



INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ
FACULTAD DE INGENIERIA ELECTRONICA.

REPORTE DE RESIDENCIA:

**“MONITOREO Y REGULACION PARA LUMINARIAS DE CENTROS
COMERCIALES CON FINES DE AHORRO DE ENERGIA BASADO
EN ANDROID”.**

ASESOR INTERNO:

ING. ODILIO OROZCO MAGDALENO.

ASESOR EXTERNO:

DR. RUBEN HERRERA GALICIA.

REVISORES:

ING. VICENTE LEON OROZCO.

M. en C. ARNULFO CABRERA GOMEZ.

RESIDENTE:

MANUEL ALEJANDRO AGUILAR SOLIS.

TUXTLA GUTIERREZ, CHIAPAS.

ÍNDICE

CAPITULO 1. Generalidades	¡Error! Marcador no definido.
1.1 Introducción.....	¡Error! Marcador no definido.
1.2 Información general de la empresa o institución donde se desarrolló el proyecto.	3
1.3 Área específica relacionada directamente con el proyecto.	3
1.4 Antecedentes.	3
1.5 Planteamiento del problema	4
1.6 Nombre del proyecto.....	4
1.7 Objetivos generales	4
1.8 justificaciones del proyecto	4
1.9 Alcances y limitaciones del proyecto.....	5
1.10 Metodología para el desarrollo del proyecto	5
CAPITULO 2. Fundamento teórico.....	6
2.1 Sistema operativo Android	6
2.2 App Inventor.....	8
2.3 Módem Bluetooth RN 41N	¡Error! Marcador no definido.
CAPITULO 3. Desarrollo e implementación del proyecto.....	14
3.1 Programación de la aplicación en App Inventor	14
3.2 Configuración del módem Bluetooth RN 41N	16
3.3 Resultados	17
Observaciones y sugerencias.	20
Conclusiones	20
Referencias	21

1. Generalidades.

1.1 Introducción

El trabajo aquí descrito está enfocado a implementar un sistema capaz de regular la potencia entregada a lámparas de iluminación interior de centros comerciales, con fines de uso eficiente y ahorro de energía. La calibración de la potencia entregada a las lámparas se hace desde una aplicación soportada en el sistema operativo Android con comunicación vía Bluetooth.

1.2 Información general de la empresa o institución donde se desarrolló el proyecto

Nombre: Instituto Tecnológico de Tuxtla Gutiérrez

Sector: publico.

R.F.C. SEP210905778

Dirección: carretera panamericana km. 108 s/n

Código postal: 29050 Ciudad/Estado: Tuxtla Gutiérrez, Chiapas

Teléfono / Fax Tel. (961) 61 5 04 81; Fax: (961) 61 5 15 87

Email: contacto@ittg.edu.mx

Nombre del Titular de la empresa: M. EN C. José Luis Méndez Navarro

Puesto: Director

1.3 Área específica relacionada directamente con el proyecto

Por centro comercial se entiende uno o varios edificios unidos por un mismo diseño arquitectónico que incluye servicios, pasillos, corredores, locales, y tiendas departamentales. El centro comercial de estudio tiene varias tiendas departamentales. Una de ellas, a la que aquí se hace referencia, tiene tres pisos. Y cada piso consta de múltiples lámparas divididas por secciones, donde todas las secciones de un mismo piso toman alimentación de un solo tablero eléctrico.

La idea es conservar la instalación eléctrica dividida en secciones y desarrollar un sistema básico capaz de ser aplicado para cada sección de lámparas.

1.4 Antecedentes

En la actualidad, la necesidad adquisitiva de productos o servicios de la sociedad, ha permitido que establecimientos de autoservicio establezcan nuevos protocolos de funcionamiento de acuerdo a la demanda y a los horarios de trabajo. Para esto, ha sido necesaria la implementación de sistemas de suministro de energía e iluminación, que permitan que dichos establecimientos comerciales puedan operar correctamente.

De acuerdo a datos obtenidos bajo previa investigación, se ha encontrado que es común que las luminarias de un centro comercial estén encendidas durante todo el lapso de tiempo de servicio, y que, además, operen con la misma intensidad y en todas las áreas sin importar si hay suficiente iluminación natural o sí, se es o no necesaria la iluminación de esa área en un horario determinado. Durante un día cualquiera, el nivel de iluminación se mantiene constante.

También para dos días diferentes de la semana el nivel de iluminación es el mismo. Significa que se carece de una diferenciación de las necesidades reales de iluminación durante las diferentes horas del día y entre un día y otro. Esta falta de diferenciación provoca un consumo excesivo innecesario de energía eléctrica.

En México más del 70% de la energía eléctrica se obtiene a partir de la quema de gas y petróleo, solo un 14% proviene de las centrales hidroeléctricas. Desde esta perspectiva, desperdiciar energía eléctrica equivale a contribuir con el proceso del calentamiento global de la atmósfera, dado que para haber producido esa energía eléctrica se debió haber quemado algún combustible, con su respectiva emisión de gases calientes a la atmósfera. También desperdiciar energía eléctrica es una debilidad en la administración de una empresa por el gasto económico que representa.

1.5 Planteamiento del problema

Los sistemas de iluminación de los centros comerciales carecen de un diseño adaptado a las necesidades reales. El problema consiste en que los sistemas de iluminación iluminan más o iluminan menos de lo debido. Y también iluminan con la misma intensidad en ausencia de personas.

1.6 Nombre del proyecto

“Monitoreo y regulación para luminarias de centros comerciales con fines de ahorro de energía basado en Android”.

1.7 Objetivos generales

Diseñar e implementar una aplicación que permita configurar el nivel de iluminación para unidades de iluminación, basado en una interfaz gráfica de un dispositivo móvil, desarrollada en el sistema operativo Android, con comunicación vía Bluetooth.

1.8 Justificación

El sistema que aquí se presenta tiene como finalidad permitir el ahorro de energía eléctrica, basando su operación en un control de iluminación, utilizando una aplicación móvil soportada en Android que permite configurar un nivel máximo y mínimo de iluminación. Tiene como ventaja la facilidad de operación gracias a la interfaz gráfica, además de su comunicación vía Bluetooth. Esto permite obtener los parámetros deseados de iluminación y con esto lograr el efecto visual deseado.

1.9 Alcances y limitaciones del proyecto

Los alcances del presente proyecto abrirán una gran ventana para sistemas de control remotos, ya que permite utilizar dispositivos móviles para su implementación. Con esto se tiene un gran avance tecnológico, ya que no solo se puede monitorear un sistema, sino además, controlar procesos sin la necesidad de estar físicamente presente en el lugar de operaciones. Se pretende también, lograr comunicación eficaz entre dos dispositivos utilizando puertos Bluetooth, con esto, se incursiona aún más en comunicaciones inalámbricas, utilizando lo que hoy por hoy, en dispositivos móviles es la principal herramienta de comunicación y/o transferencia de datos.

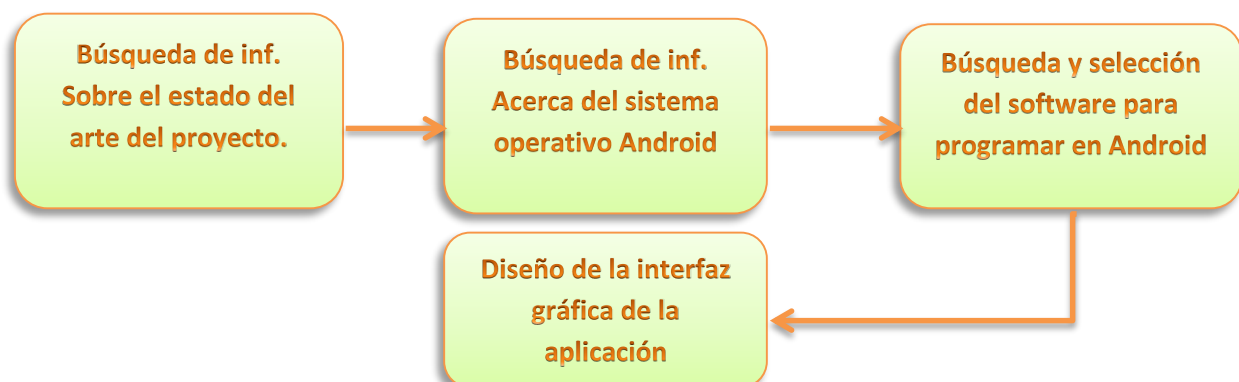
Es necesario mencionar que, la finalidad del presente proyecto pretende demostrar las infinidad de aplicaciones que se lo podrían otorgar a este tipo de sistemas, ya que basta un poco de imaginación para utilizar esta propuesta para nuevos y diversos tipos de aplicaciones móviles con fines de control.

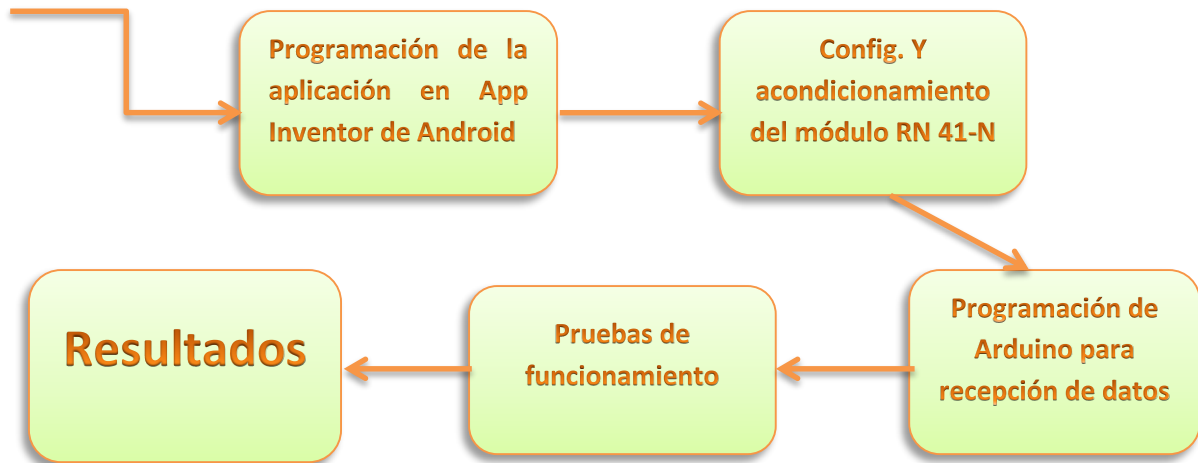
Las limitaciones importantes de este proyecto es la que únicamente se utiliza el sistema operativo Android, que es de los más importantes junto con MAC. También se tiene que se limitan a ser aplicaciones sencillas y de fácil comprensión debido a la poca capacidad de memoria de dispositivos móviles como los celulares. Otro aspecto importante a considerar sería los rangos de trabajo de los puertos Bluetooth, que en el caso del que se utiliza en este proyecto es de 100 m de alcance.

1.10 Metodología

- 1.- Búsqueda de información sobre el estado del arte del proyecto.
- 2.- Búsqueda de información acerca del sistema operativo Android.
- 3.- Búsqueda y selección del software para programar aplicaciones Android.
- 4.- Diseño de la interfaz gráfica de la aplicación.
- 5.- Programación de la aplicación en App Inventor de Android.
- 6.- Configuración y acondicionamiento del módulo Bluetooth RN 41-N en el micro maestro Arduino.
- 7.- Programación del micro maestro Arduino para la recepción de datos.
- 8.- Pruebas de funcionamiento de la aplicación desarrollada y de la transmisión de datos entre el dispositivo móvil y el micro maestro Arduino.

Diagrama de flujo de la metodología





CAPITULO 2. Fundamento teórico

2.1 Sistema Operativo Android.

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y BlackBerry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma. En la figura 2.1 se observa el logotipo oficial de Android

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.



Fig. 2.1 Logotipo de Android.

El sistema operativo Android se usa en teléfonos inteligentes, ordenadores portátiles, notebooks, tabletas, Google TV, relojes de pulsera, auriculares y otros dispositivos. La plataforma de hardware principal de Android es la arquitectura ARM. Hay soporte para x86 en el proyecto Android-x86, y Google TV utiliza una versión especial de Android x86.

Android es considerado como uno de los modelos de negocio más exitosos, pues su desarrollo estratégico contempla los factores que más se tienen en cuenta dentro de las herramientas y metodologías desarrollados por expertos en negocios. Este sistema operativo se ha convertido en un modelo a seguir por desarrolladores de tendencias y negocios de alto impacto.

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows iPhone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

En sus comienzos, Android era eminentemente un sistema operativo pensado para usar con teclado, y gracias a un cursor poder navegar entre las aplicaciones. Desde su comienzo, Android ha sido altamente personalizable. Poco después, antes del lanzamiento del primer teléfono Android, esta filosofía cambió para convertirse en eminentemente táctil, y poder competir contra el iPhone, presentado 1 año y 9 meses antes.

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente. Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

Entre las principales características que posee Android podemos mencionar:

- Conectividad: Soporta tecnologías de conectividad como Wi-Fi, Bluetooth, GSM/EDGE, UMTS, WiMAX y otras más.
- Mensajería: Las formas más comunes como SMS y MMS están disponibles además del servicio Push Messaging de Android.
- Video llamada: Por medio de la versión HoneyComb, Android soporta video llamadas a través de Google Talk.
- Soporte multimedia: Puede soportar los formatos más conocidos como JPEG, MP3, MPEG-4, WAV, además de otros como WebM, H.263 y H.264.
- Multi- táctil: Android cuenta con soporte base para equipos móviles con pantallas multi –táctiles.
- Almacenamiento: Posee una base SQLite, la cual es utilizada para almacenamiento de datos.

Programación. El desarrollo de aplicaciones para Android es sumamente sencillo y lo único que se necesita es un conocimiento básico de Java y poseer el kit de desarrollo de software provisto por Google.

Dispositivos. Android es el sistema operativo con mayor presencia en dispositivos móviles como notebooks, tablets, i-pods, reproductores de mp3 y más. Android es uno de los sistemas operativos que ha logrado establecerse firmemente en el mercado en poco tiempo y además es el sistema con el mayor potencial de desarrollo en el mundo de la telefonía móvil.

Se calcula que en el actualidad hay más 400 000 aplicaciones para Android y que diariamente se activan alrededor de 500 000 equipos móviles. Los precios de los equipos que cuentan con Android oscilan entre los 100 y 600 euros dependiendo de sus diferencias técnicas y modernidad.

2.2 App Inventor

Google App Inventor es una aplicación de Google Labs para crear aplicaciones de software para el sistema operativo Android. De forma visual y a partir de un conjunto de herramientas básicas, el usuario puede ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones fruto de App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil. En la figura 2.2 se muestra el logotipo de App Inventor.

La aplicación se puso a disposición del público el 12 de julio de 2010 y está dirigida a personas que no están familiarizadas con la programación informática. En la creación de App Inventor, Google se basó en investigaciones previas significativas en informática educativa. Fue creada a mediados del 2009 por el profesor Harold Abelson del MIT. El editor de bloques de la aplicación utiliza la librería Open Blocks de Java para crear un lenguaje visual a partir de bloques. Estas librerías están distribuidas por Massachusetts Institute of Technology (MIT) bajo su licencia libre (MIT License). El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza Kawa como lenguaje de programación, distribuido como parte del sistema operativo GNU de la Free Software Foundation



Fig. 2.2 logotipo de App inventor

Para la creación de aplicaciones, App Inventor dispone de una interfaz de programación bastante amigable con el usuario, ya que tiene una lista desplegable de herramientas, donde se encuentran desde los componentes básicos como botones, hasta sensores entre otros. Cada componente o herramienta puede programarse o editarse mediante el uso de sus propiedades que aparecen en la parte derecha de la interfaz al dar click sobre el elemento a utilizar.

Para crear una aplicación basta con presionar el comando “New” en la paleta de opciones, que abrirá una pequeña ventana que pedirá se ingrese el nombre del proyecto a desarrollar.



Fig.2.3 Comando New

Una vez realizado este paso, se abrirá la ventana principal de programación, donde podremos diseñar la aplicación y diseñarla del modo deseado. Es necesario mencionar que la ventana principal, despliega sobre el lado derecho una paleta de herramientas, en medio se muestra la pantalla de la aplicación sobre la que se está programando y del lado derecho de la ventana aparecen las propiedades de cada componente seleccionado en la paleta de herramientas, donde como ya se mencionó, se pueden editar y programar según se requiera.

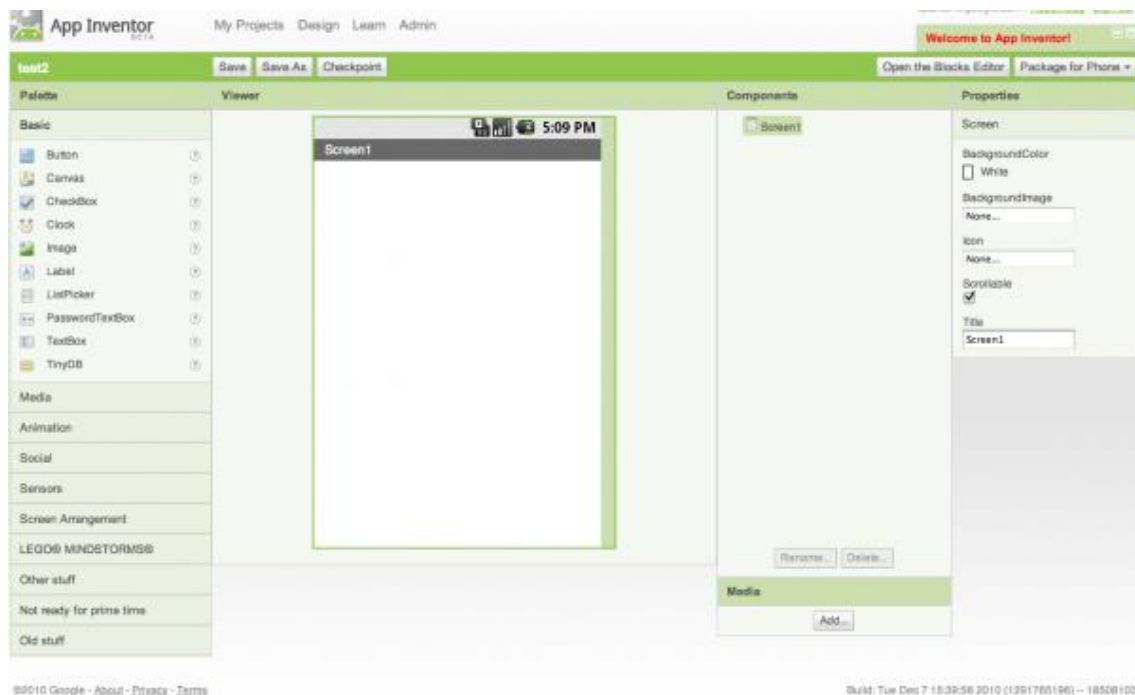


Fig. 2.4 Ventana principal de diseño de App Inventor

La forma de introducir más fácil los componentes desde la paleta de herramientas, es arrastrándolos hacia las Screen o pantalla sobre la que se está trabajando, y ubicarlos en la posición o lugar que se desee.

Una vez introducido el componente podemos disponer de sus propiedades para editarlos o configurar el modo en el que operarán al momento que se inicie la aplicación. Cabe mencionar que la programación de estos componentes se realiza en el editor de bloques de Java, donde se determinan las funciones y/o tareas que deben cumplir los componentes.

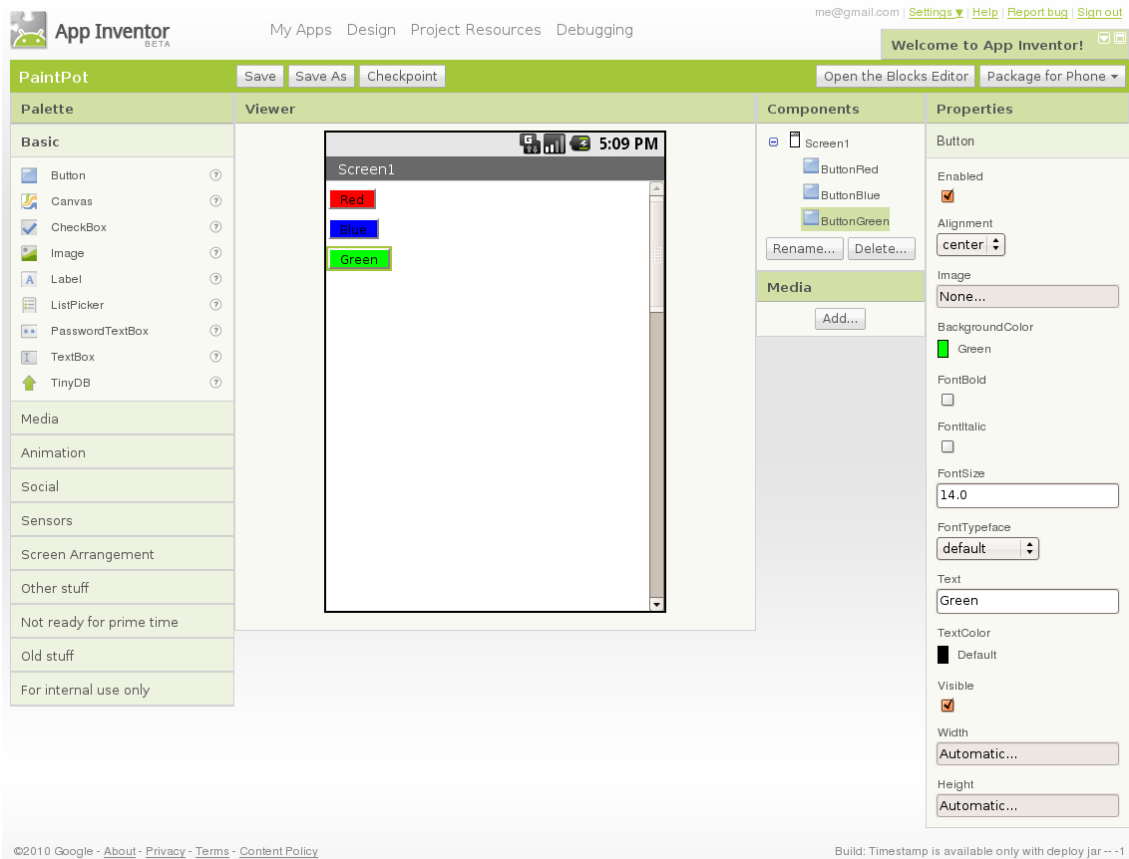


Fig. 2.5 Botones dentro de la Screen y uso de sus propiedades.

Es de esta forma como se puede diseñar la pantalla de nuestra aplicación, donde según sea el proyecto a realizar será, la cantidad y el tipo de componentes que se utilizaran. Podemos disponer las pantallas necesarias para nuestra aplicación, realizando las transiciones de una a otra con una debida programación. Es necesario que el usuario tenga un mínimo conocimiento en programación para comprender las instrucciones que se le pueden otorgar a cada componente.

Una vez realizado el proceso de diseño de la aplicación es necesario abrir el editor de bloques donde se llevará a cabo la programación de la aplicación, esto se logra presionando el comando "Open The Blocks Editor", que se encuentra en la parte superior derecha de la ventana principal de App Inventor.



Fig. 2.6 Comando *Open the Blocks Editor*

Entonces, se abrirá la ventana del editor de bloques de Java, donde se llevará a cabo la programación de los componentes utilizados en la pantalla de la aplicación.

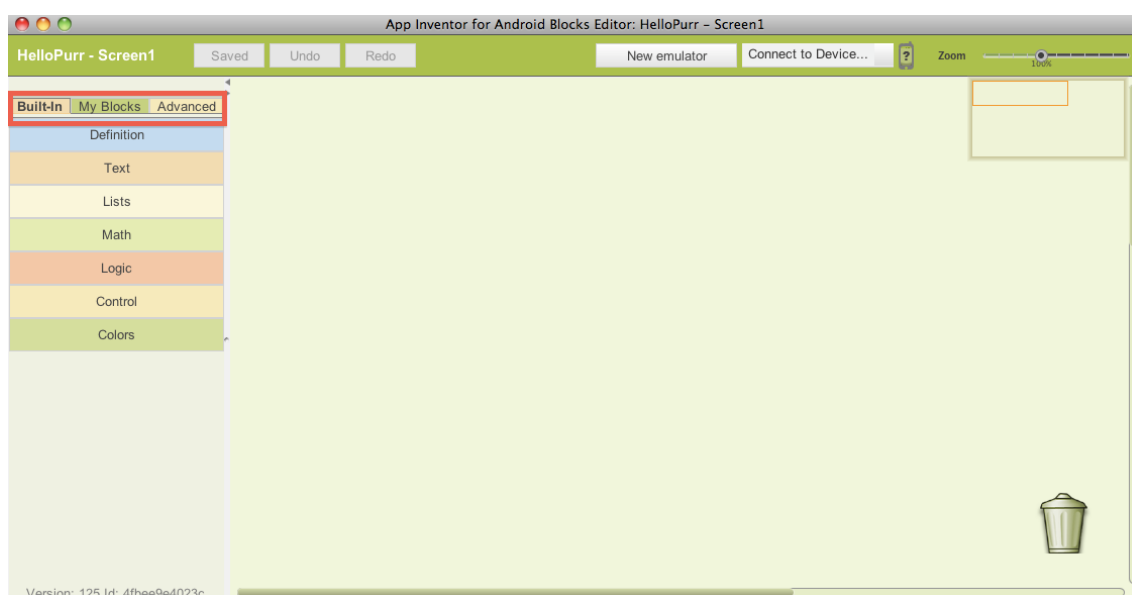


Fig. 2.7 Ventana principal del editor de bloques de *App Inventor*

En el editor de bloques se cuenta con una paleta de opciones como son las incorporadas, mis bloques y avanzadas. En las incorporadas se encuentran las funciones básicas de programación como son instrucciones matemáticas, de control, lógicas, textuales, condicionamiento y de colores. En mis bloques se aparecerán todos los elementos utilizados en la pantalla de la aplicación que se diseñó como primer paso. Y en las opciones avanzadas se disponen de precisamente opciones avanzadas para los componentes utilizados en el diseño de la aplicación.

En la parte superior derecha de la ventana se tienen las opciones de conexión de la aplicación. El comando "New emulator" permite correr un emulador en tiempo real de la aplicación realizada, donde aparecerá un emulador con sistema operativo Android simulando las funciones básicas de la aplicación

desarrollada. Y el comando “Connect to device” permite al usuario empezar a correr la aplicación en el emulador antes mencionado o, cargar dicha aplicación en un dispositivo Android conectado a la Pc o conectado a la red que se está usando en el momento del desarrollo de la aplicación.

Para empezar a programar, basta con seleccionar en la paleta de opciones el comando “My Blocks”, ahí se desplegara la lista de todos los componentes utilizados en la pantalla de la aplicación. Seleccionamos el componente a programar y damos click derecho sobre éste, se abrirán las opciones disponibles para ese componente. Por ejemplo, en la figura 2.8 se muestra algunas opciones que se disponen con un botón.



Fig. 2.8 Componente Boton1 y opciones

De esta forma seleccionamos la instrucción deseada y la arrastramos hasta el campo de trabajo del editor de bloques. Una vez esto seleccionamos en la paleta de opciones las incorporadas que nos permitirán asignarle funciones a dicho botón. Por ejemplo, dentro de las primeras aplicaciones para familiarizarse con App Inventor, está la del gatito que al presionar sobre su imagen en la pantalla se escucha el maullido de este. Para esta función al botón en el diseño de la pantalla se le asigna una imagen, que en este caso es la de una gatito, y en el editor de bloques se programa el botón con la siguiente instrucción como se ve en la figura 2.9.

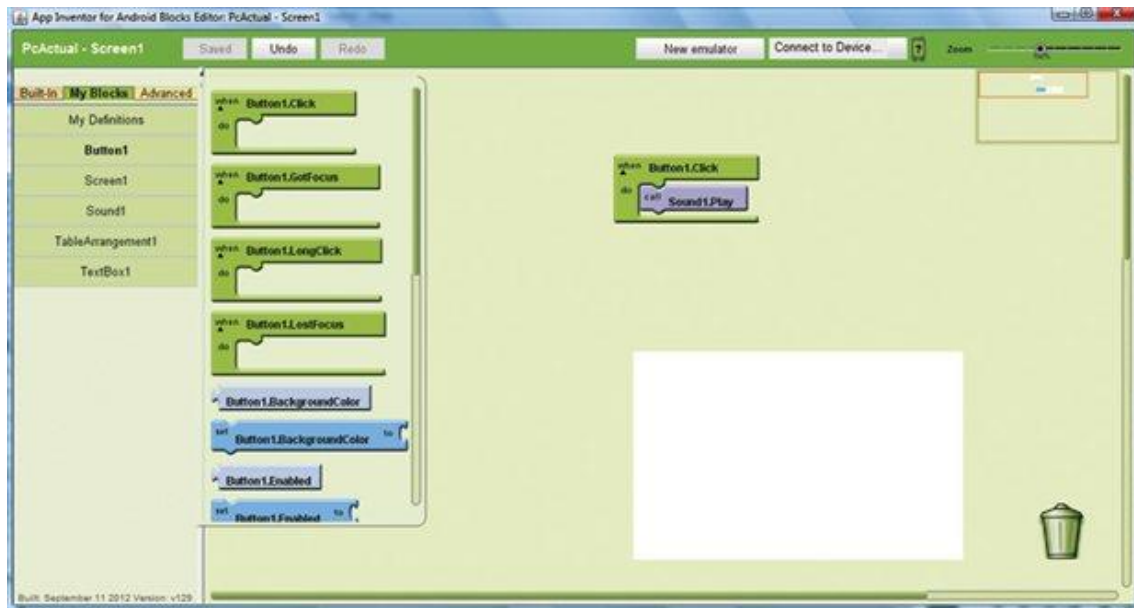


Fig. 2.9 Asignación de la función al botón

Como se observa en la figura seleccionamos la instrucción Button1.Click dentro de las opciones del botón 1 y, abrimos el componente Sound1 que previamente se asignó en la pantalla de diseño de la aplicación, y buscamos la instrucción Sound1.Play que indica que activará el sonido cada vez que se presione una vez el botón 1. Es de esta forma como se le otorgan tareas o instrucciones a nuestros componentes utilizados en nuestra aplicación. Una vez realizada la programación podemos descargar la aplicación a la computadora donde obtendremos un archivo .apk que será compatible únicamente con dispositivos Android.

Y con esto podremos estar realizando todo lo que imaginemos, utilizando App Inventor.

2.3 Módem Bluetooth RN 41N.

Módulo Bluetooth de bajo consumo clase 1, flexible y económico que cumple con el estándar 802.15.1. Soporta diferentes protocolos de comunicación. Es fácil de usar y está completamente certificado, lo que lo convierte en una solución embebida Bluetooth completa. Con su antena de alto rendimiento tipo chip y su soporte de Bluetooth enhanced data rate (EDR), el RN-41 proporciona una tasa de hasta 3Mbps para una distancia de hasta 100m.

Es una excelente solución para agregar comunicación inalámbrica Bluetooth a sistemas existentes.

Características:

Soporta Bluetooth v2.0+EDR

Interfaces de conexión de datos UART (SPP o HCI) y USB (sólo HCI)

Soporta tasas de datos SPP - 240Kbps (slave), 300Kbps (master)

Soporta tasas de datos HCI - 1.5Mbps, 3.0Mbps

Dispone de software para modo HCI ó SPP/DUN

Antena tipo chip

Alcance: hasta 100m con línea de vista.

Frecuencia: 2.402 ~ 2.48 GHz

Modulación: FHSS/GFSK (79 canales a intervalos de 1MHz)

Comunicación segura, encriptación de 128 bits

Corrección de errores

Potencia de salida: 12dBm

Sensibilidad: -80dBm

Rata de transmisión no estándar programable: 1200bps hasta 921Kbps

Consumo de corriente en transmisión: 65mA

Consumo de corriente en recepción: 35mA

Voltaje de alimentación: 3.3V

Tamaño: 13.2mm x 25.8mm.

CAPITULO 3. Desarrollo e implementación del proyecto

3.1 Programación de la aplicación en App inventor.

Para poder realizar la interfaz, se analizó la idea principal del sistema y era la de un software que permitiera al usuario determinar de forma sencilla dos valores límites [I_{max} , I_{min}], de manera que se pudiera establecer primero el valor I_{max} , y luego guardarlo en el micro maestro; una vez realizado se finalizara realizando el mismo proceso para el valor I_{min} .

Siendo así, se decidió utilizar de la paleta de herramientas de App Inventor el componente Slider, que permite deslizar una barra y ajustar el valor que se desee en un rango de 0 a 100, el valor de este Slider1 se muestra en una etiqueta que está arriba del componente para que pueda visualizarse el valor mientras se desliza la barra. Para efectos de exactitud, se decidió utilizar un segundo Slider2 que permitiera ajustar el valor decimal en un rango entre 0 y 1, y al igual que en el primer Slider1 se le colocó una etiqueta que muestra el valor decimal de la posición en que se coloque.

Entonces, primero es necesario ajustar el Slider1 en el valor que se requiera para I_{max} y ajustarlo con un valor decimal del segundo Slider2.

Cabe mencionar que se colocó una tercera etiqueta que muestra la suma de los valores de los componentes Slider1 y Slider2, ya que el resultado de esta suma es el valor de I_{max} que se enviará al micro maestro Arduino.

Una vez determinado este valor se envía al micro maestro mediante un botón con el nombre I_{max} que envía el dato y habilita el segundo botón con el nombre de I_{min}, indicando que corresponde realizar el mismo procedimiento para enviar el valor I_{min} al micro maestro. Cuando se está ajustando el valor I_{max}, el botón I_{min} está deshabilitado, y una vez que se envía el valor I_{max}, se habilita el botón I_{min} y se deshabilita el botón I_{max}, esto para evitar que por error se envíe el mismo dato.

Dicho diseño del algoritmo de programación se desarrolló en el editor de bloques de App Inventor, que para efectos de su entendimiento se explicó anteriormente como utilizar el editor de bloques. En la figura 3.10 y 3.11 se muestra la estructura de la programación a bloques de la aplicación.

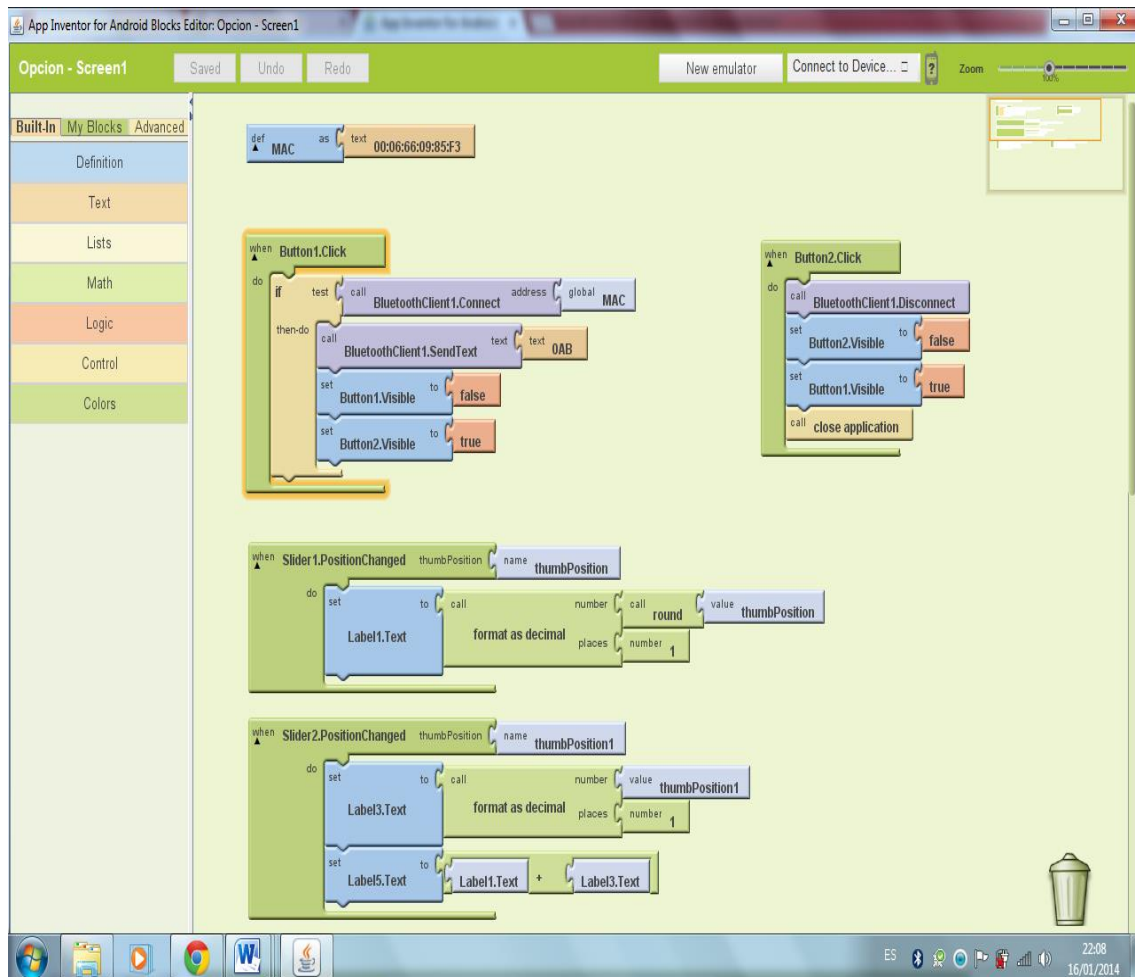


Fig. 3.10 Programación a bloques en App Inventor

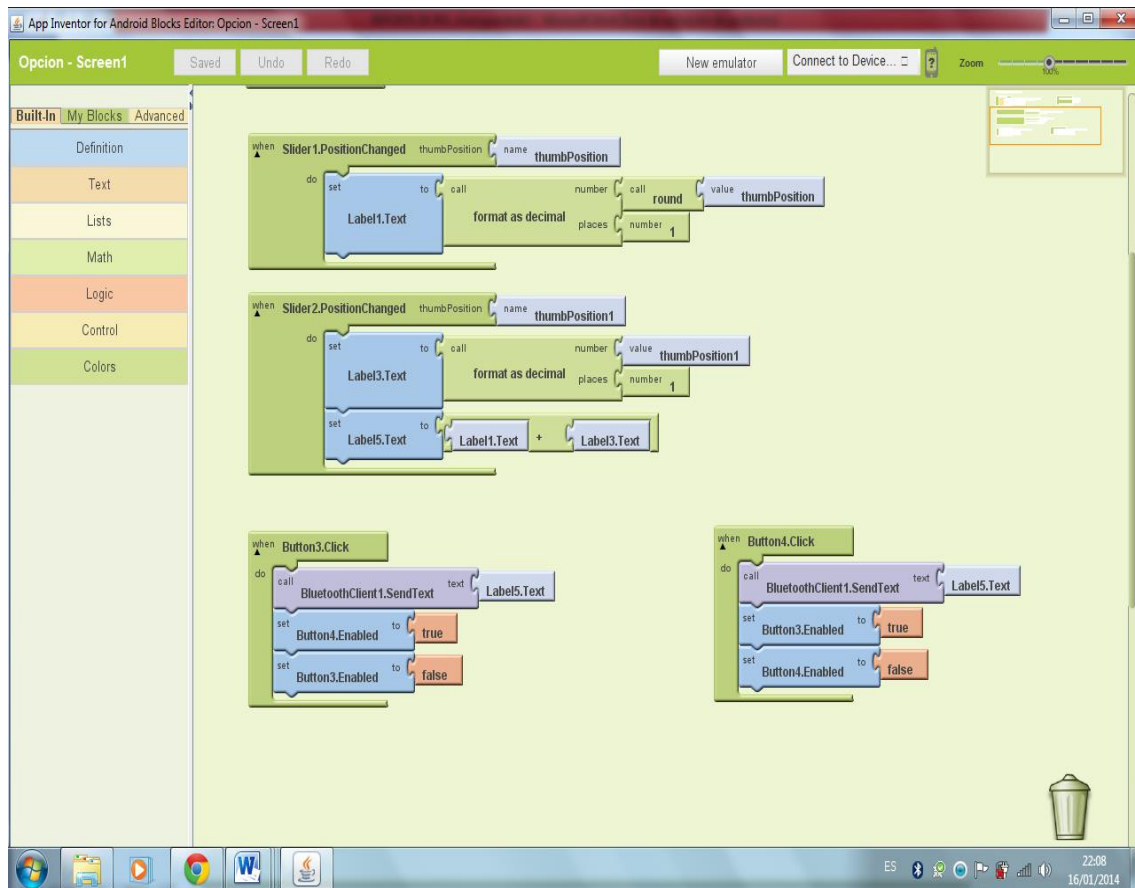


Fig. 3.11 Programación a bloques en App inventor

La finalidad de configurar en el micro maestro dos valores limites [Imax, Imin], es la de determinar el rango de porcentaje de potencia a la que operará la lámpara de Leds de acuerdo a la activación de 4 sensores de presencia que se acoplarán al sistema, es decir; una vez determinados dichos valores, el sistema hará una revisión del estado de los sensores de presencia, si ninguno está activado la potencia a la que se encontrará la lámpara de leds será igual al valor Imin configurado en el micro maestro. Si alguno de los sensores se activa, el micro maestro incrementara la potencia gradualmente según se vayan activando los demás sensores hasta llegar al valor Imax que se determinó en la configuración.

3.2 Configuración del módem Bluetooth RN 41N.

Los pines que utilizaremos para conectar el Bluetooth son el de TX, RX, GND, VDD, PIO2 (aquí se conecta un led para ver el estado del módulo; conectado=HIGH), PIO5 (aquí se conecta un led para ver el estado del módulo; cuando está parpadeando a una frecuencia de 1Hz está esperando la conexión de un dispositivo. Cuando parpadea a 10Hz está emparejado a un dispositivo, y cuando su estado es apagado, el módulo está conectado a un dispositivo). Ver figuras 3.13 y 3.14.

Por último se utilizará el PIO7 que estable la velocidad Baudrate (HIGH=9600, LOW 115). En la figura 3.12 se ve la imagen del módulo Bluetooth RN 41-1N.



Fig. 3.12 Modem Bluetooth RN 41-1N

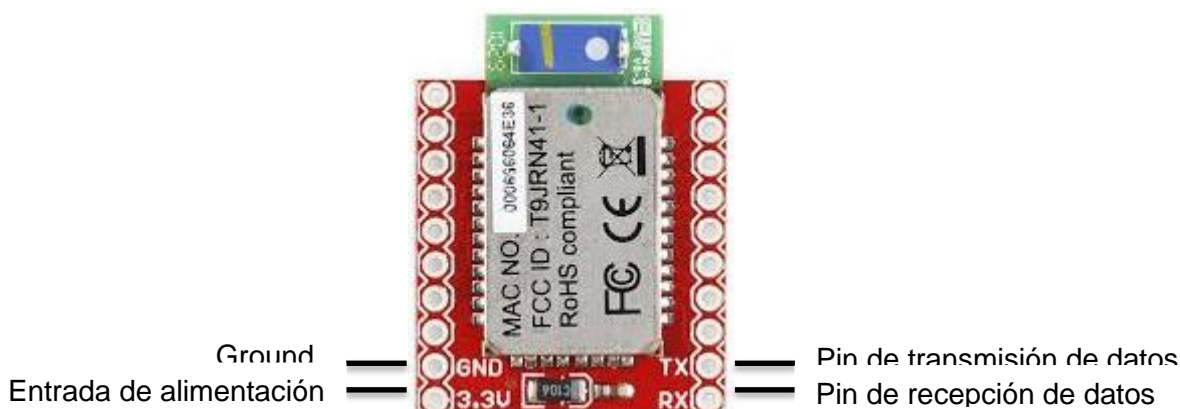


Fig. 3.13 Pines a utilizar en el modem Bluetooth RN 41-1N

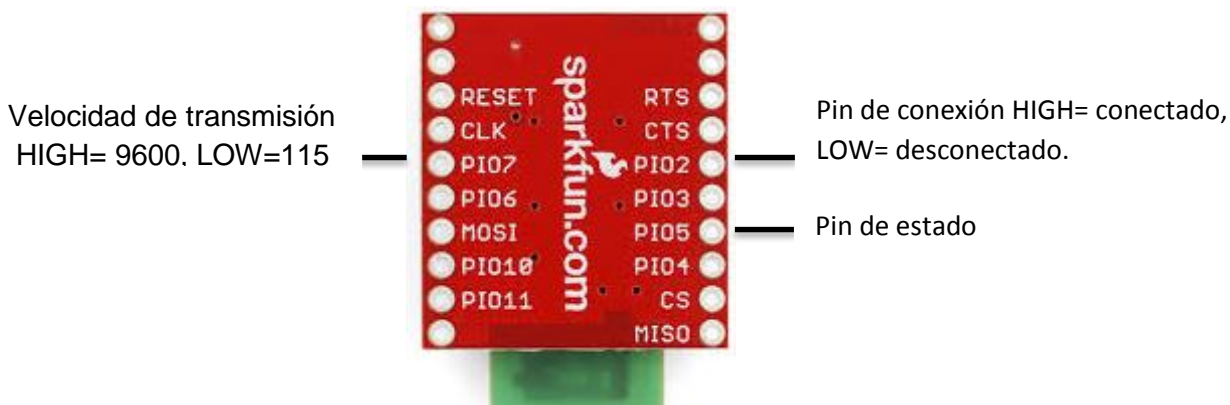


Fig. 3.14 Pines a utilizar en el modem Bluetooth RN 41-1N

Después de identificar los pines que se utilizaran y conectarlo al micro maestro Arduino, es necesario emparejar el dispositivo Bluetooth con el dispositivo móvil que contiene la aplicación a utilizar. Para esto es necesario ingresar a las opciones de conectividad del dispositivo móvil y buscar nuevos dispositivos Bluetooth, cuando haya aparecido el módulo Bluetooth seleccionamos la opción emparejar y con esto nos pedirá un código de emparejamiento que normalmente es “0000” o “1234”.

Para realizar la conexión del módulo Bluetooth Finalizado esto, el modulo estará emparejado al dispositivo móvil y listo para realizar la transmisión de información.

3.3 Resultados

El diseño que se escogió para la aplicación funciona perfectamente y el modo de operar también. La transmisión de los valores se realiza sin problemas y el módulo Bluetooth se comportó establemente. Se tenía la duda si los componentes utilizados en la aplicación funcionaria como se esperaba pero no se tuvo inconvenientes y los datos son exactos y tal y como se envían desde la aplicación.

En las siguientes imágenes se muestra como se envían datos desde la aplicación y por medio del puerto serial del micro maestro observamos los datos recibidos mediante la conexión Bluetooth. Se observa que los datos se obtienen sin ningún inconveniente. En la figura 3.15 se observa Interfaz gráfica de la aplicación desarrollada en App Inventor. En la figura 3.15 se observa el circuito utilizado para el acoplamiento del módem Bluetooth.

En la figura 3.16 se tiene la imagen de la conexión del módulo al micro maestro Arduino. Y en la figura 3.17 se observa él envió del valor I_{max} de la aplicación y recepción en el micro maestro. En la figura 3.18 se observa el envío de valor I_{min} de la aplicación y recepción en el micro maestro.

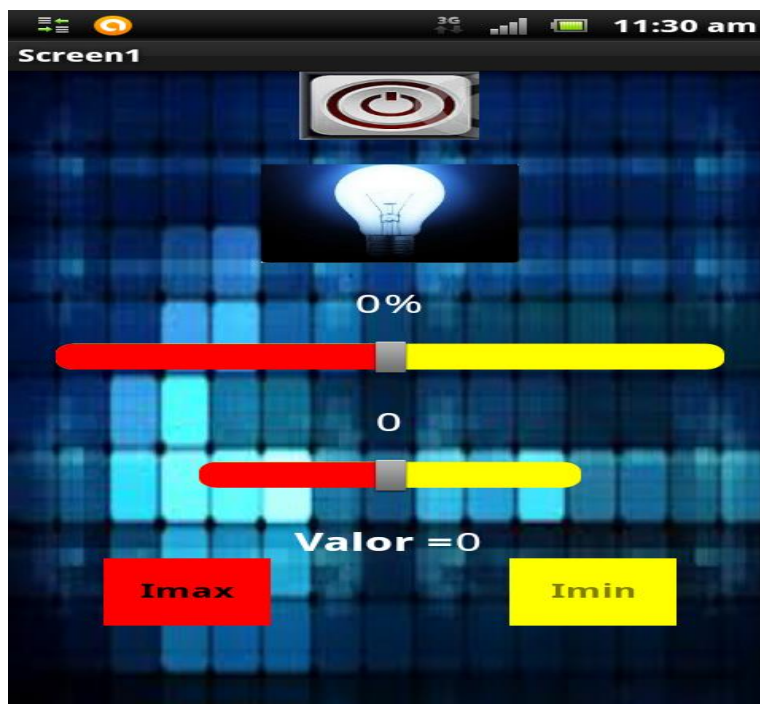


Fig.3.15 Interfaz gráfica de la aplicación desarrollada en App Inventor

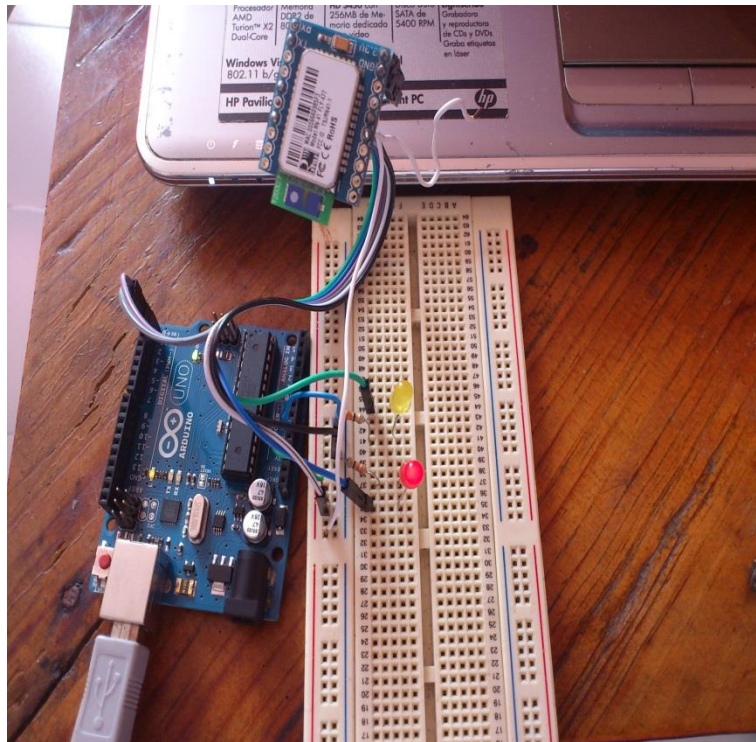


Fig.3.16 Circuito utilizado para el acoplamiento del módem Bluetooth

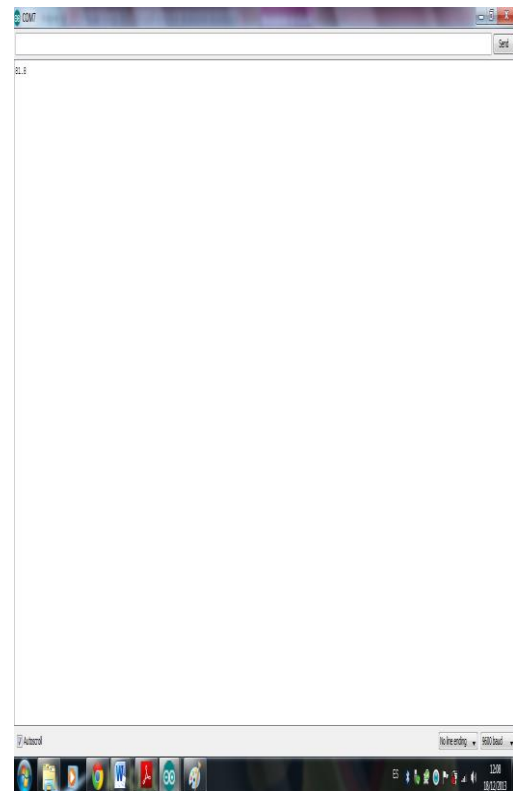
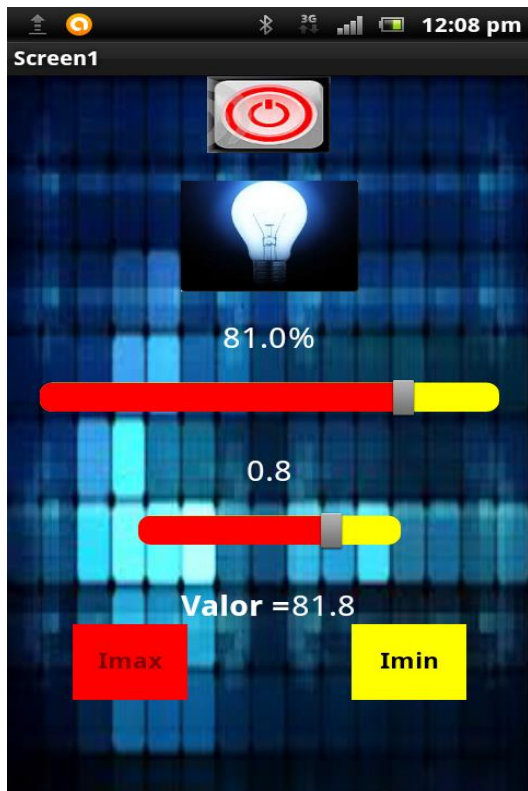


Fig. 3.17 Envío del valor I_{max} de la aplicación y recepción en el micro maestro

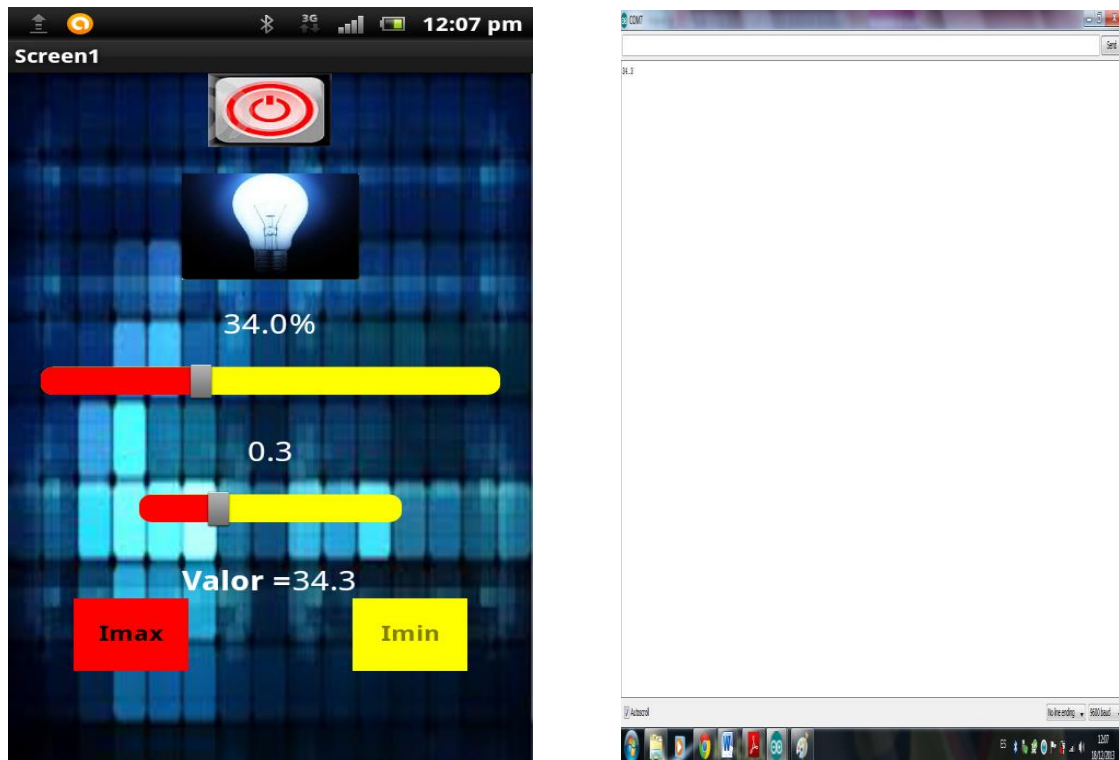


Fig. 3.18 Envío de valor Imin de la aplicación y recepción en el micro maestro

Observaciones y sugerencias

El presente proyecto se concluyó con la única observación de cambiar en la interfaz gráfica la imagen del foco incandescente por uno de leds. Y como sugerencias para futuras investigaciones, sería implementar comunicación desde la aplicación a otros dispositivos mediante Wi-Fi, con esto se tendría un sistema de comunicación más completo y se tendría una vía más de transmisión de datos aparte de la vía Bluetooth.

Conclusiones

Con la implementación del software que se presenta en este estudio se logra obtener los valores deseados con una eficacia aceptable. El comportamiento del modem Bluetooth es estable y la velocidad de transmisión de 9600 baudios hace efectiva la transmisión en el envío y recepción de datos. Se realiza una interfaz cómoda de forma que el usuario pueda interactuar sin ninguna dificultad pero con muy buenos resultados.

Dentro de las ventajas que presenta podemos destacar que gracias a los módulos bluetooth se puede realizar transmisión de datos inalámbricamente a distancias considerables sin ningún fallo de transmisión.

Referencias

[1] <https://www.youtube.com/watch?v=ocjXPIH7no>

[2]

[http://dis.um.es/~bmoros/privado/bibliografia/tutorial%20eclipse%20para%20novatos%20java%20\(Pollino\).pdf](http://dis.um.es/~bmoros/privado/bibliografia/tutorial%20eclipse%20para%20novatos%20java%20(Pollino).pdf)

[3] <http://www.neoteo.com/bluetooth-android-pic-led-hola-mundo/>

[4] <http://www.proyectosarduino.com.ar/creando-una-aplicacion-con-app-inventor/>

[5] <http://www.proyectosarduino.com.ar/adaptador-bluetooth-para-arduino/>

[6] <http://www.consejosgratis.es/todo-sobre-sistema-operativo-android/>

[7] <http://cosasdeingenieria.com/esp/index/item/15/modulo-bluetooth-para-microcontroladores-rn-41>

[8] <http://www.microelectronicos.net/?p=1075>

[9] <http://www.microelectronicos.net/?p=1075>