurre el fallo, normalmente por conexión en otros pines, mediante un multímetro que ajustaremos debidamente (20V en la posición de corriente continua). Si este multímetro no indica el valor aproximado de 3,3 o 5 V, ya que son las dos tensiones que puede suministrar la Raspberry, significa que habrá que reconfigurar de nuevo el cableado.

*En el Anexo se encuentra una fotografía tomada del cableado en un entorno real.



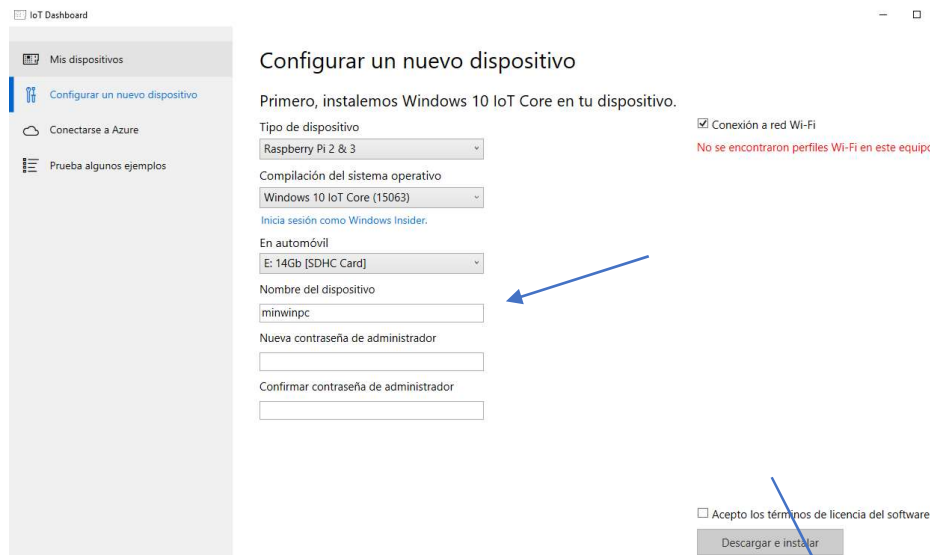
Programación del sistema

Como se puede observar en el apartado de Diseño del proyecto se empezará con la puesta en marcha de la Raspberry Pi, a continuación detectaremos la sonda y analizaremos el valor que proporciona para más tarde enviar este valor a la nube para posteriormente analizarlo con Power Bi y Microsoft Flow.

Raspberry Pi 2

Para este proyecto se instalará en la Raspberry el sistema operativo de Microsoft diseñado especialmente con el fin de la comunicación con la nube.

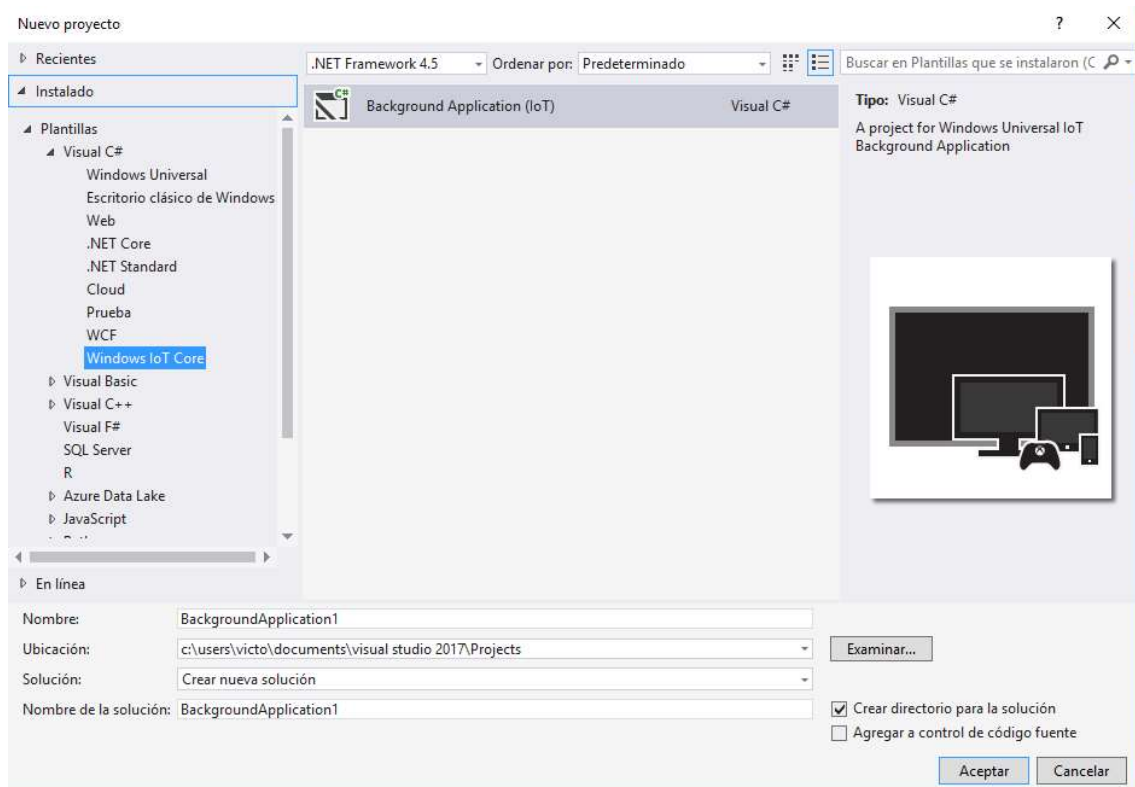
Se trata de Windows 10 IOT y para lo cual solamente se tendrá que descargar “Windows 10 IOT Core Dashboard” e instalar en el ordenador. Una vez instalado, solamente introducir la tarjeta SD y a continuación, entrar a la pestaña “Configurar un nuevo dispositivo” y pulsar sobre “Descargar e instalar” y ya únicamente esperar para que se instale. Cuando acabe insertar en la Raspberry e iniciar el microcontrolador.



Obtención de pH

Una vez la Raspberry tenga instalado el sistema operativo, abrir “Visual Studio 2017” para empezar a crear el programa que proporcione el valor que se necesita.

Para esto, abrimos el editor “Visual Studio 2017” y crearemos una nueva solución con una plantilla llamada “Windows IoT Core”.



A continuación, en el fichero “StartupTask.cs”, que es el fichero donde ejecutaremos el código que sea necesario para obtener dicho valor.

En este fichero, importaremos una librería de ayuda para el protocolo de comunicación que se utilizara siendo la librería “Windows.Devices.Spi” y además una librería para el tratamiento de JSON que se transmitirá a Azure, “Newtonsoft.Json”.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net.Http;
using Windows.ApplicationModel.Background;
using System.Threading.Tasks;
using Microsoft.Azure.Devices.Client;
using Windows.System.Threading;
using Windows.Devices.Enumeration;
using Windows.Devices.Spi;
using Newtonsoft.Json;
```


Seguidamente se añadirá una referencia “*Windows IoT Extensions for UWP*” que complementará la solución ya que esta referencia es un SDK (kit de desarrollo de software).

También se agregará un paquete NuGet llamado “*Microsoft.Azure.Devices.Client*”, el cual instala un SDK al dispositivo microcontrolador para dispositivos Azure IoT.

Una vez esté preparado el editor con todas las referencias que serán útiles para el desarrollo del proyecto, solamente habrá que saber utilizar el bus SPI.

```
//*****  
//  
//          CONEXION AL BUS SPI  
//  
//*****  
1 referencia  
private async void busSpi()  
{  
    try  
    {  
        var spiSettings = new SpiConnectionSettings(0);  
        spiSettings.ClockFrequency = 1350000;  
        spiSettings.Mode = SpiMode.Mode0;  
  
        string spiQuery = SpiDevice.GetDeviceSelector("SPI0");  
        var deviceInfo = await DeviceInformation.FindAllAsync(spiQuery);  
        if (deviceInfo != null && deviceInfo.Count > 0)  
        {  
            mcp3008 = await SpiDevice.FromIdAsync(deviceInfo[0].Id, spiSettings);  
        }  
        else  
        {  
            System.Diagnostics.Debug.WriteLine("DISPOSITIVO SPI NO ENCONTRADO!!");  
        }  
    }  
    catch (Exception ex)  
    {  
        System.Diagnostics.Debug.WriteLine("!!!! SPI NOT FOUND: " + ex.Message);  
    }  
}
```

Para la utilización de SPI se utilizará la librería importada con anterioridad la cual proporciona aquello necesario para la conexión por el bus SPI.

Primeramente, crearemos una conexión añadiendo la frecuencia de reloj, ya que es un protocolo síncrono, y el modo de comunicación (Mode0).

A continuación, se buscan todos los buses del sistema se selecciona el bus SPI que se quiera utilizar (SPI0).

Una vez el bus SPI está establecido se leerán los datos que proporciona la sonda y se transformará a un valor útil.

```
private float obt_ph()
{
    try
    {
        //1 - single ended
        // Canal 0 = 10000000 = 0x80
        // 3 Byte is empty
        var transmitBuffer = new Byte[3] { 1, 0x80, 0x00 };
        var receiveBuffer = new Byte[3];

        mcp3008.TransferFullDuplex(transmitBuffer, receiveBuffer);

        var result = ((receiveBuffer[1] & 3) << 8) + receiveBuffer[2];

        voltios = (result * 5.0f) / 1024;
        ph = 3.50f * voltios + offset; //Offset visto entre la diferencia del ph de un liquido calibrador = 7.0 y la lectura dada por el dispositivo la 1ª vez
    }

    catch (Exception ex)
    {
        System.Diagnostics.Debug.WriteLine("ERROR OBTENCION DE DATOS!");
        System.Diagnostics.Debug.WriteLine("!!!! WRONG CONVERSION: " + ex.Message);
    }

    return ph;
}

//*****
//
//                               OBTENCION DEL PH
//
//*****
1 referencia
private float obt_ph()
{
    try
    {
        //1 - single ended
        // Canal 0 = 10000000 = 0x80
        // 3 Byte is empty
        var transmitBuffer = new Byte[3] { 1, 0x80, 0x00 };
        var receiveBuffer = new Byte[3];

        mcp3008.TransferFullDuplex(transmitBuffer, receiveBuffer);

        var result = ((receiveBuffer[1] & 3) << 8) + receiveBuffer[2];

        voltios = result * (5000.0f / 1024); //5V = 5000 mV
        ph = (3.5f * voltios);
        //Console.WriteLine("VOLTAJE: " + voltios + " V || PH: " + ph);
    }

    catch (Exception ex)
    {
        System.Diagnostics.Debug.WriteLine("ERROR OBTENCION DE DATOS!");
        System.Diagnostics.Debug.WriteLine("!!!! WRONG CONVERSION: " + ex.Message);
    }

    return ph;
}
```

Para tal fin se establecerán dos buffers puesto que se necesitan 2 para cumplir con la condición de full-duplex y se asignarán a la variable que representa el microchip conversor.

Ahora, se filtran los datos recibidos y se almacenan en *result* para después transformarlas.

Para transformar los datos se hará notar que el MCP3008 tiene una resolución de 10 bits por tanto el voltaje leído es representado por un valor dentro del rango de 0-1023, 1024 posiciones. También se tendrá en cuenta que el voltaje de entrada al conversor es de 5 Voltios o 5000 milivoltios y se ajusta un valor, llamado offset, mediante un calibrador de pH (sustancia con un pH determinado).

EQUATION 4-2: DIGITAL OUTPUT CODE CALCULATION

$$Digital\ Output\ Code = \frac{1024 \times V_{IN}}{V_{REF}}$$

Where:

V_{IN} = analog input voltage

V_{REF} = analog input voltage

Transmisión a Azure

Obtenido ya el pH la finalidad es transmitir el dato a Azure para tratarlo más adelante.

Para esto solamente es necesario un método para realizar la transmisión mediante la creación de un fichero JSON y codificar dicho fichero correctamente.

```
//*****
//
//          SUBIR DATOS A AZURE
//
//*****
//Metodo para enviar el ph al CLOUD
1 referencia
private async Task SendMessageToIotHubAsync(string sensorType, float phValue)
{
    try
    {
        var phConverted = phValue;
        //Prueba de Serializacion
        var telemetry = new
        {
            sensorType,
            phConverted,
        };

        var messageString = JsonConvert.SerializeObject(telemetry);
        var message = new Message(Encoding.UTF8.GetBytes(messageString));

        await deviceClient.SendEventAsync(message);
    }
    catch (Exception ex)
    {
        System.Diagnostics.Debug.WriteLine("!!!! " + ex.Message);
    }
}
```

Y solo falta un método que actúe como “reloj” para que las mediciones se realicen en un tiempo determinado (segundos, minutos...).

```
//Enviamos un mensaje a Azure cada 1 Minuto
messageTimer = ThreadPoolTimer.CreatePeriodicTimer(MessageTimer_Tick, TimeSpan.FromMinutes(1));

private async void MessageTimer_Tick(ThreadPoolTimer timer)
{
    obt_ph();
    await SendMessageToIotHubAsync("ph", ph);
}
```

Configuración de Azure

Creación de un IOT HUB

La primera tarea que se debe hacer es crear un IOT HUB. Para ello solamente debe crearse por defecto con un nombre y un grupo de recursos nuevo.

The screenshot shows the Azure portal interface for creating an IoT Hub. The 'Choose your pricing and scale tier' dialog is open, showing three pricing tiers: S1 Standard (400k messages/unit/day), S2 Standard (6M messages/unit/day), and S3 Standard (300M messages/unit/day). The F1 Free tier is also visible, with a note that only one Free IoT hub is allowed per subscription. Blue arrows point to the 'Name your hub' field, the 'Pricing and scale tier' dropdown (set to S1 - Standard), the 'IoT Hub units' input (set to 1), the 'Device-to-cloud partitions' dropdown (set to 4), the 'Subscription' dropdown (set to Pago por uso), the 'Resource group' dropdown (set to Create new), and the 'Location' dropdown (set to West Europe). The 'Create' button is visible at the bottom.

Selección de la tarifa Free que permite tener un único IOT HUB.

Una vez creado el IOT Hub, se deberá acceder a las “Políticas de acceso compartido” donde estará la cadena “*ConnectionString-Primary Key*” que será necesario para añadir al código desarrollado para el correcto envío del pH.

Finalmente, en el código desarrollado se debe añadir esta cadena.

```
private const string IOT_HUB_CONN_STRING = "HostName=phMeter-iotHub.azure-devices.net;DeviceId=PHMeter;SharedAccessKey=va+C4Uxyy0s/ePQPlfWsp7Q/kbv+7/0lULyxafIHvA=";  
  
//Instanciamos el dispositivo cliente Azure  
deviceClient = DeviceClient.CreateFromConnectionString(IOT_HUB_CONN_STRING);
```

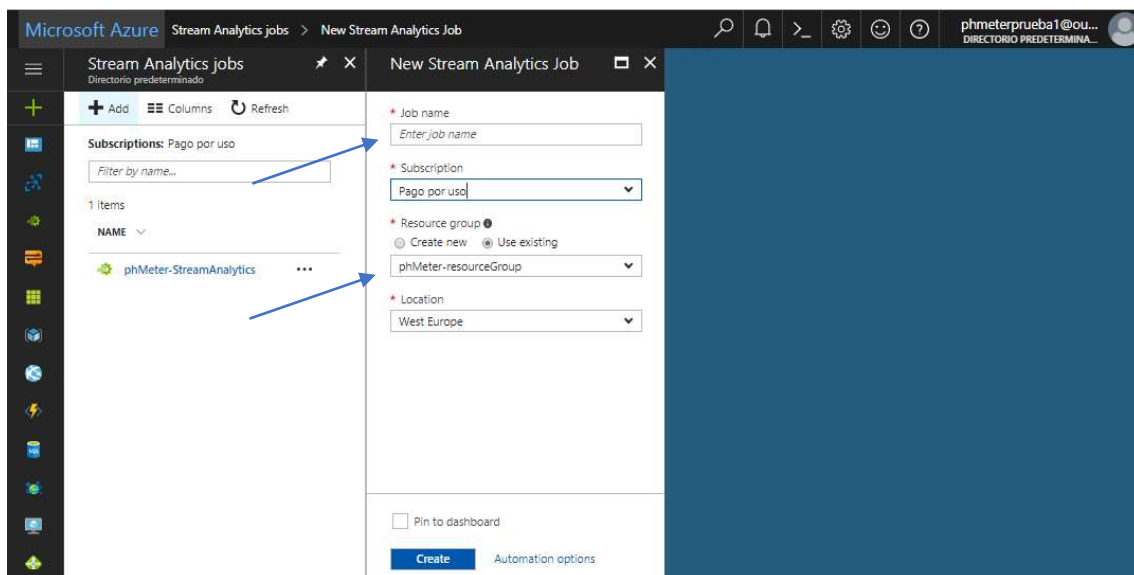
De esta forma se consigue una transmisión correcta de los datos a la nube.

Creación de Stream Analytics Job

Para la visualización de los datos se utilizará esta funcionalidad que permite la emisión en directo, *streaming*, y que posteriormente, se visualizará tanto en directo como en una gráfica a modo de histórico.

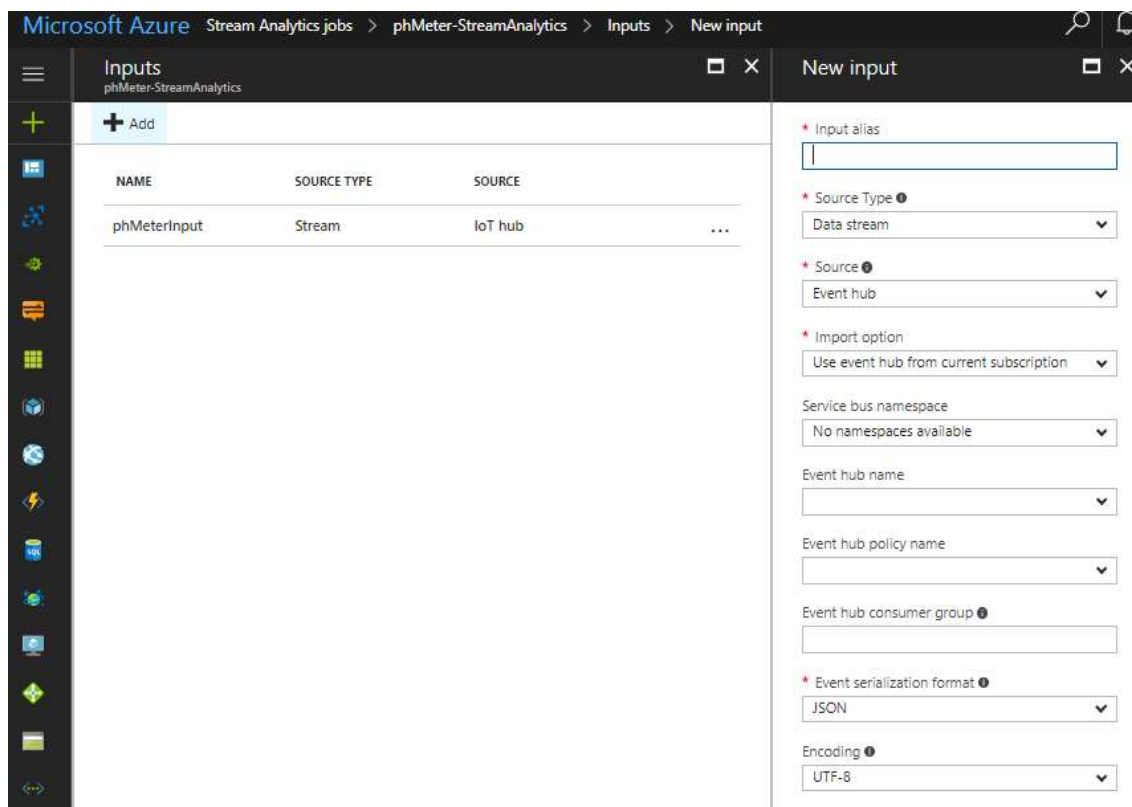
Solamente queda añadir un servicio más para ello pulsar sobre el símbolo +, buscar “Internet of Things” y crear un Stream Analytics.

Ya solo queda introducir el nombre del servicio y pulsar sobre “*Use existing resource group*” seleccionando el grupo de recursos creado para el IOT Hub.



Una vez el servicio esté creado, se creará una entrada de datos “*Input*”, una consulta “*query*” y una salida “*output*”.

Para ello, se entra dentro del menú Input se completa el formulario teniendo en cuenta que la fuente de datos es IOT HUB y los demás campos se mantendrán por defecto.



Seguidamente se creará la salida de los datos para lo que se configurará según la fuente de salida. En este caso será Power BI por lo que solamente será necesario una cuenta gratuita para poder autenticarse en el momento que se solicite en la creación de la salida.

"Click on "+Add"

- *Output Alias: "----"*
- *Sink: "Power BI"*
- *Click on "Authorize" if asked to and enter your Power BI account credentials"*

Y finalmente, se creará la consulta donde se utilizará el alias que se haya facilitado en la creación de la entrada (Input) y el alias de salida que se haya facilitado en la creación de la salida (output) y en la parte SELECT utilizaremos el operador * para indicar que se transmitirá todo.

Ya faltará pulsar sobre “Start” para que empiece a transmitir datos en directo.

**en anexo están las imágenes del proyecto real.

Configuración de Power Bi

Power BI es un conjunto de herramientas de análisis de datos empresariales y también ofrece un visor de datos configurando una entrada de datos desde la nube.

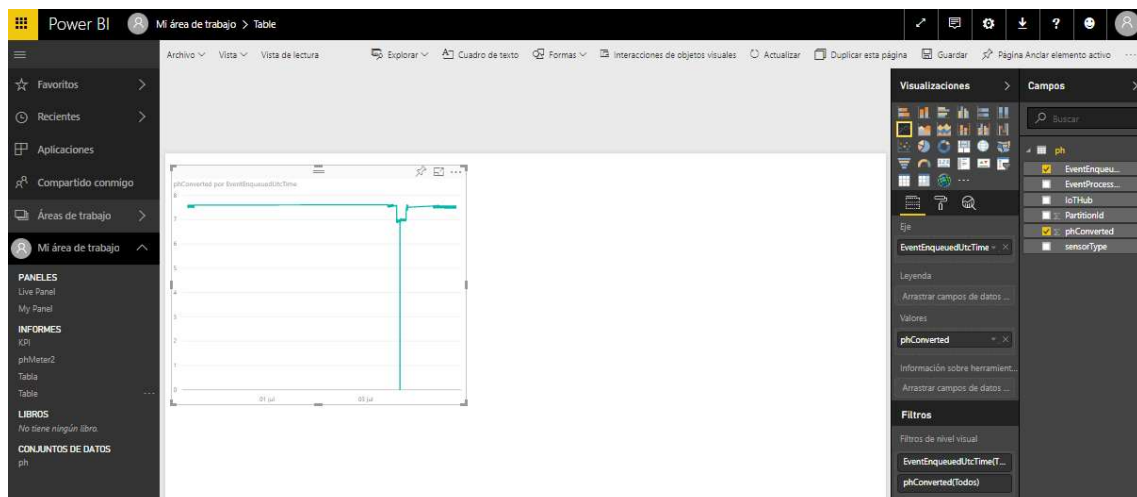
Una vez la cuenta haya sido creada, se tiene que entrar y configurar la entrada de datos.

Visualización de los datos

Esto será automático ya que en la salida de datos del servicio de Stream Analytics se configuró Power BI como receptor de estos datos.

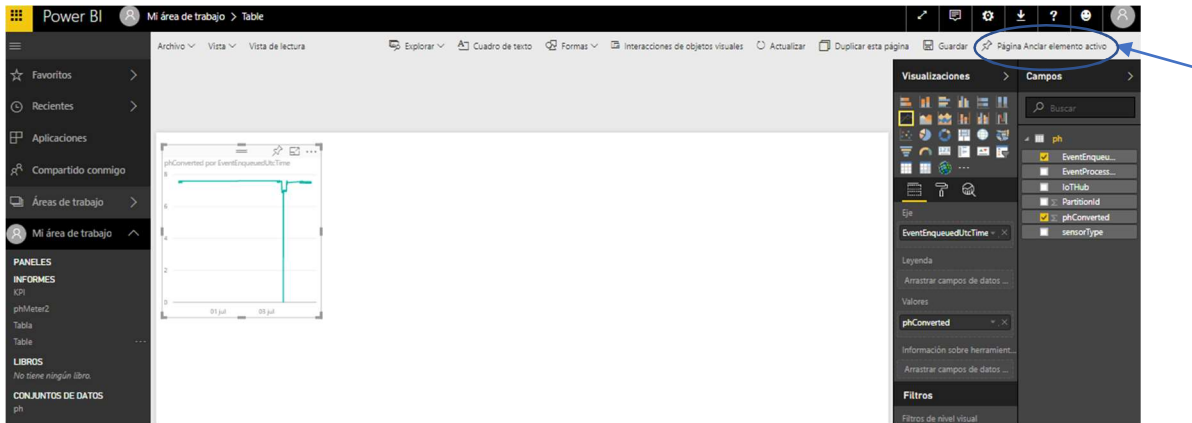


Una vez el conjunto de datos haya aparecido se creará un informe detallando la variable que se quiera visualizar. Seguidamente creado el informe, con la forma que sea idónea, se creará un Panel para que la medición se visualice en tiempo real.

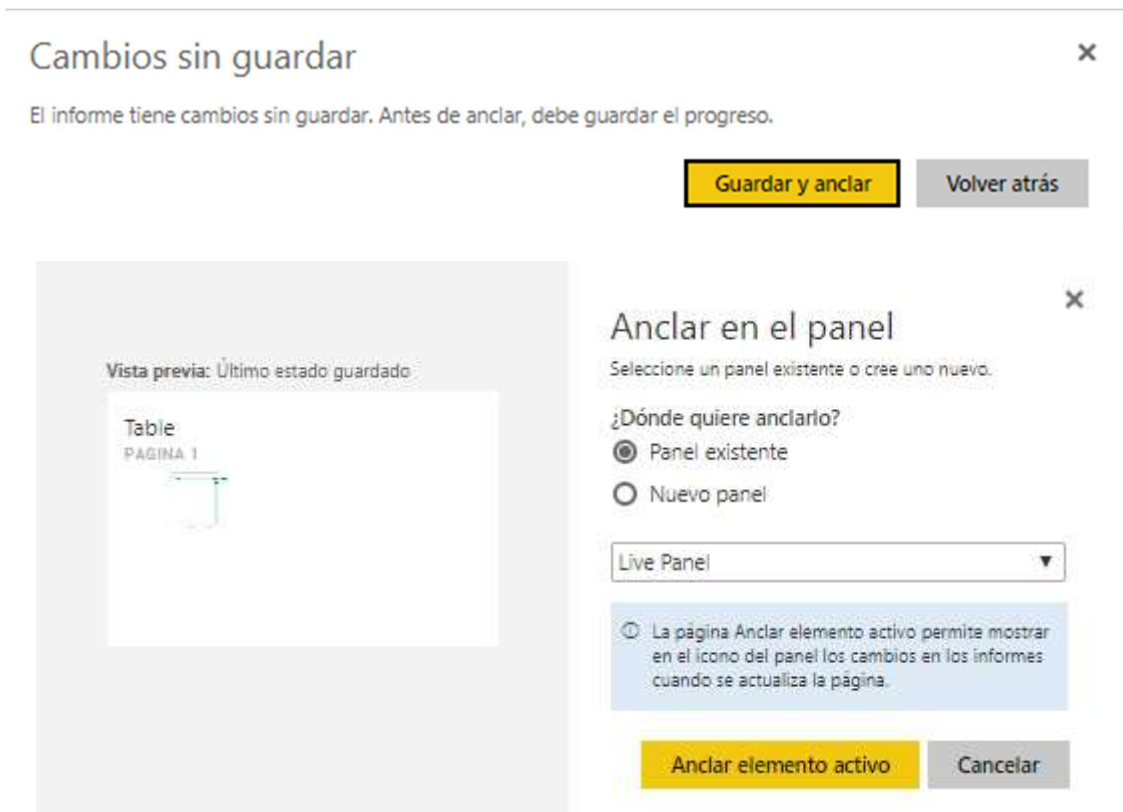


Anteriormente se ha creado el informe en forma de tabla seleccionando las variables a visualizar en los ejes x e y. Esta variable será el pH como campo de datos y el tiempo UTC de encolado del evento.

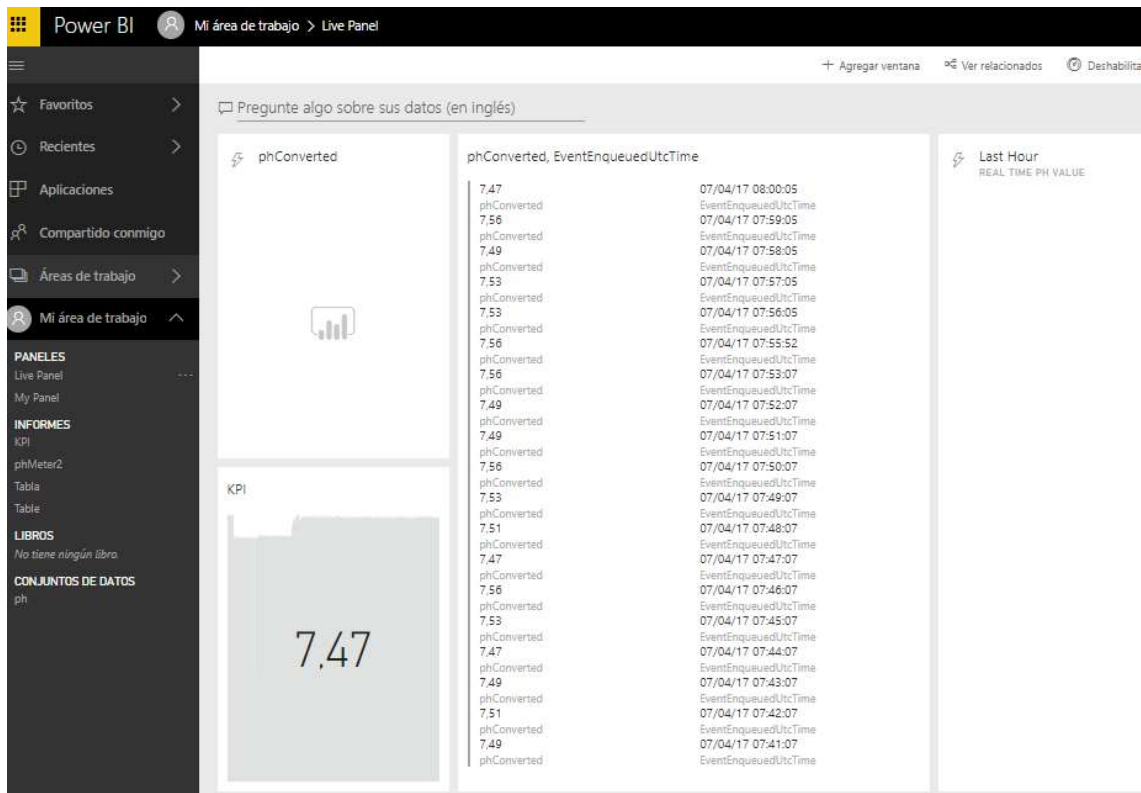
A continuación, se creará el panel donde se visualizará en tiempo real. Para ello solamente se debe de pulsar sobre el botón “Página Anclar a elemento activo”



Seguidamente se guardará el informe y se creará un panel nuevo.



Y al final, se pulsa dentro del panel y se podrá visualizar el panel en tiempo real.



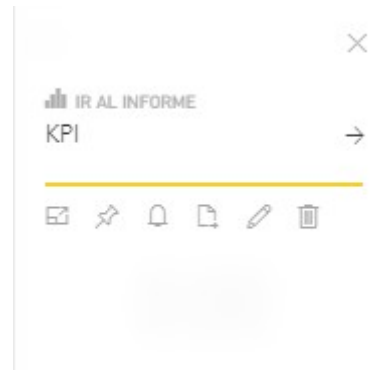
Aquí se puede ver que hay un marcador a modo de visualizador en tiempo real y una tabla a modo de histórico y una gráfica del pH durante la última hora.

Creación de Alertas

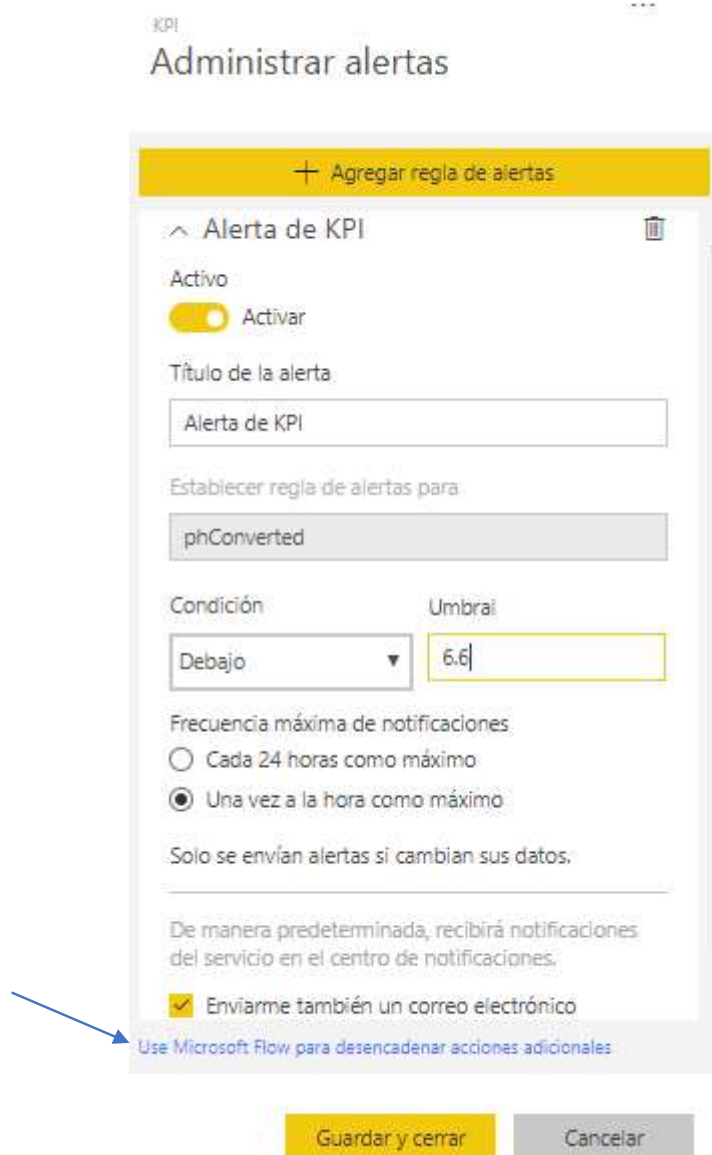
En el momento en el que el valor del pH sea inferior al límite establecido por la ley o según el límite que determina la acidez de una disolución, será interesante configurar una alerta para que ésta avise al usuario final para que dicha deficiencia sea corregida a la mayor brevedad posible.

Para ello se puede configurar una alerta en los gráficos de Power Bi y más adelante se utilizará un servicio de Microsoft llamado Microsoft Flow. Este servicio se aprovecha de la alerta lanzada por Power Bi y envía un correo de alerta al usuario final que sea el encargado de subsanar la deficiencia.

Mediante Power Bi las alertas se crean en un gráfico, que tendrá que ser creado como se ha explicado anteriormente, llamado KPI. Una vez éste se haya creado pulsando sobre 3 puntos suspensivos que aparecen cuando se arrastra el cursor por encima del gráfico y a continuación sobre un símbolo de campana.



Ahora ya solamente queda configurar la alarma para que ésta se lance cuando se reciba un valor determinado, que para este proyecto el valor será de 6,6 ya que contando el error en la medición de la sonda ($\pm 0,1$).



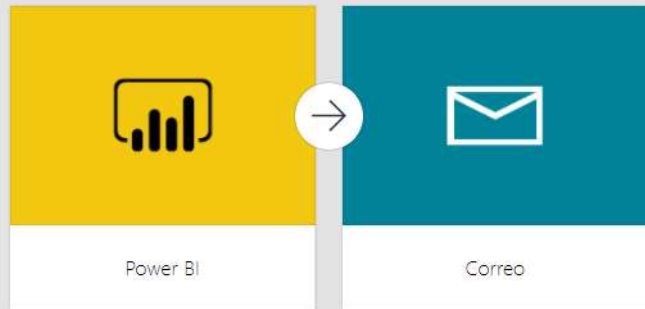
Configuración de Microsoft Flow

Como solamente se enviará un correo al usuario de la cuenta de Power Bi y pensando que el usuario final puede no ser el mismo que el usuario de la cuenta de Power Bi, se añadirá un servicio externo llamado Microsoft Flow que se servirá de la alerta de Power Bi como desencadenante de un flujo de acciones como por ejemplo del envío de un correo electrónico a otra persona entre otros.

Para agregar este servicio, solamente pulsando sobre la letra azul con texto: *“Use Microsoft Flow para desencadenar acciones adicionales”* (señalado en la imagen anterior)

Una vez pulsado en enlace, éste nos redirigirá a la página de Microsoft Flow dónde habrá que autenticarse con la misma cuenta que en Power Bi.

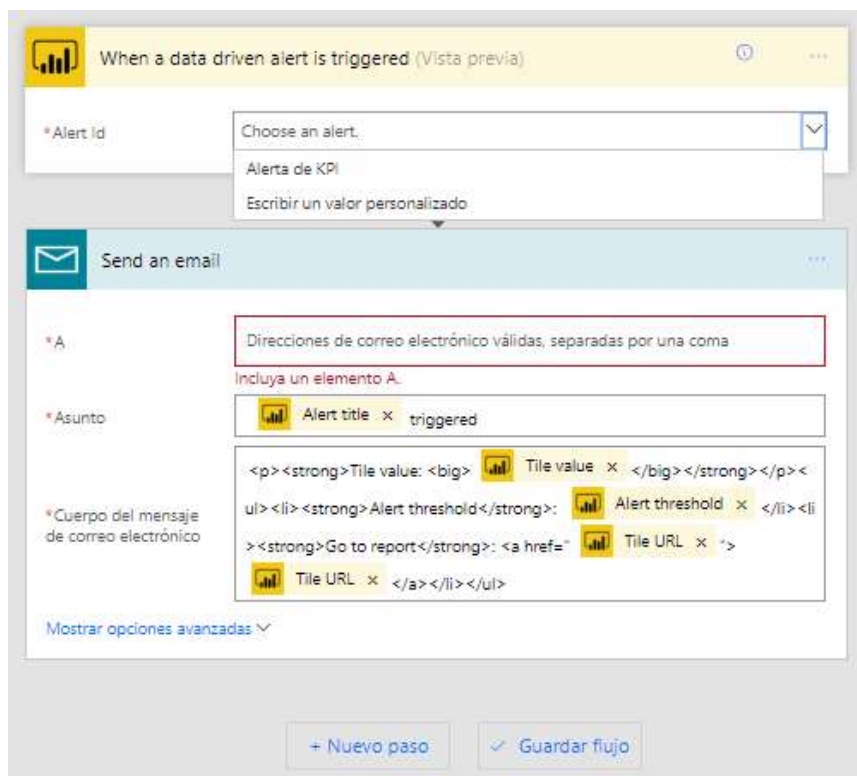
Enviar un correo electrónico al público cuando desencadene una alerta de datos de Power



Permite usar esta plantilla para enviar un correo electrónico a cualquier público cuando se desencadene una alerta controlada por datos de Power BI. El correo electrónico procederá de Microsoft Flow. Por ejemplo, puede enviar un aviso al equipo de soporte técnico cuando el volumen de incidentes sea superior a 100.

Usar esta plantilla

Seguidamente hay que pulsar sobre la plantilla que ofrece un servicio de entrega de correos al público cuando se desencadene una alerta de Power Bi.

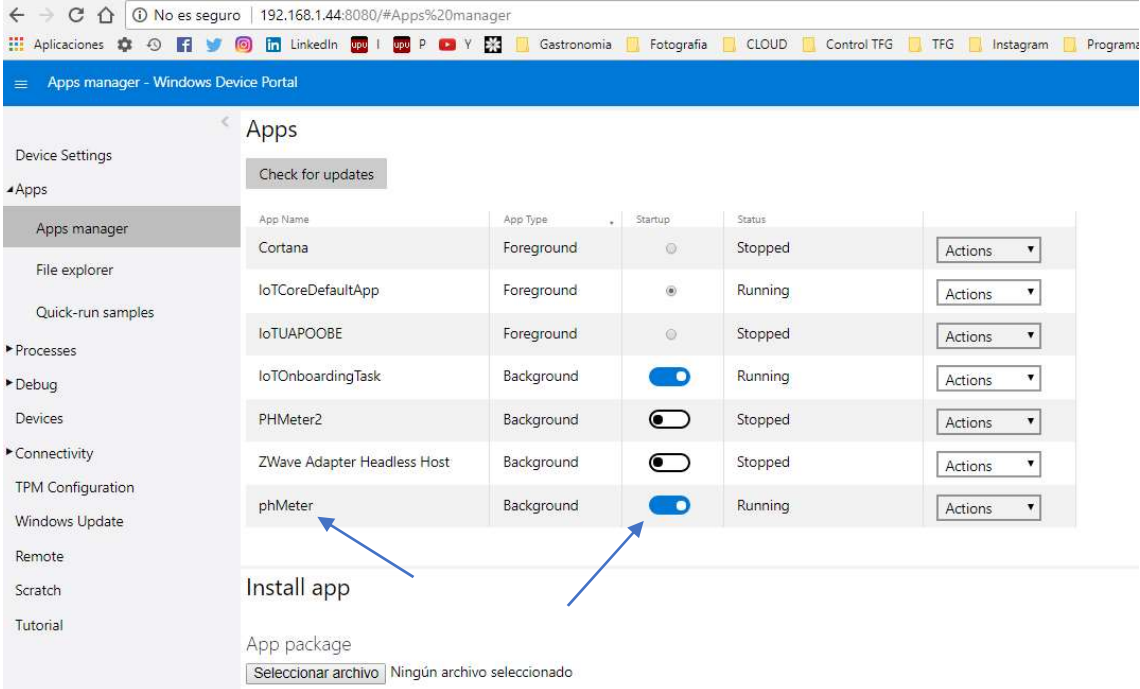
La imagen muestra la interfaz de configuración de un flujo de Microsoft Flow. El título del flujo es "When a data driven alert is triggered (Vista previa)". El primer paso es "Alert id" con un menú desplegable que muestra "Alerta de KPI" y "Escribir un valor personalizado". El segundo paso es "Send an email" con campos para "A" (direcciones de correo), "Asunto" (Alert title x triggered) y "Cuerpo del mensaje de correo electrónico" (conteniendo HTML con variables de Power BI como Tile value, Alert threshold y Tile URL). En la parte inferior hay botones para "+ Nuevo paso" y "Guardar flujo".

A continuación, habrá que seleccionar la alerta que se haya configurado en Power Bi, en este caso *“Alerta de KPI”* y solamente rellenar los campos que faltan.

Integrar “como servicio” el proyecto

Para que la Raspberry ejecute sin ayuda de Visual Studio y con el fin de hacer el proyecto autónomo, se debe integrar la solución “como servicio”. Esto significa que se ejecutará automáticamente en la rutina de arranque de la Raspberry.

Para ello solamente con acceder a la funcionalidad “Windows Device Portal” y en la pestaña de Apps únicamente hacer clic en la opción de Startup.



The screenshot displays the Windows Device Portal interface for managing apps. The left sidebar shows navigation options like Device Settings, Apps, File explorer, and Processes. The main area is titled 'Apps' and contains a table of installed applications. The 'phMeter' app is selected, and its 'Startup' toggle is turned on. Below the table, there is an 'Install app' section with an 'App package' field and a 'Seleccionar archivo' button.

App Name	App Type	Startup	Status	Actions
Cortana	Foreground	<input type="radio"/>	Stopped	Actions
IoTCoreDefaultApp	Foreground	<input checked="" type="radio"/>	Running	Actions
IoTUAPOOBE	Foreground	<input type="radio"/>	Stopped	Actions
IoTOnboardingTask	Background	<input checked="" type="checkbox"/>	Running	Actions
PHMeter2	Background	<input type="checkbox"/>	Stopped	Actions
ZWave Adapter Headless Host	Background	<input type="checkbox"/>	Stopped	Actions
phMeter	Background	<input checked="" type="checkbox"/>	Running	Actions

Install app

App package

Seleccionar archivo Ningún archivo seleccionado

Conclusiones

Concluido este trabajo se puede ver como se han cumplido los objetivos para los que fue creado como la automatización de una tarea periódica y manual.

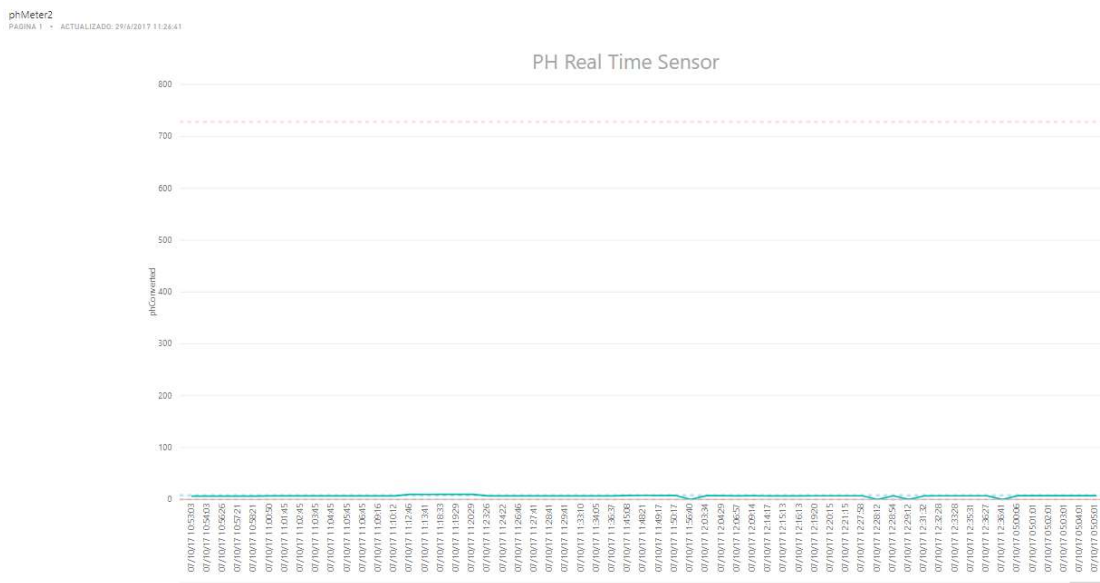
Mediante la Raspberry se puede ver que es una solución lo más económica posible, aunque se podría encarecer con una sonda profesional para obtener una precisión mayor y más exacta en el cálculo del pH.

En resumen, este sistema permite la automatización del pH de cualquier recinto con agua que sea necesario o que se quiera monitorizar siendo compatible este proyecto con cualquier dispositivo móvil gracias a la aplicación móvil de Power Bi o simplemente del enlace para visualizar el informe que se ha creado anteriormente.

Informe Power Bi

Para ver que se ha logrado el objetivo, se adjunta una captura de una simulación de medición del pH de agua, calibrador de pH (pH=7.0) y 3KNCL que se ha llevado a cabo para el correcto desarrollo del proyecto.

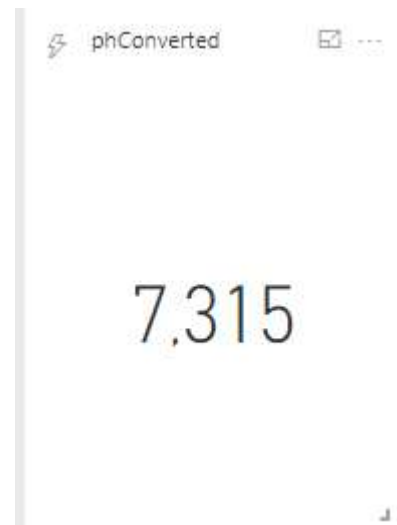
En el panel creado, se tiene la opción de la visualización en directo del valor de pH (se obtiene un histórico de valores):



Una tabla para la visualización de los datos de forma automática. Y, finalmente un visualizador en tiempo real únicamente del valor en función del tiempo y una tabla grafica de los valores de la ultima hora.

phConverted, EventEnqueuedUtcTime

7,30	07/10/17 05:07:01
phConverted	EventEnqueuedUtcTime
7,26	07/10/17 05:06:01
phConverted	EventEnqueuedUtcTime
7,26	07/10/17 05:05:01
phConverted	EventEnqueuedUtcTime
7,26	07/10/17 05:04:01
phConverted	EventEnqueuedUtcTime
7,26	07/10/17 05:03:01
phConverted	EventEnqueuedUtcTime
7,28	07/10/17 05:02:01
phConverted	EventEnqueuedUtcTime
7,33	07/10/17 05:01:01
phConverted	EventEnqueuedUtcTime
7,28	07/10/17 05:00:06
phConverted	EventEnqueuedUtcTime
0,00	07/10/17 12:36:41
phConverted	EventEnqueuedUtcTime
6,96	07/10/17 12:36:27
phConverted	EventEnqueuedUtcTime
7,01	07/10/17 12:35:31
phConverted	EventEnqueuedUtcTime
6,99	07/10/17 12:33:28
phConverted	EventEnqueuedUtcTime
6,97	07/10/17 12:32:28
phConverted	EventEnqueuedUtcTime
6,94	07/10/17 12:31:32
phConverted	EventEnqueuedUtcTime
0,00	07/10/17 12:29:12
phConverted	EventEnqueuedUtcTime
6,90	07/10/17 12:28:54
phConverted	EventEnqueuedUtcTime
0,00	07/10/17 12:28:12
phConverted	EventEnqueuedUtcTime
7,01	07/10/17 12:27:58
phConverted	EventEnqueuedUtcTime
7,01	07/10/17 12:21:15
phConverted	EventEnqueuedUtcTime



Cuando en un momento determinado el pH pueda descender de 6,6, el servicio que se ha configurado para que avise al usuario que deba corregir la anomalía de funcionamiento, se le enviará un correo electrónico avisando del error y un aviso visual en el informe de Power Bi.

MF Microsoft PowerApps and Flow 12:21

Alerta de KPI triggered PH METER PROVE EMAIL ALERT MICROSOFT FLOW
Para: victor.sanchez.ribes@gmail.com

Tile value:

- Alert threshold: 6.1
- Go to report: <https://app.powerbi.com/MobileRedirect.html?action=OpenTile&dashboardObjectId=5b9b3e45-5d30-4334-8df9-b609404512a6&tileObjectId=063e0795-b91e-4f61-be76-72b279fad7e7&OpenAppFromWindowsPCAndTablet=false&context=Flow>

If you'd like to unsubscribe and stop receiving these emails [click here](#).

The screenshot shows a Power BI dashboard with the following components:

- Navigation Pane (Left):** Includes 'Favoritos', 'Recientes', 'Aplicaciones', 'Compartido conmigo', 'Áreas de trabajo', and 'Mi área de trabajo' (selected). Under 'PANELES', 'Live Panel' is selected. Under 'INFORMES', 'KPI' is selected.
- Main Content Area:**
 - KPI Tile:** Displays the value '7,182'.
 - Data Table:** Titled 'phConverted, EventEnqueuedUtcTime'. It lists 20 rows of data with columns for 'phConverted' and 'EventEnqueuedUtcTime'. The values for 'phConverted' range from 7.18 down to 6.77.
 - Line Chart:** Titled 'Last Hour REAL TIME PH VALUE'. The y-axis ranges from 2 to 10. The chart shows a line that starts at approximately 9.5, remains relatively flat until about 11:40, then drops sharply to around 6.5 by 11:50, and continues to fluctuate slightly at that level.
- Notification Center (Right):** Titled 'CENTRO DE NOTIFICACIONES' and 'Todas las notificaciones'. It contains two alerts:
 - Alerta de KPI hace 7 minutos:** 'Alerta de KPI en Live Panel es 0.032827016, que se encuentra por debajo del umbral de 6.1. Ir al icono'.
 - Alerta de KPI hace una hora:** 'Alerta de KPI en Live Panel es 0.00736572297, que se encuentra por debajo del umbral de 6.1. Ir al icono'.

Posibles Mejoras

También, más allá, es posible conectar otros dispositivos para mejorar la interfaz y poder llegar a controlar y visualizar otras variables ambientales. Como por ejemplo un sensor de temperatura y mediante del potencial dado por la sonda para el cálculo con mayor exactitud del pH mediante la ecuación de Nerst.

Además, es posible, teniendo en cuenta que el conversor MCP3008 tiene 8 canales, la conexión de nuevos sensores para la “profesionalización” del proyecto. Es decir, aplicar la domótica.

Un ejemplo podría ser la conexión de un sensor de humedad como el DHT11, un sensor de intensidad lumínica LDR,

un sensor de temperatura LM35 o hasta un sensor de presión atmosférica.



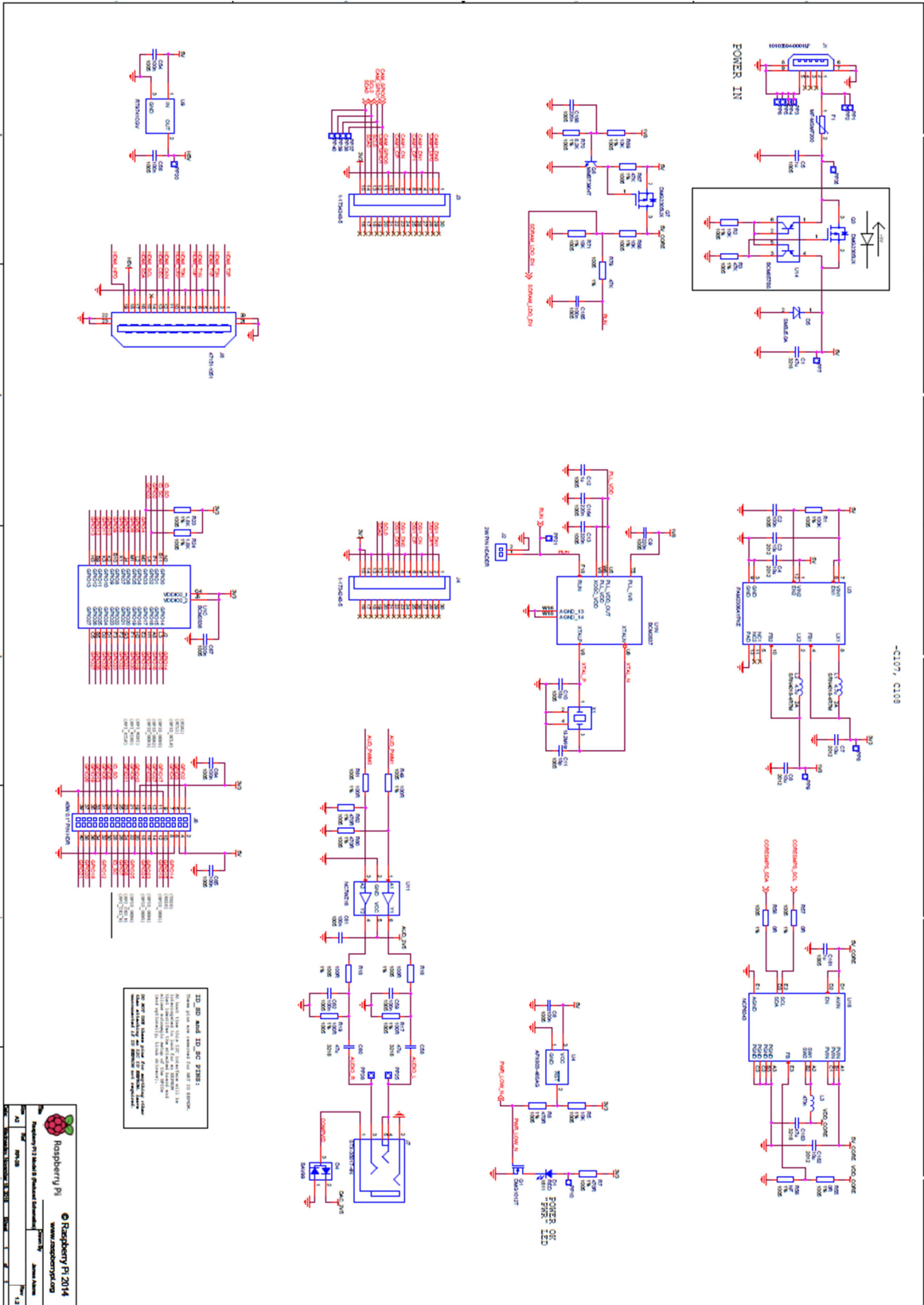
Bibliografía

- <http://www.prometec.net/bus-spi/>
- <http://www.prometec.net/bus-i2c/>
- <https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf>
- http://tienda.bricogeek.com/home/581-sensor-analogico-de-ph.html?gclid=CNuZ8a_m09MCFXEz0wodzCcFXg
- <https://developer.microsoft.com/es-es/windows/iot/Docs/GetStarted/noobs/getstartedstep1>
- <https://developer.microsoft.com/es-es/windows/iot/Docs/GetStarted/noobs/getstartedstep2>
- <https://developer.microsoft.com/es-es/windows/iot/Docs/GetStarted/noobs/getstartedstep3>
- <https://developer.microsoft.com/es-es/windows/iot/Docs/GetStarted/noobs/getstartedstep4>
- <https://developer.microsoft.com/en-us/windows/iot/docs>
- <https://docs.microsoft.com/es-es/windows/uwp/get-started/universal-application-platform-guide>
- <http://www.diarioelectronicohoy.com/blog/introduccion-al-i2c-bus>
- <http://tienda.bricogeek.com/raspberry-pi/750-mcp3008-conversor-adc-8-bits.html>
- <http://bot-boss.com/lectura-senales-analogicas-raspberry-pi-2-mcp3008/>
- <http://bot-boss.com/habilitar-spi-raspberry-pi-2/>
- <https://www.quora.com/How-do-I-connect-pH-meter-with-arduino-and-get-the-readings-on-given-time>
- <https://learn.adafruit.com/reading-a-analog-in-and-controlling-audio-volume-with-the-raspberry-pi/script>
- <https://aprendiendoarduino.wordpress.com/2016/11/13/bus-spi/>
- <http://www.instructables.com/id/Wiring-up-a-MCP3008-ADC-to-a-Raspberry-Pi-model-B-/>
- <https://pimylifeup.com/raspberry-pi-adc/>
- <https://learn.adafruit.com/reading-a-analog-in-and-controlling-audio-volume-with-the-raspberry-pi?view=all>
- https://www.youtube.com/watch?v=oW0mTWFck_c
- <https://www.youtube.com/watch?v=QHIWV904jNw>
- <https://www.raspberrypi.org/learning/physical-computing-with-python/analogue/>
- <http://wiringpi.com/reference/spi-library/>
- <http://thinglabs.io/sending-telemetry/>
- <http://ww1.microchip.com/downloads/en/DeviceDoc/21295d.pdf>
- <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-get-started-physical>
- <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-raspberry-pi-kit-node-get-started>
- <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi>

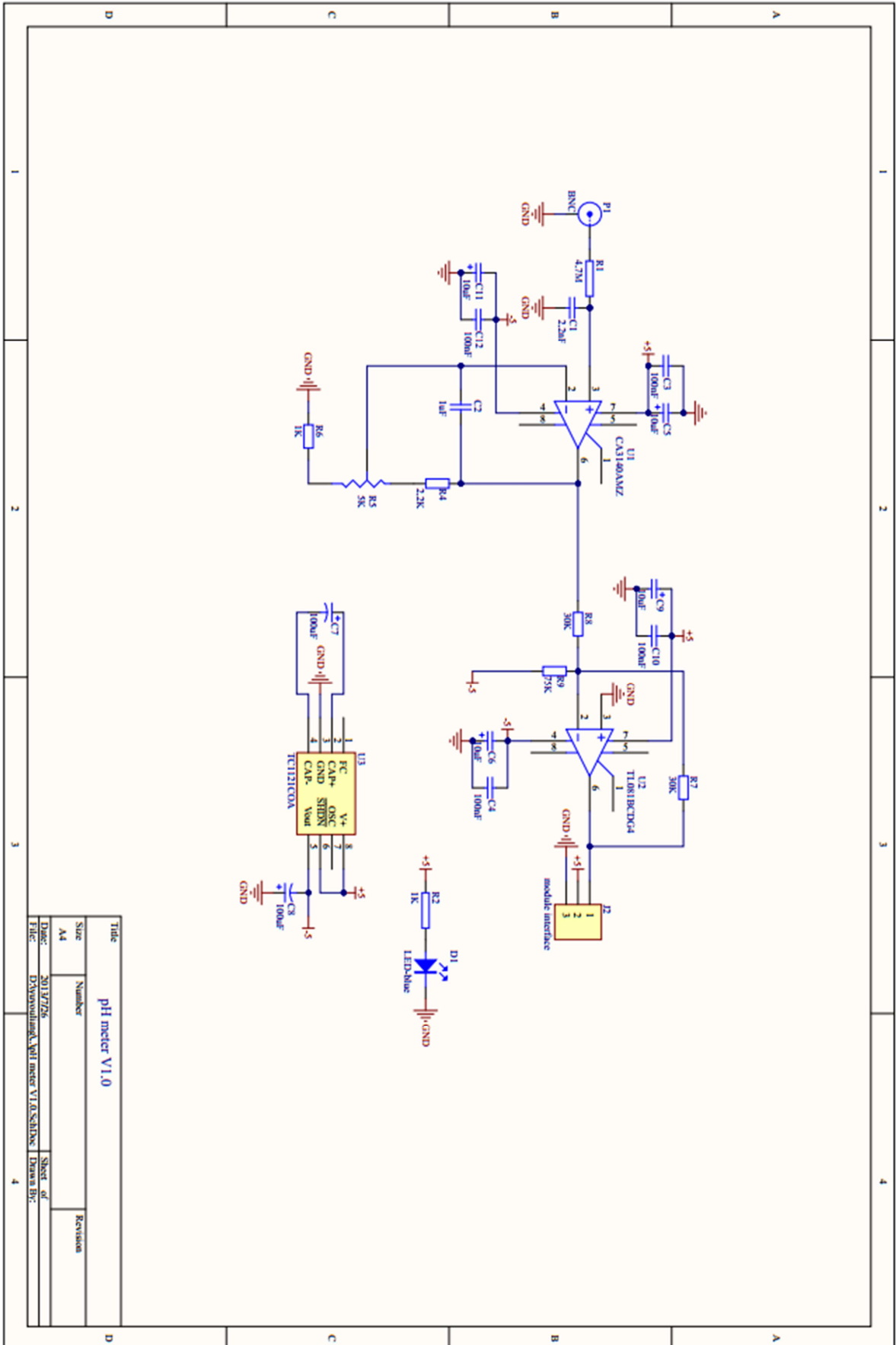
- <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-csharp-csharp-process-d2c>
- <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-node-node-getstarted>
- <https://blogs.msdn.microsoft.com/iot/2016/01/26/using-power-bi-to-visualize-sensor-data-from-windows-10-iot-core/>
- <https://powerbi.microsoft.com/en-us/blog/monitor-your-iot-sensors-using-power-bi/>
- <https://developer.microsoft.com/en-us/windows/iot/samples/azuredatauploader>
- <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-node-node-getstarted#create-an-iot-hub/>
- <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-common-deployment-errors#noregisteredproviderfound>
- <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-portal>
- <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-web-apps>
- <https://developer.microsoft.com/en-us/windows/iot/docs/pinmappingsrpi>
- https://microsoft.hackster.io/en-US/brane/003-how-to-stream-device-data-to-the-azure-iot-hub-d99807?ref=channel&ref_id=4087_trending___&offset=377
- <https://developer.microsoft.com/en-us/windows/iot/samples/helloworld>
- <http://thinglabs.io/workshop/thingy-4-windows/>
- <http://thinglabs.io/workshop/thingy-4-windows/setup-azure-iot-hub/>
- <http://thinglabs.io/workshop/thingy-4-windows/sending-d2c-messages/>
- <http://thinglabs.io/workshop/thingy-4-windows/storing-displaying-data/>
- <http://thinglabs.io/labs/edison/grove/d2c/>
- <https://github.com/ThingLabsIo/Thingy4Windows>
- <https://channel9.msdn.com/Events/Microsoft-Azure/AzureCon-2015/ACON305>
- <https://github.com/Azure/azure-iot-sdk-csharp/tree/master/tools/DeviceExplorer>
- https://github.com/Azure/azure-iot-sdk-csharp/blob/master/doc/setup_iothub.md
- <https://github.com/Azure/azure-iot-sdk-csharp>
- https://github.com/Azure/azure-iot-sdk-csharp/blob/master/doc/manage_iot_hub.md
- <https://blog.falafel.com/mcp3008-analog-to-digital-conversion/>
- https://github.com/falafelsoftware/SPI_ADC_Sample

Anexo

Raspberry Pi 2 Model B (schematics)

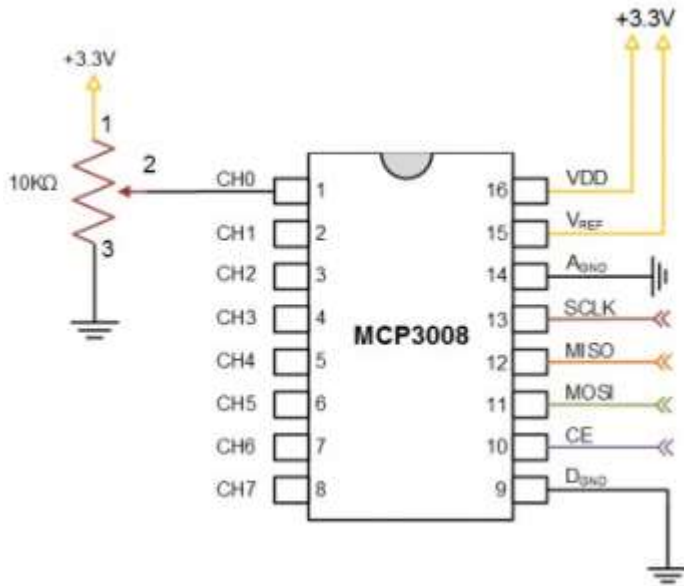


pH Meter v1.0



Title		Revision	
pH meter V1.0			
Size	Number		
A4			
Date:	2017/7/26	Sheet of	
File:	D:\yovon\img\pH meter V1.0.schDoc	Drawn By:	

MCP3008



3.0 PIN DESCRIPTIONS

The descriptions of the pins are listed in [Table 3-1](#). Additional descriptions of the device pins follows.

TABLE 3-1: PIN FUNCTION TABLE

MCP3004	MCP3008	Symbol	Description
PDIP, SOIC, TSSOP	PDIP, SOIC		
1	1	CH0	Analog Input
2	2	CH1	Analog Input
3	3	CH2	Analog Input
4	4	CH3	Analog Input
–	5	CH4	Analog Input
–	6	CH5	Analog Input
–	7	CH6	Analog Input
–	8	CH7	Analog Input
7	9	DGND	Digital Ground
8	10	$\overline{\text{CS}}/\text{SHDN}$	Chip Select/Shutdown Input
9	11	D _{IN}	Serial Data In
10	12	D _{OUT}	Serial Data Out
11	13	CLK	Serial Clock
12	14	AGND	Analog Ground
13	15	V _{REF}	Reference Voltage Input
14	16	V _{DD}	+2.7V to 5.5V Power Supply
5,6	–	NC	No Connection

Adaptador Wifi TP-Link TL-WN725N

CARACTERÍSTICAS DE HARDWARE

Interfaz	USB 2.0
Antenas	Antena interna
LED	Estatus
Peso	0.07 onzas / 2.1 gramos (Sin empaque)
Dimensiones	0.73x0.59x0.28in.(18.6x15x7.1mm)

CARACTERÍSTICAS INALÁMBRICAS

Estándares Inalámbricos	IEEE 802.11b, IEEE 802.11g, IEEE 802.11n
Frecuencia	2.400-2.4835GHz
Tasa de Señal	11b: Hasta 11Mbps (dinámico) 11g: Hasta 54Mbps (dinámico) 11n: Hasta 150Mbps (dinámico)
Sensibilidad de Recepción	130M: -68dBm@10% PER 108M: -68dBm@10% PER 54M: -68dBm@10% PER 11M: -85dBm@8% PER 6M: -88dBm@10% PER 1M: -90dBm@8% PER
Potencia de Transmisión	<20dBm
Modos Inalámbricos	Ad-Hoc / Modo de infraestructura
Seguridad Inalámbrica	Soporta WEP, WPA/WPA2, WPA-PSK/WPA2-PSK (TKIP/AES) de 64/128, soporta IEEE 802.1X
Tecnología de Modulación	DBPSK, DQPSK, CCK, OFDM, 16-QAM, 64-QAM

OTROS

Certificaciones	CE, FCC, IC, RoHS
Contenido del Paquete	Adaptador USB Nano Inalámbrico N de 150Mbps, TL-WN725N CD de Recursos Guía de Rápida Instalación
Requisitos del sistema	Windows 8(32/64bits), Windows 7(32/64bits), Windows Vista(32/64bits), Windows XP(32/64bits)
Factores Ambientales	Temperatura de Operación: 0°C~40°C (32°F~104°F) Temperatura de Almacenamiento: -40°C~70°C (-40°F~158°F) Humedad de Operación: 10%~90% sin condensación Humedad de Almacenamiento: 5%~90% sin condensación

Ensamblado Completo

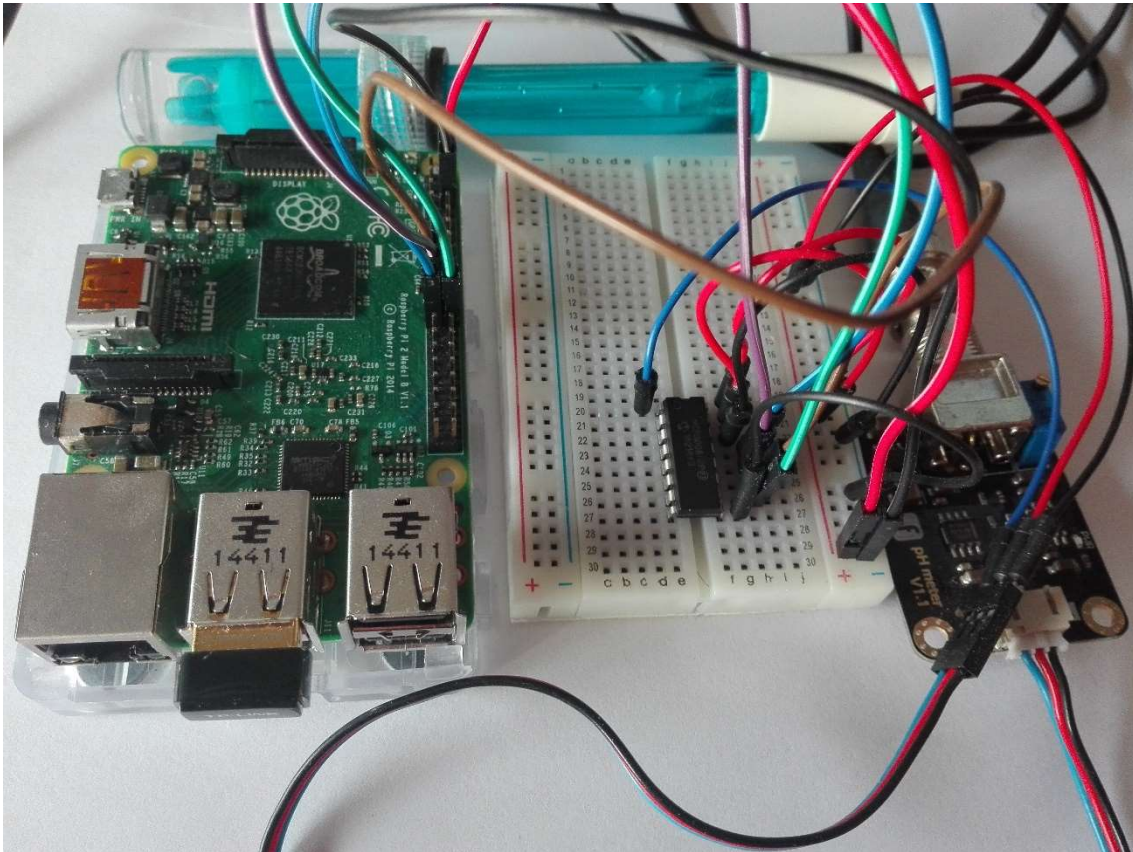
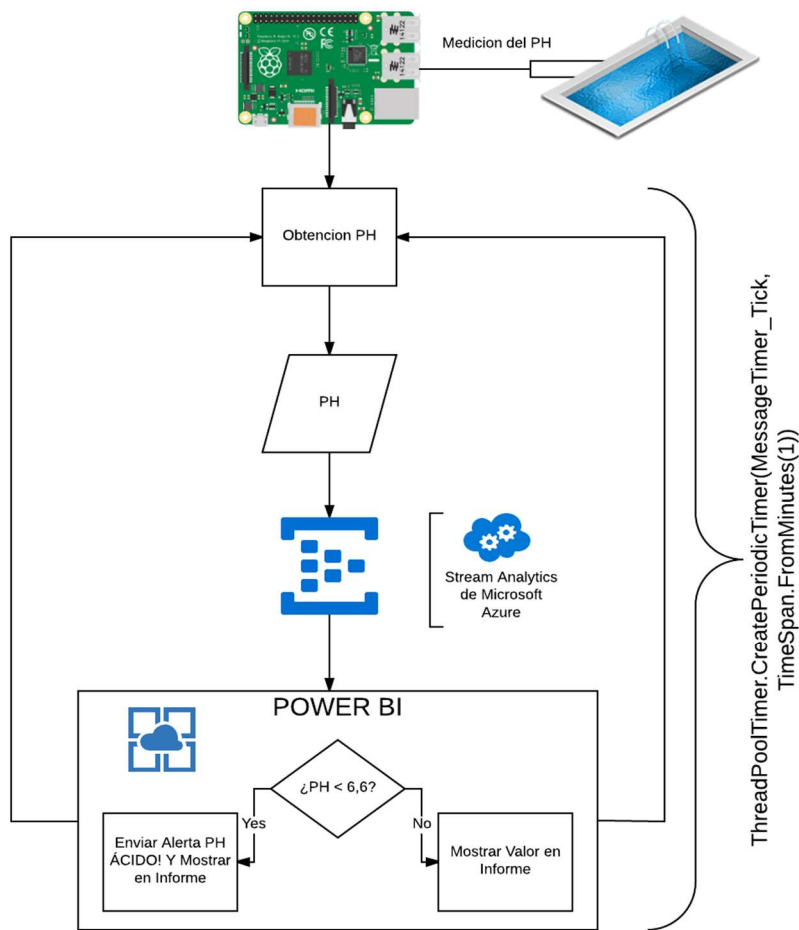


Diagrama de funcionamiento



En este diagrama se muestra el funcionamiento del proyecto a realizar.

Primeramente, se obtendrá el valor del potencial eléctrico proporcionado por la Raspberry Pi. Será transformado a continuación para conseguir el pH y una vez se consiga el pH será transmitido a Azure por el que éste proporcionará un servicio de “streaming” por el que se utilizará un visor, Power Bi, para realizar las operaciones y la visualización de dicho pH, en tiempo real. Esta operación se repetirá cada minuto según indica la llave localizada en la parte derecha (*ThreadPoolTimer.CreatePeriodicTimer(MessageTimer_Tick, TimeSpan.FromMinutes(1))*).

Código fuente

```
using System;
using System.Text;
using Windows.ApplicationModel.Background;
using System.Threading.Tasks;
using Microsoft.Azure.Devices.Client;
using Windows.System.Threading;
using Windows.Devices.Enumeration;
using Windows.Devices.Spi;
using Newtonsoft.Json;
using System.Diagnostics;
using Windows.System;

namespace phMeter
{
    public sealed class StartupTask : IBackgroundTask
    {
        //*****
        //
        //          CONSTANTS & VARIABLES
        //
        //*****
        private BackgroundTaskDeferral deferral; //Creacion de un deferral para prevenir la
terminacion repentina
        private DeviceClient deviceClient;
        private ThreadPoolTimer messageTimer;
        private float ph, voltios = 0.0f;
        private SpiDevice mcp3008;
        private float offset = 0.12f;
        private const string IOT_HUB_CONN_STRING = "HostName=phMeter-iotHub.azure-
devices.net;DeviceId=PHMeter;SharedAccessKey=va+C4Uxyy0s/eQP1fWsp7Q/kbv+7/01ULyxafIHnVA=";

        public void Run(IBackgroundTaskInstance taskInstance)
        {
            deferral = taskInstance.GetDeferral();

            //Instanciamos el dispositivo cliente Azure
            deviceClient = DeviceClient.CreateFromConnectionString(IOT_HUB_CONN_STRING);

            //Iniciamos la comunicacion con el sensor mediante SPI
            busSpi();

            //Enviamos un mensaje a Azure cada 1 Minuto
            messageTimer = ThreadPoolTimer.CreatePeriodicTimer(MessageTimer_Tick,
TimeSpan.FromMinutes(1));
        }

        //*****
        //
        //          CONEXION AL BUS SPI
        //
        //*****
        private async void busSpi()
        {
            try
            {
                var spiSettings = new SpiConnectionSettings(0);
                spiSettings.ClockFrequency = 1350000;
                spiSettings.Mode = SpiMode.Mode0;

                string spiQuery = SpiDevice.GetDeviceSelector("SPI0");
                var deviceInfo = await DeviceInformation.FindAllAsync(spiQuery);
                if (deviceInfo != null && deviceInfo.Count == 1)
                {
                    mcp3008 = await SpiDevice.FromIdAsync(deviceInfo[0].Id, spiSettings);
                }
                else
                {
                    Debug.WriteLine("DISPOSITIVO SPI NO ENCONTRADO!!");
                }
                //led parpadenado (OPCIONAL)
            }
            catch (Exception ex)
            {
            }
        }
    }
}
```

```

        {
            Debug.WriteLine("!!!! SPI NOT FOUND: " + ex.Message);
        }
    }
    /*
    * https://www.dfrobot.com/wiki/index.php/PH_meter_V1.1_SKU:SEN0161
    */

    //*****
    //
    //                OBTENCION DEL PH
    //
    //*****
    private float obt_ph()
    {
        try
        {
            //1 - single ended
            // Canal 0 = 10000000 = 0x80
            // 3 Byte is empty
            var transmitBuffer = new Byte[3] { 1, 0x80, 0x00 };
            var receiveBuffer = new Byte[3];

            mcp3008.TransferFullDuplex(transmitBuffer, receiveBuffer);

            var result = ((receiveBuffer[1] & 3) << 8) + receiveBuffer[2];

            voltios = (result * 5.0f) / 1024;
            ph = 3.50f * voltios+offset; //Offset visto entre la diferencia del ph de un
            liquido calibrador = 7.0 y la lectura dada por el dispositivo la 1ª vez
        }

        catch (Exception ex)
        {
            Debug.WriteLine("ERROR OBTENCION DE DATOS!");
            Debug.WriteLine("!!!! WRONG CONVERSION: " + ex.Message);
        }

        return ph;
    }

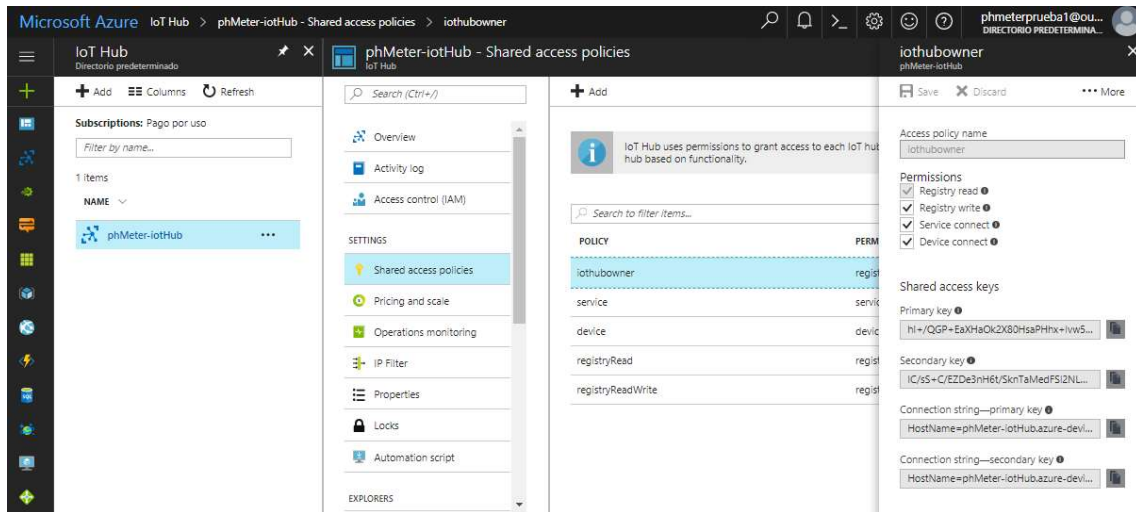
    //Funcion que enviara a Azure el ph (se invocara cada 1 minuto)
    //Llamamos a obt_ph que nos dara el ph
    //Enviaremos el ph a Azure
    private async void MessageTimer_Tick(ThreadPoolTimer timer)
    {
        obt_ph();
        await SendMessageToIotHubAsync("ph", ph);
    }

    //*****
    //
    //                SUBIR DATOS A AZURE
    //
    //*****
    //Metodo para enviar el ph al CLOUD
    private async Task SendMessageToIotHubAsync(string sensorType, float phValue)
    {
        try
        {
            var phConverted = phValue;
            //Prueba de Serializacion
            var telemetry = new
            {
                sensorType,
                phConverted,
            };
            Debug.WriteLine(phConverted);
            var messageString = JsonConvert.SerializeObject(telemetry);
            var message = new Message(Encoding.UTF8.GetBytes(messageString));
        }
    }
}

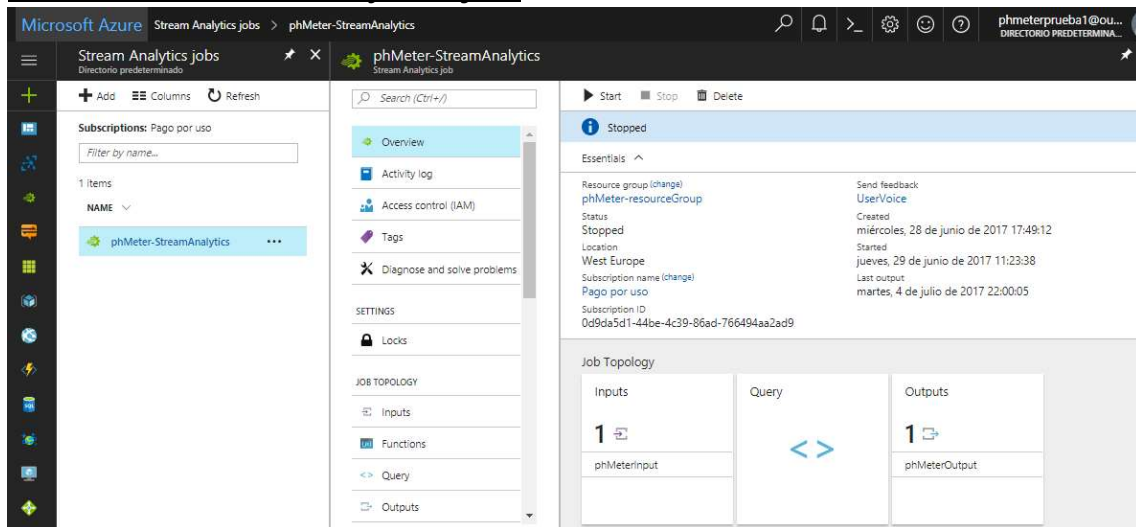
```

```
        await deviceClient.SendEventAsync(message);
    }
    catch (Exception ex)
    {
        Debug.WriteLine("!!!! " + ex.Message);
    }
}
}
```

Azure IOT HUB



Azure Stream Analytics job



Stream Analytics Job

Input

The screenshot shows the 'Input details' configuration page for an IoT hub input in a Stream Analytics job. The left pane shows a table of inputs:

NAME	SOURCE TYPE	SOURCE
phMeterInput	Stream	IoT hub

The right pane shows the configuration for 'phMeterInput':

- Import option: Provide IoT hub settings manually
- IoT hub: phMeter-iotHub
- Endpoint: Messaging
- Shared access policy name: iotHubowner
- Shared access policy key: [Redacted]
- Consumer group: phmeterconsumergroup
- Event serialization format: JSON
- Encoding: UTF-8

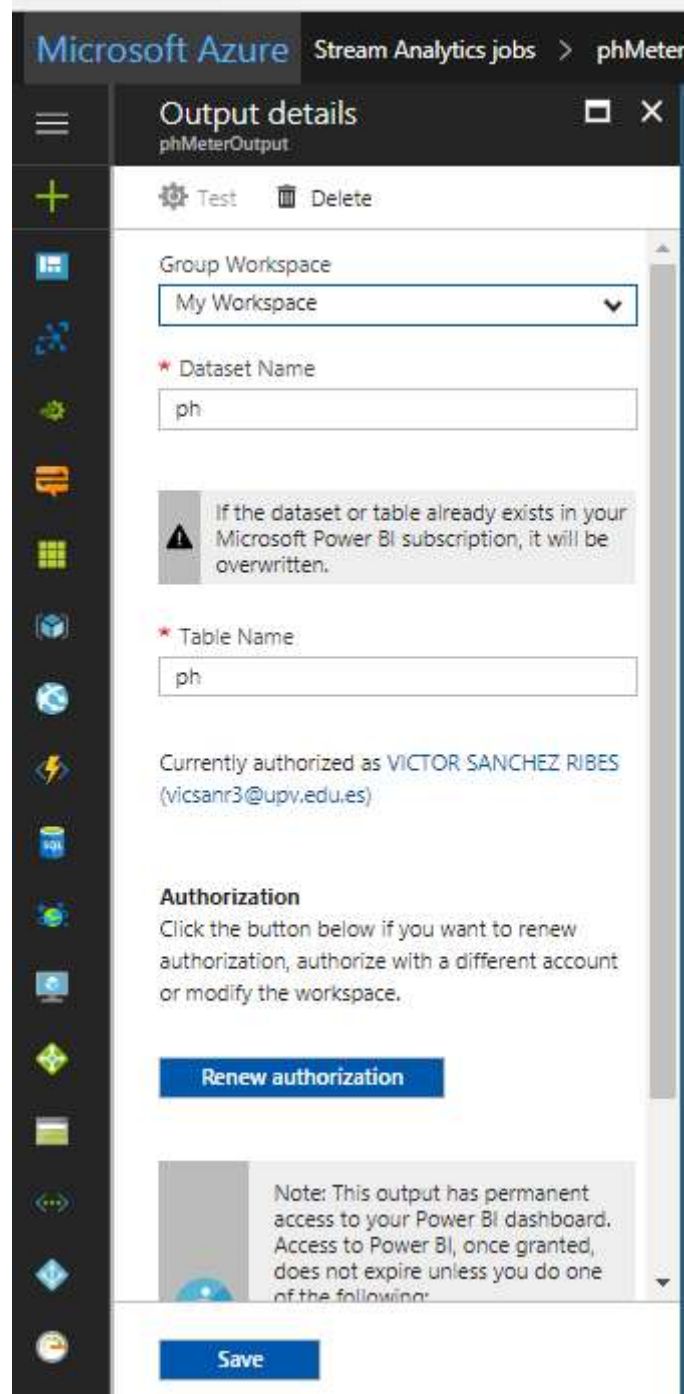
Consulta (Query)

The screenshot shows the 'Query' configuration page. On the left, the query graph shows one input ('phMeterInput') and one output ('phMeterOutput'). The query text is:

```
1 SELECT
2 *
3 INTO
4 phMeterOutput
5 FROM
6 phMeterInput
```

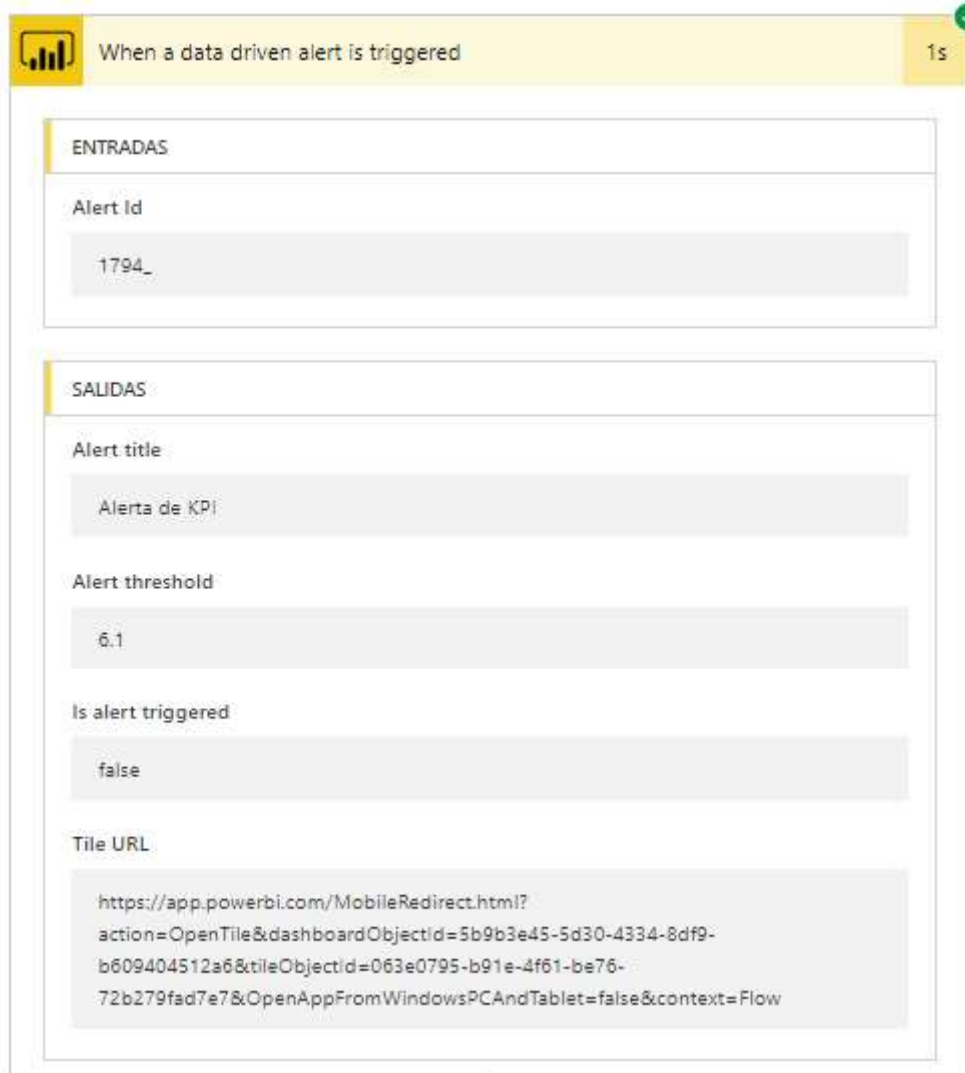
There is a link for help: "Need help with your query? Check out some of the most common Stream Analytics query patterns [here](#)."

Output



Alerta Microsoft Flow en un entorno real

Tipo de alerta desencadenada



The screenshot displays a Microsoft Flow alert configuration window titled "When a data driven alert is triggered" with a 1s timer. It is divided into two sections: "ENTRADAS" (Inputs) and "SALIDAS" (Outputs).

ENTRADAS

- Alert Id: 1794_

SALIDAS

- Alert title: Alerta de KPI
- Alert threshold: 6.1
- Is alert triggered: false
- Tile URL: <https://app.powerbi.com/MobileRedirect.html?action=OpenTile&dashboardObjectId=5b9b3e45-5d30-4334-8df9-b609404512a6&tileObjectId=063e0795-b91e-4f61-be76-72b279fad7e7&OpenAppFromWindowsPCAndTablet=false&context=Flow>

Destinatario alertado e información enviada

Send an email 1s

ENTRADAS

A

victor.sanchez.ribes@gmail.com

Asunto

Alerta de KPI triggered PH METER PROVE EMAIL ALERT MICROSOFT FLOW

Cuerpo del mensaje de correo electrónico

```
<p><strong>Tile value: <big></big></strong></p><ul><li><strong>Alert threshold</strong>: 6.1</li><li><strong>Go to report</strong>: <a href="https://app.powerbi.com/MobileRedirect.html?action=OpenTile&dashboardObjectId=5b9b3e45-5d30-4334-8df9-b609404512a6&tileObjectId=063e0795-b91e-4f61-be76-72b279fad7e7&OpenAppFromWindowsPCAndTablet=false&context=Flow">https://
```

••• Mas

SALIDAS

Encabezados

```
{
  "X-Message-Id": "ijZBrHwwRx2FfGi832Gvqg",
  "X-Frame-Options": "DENY",
  "Access-Control-Allow-Origin": "https://sendgrid.api-docs.io",
  "Access-Control-Allow-Methods": "POST",
  "Access-Control-Allow-Headers": "Authorization, Content-Type, On-behalf-of, x-sg-elas-acl",
  "Access-Control-Allow-Credentials": true
}
```

Cuerpo

```
{}
```