

Diseño novedoso de un sistema domótico e inalámbrico basado en Raspberry-Pi y Arduino

GARCÍA-MARTÍNEZ, Mario Alberto^{†*}, HERRERA-AGUILAR, Ignacio, REYES-NICOLÁS, Rafael, ALEGRÍA-VÁZQUEZ, Oliver

Instituto Tecnológico de Orizaba

Recibido Septiembre 29, 2016; Aceptado Febrero 08, 2017

Resumen

En este trabajo se presenta el diseño e implementación de un sistema domótico que usa una plataforma de software/hardware basada en dos dispositivos novedosos de desarrollo: las tarjetas Raspberry-Pi® y Arduino®. El módulo de adquisición de datos opera de manera inalámbrica, y monitorea de manera remota diversas variables de interés en esta aplicación domótica. Para la comunicación inalámbrica, se utilizó el protocolo TCP/IP, el cual consta de dos bloques básicos: el primero denominado el cliente, está conformado por la interfaz gráfica en el entorno de programación Gambas® y la tarjeta de desarrollo Raspberry-Pi, la cual sustituye el uso común de una computadora, trabajando como un procesador de uso específico; y el segundo bloque denominado como el servidor, ha sido integrado por una tarjeta Arduino y su módulo de adquisición de sensores. El procesamiento, almacenamiento y la presentación de la información de las variables analógicas se presenta de una manera intuitiva y natural con elementos visuales como gráficas e indicadores numéricos dinámicos.

Sistema domótico, Tarjeta Raspberry-Pi, Tarjeta Arduino

Abstract

In this paper it is presented the design and implementation of a home automation system using a software/hardware platform based on two novel development devices: Raspberry-Pi and Arduino boards. The data acquisition module operates wirelessly, and remotely monitors several variables of interest in this domotic application. For wireless communication, the TCP / IP protocol was used, which consists of two basic blocks: the first one called the client, is made up of the graphical interface in the Gambas programming environment and the Raspberry-Pi development board, which replaces the common use of a computer, working as a specific user processor; and the second block that is called as the server, has been integrated by an Arduino board and its sensor acquisition module. The processing, storage and presentation of the information of the analog variables is presented in an intuitive and natural way with visual elements such as graphs and dynamic numerical indicators

Home automation system, Raspberry-Pi board, Arduino board.

Citación: GARCÍA-MARTÍNEZ, Mario Alberto, HERRERA-AGUILAR, Ignacio, REYES-NICOLÁS, Rafael, ALEGRÍA-VÁZQUEZ, Oliver. Diseño novedoso de un sistema domótico e inalámbrico basado en Raspberry-Pi y Arduino. Revista de Sistemas Computacionales y TIC'S 2017, 3-9: 25-34

[†] Investigador contribuyendo como primer autor.

Introducción

El continuo desarrollo tecnológico de los circuitos integrados permite contar en la actualidad con una diversidad de dispositivos electrónicos novedosos y de bajo costo, cuyas prestaciones permiten su uso como herramientas eficientes en una amplia gama de aplicaciones que requieren el procesamiento de señales analógicas y digitales. Tal es el caso de las tarjetas Raspberry-Pi® y Arduino®, cuyo empleo se ha popularizado los últimos años en las áreas mencionadas (Pastor, 2015) (Richardson and Wallace, 2012)(Rereira, Fernandes and Sequeira, 2014).

Los sistemas domóticos son cada vez más requeridos en la actualidad debido a la búsqueda de seguridad y control de una diversidad de variables en una casa habitación o edificio, como se muestra en la figura 1 (Vaughan, 2016) (Hill, 2015) (Brennan, 2013).

Se reporta en el estado del arte una diversidad de aplicaciones domóticas basadas generalmente en el uso de procesadores de uso general, como una computadora (Benmansour, Bouchachia and Feham, 2015) (Reaz MBI. 2013). En este trabajo se presenta el diseño de un sistema domótico embebido basado en una plataforma de software/hardware de uso específico que aprovecha las prestaciones de las tarjetas Raspberry-Pi y Arduino (hacedores, 2016) (create.arduino.cc, 2016) (stackoverflow, 2016). Es un sistema, amigable y de bajo costo y muestra cómo se puede implementar la medición y control inalámbrico de diversas variables domóticas aprovechando el procesamiento complementario de las dos tarjetas y la disponibilidad del software libre para su implementación.

Metodología

En la siguiente sección se presenta una breve descripción del hardware y del software utilizados; posteriormente se considera el diseño y desarrollo del sistema. Después se muestran las pruebas realizadas y los resultados obtenidos.

Finalmente se hacen algunas conclusiones relevantes y se proponen trabajos futuros que pueden explotar aún más las ventajas de estas dos tarjetas electrónicas.

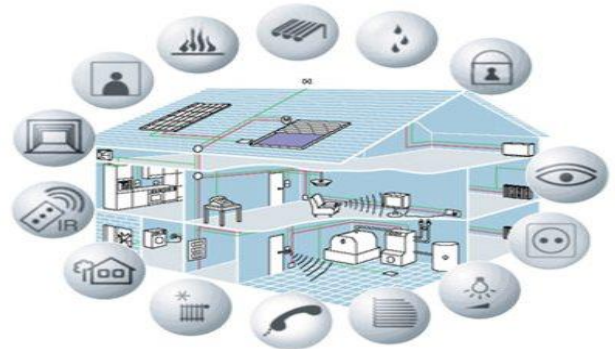


Figura 1 Variables en un sistema domótico

Fuente:(<http://www.arqhys.com/arquitectura-de-sistemas-domoticos.html>)

Descripción del hardware

Desde su fundación en 2006 bajo el liderazgo de Eben Upton y compañeros de la Universidad de Cambridge, el proyecto Raspberry-Pi ha representado toda una innovación en el área de la arquitectura de computadoras (Cellan-Jones, 2011). El Raspberry-Pi 2, el más conocido en la actualidad, es un procesador de tarjeta reducida (SOC) basado en un procesador ARM1176JZF-S corriendo a 700 MHz y con una diversidad de puertos para su interacción con dispositivos externos (Price, 2011). Son muchos los sistemas operativos que soporta Raspberry-Pi, y la mayoría están basados en Linux®. Actualmente, el más utilizado es Raspbian® (una versión de Linux derivada de Debian), lanzado en 2012 y recomendado por los creadores para iniciarse. En la figura 2 se muestra la tarjeta del Raspberry-Pi 2.

Por su parte, Arduino es una tarjeta de desarrollo que integra un microprocesador Atmel AVR con puertos de I/O analógicos y digitales y que usa un entorno de programación basado en “Wiring” (arduino, 2016)(wiring, 2016).

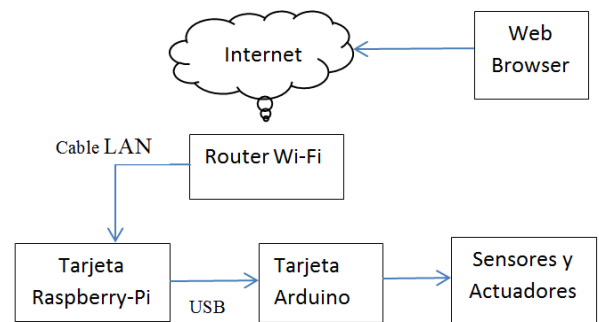

Figura 2 Tarjeta de Raspberry-Pi 2

Fuente: (<http://terceravia.mx/2016/02/raspberry-pi-el-microcomputador-educativo-de-codigo-abierto/>)

Se programa desde una computadora haciendo uso de un puerto serial. La primera versión de Arduino fue presentada en el 2005, con un bajo costo y facilidad de uso para novatos y profesionales con el propósito de desarrollar proyectos interactivos con su entorno mediante actuadores y sensores (arquitecturadecomputadoras, 2010). Desde Octubre de 2012, se incorporaron nuevos modelos de tarjetas de desarrollo que hacen uso de microcontroladores CortexM3, ARM de 32 bits, los cuales coexisten con los originales modelos que integran microcontroladores AVR de 8 bits. La tarjeta se muestra en la figura 3. Como se ha mencionado anteriormente, el objetivo del presente trabajo ha sido explorar las prestaciones de un trabajo complementario con estas dos tarjetas. El sistema propuesto se muestra en la figura 4.


Figura 3 Tarjeta de Arduino

Fuente: (<http://animala.mx/noticias/2016/04/18/arduino-que-es-arduino/>)


Figura 4 Sistema domótico inalámbrico basado en Raspberry-Pi y Arduino

Descripción del software

El sistema operativo más común utilizado con la Raspberry-Pi es el Raspbian, que como ya se mencionó, es una versión de Linux derivada de Debian®. Raspbian es un sistema operativo libre y gratuito optimizado para el hardware de la tarjeta Raspberry-Pi. Cabe mencionar que Raspbian posee más de 35,000 paquetes precompilados, lo que representa una ventaja para el desempeño de la tarjeta (López, 2012).

En cuanto al lenguaje de programación, aunque el más común es Python®, se ha optado por el uso de Gambas, debido a la facilidad para la creación de la interfaz gráfica. Al igual que en Visual Studio®, Gambas permite el desarrollo de aplicaciones de consola, aplicaciones de ventana o componentes para utilizar en otros proyectos.

En el IDE (*Integrated Development Environment*) de Gambas se pueden crear formularios con sus correspondientes controles, los cuales pueden ser botones de comandos, cuadros de texto, listas y muchos otros dispositivos; estos formularios se pueden conectar a bases de datos como MySQL, PostgreSQL o SQLite; se pueden gestionar protocolos de red como HTTP, FTP, SMTP y DNS. También es posible acceder al puerto serie, paralelo, USB o cualquier otro puerto. Gambas permite la creación de todo tipo de aplicaciones de una forma rápida y sencilla, ya sea para programadores expertos o principiantes (Campos y Redrejo, 2010).

Diseño y desarrollo del Proyecto

El sistema domótico está integrado por tres componentes básicos como se muestra en la figura 5.



Figura 5 Componentes básicos del Sistema Domótico

Se ha utilizado la filosofía de Cliente-Servidor para esta implementación, y la plataforma de hardware se ha basado en los dos dispositivos principales del sistema: el microprocesador Raspberry-Pi como cliente, y el microcontrolador Arduino como servidor.

En cuanto a lo referente al concepto de Cliente, fue necesario diseñar una aplicación de software en la plataforma Raspberry-Pi 2, mediante el uso de una interfaz gráfica desarrollada con la ayuda del programa Gambas 3.2, el cual es capaz de formular solicitudes al servidor, recibir datos procedentes del mismo y mostrar los datos procesados y de una manera organizada. Se usó como base un ejemplo ya existente en dicho programa y se le realizaron diversas adecuaciones (cursogambas, 2016) (tuinstitutoonline, 2016).

Como se muestra en la figura 6, en la pestaña número 1 del *TabStrip*, se establecen los parámetros que sirven para establecer la comunicación TCP/IP, como son Protocolo, Servidor y Puerto. Existe además un *TextArea*, donde es posible visualizar el estado de la conexión. En la pestaña número 2 de visualización de las variables, se dispuso un conjunto de *Labels*, que muestran un valor independiente separado de la cadena de caracteres original.

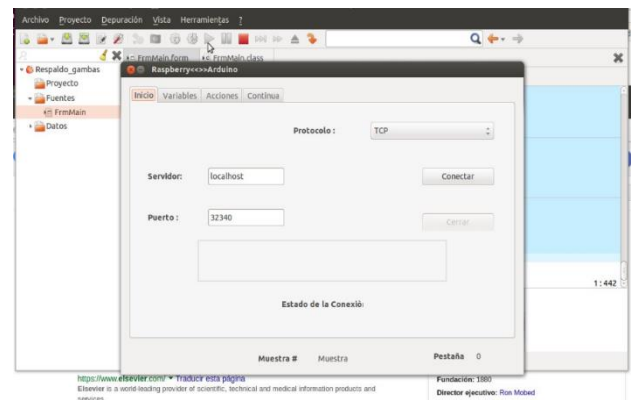


Figura 6 Ejemplo *Socket* modificado con contenedor *TabStrip*

Al inicio, el programa *Socket* almacena toda la trama de entrada en una variable de tipo *String*, pero al utilizar los comandos de separación y conversión de caracteres, se obtenían resultados carentes de sentido. La solución fue asignar el valor de la variable *String* temporalmente a un *TextArea* mediante la propiedad *Text*, para posteriormente obtener un nuevo vector de tipo *String* a partir de esta asignación del vector original al *TextArea*. Así fue posible que el comando *Split* buscara los caracteres “,” y “#” dentro del vector de Prueba y cualquier trama existente entre estos caracteres para almacenarla en un Array. Parte del código se muestra en la tabla 1.

Después de la separación elemento a elemento del *String* de Prueba, se asignan las variables con sus respectivas *Labels* y posteriores acciones especiales, como son la creación de las gráficas de distancia obtenida del ultrasónico HC-SR04 y el sensor de temperatura LM35, como se describe en la tabla 2.

```
Label24.Text = TextArea1.Text
Prueba = TextArea1.Text
ArrayCadenas = Split(Prueba, ",", "#")
TextArea1.Clear
```

Tabla 1 Separación de los caracteres de la trama

Existe la posibilidad, además de recibir datos, también de enviar instrucciones para que se ejecuten tareas en específico, tales como variar la intensidad de un actuador de CA cambiando el valor PWM (*Pulse Width modulation*), lo cual se hace desde el Servidor Arduino. Así mismo, se pueden cambiar de estado alto a bajo los pines del propio Servidor, mediante el envío de instrucciones específicas. Esta sección de código corresponde a la pestaña número 3 de la aplicación gráfica mostrada en la figura 6.

Una de las aportaciones significativas de este trabajo fue la creación de gráficas dinámicas en un área delimitada por un *Chart*, donde dos variables compartieron el mismo espacio; esta sección de la interfaz corresponde a la pestaña 4 del *TabStrip*, como se muestra en la figura 7 (helpgambas, 2016).

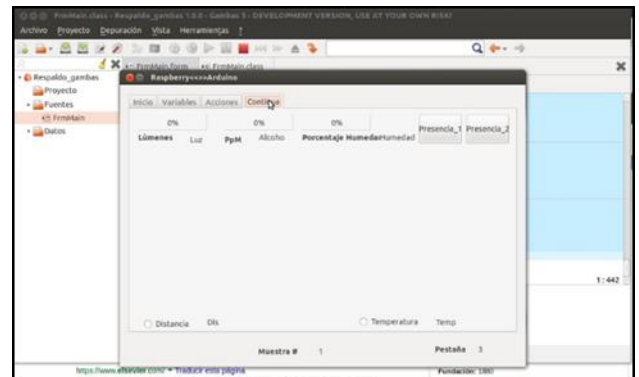


Figura 7 Pestaña 4: “Continua” del *TabStrip*

De manera análoga a la comunicación humana, la comunicación establecida entre el Cliente-Servidor, solo puede efectuarse por turnos, es decir, uno envía mientras otro recibe y viceversa. Este concepto es empleado para la muestra continua de las variables de interés, mediante la utilización de un *timer* que envía constantemente los datos. Para la recepción de los datos se generan los valores necesarios correspondientes al eje X, necesarios para la construcción de la gráfica.

En lo concerniente a la parte del Servidor, como punto de partida se tomó la información disponible en la página oficial de desarrollo del Arduino Ethernet Shield (arduino, 2016). Se realizó la conexión en la red LAN, llevando a cabo el servicio de servidor en la tarjeta Arduino y establecer a través del protocolo de comunicación TCP (Protocolo de Control de Transmisión) el envío y recepción de datos del presente proyecto.

Dentro del formato de petición más básico HTTP, se llevó a cabo la siguiente sintaxis: (método *URL version*). El método le indica al servidor qué hacer con el URL mientras que la versión simplemente indica el número de versión del protocolo que el cliente entiende (Joyanes y Zahonero, 2010). Las líneas de código mostradas en la tabla 3, son las instrucciones recibidas por parte del cliente (Raspberry Pi) en el servidor (Arduino).

```
Label7.Text = ArrayCadenas [1] 'PIN7
Label8.text = ArrayCadenas[2] 'PIN8_RELEVADOR
Label9.Text = ArrayCadenas[4] 'DISTANCIA
Label39.Text = ArrayCadenas[4]
Valor_D.Add(Val(ArrayCadenas[4]))
Label10.text = ArrayCadenas[5] 'LUZ
Luz = Val(ArrayCadenas [5])
P_Luz = 0.0009775 * Luz
ProgressBar1.Value = P_Luz
Label25.Text = Str(Luz)
Label11.Text = ArrayCadenas [6] 'PRESENCIA1
If Val(ArrayCadenas [6]) = 1 Then
ToggleButton1.Foreground = Color.Green
Else
ToggleButton1.Foreground = Color.Red
Endif
```

Tabla 2 Asignación, elemento a elemento, del Array con su respectiva operación.

La tarjeta electrónica “ultrasónico”, que se muestra en la figura 10, cumple la función de “*Breakout Board*” al conectar el sensor HC-SR04 a la terminal hembra RJ11 para su posterior comunicación con el módulo concentrador. Se ha construido en función del tamaño y distribución de pines del sensor. Lo anterior, se realizó con el software EAGLE 7.4.0, el cual permitió realizar diseños de las tarjetas electrónicas y diagramas eléctricos. Además, el uso del conector RJ11, proporcionó protección ante el ruido de señales parásitas, seguridad de conexión y nulas caídas de tensión, lo que permitió la correcta conexión del sensor al módulo “concentrador”. La integración de estas tarjetas en el módulo “concentrador” se puede observar en la figura 11, donde se pueden apreciar los distintos elementos que la integran.

La siguiente etapa de pruebas consistió en la conexión entre cliente-servidor y la prueba de comunicación entre ellos. La figura 12 muestra el resultado exitoso de la comunicación.



Figura 11 Integración del módulo “concentrador”

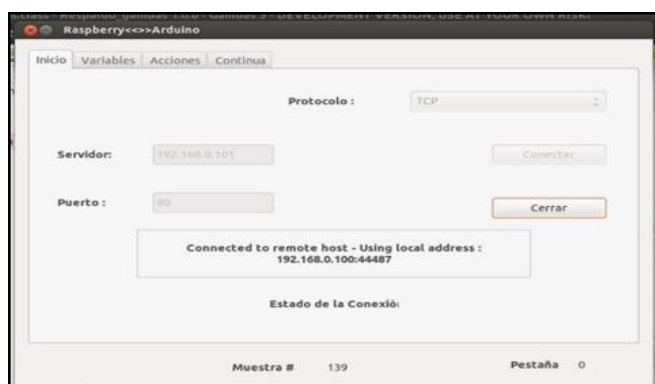


Figura 12 Prueba de comunicación exitosa entre cliente-servidor

Como se ha explicado antes, al llegar la trama de información del servidor Arduino como respuesta del cliente Raspberry, ésta es mostrada en un panel sin ninguna modificación, como se aprecia en la figura 13. Una vez recibida, se lleva a cabo la separación de la cadena *string* a través del identificador elegido en el software Gambas 3, que para este caso se eligió el carácter coma (.). Finalmente, cada sección de la trama recibida de información es procesada por separado y mostrada en sus respectivos lugares seleccionados. En este caso se puede apreciar que en la parte superior de la figura 13 aparecen de manera independiente cada uno de los valores de la cadena de información entrante.

Una prueba más se realizó verificando una de las aportaciones principales de este trabajo, que fue la graficación de los resultados de las mediciones. La figura 14 muestra estos resultados. En este caso se observa la posibilidad de ir graficando las variables de distancia y temperatura, así como gráficas porcentuales de la variable de luminosidad, gas y humedad, según el diseñador lo establezca. La prueba final del sistema consistió en la integración de todos los componentes del hardware y en la verificación de los resultados proporcionados por el software.

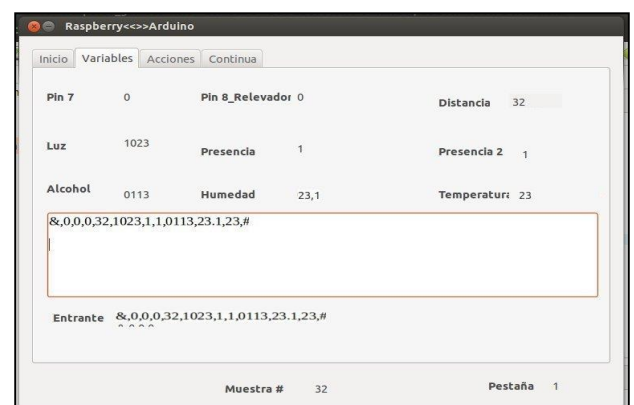


Figura 13 Respuesta del servidor Arduino en la interfaz GAMBAS

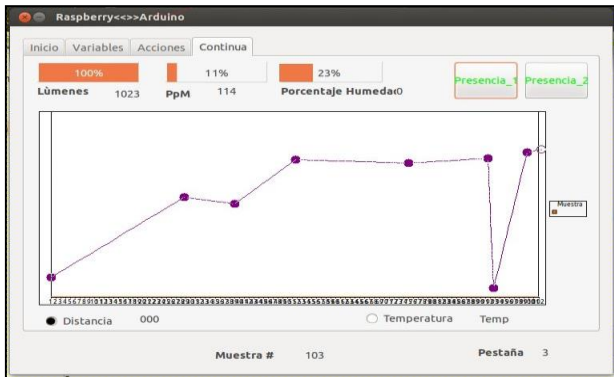


Figura 14 Gráfica de la variable "distancia" en función del tiempo

En la figura 15 se muestra el sistema domótico completo en funcionamiento, con su etapa de hardware (Sensores, actuadores, tarjetas electrónicas, etc.) y de software (Interfaz gráfica en GAMBAS).

Conclusiones

En el presente proyecto se ha mostrado el diseño de un sistema domótico novedoso que aprovecha los recursos del software libre y hardware económico para desarrollar una aplicación en la automatización del hogar. Esta es un área de estudio novedosa y en desarrollo que requiere de propuestas pertinentes ante la necesidad de seguridad que se necesita en una casa habitación.

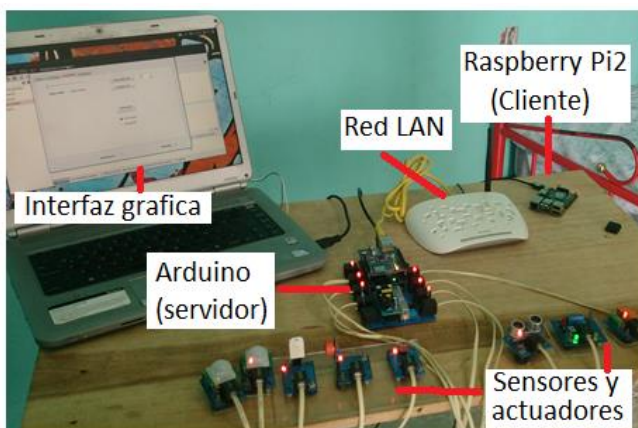


Figura 15 Componentes de hardware y software del sistema domótico inalámbrico

Se han explorado las ventajas del trabajo conjunto entre las tarjetas Raspberry-Pi y Arduino como plataforma de hardware y se ha aprovechado la ventaja del software libre GAMBAS y la disponibilidad de diversas librerías que han facilitado el desarrollo del proyecto y que han permitido diseñar las funciones de procesamiento de señales, de comunicación, el control de actuadores y la lectura de sensores analógicos y digitales.

Se resalta el uso del software GAMBAS, como una alternativa viable y eficiente para realizar interfaces gráficas que lleven a cabo la interacción entre el usuario y las variables en cuestión. No obstante, una de las principales desventajas observada en el desarrollo del proyecto, es la limitada información en relación al funcionamiento del software GAMBAS, específicamente en la etapa de diseño y manipulación de cadenas de información, proveniente de la red LAN, WAN o MAN.

Mediante el diseño e implementación de este proyecto se ha mostrado la factibilidad de su integración al verificar las siguientes características básicas: reducción de costos de diseño, interfaz amigable al usuario final, monitoreo de variables ambientales, control ON/OFF de actuadores finales y escalabilidad del sistema.

Finalmente se ha de tomar en cuenta que el continuo desarrollo tecnológico en el área de los circuitos integrados y del software libre, permiten cada día contar con herramientas más accesibles para realizar diseños más eficientes en el área de la domótica.

Referencias

Arduino o Raspberry Pi,Cuál es la mejor herramienta para ti? <http://hacedores.com/arduino-o-raspberry-pi-cual-es-la-mejor-herramienta-para-ti/>. Consultado en septiembre 2016.

Benmansour A., Bouchachia A. and Feham M. (2015). Multioccupant Activity Recognition in Pervasive Smart Home Environments. *ACM Computing Surveys*. 48(3):34:1-6.

Brennan, M. (2013). House Of The Future: How Automation Tech Is Transforming The Home. *Forbes*.

Cellan-Jones, R. (2011). «A £15 computer to inspire young programmers». *BBC News*. «*blog with video*».

Como generar gráficas con el componente Gb.Chart. <http://help-gambas.blogspot.mx/>. Consultado en noviembre 2016.

Como usar el componente Gb.Chart. <http://help-gambas.blogspot.mx/>. Consultado en noviembre 2016.

Communicating between Raspberry Pi and Arduino over LAN. <https://stackoverflow.com/questions/31193686/communicating-between-raspberry-pi-and-arduino-over-lan>. Consultado en enero 2017.

Contenedores y organizadores. <http://cursogambas.blogspot.mx/>. Consultado en octubre 2016.

Getting Started with the Arduino Ethernet Shield. www.arduino.cc. Consultado en diciembre 2016.

Hill J. (2015). The smart home: a glossary guide for the perplexed | T3. www.t3.com/features/the-smart-home-guide. Consultado en enero 2017.

Interfacing Arduino with Raspberry Pi. <https://create.arduino.cc/projecthub/sankar-Cheppali/interfacing-arduino-with-raspberry-pi-6d9870>. Consultado en febrero de 2017.

Joyanes L. y Zahonero I. (2010). Programación en C, C++, Java y UML. Primera edición. McGraw-Hill. Madrid, España. 190-223.

López M. (2012). Raspbian: Sistema operativo gratuito para la Raspberry Pi <https://www.unocero.com/2012/08/02/raspbian-sistema-operativo-gratuito-para-la-raspberry-pi/> Consulta en diciembre 2016.

Operadores de cadenas de texto. <http://www.tuinstitutoonline.com>. Consultado en octubre 2016.

Pastor J. (2015). Raspberry Pi, presente y futuro del mini ordenador que está revolucionando el sector. <https://www.xataka.com/internet-of-things/entrevista-con-eben-upton-la-raspberry-pi-se-prepara-para-la-internet-de-las-cosas>. Consultado en agosto 2016.

Price P. (2011). «Can a £15 computer solve the programming gap?». *BBC News*. «*video report*».

¿Qué es Arduino? <https://arquitecturadecomputadora.wordpress.com/2013/06/07/arduino/>. Consultado en noviembre de 2016.

Reaz MBI. (2013). Artificial Intelligence Techniques For Advanced Smart Home Implementation. *Acta Technica Corviniensis – Bulletin of Engineering*. 6(2): 51-7.

Redrejo J.L. y Campos D. (2010). GAMBAS. Programación Visual con Software Libre. Edit Lin Editorial.

Rereira V., Fernandes V. and Sequeira J. (2014). Low Cost Objet Sorting Robotic Arm using Raspberry Pi. IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS). September 26-27. Trivandrum.

Richardson M. and Wallace S. (2012). Getting started with Raspberry Pi, Published by O'Reilly Media, Inc.

Vaughan, E. (2016). The future of home automation. PC World.

¿What is Arduino? <https://www.arduino.cc>. Consultado en diciembre 2016.

¿What will you do with the W? <http://wiring.org.co/>. Consultado en diciembre 201