

PLACA DE ENTRENAMIENTO PARA APLICACIONES ELECTRÓNICAS CON TERMINALES MÓVILES BASADOS EN SISTEMA OPERATIVO ANDROID.

Edgar A. Maya, Byron R. Valenzuela

Resumen— El presente documento da a conocer el diseño e implementación de un entrenador electrónico para dispositivos móviles basados en el sistema operativo Android, este sistema incluye varios dispositivos como led's, pulsadores, switch, buzzer, releé, display's, sensores de temperatura y módulos de comunicación como GPS, Wi-Fi, Xbee, I2C, con los que el móvil y el entrenador interactuarán para desarrollar aplicaciones.

Términos Indexados— Android, GPS, Wi-fi, Xbee, I2C

I. INTRODUCCIÓN

El crecimiento de la tecnología y la gran acogida de los equipos móviles basados en sistema operativo Android han hecho que se considere el desarrollo de un entrenador para los estudiantes de la carrera de Ingeniería en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte, con el que los mismos podrán realizar varias aplicaciones basadas en terminales móviles con sistema operativo Android e interactuar con los distintos periféricos con los que el entrenador cuenta.

Documento recibido el 12 de octubre de 2014. Esta investigación se realizó como proyecto previo para obtener el título profesional en la carrera de Ingeniería en Electrónica y Redes de Comunicación de la Facultad de Ingeniería en Ciencias Aplicadas (FICA) de la Universidad Técnica del Norte.

E.A. Maya, trabaja en la Universidad Técnica del Norte, en la Carrera de Ingeniería en Electrónica y Redes de Comunicación, Av. 17 de Julio sector el Olivo, Ibarra-Ecuador (teléfono: 5936-2955-413; e-mail: eamaya@utn.edu.ec).

B.R. Valenzuela, egresado de la Carrera de Ingeniería Electrónica y Redes de Comunicación (teléfono: 5936-2604-664; e-mail: byron_101990@hotmail.com).

Por su parte los módulos y periféricos que el estudiante tendrá a su disposición para el desarrollo de las aplicaciones son los siguientes: entrada/salida digitales, led's, conversores análogo digitales, teclado matricial, Lcd 16x2, pulsadores, matriz de leds, buzzer, comunicaciones: Serial, Pwm, I2C, módulo Xbee, Bluetooth, Gps, Wi-fi y sensor de temperatura. Impulsando así al desarrollo tecnológico de la educación y el crecimiento de conocimientos por parte de los estudiantes.

II. CONCEPTOS BÁSICOS

A. ¿Qué es IOIO?

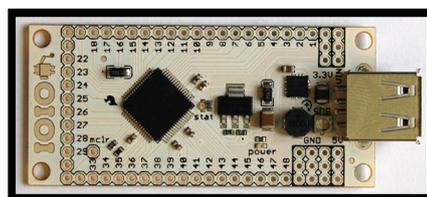


Figura 1. Placa IOIO

Referencia: Monk Simon, (2012). Making Android Accessories with IOIO

IOIO es una placa diseñada y fabricada especialmente para trabajar con dispositivos móviles basados en el sistema operativo Android, desde la versión 1.5 (Donut) hasta la versión actual de Android que es la versión 4.1

La conectividad con el dispositivo móvil se la puede realizar mediante la conexión del cable USB o mediante Bluetooth; IOIO contiene un microcontrolador PIC que interpreta el

código escrito en la aplicación Android, y permite interactuar con los distintos módulos con los que cuenta como son: Entrada / Salida Digital, PWM, conversores análogo-digitales, I2C, SPI, UART.

B. Características y estructura de la placa IOIO

La placa IOIO tiene un peso aproximado de 13gr., 6,8cm de largo por 3cm. de ancho, cuenta físicamente con 55 pines, se alimenta con una fuente externa entre los pines GND y Vin, cuenta con un conector USB micro hembra por el que se conecta con un cable adaptador al equipo móvil Android, posee dos led's, un led indica si la placa está alimentada y el otro si la placa pudo conectarse con el dispositivo Android.

IOIO contiene un microcontrolador PIC de la familia 24FJ128DA210, que es un microcontrolador de 16 bits que actúa como un host el cual recibe instrucciones de la aplicación Android instalada en el dispositivo móvil. Este microcontrolador posee una arquitectura Harvard modificada, un oscilador interno de 8Mhz y permite una corriente de 20mA de entrada y salida en todos sus puertos.

La placa IOIO contiene los siguientes pines y componentes:

- Conector micro USB (tipo A).- conector usado para vincular el dispositivo móvil Android.
- Pines GND (9 pines).- Pines de conexión a tierra.
- Pines Vin. (3 pines).- Pines de alimentación de la IOIO, el voltaje suministrado va desde 5V a 15V.
- Pines 5V (3 pines).- Se usa como salida de 5V; también puede ser usado como alimentación de 5V.

- Pines 3.3V (3 pines).- Son pines de salida de 3.3V.
- Pin Mclr (1 pin).- Este pin solo se utiliza para la actualización de firmware.
- Power led.- Led que se enciende cuando la IOIO es alimentada.
- Stat led.- Led que se enciende cuando la IOIO se conecta con el dispositivo móvil.
- Charge current trimer (CHG).- Sirve para ajustar la cantidad de corriente que suministra a la línea VBUS del USB.
- Pines de entrada y salida.- Los pines de entrada y salida de la IOIO, son usados para conectar circuitos externos a las distintas interfaces de la placa, todos los pines de la placa IOIO pueden ser usados como entradas y salidas digitales de 3.3V
 - Pines rodeados por un cuadrado, pueden ser usados como entradas analógicas de 3.3V
 - Pines rodeados con un círculo, pueden ser usados como entradas y salidas digitales tolerantes a 5V.

C. Conexión de la placa IOIO

La IOIO integra un firmware que permite la comunicación entre el dispositivo móvil Android y la placa, este firmware se divide en dos programas los dos son grabados en la memoria flash del microcontrolador, los programas son “aplicación” y “gestor de arranque (bootloader)”.

El gestor de arranque es el primer programa que se ejecuta cuando la IOIO se reinicia este firmware establece la comunicación entre la placa y el móvil Android, comprueba si existe el código escrito de la aplicación si es así esta lo ejecuta caso contrario permanece en espera. Aplicación este firmware se comunica con las diferentes librerías que contiene

IOIOLib para el control de los pines y sus módulos internos.

Tanto la comunicación por cable como por Bluetooth crea un socket para la transferencia de información y comandos entre el dispositivo Android y el tablero IOIO.

D. ¿Qué es Android?

Android es un sistema operativo para dispositivos móviles al igual que iOS, Symbian, o Blackberry OS, desarrollado inicialmente por Android Inc. en 2003 y comprada posteriormente por Google en 2005, basado en el kernel de Linux 2.6.0, que es el encargado de gestionar aspectos como seguridad, manejo de memoria, procesos, networking y modelo de controladores (Drivers).

Android permite desarrollar aplicaciones mediante el SDK (Software Development Kit), aplicaciones que puedan sacar provecho a las diversas funciones del terminal móvil tales como: GPS, cámara, contactos, acelerómetro, llamadas, etc.

El lenguaje de programación utilizado para la creación de aplicaciones Android es Java, junto al kernel de Linux trabaja Dalvik, es una máquina virtual Java que se ejecuta por encima del núcleo del sistema y permite la ejecución las aplicaciones.

E. Fundamentos para el desarrollo de aplicaciones Android

Las aplicaciones Android están escritas en el lenguaje de programación Java. El código compilado de Java, junto con todos los archivos que forman parte de la aplicación desarrollada luego de ser compilada, forman un paquete con la

extensión .apk que es la que se instalará en el terminal móvil Android.

SDK Android

Android SDK provee un sin número de herramientas y bibliotecas API necesarias para la creación y depuración de las aplicaciones Android; se compone de diversos paquetes, a continuación se describe cada uno ellos.

- **SDK Tools.**

Este paquete contiene las herramientas necesarias para la depuración y pruebas de la aplicación Android, es muy importante tener este paquete siempre actualizado, está ubicado en <SDK>/tools/.

- **Documentation.**

Una copia de la última documentación para la plataforma Android APIs, se encuentra en <SDK>/docs/.

- **SDK platform tools.**

Son herramientas dependientes de la plataforma, necesarias para el desarrollo y depuración de la aplicación, se actualiza cuando aparece una nueva plataforma, está ubicado en <SDK>/platform-tools/.

- **SDK Platform.**

Existe una plataforma disponible para cada versión de Android, incluye un paquete android.jar que contiene todas las librerías para la creación de la aplicación. Para construir la aplicación Android se debe declarar para que plataforma se va crear la aplicación. Se encuentra en <SDK>/platforms/<android-version>/.

- **Sources for Android SDK.**

Es una copia del código fuente de la plataforma Android, es útil para recorrer el código mientras depura la aplicación. Se encuentra en <SDK>/sources/.

- **Samples for SDK.**
Aplicaciones de ejemplo que pueden servir de guía en la construcción de la aplicación Android. Ubicado en <SDK>/platforms/<android-version>/samples/.
- **Google APIs.**
Un SDK adicional que proporciona una plataforma que se puede utilizar para desarrollar una aplicación utilizando los APIs de Google, se utiliza los APIs de Google para crear aplicaciones que necesiten del servicio localización.
- **Android Support.**
Es una biblioteca que se puede añadir a la aplicación y permite usar APIs que no estén en la plataforma estándar. <SDK>/extras/android/support/.
- **Google play billing.**
Proporciona las librerías que permiten integrar los servicios de facturación de la aplicación con Google play. <SDK>/extras/google/
- **Google play licencing.**
Proporciona las librerías y ejemplos que permiten llevar a cabo la verificación de la licencia de la aplicación, cuando se distribuya en Google play. <SDK>/extras/google/.

SDK Manager

El SDK manager muestra los paquetes SDK que necesitan actualización, que están disponibles, o que ya se encuentran instalados los ordena y los separa en herramientas, plataformas y otros componentes para facilitar la descarga de los mismos.

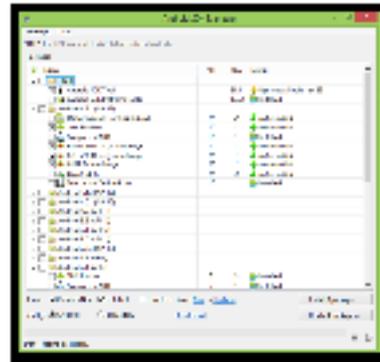


Figura 2. Pantalla principal Android SDK Manager
Referencia: Elaborado por Byron Valenzuela.
Basado en el SDK Manager

Android virtual device (AVD)

El SDK de Android incluye un emulador de dispositivo móvil virtual llamado Android Virtual Device (AVD), este imita todas las características de hardware y software de un dispositivo móvil real y permite comprobar el funcionamiento de las aplicaciones Android sin necesidad de contar con un equipo real.

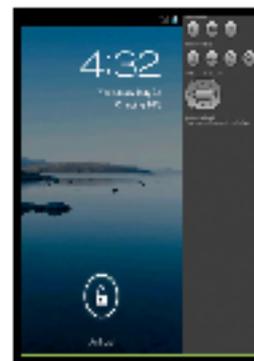


Figura 3. Interfaz de un Dispositivo virtual Android (AVD)
Referencia: Elaborado por Byron Valenzuela.
Basado en el AVD.

F. Estructura de un proyecto Android

Un proyecto Android se compone básicamente por un descriptor (AndroidManifest.xml), el código fuente en

Java y una serie de ficheros con recursos. Cada elemento se almacena en una carpeta específica, como se puede observar en la siguiente figura.



Figura 4. Contenido de un proyecto Android
Referencia: Recuperado de <http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/148-elementos-de-un-proyecto-android>.

- **src**
Esta carpeta contiene todo el código de la aplicación, así como el paquete que acoge a las diferentes clases que componen el mismo. Eclipse crea automáticamente la primera clase correspondiente al primer formulario que tendrá el proyecto.
- **gen**
Contiene las clases generadas automáticamente por el SDK. No el código de estas clases
BuildConfig.java.- Define la constante DEBUG utilizado para ejecutar el código en modo depuración.
R.java.- Contiene la clase que hace referencia al ID de todos los recursos de la aplicación.

- **Android x.x.x**
Contiene el código .jar y el API de Android según la versión.
- **Android Dependencies**
Aquí se encuentran las librerías asociadas con el proyecto.
- **assets**
Se encuentra vacía por defecto, en esta carpeta se pueden colocar archivos o carpetas como: registros de datos, fuentes, ficheros .jar externos que serán usados en la aplicación.
- **bin**
(Universidad Politécnica de Valencia, 2013) Afirma que esta carpeta se compila la aplicación y se genera el .apk que se instalará en el dispositivo móvil Android.
- **libs**
Aquí encontramos las librerías .jar que se usa en el proyecto.
- **res**
Contiene los ficheros de recursos de la aplicación como: imágenes, animaciones, sonidos, texto, etc.
- **AndroidManifest.xml**
Este fichero contiene información esencial acerca de la aplicación para el sistema Android, información necesaria para la ejecución del proyecto. El AndroidManifest contiene: el nombre de la aplicación, descripción de los componentes del proyecto, activity's, permisos necesarios que la aplicación utiliza, librerías requeridas, versión Android de ejecución, etc.

- **ic_launcher-web.png**
Representa el icono del proyecto que es usado en aplicaciones web, el nombre puede variar dependiendo de la aplicación.
- **proguard-project.txt**
Fichero de configuración de la herramienta ProGuard, permite optimizar el código generado.
- **default.properties**
Este fichero es generado automáticamente por el SDK, es utilizado para comprobar la versión del API y otras características cuando se instala la aplicación en el dispositivo móvil. Este archivo no debe ser modificado.

G. Librerías de la placa IOIO

La placa IOIO tiene un acumulado de librerías que ayudarán a realizar las aplicaciones, estas librerías se denominan IOIOLib, IOIOLibPC, IOIOLibBT y IOIOLibAccessory, estas a su vez contienen un conjunto de clases y métodos que permiten controlar todas las características y funciones de la IOIO.

El corazón de todas las aplicaciones IOIO se basa en realizar una instancia hacia el objeto ioio de la librería IOIOLib, que representa una interfaz entre la aplicación y la IOIO permitiendo el control de todas las funciones y módulos de la misma.

Todas las aplicaciones IOIO para Android se componen de las siguientes características y métodos.

- La clase actual del proyecto se debe extender de la clase AbstractIOIOActivity o de la clase IOIOActivity, que permite hacer uso de

los métodos que se encuentran en las librerías de la IOIO.

- Una clase propia extendida de la clase AbstractIOIOActivity.IOIOThread o de la clase BaseIOIOLooper en caso de usar IOIOActivity; en esta clase se aloja los métodos de la aplicación Android que controlan la conexión y uso de las diferentes funciones. Dentro de esta clase se tiene los siguientes métodos.
 - Método setup(); en este método se inicializa y configura las características que van a ser usadas en la aplicación. Ej. Se configura el pin a ser usado y el estado que tendrá inicialmente, además se configura los pines que actuarán con el módulo de comunicación serial, la velocidad de transmisión, paridad, etc. Para crear una instancia hacia la clase IOIO se realiza mediante ioio_.
 - Método loop(); dentro de este método se desarrolla la aplicación Android una vez establecida la conexión, es un bucle infinito donde se controla las funciones y características de la placa.

Tanto el método setup() y el método loop() permiten lanzar excepciones ConnectionLostException e InterruptedException

Cada vez que se ejecuta una aplicación IOIO determina automáticamente la forma de conexión ya sea por cable USB o por Bluetooth, si la conexión falla inmediatamente la aplicación llama a un método incompatible() que verifica la causa y la solución del problema, en caso de no encontrar el posible error, la aplicación

llama a un método disconnected() que hace que la aplicación IOIO sea abortada.

Las aplicaciones IOIO necesitan de permisos para su posible ejecución, estos permisos se incluyen dentro del desarrollo de la Aplicación en el AndroidManifest del proyecto y son los siguientes: Bluetooth (android.permission.BLUETOOTH) e Internet (android.permission.INTERNET) (GitHub, 2008)

IOIO presenta librerías para el control absoluto de todos sus módulos como son: módulo de entrada y salida digital, conversor análogo digital, módulo Pwm, módulo Uart, y módulo I2C.

III. DISEÑO Y CONSTRUCCIÓN DEL ENTRENADOR

A. Descripción general del entrenador.

El entrenador se divide en ocho secciones, cada sección es un módulo del entrenador con la cual se realizará las prácticas, de igual manera cada sección consta de varios elementos electrónicos de acuerdo al módulo al que pertenezcan, además se incluye un protoboard con el que fin de facilitar la conexión de dispositivos que no estén contemplados en el entrenador.

Los ocho bloques constitutivos del entrenador son: módulos: entrada y salida digital, conversión análogo-digital, comunicación serial, comunicación ZigBee y Wi-fi, comunicación Bluetooth y Gps, módulos adicionales, fuente de alimentación, y protoboard.

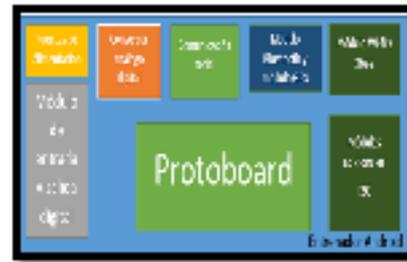


Figura 5. Descripción general del entrenador
Referencia: Elaborado por Byron Valenzuela.

Los módulos Bluetooth, y GPS, cuentan con el estándar mikroBus de miKroelectrónica, este estándar hace que los dispositivos anteriormente mencionados compartan el mismo sócalo de conexión, eliminando así la conexión de accesorios adicionales para el funcionamiento de los equipos y la necesidad de realizar una interfaz por cada módulo que se requiera conectar.

De igual manera los módulos ZigBee y Wi-fi cuentan con un sócalo en común por el mismo hecho de tener las mismas dimensiones, el dispositivo que comparten es un regulador para equipos Xbee denominado Xbee Explorer, este dispositivo cuenta con 10 pines y es capaz de regular la alimentación y los niveles de voltaje de transmisión y recepción de 5V a 3.3V tanto del módulo ZigBee y del Wi-fi.

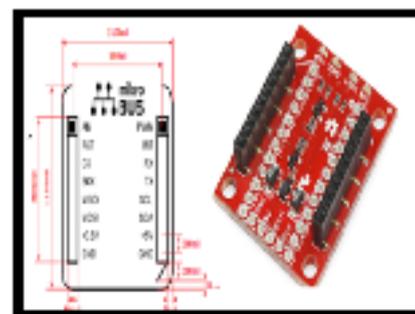


Figura 6. Conector mikroBus(i), Conector Xbee Explorer (d).
Referencia: (mikroElektronika, 2012), (Sparkfun 2012).

B. Módulo de entrada y salida digital
Salida digital

Led's

El entrenador cuenta con 8 diodos led 5mm de color rojo, que ayudan a identificar los niveles de voltaje de salida uno y cero lógico e interactuar con la placa IOIO. La forma de conexión se muestra en la siguiente figura.

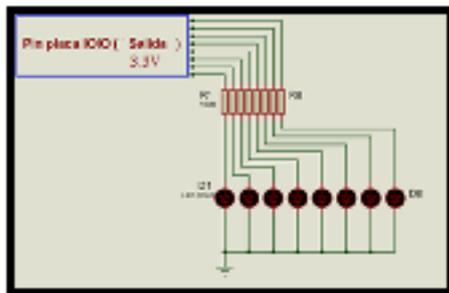


Figura 7. Esquema de conexión led's
Fuente: Elaborado por Byron Valenzuela

Buzzer

Buzzer o zumbador, es un transductor electro acústico que emite sonido continuo al ser polarizado, este dispositivo se usa como alarma frente algún evento ya sea temporizadores, aviso de ingreso de usuario, error de clave entre otras aplicaciones.

El entrenador cuenta con un buzzer HSD que funciona a 12V DC, para polarizar el buzzer se usa un transistor en corte y saturación, con el propósito de proteger al pin IOIO, ya que el buzzer utiliza una corriente de 40mA (datasheet) a 12V.

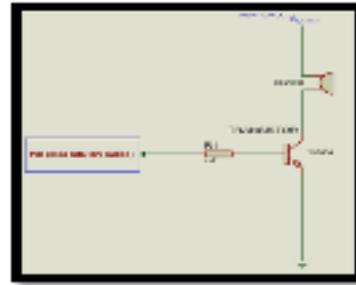


Figura 8. Esquema de conexión buzzer
Fuente: Elaborado por Byron Valenzuela

Display 7 segmentos

Dispositivo electrónico que consta de 7 diodos led que forman una especie de número 8 y su principal uso es para visualizar los números del 0 al 9; aunque se puede visualizar los caracteres que se desee encendiendo y apagando los leds que conforman el display. El entrenador Android cuenta con dos display ánodo común conectados respectivamente con su decodificador 74LS47 como se muestra en la siguiente figura.

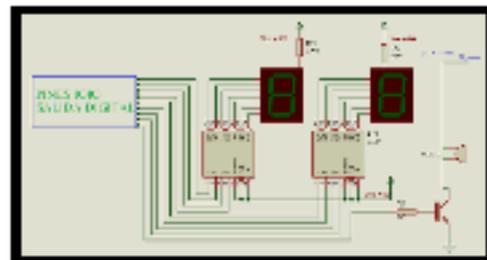


Figura 9. Esquema de conexión IOIO-displays
Fuente: Elaborado por Byron Valenzuela

Pantalla LCD

La pantalla LCD es un dispositivo electrónico que sirve para la representación de caracteres, símbolos e incluso dibujos, el entrenador contiene una LCD de 16x2 es decir dieciséis columnas, y dos filas.

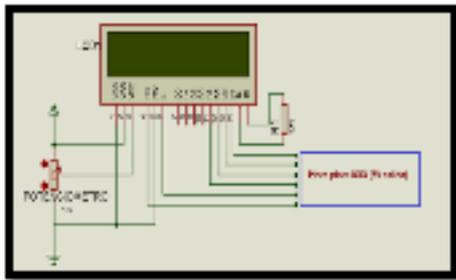


Figura 10. Diagrama de conexión LCD
Fuente: Elaborado por Byron Valenzuela

Relé

Es un dispositivo electromecánico formado por una pequeña barra de hierro llamado núcleo y alrededor una bobina de hilo de cobre, funciona como una especie de interruptor accionado por un electroimán al existir corriente eléctrica por la bobina el núcleo se magnetiza y por ende este se acciona. El relé es un mecanismo que permite controlar un circuito eléctrico de más potencia con la que se está trabajando.

El entrenador cuenta con un relé de 12V de propósito general para polarizar este dispositivo es necesario un transistor en corte y saturación.

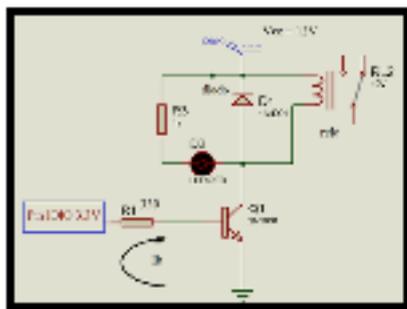


Figura 11. Conexión relé
Fuente: Elaborado por Byron Valenzuela

Entrada digital

Dip-switch

Este dispositivo electrónico es la unión de interruptores separados individualmente, el entrenador contiene un dip-switch de cuatro posiciones, usado para identificar el estado de un pin de entrada de la IOIO ya

sea este un cero o un uno lógico. El entrenador cuenta con un dip-switch de 4 posiciones y la forma de conexión es la siguiente

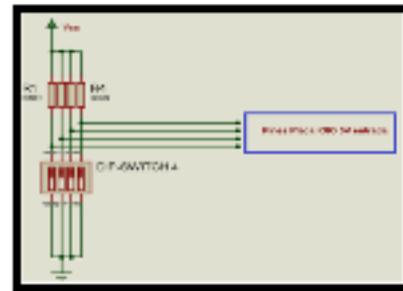


Figura 12. Esquema conexión dip-switch
Fuente: Elaborado por Byron Valenzuela

Pulsadores

Un pulsador o un botón es un dispositivo electrónico que permite cambiar de estado lógico el pin al que está conectado, es decir es un interruptor instantáneo que cambia de estado siempre y cuando se mantenga presionado el mismo. Existen dos tipos de pulsadores normalmente abiertos y normalmente cerrados.

El entrenador cuenta con cuatro pulsadores normalmente abiertos conectados en serie con una resistencia a 5V, el estado de reposo es 0V, cuando se activa el botón permite el paso de corriente y cambia de estado a 5V.

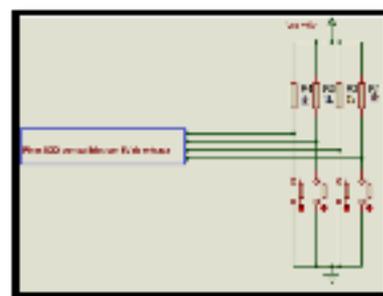


Figura 13. Conexión pulsadores con la placa IOIO
Fuente: Elaborado por Byron Valenzuela

C. Módulo de conversión análogo digital

El módulo de conversión análogo digital del entrenador cuenta con el dispositivo electrónico LM-35 un circuito integrado en forma de transistor, que es un sensor de temperatura.



Figura 14. Sensor de temperatura LM 35
Fuente: LM35. Recuperado de:
<http://www.rapidsignalph.com/shop/lm35-temperature-sensor/>

El LM35 no necesita ningún circuito externo para su funcionamiento (datasheet), la forma de conexión se la puede observar en la siguiente figura.



Figura 15. Diagrama de conexión LM35
Fuente: Texas Instrument. (2000). LM35.

Además de contar con el sensor de temperatura el entrenador cuenta con un potenciómetro o resistencia variable de $10k\Omega$, con el que también se puede realizar prácticas de conversión análogo digital. La conexión del potenciómetro es la siguiente.

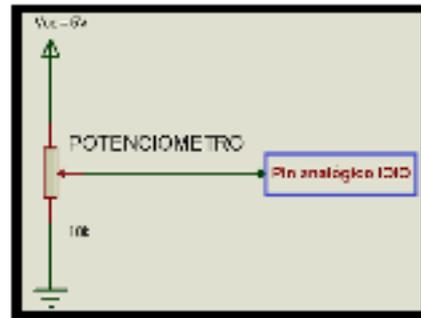


Figura 16. Diagrama de conexión potenciómetro
Fuente: Elaborado por Byron Valenzuela

D. Módulo de comunicación serial

Este módulo consta de un circuito integrado Max232 que permite la comunicación serial con la IOIO, este integrado es capaz de convertir voltajes de nivel RS232 (3V a 15V en uno lógico y -3V a -15V en cero lógico), a voltajes TTL (5V en uno lógico y 0V en cero lógico) y viceversa.

El entrenador cuenta con un módulo Max232 del fabricante Mikroelektronika el módulo es Max232 Board. A continuación se presenta dicho módulo que servirá de interfaz de comunicación entre la PC y la IOIO.



Figura 17. Max232 Board
Referencia: Max232 Board.
<http://www.mikroe.com/add-on-boards/communication/max232/>

E. Módulo ZigBee

Este es uno de los módulos adicionales al entrenador, es decir este módulo no forma parte de la placa central del entrenador, pero tiene su sócalo para

conectarlo fácilmente y hacer uso del mismo.

El entrenador es un equipo de laboratorio que será usado para motivos académicos por tal razón no se necesita de muchas distancias de operación, ni considerable potencia de transmisión, es así que el entrenador cuenta con el módulo XbeeS1®. Y la forma de conexión es la siguiente.

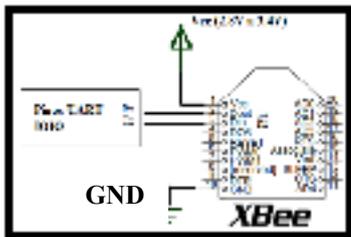


Figura 18. Conexión Xbee® con la IOIO
Referencia: Elaborado por Byron Valenzuela

F. Modulo Bluetooth

El dispositivo escogido para esta sección es el módulo Bluetooth2 Click fabricado por la empresa Mikroelektronika, este dispositivo puede alcanzar una cobertura de hasta 1000m, ofrece un bajo consumo de energía, e interfaz Uart que permite la comunicación con los dispositivos exteriores.

BlueTooth2 Stick se comunica con la IOIO mediante la comunicación Uart, y la forma de conexión con la IOIO es la siguiente.

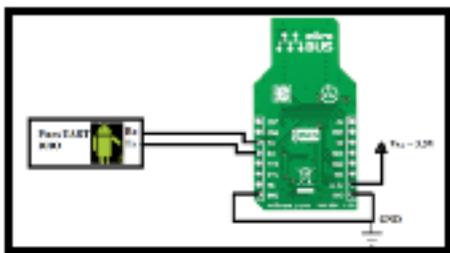


Figura 19. Módulo Bluetooth2 Click
Referencia: Bluetooth2 Click. Recuperado <http://www.mikroe.com/click/bluetooth2/>

G. Modulo GPS

El módulo GPS adoptado para esta sección es el dispositivo GPS2 Click de mikroelectronica, es un equipo compacto que se adapta al sócalo mikroBus, este módulo soporta navegación, localización, y aplicaciones industriales como GPS C/A, S-BAS, y A-GPS, es compatible con comunicación UART, SPI, I2C configurable mediante jumpers o puentes en la placa, ofrece bajo consumo de potencia y alto rendimiento.

El dispositivo GPS2 Click interactúa con la IOIO a través de la interfaz de comunicación serial, debido a que los niveles UART no son TTL es necesario utilizar un amplificador operacional para el uso del GPS y la forma de conexión es la siguiente.

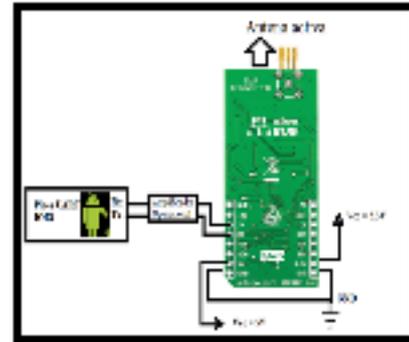


Figura 20. Conexión GPS2 Click con la IOIO
Fuente: Elaborado por Byron Valenzuela

H. Módulo Wi-Fi

El dispositivo Wi-fi usado para esta sección es el equipo WiFly RN-XV fabricado por Sparkfun; WiFly RN-XV es una solución compacta para añadir comunicación Wi-fi bajo el estándar IEEE 802.11 en los diseños electrónicos. Este dispositivo tiene las siguientes características: posee una antena integrada capaz de alcanzar 100m de distancia, 8

pinos de propósito general, 3 conversores análogo digitales, y una interfaz de comunicación UART.

La conexión se muestra en la siguiente figura.

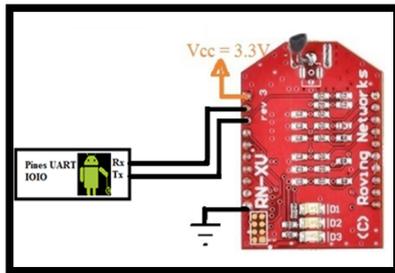


Figura 21. Conexión Wi-Fly con la IOIO
Fuente: Elaborado por Byron Valenzuela

I. Módulo I2C

Este dispositivo es otro de los módulos adicionales al entrenador, el equipo I2C escogido para esta sección es el sensor de temperatura TMP-102, es un sensor creado por la compañía Sparkfun Electronics, usa un conversor análogo digital de 12 bits con una resolución de 0.0625°C , puede medir temperaturas desde -25°C a $+85^{\circ}\text{C}$, tiene un bajo consumo de potencia $10\mu\text{A}$ en estado activo, $1\mu\text{A}$ en estado pasivo, y una alimentación desde 1.6V a 3.3V .

La conexión del sensor de temperatura con la IOIO se muestra en la siguiente figura.

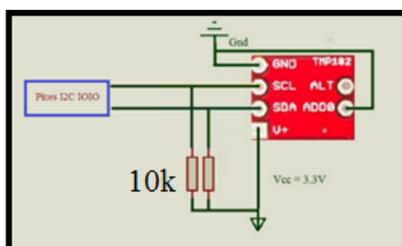


Figura 22. Conexión del módulo I2C con la IOIO
Fuente: Elaborado por Byron Valenzuela

J. Fuente de alimentación

Este bloque del sistema es el encargado de suministrar la energía necesaria para que todos los módulos del entrenador puedan trabajar, esta sección a su vez se divide en dos fuentes de alimentación totalmente independientes la una se encarga de alimentar a el entrenador en sí y cuenta con varios pines de salida donde se puede alimentar los circuitos externos, la otra es la encargada de energizar a los módulos adicionales que cuenta el entrenador como son GSP, Bluetooth, Wi-fi, I2C, Xbee.

La fuente de alimentación del entrenador está formada principalmente por un transformador de 120VCA en 24V de corriente alterna CA a 1 amperio, luego pasa por un rectificador de onda completa que convertirá la corriente alterna en corriente continua CD formado por diodos rectificadores 1N4001 capaz de soportar 50VCA a 1 amperio (A) que es lo que la fuente necesita, seguido de esto se compone de un capacitor de $2200\mu\text{F}$ y un capacitor de $47\mu\text{f}$ (datasheet) en paralelo con el objetivo de eliminar los picos de voltaje aún existentes. A la salida de este proceso ya se cuenta con 12VDC .

A continuación viene el proceso de regulación, los voltajes requeridos son de 12VDC , 5VDC , y 3.3VDC ; para esto se hace uso de los reguladores LM7812, LM7805, LM1117V33 respectivamente acompañado de un capacitor de $33\mu\text{f}$ en paralelo que determina la hoja de datos del integrado.

El siguiente diagrama muestra la fuente de alimentación del entrenador.

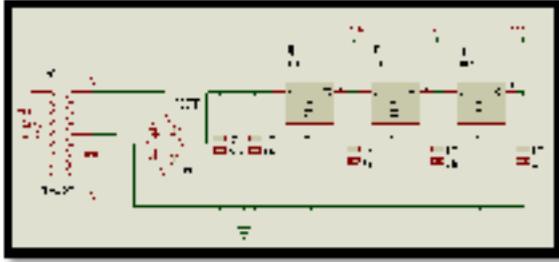


Figura 23. Diagrama de la fuente de alimentación del entrenador
Referencia: Elaborado por Byron Valenzuela

La fuente de alimentación para los módulos adicionales es muy similar a la principal la única diferencia es que cuenta con una salida de voltaje variable de 0VDC a 12VDC; de la misma manera esta cuenta con un transformador de 120V en 24V de corriente alterna CA a 1 amperio, un rectificador de onda completa formado por los mismos diodos rectificadores de la fuente principal, el proceso de filtrado cuenta con los mismos capacitores de 2200 μ F y de 47 μ F.

Para el proceso de regulación se tiene los mismos voltajes anteriores y adicionalmente el voltaje variable 0V-12V y para esto se usa el regulador LM317 que es un regulador capaz de emitir un voltaje variable de 1.2VDC a 37VDC con una corriente de 1.5A (datasheet), el fabricante impone su propia configuración para el trabajo de este integrado; la configuración tiene un capacitor de 1 μ F en paralelo con la salida, una resistencia de 1k Ω entre los pines VO (voltaje de salida) y VADJ (voltaje variable) y un potenciómetro o resistencia variable de 5k Ω entre los pines VADJ y GND.

Adicional a la fuente se tiene dos diodos led con su respectiva resistencia de protección que actúan como indicadores visuales el diodo 1 uno se activa cuando la fuente se enciende, el diodo led 2 se

enciende directamente proporcional al voltaje variable que está saliendo de la fuente es decir que mientras más voltaje exista el led emitirá más luz.

A continuación se tiene un diagrama de la fuente de alimentación para los módulos adicionales.

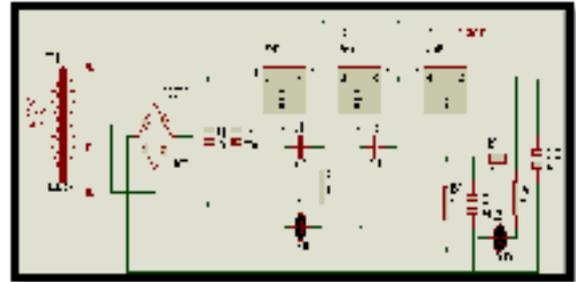


Figura 24. Fuente de alimentación variable
Referencia: Elaborado por Byron Valenzuela

IV. PRÁCTICAS REALIZADAS CON EL ENTRENADOR ANDROID

Por cada uno de los módulos del entrenador se realizó una práctica con el propósito de demostrar el funcionamiento tanto de la placa de entrenamiento como de la aplicación cargada en el dispositivo móvil Android. A continuación se enumeran los enunciados de las prácticas realizadas.

1. Desarrollar un juego de luces secuenciales de 8 led's de izquierda a derecha, conectados a los pines de salida digital no tolerantes a 5V de la IOIO, con un retardo de 500ms en cada transición.
2. Crear una aplicación que permita verificar los pulsos provenientes de tres pulsadores, adelantar, retroceder y encerrar respectivamente. Para esta aplicación hacer uso de la entrada digital a 5V.

3. Elaborar un contador decimal 0-99, con alarma sonora en un intervalo de diez números
4. Crear un letrero dinámico de izquierda a derecha con la frase "HOLA IoIo" mediante la matriz de led's.
5. Elaborar un medidor de voltaje y temperatura, mediante el potenciómetro y el sensor de temperatura que forman parte del módulo analógico digital del entrenador.
6. Elaborar una aplicación que controle la luminosidad de un led conectado a la IOIO mediante PWM.
7. Desarrollar un lector de temperatura mediante un dispositivo que trabaja bajo el protocolo de comunicación I2C.
8. Desarrollar una aplicación que permita leer la temperatura de un sensor LM35 de forma inalámbrica mediante Bluetooth.
9. Realizar un medidor de distancia mediante una red punto a punto con módulos ZigBee, el transmisor envía la distancia del sensor, el receptor recibe la información de la distancia y en caso de que la distancia sea menor a 5m se emite una alarma visual.
10. Realizar una aplicación que reciba la información de ubicación del GPS y grafique el punto recibido en un mapa.
11. Desarrollar una aplicación que permita monitorear remotamente desde la Pc la humedad de suelo mediante un módulo Wi-Fi; en caso que se necesite riego

enviar una alarma visual de una bombilla a 110V.

V. CONCLUSIONES

Al culminar este proyecto de titulación el estudiante de la carrera de ingeniería en Electrónica y Redes de Comunicación podrá familiarizarse más con el sistema operativo Android y realizar varias prácticas electrónicas con los diferentes dispositivos con los que el entrenador cuenta.

Este trabajo ayudará a muchos estudiantes que les gusta el campo de la electrónica y el mundo Android a desarrollar aplicaciones primeramente básicas, conocer los parámetros de operación de la placa y el desarrollo de software Android, para luego manejar módulos adicionales y con esto realizar proyectos que permitan el avance tecnológico de la Universidad y del país.

Android es uno de los sistemas operativos más usados en los dispositivos móviles en la actualidad, al ser de código abierto es libre esto quiere decir que cualquier persona puede desarrollar aplicaciones para su equipo móvil.

Android cuenta con el soporte oficial y herramientas necesarias para las personas que se interesan en desarrollar las aplicaciones, esto quiere decir que brinda paso a paso capacitación, ejemplos y tutoriales para adentrarse en el apasionante mundo del desarrollo de aplicaciones.

La placa de desarrollo electrónico IOIO cuenta con un firmware incluido que facilita el uso y la comunicación ya sea por cable o vía Bluetooth, además cuenta con librerías para el uso de los diferentes módulos que la junta posee como son:

módulo de entrada y salida digital, comunicación serial, PWM, I2C, y conversores análogo digitales.

Al finalizar este proyecto se logró comunicar la IOIO con varios módulos adicionales, con el propósito de que el estudiante tenga la capacidad de interactuar con los distintos dispositivos electrónicos como Xbee, Wi-Fi, Bluetooth, GPS y pueda controlarlos desde su equipo móvil.

El lenguaje de programación oficial para desarrollar aplicaciones Android es Java un lenguaje libre orientado a objetos y el IDE es Eclipse de igual forma un entorno de desarrollo gratuito que trabaja bajo la plataforma de java.

Al finalizar las prácticas con la IOIO vemos que es obligatorio incluir los permisos de uso Bluetooth e Internet dentro de la aplicación a desarrollarse, además se debe adjuntar las librerías IOIO para poder acceder a todos los métodos y librerías que controlan los módulos electrónicos de la misma.

La implementación del entrenador electrónico para dispositivos Android permitió afianzar los conocimientos adquiridos en las aulas de clase debido a que el sistema abarca conceptos tanto de redes como electrónica.

Todas las aplicaciones realizadas en esta obra son prácticas que el estudiante de Ingeniería en Electrónica y Redes de Comunicación está en la capacidad de realizarlas con un previo conocimiento adquirido en las aulas.

VI. RECOMENDACIONES

Antes de realizar las prácticas es muy importante conocer la estructura física e interna que la IOIO posee para no causar daños parciales o totales dentro de la placa, al circuito externo, módulos adicionales con los que estemos trabajando e incluso al dispositivo móvil al que está conectado.

De igual manera previa al uso de los distintos módulos adicionales el estudiante debe estar familiarizado con el dispositivo, conocer los parámetros de funcionamiento tanto eléctricos como de configuración con el objetivo de impedir que los módulos sufran averías.

Luego de haber desarrollado la aplicación Android IOIO, probar el funcionamiento de la aplicación conectando la placa al móvil vía cable USB y mediante Bluetooth y así comprobar el funcionamiento a través de las dos conexiones disponibles.

Al momento de realizar las prácticas con el módulo GPS, es muy conveniente comprobar el funcionamiento de la aplicación en un ambiente libre de obstáculos para así evitar interferencias al momento de recibir los datos satelitales de longitud y latitud.

En el presente trabajo se usó la IOIO como dispositivo conectado a un terminal Android, la versión IOIO OTG también es capaz de trabajar como un host conectado a un computador, se recomienda continuar la investigación y realizar prácticas con la IOIO conectado al ordenador.

Adicionalmente a las prácticas presentadas, el estudiante una vez que haya dominado el tema puede incursionar en el

desarrollo de aplicaciones electrónicas con Android; si la aplicación es necesaria e innovadora subirla al App Store y venderla obteniendo así una remuneración por el trabajo realizado.

RECONOCIMIENTOS

Se expresa un especial reconocimiento al Ing. Msc. Edgar Maya director del mismo por el apoyo y la colaboración brindada para desarrollar este trabajo.

REFERENCIAS

- Castillo, D. (2012). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ALARMA COMUNITARIA A BASE DE MÓDULOS INALÁMBRICOS UTILIZANDO TECNOLOGÍA ZIGBEE. Ibarra.
- Clanar Internacional. (s.f.). Internet y redes inalámbricas. Arequipa: Clanar.
- Collaguazo, G. (2009). Sistemas Basados en Microprocesadores. Ibarra.
- Correia, P. (2002). Guía práctica del GPS. Barcelona: Marcombo.
- Creative Commons. (s.f.). Arduino. Obtenido de <http://arduino.cc/en/Main/CopyrightNotice>
- Daniel Benchimol. (2011). Microcontroladores. Buenos Aires: DALAGA S.A.
- Digi International, Inc. (2009). XBee®/XBee PRO® RF Modules. New York.
- Dignani, J. P. (2011). Análisis del protocolo ZigBee.
- GitHub. (2008). Analog Input. Obtenido de <https://github.com/ytai/arduino/wiki/Analog-Input>
- Granadino, C., & Suárez, J. (s.f.). El bus I2C. Chile: Universidad Técnica Federico Santa María.
- IOIO for Android. (s.f.). Obtenido de <https://www.sparkfun.com/products/10748>
- Jara, P., & Nazar, P. (s.f.). Estándar IEEE 802.11 X de las WLAN. Buenos Aires: Edutecne.
- Monk, S. (2012). Making Android Accessories With IOIO (Vol. I). O'Reilly Media.
- Pérez, E. L. (s.f.). Curso de Redes de Microcontroladores PIC (PROTOCOLO SPI). México: Ingeniería en Microcontroladores.
- Pérochon, S. (2012). Android Guía de desarrollo de aplicaciones para smartphones y tabletas. Barcelona: Ediciones ENI.
- Quectel. (2011). L30 Quectel GPS Engine. Shanghai: Quectel.
- Quectel. (2013). Quectel L30 Compact GPS Module.
- Raúl Esteve Bosch, J. F. (2005). Fundamentos de electrónica digital. Valencia.
- RobotFreak. (2008). IOIO-Rover. Obtenido de <http://letsmakerobots.com/node/33968>
- ROVING NETWORKS. (2011). RN-XV Data Sheet. Arizona: ROVING NETWORKS.
- sgoliver.net foro. (s.f.). Estructura de un proyecto Android. Obtenido de <http://www.sgoliver.net/blog/?p=1278>
- Ucontrol. (2008). Matrices de LEDs. Ucontrol Electrónica General Pic's en particular, 68.
- Universidad Politécnica de Valencia. (Abril de 2013). Elementos de un proyecto Android. Obtenido de <http://www.androidcurso.com/index.php/recursos-didacticos/tutorialesandroid/31-unidad-1-vision-general-y-entorno-de-desarrollo/148-elementos-de-un-proyecto-android>.
- Valverde Rebaza, J. C. (2007). El Estándar Inalámbrico ZigBee. Trujillo: Perú.



Edgar A. Maya O.

Nació en Ibarra provincia de Imbabura el 22 de abril de 1980. Ingeniero en Sistemas Computacionales, Universidad Técnica del Norte – Ecuador en 2006. Actualmente es docente en la carrera de

Ingeniería en Electrónica y Redes de Comunicación en la Universidad Técnica del Norte, Ibarra – Ecuador, Maestría en Redes de Comunicación, Pontificia Universidad Católica del Ecuador, Quito – Ecuador.



Byron R. Valenzuela M.

Nació en la ciudad de Ibarra-Ecuador, el 20 de Mayo de 1990. Realizó sus estudios primarios En la Escuela Santa “La Merced” y sus Estudios secundarios en el Colegio Nacional “Teodoro Gómez de la Torre”, donde finalizó

en el año 2007, obtuvo el título de Bachiller en Ciencias Especialización Físico Matemático. Actualmente, es egresado de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte .