

Monitoreo De Sensores En Aplicaciones Web Embebidas

Avance en proyecto “Sistema de adquisición de imágenes multiespectrales para vehículos de uso en agricultura de precisión”

Andrés Felipe Gómez Rivera
Grupo de investigación MACRYPT
FCBI Universidad de los Llanos
Villavicencio, Colombia
andres.gomez.rivera@unillanos.edu.co

Fabián Velásquez Clavijo
Grupo de investigación MACRYPT
FCBI Universidad de los Llanos
Villavicencio, Colombia
fvelasquez@unillanos.edu.co

Msc. Andrés Fernando Jiménez López
Grupo de investigación MACRYPT
FCBI Universidad de los Llanos
Grupo de investigación UN-robot
Universidad Nacional de Colombia
ajimenez@unillanos.edu.co

Resumen— Este trabajo describe el proceso de diseño, implementación y evaluación de un sistema de monitoreo en tiempo real de sensores en una aplicación web embebida utilizando como plataforma hardware un ordenador de placa reducida basado en SoC ARM. Se describe la arquitectura de integración del hardware utilizado, el software de adquisición de variables físicas y el servidor web implementado para la generación de una interfaz de usuario accesible mediante un navegador web utilizando las tecnologías disponibles. Se evalúa el desempeño general del sistema y se presentan las ventajas y desventajas de la aplicación desarrollada. El sistema implementado presentado hace parte del desarrollo de un prototipo de cámara multiespectral para uso en agricultura de precisión.

Palabras clave — Monitoreo de sensors; sistemas embebidos; servidor web; ordenador de placa única, ARM

Abstract— This paper describes the design, implementation and evaluation process of a real time sensor monitoring system on an embedded web application using an ARM SoC based single board computer as hardware platform. The integration architecture of the used hardware, physical variables acquisition software and implemented web server for the generation of a user interface accessible through a web browser using the available technologies is described. The general performance of the system is evaluated and the advantages and disadvantages of the developed application are presented. The implemented system is part of the development of a multispectral camera prototype for use in precision agriculture.

Keywords— sensor monitoring; embedded system; web server; single board computer, ARM

I. INTRODUCCIÓN

La disponibilidad de plataformas de desarrollo embebidas de altas prestaciones, tales como los ordenadores de placa única y dimensiones reducidas han dado paso a una nueva tendencia de prototipado electrónico, aprovechando el bajo costo que supone su adquisición y la capacidad de trabajar sobre sistemas operativos como aquellos basados en linux, y utilizando lenguajes de alto nivel como python para controlar hardware de una manera más sencilla en comparación con los sistemas basados en lógica reconfigurable como FPGA y programables como microcontroladores. La posibilidad de usar

hardware de telecomunicaciones con protocolos como Wifi y tecnologías web dentro de estos sistemas embebidos ha permitido interconectar sensores a la red, realidad clave en el concepto de la tecnología emergente de Internet de las cosas [1], y que da paso a una mayor interactividad y facilidad de uso con el usuario al permitir su acceso desde virtualmente cualquier dispositivo provisto de un navegador web.

II. MÉTODO

El objetivo de este avance fue el diseño de una interfaz de usuario para la visualización de variables físicas medidas con sensores conectados en una plataforma embebida, mediante el uso de tecnologías web. Este avance se realizó con la intención de empezar con el desarrollo de la interfaz al usuario de la cámara multiespectral del proyecto principal, titulado “Sistema de adquisición de imágenes multiespectrales para vehículos de uso en agricultura de precisión”.

A. Materiales y dispositivos

Se utilizaron para el desarrollo de este avance una tarjeta Raspberry Pi 3, una tarjeta de sensores SensorHub BoosterPack de Texas Instruments y se construyó una tarjeta de interconexión entre las dos mencionadas.

1) Raspberry Pi 3

La Raspberry Pi 3 es una computadora de placa reducida cuyas principales características son:

- Procesador ARM Cortex-A53 de 64-bit de 4 núcleos a 1.2GHz.
- Memoria RAM de 1GB
- LAN inalámbrica 802.11n y bluetooth integrados.
- Puertos USB, Ethernet y GPIO

La familia de computadoras Raspberry Pi ha sido ampliamente utilizada por la comunidad académica para el prototipado de sistemas en robótica, visión artificial, teledetección, monitoreo de sensores, Internet de las cosas, entre otros, aprovechando su bajo costo y capacidad de funcionar con sistemas operativos basados en Linux con distribuciones como Raspbian.

2) BOOSTXL-SENSHUB Sensor Hub BoosterPack

Esta tarjeta (Fig 1), originalmente diseñada para acoplarse a una tarjeta de desarrollo Tiva launchpad de Texas Instruments consta de varios sensores ambientales y de movimiento:

- Sensor de temperatura de objetos TMP006.
- Sensor de movimiento de 9 ejes InvenSense MPU9150 (3 ejes de aceleración, 3 de velocidad angular, 3 de flujo magnético).
- Sensor de presión atmosférica Bosh BMP180.
- Sensor de luz ambiente e infrarroja Intersil ISL29023.
- Sensor de humedad SHT21.
- Dos Botones pulsadores para usuario.
- Un LED para usuario y uno de indicación de alimentación.
- Conectores de expansión para módulos inalámbricos

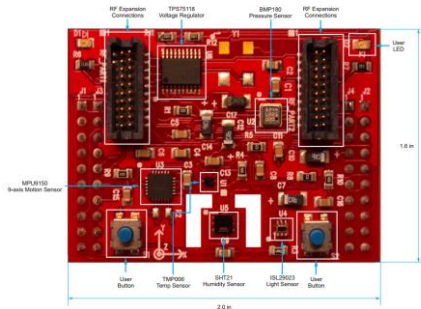


Fig. 1. BOOSTXL-SENSHUB Sensor Hub BoosterPack de Texas Instruments [2].

B. Arquitectura propuesta del sistema

El sistema, al estar construido sobre una plataforma embebida, debe contener todos los elementos para su funcionamiento, tanto hardware como software para comunicarse mediante protocolos web a un cliente externo. La arquitectura debe obedecer los requerimientos que comprende la inclusión de un servidor web con la capacidad de responder a solicitudes de información a un cliente externo y debe permitir la transmisión de información en tiempo real.

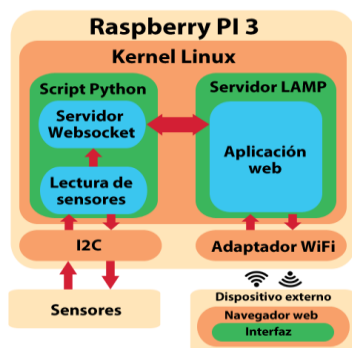


Fig. 2. Arquitectura general propuesta para el sistema

C. Procedimiento

El procedimiento del trabajo realizado a la fecha se expone a continuación:

1) Diseño de interfaz física para interconexión de tarjeta de sensores y tarjeta Raspberry Pi 3.

Para conectar las tarjetas de sensores junto con la Raspberry Pi se decidió diseñar y fabricar un circuito impreso. El diseño del circuito de interconexión (Fig.3) se realizó en el software Proteus ARES teniendo en cuenta las dimensiones de ambas tarjetas y las posiciones de los pines de alimentación y comunicaciones. Se incluyó un puerto de conexión auxiliar en caso de necesitarse expansión adicional en los puertos I2C y serial de la Raspberry.

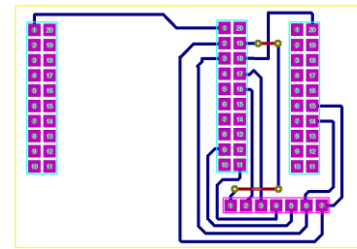


Fig. 3. Diseño de circuito impreso para interconexión de las tarjetas.

Una vez realizado el procedimiento de diseño CAD de la tarjeta, se imprimió con tóner con una impresora láser sobre papel recubierto (mejor conocido como propalcote) para realizar la transferencia con calor a una placa de circuito virgen a la que se le realizó previamente una limpieza y adecuación de la superficie de cobre. Al completar la transferencia del diseño al cobre se procedió a sumergir la placa en una mezcla de ácido clorhídrico, peróxido de hidrógeno y cloruro de sodio. Esta mezcla disuelve el cobre que no está cubierto por el tóner, dejando como resultado una placa de circuito lista para recibir componentes. En este caso se soldaron 4 conectores y dos jumpers. Se realizó el montaje del circuito para empezar a realizar la configuración y la programación (Fig. 4).



Fig. 4. Montaje sobre tarjeta Raspberry Pi.

2) Habilitación de bus I2C en Raspberry Pi

Luego de la instalación del sistema operativo Raspbian en la tarjeta Raspberry Pi 3, se configuró para que cargara los drivers I2C del kernel por defecto y poder acceder al puerto I2C. Esto se realizó mediante la ejecución del comando “sudo raspi-config”, que al ser ingresado en consola, habilita una interfaz de selección (Fig. 5) de opciones donde el usuario puede activar y desactivar funciones de configuración avanzada sobre el sistema operativo.

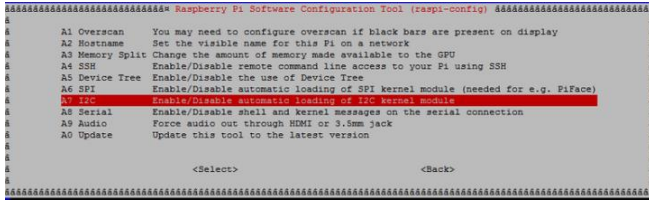


Fig. 5. Herramienta de configuración de Raspberry Pi

Se probó la conexión de la placa con la tarjeta Raspberry Pi y la tarjeta de sensores y se comprobó su correcto funcionamiento mediante el escaneo de dispositivos en el puerto I2C utilizando el comando de Linux en consola ‘sudo i2cdetect -y 1’. Al ingresar el comando aparece un mapa de direcciones en el cual se pudieron detectar todos los dispositivos presentes en la tarjeta de sensores.

3) Programación de lectura de sensores de intensidad luminica, inercial de 9 ejes y de temperatura ambiente y presión atmosférica:

Se hizo uso de la librería de Python SMBus para acceder al bus I2C de la Raspberry Pi. Primero se realizó el código de lectura del sensor de luz ambiente e infrarroja Intersil ISL29023. Este sensor entrega valores de niveles de iluminación en lux y el rango de funcionamiento (lux máximo) es seleccionable. Se crearon funciones para realizar la configuración y lectura del dispositivo y se creó un programa para observar los datos de nivel de iluminación en tiempo real.

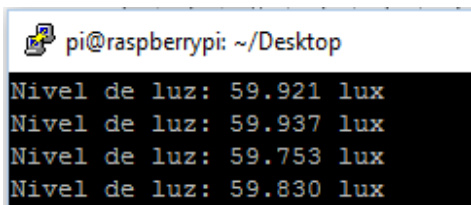


Fig. 6. Visualización de nivel de luz incidente en el sensor

Se realizó la prueba de lectura del sensor inercial InvenSense MPU-9150 mediante la ejecución de dos programas escritos en C disponibles en GitHub. El código fuente de ambos proyectos se transfirió a la Raspberry Pi 3 y se compiló mediante el comando “make”.

Se evaluó la capacidad de procesamiento de los datos provenientes del sensor en cada uno de los programas y se encontró que aunque el dispositivo es un sensor inercial de 9 ejes, uno de los códigos (MPU6050-Pi-Demo) procesaba

únicamente 6 ejes (acelerómetro y giroscopio) y en su salida no se indicaba el ángulo de la tarjeta en el eje z respecto al marco de referencia inercial de la Tierra (el cual está definido por la dirección de los vectores de aceleración gravitatoria y norte magnético) al no utilizar los datos del magnetómetro.

El otro código (linux-mpu9150) si proveía la información de inclinación de la tarjeta junto con la orientación respecto al norte magnético, pues integra las lecturas del magnetómetro. Se revisó la documentación del dispositivo y el software, notándose el uso del procesador digital de movimiento interno, el cual se activa mediante la inclusión en el código fuente de una librería binaria que se transfiere al dispositivo en el inicio de la ejecución del programa y activa el procesamiento de movimiento, permitiendo leer datos de posición ya calculados y reduciendo el uso de recursos computacionales en la Raspberry Pi 3.

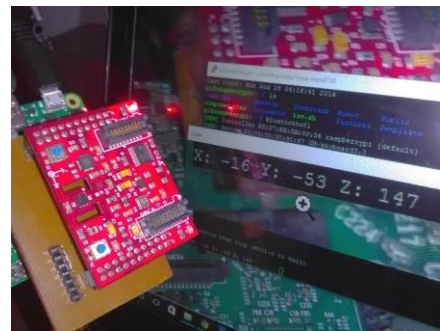


Fig. 7. Visualización de inclinación y giro del dispositivo en grados

Se espera modificar el código en C para que los datos de posición sean accesibles a través de la ejecución de un programa en Python.

Se escribió un programa en Python para la lectura del sensor de presión atmosférica y temperatura Bosh BMP180. Se escribieron funciones para determinar la temperatura, la presión atmosférica medida en pascales en distintos modos de funcionamiento (que definen la resolución de la lectura) y para hallar la altura sobre el nivel del mar. El sensor BMP180 cuenta con 11 coeficientes de calibración de fábrica que son constantes dentro de las funciones implementadas para hallar temperatura y presión. Estos parámetros son leídos al inicio del programa de prueba implementado y son utilizados en las ecuaciones de corrección sobre los valores no compensados de temperatura y presión disponibles tras el envío de una orden de lectura al sensor.

La ecuación sugerida por la hoja de datos del sensor para hallar altura a nivel del mar se muestra a continuación:

$$\text{altitude} = 44330 * \left(1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5.255}} \right)$$

Esta ecuación fue implementada en el código para hallar altura sobre el nivel del mar, basado en la presión atmosférica

leída. El término p_0 corresponde a la presión atmosférica a nivel del mar. Este término fue hallado mediante el despeje de la ecuación anterior, conociendo la altura sobre el nivel del mar del lugar en el cual se realizó una serie de lecturas de presión atmosférica.

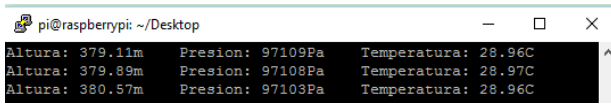


Fig. 8. Visualización de valores de presión, altura y temperatura.

4) Instalación de servidor LAMP y framework Laravel en Raspberry Pi:

LAMP es el acrónimo de Linux, Apache, MySQL y PHP. Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Se instaló un servidor web en la Raspberry Pi mediante la ejecución de los siguientes comandos:

```
“sudo apt-get install mysql-server mysql-client apache2
apache2-utils php5 php5-cli libapache2-mod-php5 php5-
mysql php5-curl php5-gd php-pear php5-imagick php5-
mcrypt php5-memcache php5-mhash php5-sqlite php5-xmllrpc
php5-xsl php5-json php5-dev libpcre3-dev”
“sudo curl -sS https://getcomposer.org/installer | php”
“sudo ~/composer.phar global require "laravel/installer" ”
```

Y se creó un nuevo proyecto de laravel con el comando:

```
“composer create-project --prefer-dist laravel/laravel
apsensorshow”
```

Se le dieron los permisos correctos al sistema operativo para el acceso a la carpeta en la cual se instaló el proyecto.

A partir de este punto fue posible escribir páginas web en la Raspberry Pi que son accesibles a través de un navegador web, accediendo a su dirección IP.

5) Implementación de websocket en Python

Se implementó sobre el código de lectura del sensor de temperatura y barométrico un servidor websocket. Websocket es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP [3]. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor. El código se adecuó para enviar a un cliente los valores de temperatura y altura sobre el nivel del mar en una estructura JSON cuando el cliente requiera la información. JSON es un formato de texto ligero para el intercambio de datos [4]. Un ejemplo de estructura JSON enviado por el programa de prueba es:

```
{“temperatura”: 28.47, “altura”: 420.14}
```

6) Diseño de interfaz web para visualización de datos en tiempo real:

Se realizó un código en PHP alojado en el servidor, con sus respectivos apartes HTML y uso de jQuery de JavaScript para acceder a la información de los sensores y visualizarla en dos gráficos en tiempo real. jQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web [5]. Se utilizó la librería de jQuery flot como herramienta de visualización de gráficos. Se ingresó desde un navegador web de un dispositivo externo conectado a la misma red que la Raspberry Pi y se accedió a la interfaz indicando la dirección IP de la tarjeta. A continuación se muestra la interfaz sobre el navegador web:

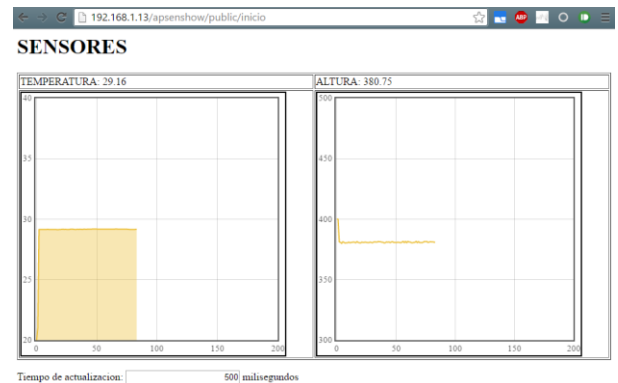


Fig. 9. Sitio WEB con visualización en tiempo real de valores de sensores

III. CONCLUSIONES

El ordenador de placa única Raspberry Pi 3 ha probado ser muy versátil para la integración de sensores, permitiendo su programación a través de aplicaciones desarrolladas en varios lenguajes de programación y su interacción con el hardware de comunicaciones integrado.

La plataforma Raspberry Pi 3 es idónea para la implementación de un servidor web para el monitoreo de sensores en tiempo real y para demostrar la capacidad de un sistema embebido para implementar tecnologías web sin depender de infraestructura de red o computacional externa.

El uso de tecnologías como websocket facilita la interacción entre programas, haciendo que el diseño de un sistema sea más modular y fácil de mantener que si se realizara en un solo entorno.

IV. TRABAJO FUTURO

Se realizará la integración de los demás sensores presentes en la tarjeta y se realizarán las primeras pruebas con sensores de imagen, permitiendo al usuario capturar fotografías por demanda a través de la interfaz web, realizando anotaciones de posición y variables ambientales al momento de capturar la imagen.

RECONOCIMIENTOS

Para la lectura del sensor inercial MPU9150 se utilizó código disponible en los repositorios GitHub de los usuarios Richard Hirst y Víctor Mayoral Vilches.

Los respectivos enlaces a los repositorios son <https://github.com/richardghirst/PiBits/tree/master/MPU6050-Pi-Demo> y https://github.com/vmayoral/bb_mpu9150/tree/master/src/linux-mpu9150.

REFERENCIAS

- [1] Shelby, Z. (2010). Embedded web services. IEEE Wireless Communications, 17(6), 52-57.
- [2] BOOSTXL-SENSHUB Sensor Hub BoosterPack User Manual, Texas Instrument (2013).
- [3] Pimentel, V., & Nickerson, B. G. (2012). Communicating and displaying real-time data with WebSocket. IEEE Internet Computing, 16(4), 45-53.
- [4] Crockford, D (2006). "The application/json media type for javascript object notation (json)."
- [5] De Volder, K. (2006, January). JQuery: A generic code browser with declarative configuration language. In International Symposium on Practical Aspects of Declarative Languages (pp. 88-102). Springer Berlin Heidelberg.