



**FACULTAD DE INGENIERÍA**  
**ESCUELA INGENIERÍA DE TELECOMUNICACIONES**

***LECTURA EN UN MEDIDOR ELÉCTRICO Y TRANSMISIÓN VÍA  
BLUETOOTH DE LOS DATOS A UN DISPOSITIVO ANDROID PARA  
SU PROCESAMIENTO Y FACTURACIÓN***

**TRABAJO ESPECIAL DE GRADO**

Presentado ante la

**UNIVERSIDAD CATÓLICA ANDRÉS BELLO**

Como parte de los requisitos para optar al título de

**INGENIERO DE TELECOMUNICACIONES**

**REALIZADO POR**

*Hernández G., Guillermo Eloy*

*Zapata H., Marlyn Andreina*

**PROFESOR GUÍA**

*Buonanno R., Nicola*

**FECHA**

*Febrero de 2013*



**FACULTAD DE INGENIERÍA**  
**ESCUELA INGENIERÍA DE TELECOMUNICACIONES**

***LECTURA EN UN MEDIDOR ELÉCTRICO Y TRANSMISIÓN VÍA  
BLUETOOTH DE LOS DATOS A UN DISPOSITIVO ANDROID PARA  
SU PROCESAMIENTO Y FACTURACIÓN***

**REALIZADO POR**

*Hernández G., Guillermo Eloy  
Zapata H., Marlyn Andreina*

**PROFESOR GUÍA**

*Buonanno R., Nicola*

**FECHA**

*Febrero de 2013*

**FACULTAD DE INGENIERÍA**  
**ESCUELA INGENIERÍA DE TELECOMUNICACIONES**

***LECTURA EN UN MEDIDOR ELÉCTRICO Y TRANSMISIÓN VÍA BLUETOOTH  
DE LOS DATOS A UN DISPOSITIVO ANDROID PARA SU PROCESAMIENTO Y  
FACTURACIÓN***

**Este jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido  
con el resultado: \_\_\_\_\_**

**JURADO EXAMINADOR**

Firma: \_\_\_\_\_ Firma: \_\_\_\_\_ Firma: \_\_\_\_\_  
Nombre: \_\_\_\_\_ Nombre: \_\_\_\_\_ Nombre: \_\_\_\_\_

**REALIZADO POR**

*Hernández G., Guillermo Eloy  
Zapata H., Marlyn Andreina*

**PROFESOR GUÍA**

*Buonanno R., Nicola*

**FECHA**

*Febrero de 2013*



Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

**UNIVERSIDAD CATÓLICA ANDRÉS BELLO**  
**LECTURA EN UN MEDIDOR ELÉCTRICO Y TRANSMISIÓN VÍA BLUETOOTH**  
**DE LOS DATOS A UN DISPOSITIVO ANDROID PARA SU PROCESAMIENTO Y**  
**FACTURACIÓN**

Hernández González, Guillermo Eloy

[guillermohdezg@gmail.com](mailto:guillermohdezg@gmail.com)

Zapata Hernández, Marlyn Andreina

[marlynzapata@gmail.com](mailto:marlynzapata@gmail.com)

**RESUMEN**

Actualmente el proceso de facturación que ofrece la compañía que presta el servicio eléctrico en Venezuela es lento y tedioso. Esto se debe a las distintas fases que debe cumplir el procesamiento de los datos desde el momento en que son extraídos por el efectivo de la empresa, hasta que finalmente, la factura llega a manos del consumidor. Por consiguiente se propone la modificación de este proceso.

Dicha modificación se basa en la innovación de un vatihorímetro o medidor de consumo eléctrico convencional, mediante la incorporación de una memoria (la cual almacenara los datos del cliente y de su respectivo consumo) y un módulo Bluetooth que establecerá una conexión con un dispositivo móvil que funciona bajo el sistema operativo OS Android que se encargará de mostrar en pantalla los datos antes mencionados, así como el envío a un servidor virtual que se ocupará de destinar dicha información tanto a la central de la empresa proveedora del servicio, como al usuario en cuestión.

**Palabras Clave:** Medidor de consumo eléctrico, Facturación, Bluetooth, Dispositivo móvil, Android OS.

## **DEDICATORIA**

Este logro esta dedicado a las personas que no solamente forman parte de mi vida sino que siempre fueron de gran apoyo para mí:

Primeramente quiero agradecer a Dios, que a pesar de todas las pruebas que me impuso en el camino siempre tiene una solución. Siempre me guió con su luz divina y por él estoy donde estoy.

A mi madre Marisol Hernández, una mujer excepcional, no solamente cumple su rol de madre, sino también es mi confidente y cómplice, siempre estuvo ahí para mí, brindándome su apoyo y consideración. Eres única y jamás podré agradecerte ni compensarte todo lo que has hecho por mí.

A mi padre Freddy Zapata, no tengo palabras para describir y agradecerle TODO el esfuerzo y dedicación que realizó para que pudiera lograr esta meta, sin tu ayuda no lo hubiese logrado. Eres mi todo.

A mi hermano Reynaldo Zapata, siempre brindándome amor a su manera.

A mi madrina Fabiola Martínez por ser mi apoyo incondicional, haber estado presente siempre en mi vida, y brindarme ayuda en los momentos más difíciles.

A Gisela González Coll una mujer única que se ha convertido en una segunda madre para mi, siempre estando ahí con sus sabios consejos y paciencia. GRACIAS.

La escuela de Ingeniería de Telecomunicaciones y todos sus miembros, por haberme brindado la oportunidad de compartir con personas tan agradables y ayudarme a culminar mis estudios en esta institución. Especialmente a Mileydis Martínez, José Pirrone, Abelis Salazar, y Nicola Buonanno.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo  
Android para su procesamiento y facturación.

---

A mis dos grandes amigas María José Martínez y María Alejandra Cabanzo, gracias por demostrarme el verdadero significado de la amistad y SIEMPRE estar a mi lado en todo momento.

A mis compañeros de toda la carrera y futuros colegas: Freddy Vásquez, Lisbeth Uruburu y Gabriel Guzmán. ¡Lo LOGRAMOS!

Y por ultimo pero no menos importante, le dedico este logro a mi compañero de Tesis Guillermo Eloy Hernández González, que a pesar de ser mi colega, es mi novio, amigo y confidente. Siempre me brindó su comprensión y está ahí con su luz en todo momento. Eres una persona maravillosa y no puedo alegar nada más que: TE AMO PROFUNDAMENTE. ¡Esta tesis no pudo con nosotros!

GRACIAS A TODOS NUEVAMENTE.

*Marlyn Zapata.*

## **DEDICATORIA**

Quisiera dedicar este proyecto, primeramente a mi madre Gisela María González Coll, quien siempre ha sido no solo un apoyo en los estudios y en todos los aspectos de mi vida, en los momentos bonitos y en los momentos más difíciles; sino que también es un ejemplo de superación ante la adversidad y las dificultades. Te amo mamá.

A mi padre Jesús Antonio Hernández Pérez, quien ha sido mi consejero, mano derecha y maestro de vida, enseñándome que aún existen personas en este mundo que son capaces de dejarlo todo y pasar trabajo, a cambio de saber que sus seres queridos están bien. Te amo viejito.

A tres personas que siempre han estado presentes en el largo camino universitario que pronto está por culminar: María Isabel González Coll, Delrosi Porras y Reina González Coll. Gracias por su apoyo infinito y fe en mí.

A quien sanguíneamente es mi primo pero la realidad es que es más que un hermano, Jorge Andrés López González. Gracias por tu presencia, por tus palabras, por tu sabiduría y por ser mi compañero de lucha en todas las batallas. Te amo hermano.

A mi compañera de tesis, amiga, compañera, novia. Sin ti, ni hubiese sido posible lograrlo ni hubiese sido tan grato llegar a la meta. Gracias por compañía, por tu amor incondicional, por tus palabras llenas de buena voluntad en todo momento, pero sobre todo, por tu presencia en mi vida y ese toque especial que le pones. Te amo. Lo logramos!!

A todos mis amigos y colegas que ya se graduaron, pero especialmente a aquellos con quienes tengo el placer y privilegio de compartir este grato momento, principalmente mis queridos Freddy Vásquez, Lisbeth Uruburu, y Gabriel Guzmán. Gracias por todo, muchachos.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo  
Android para su procesamiento y facturación.

---

Finalmente a Dios por haberme dado fuerzas siempre para continuar, aun cuando el recorrido se hacía más complejo e intransitable.

Gracias a todos.

*Guillermo Hernández.*

## **AGRADECIMIENTOS**

Agradecemos a todas las personas que estuvieron involucradas en la realización de este proyecto y que directa o indirectamente contribuyeron en él.

Al Ingeniero Nicola Buonanno por guiarnos en este largo camino y convertirse en más que un tutor y profesor para nosotros.

Al Ingeniero Biagio Cante quien a pesar de ser nuestro revisor, fue de MUCHISIMA ayuda y con sus conocimientos nos incentivó a lograrlo.

A Kenny, quien fue sumamente solidario, comprensivo y nunca dudo en brindarnos su ayuda, tiempo y conocimiento.

A Neida Zapata, quien con su buena voluntad nos facilitó lo que fue el equipo más importante y difícil de conseguir para el proyecto.

A Andrés Luzón Gómez, Manuel Santiago Yépez y Gabriel Alfonso Guzmán Rosario por dedicar parte de su tiempo en colaborar con nosotros.

A nuestros padres y familiares más cercanos por haber estado presentes en todo momento, con su apoyo incondicional en las buenas y en las malas.

Por último, pero no menos importante, a DIOS por iluminarnos siempre y guiarnos en el camino.

Gracias totales.

## ÍNDICE GENERAL

RESUMEN .....	i
DEDICATORIA .....	ii
DEDICATORIA .....	iv
AGRADECIMIENTOS .....	vi
ÍNDICE GENERAL .....	vii
ÍNDICE DE FIGURAS .....	xi
ÍNDICE DE TABLAS .....	xiii
CAPÍTULO I .....	1
PLANTEAMIENTO DEL PROYECTO .....	1
I.1.- Planteamiento del Problema .....	1
I.2.- Objetivos .....	2
I.2.1.- Objetivo General .....	2
I.2.2.- Objetivos Específicos.....	3
I.3.- Alcances y Limitaciones .....	3
CAPÍTULO II.....	5
MARCO TEÓRICO.....	5
II.1.- Medidor de Consumo Eléctrico .....	5
II.1.1.- Medidor de consumo eléctrico electromecánico convencional .....	5
II.1.2.- Medidor de consumo eléctrico digital. ....	8
II.2.- Tipos de Comunicación.....	12

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

II.2.1.- Comunicación Serial.....	12
II.2.2.- Estándar Bluetooth.....	14
II.3.- Microcontrolador .....	17
II.3.1.- Microcontrolador PICAXE .....	18
II.3.2.- Bluetooth Modem BlueSMiRF Silver .....	22
II.4.- Sistema Operativo .....	24
II.4.1.- Android O.S.....	24
II.5.- Lenguajes de Programación.....	26
II.5.1.- Java. ....	26
II.5.2.- BASIC .....	28
II.6.- Software .....	30
II.6.1.- PICAXE Programming Editor .....	30
II.6.2.- Processing .....	31
II.6.3.- TeraTerm Web 3.1.....	32
CAPÍTULO III .....	33
METODOLOGÍA.....	33
III.1.- Tipo de Investigación.....	33
III.2.- Fases del proyecto. ....	34
III.3.- Descripción de las actividades.....	35
CAPÍTULO IV.....	37
DESARROLLO .....	37
IV.1.- Rediseño del medidor eléctrico. ....	37

IV.1.1.- Estudio de los componentes y su funcionamiento.....	37
IV.1.2.- Selección del medidor, y diseño previo. ....	38
IV.1.3.- Construcción del circuito electrónico final y su adaptación al medidorelctrico. ....	39
IV.1.4.- Pruebas. ....	50
IV.2.- Diseño de la aplicación.....	51
IV.2.1.- Estudio del Sistema Operativo Android OS.....	51
IV.2.2.- Elección e Investigación referente al lenguaje de programación a utilizar.....	51
IV.2.3.- Desarrollo de la aplicación. ....	52
IV.2.4.- Pruebas. ....	56
IV.3.- Pruebas Finales.....	58
CAPÍTULO V.....	59
ANÁLISIS DE LOS RESULTADOS .....	59
V.1.- Medidor de consumo eléctrico. ....	59
V.2.- Aplicación.....	61
V.3.- Pruebas Finales.....	67
CONCLUSIONES Y RECOMENDACIONES .....	68
BIBLIOGRAFÍA .....	70
APÉNDICE A .....	76
APÉNDICE B.....	85
APÉNDICE C .....	102
ANEXO A.....	105

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo  
Android para su procesamiento y facturación.

---

**ANEXO B..... 113**

## ÍNDICE DE FIGURAS

FIGURA 1. ESTRUCTURA INTERNA DE UN MEDIDOR DE CONSUMO ELÉCTRICO ELECTROMECAÁNICO. ....	7
FIGURA 2. ESTRUCTURA DEL MEDIDOR DE CONSUMO ELÉCTRICO DIGITAL. (VISTA FRONTAL). ....	10
FIGURA 3. ESTRUCTURA INTERNA DEL MEDIDOR DE CONSUMO ELÉCTRICO DIGITAL. (VISTA ANTERIOR DE LA TARJETA). ....	11
FIGURA 4. ESQUEMATIZACIÓN DE UNA COMUNICACIÓN SERIAL. ....	14
FIGURA 5. UBICACIÓN DE PINES GROUND O DE TIERRA.) ....	20
FIGURA 6. . UBICACIÓN DE PINES DE ALIMENTACIÓN O VCC. ....	21
FIGURA 7. UBICACIÓN DE PINES DE ENTRADAS Y SALIDAS. ....	21
FIGURA 8. BLUE SMIRF SILVER (VISTA SUPERIOR) ....	23
FIGURA 9. BLUE SMIRF SILVER (VISTA INFERIOR). ....	23
FIGURA 10. FASES DEL PROYECTO. ....	34
FIGURA 11. MEDIDOR DE CONSUMO UTILIZADO (VISTA FRONTAL). ....	38
FIGURA 12. ESPECIFICACIONES DEL MEDIDOR DE CONSUMO ELÉCTRICO.. ....	39
FIGURA 13. CONEXIÓN DE LA PUNTAS DEL OSCILOSCOPIO PARA OBSERVAR LA SEÑAL EMITIDA. ....	40
FIGURA 14. CAPTURA DE PANTALLA DEL CÓDIGO DE PRUEBA. ....	43
FIGURA 15. DIAGRAMA DE FLUJO DEL FUNCIONAMIENTO DEL CÓDIGO.. ....	45
FIGURA 16. BOSQUEJO DEL CIRCUITO A IMPLEMENTAR.. ....	47
FIGURA 17. AMPLIFICADOR NO INVERSOR DE GANANCIA 4.. ....	47
FIGURA 18. MODEM BLUETOOTH RN-42 EN "MODO COMANDO". ....	49
FIGURA 19. CONFIGURACIÓN DEL TERATERM WEB 3.1. ....	49
FIGURA 20. CONFIGURACIÓN DEL MODEM BLUETOOTH RN-42. ....	50
FIGURA 21. INTERFAZ DEL IDE PROCESSING. MODO ANDROID. ....	53
FIGURA 22. PRUEBA DE CONEXIÓN ENTRE DISPOSITIVOS. ....	57
FIGURA 23. CONEXIÓN FINAL DEL PROYECTO. ....	58
FIGURA 24. CAPTURA PANTALLA DE LA SIMULACIÓN DE PRUEBA REALIZADA ....	60
FIGURA 25. CAPTURA PANTALLA DEL MENSAJE DE SOLICITUD DE PERMISO BLUETOOTH.. ....	62
FIGURA 26. CAPTURA PANTALLA DE LA LISTAS DE DISPOSITIVOS VINCULADOS.. ....	63
FIGURA 27. CAPTURA PANTALLA DE LA FACTURA DE PRUEBA. ....	64
FIGURA 28. CAPTURA PANTALLA DE LAS APLICACIONES DISPONIBLES PARA EL ENVÍO DEL CORREO ELECTRÓNICO. ....	65

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

FIGURA 29. CAPTURA PANTALLA DEL CORREO QUE SE DISPONE A ENVIAR A LAS DIRECCIONES GUARDADAS.. ..... 66

## ÍNDICE DE TABLAS

TABLA 1. DESCRIPCIÓN DE LAS ACTIVIDADES.....	36
--	----



## **CAPÍTULO I**

### **PLANTEAMIENTO DEL PROYECTO**

En este capítulo se puede observar el planteamiento del problema que llevó a cabo la ejecución de este Trabajo Especial de Grado. Así mismo se describe el problema que motivó la investigación, objetivos de estudio, y finalmente las limitaciones y el alcance del proyecto.

#### **I.1.- Planteamiento del Problema**

Actualmente el procedimiento que debe llevarse a cabo para que un usuario reciba su factura del consumo eléctrico mensual en su domicilio es lento y tedioso, tanto para la compañía, el designado de la compañía como para el cliente. Básicamente este proceso consta de cuatro (4) fases principales:

- Fase 1: El efectivo de la empresa se dirige a la residencia del cliente para recopilar la información correspondiente a su consumo mensual, visualizándola en un medidor eléctrico.
- Fase 2: Posteriormente esta misma persona se encarga de trasladar la información recopilada a la empresa.
- Fase 3: Esta información es almacenada en la base de datos de la compañía y es analizada con el fin de generar la factura con todo el reporte claramente especificado.
- Fase 4: Dicha factura es enviada al consumidor.

## Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

Actualmente se generan cantidades considerables de quejas con respecto al método mencionado, debido a fuertes retrasos en la recepción de la facturación con la consecuente suspensión del servicio al suscriptor antes de este tener en sus manos el recibo del mes.

Con el objetivo de optimizar, en cuanto a costo y tiempo, las fases del servicio de cobro del consumo de electricidad, y tomando en cuenta los inconvenientes mencionados, el avance tecnológico de los componentes electrónicos y los sistemas de comunicación inalámbricos, se plantea en este Trabajo Especial de Grado el rediseño del medidor eléctrico actual complementándolo con el desarrollo de una aplicación programada bajo ambiente Android, que tendrá como función la recepción de los datos almacenados en el medidor eléctrico, en un dispositivo que trabaja con dicho sistema operativo y el envío de dichos datos en forma de factura al correo electrónico del usuario al que corresponde el medidor en cuestión. La comunicación entre estos dos elementos se establecerá mediante una conexión basada en el estándar Bluetooth.

### **I.2.- Objetivos**

#### **I.2.1.- Objetivo General**

Optimizar el proceso de las fases del servicio de cobro de consumo de electricidad mediante el diseño e implementación de una aplicación basada en el sistema operativo Android fundamentado sobre aplicaciones Bluetooth para transmitir datos y cálculo de consumo en medidores eléctricos compatibles con la especificación Bluetooth, para su inmediata facturación.

### **I.2.2.- Objetivos Específicos**

- a. Realizar un estudio a fondo sobre la arquitectura interna y el funcionamiento de un medidor eléctrico convencional.
- b. Investigar acerca de los diferentes microprocesadores existentes en el mercado y su funcionamiento, compatibles con los requerimientos.
- c. Estudiar los distintos componentes electrónicos que permitirán el rediseño de la arquitectura interna del medidor eléctrico.
- d. Estudiar los dispositivos candidatos y las versiones del sistema operativo Android que cumplan los requerimientos establecidos en el proyecto.
- e. Desarrollar la aplicación de transmisión y cálculo que será capaz de recibir, analizar los datos y enviar la factura correspondiente al consumo.
- f. Realizar pruebas que evidencien que la conexión entre ambos dispositivos fue exitosa.

### **I.3.- Alcances y Limitaciones**

Una limitación que será decisiva para el desarrollo de este Trabajo Especial De Grado es la factibilidad de conseguir algunos de los componentes electrónicos que se usaran para el rediseño del medidor eléctrico, fundamentalmente los microprocesadores

## Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

aptos para el almacenamiento de data y que operen usando transmisión Bluetooth, ya que son difíciles de conseguir en el mercado venezolano.

Otra barrera será encontrar un medidor eléctrico en buenas condiciones, debido a que, de no estarlo retrasará la ejecución de lo que se tiene planteado en los objetivos, al invertir tiempo en revisar y corregir la(s) fallas(s) que este pueda presentar.

Para la realización de este estudio dependerá de un espacio físico amplio y equipado con todos los dispositivos necesarios para realizar las mediciones respectivas.

El sistema desarrollado para este Trabajo Especial De Grado incluirá una factura que contendrá la información necesaria para que se pueda efectuar el pago del servicio sin ningún inconveniente, sin embargo no tendrá el mismo formato que las facturas del consumo eléctrico habituales.

Este trabajo incluirá el diseño del medidor eléctrico modificado junto con el diseño de la aplicación Android, ambos bien especificados.

Se añadirá al trabajo un Manual de Usuario que contendrá tanto las instrucciones de funcionamiento de la aplicación como las del medidor eléctrico, y también incluirá las medidas necesarias de precaución para asegurar su durabilidad en el tiempo y su correcto uso.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

A continuación, se describe toda la teoría necesaria sobre la cual estará fundamentado el proyecto. Este capítulo está compuesto por: Medidor de consumo eléctrico, Tipos de comunicaciones utilizados en esta investigación, Microcontrolador, Sistema operativo Android O.S y finalmente los diferentes Lenguajes de programación y Softwares aplicados para desarrollo de este Trabajo Especial de Grado.

#### **II.1.-Medidor de Consumo Eléctrico**

Un medidor de consumo eléctrico también conocido como: contador eléctrico ó Vatihorímetro es un dispositivo que se encarga de medir el consumo de potencia de un circuito o un servicio eléctrico doméstico. Existen distintos tipos de Vatihorímetros, según su tecnología de construcción, su conexión interna y de acuerdo a la conexión de la red. A continuación se describen los tipos de medidores que fundamentaron el desarrollo del proyecto.

##### **II.1.1.-Medidor de consumo eléctrico electromecánico convencional**

Un medidor de energía tipo inducción está constituido por un núcleo de chapa magnética en el que van montados dos bobinas, una en serie con el conductor por el que

circula la corriente principal, y que se denomina bobina de intensidad (ó corriente), y otra en bobina en derivación sobre los dos conductores, denominada bobina de tensión. Los flujos magnéticos producidos por ambas bobinas están desfasadas 90° y actúan sobre un disco rotórico de aluminio. Estos flujos producen pares de torque de giros, que a su vez provocan un movimiento de rotación del disco de aluminio a una velocidad angular proporcional a la potencia. El disco de aluminio es, además, frenado por un imán (freno de corrientes parásitas) de tal forma que la velocidad angular del disco sea proporcional a la carga. El aparato está completado por un registrador, que mediante un sistema de transmisión indica los kilovatios-hora consumidos, especificación según Ríos, F. (2000).

#### II.1.1.1.- -Estructura

El medidor está constituido por las siguientes partes mas importantes:

1. Bobina de Tensión
2. Bobina de Intensidad
3. Imán de frenado
4. Regulación fina
5. Regulación gruesa
6. Disco
7. Sistema de Transmisión
8. Terminales de conexión

La representación esquemática de la estructura de un medidor de inducción, se visualiza en la Figura 1:

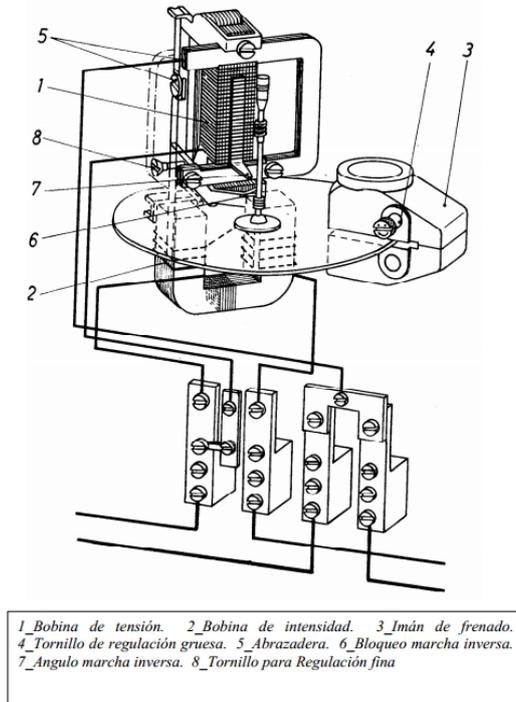


Figura 1. Estructura interna de un medidor de consumo eléctrico electromecánico. Fuente Ríos, F. (2002)

#### II.1.1.2.- Características Principales

En la placa de características de un medidor de energía se indica:

a) Corriente nominal ( $I_n$ ): corriente para la cual el medidor es diseñado y que sirve de referencia para la realización de ensayos y verificaciones.

b) Corriente máxima ( $I_{m\acute{a}x}$ ): es la intensidad límite, es decir, el máximo amperaje que puede ser conducido en régimen permanente por la corriente del medidor, sin que su error porcentual y temperatura admisible sean superados. Este valor de la corriente límite se

indica entre paréntesis detrás de la corriente nominal  $I_n(I_{max})$ ; por ejemplo: 10 (20) A, 10(40) A, 15(60) A, 15 (100)A., etc.

c) Tensión nominal: Tensión para la cual el medidor es diseñado y sirve de referencia para la realización de pruebas.

d) Constante del disco (Kh): expresada en Wh/revolución, representa el número de vatios-hora correspondientes a una revolución o vuelta completa del disco; la misma expresada en revolución/KWh, define el número de revoluciones correspondiente a un kWh que debe dar el disco.

e) Clase de precisión: Es el valor máximo del error de medición expresado en porcentaje para el cual fue diseñado el medidor dentro del rango 10% de corriente nominal y su corriente máxima.

### **II.1.2.-Medidor de consumo eléctrico digital.**

La medición de energía a través de estos dispositivos se realiza por medio de un proceso análogo-digital utilizando un microprocesador para producir pulsos de salida y cuya frecuencia es proporcional a los Vatios-Hora consumidos. Están contruidos con dispositivos electrónicos, generalmente son de mayor precisión que los electromagnéticos.

El contador utilizado para el desarrollo del proyecto es de tipo: estático monofásico, lo que refiere a tenencia de un solo conductor activo o fase y un conductor no activo o neutro. Este Vatihorímetro mide la energía activa en redes de energía eléctrica monofásica de dos hilos de corriente alterna. El medidor cumple con la norma internacional IEC 61036 según la *International Electrotechnical Commission* (Comisión Internacional de Electrotecnia). Este estándar trata sobre los requisitos generales y las condiciones de

ensayo respecto a los equipos de medición de electricidad. La gran precisión y fiabilidad de estos dispositivos impide la acción ilegal de fraude de electricidad. Se debe indicar que los medidores electrónicos se diseñan con un rango de tensión sin que se vea afectada su precisión.

#### II.1.2.1.- Estructura

Las especificaciones mencionadas a continuación provienen del Grupo HOLLEY (2010). La estructura exterior de los medidores adopta base de policarbonato y tapa principal, de acuerdo con la exigencia de aislamiento de protección, la estructura es sólida y resistente. Este dispositivo esta constituido por las siguientes partes más importantes:

1. **Contador de pantalla protectora simple**, Dispositivo que consta de un motor que hace girar los cilindros numerados. Este sistema cuenta con cinco (5) dígitos enteros y un (1) decimal (cilindro de color rojo).
2. **Memoria y Microcontrolador**, El microcontrolador posee dos entradas analógicas de corriente y una entrada de tensión. El microcontrolador convierte esas señales análogas en digitales, con conversores A/D, y la memoria no-volátil tiene como función, mantener los datos de manera invariable durante 40 años después de apagar.
3. **Varistor**, componente electrónico utilizado para proteger el circuito contra variaciones de tensión de forma que, cuando se active la corriente no afecte componentes sensibles.
4. **Led de Encendido**, indica el estado de funcionamiento del medidor.
5. **Led de Parpadeo**, indica cuando se recibe un impulso, con una frecuencia de 1600 [pulsos/Kwh].

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

6. **Capacitador CBB62**, dispositivo especial utilizado en circuitos de impulsos, la función de este capacitador es generar un impulso una vez consumido 1Kwh, el cual activa el motor del contador de pantalla protectora simple.

7. **Terminales de conexión**, se refiere a los terminales donde se conecta la carga y al sistema eléctrico. Son cuatro terminales: Línea, fase, línea prima, y fase prima.

La representación gráfica de la estructura del dispositivo se muestra en las Figuras 2 y 3.

Figura 2. Estructura del Medidor de Consumo Eléctrico Digital. (Vista frontal).

Autoría Propia



Figura 3. Estructura interna del medidor de consumo eléctrico Digital. (Vista anterior de la tarjeta). Autoría propia

#### II.1.2.2.- Mantenimiento

El medidor debe ser instalado en interiores, de lo contrario hay que utilizar la caja del medidor especial para protegerlo. El ambiente cercano de los medidores no debe contener aire cáustico y el medidor debe evitar la influencia de la arena sucia, sal-niebla y el rocío. El transporte y almacenamiento de los medidores deberá estar en buena condición de empaque y por último el cambio repentino de la temperatura del ambiente y la humedad relativa debe ser evitado. (HOLLEY, 2010).

## **II.2.-Tipos de Comunicación**

La comunicación entre dispositivos diferentes representó un elemento esencial para la ejecución del diseño del proyecto. Los tipos de comunicaciones manejados serán explicados a continuación.

### **II.2.1.-Comunicación Serial**

La comunicación serial consiste en el envío de un bit de información de manera secuencial, es decir un bit a la vez y a un ritmo acordado entre el emisor y el receptor, aun cuando esta comunicación es más lenta que la comunicación en paralelo, la cual permite la transmisión de un byte completo por vez, este método es más sencillo y puede alcanzar mayores distancias

La comunicación serial en computadores ha seguido los estándares definidos en 1969 por el RS-232 (*Recommended Standard 232*). Este estándar define las características eléctricas, la sincronización de las señales, el significado de las señales, el tamaño físico y la ubicación de los pines conectores

Para realizar la comunicación se utilizan 3 líneas de transmisión: Tierra, Transmitir (Tx) y Recibir (Rx), ver Figura 4. Según *National Instruments* (2004) las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

a. Velocidad de transmisión (*baud rate*): Indica el número de bits por segundo que se transfieren, y se mide en baudios (*bauds*). Esta velocidad de transmisión de inversamente proporcional a la distancia máxima posible entre los dispositivos que se desean comunicar.

b. Bits de datos: Se refiere a la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits. Las cantidades más comunes de bits por paquete son 5, 7 y 8 bits. Un paquete se refiere a una transferencia de byte, incluyendo los bits de inicio/parada, bits de datos, y paridad. Debido a que el número actual de bits depende en el protocolo que se seleccione, el término paquete se usa para referirse a todos los casos.

c. Bits de parada: Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits. Debido a la manera como se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj, es posible que los dos dispositivos no estén sincronizados. Por lo tanto, los bits de parada no sólo indican el fin de la transmisión sino además dan un margen de tolerancia para esa diferencia de los relojes. Mientras más bits de parada se usen, mayor será la tolerancia a la sincronía de los relojes, sin embargo la transmisión será más lenta.

d. Paridad: Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible.

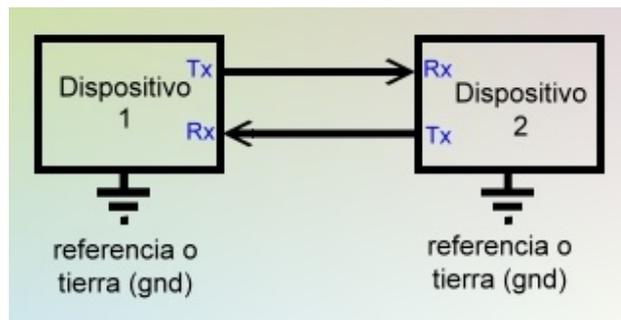


Figura 4. Esquematación de una comunicación serial. Fuente: Sistemas Digitales de México. (2012)

### II.2.2.-Estándar Bluetooth

Bluetooth es un estándar global de comunicación inalámbrica definido en el estándar IEEE 802.15.1, permitiendo la transmisión de voz y datos entre diferentes equipos mediante un enlace de radiofrecuencia.

Los principales objetivos que se pretendieron conseguir con esta norma fueron: facilitar la comunicación entre equipos móviles y fijos eliminando cables entre éstos, ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Bluetooth se ha diseñado para permitir conexiones de bajo ancho de banda inalámbrico.

El origen de la tecnología Bluetooth se remonta al año 1994 cuando la empresa Ericsson inició una investigación para crear una interfaz a través de ondas de radio, mínimo consumo de energía y bajo costo, para la interconexión entre sus teléfonos móviles y otros accesorios (por ejemplo un dispositivo manos libres), con la intención de eliminar los cables entre los dispositivos. Este estudio se inició a partir de un largo proyecto llamado MC link que investigaba sobre multicomunicadores conectados a una red celular hasta llegar a un enlace de radio de corto alcance, con el avance de este proyecto se descubrió que este tipo de enlace podía ser utilizado ampliamente en un gran número de aplicaciones, ya que su principal virtud es el uso de un chip de radio relativamente económico. A comienzos de 1997, conforme avanzaba el proyecto MC link, la empresa Ericsson fue despertando el interés de otros fabricantes de equipos portátiles. Sin embargo, para que el sistema fuera exitoso, se debía buscar la manera de implementar ésta tecnología en una gran cantidad de dispositivos. Esto fue lo que originó a principios de 1998, la creación de un grupo llamado SIG (Special Interest Group) ó Grupo de Interés Especial constituido por cinco empresas: Ericsson, Nokia, IBM, Toshiba e Intel denominadas como compañías promotoras. En Julio de 1999, el SIG publicó la versión 1.0 del estándar 802.15.1. Las compañías 3Com, Agere, Microsoft y Motorola se sumaron al SIG. La idea de nombrar “Bluetooth” al estándar IEEE 802.15.1 surgió debido al personaje de “Harald Blatand”, quien fue un rey danés vikingo, durante su reinado (940-981) fue supuesto a unir y controlar los países de Dinamarca y Noruega. Debido a que la nueva tecnología de radio fue esperada para unificar la telecomunicación y las industrias de computación, pareció adaptarse el nombre de este personaje. Un analista historiador del equipo

## Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

(SIG) propuso “Bluetooth” como el nombre interno. (Gama, Villa & Noguero, s.f., p. 2).

Este importante protocolo es muy usado debido a su gran variedad de aplicaciones, según la Bluetooth SIG Inc. (2012) esta tecnología tenía como función principal reemplazar los cables entre objetos específicos, pero ahora se puede lograr mucho más por el desarrollo del mismo como:

- Conectar televisores
- Sincronización con impresora y fax
- Conectar reproductores de música al vehículo
- Transferencia de fichas de contactos, citas y recordatorios entre dispositivos
- Reemplazo de la tradicional comunicación por cable entre equipos GPS (*Global Positioning System* ó Sistema de Posicionamiento Global) y equipos médicos.
- Controles remotos (tradicionalmente dominado por el infrarrojo).

Solo por nombrar algunos, antes era imposible de creer las conexiones que se pueden lograr en esta época debido al Bluetooth.

Una muy buena información, proviene de Curt Franklin y Julia Layton (2011). Estos explican que Bluetooth transmite datos a través de ondas de radio de baja potencia, utilizando como frecuencia la banda de 2.45 GHz (entre 2.402 GHz y 2.480 GHz más precisamente). Puede conectar hasta 8 dispositivos al simultáneo que se encuentren a unos 10 metros (32 pies) de radio. Utiliza la técnica SSFH (*Spread Spectrum Frequency Hopping* o Espectro Disperso con Saltos de Frecuencia), de forma de lograr que ninguno de los dispositivos interfiera con otro. SSFH funciona de forma que un dispositivo use 79 frecuencias aleatorias e individuales dentro de un rango designado, que irán variando con el

tiempo (en el caso de Bluetooth el cambio de frecuencia ocurre 1600 veces cada segundo), significando esto que más dispositivos pueden hacer uso más completo y eficaz de un trozo limitado del espectro de frecuencia destinado.

Cuando los dispositivos con capacidad de utilizar Bluetooth entran en rango con otros, se comunican automáticamente con el fin de determinar si tienen datos que compartir o alguno necesita controlar al otro (caso ejemplo, un dispositivo manos libres conectado vía Bluetooth a un Smartphone). El usuario no necesita presionar ningún botón o indicar un comando para que ocurra dicha comunicación, el proceso es automático. Una vez determinado esto, se forma una red. Los sistemas Bluetooth crean una PAN (*Personal Area Network* ó Red de Área Personal) o PICONET que trata de una red donde todos sus nodos se conectan usando el Estándar Bluetooth. Una vez que la PICONET se establece, los miembros de la red alternan de forma aleatoria los saltos entre las frecuencias disponibles al unísono, de forma de evitar colisionar con otras PICONET. Un ejemplo característico es el de una habitación moderna, que posee un sistema Home Theater, un BlueRay, un receptor de TV Digital, una TV LED, un teléfono inalámbrico y un computador personal. Cada uno de estos dispositivos trabaja bajo el estándar Bluetooth y cada uno, como sistema propio, conforma una PICONET para comunicarse entre la unidad principal y periférica

La versión inicial de Bluetooth (Bluetooth 1.0) tenía como velocidad máxima de transferencia 1 Mbps mientras que Bluetooth 2.0 puede manejar hasta velocidades de 3 Mbps, además, la versión 2.0 es compatible con la versión anterior.

### **II.3.-Microcontrolador**

Un microcontrolador como su nombre indica es un circuito integrado programable muy pequeño (micro), capaz de ejecutar las órdenes grabadas en su memoria (controlador). Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

Según Marco A. López Grande y José Luis Fernández Souto (2006) indican que un microcontrolador PIC (microcontrolador programable) es a menudo descrito como un "ordenador en un chip". Es un circuito integrado que contiene memoria, unidades procesadoras y circuitos de entrada y salida, en una sola unidad. Los microcontroladores son comprados sin ninguna información grabada en ellos y luego programados con un código de línea específico de control. Una vez programado, este microcontrolador es introducido en algún producto para hacerlo más inteligente y fácil de usar. Un microcontrolador puede a menudo reemplazar a un gran número de partes separadas, o incluso a un circuito electrónico completo. A continuación se indican los diferentes microcontroladores utilizados en el desarrollo del proyecto.

### **II.3.1.-Microcontrolador PICAXE**

El sistema "*PICAXE*" es un sistema de microcontrolador fácil de programar que utiliza un lenguaje BASIC muy simple. Una fuente importante viene dada por Marco A. López Grande y José Luis Fernández Souto (2006) donde mencionan que el "sistema *PICAXE* explota las características únicas de la nueva generación de microcontroladores de bajo costo FLASH. Estos microcontroladores pueden ser programados una y otra vez sin la necesidad de un costoso programador PIC".

Unas de las principales ventajas de este microcontrolador radica en su sencillez, debido a que no obliga la demanda de algún programador experto en un lenguaje de programación en específico, ya que *PICAXE* es configurado con un simple programa en BASIC o un diagrama de flujo, mediante una conexión de tres alambres conectada al puerto serie del ordenador.

Según Marco A. López Grande y José Luis Fernández Souto (2006) el circuito *PICAXE* utiliza únicamente tres componentes y puede ser ensamblado fácilmente en un tablero experimental para componentes electrónicos, en una placa corriente o en una placa PCB. EL sistema *PICAXE* está disponible en dos variedades - 18 pines y 28 pines. El controlador *PICAXE-28* provee 22 pines de entrada/salida (8 salidas digitales, 8 entradas digitales y 4 entradas analógicas). El sistema *PICAXE-18* provee 8 salidas y 5 entradas.

Las características principales del sistema *PICAXE* son las siguientes:

- Es un microcontrolador de bajo costo
- Circuito de fácil construcción.
- Hasta 8 entradas, 8 salidas y 4 canales analógicos.
- El proceso de descarga mediante el cable serial al ordenador, es de rápida velocidad.
- Software "Editor de Programación" gratuito y de fácil uso. (Editor de diagramas de flujo incluido).
- Extenso número de manuales gratuitos en línea.

El sistema *PICAXE* consiste en tres componentes principales dadas por Marco A. López Grande y José Luis Fernández Souto (2006):

- El Software "Editor de Programación": Este software debe ser ejecutado en un ordenador y permite utilizar el teclado del ordenador para escribir programas en un simple lenguaje BASIC. Los programas también pueden generarse dibujando diagramas de flujo.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

- El cable serie: Este es el cable que conecta el sistema *PICAXE* al ordenador. El cable sólo necesita ser conectado durante la descarga de programas. No debe ser conectado cuando el *PICAXE* está siendo ejecutado debido a que el programa está permanentemente almacenado en el chip *PICAXE*. Para la descarga del software se debe tener un voltaje equivalente a 5 V en el tablero electrónico.
- El chip *PICAXE* y el tablero electrónico: El microcontrolador *PICAXE* ejecuta programas que han sido descargados al mismo. Sin embargo, para operar, el chip debe ser montado en un tablero electrónico que provea una conexión al chip microcontrolador. En las Figuras 5, 6 y 7 se muestra detalladamente los pines de conexión del tablero *PICAXE*.

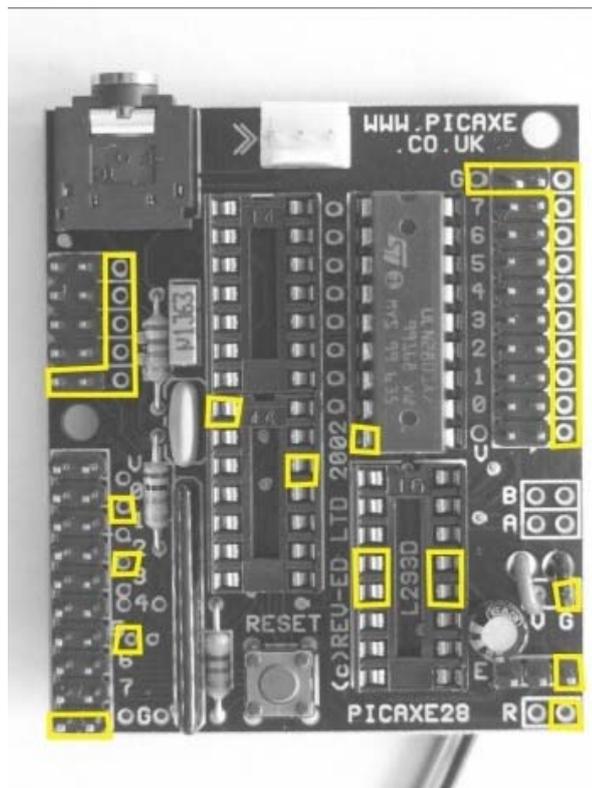


Figura 5. Ubicación de pines Ground o de Tierra. Fuente: *Let's make Robots*. (2008)

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

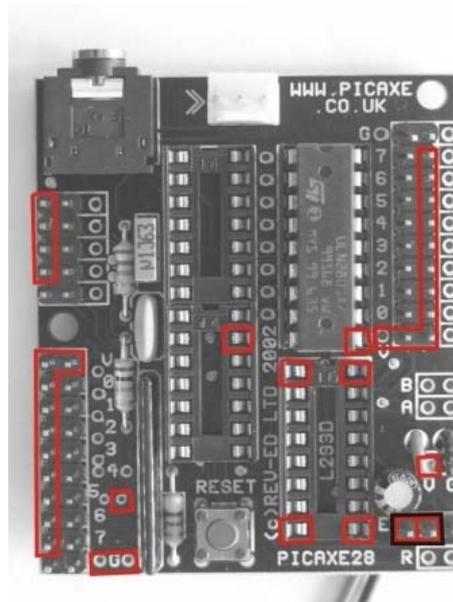


Figura 6. . Ubicación de pines de Alimentación o VCC. Fuente: Let`s make Robots. (2008)

Figura 7. Ubicación de pines de Entradas y Salidas. Fuente: Let`s make Robots. (2008)

Las especificaciones de este Microcontrolador PICAXE se pueden observar en el Datasheet contenido en el Anexo A.

### **II.3.2.-Bluetooth Modem BlueSMiRF Silver**

El *BlueSMiRF* es un módem que reemplaza el cable serial permitiendo el establecimiento de una comunicación de tipo serial, inalámbrica, a través de Bluetooth. Los dos modelos más utilizados son las versiones *Gold* (Oro) y *Silver* (Plata), los cuales emplean los módulos RN-41 y RN-42 respectivamente, diferenciándose principalmente en que el RN-41 tiene un rango más amplio de distancia de cobertura (100 m), que el RN-42 (18 m).

#### **II.3.2.1.- Blue SMiRF Silver.**

Este módem trabaja como una línea de comunicación serial común tipo Transmisor-Receptor (Tx/Rx) por el cual puede transportarse cualquier flujo de datos seriales en el rango de velocidades comprendidos desde 2.400 bps hasta 115.200 bps. La unidad se puede alimentar desde los 3.3 V hasta los 6 V.

Según la web Sparkfun.com, las características que presenta este módulo RN-42 son las siguientes:

- Alcance de 18 m.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

- Antena Incorporada.
- Comunicación Serial: 2.400 a 115.200 bps.
- Voltaje de Operación: 3.3 a 6 V.
- Frecuencia de Operación: 2.4 – 2.524 GHz.
- Conexión Encriptada.



Figura 8. Blue SMiRF Silver (Vista Superior). Fuente: Sparkfun. (2012)



Figura 9. Blue SMiRF Silver (Vista Inferior). Fuente: Sparkfun. (2012)

Las especificaciones de este modem Bluetooth se pueden observar en el Datasheet contenido en el Anexo B

## **II.4.-Sistema Operativo**

Un Sistema Operativo es el software encargado de ejercer el control y coordinar el uso del hardware entre diferentes programas de aplicación y los diferentes usuarios. Para este Trabajo Especial de Grado se trabajó específicamente con el sistema operativo Android O.S, el cual es explicado de inmediato.

### **II.4.1.-Android O.S.**

Al referirse al sistema operativo Android OS, se habla de una verdadera revolución. Este sistema operativo, está basado en Linux, junto con aplicaciones *middleware* y aplicaciones clave, está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la *Open Handset Alliance*, la cual es liderada por Google.

Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la *Open Handset Alliance*, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se han sobrepasado las 400.000 aplicaciones (de las cuales, dos tercios son gratuitas) disponibles para la tienda de aplicaciones oficial de Android: *Google Play*, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android, como pueden ser la *App Store* de Amazon o la tienda de aplicaciones *Samsung Apps* de Samsung.

## Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un *framework* (Marco) Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen: un administrador de interfaz gráfica, un *framework* OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica *OpenGL ES 2.0 3D*, un motor de renderizado *WebKit*, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

Las versiones de este importante sistema operativo son las siguientes:

- Versión 1.0.
- Versión 1.1.
- Versión 1.5 (Conocida como *Cupcake*).
- Versión 1.6 (Conocida como *Donut*).
- Versiones 2.0 y 2.1 (Conocida como *Eclair*).
- Versión 2.2 (Conocida como *Froyo*).
- Versión 2.3 (Conocida como *Gingerbread*).
- Versiones 3.0, 3.1 y 3.2 (Conocida como *Honeycomb*).
- Versión 4.0 (Conocida como *Ice Cream Sandwich*).

Uno de los puntos positivos que tiene Android es con respecto a la programación de sus aplicaciones. Android ofrece una plataforma abierta para desarrollar cualquier aplicación que el usuario se pueda imaginar, y un mercado abierto para distribuir sus productos a una gran y creciente comunidad de usuarios. Una forma de desarrollar dichas aplicaciones es a través de un IDE (*Integrated Development Enviroment*).

## II.5.-Lenguajes de Programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

### II.5.1.-Java.

Se conoce como Java a un lenguaje de programación orientado a objetos, desarrollado por *Sun Microsystems* a principios de los años 90. “Java está relacionado con C++, que es descendiente directo de C. Java hereda la mayor parte de su carácter de estos dos lenguajes. De C, Java deriva su sintaxis y muchas de sus características orientadas a objetos fueron consecuencia de la influencia de C++.” (Schildt, 2007).

Java fue concebido por James Gosling, Patrick Naughton, Chris Warth, Ed Frank y Mike Sheridan en *Sun Microsystems, Inc* (Actualmente propiedad de Oracle). El impulso inicial por el cual fue creado no fue el Internet, sino la necesidad de un lenguaje de programación que pudiera ser utilizado para crear *software* que pudiera correr en dispositivos electrodomésticos como controles remotos u hornos microondas. Un segundo factor, más importante, surge; y fue, la *World Wide Web*. De hecho, Schildt (2007) indica que:

Si el mundo de la *Web* no se hubiese desarrollado al mismo tiempo que Java estaba siendo implementado, Java podría haber sido simplemente un lenguaje útil para programación de dispositivos electrónicos. Sin embargo, con la aparición de la *World Wide Web*, Java fue lanzado a la vanguardia del diseño de lenguajes de programación, porque la *Web* también demandaba programas que fuesen portables. (Schildt. 2007 p. 10)

Algunas de las cualidades que describen a Java son:

- Simple: Su aprendizaje y utilización resultan sencillos para el programador profesional.
- Seguro: Java confina los programas a ser ejecutados en un ambiente controlado sin permitirles el acceso completo a los recursos de la computadora.
- Portable: Un programa Java puede ser ejecutado en diferentes sistemas conectados a la red.
- Orientado a Objetos: Facilita la aproximación clara y aprovechable de los objetos, al tomar ideas de entornos de orientación a objetos de las últimas décadas.
- Robusto: Java restringe al programador algunas áreas clave con la finalidad de hallar rápidamente errores en el desarrollo de un determinado programa.
- Multihilos: Permite la escritura de programas que realicen distintas operaciones simultáneamente.
- Alto rendimiento: Está diseñado para que exista máxima efectividad y para aprovechar en la mayor cantidad posible los recursos existentes.
- Distribuido: Permite a un programa invocar métodos situados en computadoras diferentes a través de la red.

- Dinámico: Java permite enlazar el código dinámicamente de una forma segura y viable.

En la actualidad, uno de los sistemas operativos más importantes, como lo es el Android OS, basa el diseño de programas y aplicaciones en el lenguaje de programación Java; las aplicaciones se ejecutan en un entramado Java (aplicaciones orientadas a objetos) sobre el núcleo de bibliotecas Java en una máquina virtual. Aplicaciones como correo electrónico, programa de SMS, calendario, mapas, navegador, contactos, etc., están escritas bajo el lenguaje de programación Java. Para poder desarrollarlas, es necesario tener conocimiento de Java, y poseer el Android SDK (*Software Development Kit*) el cual utiliza habitualmente dicho lenguaje de programación.

### **II.5.2.-BASIC**

Es una familia de lenguajes de programación de alto nivel, que inicialmente estaba ideado para fines educativos, principalmente para aquellos profesores y estudiantes que no eran parte de la especialidad de ciencias de tal forma que pudieran aprender y practicar como programar una computadora. Fue diseñado en New Hampshire, en los Estados Unidos, por John George Kemeny y Thomas Eugene Kurtz en el año 1964 con el nombre original *Dartmouth BASIC*.

Sus siglas significan *Begginer's All-purpose Symbolic Instruccion Code* o Código Simbólico de Instrucciones de Propósito General para Principiantes. Es de los lenguajes de programación más sencillos de aprender y es de los más populares dentro de la Informática. *BASIC* consta de instrucciones, funciones y comandos, que son palabras en inglés que tienen un significado particular para el computador. “Los programas de *BASIC* son un

## Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

conjunto de instrucciones, comandos y funciones que realizan una tarea determinada.” (De Lucas Linares, 1991.)

Para el diseño de *BASIC* se tomaron en cuenta 8 principios básicos, según el portal Integración Tecnológica (2001):

- Ser fácil de usar para los principiantes.
- Ser un lenguaje de propósito general.
- Permitir que los expertos añadieran características avanzadas, mientras que el lenguaje permanecía simple para los principiantes.
- Ser interactivo.
- Proveer mensajes de errores claros y amigables.
- Responder rápido a los programas pequeños.
- No requerir conocimiento del Hardware de la computadora.
- Proteger al usuario del Sistema Operativo.

Por último, la sintaxis mínima de BASIC requiere de los comandos *Let*, *Input*, *Print*, *If* y finalmente el *Goto*. En la actualidad existen una enorme cantidad de variaciones del lenguaje *BASIC*, que son aplicables a distintas aplicaciones y formatos de operación.

### II.5.2.1.- PICAXE BASIC.

Este lenguaje es mucho más sencillo para la programación de microcontroladores que los lenguajes tradicionales como *Assembler* o *C*. Está diseñado para permitir que hasta los usuarios con la menor experiencia y conocimiento de programación, puedan desarrollar de formar rápida, sencilla y efectiva, programas para los microprocesadores PICAXE.

PICAXE BASIC está conformado por los comandos más clásicos y básicos del lenguaje sobre el que esta soportado (*BASIC*), así como se complementa muy bien con unos comandos específicos que facilitan la ejecución de instrucciones y rutinas al microprocesador PICAXE.

## **II.6.-Software**

Conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora.

El software es desarrollado mediante distintos lenguajes de programación que permiten controlar el comportamiento de una máquina.

### **II.6.1.-PICAXE Programming Editor**

Aplicación gratis utilizada con fines educativos, privados y comerciales que facilita el desarrollo y simulación de programas en lenguaje PICAXE BASIC bajo ambiente Windows. Para los Sistemas Operativos GNU/Linux y MAC OS X, se debe utilizar el software AXEpad.

Las características más destacadas de este software, en palabras de sus propios desarrolladores, recuperado del portal oficial de PICAXE (2012), son:

- Auto identificación.
- Corrector de Sintaxis y programa para descargar el código.
- Simulación a pantalla completa con Chips animados.

- Varias herramientas de prueba para calibración.
- Se destacan utilizando colores distintos el código fuente y la sintaxis.
- Explorador que muestra valores de variables, constantes y etiquetas.
- Terminal serial y ventana para realizar debug.

### **II.6.2.-Processing**

Es un entorno de desarrollo integrado de código abierto (*Open Source*) y lenguaje de programación que se fundamenta en Java, por ello, puede heredar todas sus funcionalidades por lo que es una gran herramienta al momento de realizar proyectos complejos de envergadura.

El proyecto *Processing* fue iniciado por Ben Fry y Casey Reas con la intención de que se creara una herramienta alternativa al Software Propietario. En la actualidad es altamente utilizado en muchas ramas para el desarrollo de actividades y proyectos multimedia e interactivos; también sirve como medio para enseñar todo lo básico acerca de la programación enfocada al diseño digital.

El lenguaje usa una sintaxis más simplificada que como normalmente se consiguen en Java. Posee más de 1000 librerías que extienden el software no solo al aspecto gráfico, también lo expanden hacia el sonido, el audio, la computación, aplicaciones de redes, Bluetooth, NFC, entre otros. Una de sus grandes ventajas es que su descarga es gratuita y puede utilizarse en ambiente GNU/Linux, Windows y MAC OS X. Facilita el desarrollo de aplicaciones compatibles con el sistema operativo Android con tan solo cambiar el programa de modo entre sus distintas opciones. (Portal oficial de Processing, 2013).

### **II.6.3.- TeraTerm Web 3.1.**

Es un emulador de terminal (programa de comunicaciones) de Código Abierto (*Open Source*) y gratis. Este software permite establecer conexiones y comunicaciones de los siguientes tipos, según lo establecido en el portal de Cursomicros (2009-2013).

- Comunicaciones con el puerto serial.
- Conexiones TCP/IP.
- Comunicaciones IPv4 e IPv6.
- Emulación de VT100 y de VT200/300.
- Emulación de TEK4010.
- Protocolos de transferencia de Archivos.
- Codificación de Caracteres UTF-8.
- Scripts utilizando el Lenguaje Tera Term.
- Sets de caracteres japoneses, rusos, ingleses y coreanos.

Tera Term facilita la posibilidad de establecer la comunicación con un módem en estado esclavo (*Slave Mode*) vía Serial para realizar la configuración del mismo de acuerdo a las necesidades.

## **CAPÍTULO III**

### **METODOLOGÍA**

A continuación se presentan la totalidad de pasos que se siguen para la realización del proyecto. El capítulo se conforma por los siguientes puntos: Fases del Proyecto, junto con sus respectivas Actividades a cumplir para cada una de dichas fases, y el Tipo de Investigación.

#### **III.1.- Tipo de Investigación.**

Esta investigación corresponde a un proyecto del tipo factible, debido a que consiste en la elaboración y al desarrollo de una propuesta aplicable capaz de servir como un modelo viable para solucionar una situación específica en una organización, de acuerdo con sus necesidades y requerimientos. (Ortíz Lourdes, 2012).

### III.2.- Fases del proyecto.

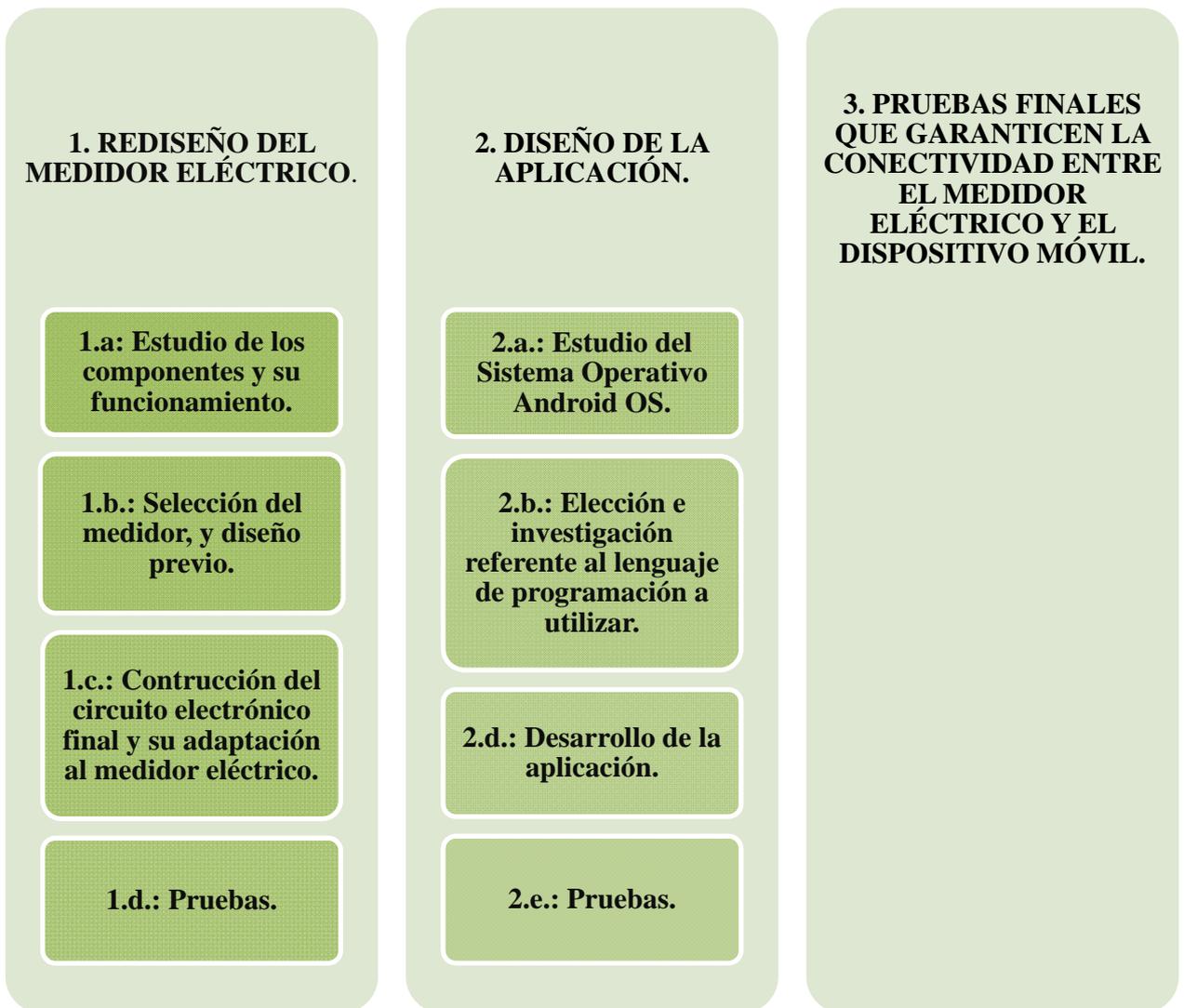


Figura 10. Fases del proyecto. Autoría Propia

### III.3.- Descripción de las actividades.

<b>Descripción de las actividades</b>	
<b>Fase 1</b>	
<b>Estudio de los componentes y su funcionamiento</b>	Para esta actividad, se realizó un estudio sobre la funcionalidad de los componentes electrónicos que serían útiles para la implementación del circuito externo aplicado al medidor de consumo eléctrico.
<b>Selección del medidor, y diseño previo.</b>	Para cumplir con esta fase se realizó una búsqueda exhaustiva en todos los sitios que disponen y proveen medidores de consumo eléctrico (en buen estado). Una vez encontrado el aparato, comenzó el estudio sobre el funcionamiento del mismo, la finalidad de todos los componentes electrónicos que posee y donde se podía anexar la tarjeta del circuito externo. Finalizada esta primera parte, se procedió a crear el borrador del diseño referente a la tarjeta externa, que se acoplará al presupuesto, al tamaño del medidor, y al cumplimiento de los objetivos del Trabajo de Grado
<b>Construcción del circuito electrónico final y su adaptación al medidor eléctrico.</b>	Una vez culminadas las dos actividades anteriores, se construyó e implementó el circuito externo, anexándolo finalmente al medidor de consumo eléctrico
<b>Pruebas</b>	Se procedió a realizar las pruebas en el laboratorio que garantizaron el pleno funcionamiento del aparato.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

<b>Fase 2</b>	
<b>Estudio del Sistema Operativo Android OS.</b>	Para el cumplimiento de esta actividad se realizó una investigación exhaustiva en documentación bibliográfica e internet sobre el sistema operativo Android.
<b>Elección e investigación referente al lenguaje de programación a utilizar.</b>	Una vez estudiado el Sistema Operativo Android en su totalidad, se procedió a escoger cual era el lenguaje y la plataforma de programación más factible para el desarrollo de este proyecto, que se ajustara a la versión del sistema operativo del dispositivo a utilizar, el más amigable y que fuese fácil de conseguir.
<b>Desarrollo de la aplicación.</b>	Una vez concretada las actividades anteriores, se desarrolló la aplicación en el sistema operativo mencionado anteriormente, que tiene como función mandar un correo a la empresa prestadora del servicio y al usuario con la factura respectiva.
<b>Pruebas.</b>	Se procedió a realizar las pruebas respectivas en el dispositivo que garantizaron el pleno funcionamiento de la aplicación.
<b>Fase 3</b>	
<b>Pruebas finales que garanticen la conectividad entre el medidor eléctrico y el dispositivo móvil.</b>	Ya finalizada las dos primeras fases del proyecto, se procedió a efectuar las pruebas definitivas que garantizaron el pleno funcionamiento entre el medidor de consumo eléctrico (transmisión de los datos almacenados) a cualquier otro dispositivo con el sistema operativo Android que tenga la aplicación instalada.

Tabla 1. Descripción de las Actividades. Autoría Propia.

## **CAPÍTULO IV**

### **DESARROLLO**

El desarrollo de este trabajo especial de grado, para el cumplimiento de los objetivos, se fundamentó en llevar a cabo el rediseño del medidor eléctrico conjuntamente con el diseño de un software basado en el sistema operativo Android. A continuación se describe lo realizado en las diferentes fases del proyecto.

#### **IV.1.- Rediseño del medidor eléctrico.**

El rediseño de este artefacto representó un valor importante en el desarrollo de este proyecto, debido que en esta fase se asenta la base del trabajo especial de grado. Dicho rediseño constó de una etapa analítica y una técnica.

##### **IV.1.1.- Estudio de los componentes y su funcionamiento.**

Para esta actividad, se llevó a cabo un estudio sobre la funcionalidad de los componentes electrónicos que serían útiles para la implementación del circuito externo aplicado al medidor de consumo eléctrico, dejando como resultado las opciones más viables e importantes para este proyecto que en este caso fueron el Microcontrolador PICAXE, suministrado por la Escuela de Ingeniería Informática de la Universidad Católica

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

Andrés Bello y el BlueSMiRF Silver Bluetooth Modem adquirido a través de la compra electrónica vía Internet, junto con sus componentes externos que perfeccionan el desempeño de los mismos.

#### IV.1.2.- Selección del medidor, y diseño previo.

Para cumplir con esta fase se realizó una búsqueda exhaustiva en todos los sitios que disponen y proveen medidores de consumo eléctrico (en buen estado), dicho artefacto se consiguió en la oficina comercial de la Corporación Eléctrica Nacional CORPOELEC ubicado en el municipio Chacao. Una vez encontrado el aparato, comenzó el estudio sobre el funcionamiento del mismo, la finalidad de todos los componentes electrónicos que posee y por último, donde se realizaría la interconexión entre la tarjeta del circuito externo y el Vatímetro.



Figura 11. Medidor de consumo utilizado (Vista Frontal). Autoría Propia

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---



Figura 12. Especificaciones del Medidor de Consumo Eléctrico. Autoría Propia.

Finalizada esta primera parte, se procedió a crear el borrador del diseño referente a la tarjeta externa, que se acoplara al presupuesto y al cumplimiento de los objetivos del Trabajo Especial de Grado. Cabe destacar que a medida que avanzaba la investigación se agregarían más elementos al circuito

#### **IV.1.3.- Construcción del circuito electrónico final y su adaptación al medidor eléctrico.**

A continuación se presenta el procedimiento utilizado para llevar a cabo el circuito electrónico final para posteriormente adaptarlo al medidor de consumo eléctrico.

IV.1.3.1.- Cálculos teóricos y prácticos de la señal emitida por el medidor de consumo eléctrico.

Se observó en el osciloscopio la señal emitida por el medidor eléctrico, la conexión para esta medición se puede ver en la Figura 13, esta conexión se realizó de esta manera ya que el LED indica cuando se consume 1/1600 [kw/h] emitiendo un pulso.



Figura 13. Conexión de la puntas del osciloscopio para observar la señal emitida. Autoría Propia

Ante las características del medidor de consumo eléctrico (véase Figura 12) se conoció que cada mil seiscientos (1600) pulsos eléctricos corresponden al consumo de 1

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

[kw/h] permitiendo establecer una relación, donde se puede calcular de manera teórica el consumo para un bombillo de setenta y cinco (75) vatios de potencia, consumidos en una hora. La ecuación se muestra a continuación:

$$\text{pulsos} = \text{consumo} \left[ \frac{\text{kw}}{\text{h}} \right] * 1600 \left[ \frac{\text{Pulso}}{\frac{\text{Kw}}{\text{h}}} \right]$$

Para comprobar el correcto funcionamiento del medidor eléctrico se realizaron los siguientes cálculos teóricos y prácticos con un bombillo de (75) vatios de potencia consumidos en una hora:

- Cálculos teóricos:

$$\text{pulsos} = 0,075 \left[ \frac{\text{kw}}{\text{h}} \right] * 1600 \left[ \frac{\text{imp}}{\frac{\text{kw}}{\text{h}}} \right]$$

$$\text{pulsos} = 120 \text{ imp}$$

- Cálculos prácticos:

Estos cálculos se realizaron observando el comportamiento del Led indicador del medidor al conectársele un bombillo de setenta y cinco (75) vatios de potencia consumidos en una hora, el estudio se hizo durante un periodo de tiempo de 5 minutos para luego extrapolarlos a los 60 minutos.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

El resultado arrojado fue de 12 pulsos transcurridos 5 minutos. Luego se realizó la siguiente multiplicación para llevarlos a una hora

$$\text{pulsos} = 12 * 2 * 6$$

$$\text{pulsos} = 144 \text{ imp}$$

#### IV.1.3.2.- Materiales.

A continuación se señalan todos los materiales e instrumentos que fueron utilizados para el rediseño del medidor de consumo eléctrico.

- Medidor electrónico monofásico HOLLEY.
- Microprocesador AXE001 PICAXE 28X1 Starter Pack.
- (3) Baterías AA 1,5 V.
- BlueSMiRF Silver RN-42.
- Cable Serial RS-232.
- Amplificador operacional.
- Resistencias de  $1K\Omega$  y  $3K\Omega$ .
- Diodo LED.
- Cables de cobre.

#### IV.1.3.3.- Programación del micro controlador PICAXE.

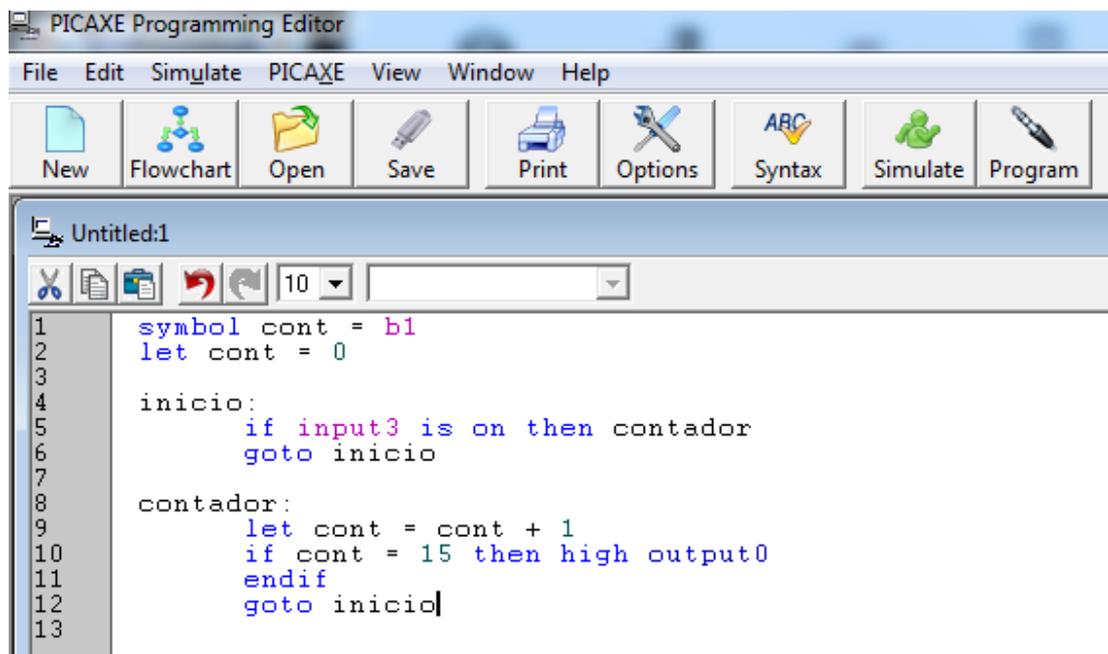
Se utilizó el lenguaje de programación PICAXE BASIC. El primer programa empleado en esta sub-fase se trata de un código de prueba, el cual opera con un contador

## Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

que almacena en una variable denominada “b1” un incremento en una unidad, una vez recibido un uno (1) lógico en el puerto de entrada digital número tres (3).

El código mostrado en la Figura 14, se le agregó una validación que comprueba que este funcione correctamente. Este segmento del programa enciende un LED que está conectado de forma inversa (el pin negativo del LED conectado a 5V) debido que, al trabajar con un Protoboard, la documentación del PICAXE indica que la conexión debe realizarse de esta forma. Está ubicado en el puerto de salida número cero (0), una vez que la variable “cont” llegue al valor de quince (15).



```
1  symbol cont = b1
2  let cont = 0
3
4  inicio:
5      if input3 is on then contador
6      goto inicio
7
8  contador:
9      let cont = cont + 1
10     if cont = 15 then high output0
11     endif
12     goto inicio
13
```

Figura 14. Captura de pantalla del código de prueba. Autoría Propia

Una vez comprobado el pleno funcionamiento del microcontrolador se procedió a crear el código que envía a través del puerto de comunicación serial la variable “cont”, que será recibida por el modem Bluetooth, dicho código se puede observar en el Apéndice C.

#### IV.1.3.3.1 Funcionamiento del Código.

- Se declararon dos variables: “b0” y “b1”. “b0” se asignó como “cont” la cual lleva el valor del contador, y “b1” se asignó como “cond” teniendo como función ser un condicional. Ambas variables fueron inicializadas con el valor cero (0).
- Se utilizó el comando “setfreq m8” con la intención de aumentar la frecuencia del microprocesador de 4 MHz (por defecto) a 8Mhz. Esto genera un aumento de la velocidad de transmisión de 4800 bps a 9600 bps, de acuerdo con la documentación del microprocesador.
- Dentro de la rutina declarada como “inicio”, se comprueba que el puerto de entrada digital (3) haya recibido un (1) lógico proveniente del medidor de consumo eléctrico, pudiendo ocurrir dos panoramas: al recibir una variación de voltaje, se evalúa la variable ”cond”. Si la variable “cond” es igual a cero (0) entonces, se asigna a esta variable el valor uno (1) y el programa realiza un salto a la etiqueta “contador”, de lo contrario si la variable “cond” tiene el valor de uno (1) salta a la etiqueta “inicio”, el motivo de utilizar esta variable condicional es para evitar que ocurra un rebote de voltaje, que podría aumentar la variable “cont” más veces de lo normal, lo que podría generar errores en el funcionamiento del proyecto. Para el segundo panorama si no

se recibe ninguna cambio de voltaje se asigna el valor (0) a la variable “cond” y el programa se devuelve al inicio.

- La etiqueta “contador” se encarga de aumentar en una unidad la variable “cont” cada vez que el programa entre a esta rutina, al finalizar el programa vuelve al inicio.
- El primer comando dentro de la rutina “inicio” es: “sertxd (b0)”, esta instrucción es de vital importancia, ya que es la encargada de enviar los datos a través del puerto serial de la tarjeta PICAXE; se colocó en esta posición dentro del código con la finalidad de que la rutina siempre se ejecute, de manera tal que los datos se envíen constantemente.

La representación gráfica del código se muestra en la Figura 15.

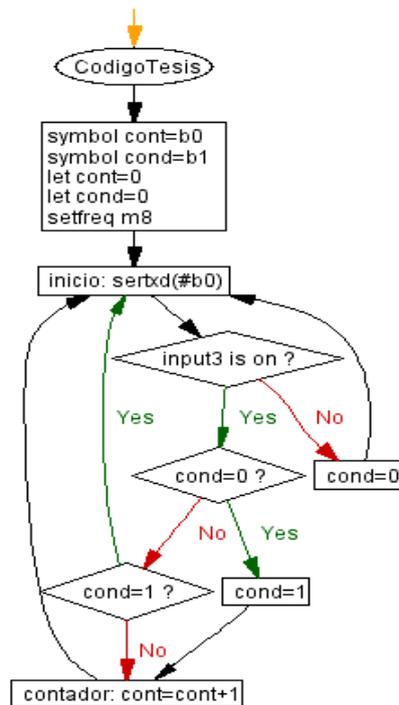


Figura 15. Diagrama de flujo del funcionamiento del código. Autoría Propia.

#### IV.1.3.4.- Diseño del circuito.

El medidor eléctrico genera impulsos que se reflejan en el LED de parpadeo, indicando que se ha consumido:

$$consumo = \frac{1}{1600} \left[ \frac{imp}{\frac{Kw}{h}} \right] * Carga \left[ \frac{Kw}{h} \right]$$

Este LED está conectado al amplificador ya que el voltaje de la señal tomada (1,2 V) es muy bajo para el procesamiento en el microprocesador. La ganancia del amplificador no inversor es  $A_v=4$  (ver Figura 17), obteniendo una señal de salida de aproximadamente 4,8 V. Este valor si puede ser analizado por el PICAXE de acuerdo a su documentación.

La salida del amplificador se conecta directamente al puerto de entrada digital (3) del PICAXE, el cual contiene el código que se encargará de almacenar, procesar y enviar por el puerto de transmisión serial (Tx) la variable respectiva.

El pin Tx del PICAXE se conecta al pin de recepción (Rx) del modem Bluetooth RN-42, debido que de acuerdo con las características técnicas del establecimiento de una comunicación serial, para conectar un dispositivo Maestro (*Master*) a un dispositivo Esclavo (*Slave*) es necesario que la conexión se realice de forma cruzada. El modem RN-42 será el encargado de enviar la data al dispositivo móvil que contiene la aplicación.

El bosquejo del diseño del circuito externo se puede observar en la Figura 16.

Figura 16. Bosquejo del circuito a implementar. Autoría Propia.

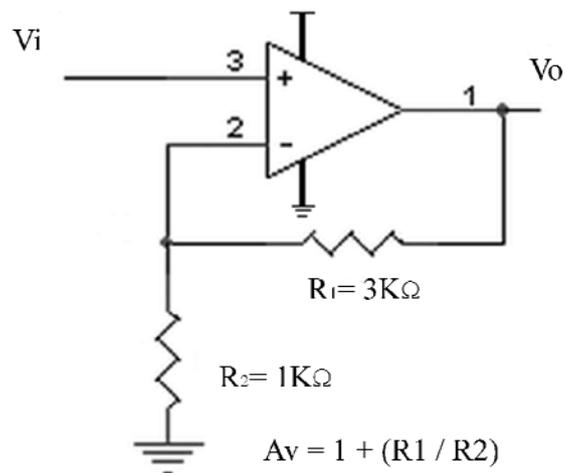


Figura 17. Amplificador no Inversor de Ganancia 4. Autoría Propia.

#### IV.1.3.5.- Configuración del BlueSMiRF Silver Bluetooth Modem

Se realizó la configuración del Modem BlueSMiRF Silver Bluetooth RN-42 con la intención de modificar algunas de sus características de fábrica para que sea posible la comunicación con el PICAXE 28X1. Para ello fue necesario el uso de una Laptop que tenga instalado un adaptador Bluetooth capaz de reconocer, vincular y conectar con otros dispositivos que utilicen el mismo estándar, además se requirió que esta tuviese instalado un software emulador de terminal, como por ejemplo: HyperTerminal. En este caso el usado fue TeraTerm Web 3.1.

El procedimiento implementado fue el siguiente: como primer paso se buscó, reconoció y pareó el modem RN-42 con el ordenador. Posteriormente se inició el software TeraTerm Web 3.1. y se configuró para establecer un socket o puerto de comunicación entre ambos dispositivos (COM3 por defecto de la Laptop, Ver Figura 19). Se procedió a conectar los dos dispositivos; una vez establecida la conexión se escribió el comando “\$\$\$” en el software; esta instrucción coloca al RN-42 en Modo Comando (*Command Mode*, Ver Figura 18). En esta modalidad es cuando se pueden realizar cambios a las características del modem Bluetooth. El valor que se modificó fue la velocidad de transmisión que por defecto equivale a 115,2 Kbps, a 9600 bps; se usó el comando “SU,96” para tal objetivo. Finalmente para salir del Modo Comando, escribe “---“ (Ver Figura 20).

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---



Figura 18. Modem Bluetooth RN-42 en "Modo Comando". Autoría Propia

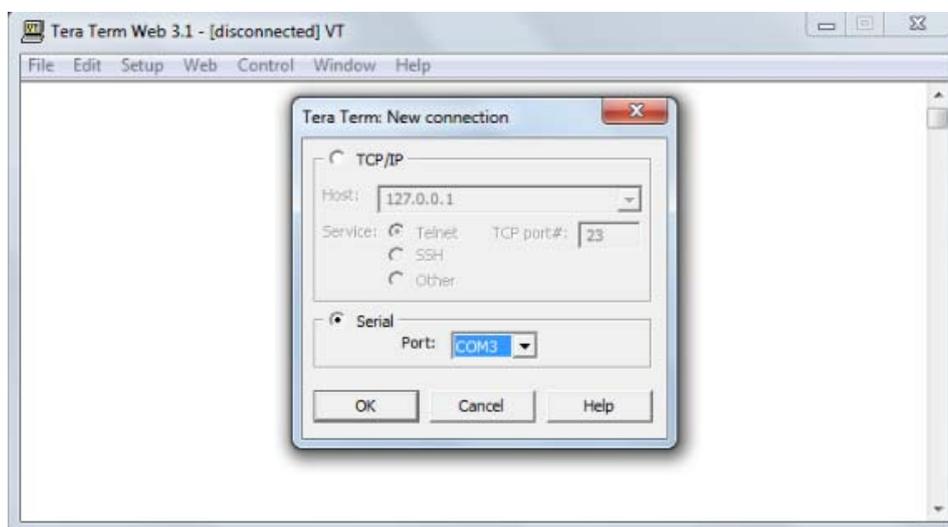
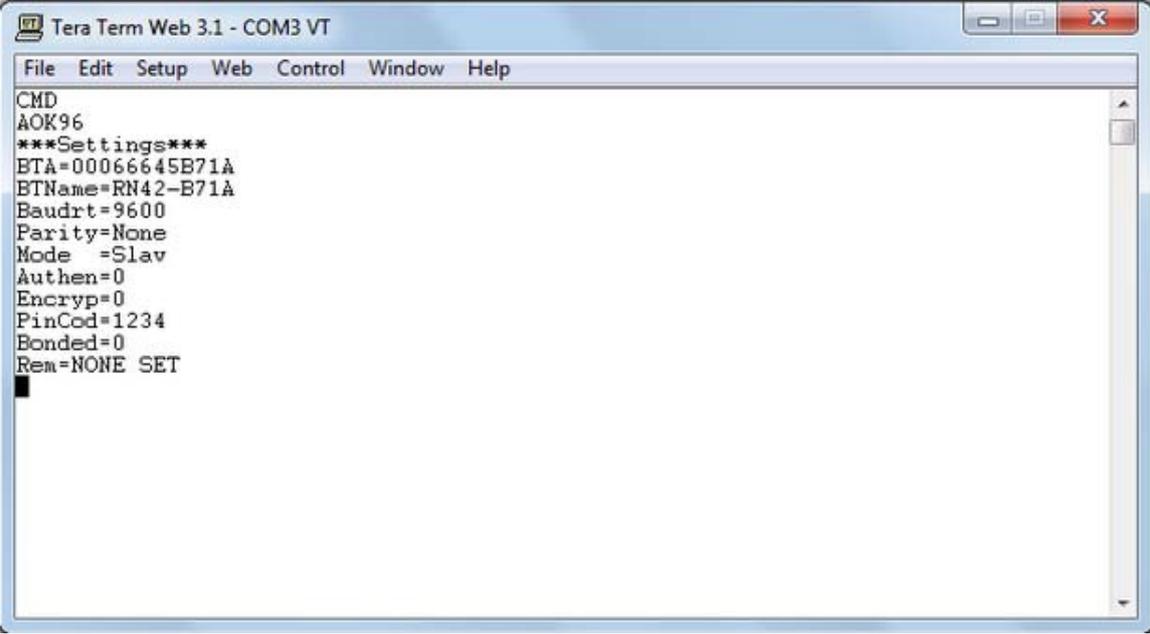


Figura 19. Configuración del TeraTerm Web 3.1. Autoría Propia.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---



```

Tera Term Web 3.1 - COM3 VT
File Edit Setup Web Control Window Help
CMD
AOK96
***Settings***
BTAddress=00066645B71A
BTName=RN42-B71A
Baudrate=9600
Parity=None
Mode =Slav
Authen=0
Encryp=0
PinCod=1234
Bonded=0
Rem=NONE SET

```

Figura 20. Configuración del Modem Bluetooth RN-42. Autoría Propia.

#### IV.1.4.- Pruebas.

En esta fase se realizaron las pruebas en el laboratorio de la Escuela de Ingeniería Informática de la Universidad Católica Andrés Bello, estas se efectuaron conectando todos los componentes y dispositivos mencionados anteriormente con su respectiva configuración y probados en etapas diferentes. Las pruebas que se realizaron fueron las siguientes:

- Se verificó la señal de salida del amplificador no inversor.
- Se conectó la salida del amplificador a la tarjeta del microcontrolador PICAXE, y se verificó el programa de prueba mencionado en la fase “Construcción del circuito electrónico final y su adaptación al medidor eléctrico”.
- Se probó que fuese posible la vinculación de varios dispositivos con el modem Bluetooth RN-42, los dispositivos utilizados fueron: Samsung SIII GT-I9300, BlackBerry Torch 9800 y BlackBerry Gemini 8520.

## **IV.2.- Diseño de la aplicación.**

### **IV.2.1.- Estudio del Sistema Operativo Android OS.**

Para esta fase del proyecto se realizó una investigación teórica con relación al Sistema Operativo sobre el cual funciona el dispositivo que fue utilizado para el desarrollo de la aplicación. El móvil escogido para tal cometido fue el Samsung SIII GT-I9300 el cual cuenta con la versión del Sistema Operativo 4.1.2 (*Jelly bean*), además posee adaptador Bluetooth. La información sobre este tópico se encuentra especificada en el Capítulo II.

### **IV.2.2.- Elección e Investigación referente al lenguaje de programación a utilizar.**

El lenguaje de programación sobre el cual se fundamenta el desarrollo de aplicaciones para el Sistema Operativo Android es Java. Por ello la elección fue realizar una búsqueda en distintos portales en Internet que facilitaron la descarga de varios Entornos de Desarrollo Integrado (IDE) que trabajan con este lenguaje. Los tres softwares escogidos y descargados en una primera instancia fueron: IDE Eclipse, IDE IntelliJ IDEA y Processing.

Luego de poner en funcionamiento algunos códigos de prueba para la familiarización con los diversos IDE, se eligió utilizar el software Processing. Se descartó el IDE Eclipse debido a que muchas veces ocurrió que, pese a estar correcto un código, se

generaban errores y se cerraba el software, así como también ocurrió que el rendimiento del computador se hacia lento cuando se simulaba algún programa. La decisión de prescindir del IDE IntelliJ IDEA, se debió a que es un software creado para usuarios profesionales con un alto nivel de dominio en conocimientos, lo que lo hace poco amigable.

Se seleccionó Processing ya que fue el que ofreció mayor diversidad de opciones al momento de programar, además que sus tutoriales son de gran ayuda y se explican de forma sencilla tal que cualquier persona con un conocimiento regular en materia de programación Java, puede desarrollar aplicaciones con fines profesionales o educativos.

#### **IV.2.3.- Desarrollo de la aplicación.**

Como parte del procedimiento para iniciar la programación bajo ambiente Android se requiere en primera instancia descargar el JDK de Android (Kit de desarrollo Java o *Java Development Kit*) el cual es un software que contiene un conjunto de herramientas de desarrollo para la creación de programas en Android con lenguaje Java y que permitirá al IDE Processing tener las librerías necesarias para comenzar el desarrollo de la aplicación. También es necesario descargar la API (Interfaz de Programación de Aplicación ó *Application Programming Interface*) correspondiente a la versión del sistema operativo del dispositivo al que se le desee instalar la aplicación. Para este proyecto se descargó el API de la versión 4.2.

El IDE Processing presenta varios modos de operación, entre los que se destacan Java, JavaScript, Experimental y Android. Se seleccionó el modo Android (Ver Figura 21).

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

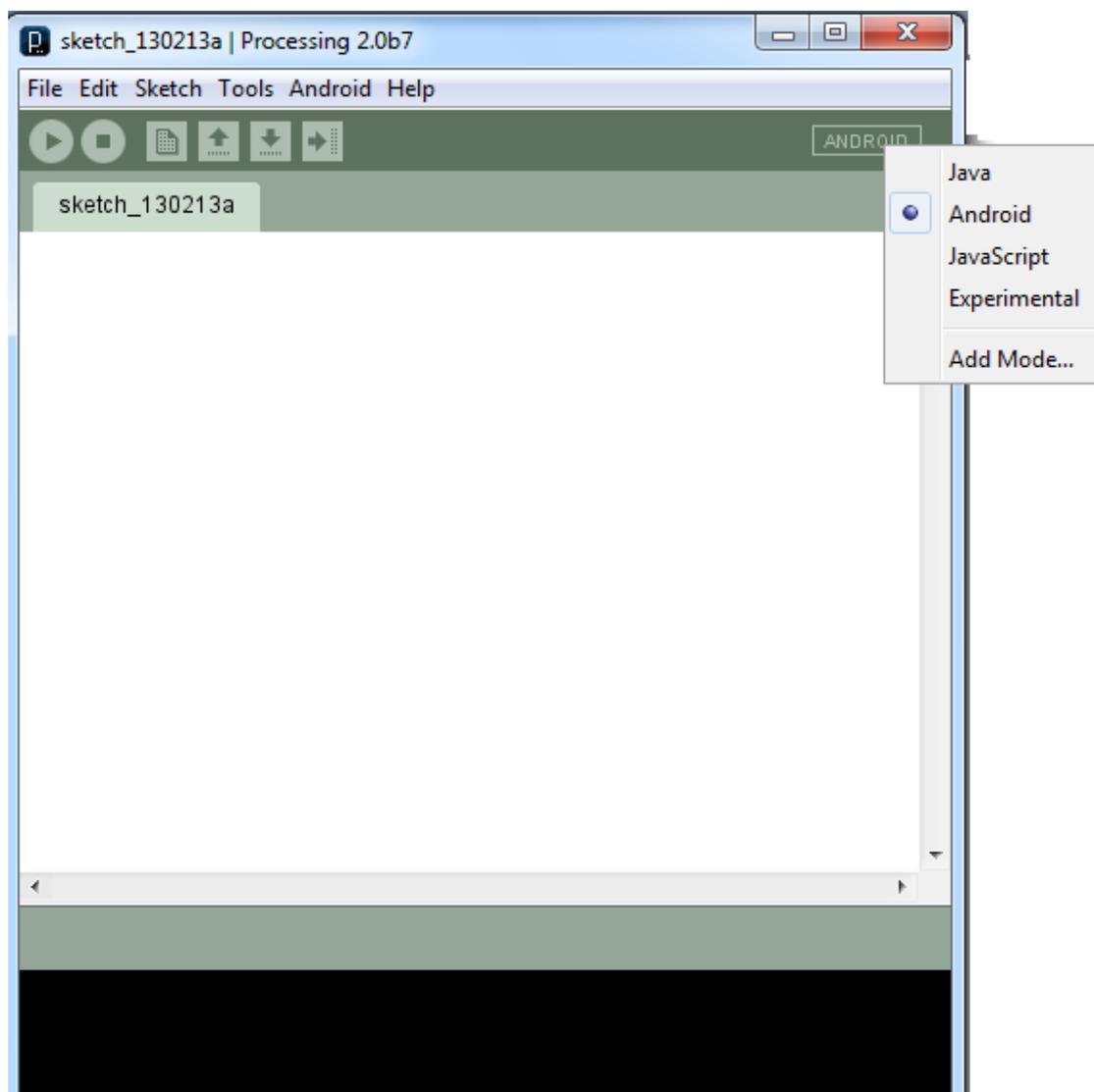


Figura 21. Interfaz del IDE Processing. Modo Android. Autoría Propia

A continuación se explicará cada método en el orden en el que aparece dentro del código empleado para la realización de la aplicación. Sin embargo el código fuente se encuentra en el Apéndice B.

- **setup():** Es un método obligatorio en todos los códigos programados en Processing que requieran de una interfaz gráfica, en esta sección se asigna el tamaño de la pantalla y se escogen los tipos de fuente y sus características que aparecerán en la misma.
- **draw():** Dependiendo del valor de la variable “estado” realiza una acción distinta o ejecuta un método particular, los cuales son: buscar dispositivos, mostrar dispositivos disponibles, ejecutar el método conectaDispositivo(), muestraDatos() o muestraError().
- **onStart():** Verifica que el móvil posea un adaptador Bluetooth. De ser así muestra un mensaje con la opción de encender o no el Bluetooth en el dispositivo y ejecuta el método empieza() de caso contrario muestra en pantalla un error.
- **onStop():** Cierra el Socket si se pudo establecer una comunicación previa, de caso contrario se muestra un error en pantalla.
- **onActivityResult():** Encargado de mostrar el error en pantalla si el usuario decidió NO encender el Bluetooth al aparecer la caja de diálogo.
- **mouseReleased():** Encargado de cambiar el valor de la variable “estado” dependiendo de las acciones posibles en la pantalla táctil del dispositivo.

- **empieza():** Se encarga de mostrar la lista de los dispositivos vinculados al celular.
- **listaDispositivos():** Llena y almacena la variable “dispositivos” con los dispositivos que estén disponibles.
- **compruebaEleccion():** Verifica el dispositivo seleccionado por el usuario.
- **conectaDispositivo():** Está encargado de conectar el celular que contiene la aplicación con el dispositivo escogido por el usuario, además recibe el dato que se transmite vía Bluetooth y lo almacena en la variable “ins”. Dado el caso que no se pueda establecer la comunicación se muestra un error en la pantalla.
- **muestraDatos():** Muestra la factura en pantalla con los datos recibidos y asimismo un botón que permite el envío de dicha factura por correo electrónico. Este método realiza la conversión a la unidad de consumo y de pago. La unidad de pago se calculó mediante la multiplicación del consumo por un coeficiente. El coeficiente real es un valor aproximado a 0,124 BsF/Kwh, para las pruebas se utilizó un coeficiente ficticio igual a dos (2).
- **enviaCorreo():** Se encarga de mostrar al usuario un mensaje de diálogo con los clientes de correo electrónico existentes en su dispositivo, para que seleccione el de su preferencia. También llena el asunto, destinatario y contenido del correo electrónico de forma automática. En caso de no tener instalado ningún cliente de correo electrónico muestra un error por pantalla.

- **compruebaBoton():** Verifica que el botón “Enviar E-Mail” fue presionado y ejecuta el método `enviaCorreo()`.
- **muestraError():** Encargado de mostrar el mensaje de error en pantalla en dado caso que se generara en alguno de los métodos anteriormente explicados.

#### IV.2.4.- Pruebas.

La manera en como se realizaron las pruebas para esta fase del proyecto fueron las siguientes:

- Se compiló y ejecutó la aplicación en un dispositivo virtual.
- Se ejecutó la aplicación en el dispositivo móvil, para verificar que no se presentaran errores espontáneos.
- Se revisó que se pudiera establecer la conexión con el modem Bluetooth, al hacer una pequeña modificación en el código, de tal forma que si se lograban conectar, en el dispositivo móvil se mostró un mensaje en pantalla que decía “Conexión Exitosa” (Ver Figura 22).

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

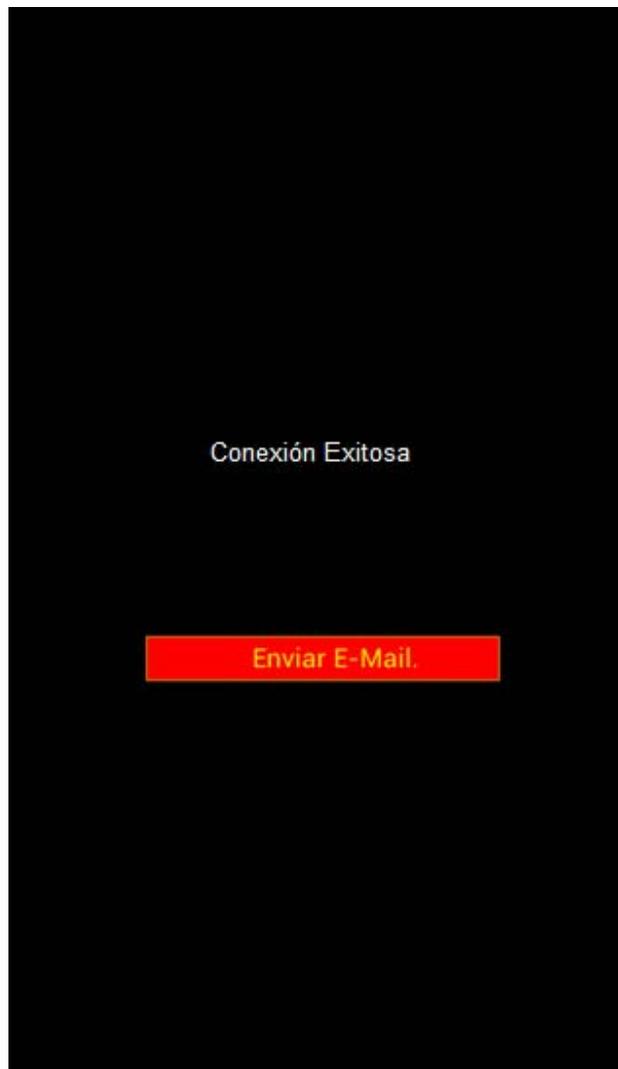


Figura 22. Prueba de conexión entre dispositivos. Autoría Propia.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

### **IV.3.- Pruebas Finales.**

Luego de ejecutadas las fases previas se procedió finalmente, a conectar todos los dispositivos (Ver figura 16): medidor de consumo eléctrico, amplificador no inversor, microcontrolador PICAXE 28X1, modem Bluetooth RN-42 con el móvil Samsung Galaxy SIII, de manera tal de que este pudiese recibir vía Bluetooth la variable “cont” enviada desde el microprocesador. La conexión se puede observar en la Figura 23.

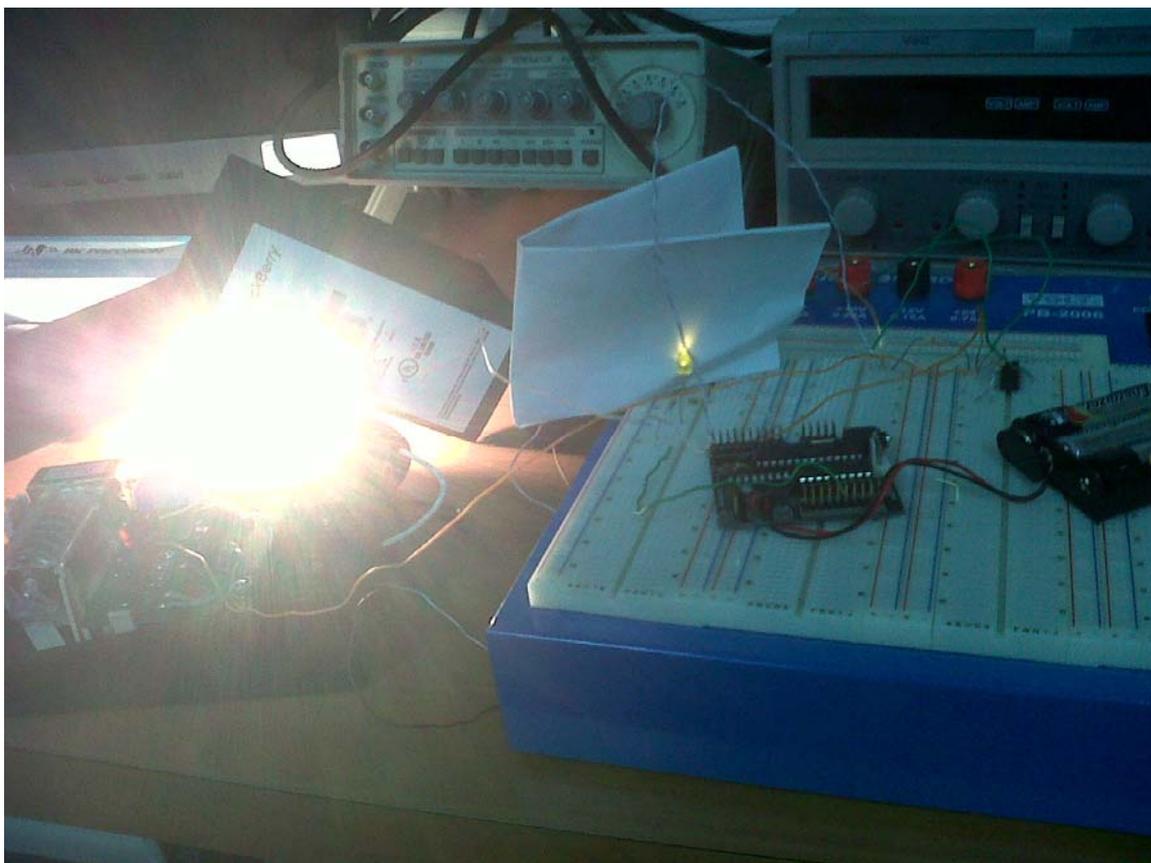


Figura 23. Conexión final del proyecto. Autoría Propia

## **CAPÍTULO V.**

### **ANÁLISIS DE LOS RESULTADOS**

En este capítulo se muestran los resultados obtenidos luego de haber desarrollado cada uno de las fases y procesos definidos y explicados en capítulos anteriores.

#### **V.1.- Medidor de consumo eléctrico.**

Una vez realizados los cálculos teóricos, estos arrojaron un valor de 120 impulsos al conectar un bombillo de 75 vatios por una hora al medidor de consumo eléctrico. Comparado con los cálculos prácticos que dieron el valor de 144 impulsos bajo las mismas condiciones anteriores, surge un error de 16,67%. En la empresa facilitadora del medidor, se realizaron pruebas con equipos especiales, que miden el patrón de calibración del aparato. Aunque no se recibieron los datos reales del resultado de la prueba, los técnicos encargados explicaron, que el valor de la calibración no está dentro del rango permitido para utilizarlos en residencias.

Ciertamente, luego de observar la señal de salida del amplificador no inversor se evidenció el aumento del valor de la amplitud de la señal original del LED de parpadeo del medidor de consumo de 1,2V a 4,8V aproximadamente.

La variable contadora utilizada dentro del código del microcontrolador efectivamente está almacenando e incrementa su valor según las instrucciones que recibe del programa, esto se pudo corroborar con la simulación mostrada en la Figura 24 obtenida

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

del simulador que tiene incorporado el software de programación PICAXE Programming Editor, así como también se comprobó esto con el montaje explicado en el punto “IV.1.3.3 Programación del microcontrolador PICAXE.”

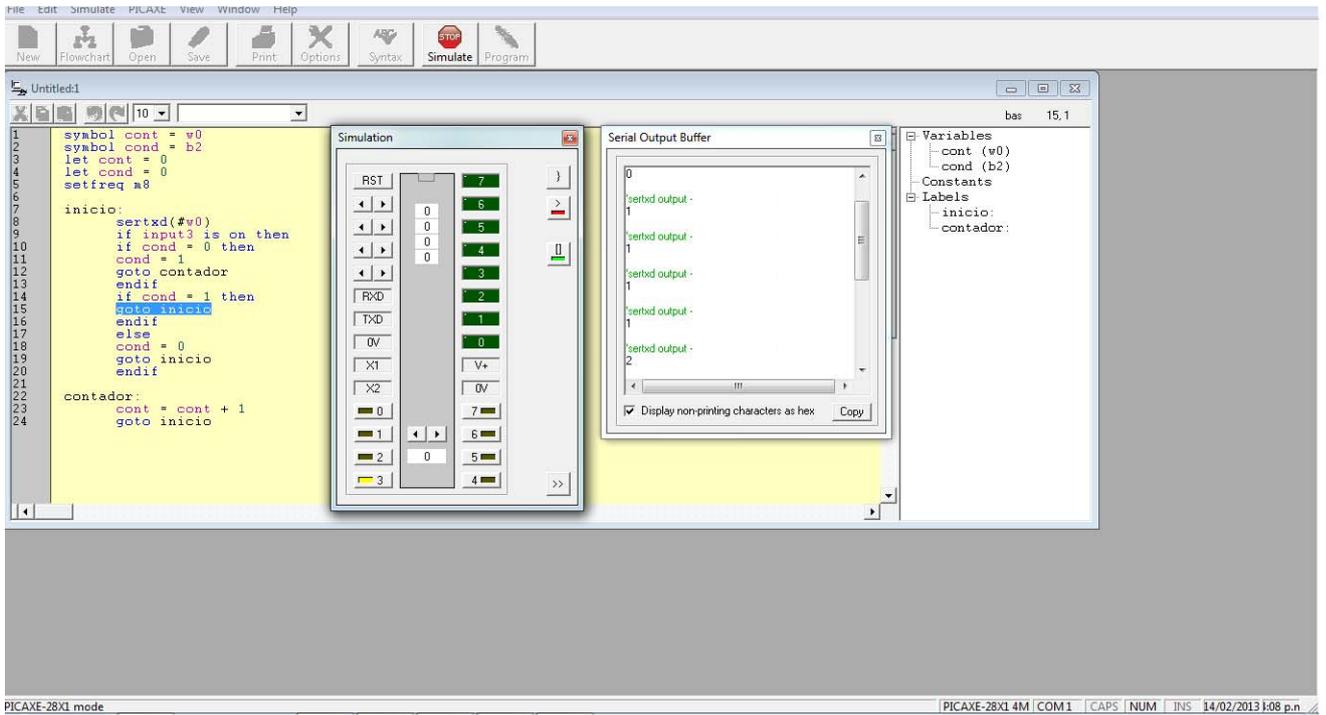


Figura 24. Captura pantalla de la simulación de prueba realizada. Autoría Propia

## **V.2.- Aplicación.**

Una vez desarrollada la aplicación, se comprobó su correcto funcionamiento vinculando el dispositivo móvil con el modem Bluetooth para establecer posteriormente la conexión entre ambos equipos. Se destaca que la conexión se realizó únicamente con el modem separado de los demás componentes que integran el circuito rediseñado del medidor de consumo, teniendo como resultado el exitoso establecimiento de la conexión al poder acceder a la pantalla de la factura. La factura mostrada no presentó ningún valor en sus campos de consumo y pago debido a que la prueba fue netamente de conectividad, es decir, no se envió ningún dato. Las Figuras 25, 26 y 27 evidencian lo mencionado anteriormente.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

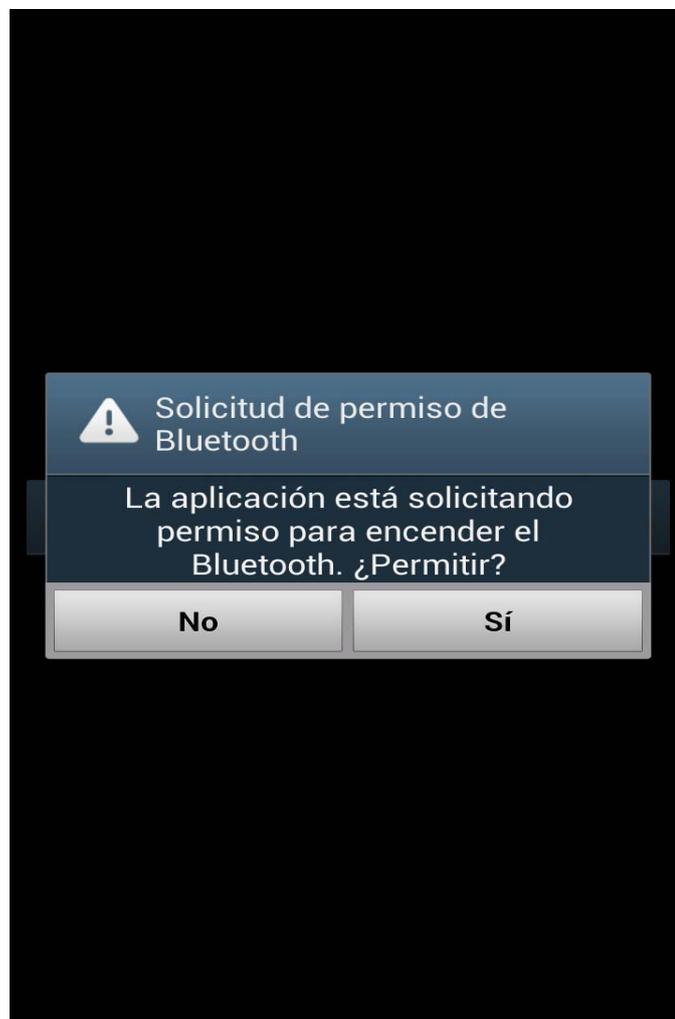


Figura 25. Captura pantalla del mensaje de solicitud de permiso Bluetooth. Autoría Propia.

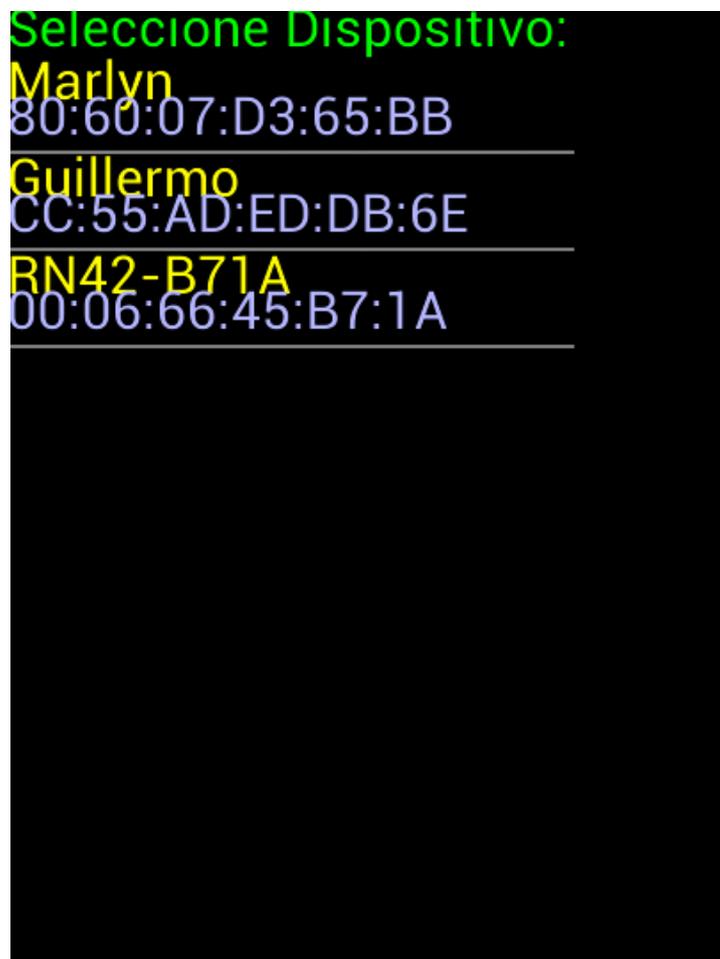


Figura 26. Captura pantalla de la listas de dispositivos vinculados. Autoría Propia.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---



Figura 27. Captura pantalla de la factura de prueba. Autoria Propia

Luego de efectuar las pruebas previas, se añadió el código necesario para enviar la factura con la información a los correos electrónicos respectivos del usuario y de la empresa reguladora del servicio. Como comentario adicional, se aclara que las direcciones

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

de correos electrónicos empleados para esta etapa fueron correos de prueba que representan al cliente y a la compañía. El resultado fue el envío exitoso de la factura a ambas direcciones; las Figuras 28 y 29 muestran el resultado antes explicado.

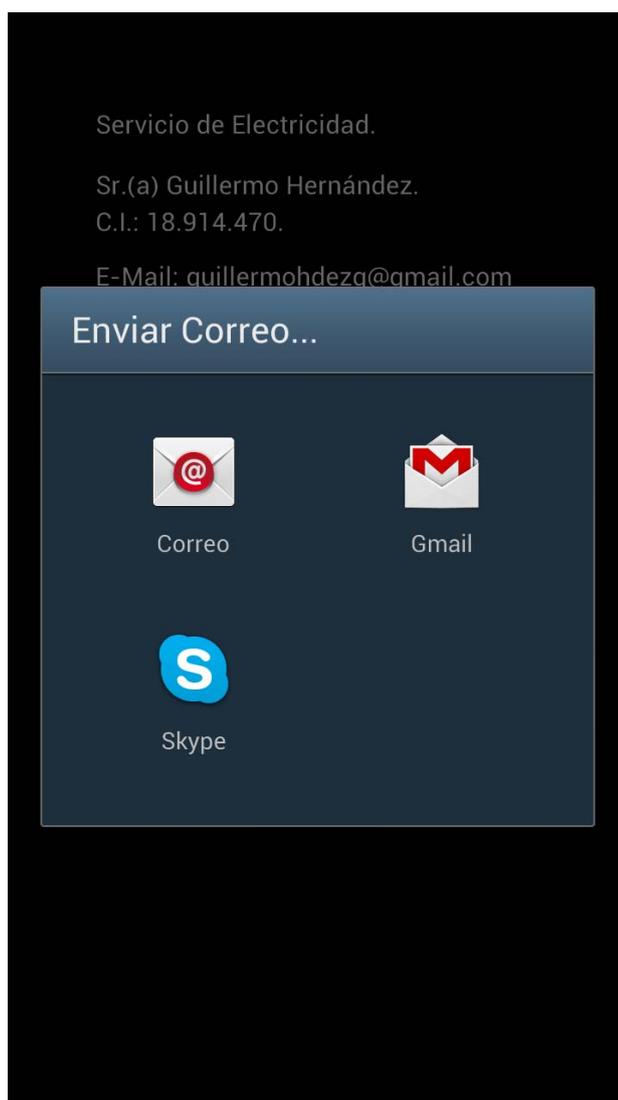


Figura 28. Captura pantalla de las aplicaciones disponibles para el envío del correo electrónico. Autoría Propia

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---



Figura 29. Captura pantalla del correo que se dispone a enviar a las direcciones guardadas. Autoría Propia.

### **V.3.- Pruebas Finales.**

Una vez conectado todo el sistema, la aplicación arrojó un resultado no esperado. A partir de las cifras de dos (2) dígitos el valor que se recibe en la aplicación no concuerda con el valor esperado. Debido a esto, se procedió a probar nuevamente el funcionamiento de cada segmento del sistema, lo que arrojó como explicación que el error se encuentra en la conexión existente entre el PICAXE 28X1 y el modem RN-42. Esto ocurre por la siguiente razón: Los datos en la comunicación serial se envían bit a bit, además se le agrega al paquete bits de paridad y bit de stop. A pesar de que para ambos dispositivos las características son las mismas, al sumarse un dígito más a la variable a transmitir se pierde la sincronización en la comunicación.

Sin embargo como resultado final de estas pruebas se logró establecer la comunicación entre el dispositivo móvil Samsung SIII con el medidor de consumo eléctrico a través del estándar Bluetooth.

Para los diez valores que se transmitieron de forma correcta se logró mostrar en la pantalla del móvil que contiene la aplicación, la factura correspondiente al consumo del mes así como también el envío exitoso del correo electrónico al usuario y a la compañía proveedora del servicio. El motivo de haber utilizado el valor del coeficiente de pago igual a 2BsF/Kwh se debe a que, por ser valores tan pequeños los recibidos por la aplicación, el valor del campo de pago sería sumamente pequeño generando que quizás no se podría apreciar ninguna cantidad en dicho campo de la factura.

## CONCLUSIONES Y RECOMENDACIONES

Una vez estudiado a fondo y analizado el sistema de comunicación implementado para el desarrollo del Trabajo Especial de Grado, se pueden plantear distintas conclusiones.

Primeramente se puede comentar acerca de la importancia de seleccionar el medidor de consumo apropiado para la implementación de este sistema digital. Es necesario que se escoja un medidor de consumo eléctrico electrónico y no un medidor electromecánico, que es uno de los más usados en las residencias domiciliarias. Esto no quiere decir que sea imposible realizar la optimización del proceso de facturación con este tipo de medidor, sin embargo se requiere de un rediseño distintito en la estructura interna del mismo.

A pesar de que se logró cumplir el objetivo principal, se presentaron dificultades con el microprocesador al momento de enviar los datos del PICAXE 28X1 al modem Bluetooth RN-42. Se considera que parte de las fallas técnicas en esta etapa del proyecto se debe a la poca compatibilidad que existe entre la tarjeta del microprocesador y el modem. En una gran cantidad de la bibliografía consultada se notó que el PICAXE es empleado en su mayoría para proyectos relacionados con robótica y activación de motores de paso. También existen otros dispositivos que para el desarrollo de este sistema hubiesen sido mucho más óptimos. Se recomienda que para proyectos que involucren el estándar Bluetooth se utilice como tarjeta proyecto del microprocesador la tarjeta Arduino.

Por otro lado una característica limitante que posee el microcontrolador PICAXE, es que sus variables solo pueden alojar valores en el rango de 0-65535; si consideramos que el medidor eléctrico utilizado contabiliza un (1) Kwh por cada mil seiscientos (1600) impulsos y además, el consumo promedio de una residencia es de aproximadamente 450 Kwh mensual, la variable debería almacenar un valor superior a 720.000. Debido a esta

## Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

premisa se recomienda la utilización de un microprocesador diferente que cumpla con estos requerimientos.

Los objetivos referentes al desarrollo y funcionamiento de la aplicación fueron cumplidos en su totalidad.

Para finalizar, se piensa que este proyecto no es apto para implementar su funcionamiento en un ambiente residencial hoy por hoy, puesto que existen una cierta cantidad de modificaciones que se pueden realizar para complementar el sistema actual de manera que, en un futuro, pueda ser inclusive un proyecto viable por parte de la compañía prestadora del servicio eléctrico.

Algunas de las modificaciones recomendadas son: cambiar el microcontrolador utilizado así como su tarjeta de proyecto Arduino, agregar a la aplicación un método que llame a la base de datos de la compañía alojada en un servidor, para extraer de ella los datos del usuario asignado al número de medidor contratado, incorporar a la aplicación un botón que tenga como función reiniciar la variable contadora ubicada en el microcontrolador, modificar las dimensiones del medidor eléctrico con la finalidad de que se pueda incluir la nueva circuitería dentro de un único dispositivo, y por último agregar una clave de vinculación al dispositivo Bluetooth más segura que la configurada por defecto.

## **BIBLIOGRAFÍA**

Android (2012), Developers, [Página Web]. Recuperado el 15 de junio de 2012, de [developer.android.com/sdk/index.html](http://developer.android.com/sdk/index.html)

Android (2012), Discover Android, [Página Web]. Recuperado el 15 de junio de 2012, de [www.android.com/about/](http://www.android.com/about/)

Ayera Technologies, INC. (2013) TeraTerm Pro Web 3.1.3 - Enhanced Telnet/SSH2 Client, [Página Web]. Recuperado el 02 de febrero de 2013, de <http://www.ayera.com/teraterm/>

Bluetooth SIG Inc. (2012), Bluetooth Basics, [Página Web]. Recuperado el 29 de abril de 2012, de <http://www.bluetooth.com/pages/basics.aspx>

Bolton, M. (2011). What is Android? A Beginner's Guide. [Página Web]. Recuperado el 15 de junio de 2012, de <http://www.techradar.com/news/phone-and-communications/mobile-phones/what-is-android-a-beginners-guide-975482>

Cursomicros.com (2013). Uso del Programa Tera Term. [Página Web]. Recuperado el 02 de febrero de 2013, de <http://www.cursomicros.com/avr/usart/tera-term.html>

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

Curt, F. & Layton, J. (2011), How Bluetooth Works? [Página Web]. Recuperado el 29 de abril de 2012, de [http:// electronics.howstuffworks.com/bluetooth.htm](http://electronics.howstuffworks.com/bluetooth.htm)

De Lucas, J. (1991). Introducción al Basic. [Página Web]. Recuperado el 14 de enero de 2013, de <http://platea.pntic.mec.es/~jdelucas/basic.htm>

Engineers Garage. (s.f.), Android. [Página Web]. Recuperado el 15 de junio de 2012, de <http://www.engineersgarage.com/articles/what-is-android-introduction>

Gama, L.A, Villa, E. & Nogueroń, C. (s.f.), Bluetooth: un enfoque al estándar IEEE 802.15.1 [Documento PDF], México. Instituto Tecnológico de Zacatepec. Recuperado el 29 de abril de 2012, de [http://www.lgama.com.mx/images/publicaciones/nousitz\\_bluetooth.pdf](http://www.lgama.com.mx/images/publicaciones/nousitz_bluetooth.pdf)

Google Play (2012). Acerca de Google Play. [Página Web]. Recuperado el 15 de junio de 2012, de <https://play.google.com/about/>

Holley Metering Ltd (2010). Medidor Electrónico. [Página Web]. Recuperado en febrero de 2012, de <http://www.holleymeter.com/es/products.php?pid=10>

ImaginaNET (2010). Programando para Android: Una sencilla aplicación. [Página Web]. Recuperado el 15 de junio de 2012, de

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

<http://www.imaginanet.com/blog/programando-para-android-una-sencilla-aplicacion.html>

Integración Tecnológica (2001). Lenguaje Basic desde el inicio. [Página Web].

Recuperado el 14 de enero de 2013, de <http://www.jlmtzhermida.net/Basic.asp>

IEC (2003). International Estándar. [Documento PDF]. Recuperado el 16 de Junio de

2012, de [http://webstore.iec.ch/preview/info\\_iec62053-21%7Bed1.0%7Den\\_d.pdf](http://webstore.iec.ch/preview/info_iec62053-21%7Bed1.0%7Den_d.pdf)

Let`s Make Robots (2008) 28 pin Project Board (AXE020), picaxe for dummies. [Página

Web]. Recuperado el 13 de enero de 2013, de <http://letsmakerobots.com/node/75>

López M. & Fernández J. (2006), Sistemas de control programado. Sistema PICAXE,

[Documento PDF]. Recuperado el 16 de Junio de 2012, de

[http://www.tecnologiafacil.net/documentacion/Curso\\_Picaxe.pdf](http://www.tecnologiafacil.net/documentacion/Curso_Picaxe.pdf)

Mejía Aurelio, (2004), Guía práctica para manejar y reparar el computador. [Documento

PDF], Colombia. Recuperado el 16 de Junio de 2012, de

<http://books.google.co.ve/books?id=eAsV2vaIenkC&printsec=frontcover&hl=es#v=onepage&q&f=false>

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

National Instrument <sup>TM</sup>, (2006), Comunicación Serial: Conceptos Generales, [Página Web]. Recuperado el 27 de Octubre de 2012, de <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>

Ortíz Lourdes, (2012), Guía para la Documentación del Trabajo Especial De Grado de la Escuela de Telecomunicaciones. [Documento Word], Venezuela.

PICAXE® (2012), PICAXE Manual, [Documento PDF]. Recuperado el 05 de febrero de 2013, de [http://www.picaxe.com/docs/picaxe\\_manual1.pdf](http://www.picaxe.com/docs/picaxe_manual1.pdf)

PICAXE® (2012), PICAXE microcontroller, [Documento PDF]. Recuperado el 16 de Junio de 2012, de <http://www.picaxe.com/docs/picaxem2.pdf>

PICAXE® (2012), PICAXE Programming editor, [Página Web]. Recuperado el 12 de Enero de 2013, de <http://www.picaxe.com/Software/PICAXE/PICAXE-Programming-Editor>

Processing (s.f.), [Página Web]. Recuperado el 13 de enero de 2013, de <http://www.processing.org/>

Rallat, C. (2011, 06 de Noviembre). Aprender a programar y desarrollar aplicaciones Android. [Página Web]. Recuperado el 15 de junio de 2012, de

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

<http://www.android.es/aprender-a-programar-y-desarrollar-aplicaciones-android.html#axzz1y1ZNW8TE>

Ríos, F. (2000). Acometidas y Medidores/ Pérdidas de Energía. Ecuador. E.E.R.S.A.

Schildt, H. (2007). Java: Manual de Referencia. Séptima Edición. McGraw Hill

SDM Sistemas Digitales de México. (2012). MODBUS Parte I: Comunicación Serial. [Página Web]. Recuperado el 27 de Octubre de 2012, de <http://www.sidmx.com/blog/24-modbus-parte-i-comunicacion-serial.html>

Siemens Aktiengesellschaft (1987). Componentes Electrónicos. Descripción técnica y características para estudiantes. Barcelona: Marcombo.

SparkFun Electronics® (2012). Bluetooth Modem –BlueSMiRF Silver, [Página Web]. Recuperado el 28 de octubre de 2012, de <https://www.sparkfun.com/products/10269>

Tamayo, A. (2009). Comunicación Serial, [Página Web]. Recuperado el 27 de octubre de 2012, de <http://galaxi0.wordpress.com/el-puerto-serial/>

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

Universitat de Valencia, (s.f.), [Página Web]. Recuperado el 12 de mayo de 2012, de <http://www.uv.es/marinjl/electro/opto.html>

4Android (s.f.). Android: Paso a Paso. [Página Web]. Recuperado el 15 de junio de 2012, de <http://4ndroid.com/android-paso-a-paso-iv/>

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

## **APÉNDICE A**

**Manual de Usuario.**

## Manual de Usuario

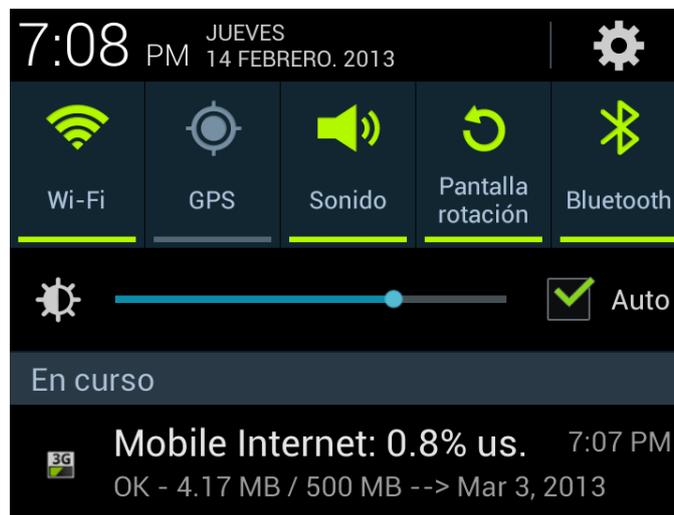
A continuación se presenta una guía paso a paso para comprender el funcionamiento y uso de la aplicación de nombre “Factura”.

Esta aplicación tiene como fin principal, generar la factura de un cliente, conectando el dispositivo móvil con el medidor eléctrico utilizando Bluetooth para luego enviarla por correo electrónico.

*Nota 1: Antes de comenzar con la guía, se debe realizar el proceso de vinculación o pareado entre el dispositivo y el medidor del usuario.*

### Pasos a seguir para la vinculación.

1. Con el dispositivo encendido y desbloqueado, encender el Bluetooth.



Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

2. Se mostrará una lista con una serie de dispositivos vinculados, y otros disponibles para vinculación.



3. Seleccionar de la lista de dispositivos disponibles, el que tiene por nombre “RN42-B71A”.
4. Aparecerá un cuadro donde se pide un código de vinculación. Aquí debe escribir la contraseña “1234” (Sin las comillas).

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---



5. Ya ha vinculado ambos dispositivos.

*Nota 2: Se recomienda con alta prioridad que se apague el Bluetooth del dispositivo antes y después de utilizar la aplicación para evitar fallas inesperadas.*

**Pasos a seguir para el funcionamiento.**

1. Con el dispositivo encendido y desbloqueado, ingresar al menú de aplicaciones.
2. Buscar entre las aplicaciones aquella que posee por nombre "Factura" y seleccionarla.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

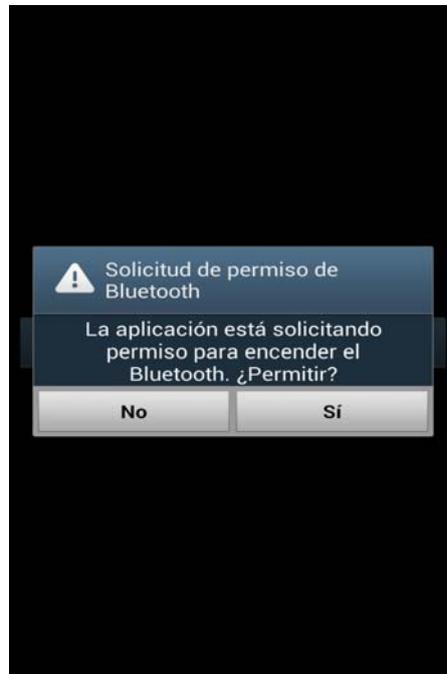
---



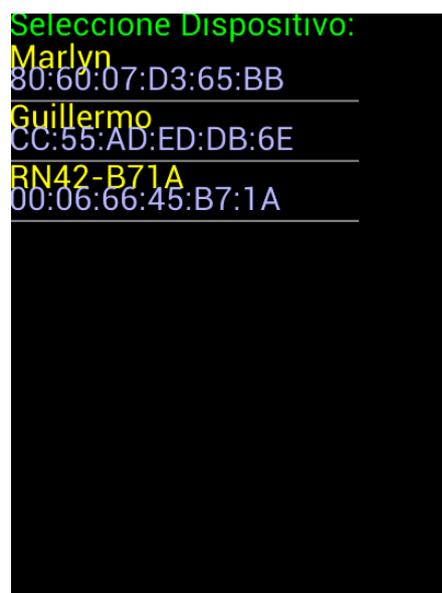
3. Si siguió correctamente la Nota 2, aparecerá un cuadro indicando que la aplicación desea encender el Bluetooth y le ofrecerá las opciones **Si** o **No**. Debe seleccionar Si para continuar con el paso 4. Caso contrario, se mostrará en pantalla un error.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---



4. A continuación, se le presenta una lista con los dispositivos vinculados. Debe seleccionar el que tiene por nombre "RN42-B71A".



Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

5. Una vez seleccionado, se muestra en pantalla la factura del usuario con formato preestablecido y un botón con la etiqueta “Enviar E-Mail.”



6. Presione dicho botón “Enviar E-Mail” y le aparecerá un cuadro que muestra los clientes de correo electrónico instalados en el dispositivo. Seleccione el de su preferencia.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

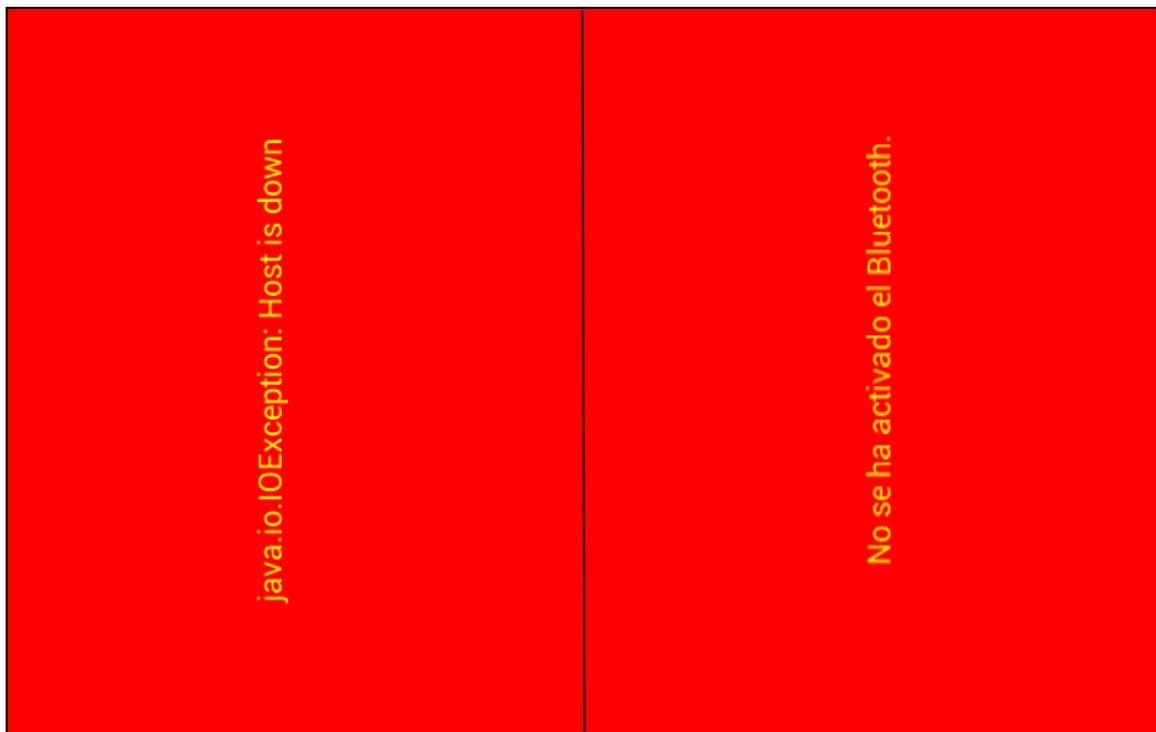


7. Ahora se muestra un formato de correo electrónico con todos los campos llenos. Solo debe presionar Enviar, de esta forma, se enviará el correo a la dirección del cliente y a la de la compañía.



**Aspectos de Interés:**

- Si al momento de seleccionar el “RN42-B71A” para visualizar la factura, se muestra un mensaje de error, pudo haber ocurrido que no se haya realizado la conexión entre los dispositivos. Se recomienda realizar nuevamente la guía paso por paso. Los errores que se pueden presentar durante la ejecución de la aplicación los siguientes:



- Para enviar el correo electrónico, es necesario que el dispositivo cuente con un plan de datos adquirido con algún operador, o en su defecto, estar conectado vía WiFi a una red inalámbrica.

## **APÉNDICE B**

### **Código de la Aplicación.**

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.bluetooth.BluetoothSocket;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.content.IntentFilter;

import java.util.ArrayList;

import java.io.IOException;

import java.io.InputStream;

import java.lang.reflect.Method;

import android.app.Activity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import java.util.Properties;

import javax.mail.*;

import javax.mail.internet.*;
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
import javax.mail.internet.MimeMessage;

import javax.activation.*;

import android.widget.Toast;

private static final int REQUEST_ENABLE_BT = 3; // Asigna a la variable
REQUEST_ENABLE_BT el valor 3

ArrayList dispositivos; // Se crea una lista de tipo ArrayList para llenarla con los
dispositivos

BluetoothAdapter adaptador; // Se asigna un adaptador bluetooth a la variable "adaptador"

BluetoothDevice dispositivo; // Se asigna un dispositivo bluetooth a la variable
"dispositivo"

BluetoothSocket socket; // Se asigna un socket bluetooth a la variable "socket"

InputStream ins; // Declara la variable "ins" de tipo InputStream

boolean registrado = false; // Declara "registrado" como un booleano inicializado en Falso

PFont f1; /* f1 y f2 asignan las

PFont f2; fuentes qu se usaran en el diseño de la aplicación */

int estado; // Declara la variable "estado" de tipo entero

String error; // Declara la variable "error" de tipo String

double contador; /* Declara las variables "contador", "consumo"
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo  
Android para su procesamiento y facturación.

---

double consumo; y "total" del tipo double para poder

double total; realizar los cálculos \*/

//

void setup() {

  size(720,1280);

  frameRate(25);

  f1 = createFont("Arial",30,true);

  f2 = createFont("Arial",20,true);

  stroke(255);

}

//

void draw() {

  switch(estado)

  {

    case 0:

      listaDispositivos("Buscando Dispositivos...", color(255, 0, 0));

      break;

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

case 1:

```
    listaDispositivos("Seleccione Dispositivo:", color(0, 255, 0));
```

```
    break;
```

case 2:

```
    conectaDispositivo();
```

```
    break;
```

case 3:

```
    muestraDatos();
```

```
    break;
```

case 4:

```
    muestraError();
```

```
    break;
```

```
}
```

```
}
```

```
////////////////////////////////////
```

```
void onStart()
```

```
{
```

```
    super.onStart();
```

```
    println("onStart");
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
adaptador = BluetoothAdapter.getDefaultAdapter();

if (adaptador != null)

{

    if (!adaptador.isEnabled())

    {

        Intent enableBtIntent = new

Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);

        startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);

    }

    else

    {

        empieza();

    }

}

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void onStop()

{

    println("onStop");

}
```

```
if(socket != null)
{
    try
    {
        socket.close();
    }
    catch(IOException ex)
    {
        println(ex);
    }
}
super.onStop();
}
////////////////////////////////////
void onActivityResult (int requestCode, int resultCode, Intent data)
{
    println("onActivityResult");
    if(resultCode == RESULT_OK)
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
{  
    println("RESULT_OK");  
    empieza();  
}  
else  
{  
    println("RESULT_CANCELED");  
    estado = 4;  
    error = "No se ha activado el Bluetooth."  
}  
}  
  
////////////////////////////////////  
  
void mouseReleased()  
  
{  
    switch(estado)  
    {  
        case 0:  
            break;  
        case 1:
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
    compruebaEleccion();

    break;

case 3:

    compruebaBoton();

    break;

}

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void empieza()

{

    dispositivos = new ArrayList();

    for (BluetoothDevice dispositivo : adaptador.getBondedDevices())

    {

        dispositivos.add(dispositivo);

    }

    estado = 1;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void listaDispositivos(String texto, color c)
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
{
background(0);
textFont(f1);
fill(c);
text(texto,0, 20);
if(dispositivos != null)
{
for(int indice = 0; indice < dispositivos.size(); indice++)
{
BluetoothDevice dispositivo = (BluetoothDevice) dispositivos.get(indice);
fill(255,255,0);
int posicion = 50 + (indice * 55);
if(dispositivo.getName() != null)
{
text(dispositivo.getName(),0, posicion);
}
fill(180,180,255);
text(dispositivo.getAddress(),0, posicion + 20);
fill(255);
}
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
        line(0, posicion + 30, 319, posicion + 30);
    }
}
}

/////////////////////////////////////////////////////////////////

void compruebaEleccion()
{
    int elegido = (mouseY - 50) / 55;
    if(elegido < dispositivos.size())
    {
        dispositivo = (BluetoothDevice) dispositivos.get(elegido);
        println(dispositivo.getName());
        estado = 2;
    }
}

/////////////////////////////////////////////////////////////////

void conectaDispositivo()
{
    try
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
{  
  
    Method m = dispositivo.getClass().getMethod("createRfcommSocket", new Class[] {  
int.class });  
  
    socket = (BluetoothSocket) m.invoke(dispositivo, 1);  
  
  
    socket.connect();  
  
    ins = socket.getInputStream();  
  
    estado = 3;  
  
}  
  
catch(Exception ex)  
  
{  
  
    estado = 4;  
  
    error = ex.toString();  
  
    println(error);  
  
}  
  
}  
  
////////////////////////////////////
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
void muestraDatos()
{
    try
    {
        while(ins.available()>0)
        {
            consumo = (double)ins.read();

            if (consumo == 230) {

                consumo = 8;

                contador = consumo;

                consumo = consumo/1600;

            }

            total = consumo*2;

        }
    }
    catch(Exception ex)
    {

        estado = 4;

        error = ex.toString();
    }
}
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
println(error);

}

background(0);

fill(255);

text("Servicio de Electricidad.",width/7,height/9);

text("Sr.(a) Guillermo Hernández.",width/7,height/6);

text("C.I.: 18.914.470.",width/7,height/5);

text("E-Mail: guillermohdezg@gmail.com",width/7,height/4);

text("Valor del Contador: "+contador+" impulsos.",width/7,height/3.3);

text("Ud. Consumió este mes: "+consumo+" Kwh.",width/7,height/2.5);

text("Ud. debe pagar: "+total+" BsF.",width/4,height/2);

stroke(255, 255, 0);

fill(255, 0, 0);

rect(160, 725, 400, 50);

fill(255, 255, 0);

text("Enviar E-Mail.", 280, 760);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void enviaCorreo() {
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
Intent correo = new Intent(Intent.ACTION_SEND);

correo.setType("message/rfc822");

correo.putExtra(Intent.EXTRA_EMAIL, new
String[]{"guillermohdezg@gmail.com", "marlynzapata@gmail.com"});

correo.putExtra(Intent.EXTRA_SUBJECT, "Factura del Mes.");

correo.putExtra(Intent.EXTRA_TEXT, "Servicio de Electricidad."+"\n"+"Sr.(a)
Guillermo Hernández."+"\n"+"C.I.: 18.914.470."+"\n"+"E-Mail:
guillermohdezg@gmail.com"+"\n"+"Ud. Consumió este mes: "+consumo+"
Kwh."+"\n"+"Ud. debe pagar: "+total+" BsF."+"\n"+"Gracias.");

try {

    startActivity(Intent.createChooser(correo, "Enviar Correo..."));

} catch (android.content.ActivityNotFoundException ex) {

    Toast.makeText(Factura.this, "Ud. no tiene clientes de Correo Electrónico instalados en
su Dispositivo.", Toast.LENGTH_SHORT).show();

}

}

////////////////////////////////////

void compruebaBoton()

{

    if(mouseX > 160 && mouseX < 560 && mouseY > 725 && mouseY < 775)
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
{
    try
    {
        enviaCorreo();
    }
    catch(Exception ex)
    {
        estado = 4;
        error = ex.toString();
        println(error);
    }
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void muestraError()
{
    background(255,0,0);
    fill(255,255,0);
    textFont(f2);
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

```
textAlign(CENTER);
```

```
translate(width/2,height/2);
```

```
rotate(3*PI/2);
```

```
text(error,0,0);
```

```
}
```

```
////////////////////////////////////
```

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

## **APÉNDICE C**

**Código del Microcontrolador PICAXE.**

**Código Fuente: PICAXE 28X1.**

**symbol cont = b0**

**symbol cond = b1**

**let cont = 0**

**let cond = 0**

**setfreq m8**

**inicio:**

**sertxd(#b0)**

**if input3 is on then**

**if cond = 0 then**

**cond = 1**

**goto contador**

**endif**

**if cond = 1 then**

**goto inicio**

**endif**

**else**

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

**cond = 0**

**goto inicio**

**endif**

**contador:**

**cont = cont + 1**

**goto inicio**

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

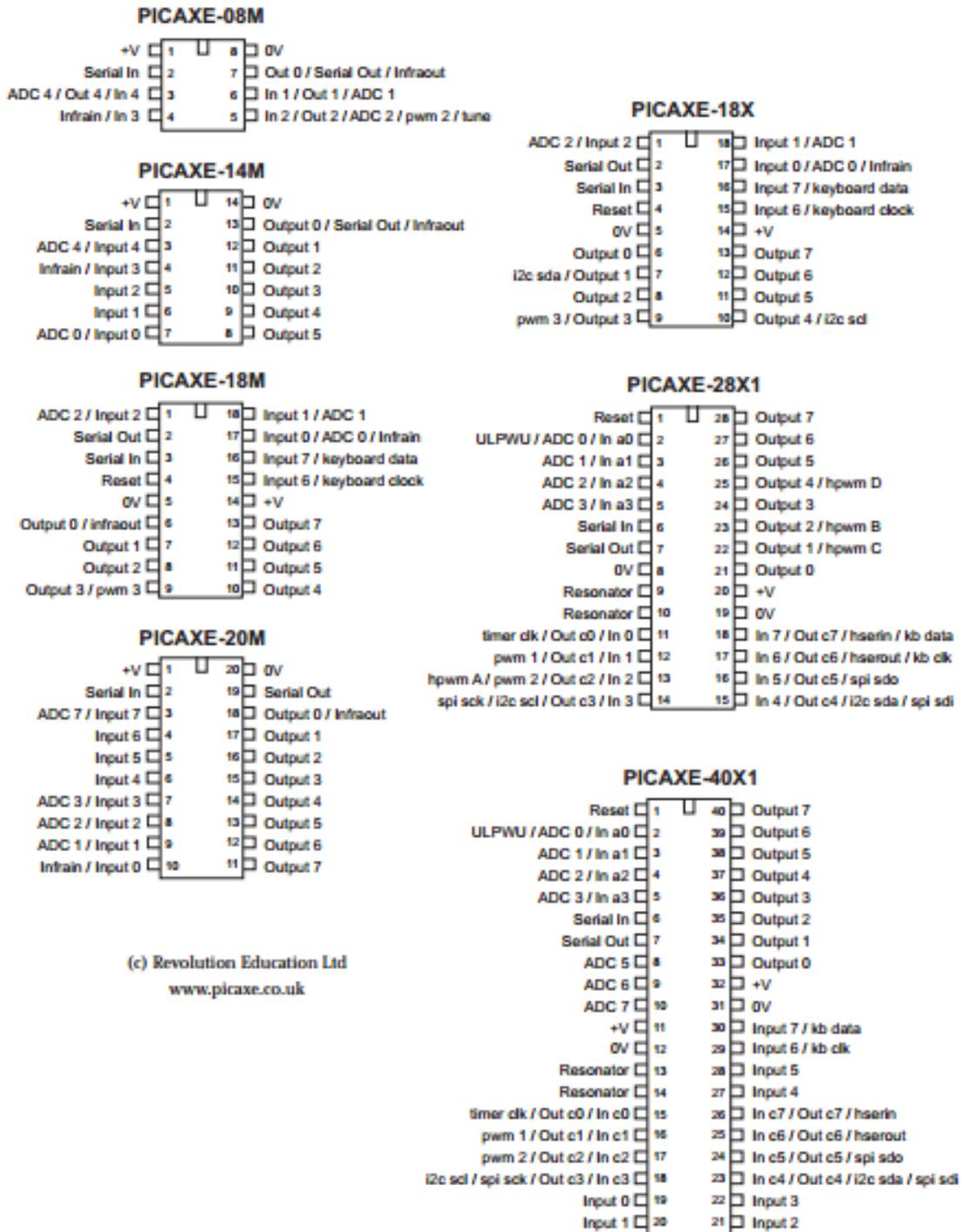
---

## **ANEXO A**

**DataSheet del Microcontrolador PICAXE 28X1**

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

At a glance - pinout diagrams (older parts):



## Which PICAXE chip?

The PICAXE system can be used with different physical sizes of PICAXE chip (8, 14, 18, 20, 28 and 40 pin). The primary difference between the sizes of chips is the number of input/output pins available – the larger chips cost a bit more but have more available input/output pins. The same BASIC language is common to all size chips.



Within a chip size there are also different variants (e.g. for the 20 pin PICAXE the 20M2 and 20X2 variants are available). The principal difference between the variants is the amount of memory (ie how long a program can be downloaded into the chip). The higher specification variants also have some increased functionality (e.g. high resolution analogue inputs and i2c compatibility, as described in the next section). Any project can be upgraded to the next level variant at any point (e.g. if your program is too long for the variant of chip used) by simply replacing the microcontroller in your circuit with the upgraded variant.

All upgraded variants are pin and program compatible with the lower specification device.

The recommended part for new designs is:

### Standard:

08	PICAXE-08M2
14	PICAXE-14M2
18	PICAXE-18M2
20	PICAXE-20M2
28	PICAXE-28X1
40	PICAXE-40X1

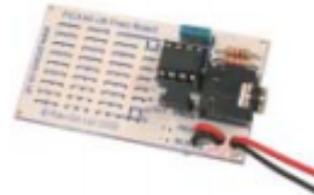
### Advanced:

20	PICAXE-20X2
28	PICAXE-28X2
40	PICAXE-40X2

### PICAXE Starter Packs

To get started with the PICAXE system a starter pack is recommended. All 5 starter packs contain the same CDROM (containing the manuals and free programming software), USB (or serial) download cable and battery box. However the project board and type of PICAXE chip varies in each starter pack as indicated below. 3 x AA batteries are also required (not included).

**PICAXE-08 Starter Pack (AXE003U)**  
PICAXE-08 proto board, PICAXE chip, CDROM, USB download cable and battery box. Self assembly kit.



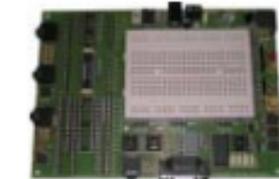
**PICAXE-14 Starter Pack (AXE004U)**  
**PICAXE-20 Starter Pack (AXE005U)**  
PICAXE-14 project board, PICAXE chip, CDROM, USB download cable and battery box. Self assembly kit.



**PICAXE-18 Starter Pack (AXE002U)**  
PICAXE-18 standard project board, PICAXE-18M2 chip, CDROM, USB download cable and battery box. Pre-assembled (18M2 chip supplied).



**PICAXE-28X1 Starter Pack (AXE001U)**  
PICAXE-28 project board, connector cables, PICAXE-28X1 chip, CDROM, USB download cable and battery box. Pre-assembled (28X1 chip supplied).



**Development Starter Pack (AXE091U)**  
Specifically designed for hobbyists with large breadboarding area and inputs/ outputs for experimentation. The development PCB can support all sizes of PICAXE chips and is supplied with a PICAXE-18M2 chip. Pre-assembled.



**Tutorial Starter Pack (AXE050U)**  
The tutorial pack is designed for school use to enable students to rapidly learn the PICAXE language by a series of structured tutorials (provided on the CDROM). Pre-assembled board with LDR, switches and output display.

# Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

## Software Installation

### Computer Requirements:

To install the software you require a computer running Windows 95 or later with approximately 50MB free space. Any computer that runs the Windows operating system will work in textual 'BASIC' mode, however a Pentium 4 processor or later is recommended for graphical flowchart work.

### Installation:

- 1) Start up and log into your computer (some operating systems require that you log in as 'Administrator' to install software).
- 2) Insert the CD, or download and run the installation file from the software page at [www.picaxe.co.uk](http://www.picaxe.co.uk)
- 3) Follow the on-screen instructions to install the software. On older computers you may be instructed to restart the computer after installation.
- 4) Insert the AXE026 cable into the 9 pin serial port at the back of the computer, or the AXE027 USB cable into the USB port. The AXE027 will require a software driver when first used, a 'New hardware found' wizard will automatically start (see the AXE027 datasheet for more details).
- 5) Click Start>Programs>Revolution Education>Programming Editor to start the software.
- 6) If the Options screen does not automatically appear, click the View>Options menu. On the 'Mode' tab select the size and type of PICAXE microcontroller you are using. On the 'Port' tab select the appropriate serial COM port then click OK.

You are now ready to use the system.

## Installation on RM CC3 networks

The software will run on all school networks, including RM CC3.

- 1) It is recommended you use the uncompressed MSI install provided on the CDROM, rather than the internet download.
- 2) Log on as System Admin and use your preferred distribution software (e.g. RM Application Wizard) to build a distribution package using the MSI install found within the /progedit folder on the CDROM. If preferred you can also manually copy the MSI files into the RMPackages\Applications area.
- 3) Update the package list of the appropriate workstations using the RM Management Console and generate shortcuts as required.
- 4) XP users - note that you may have to create two Software Restrictions 'hash' rules - one to the progedit.exe executable and another to the shortcut. To do this log on as System Admin on an XP workstation, click Start>Programs>System Management>Software Restriction settings. Open Computer Configuration>Windows Settings>Software Restriction Policies>Additional Rules. From the Action menu select 'New Hash Rule' and browse to the progedit.exe executable. Click OK.
- 5) The default save/open folder paths can be edited as required in the file called network.ini found in the main installation folder.

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

PICAXE-28X2/28X1/28X/28A Pinout and Circuit

The pinout diagrams for the 28 pin devices are as follows:  
(0.3" DIL or 300mil SOIC)

PICAXE-28X2

Reset	1	28	B.7 (In / Out)
{touch} (Comp1- / ADC0 / Out / In) A.0	2	27	B.6 (In / Out)
{touch} (Comp2- / ADC1 / Out / In) A.1	3	26	B.5 (In / Out) {ADC13 / touch / pwm}
{DAC / touch} (Comp2+ / ADC2 / Out / In) A.2	4	25	B.4 (In / Out / ADC11) {touch / hpwm D}
{touch} (Comp1+ / ADC3 / Out / In) A.3	5	24	B.3 (In / Out / ADC9) {touch}
Serial In	6	23	B.2 (In / Out / ADC8 / hint2) {touch / hpwm B}
{SRNO} (Out) Serial Out / A.4	7	22	B.1 (In / Out / ADC10 / hint1) {touch / hpwm C}
0V	8	21	B.0 (In / Out / ADC12 / hint0) {touch / pwm / SRI}
Resonator	9	20	+V
Resonator	10	19	0V
{timer clk / Out / In} C.0	11	18	C.7 (In / Out / hserin / kb data) {ADC19 / touch}
{pwm / Out / In} C.1	12	17	C.6 (In / Out / hserout / kb clk) {ADC18 / touch}
{hpwm A / touch / ADC14} {pwm / Out / In} C.2	13	16	C.5 (In / Out / hspi sdo) {ADC17 / touch}
{touch / ADC4} {i2c scl / hspi sck / Out / In} C.3	14	15	C.4 (In / Out / i2c sda / hspi sdi) {ADC16 / touch}

Features shown in brackets { } are not available in older -5V and -3V parts.

PICAXE-28X1

Reset	1	28	Output 7
ULPWU / ADC 0 / In a0	2	27	Output 6
ADC 1 / In a1	3	26	Output 5
ADC 2 / In a2	4	25	Output 4 / hpwm D
ADC 3 / In a3	5	24	Output 3
Serial In	6	23	Output 2 / hpwm B
Serial Out	7	22	Output 1 / hpwm C
0V	8	21	Output 0
Resonator	9	20	+V
Resonator	10	19	0V
timer clk / Out c0 / In 0	11	18	In 7 / Out c7 / hserin / kb data
pwm 1 / Out c1 / In 1	12	17	In 6 / Out c6 / hserout / kb clk
hpwm A / pwm 2 / Out c2 / In 2	13	16	In 5 / Out c5 / spi sdo
spi sck / i2c scl / Out c3 / In 3	14	15	In 4 / Out c4 / i2c sda / spi sdi

PICAXE-28A

Reset	1	28	Output 7
ADC 0	2	27	Output 6
ADC 1	3	26	Output 5
ADC 2	4	25	Output 4
ADC 3	5	24	Output 3
Serial In	6	23	Output 2
Serial Out	7	22	Output 1
0V	8	21	Output 0
Resonator	9	20	+V
Resonator	10	19	0V
Input 0 / In/In	11	18	Input 7 / Keyboard data
Input 1	12	17	Input 6 / Keyboard clock
Input 2	13	16	Input 5
Input 3	14	15	Input 4

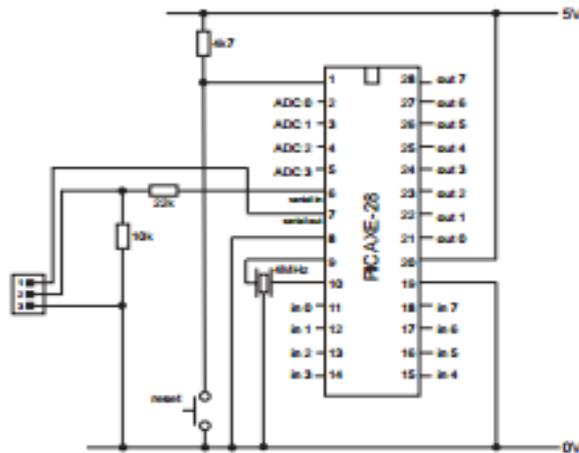
PICAXE-28X

Reset	1	28	Output 7
ADC 0 / In a0	2	27	Output 6
ADC 1 / In a1	3	26	Output 5
ADC 2 / In a2	4	25	Output 4
ADC 3 / In a3	5	24	Output 3
Serial In	6	23	Output 2
Serial Out	7	22	Output 1
0V	8	21	Output 0
Resonator	9	20	+V
Resonator	10	19	0V
In0 / Out c0 / In/In	11	18	In7 / Out c7 / keyboard data
In 1 / Out c1 / pwm 1	12	17	In6 / Out c6 / keyboard clock
In 2 / Out c2 / pwm 2	13	16	In 5 / Out c5
In 3 / Out c3 / i2c scl	14	15	In 4 / Out c4 / i2c sda

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

The minimum operating circuit for the 28 pin devices is:



See the USB / Serial Download Circuit section of this manual for more details about the download circuit.

**Notes:**

- 1) The 10k/22k resistors must be included for reliable operation.  
DO NOT leave the serial in pin floating as THE PROGRAM WILL NOT RUN!
- 2) The reset pin must be tied high with the 4k7 resistor to operate.
- 3) Resonator:
 

28X2	(optional)	4 (16), 8(32), 10 (40) or 16(64) MHz
28X2-5V	(optional)	4 (16), 8(32), or 10(40) MHz
28X2-3V	(optional)	4 (16), 8(32), 10 (40) or 16(64) MHz
28X1	(optional)	16MHz
28X		4, 8 or 16MHz
28 / 28A		4MHz

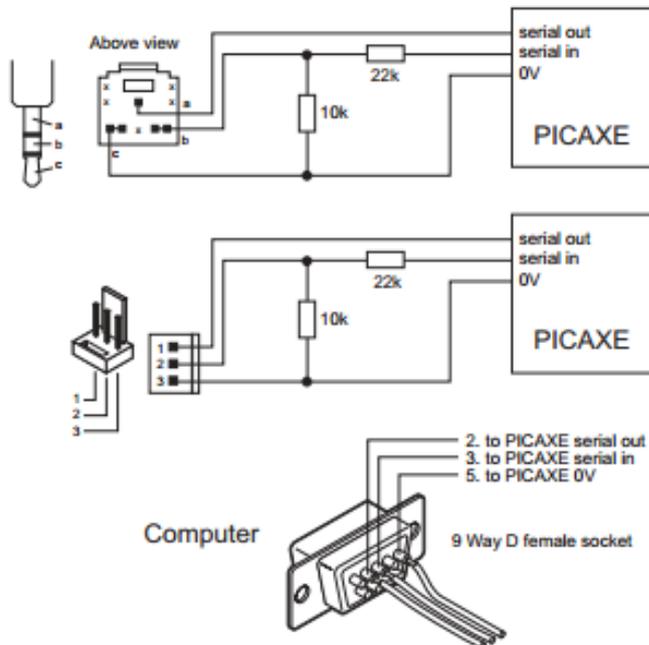
The 28X1 and 28X2 have an internal resonator (4 or 8MHz) and so the external resonator is optional. On 28A and 28X parts it is compulsory.  
The 28X2 has an internal 4xPLL circuit. This multiplies the external clock speed by 4. Therefore an external 8MHz resonator gives an actual internal operating clock frequency of 4x8MHz=32MHz.

**IMPORTANT NOTE** - this manual describes use of the standard range (3-5V) parts. The X2 parts are also available in special low power (1.8V to 3.3V) variants. Use of a 5V supply on a 3.3V part will permanently damage it!

### Serial Download Circuit

The serial download circuit is identical for all PICAXE chips. It consists of 3 wires from the PICAXE chip to the AXE026 serial cable. One wire sends data from the computer to the serial input of the PICAXE, one wire transmits data from the serial output of the PICAXE to the computer, and the third wire provides a common ground. See the USB adapter section for details on how to use the USB port adapter.

The minimum download circuit is shown here. This circuit is appropriate for most educational and hobbyist work.



Note that the two resistors are not a potential divider. The 22k resistor works with the internal microcontroller diodes to clamp the serial voltage to the PICAXE supply voltage and to limit the download current to an acceptable limit. The 10k resistor stops the serial input 'floating' whilst the download cable is not connected. This is essential for reliable operation.

The two download resistors must be included on every PICAXE circuit (i.e. not built into the cable). The serial input must never be left unconnected. If it is left unconnected the serial input will 'float' high or low and will cause unreliable operation, as the PICAXE chip will receive spurious floating signals which it may interpret as a new download attempt.

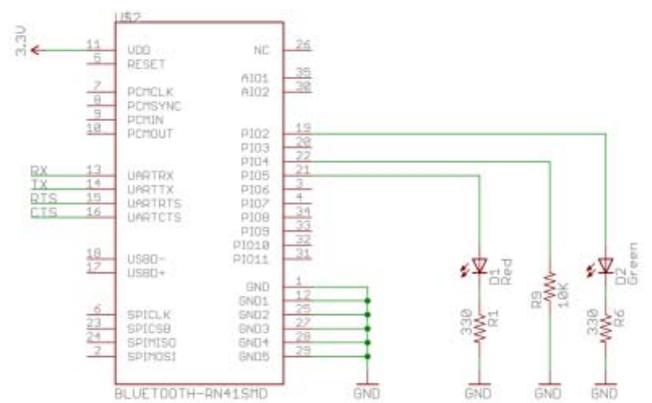
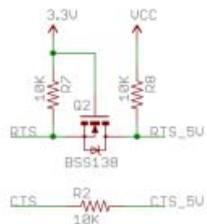
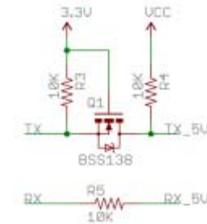
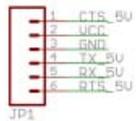
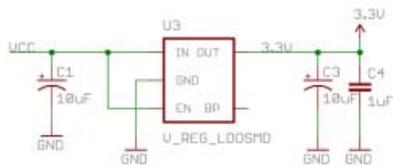
Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.

---

## **ANEXO B**

**DataSheet del Modem Bluetooth BlueSMiRF RN-42 Silver.**

Lectura en un Medidor Eléctrico y transmisión vía Bluetooth de los Datos a un dispositivo Android para su procesamiento y facturación.



⊗ ⊗	
Spark Fun Electronics	
TITLE: BlueSMiRF RN-v1	SFE
Document Number:	REV:
Date: 2/20/2008 11:09:19 PM	Sheet: 1/1