



Instituto Politécnico Nacional
Escuela Superior de Cómputo



❖ **INTEGRANTES DEL EQUIPO:**

- Carlos Tonatihu Barrera Pérez
 - López Higuera Antonio
 - Monroy Martos Elioth

❖ **UNIDAD DE APRENDIZAJE:** Application Development for Mobile Devices

❖ **PROFESOR:** Alfredo Sigfrido Cifuentes Alvarez.

❖ **PROYECTO 1:** Aplicación MIDlet-Arduino con comunicación Bluetooth.

❖ **FECHA:** 4 de marzo de 2019.

❖ **Versión del reporte:** 1.

❖ **GRUPO:** 3CM8.

Índice

Índice	2
Objetivo	3
Objetivos específicos	3
Conceptos	3
Desarrollo	4
Descripción	4
Programación del MIDlet	4
Desarrollo del circuito y programación de Arduino	4
Software y hardware utilizados	5
Diagramas	5
Código fuente	8
Conclusiones	13
Dificultades encontradas	13
Posibles aplicaciones	13
Conclusiones individuales	14
Barrera Pérez Carlos Tonatihu	14
López Higuera Antonio	14
Monroy Martos Elioth	14
Bibliografía	15

Objetivo

Este proyecto tuvo como objetivo, diseñar una aplicación móvil con Java MIDlet que a través de la comunicación Bluetooth permitiera encender y apagar un LED ubicado en una placa de arduino.

Objetivos específicos

- Desarrollar una aplicación MIDlet que muestre una interfaz gráfica con dos botones, uno para encender el led y otro botón para apagarlo.
- Para apagar o encender el LED, el MIDlet se debe comunicar vía Bluetooth con un Arduino Uno.
- El usuario tendrá la posibilidad de apagar o encender el mismo LED, mediante el Arduino, a través de un botón, en este caso, se realizó la implementación usando dos botones, uno para encender y otro para apagar. Con la finalidad de mantener la consistencia entre la aplicación móvil y el Arduino.

Conceptos

Para el entendimiento de esta proyecto es necesario conocer los siguientes conceptos los cuales fueron clave durante el desarrollo de esta aplicación.

- **MIDlet.** Los MIDlets son las aplicaciones Java desarrolladas con MIDP que se pueden ejecutar en los MIDs. Los ficheros JAD y JAR contienen un conjunto de MIDlets, lo que se conoce como suite. Una suite es un conjunto de uno o más MIDlets empaquetados en un mismo fichero. De esta forma cuando dicha suite sea instalada en el móvil se instalarán todas las aplicaciones (MIDlets) que contenga. [2]
- **Bluetooth.** Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) creado por Bluetooth Special Interest Group, Inc. que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz. [3]
- **Arduino.** Es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso. [4]
- **Java.** Es un lenguaje de programación orientado a objetos, compilado e interpretado por su máquina virtual, Java puede ser utilizado en prácticamente cualquier tipo de proyecto debido a su portabilidad.[2]

- **J2ME.** La plataforma J2ME nos ofrece una serie de APIs con las que desarrollar las aplicaciones en lenguaje Java. Una vez tengamos la aplicación podremos descargarla en cualquier dispositivo con soporte para J2ME y ejecutarla en él. J2ME soporta una gran variedad de dispositivos, no únicamente MIDs. [2]

Desarrollo

Descripción

Para el desarrollo del proyecto, se dividió el trabajo en dos secciones. La primer sección, era la programación de un MIDlet, el cual se comunica vía Bluetooth con el dispositivo Arduino UNO. La segunda, el trabajo realizado en la parte de hardware con Arduino para que este tuviera la capacidad de comunicarse vía Bluetooth, además de la elaboración del circuito con el cual se pudiera prender un led, vía remota o con el uso de botones.

Programación del MIDlet

Para la programación del mismo, se usó como ejemplo, los programas proporcionados por el profesor, y se realizaron las debidas modificaciones, como el que la aplicación permite la comunicación bidireccional, tanto que reciba datos de entrada enviados por el Arduino, como que pudiera enviar datos hacia el mismo.

Además, se realizó un pequeño menú desde el cual se puede seleccionar el dispositivo Bluetooth al cual se requiere conectarse, y otra pantalla principal desde la cual se puede seleccionar si se desea apagar, o prender el led remotamente.

Para la programación de la aplicación se usó el lenguaje de programación Java, y el SDK 3.4 de JavaME.

Desarrollo del circuito y programación de Arduino

En primer instancia, se eligió el dispositivo que le daría la capacidad de comunicación inalámbrica al Arduino Uno, en este caso, el dispositivo elegido, fue el módulo Bluetooth HC-05, debido a que era un dispositivo con el cual habíamos trabajado anteriormente. Además, de que cumple con el objetivo de permitir una comunicación inalámbrica entre Arduino y teléfono.

Posteriormente, se ensambló el circuito con el cual se pudiera prender y apagar un led usando botones y el arduino. Realizar el circuito fue relativamente sencillo debido a la cantidad de información que existe, así como a la posibilidad de acceder a las hojas de datos de nuestros dispositivos, tal como la hoja de datos del módulo Bluetooth que usamos, donde viene explicado la forma en que debe ser conectado.[1]

Software y hardware utilizados

A continuación, se presenta una lista del software y hardware usado para la elaboración del proyecto:

- **Java Development Kit 8.** Necesario para el desarrollo de aplicaciones Java.
- **Java ME SDK 3.4.** Este SDK permite el desarrollo de aplicaciones Java que funcionan en teléfonos que soporten MIDlets.
- **Netbeans 8.2.** Entorno de desarrollo Java que también permite el fácil desarrollo de aplicaciones que utilicen MIDlets.
- **Arduino UNO.** Es una placa con un microcontrolador que permite su programación para el desarrollo de aplicaciones electrónicas.
- **Arduino IDE.** Entorno de desarrollo para arduino que permite crear programas y además permite cargar los mismos a la placa de Arduino.
- **Modulo Bluetooth HC-05.** Es el módulo que se utilizó en la placa arduino para la comunicación entre dicho microcontrolador y un teléfono celular, debido a que es un módulo que se ajustaba a las necesidades del proyecto, ya que no es costoso, y su uso es sencillo.
- **Nokia xpressmusic 5310.** Modelo de teléfono que se utilizó para realizar las pruebas y comprobar el funcionamiento del proyecto.
- **Otros componentes.** Se usaron resistencias de 1K Ohms, un led verde, y dos botones de 4 “patitas”, además de varios jumpers.

Diagramas

Los diagramas siguientes muestran la aplicación hecha en hardware, la cual controla mediante botones el encendido y apagado de los led, con ayuda del Arduino Uno.

En la Figura 1 se puede observar, cómo es que se realizó la conexión entre el Arduino Uno y el módulo Bluetooth HC-05.

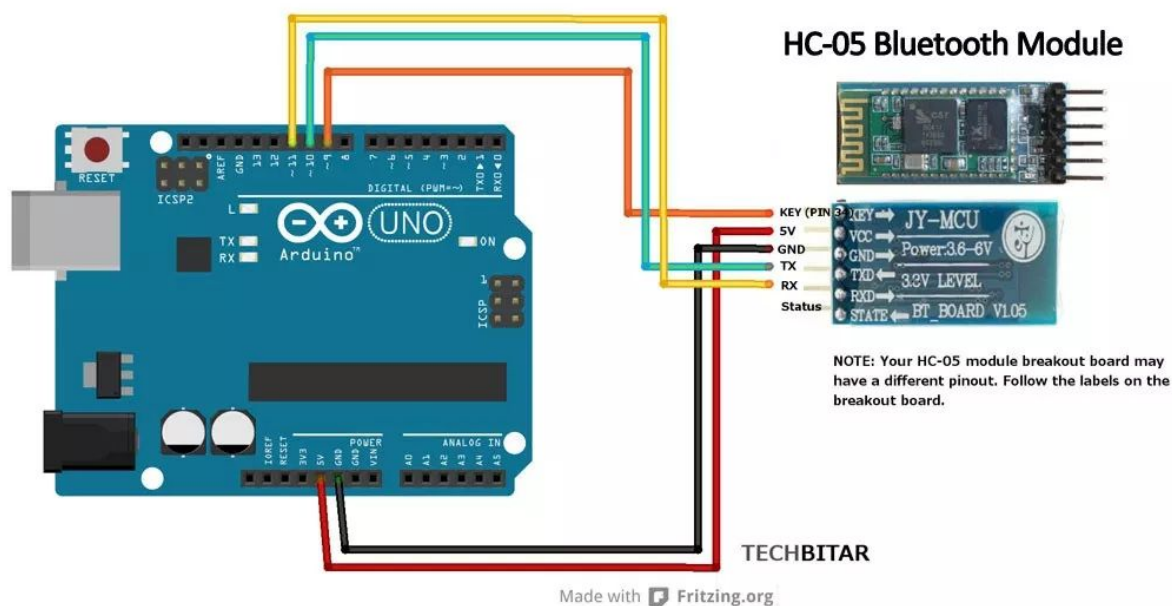


Figura 1. Conexión Arduino Uno con módulo Bluetooth HC-05.

En la Figura 2, se muestra un pequeño ejemplo de cómo puede ser conectado y controlado, un led mediante el uso de un botón y un Arduino Uno. Para el proyecto en cuestión, se utilizaron dos botones conectados de forma similar a la imagen, la principal diferencia

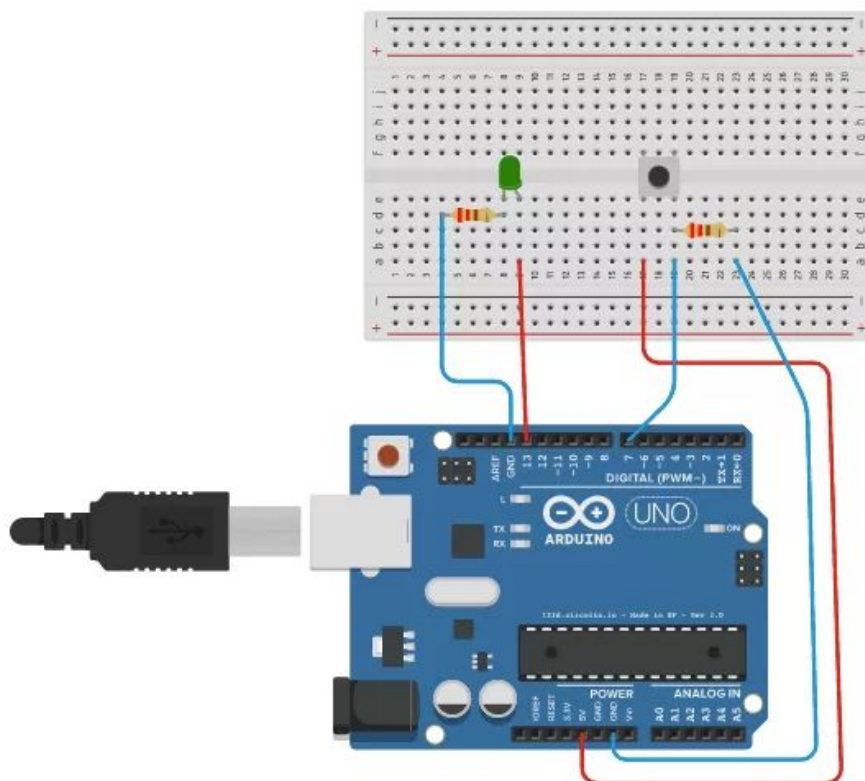


Figura 2. Ejemplo de conexión de un Arduino Uno con un Led y un botón.

En la Figuras 3 y 4, se muestran la implementación final que se tuvo de la parte de hardware del proyecto, el Arduino Uno es alimentado mediante una conexión USB a una computadora, lo cual a su vez nos permite usar esta alimentación para alimentar todo el circuito, el cual se encuentra colocado en una protoboard (placa de prueba),

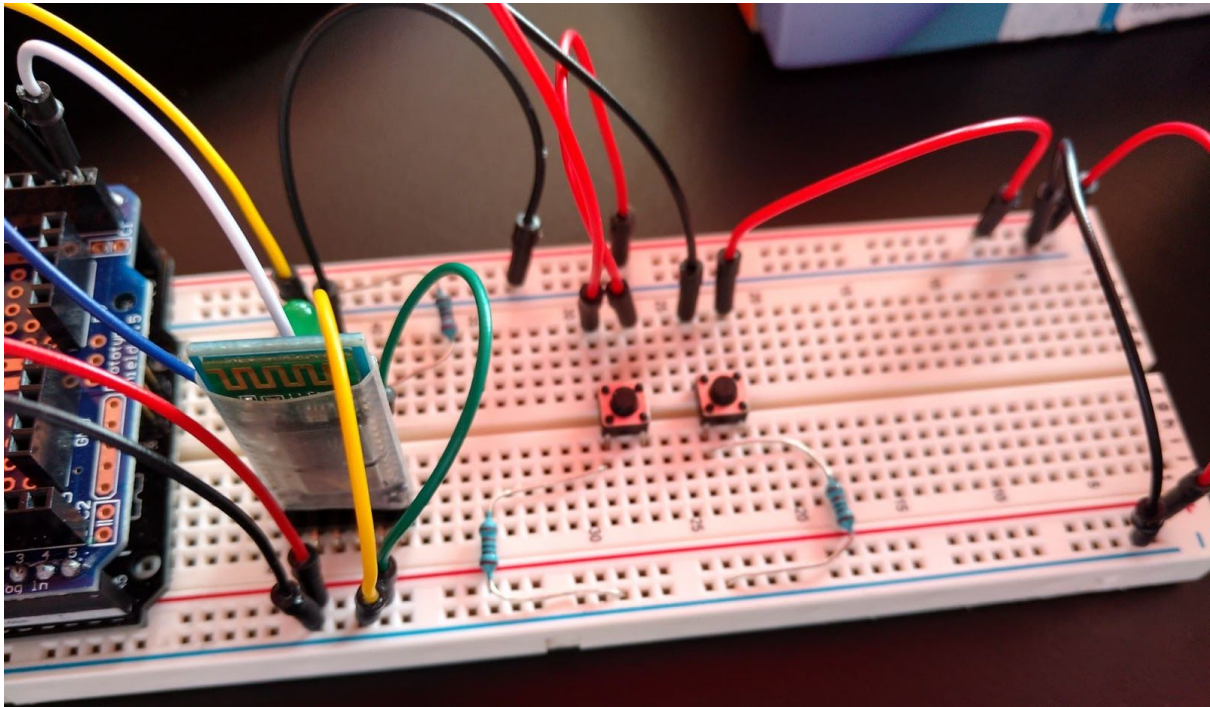


Figura 3. Implementación del proyecto (1).

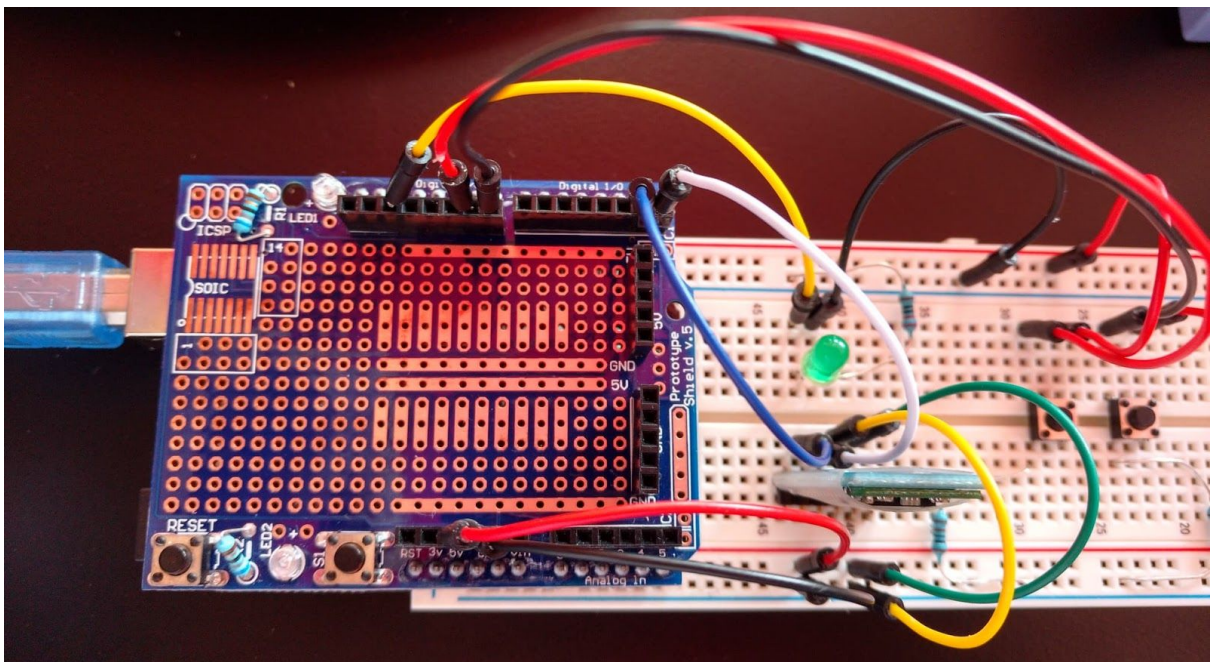


Figura 4. Implementación del proyecto (2).

Código fuente

A continuación, se presenta el código elaborado para el desarrollo del proyecto. Aplicación Java Midlet (BluetoothUI.java):

```
package com.escom.ipn.mx.ledbluetooth;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Vector;
import javax.bluetooth.*;
import javax.bluetooth.DiscoveryListener;
import javax.microedition.io.Connector;
import javax.microedition.io.StreamConnection;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
/**
 * @author Vaibhav
 */
public class BluetoothUI extends MIDlet implements CommandListener,
DiscoveryListener, Runnable {

    private final List listaDispositivos;
    private final Command commandSalir, commandBuscar;
    private Command commandSeleccionar, commandEnviar;
    private List listaOpciones;

    private Display d;
    private Vector v;

    private String deviceName;
    private DiscoveryAgent agent;
    private Alert dialog;

    private StreamConnection con;
    private OutputStream outs;
    private InputStream ins;

    private boolean noRead = false;
    private long millis1, millis2, debounce;

    private Thread t;

    public BluetoothUI() {
        listaDispositivos = new List("Lista de dispositivos",
List.IMPLICIT);

        commandSalir = new Command("Salir", Command.EXIT, 0);
        commandBuscar = new Command("Volver a buscar", Command.SCREEN, 1);

        commandSeleccionar = new Command("Seleccionar Dispositivo",
Command.SCREEN, 1);
        commandEnviar = new Command("Enviar", Command.OK, 1);

        listaDispositivos.addCommand(commandSalir);
```



```

        listaDispositivos.addCommand(commandBuscar);
        listaDispositivos.addCommand(commandSeleccionar);
        listaDispositivos.setCommandListener(this);

        v = new Vector();
        t = new Thread(this);

        d = Display.getDisplay(this);
    }

    public void startApp() {
        d.setCurrent(listaDispositivos);

        try {
            listaDispositivos.deleteAll();
            LocalDevice local = LocalDevice.getLocalDevice();
            local.setDiscoverable(DiscoveryAgent.GIAC);
            deviceName = local.getFriendlyName();
            agent = local.getDiscoveryAgent();
        } catch (BluetoothStateException ex) {
            ex.printStackTrace();
        }

        try {
            agent.startInquiry(DiscoveryAgent.GIAC, this);
        } catch (BluetoothStateException ex) {
            ex.printStackTrace();
        }
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
        try {
            this.outs.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    public void commandAction(Command c, Displayable d) {
        if (c == commandSalir) {
            this.destroyApp(true);
            notifyDestroyed();
        } else if (c == commandBuscar) {
            this.startApp();
        } else if (c == commandSeleccionar) {
            try {
                RemoteDevice rd = (RemoteDevice)
v.elementAt(listaDispositivos.getSelectedIndex());

                listaOpciones = new List("Selecciona una opcion",
List.IMPLICIT);
                listaOpciones.insert(0, "Encender LED",
Image.createImage("/Resources/turnOn48.png"));
                listaOpciones.insert(1, "Apagar LED",
Image.createImage("/Resources/turnOff48.png"));
            }
        }
    }

```

```

        listaOpciones.addCommand(commandEnviar);
        listaOpciones.setCommandListener(this);

        String URL = "btspp://" + rd.getBluetoothAddress() +
":1;authenticate=false;encrypt=false;master=false";

        this.d.setCurrent(listaOpciones);

        this.con = ((StreamConnection) Connector.open(URL));
        this.outs = this.con.openOutputStream();
        this.ins = this.con.openInputStream();

        t.start();
    } catch (IOException ex) {
        this.d.setCurrent(new Alert("No, wey", "La neta no jaló,
wey", null, AlertType.WARNING));
    }
    this.d.setCurrent(listaOpciones);
} else if (c == commandEnviar) {
    millis2 = millis1 + 500;
    try {
        if (this.listaOpciones.getSelectedIndex() == 0) {
            this.outs.write('1');
        } else {
            this.outs.write('0');
        }
        this.outs.flush();
    } catch (IOException ex) {
        //f.append(ex.toString());
        ex.printStackTrace();
    }
}
}

public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
    String deviceaddress = null;
    try {
        deviceaddress =
btDevice.getBluetoothAddress(); //btDevice.getFriendlyName(true);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    v.addElement(btDevice);
    listaDispositivos.insert(0, deviceaddress, null);
}

public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
{
}

public void serviceSearchCompleted(int transID, int respCode) {
}

public void inquiryCompleted(int discType) {
    Alert dialog = null;
    if (discType != DiscoveryListener.INQUIRY_COMPLETED) {
        dialog = new Alert("Bluetooth Error", "The inquiry failed to

```

```

complete normally", null, AlertType.ERROR);
    } else {
        dialog = new Alert("Inquiry Completed", "The inquiry completed
normally", null, AlertType.INFO);
    }
    dialog.setTimeout(500);
    d.setCurrent(dialog);
}

public void run() {
    int trys = 2;
    millis1 = System.currentTimeMillis();
    try {
        while (con != null) {
            byte buffer[] = new byte[40];
            ins.read(buffer);

            String mess = new String(buffer);
            for (int i = 0; i < mess.length(); i++) {
                if (Character.isDigit(mess.charAt(i))) {
                    trys = mess.charAt(i);
                }
            }

            if (trys == '0') {
                listaOpciones.setSelectedIndex(0, true);
            }

            if (trys == '1') {
                listaOpciones.setSelectedIndex(1, true);
            }
        }
    } catch (Exception e) {
        this.d.setCurrent(new Alert("No, funcionó", "La neta no jaló",
null, AlertType.WARNING));
    }
}
}

```

Código para dispositivo arduino (BlueLed.ino):

```

char data = 0; // Aqui se almacena lo que llega del
bluetooth (solo un byte)
int botonEncendido=8;
int botonApagado = 9;
int pinLED = 13;
unsigned long debounce = 500;
unsigned long tiempoActual;
unsigned long lastTiempo;

void setup() {
    Serial.begin(9600); //Bits por segundo usados en la transmision
del bluetooth
    pinMode(pinLED, OUTPUT); // Indicamos que pinLED sera de salida
    pinMode(botonEncendido, INPUT); // Indicamos que el boton de encendido
sera para recibir datos
}

```

```

    pinMode(botonApagado, INPUT); // Indicamos que el boton de apagado sera
para recibir datos
}
void loop(){
    // Recuperamos el estado de los dos botones
    int estadoBtnEncendido = digitalRead(botonEncendido);
    int estadoBtnApagado = digitalRead(botonApagado);
    tiempoActual = millis();
    int lastEstado = 1;
    // Checamos si fueron precionados, de serlo se prende o apaga el LED
respectivamente

    if(estadoBtnEncendido && ((tiempoActual-lastTiempo)> debounce)){
        digitalWrite(pinLED, HIGH);
        Serial.write("1");
        lastTiempo = tiempoActual;
    }

    if(estadoBtnApagado && ((tiempoActual-lastTiempo)> debounce)){
        digitalWrite(pinLED, LOW);
        Serial.write("0");
        lastTiempo = tiempoActual;
    }

    if(Serial.available() > 0) // Si la conexion serial esta disponible
activaremos o no el LED
    {
        data = Serial.read(); //Leemos los datos
        // Lineas de impresion de datos en la consola
        Serial.print(data);
        Serial.print("\n");
        // Si nos llega un 1 encendemos, si llega un cero apagamos
        if(data == '1'){
            digitalWrite(pinLED, HIGH);
        }
        else if(data == '0'){
            digitalWrite(pinLED, LOW);
        }
    }
}

```

Conclusiones

Dificultades encontradas

Durante la elaboración del proyecto, se suscitaron diversas problemáticas. Primeramente, el proyecto debía ser probado en un teléfono real, con soporte para MIDlets, por lo cual fue necesario buscar algún teléfono que cumpliera con estas características. Sin embargo, estos teléfonos en la actualidad, son difíciles de encontrar, dado el auge que los teléfonos “inteligentes” han tenido en los últimos años. Por lo cual durante dos semanas, estuvimos buscando un teléfono con el cual pudiéramos trabajar, hasta que logramos encontrar uno.

Posteriormente, otra dificultad que encontramos, fue el hecho de que no se había contemplado que existiera una comunicación bidireccional entre el dispositivo Arduino y el teléfono, ya que el diseño original solo consideraba que el teléfono pudiera enviar información. Esta situación fue solventada al preguntar al profesor nuestra duda.

Finalmente, otra situación que se presentó, fue al momento de conectar el teléfono al dispositivo Bluetooth, dado que en ocasiones cuando se encuentran muchos dispositivos Bluetooth cercanos al teléfono, el funcionamiento del teléfono no es el correcto, ya que no logra conectarse. Esto debido, a probablemente exista algún desperfecto en el teléfono, debido a su antigüedad y continuo uso.

Posibles aplicaciones

Con el auge del IoT(Internet de las Cosas), existen infinidad de aplicaciones móviles las cuales nos permiten controlar ciertos aparatos “inteligentes” de nuestros hogares, desde refrigeradores, luces, televisiones, hasta hornos de microondas. En un futuro, es posible que podamos controlar cada aspecto de nuestra casa desde nuestro teléfono móvil. Es por esto, que esta práctica nos permite comprender un poco mejor cómo es que funciona esta nueva tendencia en la tecnología, ya que lo que se programó, fue en realidad un switch, ya que era un dispositivo que dada una instrucción de un usuario, cambiaba su estado (en este caso, de encendido o apagado del led). Pero una posible aplicación, podría ser que en lugar de un led, se conectara a una lámpara o a un foco de una casa, con lo cual. Podríamos controlar desde nuestro teléfono, si se prende o apaga una luz.

Y esto, es solo el inicio de un mundo de aplicaciones que puedan comunicarse con nuestro teléfono y puedan realizar una mayor funcionalidad. Además, al trabajar con un módulo Bluetooth, podemos empezar a trabajar con otras aplicaciones que permitan la comunicación inalámbrica entre dispositivos.

Conclusiones individuales

Barrera Pérez Carlos Tonatihu

En un inicio esta práctica resultó difícil, principalmente por los elementos que involucra, sin embargo, al final se pudo concretar ya que se investigó lo suficiente para poder superar estas dificultades. Y a pesar de que los MIDlets ya no son usados actualmente los conocimientos adquiridos en este proyecto se pueden aplicar en cualquier otro y evitar el tener que lidiar con los problemas que se enfrentaron en este caso como lo fue el de la conexión bluetooth y del envío de los datos.

López Higuera Antonio

Este proyecto demuestra que la tecnología antigua y la actual pueden comunicarse y seguir creando herramientas interesantes. Cuando empezó el desarrollo de este proyecto, la primera idea que tuve fue ¿cómo es que este microcontrolador soportará la conexión con el midlet?, el mecanismo de comunicación siempre será el mismo, está tan bien diseñado que permite una comunicación sencilla y útil. Por otro lado, encontrar documentación de MIDlets y Bluetooth es complicado, sólo se pueden encontrar unas cuantas fuentes y el nivel de explicación es alto, sólo había proyectos de tesis que explicaban el funcionamiento, por lo cual fue un poco más complicado, puesto que era más técnica y menos práctica. Esa fue la única dificultad que hubo, por otra parte la conexión pudo ser un poco tediosa y fue necesario aplicar ciertos *tricks* para hacerla funcionar como queríamos que funcionara.

Monroy Martos Elioth

El desarrollo del proyecto, fue al principio un tanto complicado y lento, ya que en algunas partes nos atrasamos un poco, como fue al inicio, el conseguir el celular adecuado, que nos permitiera trabajar con el proyecto. Sin embargo, el aprendizaje obtenido al finalizar el desarrollo del proyecto, hizo que valiera la pena el trabajo, ya que en este proyecto, pudimos trabajar con la tecnología Bluetooth, la cual es un estándar en nuestros días, además de que pusimos en práctica varias de las cosas que aprendimos durante el semestre, en el desarrollo del MIDlet, el cual, la parte más complicada fue la de la comunicación Bluetooth. Ya que en sí, la interfaz gráfica fue sencilla de realizar. El trabajar con Arduino, también nos abre las puertas a poder realizar otras aplicaciones de hardware de una forma menos compleja, ya que Arduino nos da muchas herramientas para realizar el trabajo de una manera más sencilla.

Bibliografía

- [1] Electronica60Norte, *Datasheet bluetooth to serial port module HC05*. 2006. [Consultado: 25- Feb- 2019]
- [2] M. Lozano Ortega and B. Bonev, "Introducción a los MIDs. Java para MIDs. MIDlets", *Curso de Android y Java para Dispositivos Móviles*, 2010. [Online]. Disponible en: <http://www.jtech.ua.es/apuntes/ajdm2010/sesiones/sesion02-apuntes.html>. [Consultado: 25- Feb- 2019].
- [3] "Bluetooth", *Wikipedia*, 2019. [Online]. Disponible en: <https://es.wikipedia.org/wiki/Bluetooth>. [Consultado: 25- Feb- 2019].
- [4] Y. FM, "Qué es Arduino, cómo funciona y qué puedes hacer con uno", *Xataka.com*, 2018. [Online]. Disponible en: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>. [Consultado: 25- Feb- 2019].