

Cambio de la intensidad de una lámpara de LEDs desde un dispositivo móvil con SO Android

Felipe Santiago Espinosa

Universidad Tecnológica de la Mixteca, Carretera a Acatlima Km. 2.5, Huajuapán de León Oaxaca.

Tel. (01-953) 53 203 99 Ext. 555

fsantiag@mixteco.utm.mx

Jorge Ordoñez Gutiérrez

Universidad Tecnológica de la Mixteca, Carretera a Acatlima Km. 2.5, Huajuapán de León Oaxaca.

Tel. (01-953) 53 203 99 Ext. 555

jorge21482@gmail.com

David Hernández García

Universidad Tecnológica de la Mixteca, Carretera a Acatlima Km. 2.5, Huajuapán de León Oaxaca.

Tel. (01-953) 53 203 99 Ext. 555

garcia7_2@hotmail.com

Resumen

El sistema operativo Android ha tenido un gran crecimiento en dispositivos móviles en los últimos años. La mayoría de ellos cuenta con un módulo de transmisión Bluetooth para una comunicación inalámbrica entre móviles. Debido a las características de transmisión de voz y datos con este protocolo, es factible darle aplicaciones domóticas. En este documento se describe una aplicación para un dispositivo móvil como medio de manipulación de la intensidad de luz en una lámpara de LEDs. Para ello, se establece la comunicación entre un dispositivo móvil y el MCU ATmega16, el cual utiliza un módulo Bluetooth externo configurado en el modo SSP. Desde el móvil se mandan comandos, el MCU los recibe y genera una señal PWM para controlar la intensidad de luz de la lámpara de LEDs. Los resultados obtenidos muestran que este sistema es factible debido a su

bajo costo y fácil manipulación, ya que la mayoría de los dispositivos móviles cuentan con sistema Android y una interfaz Bluetooth.

Palabras Claves: Android, App Inventor, Bluetooth, MCU.

1. Introducción

Las redes de área personal en los últimos años han cobrado gran importancia debido al aumento de computadoras y dispositivos móviles que cada vez utilizan mejores procesadores y métodos de comunicación. Aunque los medios y protocolos de comunicación originalmente se desarrollaron para la transmisión de datos entre dispositivos, actualmente su uso se está ampliando para permitir el manejo de algunos actuadores, como lámparas o pequeños motores; de manera que hay aplicaciones sobre dispositivos móviles relacionadas con el campo de la domótica. La domótica es el conjunto de sistemas y/o dispositivos que se integran para automatizar espacios habitacionales para tener un mayor confort, seguridad, ahorro de energía, bienestar y comunicación, esto, de forma centralizada y/o remota [1].

En domótica existen diversos protocolos de comunicación, algunos sistemas se orientan hacia la red eléctrica de las viviendas, por ejemplo los sistemas X-10, CEBus, EHS; otros utilizan la línea telefónica como vía de transmisión, un ejemplo es el sistema Home PNA 2.0; y más recientemente se ha utilizado la tecnología inalámbrica como las redes HomeRF, Wi-fi y Bluetooth [2].

En Europa se implementó el estándar EIB que define una relación extremo a extremo entre los dispositivos, permitiendo distribuir la inteligencia entre los sensores y los actuadores instalados en una vivienda. La firma Echelon presentó el sistema de control domótico Lonworks; sin embargo, no ha logrado una implantación masiva debido a su alto costo. El sistema Jini es una tecnología de Sun Microsystems, la cual permite utilizar servicios y dispositivos de red por medio de un tono de marcado en la red [3].

Para enlaces de corto alcance, los dispositivos móviles generalmente incluyen al protocolo Bluetooth, que es una especificación abierta para la tecnología aplicada en la transmisión de voz y datos, con el fin de permitir la comunicación ininterrumpida a través de enlaces inalámbricos de corto alcance entre dispositivos estacionarios y portátiles [4].

Bluetooth se ha consolidado en el desarrollo de aplicaciones para el monitoreo e intercambio de datos a través de dispositivos inteligentes como celulares y PDAs ([5], [6]); en la recolección de datos a través de un microcontrolador (MCU) [7], en aplicaciones en ambientes inteligentes [8], entre otras áreas de investigación. Estos trabajos garantizan su consolidación en el mercado y en investigaciones futuras.

Con respecto a los sistemas operativos que dan soporte a los dispositivos móviles, de acuerdo a un estudio llevado por el sitio *BI Intelligence*, el 72% de los usuarios de smartphones usan Android, en tanto iOS tiene el 18% del mercado, BlackBerry el 4% y Windows Phone el 2.9%, considerando un total de alrededor de 1500 millones de smartphones activos en el año 2013 [9].

El desarrollo de Android es liderado por Google y el amplio crecimiento de su uso se debe principalmente a que es un proyecto de código abierto [10]. Android ha sido utilizado anteriormente en sistemas para control de luz, de aire acondicionado y otros artículos del hogar, por ejemplo, en el 2013 Ming Yan y Hao Shi publicaron un artículo relacionado con el desarrollo un sistema para comunicar un dispositivo móvil a un microcontrolador utilizando la tecnología Bluetooth [11].

Por la importancia que están teniendo estas nuevas tecnologías, con este trabajo se evalúa la comunicación entre un móvil con Android y un MCU, para el móvil se desarrolló una aplicación empleando App Inventor, la cual hace uso del módulo Bluetooth para enviar comandos al MCU. El MCU es un ATmega16 de la familia AVR de ATMEL, al MCU se le conecta un receptor de Bluetooth con número de serie JY-MCU. El módulo JY-MCU presenta una interfaz serial del tipo RS-232, por lo que se conecta directo en la USART del ATmega16.

2. Desarrollo

El sistema consta de una aplicación de software para el móvil y otra de hardware y software para el MCU, las aplicaciones fueron desarrolladas y probadas de manera independiente. El sistema en conjunto se implementó una vez que cada parte cumplió con lo que requerido. La aplicación del móvil está desarrollada bajo la plataforma Android con la herramienta de software App Inventor. El MCU trabaja por medio de interrupciones, de manera que el cambio en el ciclo de trabajo de la señal PWM, con la que se manejará una lámpara de LEDs, se hará cada vez que reciba una instrucción del dispositivo móvil. La arquitectura del sistema se muestra en la Fig. 1.

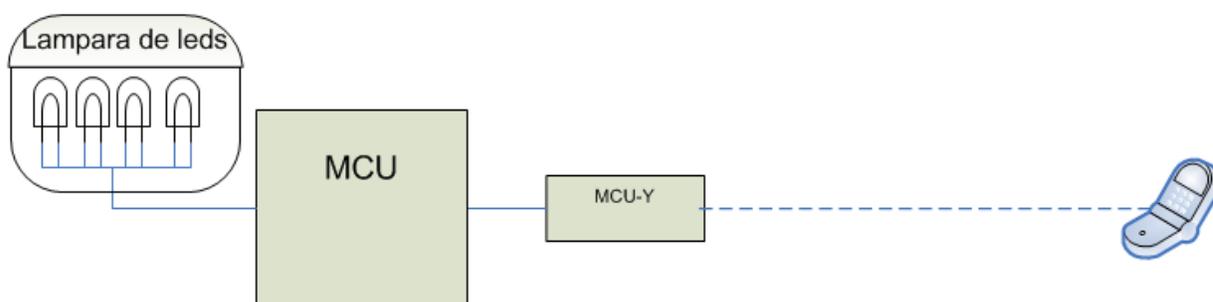


Fig. 1. Arquitectura del sistema.

El MCU se encarga de controlar la intensidad de luz por medio de una señal PWM, ajustando el ancho de pulso entre los valores del 0%, 25%, 50%, 75% y 100%; esto de acuerdo con los datos recibidos desde el dispositivo móvil.

2.1 Requerimientos Hardware y Software

Con respecto al hardware, en la Fig. 1 se observa que el sistema utiliza pocos recursos, por lo que sería posible implementarlo en un microcontrolador con menos capacidad, como el ATmega8 u otro de la gama Tiny. Se utilizó un ATmega16 porque ya se contaba con el dispositivo.

Además del ATmega16 los otros elementos de hardware requeridos son:

- Módulo JY MCU HC-05: Receptor Bluetooth.
- Lámpara de LEDs.

En cuanto al software, se tiene los siguientes requerimientos:

- Una función que se encargue de la configuración de la USART, para que trabaje por interrupciones a una velocidad de transferencia de 9600 bps, asíncrona de 8 bits y sin paridad.
- Otra función para la configuración inicial del temporizador 2, para generar la salida de PWM. El sistema consta de una salida para señal PWM en el puerto D, en este mismo puerto se encuentran la entrada y salida que se utilizaron para establecer la conexión serial. Para una adecuada iluminación, se decidió utilizar un pre escalador del temporizador 2 con un factor de pre-escala de 1024 (división de frecuencia).
- Se requiere almacenar los datos que se reciben por el puerto serie en una cadena, se utiliza el carácter '\$' como el fin de cadena.
- Se requiere la evaluación de la cadena que se recibe para el ajuste del ciclo de trabajo de la señal PWM.

La cadena que se recibe es comparada con cinco cadenas constantes, cada una tiene un valor que determina el ancho de pulso en la señal de salida. Una vez que se ha detectado la cadena recibida, de acuerdo con su valor, se configura al registro OCR2 para modificar el ancho de pulso.

El microcontrolador no utiliza un cristal externo, dado que trabaja con el circuito RC interno configurado para una frecuencia de operación de 1 MHz.

2.2 Diseño del hardware

El sistema utiliza 2 recursos del MCU:

- El temporizador 2 para la generación de la señal PWM. El recurso genera una respuesta automática en la terminal OC2 y con ésta se maneja la lámpara de LEDs.
- La USART para la conexión del módulo JY MCU. Las terminales RXD y TXD de la USART se configuran como entrada y salida, respectivamente. La USART se configura para dar respuesta a interrupciones por recepción.

El hardware resultante se muestra en la Fig. 2.

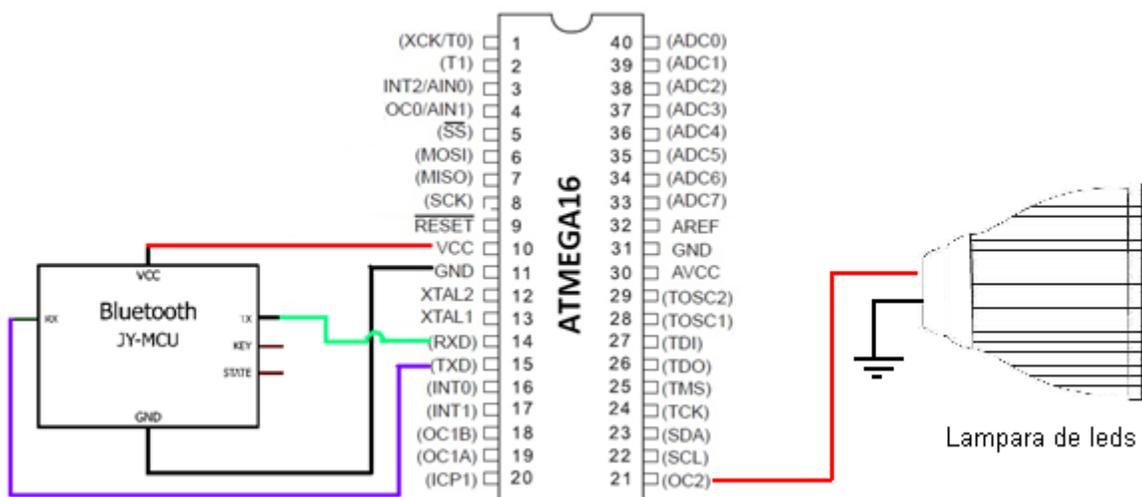


Fig. 2. Diseño de hardware de controlador de intensidad de luz.

2.3 Diseño del software

El diseño del software inicia con el desarrollo de un diagrama de flujo en el que se muestre el comportamiento global del sistema. En la Fig. 3 se ilustra tal comportamiento, cabe aclarar que es una descripción estructurada, sin embargo, algunas etapas no deberían aparecer en el diagrama por el uso de la interrupción por recepción.

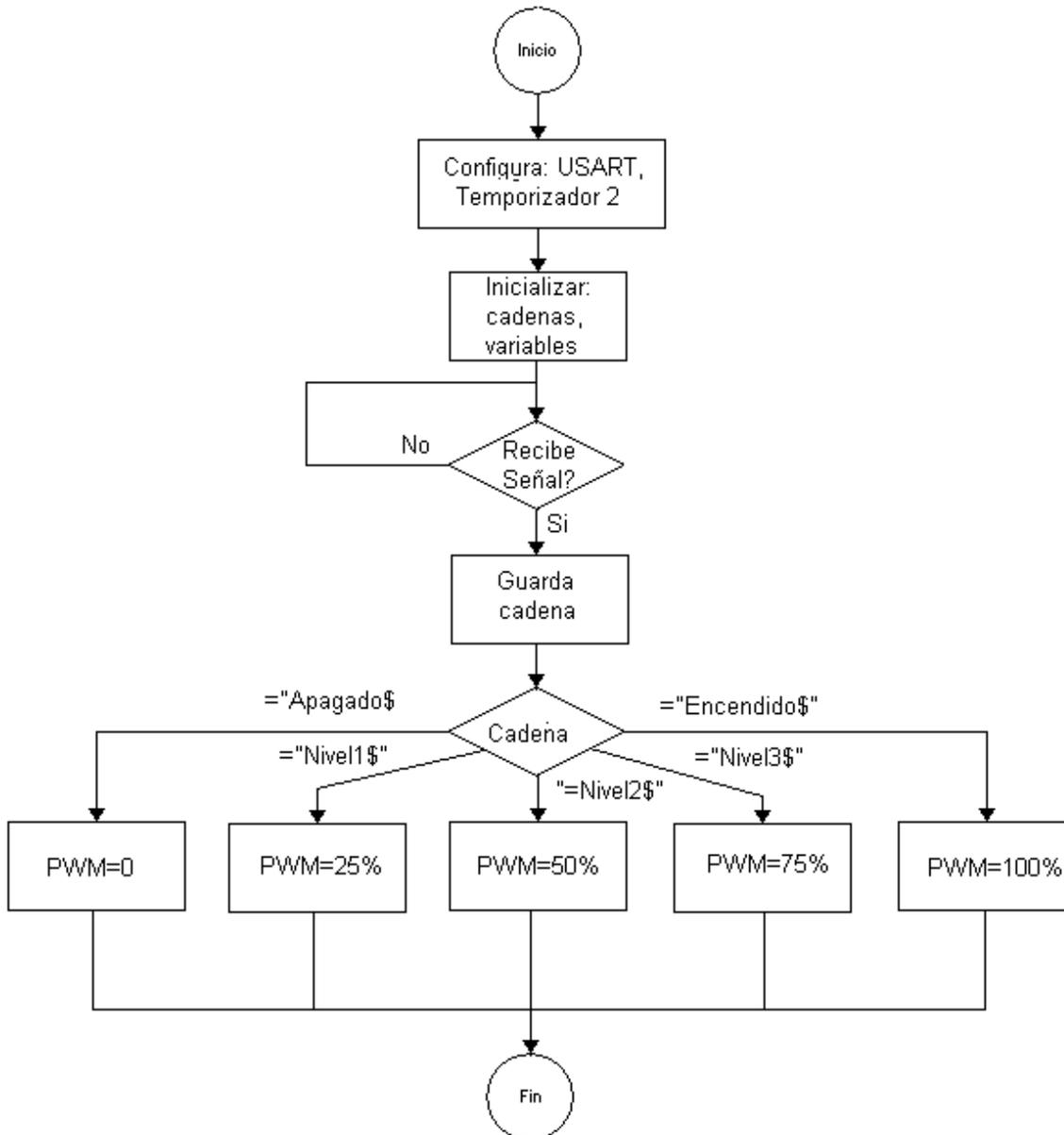


Fig. 3. Diagrama de flujo del software del sistema.

La configuración básicamente corresponde con el ajuste en el contenido de los registros I/O dedicados al manejo de los recursos del MCU.

La espera de un dato por el puerto serie no requiere de un ciclo debido a que la USART fue configurada para trabajar por medio de interrupciones. Una vez recibido un dato, éste se almacena en un arreglo mientras sea diferente del fin de cadena “\$”.

Después se hace una comparación, esto en una función que analiza carácter por carácter, en la cual además se modifica el ciclo de trabajo de la señal PWM.

2.4 Implementación del hardware

El armado físico del sistema en una protoboard se muestra en la Fig. 4.

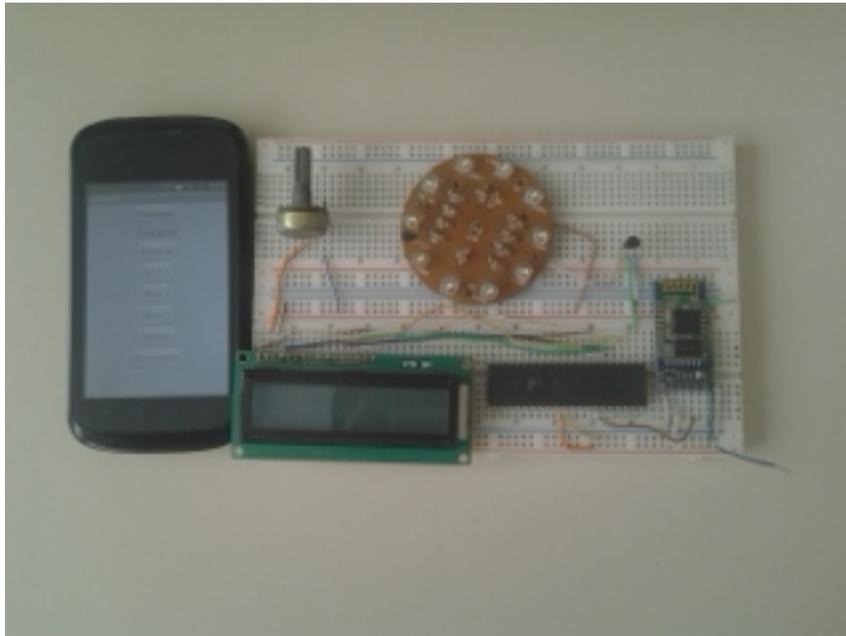


Fig. 4. Hardware del sistema.

El LCD no es necesario para el sistema, se colocó con el propósito de mostrar el porcentaje en que se encuentra el ciclo de trabajo de la señal PWM de salida, éste no sería necesario para el sistema final.

2.5 Implementación del software

El programa se desarrolló en el Lenguaje C, su comportamiento ya ha sido descrito en el diagrama de flujo de la Fig. 3. Dado que el programa es extenso solamente se muestran las funciones utilizadas para la configuración de los recursos y para la modificación del ancho de pulso.

Función de la configuración de la USART:

```
void conf_USART(){
    UBRRH=0;
    UBRRL=12;
    UCSRA=0x02;
    UCSRB=0x98;
    UCSRC=0x86;
}
```

Configuración del temporizador 2:

```
DDRD=0b11000010;
TCCR2=0x6D; //preescalador
```

Función de comparación de cadenas y configuración de ancho de pulso:

```
void ancho_pulso(char instruccion[ ]){
    if (strcmp(instruccion,"pwm9")==0){
        OCR2=250;
    }
    if (strcmp(instruccion,"pwm25")==0){
        OCR2=63;
    }
    if (strcmp(instruccion,"pwm50")==0){
        OCR2=123;
    }
    if (strcmp(instruccion,"pwm75")==0){
        OCR2=186;
    }
    if (strcmp(instruccion,"pwmof")==0){
        OCR2=256;
    }
}
```

}

3. Desarrollo de la aplicación para un móvil

La aplicación de cliente Bluetooth se desarrolló utilizando la aplicación App Inventor de Google Labs. El editor de bloques de la aplicación utiliza la librería Open Blocks de Java para crear un lenguaje visual a partir de bloques. Este software garantiza un desarrollo fácil y rápido de aplicaciones. El sistema es gratuito y se puede descargar fácilmente de la web.

Los pasos para desarrollar la aplicación son los siguientes:

- Configuración de elementos a utilizar.
- Detección de dispositivos Bluetooth en el área.
- Conexión con el dispositivo Bluetooth.
- Envío de instrucciones.
- Cierre de conexión.

3.1 Configuración de los elementos a utilizar

En la Fig. 5 se observan los elementos que utiliza la aplicación, debido a las pocas funciones de la aplicación sólo se utiliza una pantalla.

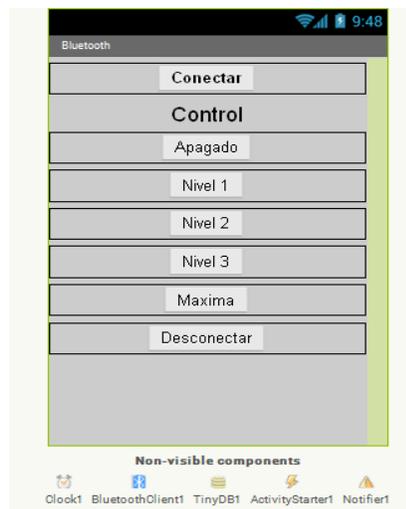


Fig. 5. Aplicación para el envío de datos vía Bluetooth.

El botón “conectar” es un elemento *listPicker* que tiene la función de mostrar una lista de los dispositivos Bluetooth disponibles en el área, por medio del elemento *bluetoothClient*. Una vez seleccionado un dispositivo se establece la conexión, en caso de que no se pueda establecer una conexión se informa al usuario por medio de un mensaje.

Un elemento del tipo *lables* es utilizado para separar la parte que se encarga de la conexión y la que se utiliza para el control. Los botones “Apagado”, “Nivel 1”, “Nivel 2”, “Nivel 3” y “Maxima”, envían una instrucción que indica con qué ciclo de trabajo debe configurarse la señal PWM. El botón “Desconectar” tiene la función de cerrar de manera segura la conexión establecida con el dispositivo Bluetooth.

3.2 Detección de dispositivos Bluetooth en el área

El elemento *BluetoothClient* accede a la lista de dispositivos que han sido localizados por el sistema Bluetooth del dispositivo móvil, por medio de la función *AddressesAndNames*, y los presenta en forma de lista para seleccionar aquel con el que se desee establecer la conexión. El diagrama de bloques realizado en esta etapa se muestra en la Fig. 6.



Fig. 6. Etapa de detección de dispositivos Bluetooth.

3.3 Conexión a un dispositivo Bluetooth

Después de haber seleccionado un dispositivo, el elemento *BluetoothClient* se encarga de realizar la conexión por medio de la función *Connectaddress*, la cual obtiene el nombre del dispositivo del elemento *listPicker*. Para realizar esto, el dispositivo receptor deberá estar previamente vinculado con el dispositivo móvil. El diagrama de bloques para esta etapa se observa en la Fig. 7.



Fig. 7. Etapa de conexión Bluetooth.

3.4 Envío de instrucciones

Para el envío de mensajes primero se debe confirmar que la conexión con el dispositivo continua activa y entonces se envía, se utiliza la función *SendText* para enviar la instrucción deseada. En la Fig. 8 se muestra el diagrama de bloques para el envío de la instrucción "pwmof\$", ésta es enviada al presionar el botón "apagado", el símbolo "\$" es utilizado para denotar el fin de la instrucción.

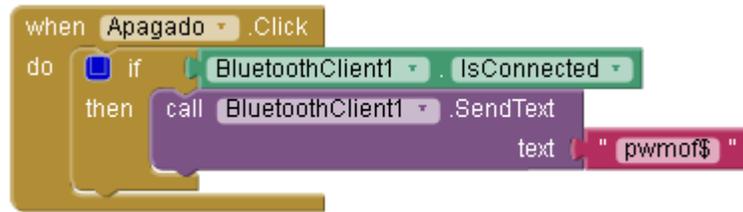


Fig. 8. Etapa de envío de instrucciones.

3.5 Cierre de conexión

Para cerrar la conexión, la función *Disconnect* comprueba que se la conexión siga activa, después el elemento bluetooth cliente se encarga de una desconexión segura. En la Fig. 9 se observa el diagrama de bloques de la etapa que se encarga de cerrar la conexión.



Fig. 9. Cierre de conexión.

4. Resultados

Las pruebas realizadas en el laboratorio mostraron un alcance en la comunicación de aproximadamente 6 m, sin línea de vista y en condiciones climatológicas estables. Un problema que presenta la aplicación es que requiere de un tiempo considerable para detectar el sistema y conectarse al mismo. En este sentido, el sistema está limitado para aplicaciones que no requieren una conexión rápida.

En la Fig. 10 se muestra una fotografía del sistema en funcionamiento, con un PWM de 25%.



Fig. 10 Lámpara con PWM de 25%.

En la Fig. 11 se muestra otra fotografía del sistema en funcionamiento, en este caso el PWM está a un 75%.



Fig. 11. Lámpara con PWM de 75%.

5. Conclusiones

La aplicación fue desarrollada través de la herramienta App Inventor para un dispositivo móvil con sistema operativo Android, que tiene la función de conectarse vía Bluetooth con un receptor conectado a un MCU, el cual recibe órdenes mediante cadenas y

modifica los parámetros de salida de la señal PWM para realizar la variación de la intensidad de luz en un la lámpara de LEDs.

El sistema presentó resultados satisfactorios debido a que cumple de manera eficiente con el problema planteado en un principio. Sin embargo, al analizar a fondo las características tanto del hardware como del software, se hace evidente que este sistema tiene un mayor potencial y por lo tanto es posible mejorar en gran proporción. Siendo la integración de diferentes tecnologías recientes, así como el desarrollo del trabajo en diferentes plataformas de software la aportación de este trabajo.

6. Referencias

- [1] Wikipedia, Domótica, <http://es.wikipedia.org/wiki/Domotica>, Consulta: Septiembre de 2014.
- [2] Sifuentes de la Hoya, Ernesto “Sistemas de comunicación en casas y edificios inteligentes”, uct vol.9 no.36 Puerto Ordaz Diciembre del 2005.
- [3] Herrera Quintero Luis Felipe, “Viviendas inteligentes (Domótica)”, Revista ingeniería e investigación (58), Vol. 25 No. 2, Agosto del 2005.
- [4] Página electrónica del grupo SIG dedicada a la divulgación de información y desarrollo relacionada a la tecnología inalámbrica Bluetooth. <https://www.bluetooth.org/apps/content>, Consulta: Abril, 2014.
- [5] Uichin, L., Sewook, J., Dae-Ki, C., Chang, A., Junho, C., Gerla, M. “P2P Content Distribution to Mobile Bluetooth Users” IEEE Transactions on vehicular Technology, 59(1): 356-367. January, 2010.
- [6] Peng, L., Tingting, H. “Desing of the embedded bluetooth server for movileterminals” Proc. Of the 3rd IEEE international Conference on Computer Science and information Technology (ICCSIT), 2010, pp. 149-152. July, 2010.

- [7] Jia LIU, Guangmin SUN*, Dequn ZHAO, Xu YAO, Yihang ZHANG “MCU-Controlling Based Bluetooth Data Transferring System, ScienceDirect.
- [8] Starsinic, M. “System architecture challenges in the home M2M network” Proc. Of the Long Island System Applications and Technology Conference (LISAT), 2010, pp.1-7. May,2010.
- [9] Adriana Varela, QORE Noticias, 29 de Noviembre 2013, <http://www.qore.com/noticias/12027/ESTUDIO-72-de-los-smartphones-en-uso-en-el-mundo-son-Android>, Consulta: Septiembre de 2014.
- [10] About the Android Open Source Project. <http://source.android.com/about/index.html>. Abril, 2014.
- [11] Ming Yan and Hao Shi, February 2013,” smart living using bluetooth based android smartphone”, International Journal of Wireless & Mobile Networks (IJWMN) Vol. 5, No. 1.

7. Autores

M. C. Felipe Santiago Espinosa es Maestro en Ciencias con especialidad en Electrónica por parte del INAOE, incorporado al IEM de la Universidad Tecnológica de la Mixteca, en donde es Profesor-Investigador desde 1998. Actualmente está cursando el Doctorado en Robótica en la misma institución. En el año de 2012 publicó su libro titulado “Los Microcontroladores AVR de ATMEL”.

Jorge Ordóñez Gutiérrez es estudiante de décimo semestre de la carrera de Ingeniería en Electrónica de la Universidad Tecnológica de la Mixteca.

David Hernández García es estudiante de décimo semestre de la carrera de Ingeniería en Electrónica de la Universidad Tecnológica de la Mixteca.