



CCADET

CENTRO DE CIENCIAS APLICADAS Y DESARROLLO
TECNOLÓGICO
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Inclinómetro con conectividad al internet de las cosas

**Benjamín Valera Orozco
Gerardo Antonio Ruiz Botello**

Marzo 2018

Proyecto Interno

Financiamiento interno

Inclinómetro con conectividad al internet de las cosas

Benjamín Valera Orozco y Gerardo Antonio Ruiz Botello

Centro de Ciencias Aplicadas y Desarrollo Tecnológico
Universidad Nacional Autónoma de México, Circuito Exterior S/N, Ciudad
Universitaria AP 70-186, C.P. 04510, México D.F. México

Resumen

En el presente informe se describe el diseño e implementación de un instrumento para la medición de tres ángulos de inclinación con conectividad al internet de las cosas. Con este concepto es posible registrar, de manera remota e inalámbrica, los ángulos de inclinación que reporta el instrumento en la nube de datos. La principal ventaja que se obtiene al crear instrumentación con computación en la nube es la enorme posibilidad de servicios que se ofrecen actualmente por compañías de conectividad y procesamiento en la nube. Entre otros los servicios que se disponen en la nube son: extensa conectividad geográfica de los dispositivos mediante estaciones de radiofrecuencia, resguardo de información en la nube, acceso a la información desde prácticamente cualquier sitio en donde exista conexión a internet, intercambio de información entre una gran variedad de aplicaciones y procesamiento de datos en ambientes amigables de programación en donde no se requieren conocimientos especializados en computación. Se presenta el diseño electrónico del instrumento y la configuración en los sitios de los proveedores de computación en la nube que proporcionan como resultado final una hoja de cálculo que se actualiza automáticamente con datos de inclinación en tres ejes y que adicionalmente puede compartirse entre usuarios de Google.

Indice

Nomenclatura	3
Introducción	4
Definición del problema	4
Entorno actual	4
Descripción del problema a resolver y motivación	4
Relevancia y limitaciones	4
Relación con otras áreas	5
Método	5
Objetivos y resultados esperados	5
Organización del informe	6
1 Inclínómetro con conectividad al internet de las cosas	7
1.1 Descripción general	7
1.2 Sistema electrónico	7
1.2.1 Kit de desarrollo NXTIoT	8
1.2.2 Inclínómetro triaxial de precisión ADIS16210	10
1.2.3 Convertidor lógico bidireccional	11
1.2.4 Algoritmo de medición	11
1.3 Proveedores de servicios en la nube	12
1.3.1 IoTnet	13
1.3.2 SigFox	13
1.3.3 Google	19
2 Resultados y discusión	25
2.1 Resultados	25
2.1.1 Prototipo desarrollado	25
2.1.2 Hoja de cálculo	25
2.2 Discusión y trabajo a futuro	27
2.3 Especificaciones del sistema electrónico	27
2.4 Requisitos de operación	28
2.5 Guía para la detección de fallas	28
Conclusiones	29
Agradecimientos	30
Anexo A Diagrama esquemático	31
Anexo B Lista de partes	32
Anexo C Código fuente Arduino	33
Anexo D Código fuente de la aplicación WEB	42
Referencias bibliográficas	43

Nomenclatura

Símbolo	Significado
ADIS16210	Circuito electrónico para la medición de inclinación en tres ejes.
App	Aplicación informática.
Arduino	Compañía de un proyecto de software y hardware libres.
AT	Estándar abierto de comandos para configurar equipos periféricos de cómputo.
Bps	Bits per second, Bits por segundo.
Callback	Código ejecutable que es pasado como argumento a otro código.
Drive	Unidad física en donde se escriben o leen datos de computadora.
Full dúplex	Transmisión simultánea bidireccional entre equipos transmisor y receptor.
Google	Proveedor de servicios de informática.
HTTP	The Hypertext Transfer Protocol, Protocolo de transferencia de hipertexto.
HV	High Voltage, Voltaje alto.
ID	Identification, Número de identificación.
IDE	Integrated Development Environment, Ambiente integrado de desarrollo.
IoT	Internet of Things, Internet de las cosas.
Jumper	En electrónica, dispositivo de conmutación.
Kit	Conjunto de herramientas.
LED	Light Emitting Diode, Diodo emisor de luz.
LV	Low Voltage, Voltaje bajo.
MEMS	Micro Electro Mechanical Systems, Sistemas micro electro mecánicos.
NXTIoT	Tarjeta electrónica de desarrollo para el IoT.
RF	Radio frecuencia.
Script	Conjunto de instrucciones.
Sigfox	Proveedor de servicios de conectividad en la nube.
SNI	Server Name Identification, Nombre de identificación de un servidor.
SPI	Serial Peripheral Interface, Interface serial de comunicaciones entre dispositivos electrónicos.
TTL	Transistor to Transistor Logic, Tecnología de circuitos digitales.
URL	Uniform Resource Locator, secuencia de caracteres que permite localizar un recurso en una red de computadoras.
USART	Universal Synchronous and Asynchronous Receiver Transmitter, Receptor y transmisor universales síncronos y asíncronos.
USB	Universal Serial Bus, Canal universal serial.
Web	Red de computadoras

Introducción

Definición del problema

En el Internet de las cosas (por sus siglas en inglés, IoT, Internet of Things) convergen una serie de tecnologías de la información con el objetivo de dotar de conectividad a los objetos cotidianos con Internet. Entre las tecnologías de la información se pueden mencionar las siguientes: Radio bases, sitios web, dispositivos electrónicos de control y de radiofrecuencia. Por otra parte, los objetos cotidianos más viables para ser conectados a Internet son los siguientes: Aparatos electrodomésticos, automóviles y alarmas. El concepto más difundido del IoT está relacionado con la automatización de los objetos cotidianos para simplificar la vida de los seres humanos. En este concepto, las cosas, dotadas con microprocesadores y conectividad a Internet, serían capaces de tomar decisiones por ellas mismas, aligerando la dependencia que los seres humanos dedican al funcionamiento de tales cosas. No obstante, el campo de aplicación del IoT se ha incrementado notablemente alcanzando nuevas aplicaciones en diversos sectores como: el monitoreo ambiental, sistemas inteligentes de transporte, diagnóstico médico de pacientes y el monitoreo de infraestructura y obras civiles, entre muchos otros.

Entorno actual

El principal antecedente del presente proyecto es el desarrollo de un inclinómetro inalámbrico reportado en (Ruiz *et al*, 2017). En este trabajo se describe un aparato para la medición de inclinación que consta de dos radio-bases, una transmisora y otra receptora. La radio-base transmisora, mide y transmite tres ángulos de inclinación. La radio-base receptora, registra la información de los ángulos de inclinación en una computadora. La aplicación de este instrumento fue en el monitoreo de la inclinación del Tláloc de Coatlinchan. El alcance del enlace de radiofrecuencia conseguido al desarrollar este aparato fue de 20m.

Descripción del problema a resolver y motivación

En el presente informe se describe el desarrollo de un inclinómetro con conectividad al IoT con las siguientes características:

- En este nuevo enfoque el enlace de radiofrecuencia debe ser más robusto, es decir, menos susceptible a los errores propios de recepción en los sistemas de comunicaciones.
- El nuevo instrumento para la medición de inclinación debe extender más allá de 20m el alcance del enlace de radiofrecuencia.
- El sistema deberá contemplar la recepción de la información por diversos usuarios validados por el encargado de su administración.
- El sistema debe ser altamente autónomo, es decir, debe operar por intervalos largos de tiempo (típicamente 6 meses) sin la intervención humana.

El enfoque hacia el IoT cubre eficientemente las características planteadas para el nuevo diseño de inclinómetro.

Relevancia y limitaciones

La principal relevancia del proyecto radica en que utilizando las tecnologías del IoT se garantiza un rendimiento altamente confiable y robusto en cuanto a las telecomunicaciones y en el aseguramiento de la información. Por otra parte, se fomenta la creación de infraestructura de desarrollo propio que puede ser ofrecida como una transferencia tecnológica.

El alcance del presente proyecto es el desarrollo de un prototipo funcional que pueda ser utilizado confiablemente para la medición de ángulos de inclinación en aplicaciones de monitoreo estructural de infraestructuras civiles.

La potencial limitación del prototipo radica en los costos que impliquen el uso de las tecnologías del IoT que se describen en el presente trabajo. Al día de hoy, la operación del inclinómetro con IoT no ha ocasionado ningún pago por renta de los servicios de IoT. No obstante, no se descarta la idea de que en un futuro se deban abonar pagos por renta de estos servicios.

Relación con otras áreas

Existe una gran relación con las áreas de sistemas de comunicaciones, instrumentación, programación de computadoras, microcontroladores y microprocesadores.

Método

En este informe se describe el desarrollo hardware y software para implementar un inclinómetro con conectividad al IoT. El esquema general se presenta en la figura 1.

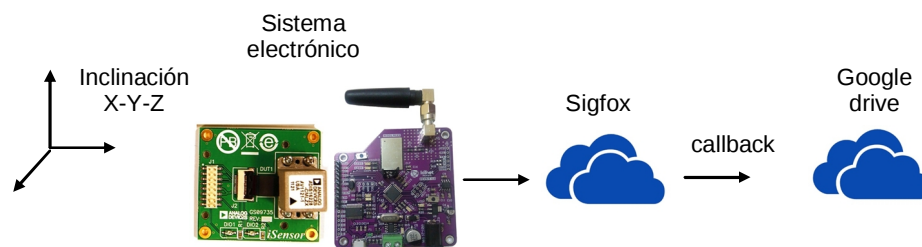


Figura 1. Método a emplear.

En el esquema de la figura 1 un sistema electrónico, con base en el kit de desarrollo NXTIoT, se posiciona sobre la estructura física de la cual se desea conocer su inclinación respecto a la vertical en el sistema cartesiano X-Y-Z. El sistema electrónico determina los ángulos de inclinación X-Y-Z, además del voltaje de la fuente de suministro y temperatura del circuito integrado sensor de inclinación ADIS16210. Estas variables se transmiten, como un conjunto de 5 datos de 16 bits cada uno, por radio-frecuencia a una de las estaciones del proveedor de servicios de conectividad IoT, Sigfox. En el sitio web de Sigfox se configura un servicio de envío de datos denominado “callback” de manera que los $5 \times 16 = 80$ bits recibidos en la red de Sigfox puedan ser re-transmitidos a otro proveedor de servicios de internet para su interpretación como ángulos de inclinación X-Y-Z más voltaje y temperatura con sus unidades dimensionales correspondientes. El sitio web de Google recibe el mensaje “callback” mediante una aplicación Web desarrollada exprofeso para el presente proyecto. La aplicación ajusta las variables de 16 bits a magnitudes dimensionadas para 3 ángulos más un voltaje y temperatura. Además la aplicación almacena automáticamente las magnitudes en una hoja de cálculo en la nube que puede ser compartida con otros usuarios.

Objetivos y resultados esperados

El objetivo del presente informe es mantener documentado el desarrollo tecnológico que aquí se describe de forma que sea de utilidad para su reproducibilidad o como referencia para desarrollos posteriores.

El objetivo general del proyecto es desarrollar un inclinómetro con conectividad al internet de las cosas. Los siguientes son los objetivos particulares:

- Desarrollar y construir un sistema electrónico que mida: 3 ángulos de inclinación coordenados, el voltaje de la batería que suministra energía al sistema y la temperatura del circuito.
- Desarrollar y programar el sistema electrónico para que ejecute los algoritmos de medición y de comunicación con el IoT.

- Configurar el sitio de Sigfox para la recepción y envío de las mediciones provenientes del sistema electrónico.
- Desarrollar y programar una aplicación Web de Google para la recepción, tratamiento y almacenamiento de las mediciones provenientes de Sigfox.
- Que el sistema, operando en conjunto, posea gran autonomía, es decir, que la intervención por parte del operador sea mínima.

Organización del informe

La sección de introducción plantea el problema, el método empleado para resolverlo y expone los resultados esperados. En el capítulo uno se describe el desarrollo del sistema de medición en sus dos componentes principales: componente electrónica y configuración o programación de sitios web. El capítulo dos contiene los resultados, conclusiones y trabajo a futuro. Finalmente los anexos contienen información detallada acerca de los recursos técnicos desarrollados como parte del presente informe.

1 Inclínómetro con conectividad al internet de las cosas

El inclinómetro que se describe en esta sección, está construido sobre la base de tarjetas electrónicas comerciales, también conocidas con el nombre de kits de desarrollo. De esta manera está garantizada la confiabilidad y compatibilidad con la infraestructura existente para el IoT.

1.1 Descripción general

La figura 2 muestra el esquema general del inclinómetro con conectividad inalámbrica al IoT.

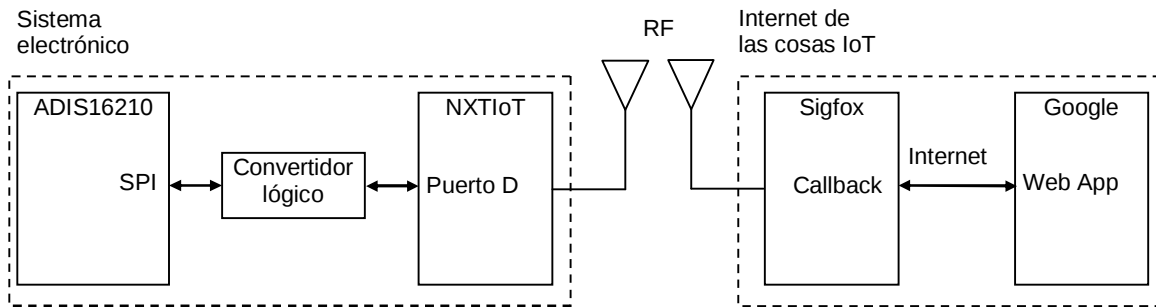


Figura 2. Esquema general.

En la figura 2 se muestran las dos componentes principales del inclinómetro:

- Sistema electrónico. Constituido sobre la base de 3 tarjetas electrónicas comerciales: Un inclinómetro triaxial de precisión ADIS16210, un convertidor lógico bidireccional de niveles de voltaje y un kit de desarrollo NXTIoT. El ADIS16210 funciona como transductor de inclinación que es conectado al puerto D del procesador en el NXTIoT mediante su interface periférica serial (SPI, Serial Peripheral Interface). Esta conexión requiere una conversión de niveles lógicos TTL de 5V en el NXTIoT a niveles TTL de 3V en el ADIS16210. En el NXTIoT está embebido un procesador Arduino que contiene los algoritmos de medición de inclinación y comunicaciones SPI y RF (Radio Frecuencia). Bajo este esquema, el procesador en el NXTIoT obtiene las mediciones de ángulo de inclinación, más voltaje de la batería y temperatura del circuito, y las envía por RF cada determinado tiempo a la componente IoT.
- Internet de las cosas IoT. Constituida por 2 servidores en la nube: El proveedor de servicios de conectividad Sigfox y el proveedor de servicios de Internet Google. El sitio Sigfox es configurado para generar un método “callback” cada vez que un mensaje es recibido, es decir, cada vez que el sistema electrónico envía un conjunto de mediciones. El mensaje “callback” es recibido por una aplicación Web, Web App, en el sitio de Google. La aplicación consiste en recibir la información de inclinación más voltaje y temperatura para decodificarlos y almacenarlos en las últimas celdas de una hoja de cálculo de Google.

A continuación una descripción detallada de cada uno de los elementos descritos en el esquema general.

1.2 Sistema electrónico

El diagrama detallado del sistema electrónico se muestra en la figura 3 y anexo A. En este diagrama se ejemplifica la conexión eléctrica entre las tres tarjetas: NXTIoT, ADIS16210 y convertidor lógico. La conexión se realiza bajo el esquema de maestro SPI, el NXTIoT, y esclavo SPI, el ADIS16210. En este esquema el maestro gobierna la operación del esclavo al enviar comandos y recibir respuestas. La iniciativa entonces es propiedad del maestro y es él que comienza cada instrucción que se ejecuta en el esclavo. Es importante mencionar que la lógica TTL con la que operan las dos interfaces SPI de la figura 3 es con diferentes niveles de voltaje; para el NXTIoT es de 5V y para el ADIS16210 es de 3V. Por esta razón se utiliza el convertidor lógico bidireccional cuya función exclusiva es la de desplazar niveles

lógicos de 3 V (LV) hacia niveles lógicos de 5V (HV) y viceversa. En este convertidor existen 4 entradas/salidas de bajo voltaje (LV), etiquetadas con A, y cuatro salidas/entradas de alto voltaje (HV), etiquetadas con B. Las terminales del convertidor lógico son bidireccionales, es decir pueden actuar como entradas o salidas. El NXTIoT se polariza con una batería de 9V y éste redistribuye 3.3V y 5V al resto del sistema.

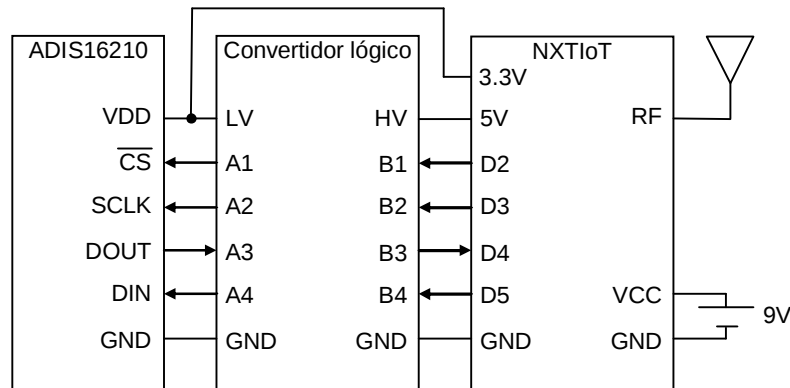


Figura 3. Sistema electrónico.

El kit de desarrollo NXTIoT contiene el algoritmo de medición que instruye al esclavo ADIS16210 en la configuración de su operación y en la obtención de mediciones de 3 ángulos de inclinación, voltaje de la batería de 3.3V y temperatura interna del circuito. Por otra parte, las 5 mediciones obtenidas por el NXTIoT del ADIS16210, son enviadas, mediante su sección de RF, hacia el IoT.

Las señales de comunicación SPI se describen en la tabla 1.

ADIS16210		NXTIoT	
Señal SPI	Descripción	Puerto D	Descripción
$\overline{CS}$	Chip Select, habilitador del circuito integrado	D2	$\overline{SS}$ Slave Select, selector de esclavo
SCLK	Serial Clock, reloj serial	D3	Serial Clock, Reloj serial
DOUT	Data Output, salida de datos	D4	MISO Master Input Slave Output, entrada maestro salida esclavo
DIN	Data Input, entrada de datos	D5	MOSI Master Output Slave Input, salida maestro entrada esclavo

Tabla 1. Descripción de las señales SPI del sistema electrónico

1.2.1 Kit de desarrollo NXTIoT

El kit de desarrollo NXTIoT (IoTnet, 2018b) es una tarjeta electrónica para el desarrollo de aplicaciones del IoT con base en el microcontrolador ATMEGA 328P (Atmel, 2016). Si bien no existe documentación detallada por parte del fabricante (IoTnet, 2018^a), se puede deducir que la organización del kit es como se muestra en la figura 4. En donde las terminales de conexión y LED's con el símbolo "¿", no se sabe de qué forma están relacionadas con el microcontrolador 328P. Por otra parte se desconoce la terminal de conexión para los puertos PB1 y PB2 del mismo microcontrolador.

Para la siguiente explicación refiérase a la figura 4.

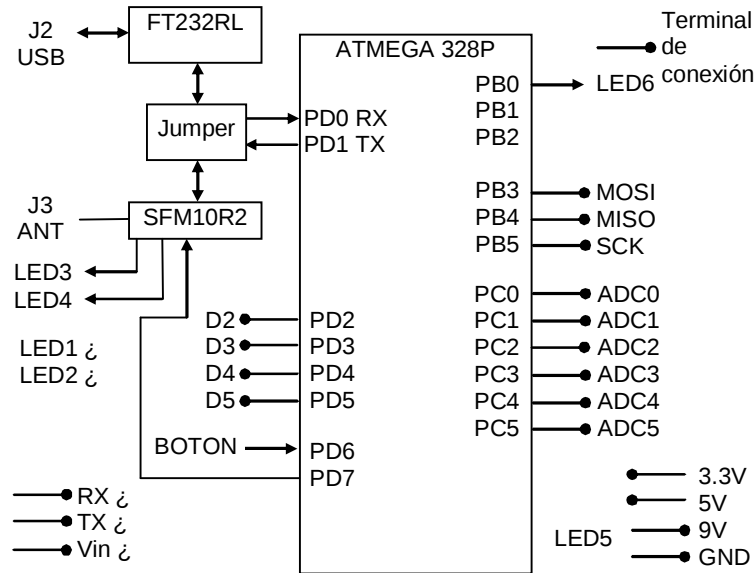


Figura 4. Esquema general del kit de desarrollo NXTIoT.

El kit NXTIoT puede ser programado con una computadora y el software de Arduino (Arduino, 2018a) mediante el conector micro USB J2 y retirando el “Jumper” de su posición. De esta forma se habilita la conexión USB de la computadora hacia el puerto serie UART del 328P a través del convertidor USB a serie FT232RL. En esta situación es probable que la primera vez que se conecte el kit NXTIoT a la computadora, se solicite la instalación de los drivers del dispositivo FT232RL. Los drivers se pueden conseguir en la página de la empresa FTDI, Future Technology Devices International (FTDI Chip, 2018). Una vez establecida la conexión del kit NXTIoT con la computadora, éste puede ser programado desde el IDE (Integrated Development Environment, Ambiente Integrado de Desarrollo) de Arduino (Arduino 2018b) de la misma manera en que se hace para la popular tarjeta de Arduino UNO (Arduino, 2018c).

El circuito integrado SFM10R2 es un dispositivo de comunicación RF con el servidor IoT de la empresa Sigfox (Wisol, 2016). Cuando se posiciona el “Jumper” sobre su terminal, el puerto serie del microcontrolador 328P queda conectado físicamente al SFM10R2, posibilitando de esta manera la transmisión inalámbrica al IoT desde un algoritmo previamente grabado en el microcontrolador. El estado de la transmisión inalámbrica en el SFM10R2 es visualizado mediante los LED 3 y 4. Además el conector J3 se utiliza para la antena de 902.2 MHz. La comunicación entre el microcontrolador 328P y el SFM10R2 es a través del puerto serie asíncrono a 9600 bps utilizando el estándar de comandos AT y la señal de habilitación/deshabilitación en PD7. En este esquema, el 328P forma las cadenas conteniendo los comandos apropiados para la operación del SFM10R2 y éste último ejecuta las operaciones indicadas. La tabla 2 muestra los comandos AT utilizados en la presente aplicación.

Comando AT	Descripción
AT	Puesta inicial
AT\$RC	Restablece a cero el canal de radio frecuencia
AT\$SF=cadena	Envía por RF los caracteres contenidos en el parámetro “cadena”.

Tabla 2. Comandos comunes en el SFM10R2.

El puerto D del 328P está disponible para su conexión con otros periféricos desde la pata 2 a la 5. Para la presente aplicación, el puerto D se utiliza como un emulador por software de la interface periférica serial, SPI, con el inclinómetro triaxial de precisión ADIS16210, como se muestra en la tabla 1. La funcionalidad de la interface SPI está estrechamente ligada al ADIS16210, razón por la cual su descripción se

desarrolla en la siguiente sección. Los puertos B y C del NXTIoT no se usan en la presente aplicación, a excepción del PD6 (entrada digital) y PB0 (salida digital) que son utilizados con fines de depuración.

El kit NXTIoT recibe 9V de entrada e internamente los regula a 5V y 3.3V de salida para ser utilizados en el resto del sistema electrónico.

1.2.2 Inclinómetro triaxial de precisión ADIS16210

El ADIS16210 es un sistema digital con tecnología MEMS (MicroElectroMechanical Systems, Sistemas Electro-Mecánicos) para la medición de inclinación en un rango completo de orientación ($\pm 180^\circ$) (Analog Devices, 2015). Su interface SPI proporciona el método para acceder a los datos de inclinación y controles de su configuración. El diagrama de conexión eléctrico típico para este dispositivo consiste en el esquema básico maestro-esclavo SPI que se muestra en la figura 5.

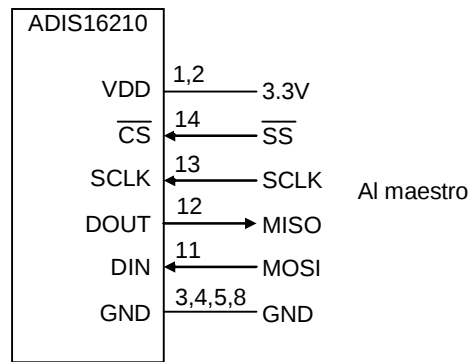


Figura 5. Diagrama de conexión para el ADIS16210.

El ADIS16210 es un sistema autónomo que no requiere inicialización por parte del usuario maestro. La operación básica se consigue al enviar comandos SPI desde el maestro. Los comandos SPI en el ADIS16210 están conformados por dos ciclos full dúplex (transmisión recepción simultánea) de 16 bits cada uno como se muestra en la figura 6.

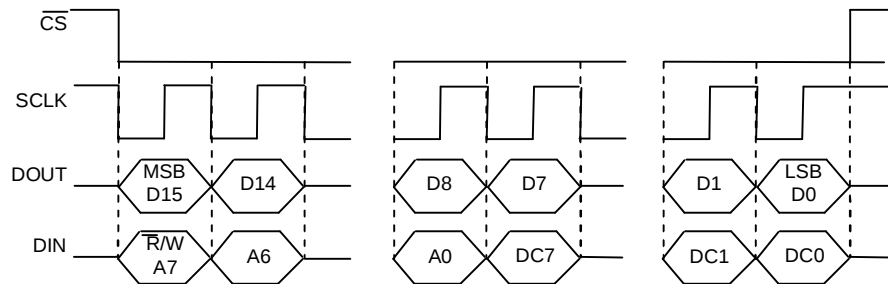


Figura 6. Ciclo SPI de 16 bits.

En donde el comando SPI que envía el maestro al ADIS16210, por la línea DIN, es una palabra de 16 bits compuesta por 8 bits de dirección, A7 a A0 más 8 bits de datos, DC7 a DC0. Por otra parte la respuesta del ADIS16210 al ciclo SPI anterior de 16 bits, por la línea DOUT, está conformada por la palabra de 16 bits D15 a D0. La secuencia de bits en DIN y DOUT está en sincronía con la línea de selección de esclavo $\overline{CS}$ y con la señal de reloj SCLK que son generadas por el maestro SPI.

Los comandos del ADIS16210 utilizados en la presente aplicación son los que se muestran en la tabla 3.

Comando SPI	DirecciónDato	Descripción
PROD_ID	0x5600	Leer identificador del dispositivo
XINCL_OUT	0x0C00	Leer ángulo de alabeo
YINCL_OUT	0x0E00	Leer ángulo de elevación
ZINCL_OUT	0x1000	Leer orientación con el eje de gravedad
SUPPLY_OUT	0x0200	Leer voltaje de suministro
TEMP_OUT	0x0A00	Leer temperatura interior
SLP_CNT	0xBA00	Periodo de bajo consumo infinito
GLOB_CMD	0xBE02	Ingreso al periodo de bajo consumo

Tabla 3. Comandos comunes en el ADIS16210.

1.2.3 Convertidor lógico bidireccional

El convertidor lógico bidireccional de la figura 7, es una tarjeta de circuito electrónico que tiene el propósito de desplazar niveles de 3.3V (LV) a 5V (HV) y viceversa (Ada fruit, 2018).

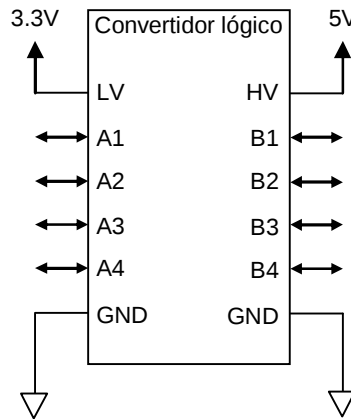


Figura 7. Convertidor bidireccional.

Existen un total de 4 canales bidireccionales A y B separados para la conversión de alto (HV) a bajo voltaje (LV) y viceversa. Como ejemplo de conexión, un una señal de 3.3V en la terminal A1 producirá un voltaje de 5V en la terminal B1. De forma similar, un una señal de 0V en la terminal A1 producirá un voltaje de 0V en la terminal B1. De forma contraria, un una señal de 5V en la terminal B1 producirá un voltaje de 3.3V en la terminal A1. De forma similar, un una señal de 0V en la terminal B1 producirá un voltaje de 0V en la terminal A1.

1.2.4 Algoritmo de medición

El algoritmo que gobierna la operación del inclinómetro mostrado en el esquema general de la figura 3 es el mostrado en la figura 8. Este algoritmo se ejecuta en el procesador ATMEGA 328P de la tarjeta NxtIoT.

En donde la variable “bufer” resulta en una cadena de 10 caracteres hexadecimal, es decir, las siguientes 5 variables de 2 caracteres hexadecimal cada una: prod_id, xincl_out, yincl_out, zincl_out, supply_out y temp_out, que se obtienen mediante la implementación de los comandos de la tabla 3. La cadena “bufer” se envía por RF al servidor Sigfox del IoT utilizando los comandos AT de la tabla 2.

Para complementar el algoritmo de la figura 8 se desarrollaron las siguientes funciones:

- spi_tx(dato): Transmite un dato de 16 bits por la interface SPI y regresa el registro de 16 bits leído en el ciclo SPI previo.

- `angulo(dato)`: Convierte un dato de 16 bits a flotante en grados (Analog Devices, 2015).
- `temperatura(dato)`: Convierte un dato de 16 bits a flotante en centígrados (Analog Devices, 2015).
- `voltaje(dato)`: Convierte un dato de 16 bits a flotante en volts (Analog Devices, 2015).

Adicionalmente al algoritmo de la figura 8, se tienen rutinas para desplegar en el emulador de terminal del IDE de Arduino los valores de las 5 variables en formatos decimal, hexadecimal y convertidas a magnitudes dimensionadas.

La implementación del algoritmo es en lenguaje C para Arduino y el código fuente se presenta en el Anexo C del presente informe.

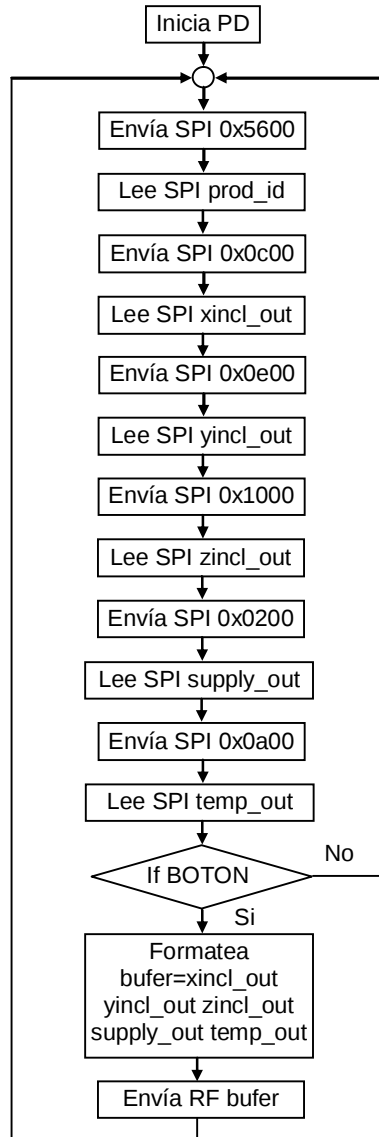


Figura 8. Algoritmo de medición.

1.3 Proveedores de servicios en la nube

El sistema electrónico para el inclinómetro se encarga de medir tres ángulos de inclinación más voltaje y temperatura, para enviar las mediciones a la nube del IoT. El primer paso para acceder a servicios en la

nube es adquirir un kit de desarrollo o dispositivo básico a través de un proveedor nacional que permita el enlace entre las variables físicas y el servidor de conectividad en la nube. En el ámbito nacional, IoTnet México constituye ese primer eslabón en la conectividad ya que provee servicios de conectividad a nivel regional. IoTnet es el operador local de la red Sigfox para el IoT en México y se encarga de comercializar dispositivos y proporcionar asesoría para el desarrollo de soluciones con base en el IoT. En esta cadena de proveedores, Sigfox es el proveedor de conectividad global en donde residen los datos e información que el usuario final desea como parte de su solución. Sigfox mantiene una serie de radio-bases para proporcionar conectividad a los dispositivos y un servidor de datos para el almacenamiento de la información que generan dichos dispositivos. También los servidores de Sigfox pueden ser automatizados para generar diversos eventos en respuesta al arribo de datos provenientes de los diversos dispositivos que atienden. Como parte de esta automatización y considerando que Google se ha convertido en una herramienta ampliamente difundida para el procesamiento de información en la nube, el servidor de Sigfox se puede configurar para generar mensajes “callback” o peticiones que Google puede atender y poner a la disposición de los usuarios finales los datos en la nube en una plataforma amigable. En nuestro concepto, una hoja de cálculo compartida en el drive de Google, es la forma más conveniente para que los usuarios finales del inclinómetro procesen las mediciones de tres ángulos más voltaje de suministro y temperatura. El proceso descrito se muestra gráficamente en la figura 9.

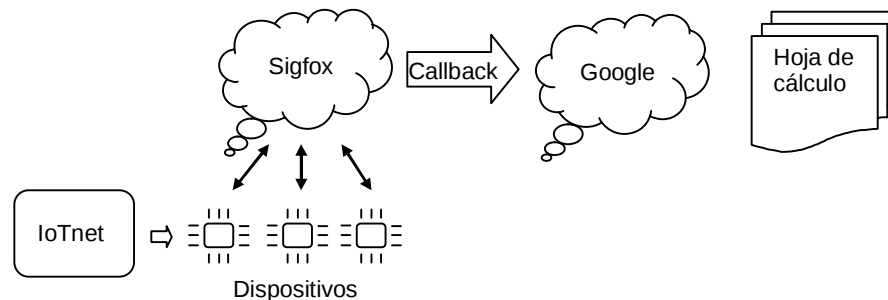


Figura 9. Red de proveedores en la nube para el IoT.

A continuación una descripción detallada de los servicios y operaciones en la red de la figura 9.

1.3.1 IoTnet

IoTnet México (IoTnet, 2018a) es el proveedor de dispositivos y el operador local de la red Sigfox para México. El primer paso para desarrollar un proyecto con conectividad al IoT es ponerse en contacto con un representante de IoTnet para plantearle el proyecto que se desea desarrollar y con base en esto el proveedor sugiere una solución y la cotiza. Posteriormente, y una vez aceptados los términos de la cotización y fincada la compra, en este caso del dispositivo NXTIoT, se solicita al representante la creación de una cuenta en el sitio de Sigfox para dar de alta y registrar el dispositivo. El representante gestiona la cuenta en Sigfox y envía un correo conteniendo el sitio web de Sigfox para la creación de la cuenta mediante el correo electrónico del desarrollador y un password. Al ingresar al sitio de SigFox a través de la cuenta ya creada, se debe registrar el dispositivo mediante un ID y un PAC. Estos parámetros se pueden obtener al ejecutar el ejercicio “Leer ID/PAC” del tutorial “NXTIoT_DEVKIT” (IoTnet, 2018b). En esta circunstancia el dispositivo queda registrado y listo para enviar información en la nube de Sigfox.

1.3.2 Sigfox

Sigfox es el proveedor de servicios de conectividad global (Sigfox, 2018). Proporciona conectividad y acceso a la nube a los diversos dispositivos fabricados y comercializados por socios locales del grupo Sigfox. La tecnología que provee SigFox se puede resumir en los siguientes 4 puntos:

- El dispositivo de un cliente local es habilitado y envía un mensaje que contiene información útil para diversos usuarios.

- Las radio-bases al alcance del dispositivo reciben el mensaje.
- Las radio-bases envían la información al servidor en la nube de Sigfox.
- La nube de Sigfox envía el mensaje a alguna de las plataformas que puede definir el cliente del dispositivo.

Si bien la plataforma de administración de la información que generan los dispositivos puede ser el mismo portal de Sigfox, lo más común entre los desarrolladores es que otro proveedor de servicios en la nube procese la información de los dispositivos. Por esta razón, Sigfox implementa en su sitio métodos para enviar mensajes a otros servidores. En particular, el método denominado “callback” es una solicitud HTTP que permite enviar mensajes a otros servidores en el instante preciso en que un dispositivo genera la información.

Para la presente aplicación, se personaliza un método “callback” en el portal de Sigfox mediante los 4 pasos siguientes:

1. Con clave y password se ingresa al portal de Sigfox en <https://backend.sigfox.com/auth/login>. A continuación aparece la pantalla inicial de bienvenida de la figura 10.

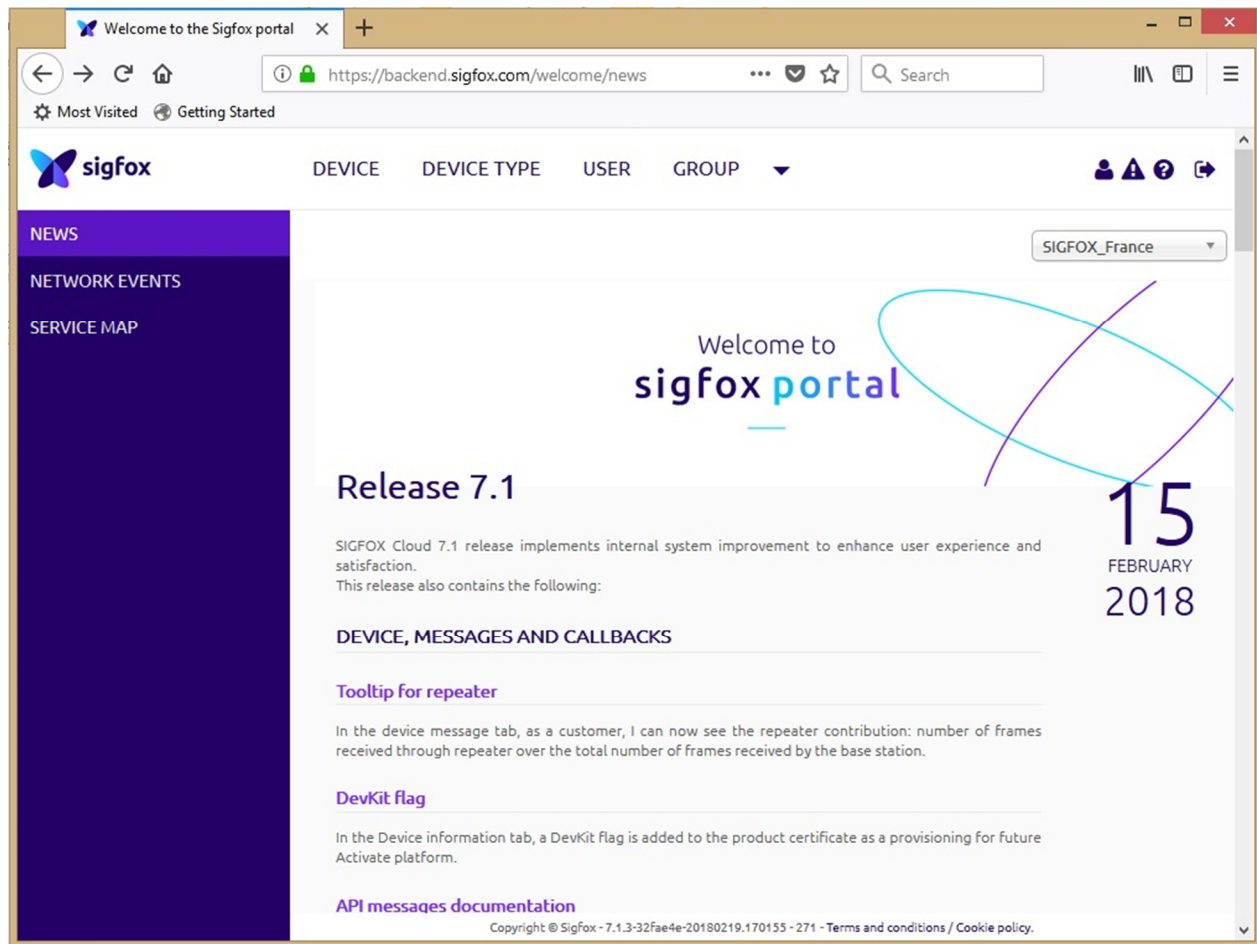


Figura 10. Pantalla de bienvenida al sitio de usuarios de Sigfox.

2. En esta pantalla se elige la opción “DEVICE” del menú superior. A continuación aparece la pantalla de la figura 11 en donde se despliega la información del dispositivo que se registró en esta cuenta.

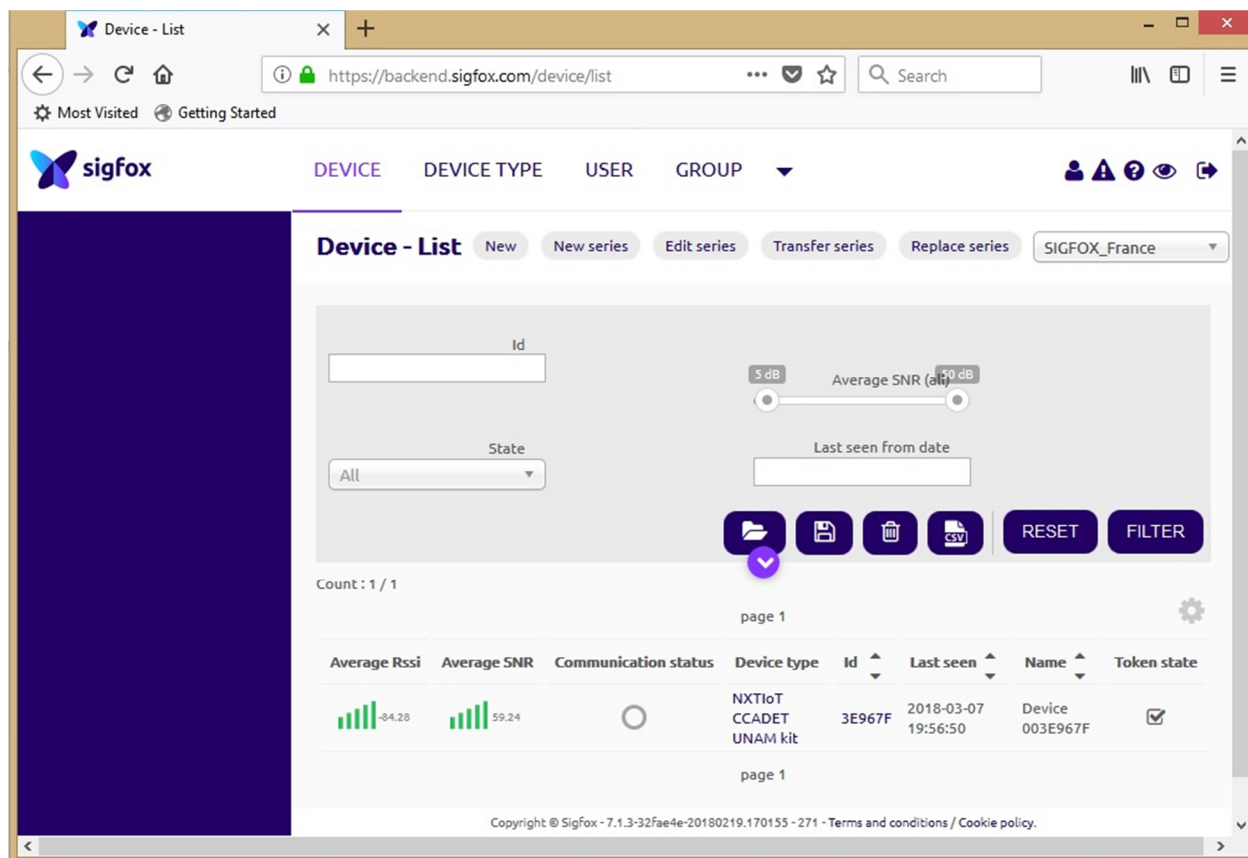


Figura 11. Pantalla para la lista de los dispositivos registrados en la cuenta.

- En esta pantalla se elige el dispositivo al presionar su clave de identificación "Device type" de referencia. A continuación se despliega la pantalla de la figura 12 en donde se puede observar la información detallada del dispositivo y a la izquierda un menú de opciones de entre las cuales se encuentra la opción "CALLBACKS".

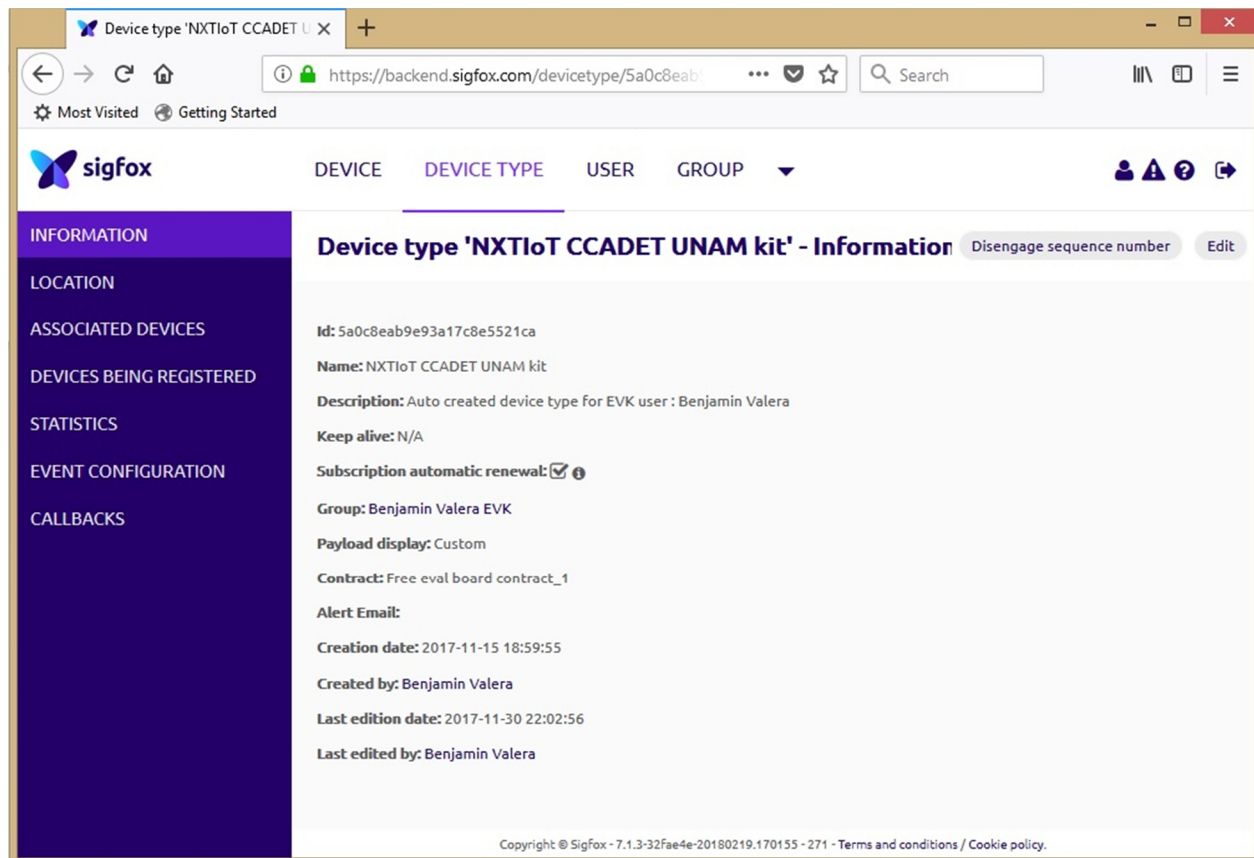


Figura 12. Pantalla para el despliegue de la información del dispositivo.

- Al presionar la opción CALLBACKS se despliega la lista de mensajes "callback" configurados para el dispositivo en cuestión. En particular para la presente aplicación, la figura 13 muestra el "callback" que permite el envío de información al servidor de Google (el último en la lista con la casilla "Enable" habilitada). En la figura 14 se muestra la pantalla para la configuración del "callback" que hace posible el envío de información a la nube de Google y la tabla 4 muestra la descripción de los parámetros del "callback".

Device type NxtIoT CCADET UNAM kit - Callback edition

Callbacks

Type: DATA UPLINK

Channel: URL

Send duplicate: ☐

Custom payload config: `anguloX::int:16:little-endian anguloY::int:16:little-endian anguloZ::int:16:little-endian volt: ?`

URL syntax: `http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...`
 Available variables: `device, time, duplicate, snr, station, data, avgSnr, lat, lng, rssi, seqNumber`
 Custom variables: `customData#anguloX, customData#anguloY, customData#anguloZ, customData#voltaje, customData#temperatura`

Url pattern: `https://script.google.com/a/ccadet.unam.mx/macros/s/ID de la App de google/exec?`

Use HTTP Method: GET

Send SNI: ☒ (Server Name Indication) for SSL/TLS connections

Headers:

header	value
--------	-------

Ok Cancel

Copyright © Sigfox - 7.1.3-32fae4e-20180219.170155 - 271 - Terms and conditions / Cookie policy.

Figura 14. "Callback" para el envío de información a la nube de Google.

Parámetro	Descripción	Contenido
Type	Tipo para la entrega del mensaje	DATA UPLINK
Channel	Medio de comunicación	URL
Send duplicate	Si la información se envía por cada radio-base que la recibió	Ninguno
Custom payload config	Definición del paquete de información	anguloX::int:16:little-endian anguloY::int:16:little-endian anguloZ::int:16:little-endian voltaje::int:16:little-endian temperatura::int:16:little-endian
Url pattern	Dirección electrónica para la entrega del paquete de información	<code>https://script.google.com/a/ccadet.unam.mx/macros/s/ID de la APP de google/exec?timestamp={time}&device={device}&anguloX={customData#anguloX}&anguloY={customData#anguloY}&anguloZ={customData#anguloZ}&voltaje={customData#voltaje}&temperatura={customData#temperatura}</code>

Use HTTP Method	Tipo de método para la entrega	GET
Send SNI	Si el envío contiene el Nombre de Identificación del Servidor	Habilitado
Headers	El encabezado que acompaña a la información	Ninguno

Tabla 4. Parámetros del “callback” con Google.

En la tabla 4 es importante destacar lo siguiente:

- El “Custom payload config” representa la información de las 5 variables (3 ángulos, 1 voltaje y 1 temperatura) organizada como caracteres hexadecimales que representan variables enteras de 16 bits en formato de menor significancia a la izquierda y mayor significancia a la derecha, de acuerdo con la nomenclatura adoptada en los procesadores de Intel. Adicionalmente a estas variables, se envía el tiempo y el número identificador del dispositivo.
- En el “URL pattern”, el **ID de la APP de google** debe ser remplazado por el número con el que Google publica la aplicación, como se muestra en la siguiente sección.

1.3.3 Google

Google no solo es el motor de búsqueda de la mayoría de los navegadores que usamos en la vida diaria. Google además es una red de servidores que proporcionan servicios de respaldo de información, tienda de aplicaciones, herramientas de desarrollo de software, correo electrónico, aplicaciones de oficina entre muchos otros.

En la presente aplicación, Google es utilizado como unidad de respaldo, editor de hoja de cálculo y ambiente de desarrollo de aplicaciones, “App script”, como se muestra en la figura 15.

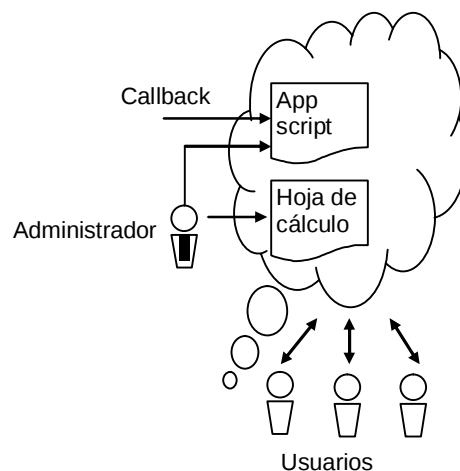


Figura 15. Google como ambiente integrado de trabajo colaborativo.

En la figura 15 el administrador del mini sitio de Google, asigna permisos para acceder a la hoja de cálculo, principalmente por parte de otros usuarios. El mismo administrador configura la aplicación de

Google, “App script” o “App de Google”, para recibir la información del inclinómetro incrustada en el “callback” como se mencionó en la sección anterior. La configuración en la nube de Google se desarrolla conforme a los siguientes 7 pasos:

1. El administrador crea una hoja de cálculo en blanco en el drive de Google al presionar el botón derecho y con el menú flotante que aparece en la figura 16. Es importante mencionar que previamente el administrador tiene una cuenta de correo electrónico en Google y tiene instaladas las aplicaciones Drive y Google Apps Script.

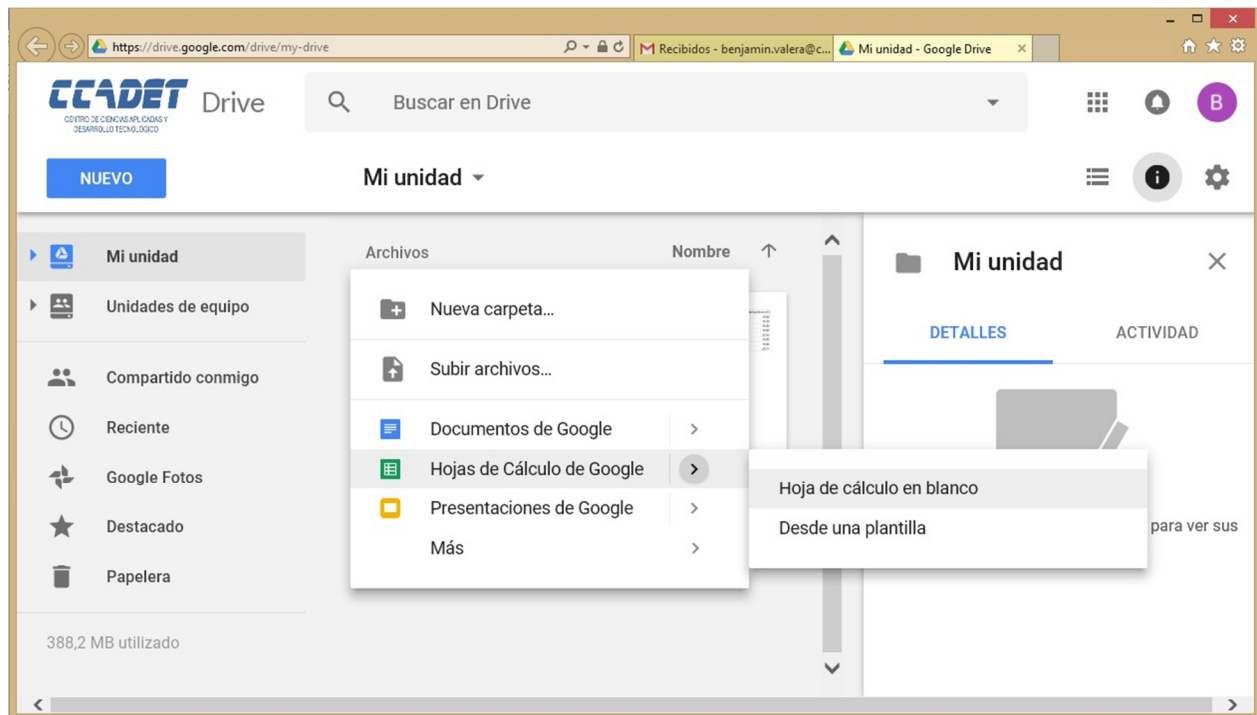


Figura 16. Menú flotante para nueva hoja de cálculo en Google Drive.

2. Inicialmente en la hoja de cálculo se puede editar el encabezado como se muestra en la figura 17. Es importante mencionar que esta hoja de cálculo tiene una clave de identificación **ID de la hoja de google**, como se muestra en la figura 17, dato que será utilizado en el paso 4.

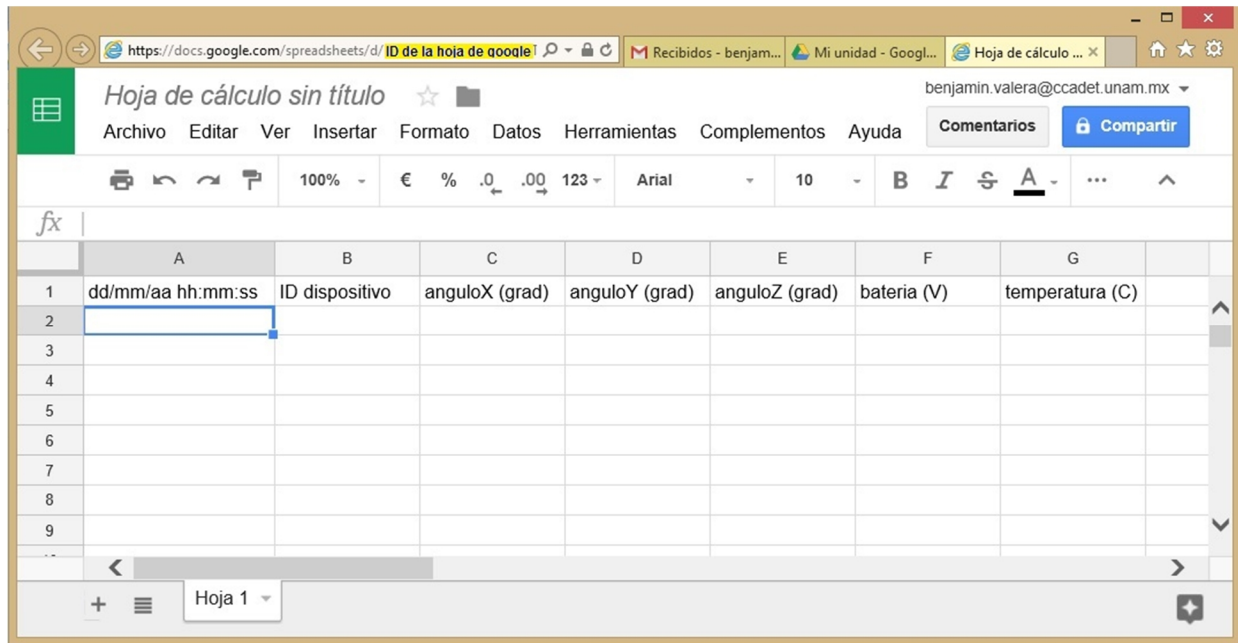


Figura 17. Hoja de cálculo en Google Drive.

3. En el Drive de Google presionar el botón derecho para crear un nuevo “Google Apps Script” como se muestra en la figura 18.

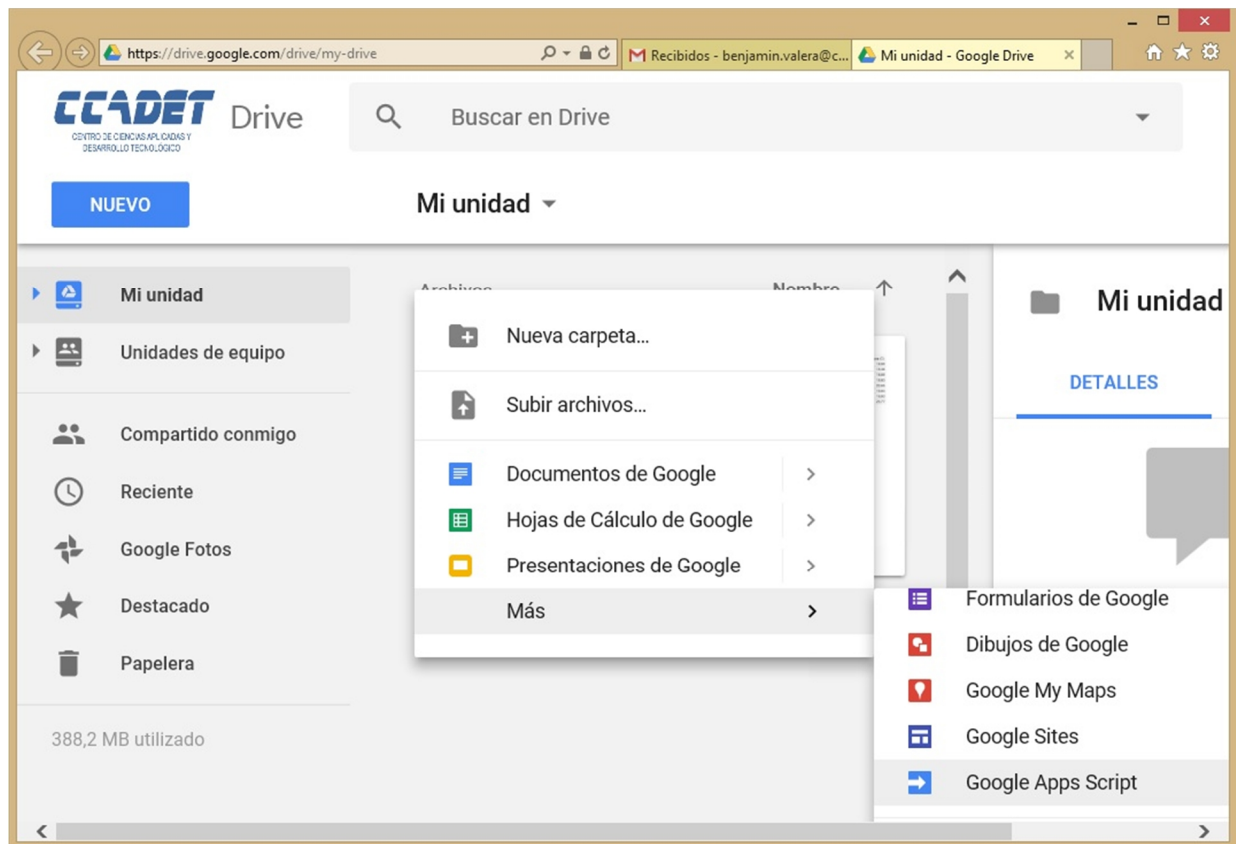


Figura 18. Menú flotante para nuevo “Google Apps Script” en Google Drive.

4. El archivo del Script contiene el código de la aplicación que recibe la información del dispositivo de Sigfox e incrusta dicha información en la hoja de cálculo referenciada por el ID de la hoja de google, como se muestra en la misma figura 19. El código fuente de esta aplicación se muestra en el anexo D.

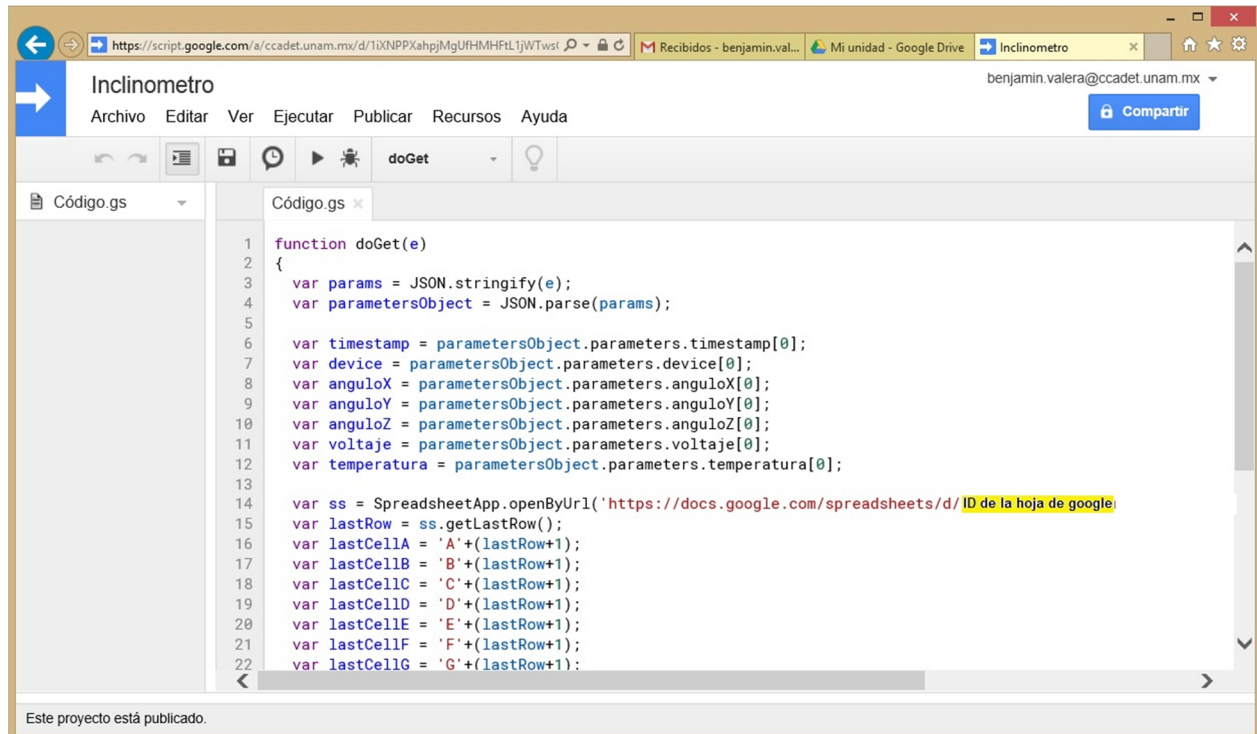


Figura 19. Nuevo archivo del “Google Apps Script”.

5. Los archivos de la hoja de cálculo y Script se pueden llamar de cualquier forma pero deben residir en el drive del administrador. El script de Google se publica como “aplicación web” al seleccionar esta opción del menú como se muestra en la figura 20.

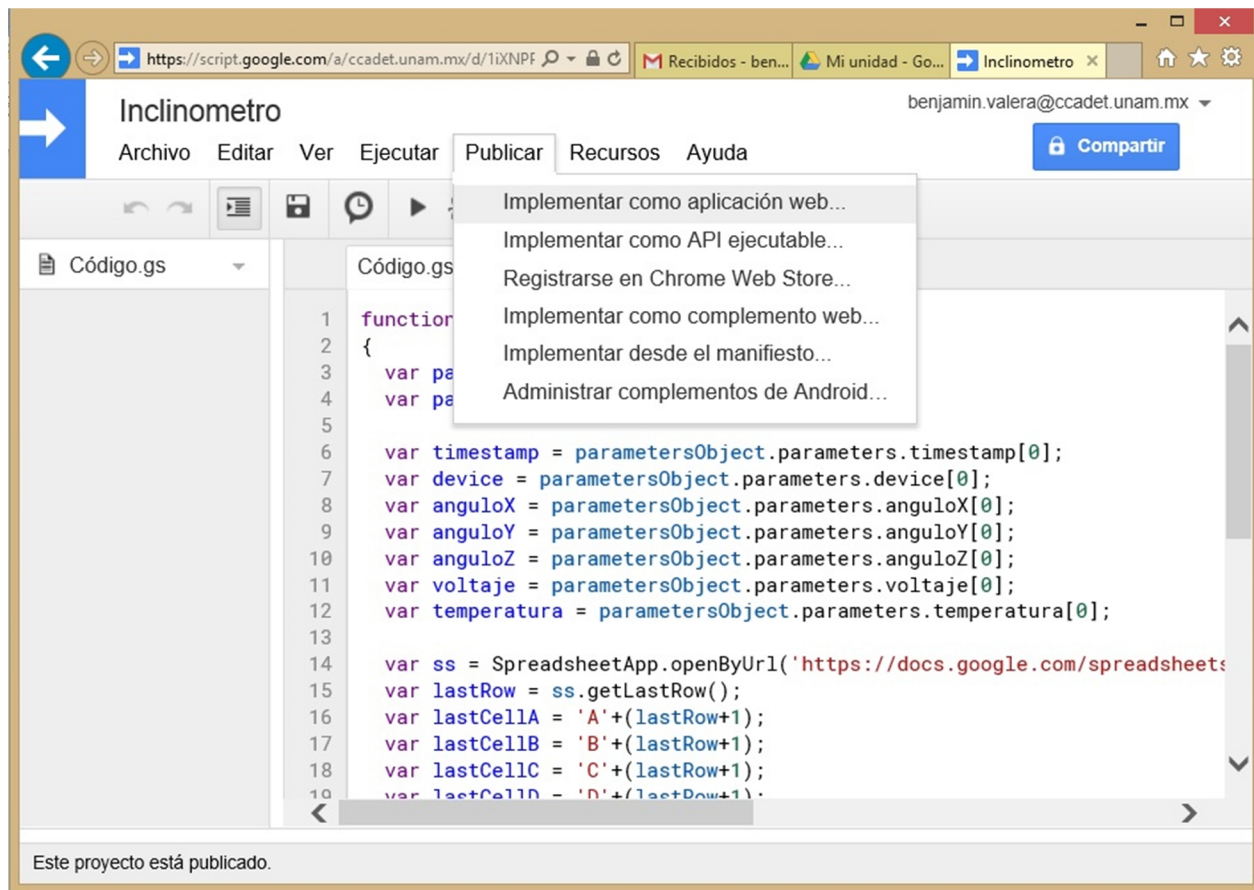


Figura 20. Publicación del "Google Apps Script".

6. Posteriormente se muestra la ventana emergente de la figura 21 que resume las propiedades de la aplicación.

Implementar como aplicación web

URL actual de la aplicación web: [Inhabilitar aplicación web](#)

https://script.google.com/macros/s/: **ID de la APP de google**

Verifica la aplicación web para el [último código](#).

Versión del proyecto:

Nuevo ▼

Describe qué ha cambiado...

Ejecutar la aplicación como:

Yo (benjamin.valera@ccadet.unam.mx) ▼

Debes autorizar la secuencia de comandos antes de distribuir la URL.

Quién tiene acceso a la aplicación:

Cualquier persona, incluso de forma anónima ▼

Actualizar

Cancelar

[Ayuda](#)

Figura 21. Resumen de la publicación del “Google Apps Script”.

7. Al actualizar los parámetros de la figura 21, aparece la confirmación de la figura 22. En este diálogo, es importante extraer la clave **ID de la APP de google** que se utiliza en la configuración del “callback” descrito en la sección anterior.

Implementar como aplicación web

Este proyecto se ha implementado ahora como una aplicación web.

URL actual de la aplicación web:

https://script.google.com/macros/s/: **ID de la APP de google**

Verifica la aplicación web para el [último código](#).

Aceptar

Figura 22. Confirmación de la publicación del Google Apps Script.

A partir del paso número 7 los servidores WEB mostrados en la figura 15 están listos para recibir información del inclinómetro y actualizar la hoja de cálculo de Google.

2 Resultados y discusión

Se diseñó y constituyó un instrumento para la medición de inclinación triaxial compuesto por dos componentes: un sistema electrónico y conectividad al IoT.

2.1 Resultados

Los resultados obtenidos para el inclinómetro se reportan de manera cualitativa en esta sección. Se reportan fotografías y capturas de pantalla de las componentes electrónica y hoja de cálculo en la nube del IoT.

2.1.1 Prototipo desarrollado

La figura 23 muestra el sistema electrónico desarrollado.

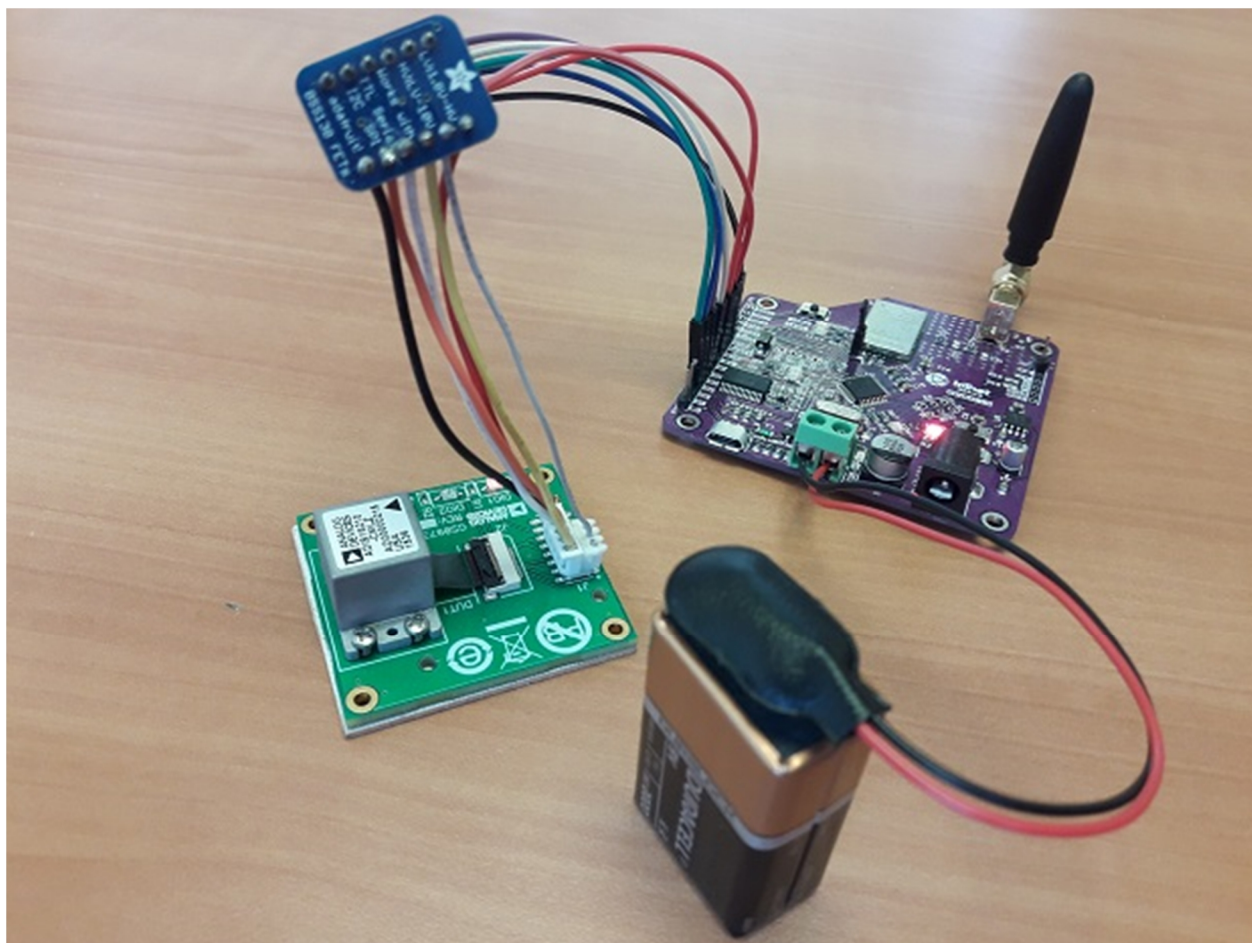


Figura 23. Sistema electrónico desarrollado.

2.1.2 Hoja de cálculo

Para verificar el desempeño del instrumento se realizan diversas ejecuciones para que el inclinómetro envíe información a la nube. La figura 24 muestra la hoja de cálculo en el Google Drive del administrador.

	A	B	C	D	E	F	G
	dd/mm/aa hh:mm:ss	ID dispositivo	anguloX (grad)	anguloY (grad)	anguloZ (grad)	bateria (V)	temperatura (C)
1	dd/mm/aa hh:mm:ss	ID dispositivo	anguloX (grad)	anguloY (grad)	anguloZ (grad)	bateria (V)	temperatura (C)
2	21/02/2018 12:02:08	3E967F	1,241455078	-2,010498047	87,63793945	3,19519043	18,89
3	21/02/2018 12:05:21	3E967F	1,159057617	-2,092895508	87,61047363	3,194274902	19,36
4	21/02/2018 12:11:04	3E967F	1,373291016	-2,026977539	87,55554199	3,196258545	18,89
5	21/02/2018 12:18:40	3E967F	1,587524414	-1,93359375	87,50061035	3,194274902	19,83
6	22/02/2018 15:16:29	3E967F	2,993774414	-1,329345703	86,72607422	3,192596436	22,65
7	07/03/2018 13:56:50	3E967F	1,208496094	-1,505126953	88,07189941	3,194580078	19,83
8	7/03/2018 16:18:29	3E967F	-0,1373291016	-0,3021240234	89,67041016	3,146820068	19,83
9	08/03/2018 16:09:32	3E967F	0,4998779297	0,6317138672	89,19799805	3,116455078	20,77

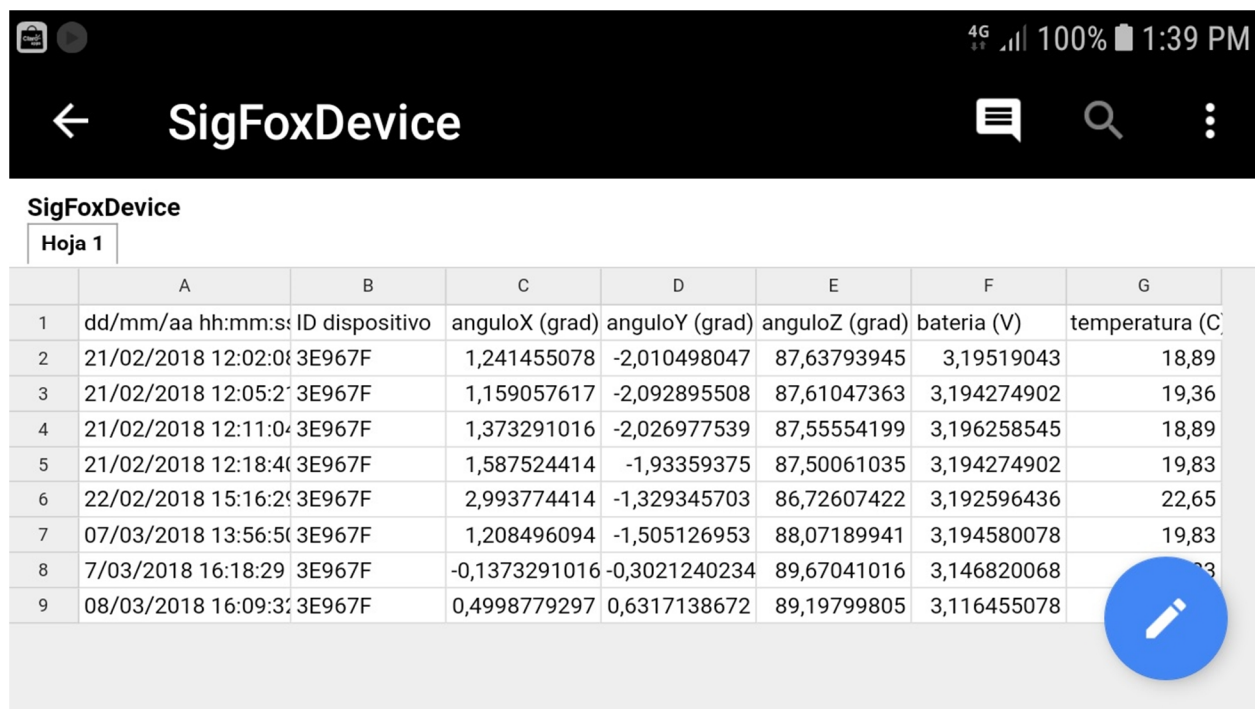
Figura 24. Hoja de cálculo en el Google Drive del administrador.

En una aplicación concreta como la descrita en (Ruiz *et al*, 2017), la información en la hoja de cálculo es de gran utilidad para evaluar la estabilidad mecánica de la estructura de la cual se desea conocer la evolución histórica de su inclinación en los ejes coordenados X-Y-Z. Por otra parte, la información de la batería es de utilidad para determinar en qué momento hay que remplazar la pila de 9 V en el sistema electrónico. La información de temperatura puede ser usada para conocer las condiciones ambientales bajo las que opera el sistema electrónico.

Se observa que la hoja de cálculo se actualiza con nueva información del inclinómetro casi al mismo instante en que se genera. En la hoja de cálculo se registran las siguientes variables:

- dd/mm/aa hh:mm:ss: Fecha y hora del evento registrada en el servidor de Sigfox.
- ID dispositivo: Número de identificación del dispositivo asignado en el servidor de Sigfox.
- anguloX: Ángulo de inclinación en grad en el eje X.
- anguloY: Ángulo de inclinación en grad en el eje Y.
- anguloZ: Ángulo de inclinación en grad en el eje Z.
- batería: Voltaje en V de la fuente de alimentación del inclinómetro ADIS16210.
- temperatura: Temperatura en C del inclinómetro ADIS16210.

La figura 25 muestra la misma hoja de cálculo en el drive de otra cuenta de Google pero accedida a través de un dispositivo móvil. El acceso se hace al ejecutar la aplicación "Drive" de Google y seleccionar la opción del menú "Compartidos conmigo". En esta situación debe aparecer el archivo "SigFox Device" como hoja de cálculo.



SigFoxDevice

Hoja 1

	A	B	C	D	E	F	G
1	dd/mm/aa hh:mm:ss	ID dispositivo	anguloX (grad)	anguloY (grad)	anguloZ (grad)	bateria (V)	temperatura (C)
2	21/02/2018 12:02:08	3E967F	1,241455078	-2,010498047	87,63793945	3,19519043	18,89
3	21/02/2018 12:05:21	3E967F	1,159057617	-2,092895508	87,61047363	3,194274902	19,36
4	21/02/2018 12:11:04	3E967F	1,373291016	-2,026977539	87,55554199	3,196258545	18,89
5	21/02/2018 12:18:40	3E967F	1,587524414	-1,93359375	87,50061035	3,194274902	19,83
6	22/02/2018 15:16:29	3E967F	2,993774414	-1,329345703	86,72607422	3,192596436	22,65
7	07/03/2018 13:56:50	3E967F	1,208496094	-1,505126953	88,07189941	3,194580078	19,83
8	7/03/2018 16:18:29	3E967F	-0,1373291016	-0,3021240234	89,67041016	3,146820068	19,83
9	08/03/2018 16:09:32	3E967F	0,4998779297	0,6317138672	89,19799805	3,116455078	

Figura 25. Hoja de cálculo en el Google Drive de un usuario habilitado para compartir el documento.

2.2 Discusión y trabajo a futuro

Se desarrolló un instrumento para la medición de tres ángulos de inclinación. Su principal uso es en el monitoreo de infraestructuras civiles de las cuales se requiere tener un registro del comportamiento estructural en el tiempo discreto. Se considera que este tiempo discreto de registro se puede desarrollar a tasas de muestreo relativamente prolongadas (10 a 20 min) pero constantes durante largos periodos de tiempo (típicamente años o meses). El instrumento solo puede generar estos registros de ángulo de inclinación y no se tiene, por el momento, algún criterio para decidir situaciones críticas de inclinación en el borde de alguna condición límite.

El prototipo descrito en este trabajo tiene el mérito de haber sido desarrollado a partir de las herramientas más básicas que se encuentran actualmente para desarrollar proyectos con base en el IoT. Como prueba de ello se pueden mencionar los siguientes aspectos. En primer lugar, el sistema electrónico fue desarrollado con base en un kit básico de desarrollo con base en un microcontrolador y dispositivo sensor MEM's. Finalmente, para la implementación del algoritmo de medición en el microcontrolador se utilizó una plataforma de programación de uso libre en la cual participan programadores que comparten sus experiencias en diversos foros abiertos sin fines de lucro.

Como trabajo a futuro se plantea lo siguiente:

- Desarrollar y construir una caja protectora para el sistema electrónico que sea inmune a condiciones ambientales severas pero que a su vez permita la propagación de la onda electromagnética hacia el IoT.
- Desarrollar un criterio para la detección de condiciones límite al borde de condiciones catastróficas de inclinación.

2.3 Especificaciones del sistema electrónico

- Dimensiones aproximadas: 14 cm X 7 cm X 6 cm.
- Peso aproximado: 80 gr.

- Voltaje de operación: 9V.
- Consumo de corriente en estado activo (Medido): 53.6 mA.
- Consumo de corriente en estado inactivo (Medido): 19.2 mA.
- Alcance de medición en el eje X: ± 180 grad.
- Medición mínima en el eje X: 0.055 grad.
- Exactitud en el eje X: 0.15 grad.
- Alcance de medición en el eje Y: ± 180 grad.
- Medición mínima en el eje Y: 0.055 grad.
- Exactitud en el eje Y: 0.15 grad.
- Alcance de medición en el eje Z: ± 180 grad.
- Medición mínima en el eje Z: 0.055 grad.
- Exactitud en el eje Z: 0.15 grad.
- Alcance de medición en temperatura: -40 C a 125 C.
- Medición mínima de temperatura: 0.47 C.
- Alcance de medición en voltaje: 3.0 V a 3.6 V
- Medición mínima en voltaje: 125.6 μ V.

2.4 Requisitos de operación

- Desempeño completamente autónomo. No requiere ajustes ni comandos de operación.
- Cuenta de Google.
- Instalación de Google Drive en el dispositivo de acceso a la cuenta.
- Dispositivo con acceso a internet (computadora o teléfono celular).
- Permiso del administrador para acceder a la información.

2.5 Guía para la detección de fallas

Para el sistema electrónico se requieren las condiciones de operación satisfactoria que se listan en la tabla 5.

Situación del sistema electrónico	Indicadores en el sistema electrónico
Energizado en espera de enviar información.	Led DIO1 en el ADIS16210 encendido. Led6 en el NXTIoT encendido.
Interruptor momentáneo S2 presionado.	Led2 en el NXTIoT parpadea una vez
Envío de información al IoT.	LED3 y LED4 en el NXTIoT parpadean 3 veces.

Tabla 5. Condiciones normales de operación para el sistema electrónico del inclinómetro.

En caso de presentarse otras circunstancias distintas a las mencionadas en la tabla 5, revise lo siguiente:

- El buen estado de la batería de 9V.
- La programación del código del anexo C en el microcontrolador del NXTIoT como se describió en 1.2.4 y 1.2.1.
- La buena conductividad en los cables del sistema electrónico.

Si las condiciones en el sistema electrónico son satisfactorias y no se percibe que llegue información a la hoja de cálculo en la nube, revise lo siguiente:

- Los permisos asignados por el administrador a la cuenta de Google.
- Que la App de Google haya sido publicada adecuadamente como se describió en 1.3.3.
- Que el método "callback" de Sigfox haya sido configurado adecuadamente como se describió en 1.3.2.

Conclusiones

Se diseñó y construyó un sistema para la medición de inclinación triaxial con aplicación en el monitoreo de infraestructuras civiles. El sistema desarrollado constituye una herramienta de evaluación para determinar la estabilidad mecánica de dichas infraestructuras en largos periodos de tiempo (típicamente 6 meses) con tasas de muestreo relativamente altas (20 min). Las componentes electrónica y configuración en la nube del IoT se construyeron con base en herramientas básicas de desarrollo (microcontroladores y software libre) lo cual facilita su eventual transferencia tecnológica y trabajo ambientes colaborativos.

Los experimentos realizados con el sistema comprueban su buen desempeño y viabilidad para usar esta tecnología en casos reales como el monitoreo de monumentos históricos en la Ciudad de México.

Agradecimientos

Al Dr. Sergio González García del Instituto Nacional de Antropología e Historia por sus aportaciones y validación del instrumento.

Al Ing. David Alonso Rodríguez Ruíz de la empresa IoTnet México por su asesoría en el kit de desarrollo electrónico.

Anexo A Diagrama esquemático

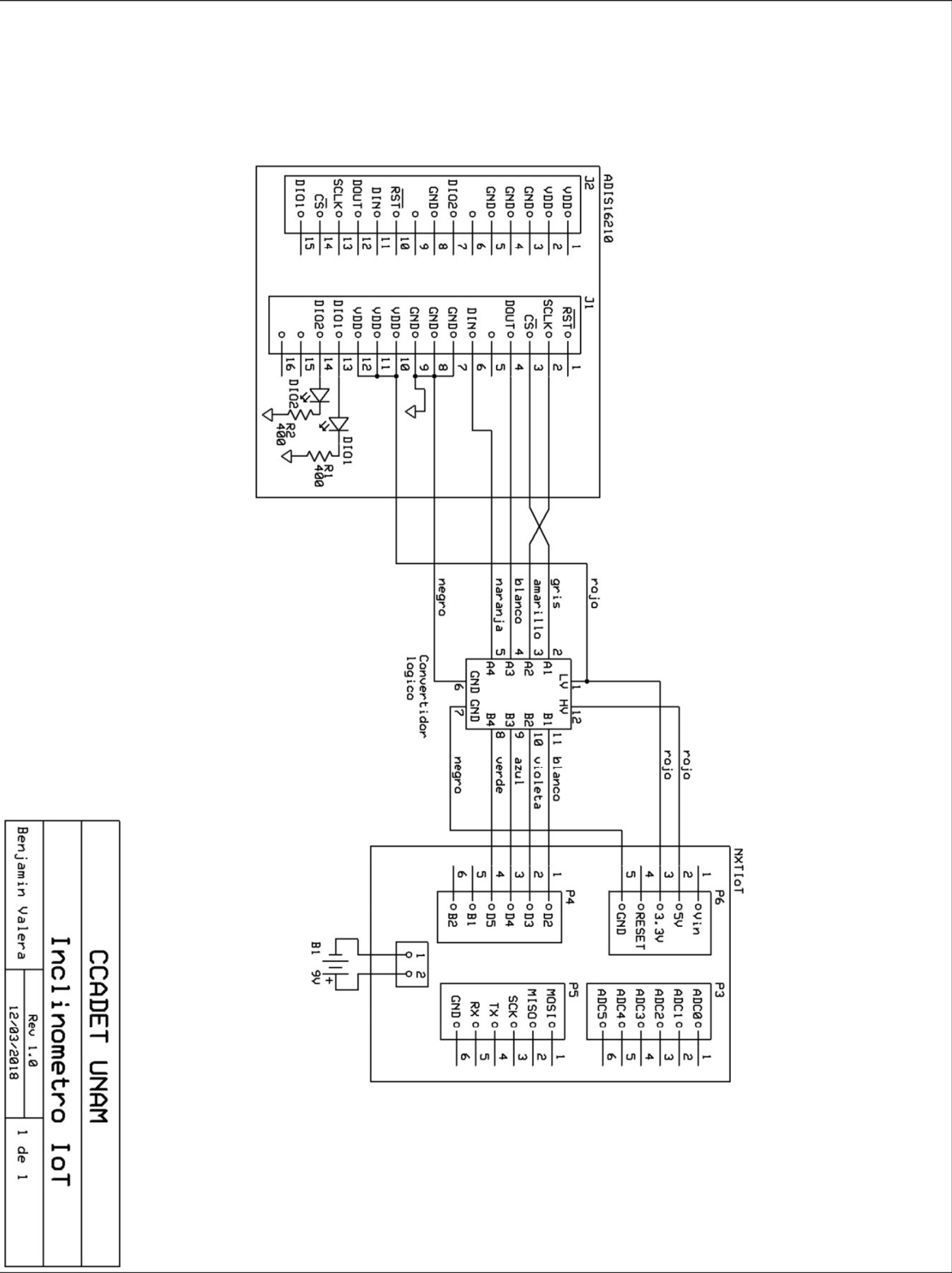


Figura 26. Dibujo esquemático.

Anexo B Lista de partes

ID	Descripción	Proveedor Num. Parte
ADIS16210	Inclinómetro triaxial de precisión	Newark 55T1448
Convertidor lógico	Convertidor lógico bidireccional de niveles TTL	330ohms AF-00757
B1	Batería de 9V	Comercio común
NXTIoT	Kit de desarrollo	IoTnet México Devkit 1.0

Tabla 6. Lista de partes.

Anexo C Código fuente Arduino

```
// Entradas:
//      D4 <- De J1 4, DOUT ADIS16210
// Salidas:
//      D2 -> A1 J1 3, CS    ADIS16210
//      D3 -> A1 J1 2, SCLK AIDS16210
//      D5 -> A1 J1 6, DIN   AIDS16210

#include <SoftwareSerial.h>
SoftwareSerial mySerial(0, 1); // RX, TX

#define D2 2
#define D3 3
#define D4 4
#define D5 5
const int boton=6;

void setup()
{
    Serial.begin(9600);
    //mySerial.begin(9600);

    pinMode(D2, OUTPUT);
    pinMode(D3, OUTPUT);
    pinMode(D4, INPUT);
    pinMode(D5, OUTPUT);
    pinMode(boton, INPUT);

    digitalWrite(D2, HIGH);
    digitalWrite(D3, HIGH);
    digitalWrite(D5, HIGH);

    delay(2000);
    mySerial.println("Setup completed");
}

unsigned int spi_tx(unsigned int dato)
{
    int aux;
    unsigned int resultado;
    resultado=0xffff;
    digitalWrite(D2, HIGH);
    digitalWrite(D3, HIGH);

    digitalWrite(D2, LOW);
    digitalWrite(D3, LOW);
    aux=digitalRead(D4);
    if(aux) resultado=resultado&0xffff;
    else resultado=resultado&0x7fff;
    if(dato&0x8000) digitalWrite(D5, HIGH);
    else digitalWrite(D5, LOW);
    digitalWrite(D2, LOW);
    digitalWrite(D3, HIGH);
    aux=digitalRead(D4);
    if(aux) resultado=resultado&0xffff;
    else resultado=resultado&0x7fff;
```

```

if(dato&0x8000) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xbfff;
if(dato&0x4000) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xbfff;
if(dato&0x4000) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xdfff;
if(dato&0x2000) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xdfff;
if(dato&0x2000) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xefff;
if(dato&0x1000) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xefff;
if(dato&0x1000) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xf7ff;
if(dato&0x0800) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);

```

```

aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xf7ff;
if(dato&0x0800) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xfbff;
if(dato&0x0400) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xfbff;
if(dato&0x0400) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xfdff;
if(dato&0x0200) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xfdff;
if(dato&0x0200) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xfeff;
if(dato&0x0100) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xfeff;
if(dato&0x0100) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xff7f;
if(dato&0x0080) digitalWrite(D5, HIGH);

```

```

else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xff7f;
if(dato&0x0080) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffbf;
if(dato&0x0040) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffbf;
if(dato&0x0040) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffdf;
if(dato&0x0020) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffdf;
if(dato&0x0020) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffef;
if(dato&0x0010) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffef;
if(dato&0x0010) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);

```

```

if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffff7;
if(dato&0x0008) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffff7;
if(dato&0x0008) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffffb;
if(dato&0x0004) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffffb;
if(dato&0x0004) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffffd;
if(dato&0x0002) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffffd;
if(dato&0x0002) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffffe;
if(dato&0x0001) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
aux=digitalRead(D4);
if(aux) resultado=resultado&0xffff;
else resultado=resultado&0xffffe;
if(dato&0x0001) digitalWrite(D5, HIGH);
else digitalWrite(D5, LOW);

```

```

    digitalWrite(D2, HIGH);
    digitalWrite(D3, HIGH);
    return resultado;
}

float angulo(unsigned int dato)
{
    int resultado;
    float retorno;
    if((dato>=0)&(dato<=32767))
    {
        resultado=dato;
    }
    if((dato>=32768)&(dato<=65535))
    {
        resultado=dato-65536;
    }
    retorno=resultado*0.0054931640625; //(180/32,768) pagina 9 del data sheet
ADIS16210
    return retorno;
}

float temperatura(unsigned int dato)
{
    int resultado;
    float retorno;
    if((dato>=0)&(dato<=32767))
    {
        resultado=dato;
    }
    if((dato>=32768)&(dato<=65535))
    {
        resultado=dato-65536;
    }
    retorno=((float)(resultado-1278)*(-0.47))+25.0; //pagina 11 del data sheet
ADIS16210
    return retorno;
}

float voltaje(unsigned int dato)
{
    int resultado;
    float retorno;
    if((dato>=0)&(dato<=32767))
    {
        resultado=dato;
    }
    if((dato>=32768)&(dato<=65535))
    {
        resultado=dato-65536;
    }
    retorno=(float)(resultado*5.0)/32768.0; //pagina 11 del data sheet
ADIS16210
    return retorno;
}

void loop()

```

```

{
  unsigned int datoli, prod_id, xincl_out, yincl_out, zincl_out, supply_out,
  temp_out;
  float ang_x, ang_y, ang_z, vcc, temp;

  digitalWrite(D2, HIGH);
  digitalWrite(D3, HIGH);
  digitalWrite(D5, HIGH);
  delay(500);

  datoli=spi_tx(0x5600);
  delay(2);
  prod_id=spi_tx(0x0c00);
  delay(2);
  xincl_out=spi_tx(0x0e00);
  delay(2);
  yincl_out=spi_tx(0x1000);
  delay(2);
  zincl_out=spi_tx(0x0200);
  delay(2);
  supply_out=spi_tx(0x0a00);
  delay(2);
  temp_out=spi_tx(0x0560);

  ang_x=angulo(xincl_out);
  ang_y=angulo(yincl_out);
  ang_z=angulo(zincl_out);
  vcc=voltaje(supply_out);
  temp=temperatura(temp_out);

  mySerial.print("ID=");
  mySerial.print(prod_id, HEX);
  mySerial.print(", ");

  mySerial.print("X=");
  mySerial.print(ang_x, 4);
  mySerial.print(", ");

  mySerial.print("Y=");
  mySerial.print(ang_y, 4);
  mySerial.print(", ");

  mySerial.print("Z=");
  mySerial.print(ang_z, 4);
  mySerial.print(", ");

  mySerial.print("Volt=");
  mySerial.print(vcc, 1);
  mySerial.print(", ");

  mySerial.print("Temp=");
  mySerial.println(temp, 1);

  if (digitalRead(boton)==LOW)
  {
    mySerial.println("");
    mySerial.print(xincl_out, HEX);

```

```

mySerial.print(" ");
mySerial.print(yincl_out, HEX);
mySerial.print(" ");
mySerial.print(zincl_out, HEX);
mySerial.print(" ");
mySerial.print(supply_out, HEX);
mySerial.print(" ");
mySerial.println(temp_out, HEX);
mySerial.print(xincl_out, DEC);
mySerial.print(" ");
mySerial.print(yincl_out, DEC);
mySerial.print(" ");
mySerial.print(zincl_out, DEC);
mySerial.print(" ");
mySerial.print(supply_out, DEC);
mySerial.print(" ");
mySerial.println(temp_out, DEC);

```

```

String bufer="AT$SF=";
String bufer2="\n";
String str1;

```

```

byte* a1 = (byte*) &xincl_out;
for(int i=0;i<2;i++)
{
    str1=String(a1[i], HEX);
    if(str1.length(<2)
    {
        bufer+=0+str1;
    }
    else
    {
        bufer+=str1;
    }
}

```

```

byte* a2 = (byte*) &yincl_out;
for(int i=0;i<2;i++)
{
    str1=String(a2[i], HEX);
    if(str1.length(<2)
    {
        bufer+=0+str1;
    }
    else
    {
        bufer+=str1;
    }
}

```

```

byte* a3 = (byte*) &zincl_out;
for(int i=0;i<2;i++)
{
    str1=String(a3[i], HEX);
    if(str1.length(<2)
    {
        bufer+=0+str1;
    }
}

```

```

    }
    else
    {
        bufer+=str1;
    }
}

byte* a4 = (byte*) &supply_out;
for(int i=0;i<2;i++)
{
    str1=String(a4[i], HEX);
    if(str1.length(<2)
    {
        bufer+=0+str1;
    }
    else
    {
        bufer+=str1;
    }
}

byte* a5 = (byte*) &temp_out;
for(int i=0;i<2;i++)
{
    str1=String(a5[i], HEX);
    if(str1.length(<2)
    {
        bufer+=0+str1;
    }
    else
    {
        bufer+=str1;
    }
}
bufer+=bufer2;
mySerial.println(bufer);
Serial.print("AT\n");
Serial.print("AT$RC\n");
Serial.print(bufer);
}
delay(1000);
}

```

Anexo D Código fuente de la aplicación WEB

```
function doGet(e)
{
    var params = JSON.stringify(e);
    var parametersObject = JSON.parse(params);

    var timestamp = parametersObject.parameters.timestamp[0];
    var device = parametersObject.parameters.device[0];
    var anguloX = parametersObject.parameters.anguloX[0];
    var anguloY = parametersObject.parameters.anguloY[0];
    var anguloZ = parametersObject.parameters.anguloZ[0];
    var voltaje = parametersObject.parameters.voltaje[0];
    var temperatura = parametersObject.parameters.temperatura[0];

    var ss = SpreadsheetApp.openByUrl('https://docs.google.com/spreadsheets/d/ Borrado intencionalmente por seguridad /edit');
    var lastRow = ss.getLastRow();
    var lastCellA = 'A'+(lastRow+1);
    var lastCellB = 'B'+(lastRow+1);
    var lastCellC = 'C'+(lastRow+1);
    var lastCellD = 'D'+(lastRow+1);
    var lastCellE = 'E'+(lastRow+1);
    var lastCellF = 'F'+(lastRow+1);
    var lastCellG = 'G'+(lastRow+1);

    var d = new Date(0);
    d.setUTCSeconds(timestamp);
    var formattedDate = Utilities.formatDate(d, "GMT-6:00", "dd/MM/yyyy HH:mm:ss");

    ss.getRange(lastCellA).setValue(formattedDate);
    ss.getRange(lastCellB).setValue(device);
    ss.getRange(lastCellC).setValue(anguloX*0.0054931640625);
    ss.getRange(lastCellD).setValue(anguloY*0.0054931640625);
    ss.getRange(lastCellE).setValue(anguloZ*0.0054931640625);
    ss.getRange(lastCellF).setValue(voltaje*5/32768);
    ss.getRange(lastCellG).setValue(((temperatura-1278)*-0.47)+25);

    return HtmlService.createHtmlOutput(ss.getName()+ ' ' + lastRow + ' ' + lastCellA + ' ' + params + ' ' + parametersObject.parameters.temperatura[0]);
}
```

Referencias bibliográficas

Ada fruit (2018) 4-channel i2c-safe bi-directional logic level converter [Internet], **Ada fruit**. Disponible desde <<https://www.adafruit.com/product/757>> [Acceso 15 de marzo de 2018].

Analog Devices (2015) Precision triaxial inclinometer and accelerometer with SPI, **Hoja de datos**, Norwood USA.

Arduino (2018a) Arduino Home [Internet], **Arduino**. Disponible desde <<https://www.arduino.cc/>> [Acceso 15 de marzo de 2018].

Arduino (2018b) Arduino Software [Internet], **Arduino**. Disponible desde <<https://www.arduino.cc/en/main/software>> [Acceso 15 de marzo de 2018].

Arduino (2018c) Arduino UNO Rev 3 [Internet], **Arduino**. Disponible desde <<https://store.arduino.cc/usa/arduino-uno-rev3>> [Acceso 15 de marzo de 2018].

Atmel (2016) ATmega328/P Datasheet complete, **Hoja de datos**, San Jose USA.

FTDI Chip (2018) Virtual COM port drivers [Internet], **Future Technology Devices International**. Disponible desde < <http://www.ftdichip.com/Drivers/VCP.htm> > [Acceso 15 de marzo de 2018].

IoTnet (2018a) IoTnet México la red nacional de internet de las cosas [Internet], **IoTnet**. Disponible desde <<http://www.iotnet.mx/>> [Acceso 15 de marzo de 2018].

IoTnet (2018b) NXTIoT_DEVKIT [Internet], **GitHub**. Disponible desde <https://github.com/NXTIoT/NXTIoT_DEVKIT> [Acceso 14 de marzo de 2018].

Ruiz B. G. A., Valera O. B., Padilla O. S., González G. S., Alba A. L. A., Flores F. J. O., Mata Z. M. E., Saniger B. J. M., Gastélum S. A., Márquez F. J. A. y Sánchez V. J. (2017) Estudios y desarrollos preliminares para la restauración y conservación del Tlálóc de Coatlinchán, **Informe Técnico**, CCADET UNAM.

Sigfox (2018) Sigfox, the world's leading IoT connectivity service [Internet], **Sigfox**. Disponible desde <<https://www.sigfox.com/en>> [Acceso 15 de marzo de 2018].

Wisol (2016) Sigfox support product information [Internet], **Wisol**. Disponible desde <<http://support.wisol.co.kr/en/>> [Acceso 15 de marzo de 2018].