

MONITOREO DE VARIABLES ANALÓGICAS CON RASPBERRY Pi B+

Vignolo, Enzo E.*; Valdez, Alberto D.*; Miranda, Carlos A.*; Schlesinger, Paola L.*; Chiozza, Juan A.*; Miranda, Carlos V.*

Resumen

En este documento se explica el diseño y realización de un sistema prototipo basado en Raspberry Pi B+ para la captura, almacenamiento, procesamiento y presentación de variables analógicas.

Palabras claves: computadora, mini, sensores, base de datos, procesamiento, programación

INTRODUCCIÓN

La plataforma Raspberry Pi, es un mini computador basado en procesadores de arquitectura ARM, el cual tiene como software de fábrica un sistema operativo basado en Linux, llamado Raspbian, aunque es posible obtener distintas distribuciones oficiales y no oficiales dependientes del proyecto que se quiera llevar a cabo[1][2].

El bajo costo, el bajo consumo, la flexibilidad de su software y el soporte de una gran comunidad, son algunas de las características por las cuales el Raspberry Pi es una de las plataformas más populares al momento del desarrollo de

proyectos de procesamiento de datos.

La detección de magnitudes físicas mediante sensores, se facilitó en el transcurso del tiempo gracias a la aparición de microcontroladores y de plataformas como la del Raspberry Pi. Por lo cual resulta de interés tener un sistema que permita de manera flexible el almacenamiento y la presentación de los datos obtenidos con estos sensores.

DESARROLLO

Raspberry Pi B+

Existen diferentes modelos de Raspberry Pi, en los cuales existen variaciones tanto de potencia, de tamaño y de posibilidad para conectar mayor o menor número de periféricos.

El desarrollo del prototipo se realizó en un Raspberry Pi modelo B+. Este posee las siguientes características técnicas [3]:

- 4 entradas USB
- Un puerto Ethernet
- Un puerto HDMI

(*) Facultad de Ciencias Exactas y Naturales y Agrimensura, Departamento de Ingeniería, UNNE, Corrientes CP3400, Argentina. dvaldez@exa.unne.edu.ar

- Un conector de salida de audio de 3.5mm
- Un conector serial para cámaras (CSI)
- Un conector serial para display (DSI)
- 40 pines de propósito general de entrada y salida (GPIOs)
- Un lector de tarjetas microSD

El modelo B+ es alimentado mediante un puerto micro-USB de 5V con una fuente de poder de 1.8A como capacidad máxima. El consumo de corriente es dependiente tanto del consumo del microprocesador como de los periféricos a conectarse en la placa. Los

puertos USB son capaces de soportar hasta 600mA cada uno, y los pines GPIOs permiten entregar hasta 50mA en total.[4]

GPIOs

Los pines de propósito general de entrada y salida o GPIOs representan una parte fundamental del Raspberry Pi ya que mediante ellos se permite la interacción con distintos actuadores, así como la entrada y salida de datos. Los GPIOs además, son capaces de proveer referencias de tensión de 5V, 3.3V o tierra (GND). El modelo B+ posee 40 pines de propósito general de entrada y salida, los cuales se distribuyen de la siguiente manera sobre la placa:

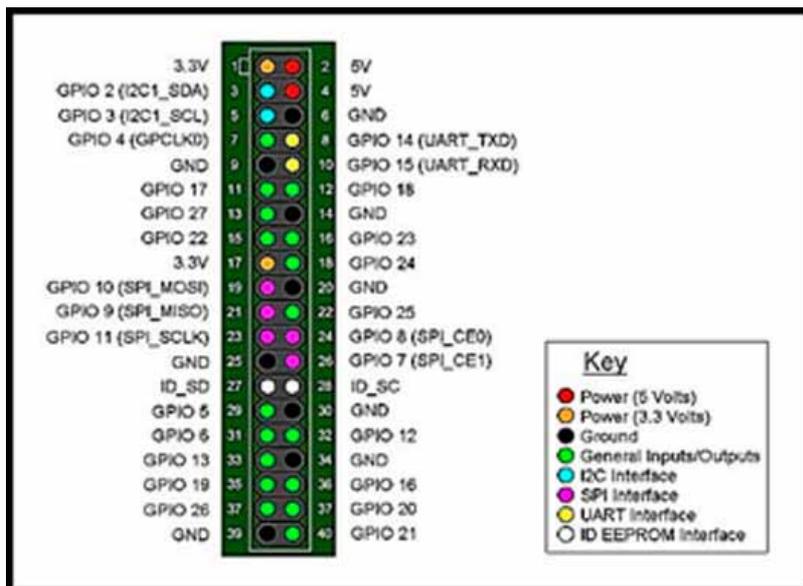


Figura 1 – Disposición de conectores de propósito general de entrada y salida[3].

Como puede observarse en la *figura 1*, existen 4 conectores de alimentación (dos de 5V y dos de 3.3V), 8 pines de tierra, dos pines especiales para la lectura de una ID EEPROM y 26 pines que pueden utilizarse como entrada o salida de datos digitales. De estos 26 pines utilizados como entrada y

salida de datos generales, existen algunos de ellos que pueden utilizarse también como bus de comunicación con periféricos de distintos protocolos, como I2C, SPI y UART.

Debido a que Raspbian es un freeware (software de licencia gratuita), a la hora

de la programación, la disposición lógica dependerá de que librería se utilicen para controlar los GPIOs. En la figura 1 puede observarse la disposición lógica para la librería GPIOs compatible con Python.

Software

Los sistemas operativos que pueden utilizar los Raspberry Pi son variables, no dependen del modelo y son montados en una memoria flash MicroSD que se coloca en el lector de tarjetas MicroSD del Raspberry Pi.

El sistema operativo utilizado para el desarrollo del prototipo es el Raspbian, un sistema basado en Linux, el cual es una de las alternativas oficiales que se ofrecen[2][4]. Este sistema cuenta con varias herramientas preinstaladas para el desarrollo de programas y scripts en diferentes lenguajes. Algunos de ellos son C, C++, Python, Ruby, etc[4][5]. Los scripts del prototipo realizado se encuentran escritos en Python 2.7. Se eligió Python ya que cuenta con un gran soporte de librerías específicas para Raspberry, permite la interacción con otras aplicaciones de manera sencilla y también debido a su facilidad de aprendizaje.

El Raspberry Pi modelo B+ puede conectarse a internet mediante su puerto Ethernet o bien mediante el uso de una placa WiFi USB. Esto permite la descarga de diferentes paquetes y aplicaciones para el desarrollo del proyecto.

Prototipo

El prototipo de monitoreo de variables analógicas se basó en tres funciones que se cumplen de manera cíclica:

- Captura y almacenamiento de da-

tos: realiza la conversión analógica a digital de la tensión a medir y se los almacena en un servidor MySQL.

- Procesamiento de datos: se grafican los datos obtenidos.

- Presentación de datos: mediante un servidor HTTP se presentan los datos obtenidos.

Captura y almacenamiento de datos

Si bien el Raspberry Pi posee muchos pines de propósito general que pueden funcionar como entrada de datos, estos solo permiten datos digitales, por lo que primero se debe realizar una conversión analógica digital de la tensión que pudiese llegar desde el sensor al Raspberry.

Para este proyecto se utilizó el conversor MCP3208, el cual consta de 8 canales para la conversión de tensión analógica a digital[5][6]. El circuito integrado MCP3208 es un conversor de 12 Bits de resolución por aproximaciones sucesivas, que permite obtener hasta 100 kilo muestras por segundos y posee una interface de comunicación SPI. Este cuenta además con dos métodos de conversión, uno simple y el otro diferencial.

La programación del script que permite la conversión, debe tener en cuenta que tanto la selección del modo de conversión como la selección del modo de canal en el cual ingresa la tensión analógica, se realiza con el envío de un tren de 4 bits, en el cual el primero representa el modo y los tres restantes representan el canal. Para el sistema diseñado se elige el modo normal de conversión, el cual se logra enviando un bit en 0, y el número de canal se forma con los tres bits restantes que van desde el canal 0 (000) al 7 (111).

Además debe tenerse en cuenta que el resultado de la conversión consta de 14 bits, de los cuales el primero es un bit nulo que puede quedar tanto en 1 o 0, por lo que hay que descartarlo a fin de que no modifique la lectura; el segundo es un bit nulo que siempre se en cuenta en estado bajo, por lo cual se lo puede omitir; y luego se leen los 12 bits de conversión, empezando por el bit más significativo.

Para lograr el monitoreo de una variable analógica y las variaciones que esta sufre en el tiempo, es necesario llenar una base de datos con los resultados obtenidos en la etapa de captura de datos. Para lograr esto, la combinación del uso entre un minicomputador como el Raspberry y Python como lenguaje de programación permite varias alternativas que se ajustan a la finalidad del proyecto a desarrollar.

El caso del prototipo realizado, está pensado para variables de las cuales se requieren registros las 24 horas del día y que se pueda realizar un estudio posterior de su variación en el tiempo en distintas épocas del año. Por estas razones, se decidió montar en el Raspberry un servidor MySQL y armar una base de datos relacionales.

Una base de datos relacionales es aquella que cumple con el modelo relacional, se encuentra compuesta por varias tablas o relaciones y cada tabla a su vez se compone de un conjunto de registros ordenados en filas y columnas. Para el sistema desarrollado, se conformó la base de datos creando una tabla por día, cuyo nombre es la fecha del mismo, y dentro de la cual se almacenan la medición de la variable analógica y la hora a la cual esta fue medida.

MySQL es un sistema gestor de base de

datos muy conocido y ampliamente utilizado debido a su sencillez y a su buen rendimiento [7]. El mismo se gestiona mediante el lenguaje SQL (Structured Query Language), el cual es sencillo de comprender ya que se trata de alto nivel. La gestión de datos en MySQL se realiza mediante la comunicación entre un servidor (el cual puede ser local o remoto) y un cliente. Esta desarrollado en C++ y cuenta con una API (application programming interface) que permite al mismo interactuar con otro software [7].

Resulta sumamente sencillo la instalación del servidor MySQL en Raspberry y la configuración del mismo. Una vez conectados a internet, en la consola del Raspbian debe colocarse:

```
$sudo apt-get install mysql-server mysql-client
```

Durante el proceso de instalación, se pedirá la configuración de una contraseña la cual será utilizada para ingresar con el cliente al servidor. Una vez instalado este, se ingresa al mismo desde la consola de Raspbian colocando:

```
$mysql -u root -p
```

A lo que le seguirá una petición con la contraseña configurada. Una vez ingresada la contraseña, se podrá controlar el servidor de base de datos para realizar modificaciones sobre la misma. Para el sistema desarrollado, se crea una base de datos con el nombre "Mediciones", la cual será utilizada para el almacenamiento de las tablas de datos. Para crear la base de datos, basta con ingresar en el servidor MySQL el comando:

```
>create database Mediciones;
```

Luego de la instalación del servidor, se procede con la instalación de la librería que permite la interacción entre Python y el servidor de base de datos. Esto se logra colocando en la consola del Raspbian:

```
$sudo apt-get install Python-mysqldb
```

Una instalada la librería, basta con realizar un script en Python el cual se conecte a la base de datos, cree las tablas correspondientes y almacene los datos obtenidos en la etapa de captura de datos.

Procesamiento de datos

La referencia a la etapa de procesamiento de datos, no es más que la creación de un gráfico diario que permita una observación más detallada del cambio que sufre la variable analógica a monitorear. Existen diferentes modos de realizar gráficos científicos con Python [8], que permitan una correcta observación de los datos. En el caso del prototipo desarrollado, se optó por una plataforma conocida como Plotly.

Plotly es una plataforma que permite la creación de gráficos interactivos en la nube [9]. Esto significa que basta con pasar los datos al servidor de Plotly donde se realizarán los gráficos. Existen ventajas y desventajas a la hora de la utilización de Plotly:

Ventajas:

- Libera de procesamiento al Raspberry
- Los gráficos y datos pueden ser observados por internet
- Gráficos dinámicos y con buena visual
- Cuenta con una API compatible con Python

Desventajas:

- Necesita conexión a internet
- Las cuentas gratis permiten solo 50 gráficos por día

Para la instalación de las librerías que Python utiliza para conectarse con Plotly, primero debe instalarse un gestor de paquetes de Python conocido como pip, lo cual se logra ingresando en la consola de Raspbian:

```
$sudo apt-get install pip
```

Y luego se instala la librería ingresando en la consola[8]:

```
$pip install plotly
```

Ya instaladas las librerías necesarias, se debe crear una cuenta en la página de Plotly, con la cual se obtiene una clave de autenticación para lograr la conexión entre el Python y los servidores de Plotly mediante la API.

El sistema desarrollado utiliza un script que se programó con una cuenta gratuita de Plotly, con el cual se conecta con el servidor de este y se pasan los datos a graficar y los parámetros del estilo de gráfico que se requieren. Plotly permite realizar distintos tipos de gráfico que pueden adaptarse según los requisitos que debe cumplir el proyecto a desarrollar. El gráfico realizado con el prototipo es un gráfico de dispersión, en el cual se realiza un gráfico de amplitud de tensión contra tiempo en el que se capturó la muestra.

El gráfico realizado puede verse en la página de Plotly ingresando con la cuenta de usuario creada:

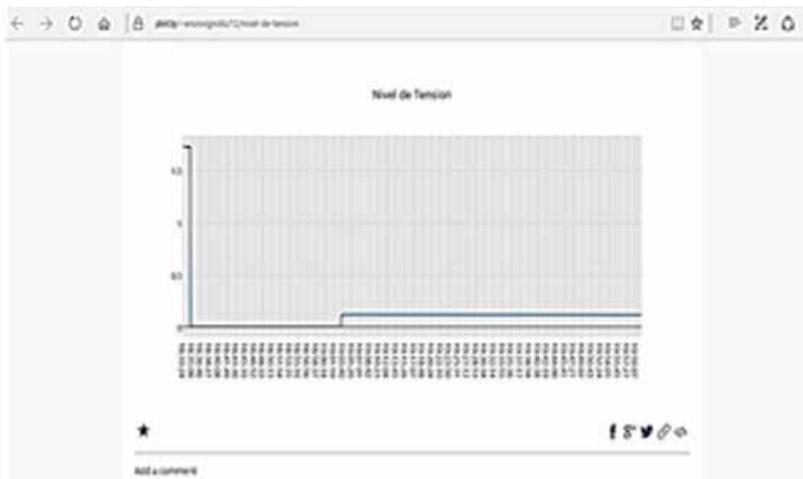


Figura 2 – Gráfico obtenido visto desde la página de Plotly

Presentación de datos

La posibilidad de trabajar con un Raspberry Pi corriendo un sistema basado en Linux como lo es Raspbian, facilita la presentación de los datos obtenidos en las etapas previas, mediante la instalación de un servidor HTTP. Una vez más, existen muchas variedades de servidores HTTP que corren bajo sistemas basados en Linux. En particular para el prototipo de monitor de variables analógicas, se eligió Lighttpd.

Lighttpd es un servidor web sencillo y liviano, lo cual lo hace ideal para trabajar con el Raspberry Pi. Para instalarlo basta con correr en la terminal de Raspbian el siguiente comando[10]:

```
$sudo apt-get install lighttpd
```

Una vez instalado, se podrá verificar su instalación ingresando mediante un navegador web a la dirección “http://x.x.x.x:80” reemplazando x.x.x.x por la dirección IP del Raspberry en la red.

Las librerías de Plotly utilizadas para realizar los gráficos, permiten obtener la

dirección URL del gráfico y así poder mostrarlo en nuestro servidor web. Se creó un script que al ejecutarse obtiene la dirección URL del último gráfico realizado y lo reemplaza en el servidor web del Raspberry Pi.

Ahora ingresando con cualquier dispositivo que se encuentre dentro de la misma red del Raspberry puede visualizar el gráfico de los datos capturados con solo ingresar la dirección ya mencionada en un navegador web.

Automatización de mediciones

Por último queda explicar que para lograr automatizar estos procesos y realizar mediciones las 24 horas del día, los sistemas basados en Linux poseen una herramienta muy útil llamada crontab. Crontab es una lista de tareas que debe realizar el sistema, programadas por minuto, hora, día de la semana y por si se desea, cada vez que se inicia el sistema operativo.

Con crontab se programó que los *scripts* creados para las etapas de captura de datos y almacenamiento de datos se ejecuten cada minuto del día, y que tanto los

de procesamiento como los de presentación se ejecuten una vez cada 15 minutos.

CONCLUSIONES

Como conclusión del trabajo realizado, se tiene que si bien el Raspberry Pi nació como una minicomputadora que ayude a la hora de la enseñanza y aprendizaje de programación, su gran potencial y flexibilidad permite la realización de proyectos complejos ya sean destinados al área científica o comercial.

Además cabe destacar que debido a que el Raspberry está basado en un sistema de código abierto, existe una vasta cantidad de herramientas que pueden conseguirse de manera gratuita, o por un

bajo costo, y siempre es necesario realizar una investigación de estas en base a los requisitos que exige el proyecto a realizar.

Continuación del proyecto

El prototipo de monitor de variables analógicas con Raspberry Pi sigue en desarrollo. En un futuro cercano se tiene como objetivo diseñar y construir una fuente de poder de respaldo para el Raspberry Pi, ya que este demostró inestabilidades cuando ocurre un corte abrupto de energía. Además se analizará la mejor manera de publicar el servidor que corre en el Raspberry para poder ingresar a los datos de manera remota a través de internet.

BIBLIOGRAFÍA

- [1]Caprio, Candela, Armendáriz. *Utilización de la Minicomputadora Raspberry Pi para la Adquisición y Evaluación de Datos de Niveles de Radiación Solar*. 2014. <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/25442>
- [2]Matt Richardson & Shaw Wallace, *Getting Started with Raspberry Pi*. Maker Media Inc. 2013.
- [3]<http://www.rs-online.com/designspark/electronics/eng/blog/introducing-the-raspberry-pi-b-plus>
- [4]Sitio Oficial de Raspberry Pi, www.raspberrypi.org
- [5]Microchip, MCP3208 Datasheet. 2002.
- [6]Escobar, Oñates, Yévenes. *Proyecto Raspberry Pi*. 2011.
- [7]Casillas Santillán, Giber Ginestá. Pérez Mora. *Base de datos en MySQL*. UOC. 2006.
- [8]Sitio Oficial Python, www.python.org
- [9]Sitio Oficial Plotly, www.plotly.org
- [10]Sitio Oficial Lighttpd, www.lighttpd.net