**Variable Analog and Digitals Monitoring using Tiva™ C and Ubidots**

**Monitorización de Variables Análogas Y Digitales Usando Tiva™ C Y Ubidots**

**Cesar A. Romero1, Andrés F. Jimenez2, Sandro D. Martinez 3**

1Universidad de los Llanos. *Cesar.romero@unillanos.com*; 2Universidad de los Llanos. a*ndres.jimenez@unillanos.com*; 3 Universidad de los Llanos. sandro.martinez*@unillanos.com*

**INFORMACIÓN DEL ARTÍCULO**

*Tipo*

*Investigación.*

*Historia*

Recibido: 30-07-2015

Correcciones:

Aceptado: XX-XX-2015

*Keywords*

Internet of Things, Ubidots,

Tiva™ C.

*Palabras clave*

Internet de las cosas, Ubidots,

Tiva™ C.

**ABSTRACT**

In this paper the design there wil be a data acquisition system in real time using the Evaluation Board Connected LaunchPad TM4C1294NCPDT a 32-bit microcontroller from Texas Instruments and Ubidots latter is a cloud service lets you store sensors information in real time, enabling the creation of applications for the ‘*Internet of Things’* (IoT) easily and quickly.

The acquisition of analog or digital signals from sensors of physical world which need to be managed, stored and transmitted in real time is first obstacle to overcome in applications oriented to‘Internet of things’ as currently acoording to Cisco Systema stadistics the global trend is that there will be more things connected to the cloud that people.

This article describes the process designing a data acquisition system in real time using the Connected Evaluation Board LaunchPad as hardware processing element and data transmission from the sensors and Ubidots as management software and storage sensor status. For the end application design process. They were defined three level: Level one Electronic focuses on sensors and microcontroller, level two focuses on communication, radios, Wi-fi, Bluetooth, zigbee, mobile network among others and last level focuses on need for use of databases, development, graphic design software, Apis.

**RESUMEN**

En este trabajo se realizara el diseño un sistema de adquisición de datos en tiempo real usando la Board de evaluación Connected LaunchPad con un microcontrolador de 32 bits TM4C1294NCPDT de Texas Instruments y Ubidots este ultimo es un servicio en la nube que permite almacenar y monitorear información de sensores en tiempo real, posibilitando la creación de aplicaciones para el internet de las cosas (IoT) de una forma fácil y rápida.

La adquisición de señales análogas o digitales provenientes de sensores del mundo físico las cuales requieren ser gestionadas, almacenadas y transmitidas en tiempo real es el primer obstáculo a vencer en aplicaciones orientadas para el internet de las cosas [[1](#Referencia01)] ya que en este momento según estadísticas de cisco system la tendencia mundial es que existiran más cosas conectadas a la red que personas.

Este articulo describe el proceso del diseño de un sistema de adquisición de datos en tiempo real usando la Board de evaluación Connected LaunchPad como elemento hardware de procesamiento y transmisión de los datos provenientes de los sensores y Ubidots como software para la gestión y almacenamiento del estado de los sensores. Para el proceso del diseño de la aplicación final se utilizó un enfoque en el cual se definieron tres niveles los cuales son: Nivel uno electrónica se enfoca en sensores y microcontroladores, nivel dos comunicaciones se enfoca en la comunicación radios WI-FI, Bluetooth, zigbee, red celular entre otras y un último nivel tres enfocado en el software necesario para el uso de bases de datos, desarrollo web, diseño gráfico, Apis.

© 2015. All rights reserved

1. **Introducción**

Con la aparición de sensores capaces de medir variables físicas se dio paso al monitoreo de sistemas lo cual es de gran ayuda para prevenir fallas y mejorar el funcioamiento de estos, hoy en dia por la gran cantidad de dispositivos conectados a internet se ha logrado monitorear sensores en tiempo real y almacenar la información obtenida en la nube por ejemplo las grandes ciudades están empezando a usar sensores que miden las vibraciones y el estado de los materiales de edificios, puentes, vías de comunicación y otras infraestructuras para evaluar su salud estructural y saber cuándo se deben hacer reparaciones [2] asi mismo las grandes industrias y hasta los hogares monitorean diferentes sensores cada vez con mas facilidad.

A continuación se describirá el método utilizado para realizar el sistema de adquisición de datos en tiempo real usando la Board de evaluación Connected LaunchPad, como elemento hardware de procesamiento, y transmisión de los datos provenientes de los sensores y Ubidots como el elemento software para la gestión y almacenamiento del estado de los sensores.

Para el proceso del diseño de la aplicación final se utilizó un enfoque en el cual se definieron tres niveles los cuales se observan en la tabla No. 1.

Tabla 1. Niveles de aplicación.

|  |  |
| --- | --- |
| Nivel 3 | Software.  Bases de datos, desarrollo web, diseño grafico APIs |
| Nivel 2 | Comunicaciones.  Radios, WI-FI, Bluetooth, zigbee, red movil, Ethernet |
| Nivel 1 | Electrónica.  Sensores y microcontroladores |

**2. Internet de las cosas.**

Actualmente y desde hace ya un tiempo vivimos en un mundo conectado en internet, la aparición de las redes sociales, blogs, wikis y muchos otros medios ha hecho que las personas compartan cada dia mas información y se comuniquen a través de la red el internet de las cosas permite que los objetos también puedan conectarse y compartir información.

El internet de las cosas (IoT) consiste en que las cosas tengan conexión a internet en cualquier momento y lugar. En un sentido más técnico, consiste en la integración de sensores y dispositivos en objetos cotidianos que quedan conectados a Internet a través de redes fijas e inalámbricas. [3]

En 2003, había aproximadamente 6.3 mil millones de personas en el planeta, y había 500 millones de dispositivos conectados a Internet. Si dividimos la cantidad de dispositivos conectados por la población mundial, el resultado indica que había menos de un dispositivo (0.08) por persona. De acuerdo con la definición de Cisco IBSG, el internet de las cosas aun no existía en 2003 por que la cantidad de cosas conectadas era relativamente escasa, dado que apenas comenzaba la invasión de los dispositivos omnipresentes, como los smartphones. Por ejemplo, el Director General de Apple, Steve Jobs, no presentó el iPhone sino hasta el 9 de enero de 2007 en la conferencia Macworld. [4]**.**

El número de cosas conectadas a internet sobrepasó en 2008 el número de habitantes del planeta. Se estima que habrá 50.000 millones de dispositivos conectados en 2020. [5].

**2.1 Computación en la nube (cloud computing) y computación obicua (ubiquitous computing).**

El cloud computing o computación en la nube se considera el esqueleto de Web 2.0. Son muchas las empresas que, por seguridad, ya no almacenan sus datos ni alojan sus webs en los servidores que tienen en el edificio donde ejercen su actividad, sino que contratan los servicios que ofrecen los grandes centros de datos del país o de países lejanos. [6] a pesar de que los grandes centros de información cuentan con las mayores medidas de seguridad son uno de los objetivos para los intrusos informáticos los cuales siempre están buscando desafiar su seguridad.

La computación obicua se caracteriza principalmente por tres factores: por una proliferación de tecnología embebida en dispositivos de multiple naturaleza, por la integración de la informática en el ambito personal (de forma que los ordenadores no se perciban como objetos diferenciados), y por el hecho de que nuesros datos o aplicaciones estén disponibles desde cualquier lugar. Esta tendencia da lugar al internet de las cosas donde el mundo físico se esta convirtiendo en un tipo de sistema de información, y estas redes permiten crear nuevos modelos de negocio, mejorar procesos y reducir costes y riesgos.[7]

**2.2 Niveles De Aplicación**

* + Nivel de Electrónica

Este nivel del diseño se centra en la programación del microcontrolador TM4C1294NCPDT de la serie Tiva™ C fabricado por Texas Instruments, las principales características de dicho controlador son poseer un núcleo ARM Cortes M4F operando a 120 MHz con una memoria flash de 1024 Kb[[8](#Referencia02)]; además este microcontrolador posee dos nucleos ADC de 12 bit cada uno operando a una frecuencia de muestreo máxima de 2.000.000 de muestras por segundo para la adquisición de señales análogas y 15 bloques fisicones GPIO para la adquisición de señales digitales, también dispone de una interfaz de comunicación Ethernet MAC 10/100 y Ethernet PHY con soporte para IEEE 1588.

El TM4C1294NCPDT puede ser programado en lenguaje C usando como herramientas de desarrollo Code Composer Studio™ (CCStudio) IDE, IAR Systems, KEIL™ Tools by ARM®, Sourcery™ CodeBench from Mentor Embedded, GNU, Atollic® TrueSTUDIO® y Energia.nu, lo que le proporciona una gran facilidad para adaptarse a cualquier aplicación.

En relación a las características eléctricas que deben tener las señales análogas o digitales para que sean correctamente adquiridas y procesadas en el microcontrolador se debe cumplir con las especificaciones técnicas del mismo las cuales definen que el voltaje nominal de entrada para el GPIO es 3.3V y una corriente que debe estar en el rango de 2mA a 12mA dependiendo el tipo de configuración del pin del GPIO usado; En relación al uso del conversor ADC para la adquisición de las señales análogas están deben de estar en el rango de 0V a VDDA para escala completa y voltaje de referencia interno, cuando se usa escala completa y voltaje de referencia externo el rango será de 2.4V a VDDA [[9](#Referencia03)].

En cuando al código que se debe programar en el microcontrolador para realizar la correcta gestión de los pines del GPIO, del conversor análogo digital y de la comunicación Ethernet se utilizó energuia.nu, energui.nu es un IDE multiplataforma basado y completamente compatible con wiring y arduino; El código utilizado en esta aplicación se basa en el diseño de un cliente web [[10](#Referencia04)] el cual hace uso de las librerías Ethernet.h y SPI.h de arduino, este código tiene tres funciones las cuales son void setup(), void loop() y void save\_value(String value,String idvariable, String titulo).

En la función void setup() se realiza la configuración del pin digital (PE\_4) a usar como entrada haciendo uso de la función pinMode(), también se configura el puerto serial a 9600 baudios con la función Serial.begin(), para finalmente realizar la verificación del establecimiento de la conexión de red mediante la función Ethernet.begin(mac).

En la función void loop() se realiza la lectura del pin análogo (PD\_3) mediante la función analogRead() y del pin digital (PE\_4) mediante la función digitalRead(PE\_4), también se realiza el llamado de la función void save\_value(String value,String idvariable, String titulo) la cual tiene por función el envió de datos a Ubidots del pin digital y análogo leído anteriormente. Para la descarga completa del código usado visitar http:// <https://goo.gl/EQojok>

* + Nivel de Comunicaciones

Al momento de realizar el análisis para la selección de la tecnología a utilizar en la transmisión de los datos entregados por los sensores[[17](#Referencia11)] se estudiaron diferentes tecnologías como zigbee, Bluetooth, red celular, Ethernet y radios WI-FI disponibles en las diferentes Board o plataformas de desarrollo así por ejemplo se evaluaron las Board chipKIT™ Wi-FIRE [[11](#Referencia05)], arduino Uno + Ethernet[[12](#Referencia06)] y Connected LaunchPad [[13](#Referencia07)]. Optando por utilizar la Connected LaunchPad debido a que esta posee un microcontrolador de 32 bits y un módulo Ethernet MAC y PHY con soporte para IEEE1588, además de ofrecer la opción de programming y debugging en la misma board a un bajo costo. En cuando al código que se debe programar en el microcontrolador para realizar la correcta comunicación entre Ubidots[[14](#Referencia08)] y el microcontrolador se utilizó energuia.nu donde lo primero que se debe hacer es asignar la dirección MAC la cual es proporcionada por el fabricante de la Board de desarrollo mediante la siguiente línea de código se asigna la dirección MAC:

|  |
| --- |
| byte mac[] = {0x00,0x1A,0xB6,0x02,0xE2,0x6E}; |

Luego de esta asignación se realiza la inicialización de un cliente Ethernet con dirección IP asignada por DHCP y conexión por el puerto 80 el cual es usado por defecto para comunicaciones HTTP[[15](#Referencia09)], esta inicialización del cliente Ethernet es un objeto de la librería Ethernet.h y se realiza mediante la siguiente línea de código:

|  |
| --- |
| EthernetClient client; |

Finalmente en la función setup() se realiza la validación de la conexión como ya fue expuesto en el nivel electrónica y si esta comunicación se realiza satisfactoriamente[[16](#Referencia10)] se imprimirá el texto “connecting…” por el puerto serial del microcontrolador.

* + Nivel de Software

El desarrollo de aplicaciones que leen información de sensores análogos o digitales proveniente del ambiente es lo que se conoce como el internet de las cosas y es una tendencia tecnológica que promete crear una nueva era de aplicaciones [[18](#Referencia12)].

Con los niveles electrónica y comunicaciones resueltos el último paso en el desarrollo de la aplicación es el diseño e implementación de un webserver en el microcontrolador para así subir a un servidor remoto la información recolectada por los diferentes sensores; Esto permitiría tener datos históricos y graficar las mediciones lo que implicaría trabajar con scripts en PHP, una base de datos en MySQL y el stack del microcontrolador lo cual involucra un alto conocimiento en el uso de herramientas software, pero este no es el único escenario de trabajo para la gestión y almacenamiento de los datos, otra alternativa de trabajo es el uso de Ubidots como herramienta de software para la gestión y almacenamiento de la información proveniente de los sensores. El uso de esta API simplifica el nivel de conocimiento software ya que con unos pocos pasos se logra él envió y gestión de la información.

Para comenzar a utilizar el API de Ubidots se presentara a continuación algunos conceptos de la plataforma:

Data Source: Una Data Source se refiere al dispositivo conectado. Cada Data Source puede tener uno o más sensores o variables. Por ejemplo un sistema de riego sería un Data Source y sus variables pueden ser la humedad del suelo y la temperatura.

Variable: Una variable es una serie de datos que cambia con el tiempo. Por ejemplo las variables en un Data Source llamado “vehículo” pueden ser la velocidad en Km/h y la posición GPS.

Value: Es el valor medido por el sensor en un instante de tiempo determinado. Por ejemplo la temperatura en una habitación es 35 grados Celsius.

Event: Son declaraciones que mantienen el formato “IF…ELSE…” y se activan en función del ultimo valor de una variable. Por ejemplo recibir un correo electrónico o SMS si la temperatura supera los 38 grados Celsius.

Widget: Son visualizaciones personalizadas de los valores de las variables que ayudaran a extraer información de la misma. Por ejemplo un widget llamado “Historial” puede mostrar en una línea de tendencia el comportamiento de la temperatura en una habitación en la última hora, sema, mes o año.

Una vez expuestos los conceptos anteriores lo siguiente es interactuar con el API de Ubidots, Ubidots ofrece un REST API a través del cual los Data Source, Variable y value pueden ser creados, leídos, editados y borrados. Ubidots utiliza el protocolo HTTP para implementar cuatro métodos HTTP GET, POST, PUT y DELETA en formato JSON.

En energuia.nu dentro de la función void save\_value(String value,String idvariable, String titulo) se establece la comunicación mediante la siguiente línea de código:

|  |
| --- |
| client.connect("things.ubidots.com", 80) |

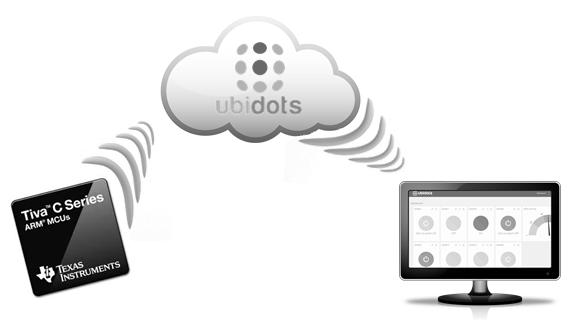
Una vez establecida la comunicación se pasa a realizar él envió del estado “value” de las variables que se requieren visualizar en el widget con el siguiente código:

|  |
| --- |
| client.println("POST /api/v1.6/variables/"+idvariable+"/values HTTP/1.1"); // envió del identificador de la variable.  client.println("Content-Type: application/json"); // conectando con la aplicación  client.println("Content-Length: "+String(num));  client.println("X-Auth-Token: "+token); // envoi del token  client.println("Host: things.ubidots.com\n");  client.print(var); |

Para la descarga completa del código usado visitar <https://goo.gl/EQojok>

**3. Resultados**

Como resultado del trabajo de investigación se presenta el diseño del sistema de adquisición de datos en tiempo real usando la Board de evaluación Connected LaunchPad y Ubidots. El funcionamiento general se observa en la figura 1.

**

*Figura 1. Funcionamiento.*

*Figura 2. Diagrama de bloques del sistema.*

**Monitorización y Control**

**Adquisición de Datos**

**Sensor**

**Microcontrolador**

**Alimentación**

**Nube (Ubidots)**

**Dispositivo con acceso a internet**

**Almacenamiento**

La figura 2 muestra un diagrama de bloques del software y hardware diseñado y utilizado en el diseño final. La forma en que un usuario final interactua con el sistema de adquisión es simple ya que este solo deberá conectar los sensores a las entradas digital y análoga de la board y luego iniciar sesión en ubidots para que el sistema realice el envio y visualización de los dados en el aplicativo web.

**4. Conclusion**

La mejora en las comunicaciones, el relativo bajo precio actual de muchos sensores y la popularización de distintas tarjeta hadware son algunos factores que han facilitado el monitoreo de variables físicas en tiempo real por medio de internet, y han hecho de esto una tendencia mundial utilizada tanto en los hogares como en las industrias, existen diferentes plataformas que brindan el servicio para la visualización de la adquisición de datos obtenidas por diferentes sensores , la información es almacenada en la nube y esto evita que esta se pierda fácilmente,Ubidots es una plataforma que brinda este servicio y es una excelente opción por su fácil comprensión y buen funcionamiento.

**5. Agradecimientos**

Este trabajo fue apoyado por la Universidad de los llanos (UNILLANOS) y el instituto de investigaciones de la Orinoquia colombiana (IIOC) y desarrollado por el Grupo de Investigación en Tecnologías Abiertas (GITECX).

**Referencias**

1. Evans, Dave (2012). [A Drop of Water Begins a Chain Reaction](http://blogs.cisco.com/ioe/a-drop-of-water-begins-a-chain-reaction-infographic-tomorrowstartshere?_ga=1.33419006.1168686822.1437726178). Computing Research, pp. 1-1
2. RIFKIN JEREMY. The Zero Marginal Cost Society. Palgrave Macmillan. United States of America 2014.
3. FUNDACIÓN DE LA INNOVACIÓN BANKINTER. El internet de las Cosas, En un mundo conectando de objetos inteligentes. 2011
4. DAVE EVANS. Internet de las cosas: cómo la próxima evolución de internet lo cambia todo. Cisco Internet Business Soutions Group (IBSG). Abril de 2011.
5. EVERLET A., PASTOR J. Introduccion al internet de las cosas, construyendo un Proyecto de iot. Carriots, Universidad Rey Juan Carlos Nov 2013.
6. Fundación Aguilera López. Seguridad informática. Editex 2010.
7. Josep Curto Díaz. Introducción al Business Intelligence. Editorial UOC 2012.
8. Texas Instruments (2014). [Tiva™ TM4C1294NCPDT Microcontroller DATA SHEET](http://www.ti.com/lit/ds/symlink/tm4c1294ncpdt.pdf). pp. 54-1890.
9. Texas Instruments (2015). [Tiva™ C Series TM4C1294 Connected LaunchPad Evaluation Kit](http://www.ti.com/lit/ug/spmu365b/spmu365b.pdf). pp. 6-43.
10. Arduino (2015). [Tutorial Web Client.](https://www.arduino.cc/en/Tutorial/WebClient) Pp. 1-1
11. Digilent ® (2015). [chipKIT Wi-FIRE](https://www.digilentinc.com/Products/Detail.cfm?NavPath=2,892,1266&Prod=CHIPKIT-WIFIRE).
12. Arduino (2015). [Arduino Board Ehternet](https://www.arduino.cc/en/Main/ArduinoBoardEthernet).
13. Texas Instruments (2015). [TM4C1294 Connected LaunchPad.](http://www.ti.com/tool/ek-tm4c1294xl)
14. Ubidots (2014). [About Ubidots.](http://ubidots.com/about-ubidots.html)
15. Tanenbaum, A. S; Redes de computadores; Tercera edición; Prentice-Hall; 1997.
16. Texas Instruments Incorporated (2015) [Virtual COM Port](http://www.ti.com/lit/ug/spmu365b/spmu365b.pdf). Pp. 24-43.
17. ACEDO, M., MOLINA, M., SILVA, R., MARCIANO, M. Y PORTILLA,R. [Revision de procesos de identificación de nodos en la wireless sensor networks.](http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/858/2112) CIDETEC-IPN. Departamento de Postgrado. Área de Telemática, Unidad Profesional Adolfo Lopez Mateos. C.P. 07700, México, D.F., MÉXICO,2009
18. YU, MISUN. [NanoMon: An Adaptable Sensor Network Monitoring Software](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4382116&queryText=NanoMon+&newsearch=true&searchField=Search_All). Electron. & Telecommun. Res. Inst. (ETRI), Daejeon.June 2007,Irving, TX,United States.