

RASPBERRY PI INTERFACES PROGRAMABLES.

RODRIGUEZ JUAN, MEDINA ALFONSO, MALDONADO CHRISTIAN, QUINTERO VERONICA Y DIAZ ARNOLDO

Resumen - Existe una nueva tendencia en el campo del desarrollo de software y en la programación de dispositivos, se trata de trabajar con una computadora de placa reducida; que cuenta con atributos de hardware que ningún otro dispositivo tan pequeño tiene. El potencial en las características que posee son una herramienta en el campo del desarrollo de software y el entretenimiento, este dispositivo cuenta con atributos para realizar interconexiones programables usando lenguaje estructurado en C con otros dispositivos. Trabaja con un sistema operativo basado en Linux, pero tiene su propio software para comunicación con redes inalámbricas la cuales compatible con muchos tipos de antenas USB, solo se tiene que tener los drivers previamente cargados.

También cuenta con su propio centro de software, el cual se conecta al repositorio de Raspberry [2] donde contiene distintas aplicaciones creadas para el sistema operativo de esta computadora, cabe mencionar que existen diferentes distribuciones de sistemas operativos compatibles con este dispositivo con características parecidas; pero conciertos parámetros gráficos y funcionales que varían. En este documento se analiza la forma de trabajar, programar y el uso de este dispositivo mediante prácticas realizadas en el Mexicali, aquí se explicaran las prácticas de un cubo de leds y un seguidor de línea.

Palabras Claves: Rasperry pi, programación, Linux, Lenguaje estructurado, parámetros gráficos, placa reducida.

I. OBJETIVO

Por medio de la programación y conocimientos básicos en electrónica se pudo demostrar como el dispositivo Raspberry PI tiene un potencial que solo se limita por nuestra creatividad, ya que desarrollamos varios proyectos de redes, brazos robóticos, seguidores de línea, matrices de leds sincronizados, etc. El dispositivo a pesar de que tiene y a tiempo en el mercado no es muy conocido.

MALDONADO LOPEZ CHRISTIAN ALBERTO a10490348@itmexicali.edu.mx
MEDINA DURAN ALFONSO a10490351@itmexicali.edu.mx
RODRIGUEZ QUIROZ JUAN a10490362@itmexicali.edu.mx
QUINTERO ROSAS VERONICA veroquintero@yandex.ru
DIAZ RAMIREZ ARNOLDO adiaz@itmexicali.edu.mx

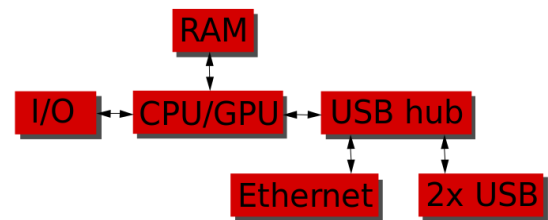


Figura1: Diagrama a bloques raspberry pi

II. RASPBERRY PI

Raspberry Pi es un ordenador de placa reducida con un costo muy bajo, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño contiene un procesador central (CPU) ARM1176JZF-Sa700MHz (el firmware incluye modos "Turbo" para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) Video Core IV, y 512 MB de memoria RAM aunque originalmente al ser lanzado eran 256MB.

El diseño no incluye una unidad de memoria de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente [3]; tampoco incluye fuente de alimentación carcasa. El 29 de febrero de 2012 la fundación empezó a aceptar órdenes de compra del modelo "B", y el 4 de febrero de 2013 del modelo "A" [12], [10].

ESTUDIANTE DE INGENIERIA ITM
ESTUDIANTE DE INGENIERIA ITM
ESTUDIANTE DE INGENIERIA ITM
PROFESORA DPTO SISTEMAS ITM
PROF INVESTIGADOR DPTO SISTEMAS ITM



Figura2: Modelo B de Raspberrypi.

A pesar de que el Modelo A no tiene un puerto RJ45, se puede conectar a una red usando un adaptador USB-Ethernet suministrado por el usuario. Por otro lado, a ambos modelos se les puede conectar un adaptador Wi-Fi por USB, para tener acceso a redes inalámbricas o internet.

El sistema cuenta con 256MB de memoria RAM en su modelo "A", y el modelo "B" con 512 MB de memoria RAM. Como es típico en los ordenadores modernos, se pueden usar teclados y ratones con conexión USB compatible con Raspberry Pi.

Raspberry Pi [3] no incluye reloj en tiempo real, por lo que el sistema operativo debe usar un servidor de hora en red; o pedir al usuario la hora en el momento de arrancar el ordenador. Sin embargo se podría añadir un reloj en tiempo real (como el DS1307) con una batería mediante el uso de la interface I²C.

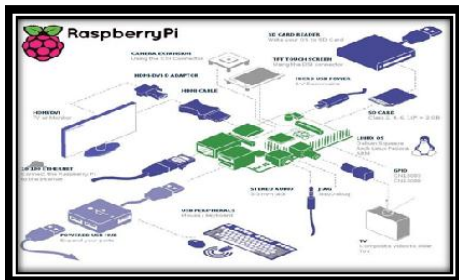


Figura3: Tipos de dispositivos para interconectar a Raspberry.

III. TIPOS DE SISTEMAS OPERATIVOS SOPORTADOS

Existen diversos sistemas operativos diseñados exclusivamente para el dispositivo Raspberrypi [6], pero no es posible utilizar sistemas operativos comercializados y no especializados ya que el procesador que este dispositivo maneja no es compatible, pero no hay problema, los sistemas operativos especializados de Raspberrypi son muy eficientes y trabajan a la perfección basados en arquitectura Linux [4].

Raspeberry es idóneo para cualquier situación, los comandos para realizar las tareas básicas desde consola se asemejan a Ubuntu.

Entre los distintos sistemas operativos que existen para este dispositivo destacan *Arch Linux*, *Raspbian*, *RiscOs*, *Pidora*, *RaspBmc* y *OpenElec* [4].

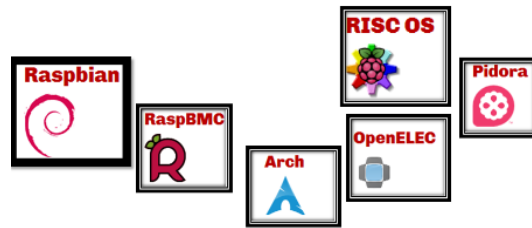


Figura4: Sistema especializado en Raspberry [4].

IV. ENTORNO DE DESARROLLO.

Entre los distintos lenguajes de programación que este dispositivo puede interpretar:

- a) C
- b) Python
- c) C++
- d) PHP

También se pueden manejar bases de datos y poder administrar las para cada lenguaje que se use, la Raspberry Pi dispone de un puerto de expansión de entradas y salidas; de propósito general conocido como GPIO y que se accede a él a través de un conector de cabecera rotulado como P1 en la placa. Todos los pines, excepto los de alimentación son reconfigurables y por tanto podremos adaptarlos a nuestras necesidades; aunque por defecto una serie de pines están configurados para la conexión de las interfaces UART, I²C y SPI. La disposición y funciones de sus pines para la placa perteneciente a la Rev2 es la que se muestra en la figura 5 abajo, los pines GPIO4, GPIO17, GPIO27, GPIO22, GPIO18, GPIO23, GPIO24 y GPIO25 son los que vienen configurados por defecto para entradas y salidas digitales [9].

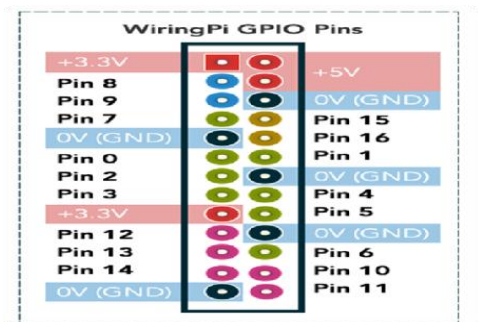


Figura 5: Puertos GPIO.

V. FUNCIONAMIENTO BÁSICO.

1. Antes de todo tenemos que tener todos nuestros elementos de hardware conectados: memoria SD con nuestro sistema operativo ya montado, mouse, monitor y un teclado.
2. Conectamos a corriente el dispositivo, encendemos y empezara inmediatamente a cargar los archivos del sistema operativo.
3. Al cargar el sistema operativo; se debe introducir por default el usuario "pi" y password "raspberrypi".
4. Al introducir el password y usuarios podrá acceder al sistema de raspberrypi, el cual es muy similar a la terminal manejada en diversos sistemas basados en Linux. Por medio de comandos podemos realizar varias tareas como la configuración de tu ip, conectarte a una red o entrar a una carpeta en específico, instalar o correr un programa etc.
5. Para poder trabajar con un entorno grafico de raspberry solo habrá que introducir el comando startx ya continuación el sistema operativo pasara de un entorno de terminal, a un entorno completamente gráfico. [1].

```
Debian GNU/Linux wheezy/sid raspberrypi tty1

raspberrypi login: pi
Password:
Last login: Tue Aug 21 21:24:50 EDT 2012 on tty1
Linux raspberrypi 3.1.9* #168 PREEMPT Sat Jul 14 18:56:31 BST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Type 'startx' to launch a graphical session

pi@raspberrypi ~ $
```

Figura6: Entorno de terminal antes de usar el comando startx.

VI. PROYECTOS DESTACADOS.

Existen varios proyectos ya diseñados o implementados con este dispositivo, los más conocidos se basan en la idea de crear "controles remotos" para un carro de juguete por medio del wifi o Bluetooth.

Pero el más conocido mundialmente es la creación de una computadora portátil o de escritorio, incluso como una consola de videojuegos, entre los más populares se encuentra un contador Geiger, conglomerados mejor conocidos como Clúster, servidores, etc.

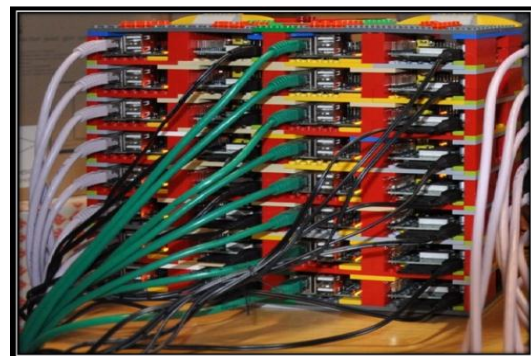


Figura 7: Clúster realizado en la universidad Southampton con 64 dispositivos Raspberry.

VII. DISEÑO DE PROYECTOS.

En una labor de investigación y de interés por el dispositivo y su funcionalidad logramos formar varios proyectos dando comienzo por una matriz de leds de 3 dimensiones, el cual por medio de un algoritmo logramos controlar el encendido y apagado de cada uno de los leds del cubo.

En cada posición se otorgaron a su vez diversas secuencias de encendido creando efectos visuales muy llamativos, esto lo desarrollamos en lenguaje C.

Después de haber terminado con este proyecto nos dimos a la tarea de realizar un seguidor de línea negra con raspberry pi; lo más sencillo era utilizar arduino.

Pero en nuestras investigaciones nos percatamos de que existía un notable abandono hacia el uso de la Raspberri Pi para trabajar con este tipo de proyecto, armamos el sistema mecánico con múltiples sensores y varios materiales reciclados, nuevamente el software fue desarrollado completamente en C [9].



Figura 8: Nuestro seguidor de línea.

Nuestro Cubo

Los algoritmos que diseñamos, no consumen demasiados recursos del sistema, fueron programados en C, y trabajamos con la librería “**WiringPi**” la cual funciona como interfaz para el acceso al puerto GPIO.

```
#include <wiringPi.h>
```

Los pines del puerto GPIO están enumerados y de esta forma se configuran entradas y salidas.

```
int pin0=0;
if(wiringPiSetup===-1) //verificación de configuración
return 1;pinMode(pin0,OUTPUT);
```

De igual forma el pin puede funcionar como entrada, por ejemplo en nuestro algoritmo se verifico primero con un ciclo para que envié un uno y un cero cada segundo.

```
for(;;)
{
digitalWrite(pin0,1);
delay(1000);
digitalWrite(pin0,0);
delay(1000);
}
```

Si compiláramos y ejecutáramos este algoritmo y conectáramos un led a al pin 0 de Raspberry Pi se observaría como se enciende y apaga cada segundo (timer).

La forma de compilar el programa es similar a la compilación de cualquier programa en c.

<pre>cc-onombre_ejecutable L/usr/local/lib -lwiringPi</pre>	<pre>nombre_programa.c-</pre>
---	-------------------------------

Hay que especificar la dirección donde esta almacenada la librería WiringPi. La forma de ejecutar es la siguiente.

```
./nombre_ejecutable
```

Algoritmo Usado en cubo de Leds.

El proyecto consta de un cubo formado por 27 leds (3x3x3), la secuencia para encender dichos leds fue controlada utilizando Raspberry Pi.

Fueron utilizados 12 pines del puerto GPIO, con los cuales definimos 9 para especificar la columna que se polarizaría con positivo y 3 pines para especificar la capa que será polarizada con negativo.

Por ejemplo: para encender el led del centro del cubo polarizamos la columna 5 (+) de la capa 2 (-)
Se diseñó un algoritmo que encendía los leds en diferentes secuencias utilizando un ciclo para cada una de ellas.

Pines:

0->8 columnas.

9->11capas.

La secuencia para encender en orden cada columna es:

digitalwrite (9,0);

digitalwrite (10,0);

digitalwrite (11,0);

Algoritmo Usado En Seguidor De Línea

En este proyecto se construyo un robot seguidor de línea el cual está formado con sensores, motores y un cerebro, es aquí donde implementamos el uso de Raspberry Pi (como cerebro).

El robot sigue una línea negra en un fondo blanco, dicha línea es detectada a través de dos sensores de infrarrojos, los cuales son colocados sobre la línea negra esto con el fin de detectar cuál de los dos sensores fue el que salió de la línea y así corregir la dirección del robot.

El funcionamiento es sencillo si el sensor de la derecha salió de la línea, cambio la dirección hacia la izquierda y viceversa; en cambio si ambos sensores están en la línea, la dirección sería recta.

El sensor manda un 0 si se encuentra sobre el color negro y envía un 1 si esta sobre el color blanco (fuera de línea) con esto podemos decir que: 0=dentro de línea, 1=fuera de línea. Algoritmo:

```
if(digitalRead(sensor1)==1)//sensor1 salió de la línea
  anegra
  Con esta condición podremos saber si el sensor
  esta sobre la línea.
  if(digitalRead(sensor1)==1)//sensor1 salió
  {
  digitalWrite(direccion1,1);//rectificación de
  dirección
  }
```

VIII. CONCLUSIONES

La creciente popularidad de Raspberry Pi, supone la eliminación definitiva del gigantesco paso que existe entre una buena idea y su plasmación en la vida real, el cual se encuentra lastimosamente minado de patentes y financiación de prototipos. Raspberry Pi es el nuevo horizonte para la materialización de lo que antes solo existiría en proyectos en el mundo digital para aquellos que no contamos con inversionistas acaudalados o empresas que nos financien, cuenta con potencial para cambiar la forma en la que nos relacionamos con la electrónica de consumo. Muchos más además de nosotros son los que han comenzado a plantearse antes de comprar el nuevo “gadget” que se encuentra actualmente de moda, que si tal vez podríamos crearlo nosotros mismos con nuestras necesidades personalizadas.

Basándonos en el costo de la Raspberry solo podemos decir que el que no lo intenta es porque no quiere.

REFERENCIAS

- [1] Eben Upton and Gareth Halfacree. Raspberry Pi User Guide. Wiley John + Sons (1 de Septiembre de 2012).
- [2] Matt Richardson. Getting Started With Raspberry Pi. Maker Media, Inc.; Edition: 1 (3 de enero de 2013).
- [3] K. Dennis, Andrew. Raspberry Pi Home Automation with Arduino. Packt Publishing (5 de febrero de 2013).
- [4] Linux User and Developer 2013. Has the Raspberry pi met its match? IP imagine publishing.
- [5] Matt Richardson is a Brooklyn-based creative technologist and video producer. Programming the Raspberry Pi: Getting Started with Python. Maker Media, Inc.; Edition: 1 (3 de enero de 2013)
- [6] Brendan Horan. Practical Raspberry Pi. Apress; Edición: 2013. (27 de marzo de 2013).
- [7] Rs components Raspberry Pi Start Guide. Rs components Vsn 1.0 (2012).
- [8] Dr. Simon Monk. Raspberry pi Cookbook. O'Reilly Media; Edición: 1 (10 de enero de 2014)
- [9] Charles Bell. Beginning Sensor Networks with Arduino and Raspberry Pi. Apress; Edition: 1 (21 de noviembre de 2013).
- [10] Andrew Hague, Karl Wright, Robert Cruse and Paul Kingett. Raspberry Pi Education Manual. Raspberry Pi Foundation.
- [11] Stefan Sjogelid. Raspberry Pi for Secret Agents. PACKT PUBLISHING (25 de abril de 2013).
- [12] Maik Schmidt. Raspberry Pi A Quick Start Guide. Pragmatic Bookshelf; Edición: 1 (6 de noviembre de 2012).

<https://www.dropbox.com/sh/dd863cwrwy0k1yz/r3-YRXFn7L/libros>