



FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS
TÉCNICO EN INGENIERÍA DE HARDWARE



TEMA: “IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO CON
TECNOLOGÍA ANDROID PARA EL TEMPORIZADO DE LAS ACTIVIDADES
DEL COMPLEJO EDUCATIVO MARCOS OCHOA”.

TRABAJO DE GRADUACIÓN PRESENTADO POR:

GRANADOS REYES, NERY DE JESÚS
OCHOA ARIAS, FRANCISCO ERNESTO
RIVERA AVILES, JONNATTAN SICFREDO

PARA OPTAR AL GRADO DE:

TÉCNICO EN INGENIERÍA DE HARDWARE.

SEPTIEMBRE, 2014

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA.

Página de Autoridades

ING. NELSON ZÁRATE SÁNCHEZ
RECTOR

LIC. JOSÉ MODESTO VENTURA
VICERRECTOR ACADÉMICO

ING. FRANCISCO ARMANDO ZEPEDA
DECANO

JURADO EXAMINADOR

ING. JOSÉ RAMÓN CORNEJO
PRESIDENTE

ING. JOSÉ MAURICIO RIVERA
PRIMER VOCAL

ING. OMAR OTONIEL FLORES
SEGUNDO VOCAL

SEPTIEMBRE, 2014
SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA

ACTA DE EXAMEN PROFESIONAL

HABIÉNDOSE REUNIDO EL JURADO CALIFICADOR INTEGRADO POR:

Ing. José Mauricio Rivera Mejía, Ing. Omar Otoniel Flores, Ing. José Ramón Cornejo Hernández, a las 12:30m. del día Miércoles, 25 de junio de dos mil catorce.

Y LUEGO DE HABER DELIBERADO SOBRE EL EXAMEN PROFESIONAL DE LOS ALUMNOS:

1-Nery de Jesus Granados Reyes	Carnet 28-5549-2011
2-Francisco Ernesto Ochoa Arias	Carnet 28-4489-2012
3-Jonnattan Sicfredo Rivera Aviles	Carnet 28-1235-2008

QUIENES PRESENTARON DEFENSA DE SU TRABAJO DE GRADUACION TITULADO:

"Implementación de un Sistema Domótico con Tecnología Android para el temporizado de las actividades del complejo Educativo Marcos Ochoa"

PARA OPTAR AL GRADO DE:

TÉCNICO EN INGENIERIA DE HARDWARE

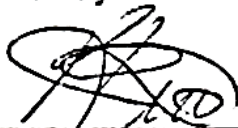
Y DEL CUAL TAMBIEN EVALUARON LOS CONOCIMIENTOS RELACIONADOS CON EL TEMA DEL MISMO. POR LO QUE ESTE JURADO RESUELVE DECLARAR EL EXAMEN COMO:

Aprobado

YA QUE CUMPLE CON LOS REQUISITOS ESTABLECIDOS EN EL REGLAMENTO DE GRADUACION DE LA UNIVERSIDAD.

San Salvador, 25 de junio de 2014.

F.



PRIMER VOCAL

Ing. José Mauricio Rivera Mejía

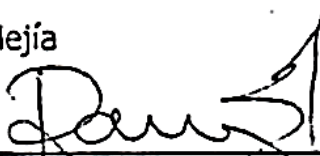
F.



SEGUNDO VOCAL

Ing. Omar Otoniel Flores

F.



PRESIDENTE

Ing. José Ramón Cornejo Hernández

Índice

Introducción.....	i
Capítulo I: Situación Actual.....	1
1.1 Situación Problemática.....	1
1.2 Enunciado del Problema.....	3
1.3 Justificación.....	3
1.4 Objetivos.....	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos.....	4
1.5 Delimitaciones.....	5
1.5.1 Delimitación Geográfica.....	5
1.5.2 Delimitación Temporal.....	5
1.5.3 Delimitación Organizacional.....	6
1.6 Alcances (Entregables).....	7
1.7 Estudio de Factibilidad.....	8
1.7.1 Factibilidad Económica.....	8
1.7.2 Factibilidad Técnica.....	10
1.7.3 Análisis General de los Componentes.....	12
1.8 Carta de Aceptación.....	13
Capítulo II: Documentación Técnica.....	14
2.1 Marco Teórico de Referencia.....	14
2.1.1 Sistema Embebido.....	14
2.1.2 Características de un Sistema Embebido.....	15
2.1.3 Hardware de sistemas embebidos:.....	16
2.1.4 Esquema de bloques funcionales de un sistema embebido:.....	16
2.1.5 Microprocesadores:.....	17
2.1.6 Microcontrolador:.....	17
2.1.7 Sensores:.....	17
2.1.8 Memoria:.....	18
2.1.9 Organización de memoria:.....	19

2.1.10 Tipos de memoria.....	19
2.1.11 Memoria RAM:	20
2.1.12 Memoria ROM:	21
2.1.13 Comunicaciones:	22
2.1.14 Presentación:	23
2.1.15 Actuadores:.....	23
2.1.16 Pines de E/S analógicos y digitales:	24
2.1.17 Reloj:	24
2.1.18 Módulo de alimentación (Power):	24
2.1.19 Arduino.....	25
2.1.20 ¿Qué es Wiring?	26
2.1.21 ¿Qué es Processing?	26
2.1.22 Sistema Android.	27
2.1.23 Las primeras versiones Android:.....	28
2.1.24 Android 1.0 Nivel de API 1 (septiembre 2008)	28
2.1.25 Android 1.1 Nivel de API 2 (febrero 2009)	29
2.1.26 Cup cake	29
2.1.27 Donut	30
2.1.28 Éclair	30
2.1.29 Froyo	31
2.1.30 Honeycomb	33
2.1.31 IceCreamSandwich.....	34
2.1.33 Kit Kat	36
2.1.34 Comunicación entre Android y Arduino:	37
2.1.35 Domótica:	37
2.2 Marco Teórico de Solución.	38
2.3 Marco Teórico Conceptual.	42
2.4 Documentación Técnica.....	51
Capítulo III Desarrollo de la Solución.	55
3.1 Propuesta de Solución.....	55
3.1.1 Firmware de control general del sistema.....	57

3.1.2 Flujograma de funcionamiento de sistema de temporizado.	58
3.1.3 Diagrama Electrónico de Temporizador.	82
3.1.4 Aplicación Móvil (APK)	82
3.1.5 Modelo ADDIE	86
3.2 Conclusiones.	96
3.3 Recomendaciones.	97
3.4 Referencias.....	98
Anexo.....	99
A1.Matriz de Congruencia	99
A2. A continuación se exponen imágenes de la instalación del proyecto.....	100

Introducción.

Actualmente en los centros educativos se necesita de un sistema de temporizado que cumpla con exactitud eficiencia y puntualidad la activación del timbre para las actividades diarias y la jornada educativa, por lo cual surge la necesidad de tener un buen sistema de temporizado para que cumpla con dicha función de timbrado. Debido a esta necesidad como grupo de trabajo se optó por realizar la “Implementación de un sistema Domótico con tecnología Android para el temporizado de las actividades del Complejo Educativo Marcos Ochoa.” Para solventar esta problemática y darle un mayor confort al encargado de dicha actividad y así el director podrá realizar todas sus actividades diarias sin la preocupación de cumplir con esta responsabilidad. Ya que en esta institución no dispone de un sistema de timbrado para anunciar las diferentes actividades de entrada, salida, recreos, etc.

En el primer capítulo se plantea la situación actual del problema que dicha institución tiene y a su vez se justifica por qué realizar dicho proyecto, presentando los objetivos que se pretenden cumplir. Otro de los temas de importancia son las de la factibilidad económica y técnica los cuáles darán a conocer que componentes electrónicos son más factibles para la implementación del proyecto.

En el segundo capítulo se mencionan todas las áreas en las cuales está basado el proyecto siendo el marco teórico de referencia el cual contiene toda la

información sobre los diferentes temas en los cuales está basado el proyecto mencionando ¿Qué es un Sistema Embebido?, ¿Qué son los microcontroladores?, plataforma Arduino y las Versiones de Android. Además se encuentra el marco teórico de solución en el cual se muestra como quedara planteado y a su vez instalado el proyecto y también se habla del funcionamiento de cada componente electrónico. Así mismo se presenta el marco teórico conceptual en el cual se definen diferentes conceptos.

Para finalizar el capítulo 2 se da a conocer la documentación técnica de cada componente electrónico donde se muestran las características técnicas.

En el tercer capítulo de este documento se muestra la propuesta de solución que fue elaborada en grupo para dar a conocer la solución que se le dará a la problemática presentada en el Complejo Educativo Marcos Ochoa, mencionando los componentes que se utilizaran y el funcionamiento que tendrá cada uno de ellos, también se dará a conocer el flujograma, y esquema electrónico realizado en fritzing el cual muestra cómo estará conectado los componentes, El firmware que controla el sistema siendo este la parte fundamental para que funcione correctamente.

Para terminar se da a conocer las conclusiones como grupo y una serie de recomendaciones para que el sistema tenga una extensa durabilidad, así concluir que se espera que la lectura de este documento sea de agrado para el lector.

Capítulo I: Situación Actual.

1.1 Situación Problemática.

En la villa de Tapalhuaca departamento de la Paz, El Salvador se encuentra el “Complejo Educativo Marcos Ochoa”, Escuela pública de Parvularia, Educación Básica y Media, la cual dispone de 14 aulas, una para sala de maestros y la otra para la dirección y administración y las 12 restantes son utilizadas para impartir clases. Cuentan con un grupo de 18 maestros, quienes son los encargados de darles clases a 405 alumnos, divididas en dos turnos, matutino de 7:15 am a 12:00 pm y vespertino de 1:45 pm a 5:00 pm. En la institución educativa se brinda una educación desde parvularia hasta tercer año de bachillerato, siendo un bachillerato único en opción contaduría, sin secciones, y como es normal en toda institución pública y privada surge la necesidad de cumplir con horarios establecidos, en el cual cabe mencionar el toque del timbrado para reuniones generales antes de clase, para los recesos y para anuncio de entrada y salida, siendo todo esto algo muy importante que no se puede obviar, pero lamentablemente en esta institución no disponen de un buen sistema que cumpla con estas tareas, pues antes disponían de un timbre manual, sufrió desperfectos y ahora no tienen más que una campana pequeña en malas condiciones, la cual tiene que ser tocada por el director o secretaria.

Por otro lado, los problemas que surgen a raíz de esta situación es que en ocasiones se les olvida estar pendientes del reloj debido a la gran demanda de trabajo. De esta forma se describen las diferentes tareas que se realizan en dicha institución y en debidas circunstancias la secretaria o director tiene que correr por habérsele pasado el tiempo o estar solventando algunos problemas y esto podría causarle una caída y sufrir una fractura en uno de sus miembros, y también se pierden minutos porque en ocasiones el director se encuentra ocupado atendiendo visitas, ya sean de parte del Ministerio de Educación o padres de familia que llegan a consultar por sus hijos, causando perdida de horas de clases y de recreo, y la secretaria como es normal pasa manejando todo lo administrativo de la institución e incluso atendiendo llamadas y es por ello, que también resulta incómodo para ella tener que ocuparse de dicha tarea; por lo tanto, esto genera una gran incomodidad para los alumnos y maestros pues en muchas ocasiones se pierden minutos de receso y a la misma vez es incómodo para el director recibir quejas de parte de los alumnos.

Tampoco se puede obviar, que como todo director, hay días en los cuales tiene que asistir a reuniones fuera de la institución, por lo que la secretaria queda encargada y se vuelve más complicado, y los maestros no pueden ayudar pues se encuentran ocupados impartiendo clases.

Y para finalizar, es bueno dejar claro que si no se realiza algo para que se resuelva este problema en la institución, siempre se seguirá dando dicha

problemática y los principales afectados serán los alumnos, y secundariamente el director y secretaria que deben cumplir con esta labor diariamente mañana y tarde durante toda la semana.

1.2 Enunciado del Problema.

¿Es necesario realizar una propuesta tecnológica de última generación que sea rentable y funcional para que cumpla y facilite las labores de temporizado del timbre de entrada y salida en el Complejo Educativo Marcos Ochoa ?

1.3 Justificación.

Es necesario que el “Complejo Educativo Marcos Ochoa”, escuela pública, tenga un sistema Domótico, que facilite la labor del Temporizado del timbre que regula las actividades del Complejo Educativo. En dicha escuela, el proyecto ayudará al director que cumpla con sus labores diarias de la manera más tranquila y eficiente, ya que con esto no tendría que salir corriendo, ni dejar en espera a las visitas que llegan a la institución, y en caso de salidas no tener que dejar encargado a ningún maestro. Así mismo. se cumplirán los horarios a cabalidad tales como los toques de cambio de clases, entradas y salidas de los recreos y, entrada y salida de la Institución.

Algunos beneficios que el Complejo Educativo obtendría son:

- Un mayor confort para el “Complejo Educativo Marcos Ochoa” en el área de timbrado.

- Nuevas tecnologías instaladas en el Complejo Educativo.
- Ya no tendrán que dejar a una persona encargada para las labores de timbrado ya que este proyecto funcionará automáticamente.
- Accesibilidad económica y calidad en los servicios de mantenimiento.

1.4 Objetivos.

1.4.1 Objetivo General.

Implementar un Sistema Domótico con tecnología Android para el temporizado de las actividades del “Complejo Educativo Marcos Ochoa” y así brindar un mayor confort a la institución y al personal que la conforman.

1.4.2 Objetivos Específicos.

- Elaborar un circuito electrónico con la plataforma Arduino que sea capaz de llevar el temporizado de las actividades del “Complejo Educativo Marcos Ochoa” y a la misma vez active el timbre según las horas que se programe.
- Elaborar un firmware o programa que permita que el circuito funcione adecuadamente, y pueda recibir información enviada por medio de una aplicación de sistema Android, instalada en un teléfono móvil o Tablet con dicho sistema.
- Integrar en un solo circuito electrónico el Hardware y Software para la realización del proyecto final y funcional.

1.5 Delimitaciones

1.5.1 Delimitación Geográfica

A realizarse en el “Complejo Educativo Marcos Ochoa” Villa de Tapalhuaca , departamento de La Paz, El Salvador.

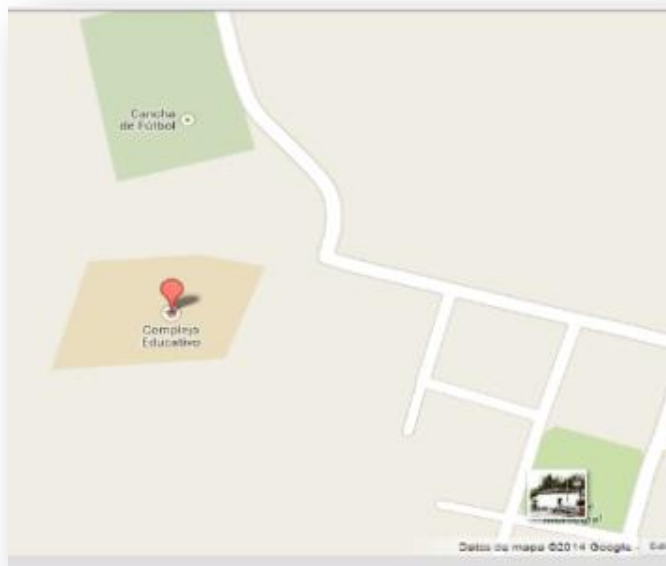


Figura 1.1 Mapa de ubicación del Complejo Escolar Marcos Ochoa.

Fuente: Google Maps.

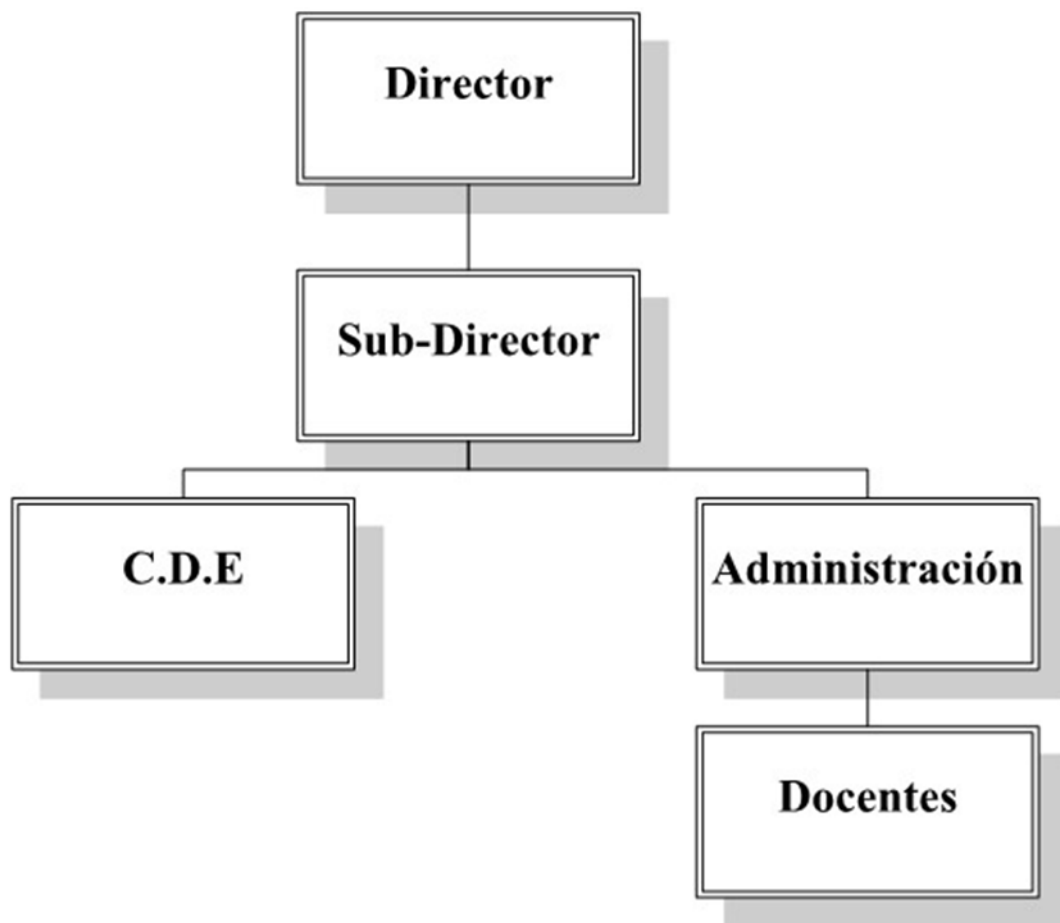
1.5.2 Delimitación Temporal

Programado para el ciclo académico 01-2014 que comprende de Marzo - Junio 2014.

1.5.3 Delimitación Organizacional

En el siguiente Organigrama se muestra como está estructurado el “Complejo Educativo Marcos Ochoa”.

Figura 1.2 Esquema que muestra la estructura organizativa del “Complejo Educativo Marcos Ochoa”.



1.6 Alcances (Entregables)

Durante la realización de este proyecto se han propuesto diferentes alcances que se convierten en promesas para el desarrollo de este proyecto y su finalización; las promesas se encuentran en el siguiente cuadro juntamente con los productos.

PROMESAS	PRODUCTOS
Elaborar un circuito electrónico el cual tenga diversas tecnologías implementadas tales como la plataforma Arduino y la tecnología Android, capaz de llevar las labores de temporizado en dicha institución.	Un circuito electrónico encargado para el temporizado de las actividades del “Complejo Educativo Marcos Ochoa”.
Elaborar un programa que controle todo el circuito electrónico para así poder cumplir con las labores de temporizado en la institución	Software compilado funcional el cual controle todos los dispositivos electrónicos para que puedan cumplir su función en el sistema electrónico
Integrar en un solo circuito electrónico el hardware y el software para la realización del sistema de temporizado juntamente con el sistema operativo Android	Proyecto funcional con el programa compilado y una aplicación Android con la cual podrán modificar los horarios en los cuales quieren que suene el timbre.

1.7 Estudio de Factibilidad.

Es necesario el estudio de factibilidad técnica y económica para el desarrollo e implementación de este dispositivo electrónico, para verificar que componentes son más recomendables en cuanto a lo económico, funcionamiento y características. Para ello se cotizaran precios y marcas de los principales dispositivos electrónicos que conforman el proyecto final.

1.7.1 Factibilidad Económica

En este apartado se hacen todas las comparaciones de los precios de los diferentes componentes que utilizan para el desarrollo del dispositivo electrónico, con el objetivo de ver la factibilidad económica en cada componente.

A continuación se presentarán una serie de tablas, en las que se muestran los precios y lugares de cotización de los diferentes dispositivos electrónicos a utilizar en el proyecto. Con el objeto de apreciar la factibilidad económica de cada elemento.

MICROCONTROLADOR			
Opción	PIC16F84A	PIC16F88	ATMEGA328
Precio de cotización	\$4.75	\$2.46	\$3.39
Lugar de cotización	www.mouser.com	www.mouser.com	www.mouser.com

Por el precio accesible y tomando en cuenta el aprendizaje adquirido en la Universidad Tecnológica de El Salvador y previo a proyectos anteriores elaborados se utilizara el microcontralador Atmega 328.

CIRCUITO INTEGRADO : Reloj de tiempo real			
Opción	74LS121	LM555	RTC DS1307
Precio de cotización	\$3.22	\$0.64	\$7.00
Lugar de cotización	es.aliexpress.com	es.aliexpress.com	www.ingnac.com

Por calidad y su proveedor garantizado se optó por RTC DS1307.

MÓDULO DE COMUNICACIÓN Bluetooth			
Opción	HC-06	VS1838	APC220
Precio de cotización	\$15.00	\$8.00	\$36.30
Lugar de cotización	www.ingnac.com	www.ingnac.com	www.tuelectronica.es

Por su precio y garantía del proveedor se optó por el módulo HC-06.

1.7.2 Factibilidad Técnica

Aquí se hace el análisis técnico para cada componente electrónico, viendo las características de cada uno para elegir el que es mejor en cuanto en rendimiento y económico.

MICROCONTROLADOR							
SE REQUIERE UN PORCENTAJE DE ACEPTACIÓN MÍNIMO DE 90%							
CARACTERÍSTICAS	PONDERACIÓN	PIC16F84A		PIC16F88	ATMEGA 328		
	%	VALOR	%	VALOR	%	VALOR	%
VELOCIDAD	30%	2	60%	2	60%	2	60%
MEMORIA	30%	1	30%	1	30%	2	60%
LENGUAJE DE PROGRAMACIÓN	15%	2	30%	2	30%	2	30%
VIA DE PROGRAMACIÓN	15%	1	15%	1	15%	2	30%
PRECIOS	10%	1	10%	2	20%	2	20%
TOTAL	100%		145%		155%		200%
PORCENTAJE FINAL	TOTAL/2		72,50%		77,50%		100%

VALORES: MALO = 0, MUY BUENO = 1, EXCELENTE =2

Después de observar la tabla anterior se concluye elegir el microcontrolador ATMEGA 328, pues cumple con todas las características necesarias para implementar nuestro proyecto.

CIRCUITO INTEGRADO: RELOJ DE TIEMPO REAL							
SE REQUIERE UN PORCENTAJE DE ACEPTACIÓN MÍNIMO DE 90%							
CARACTERÍSTICAS	PONDERACIÓN	74LS121		LM555		RTCDS1307	
	%	VALOR	%	VALOR	%	VALOR	%
BATERIA	45%	1	45%	2	90%	2	90%
TAMAÑO	45%	1	45%	1	45%	2	90%
PRECIOS	10%	2	20%	2	20%	2	20%
TOTAL	100%		110%		155%		200%
PORCENTAJE FINAL	TOTAL/2		55,00%		77,50%		100%

VALORES: MALO = 0, MUY BUENO = 1, EXCELENTE =2

Como observación a la tabla mostrada sobre diferentes tipos de Reloj, se puede concluir que la mejor opción es el RTC DS1307, pues es el que cumple con las características necesarias para implementar nuestro proyecto.

MÓDULO DE COMUNICACIÓN BLUETOOTH							
SE REQUIERE UN PORCENTAJE DE ACEPTACIÓN MÍNIMO DE 90%							
CARACTERÍSTICAS	PONDERACIÓN	HC-06		VS1838		APC220	
	%	VALOR	%	VALOR	%	VALOR	%
DISTANCIA	30%	2	60%	2	60%	2	60%
TAMAÑO	30%	2	60%	1	30%	1	30%
FRECUENCIA	30%	2	60%	1	30%	2	60%
PRECIOS	10%	2	20%	2	20%	1	10%
TOTAL	100%		200%		140%		160%
PORCENTAJE FINAL	TOTAL/2		100%		60%		18%

VALORES: MALO = 0, MUY BUENO = 1, EXCELENTE =2

Según los valores mostrados anteriormente es más factible utilizar el módulo HC-06, pues es mejor en cuanto a sus características para la implementación del proyecto.

1.7.3 Análisis General de los Componentes

ANÁLISIS GENERAL DE LOS COMPONENTES QUE SE UTILIZARAN EN EL PROYECTO	
PRODUCTO	DESCRIPCIÓN DEL PRODUCTO
MICROCONTROLADOR ATMEGA 328	Utilizaremos este microcontrolador porque es más fácil de encontrar posee una gran demanda en el país y su precio es accesible y además poseemos conocimiento de programación adquirido en nuestra carrera técnica.
RTC DS1307	Utilizaremos este reloj que trabaja en tiempo real porque es más accesible económicamente y su programación es fácil y tiene alta calidad en cuanto a su función.
MODULO BLUETOOTH HC-06	Este módulo viene diseñado para permitir conexión inalámbrica el cual ayudara a que se comunique la aplicación Android con el sistema. Este será utilizado en el proyecto por su factibilidad económica y mejor programación.

1.8 Carta de Aceptación

San Salvador, 25 de febrero de 2014

Lic. Iván Ernesto Ramírez

Director del Complejo Educativo Marcos Ochoa

Nos dirigimos a usted muy cordialmente los alumnos de la carrera Técnico en Ingeniería de Hardware de la Universidad Tecnológica de El Salvador, esperando se encuentre muy bien al lado de sus seres queridos, después de este corto pero sincero saludo pasamos a lo siguiente.

El motivo por el cual emitimos la presente es para solicitar poder desarrollar nuestro trabajo de graduación en las instalaciones del Complejo Educativo Marcos Ochoa de la villa de Tapalhuaca con el Tema “Implementación de un sistema Domótico con tecnología Android para el temporizado de las actividades del Complejo Educativo Marcos Ochoa” orientado al desarrollo de prototipos Domóticos, diseño de software, diseño de hardware y aplicación.

Esperando una respuesta positiva para el desarrollo de dicho proyecto de antemano damos gracias.

Atentamente.

F. _____

Francisco Ernesto Ochoa

Carnet: 28-4489-2012

F. _____

Nery de Jesús Granados

Carnet: 28-5549-2011

F. _____

Jonnattan Sicfredo Rivera

Carnet: 28-1235-2008

Aprobado por:

F. _____

Lic. Iván Ernesto Ramírez

Director del C.E.M.O

F. _____

Sello

Capítulo II: Documentación Técnica

2.1 Marco Teórico de Referencia.

2.1.1 Sistema Embebido

Existen numerosas definiciones de sistemas embebidos, algunas son:

- “Un sistema embebido es cualquier dispositivo que incluye un computador programable, pero en sí mismo no es un computador de propósito general”. (Pérez, 2009)
- “Un sistema embebido es un sistema electrónico que contiene un microprocesador o microcontrolador; sin embargo, no pensamos en ellos como un computador”. (Pérez, 2009)
- “Las personas usan el término sistema embebido para referirse a cualquier sistema de cómputo escondido en algún producto o dispositivo”. (Pérez, 2009)
- “Un sistema embebido es un sistema cuya función principal no es computacional, pero es controlado por un computador integrado. Este computador puede ser un microcontrolador o un microprocesador. La palabra embebido implica que se encuentra dentro del sistema general, oculto a la vista, y forma parte de un todo de mayores dimensiones”. (Pérez, 2009)

Un sistema embebido posee hardware de computador junto con software embebido como uno de sus componentes más importantes. Es un sistema computacional dedicado para aplicaciones o productos. Puede ser un sistema

independiente o parte de un sistema mayor, y dado que usualmente su software está embebido en ROM (ReadOnlyMemory) no necesita memoria secundaria como un computador. Un sistema embebido tiene tres componentes principales:

1. Hardware. Son todos los componentes físicos de un Sistemas Embebidos.
2. Un sistema operativo que permite supervisar la(s) aplicación(es), además de proveer los mecanismos para la ejecución de procesos. En muchos sistemas embebidos es requerido que el sistema operativo posea características de tiempo real.
3. Un software primario o aplicación principal. Este software o aplicación lleva a cabo una tarea en particular, o en algunas ocasiones una serie de tareas.

Es importante resaltar que el software que se ejecuta en un sistema embebido es diseñado bajo algunas restricciones importantes: (1) cantidades pequeñas de memoria, generalmente en el orden de los KB, (2) capacidades limitadas de procesamiento, generalmente los procesadores poseen velocidades que no superan los Mhz, (3) la necesidad de limitar el consumo de energía en cualquier instante, bien sea en estado de ejecución o no. (Pérez, 2009)

2.1.2 Características de un Sistema Embebido

Los sistemas embebidos poseen ciertas características que los distinguen de otros sistemas de cómputo.

2.1.3 Hardware de sistemas embebidos:

El término hardware en cualquier sistema se refiere a los componentes físicos que lo forman o constituyen; estos componentes permiten realizar un conjunto de tareas al ejecutar programas o software.

Los componentes físicos de un sistema embebido por lo general difieren en algunos aspectos de los que conforman un sistema de propósito general, como un computador de escritorio en: tamaño, capacidad de cómputo, requerimientos de energía, etc. Por esta razón es de gran importancia conocer el funcionamiento del hardware para poder desarrollar sistemas embebidos y las aplicaciones que se ejecutarán en él. A continuación se destacan los aspectos más relevantes del hardware comúnmente utilizado en sistemas embebidos. (Pérez, 2009)

2.1.4 Esquema de bloques funcionales de un sistema embebido:

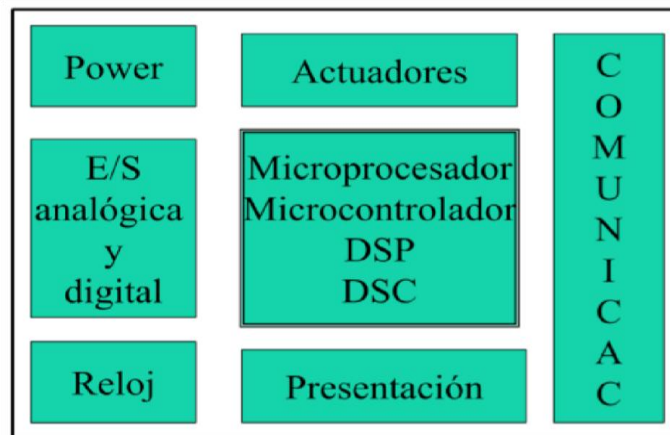


Figura 2.1.- Diagrama de bloques simplificado de un módulo típico de un SE. (Miñarro, 2009)

A continuación se describen los módulos.

2.1.5 Microprocesadores:

Es un chip que incluye básicamente la CPU y circuitería relacionadas con los buses de datos y memoria. Para poder realizar su tarea se necesitan otros chips adicionales (Sistema mínimo) tales como memoria, circuitos de entrada salida E/S (I/O) y reloj (Miñarro, 2009)

2.1.6 Microcontrolador:

Es un dispositivo que alberga el sistema mínimo dentro de un único chip, esto es, incluye CPU, buses, reloj, memoria ROM, memoria RAM, E/S, otros periféricos tales como conversores A/D, temporizadores (timers), etc. (Miñarro, 2009)

2.1.7 Sensores:

Un sensor es un dispositivo eléctrico y/o mecánico que convierte magnitudes físicas en valores medibles de dicha magnitud. Los sensores van a aportar información tanto del entorno como del estado interno del componente que mide.

La señal medida usualmente debe transformarse para poder ser interpretada. Este proceso se realiza en tres fases:

- Un fenómeno físico es captado por un sensor, como consecuencia, muestra en su salida una señal eléctrica equivalente al fenómeno captado.

- La señal eléctrica es modificada por un sistema de acondicionamiento de señal, cuya salida es un voltaje que será convertido usando un convertidor analógico/digital (A/D) para ser tratado.
- El convertidor A/D es sensible sólo a rangos limitados de tensiones, frecuentemente 0 a 5V. El convertidor hace que la salida continua se convierta en una salida discreta.

En sistemas embebidos suelen utilizarse diversos tipos de sensores como por ejemplo, sensores de luz, sensores de contacto, sensores de temperatura, etc.

2.1.8 Memoria:

Cualquiera de las funcionalidades de los sistemas embebidos están compuestas de tres aspectos: procesamiento, almacenamiento y comunicación. El procesamiento es la transformación de los datos, el almacenamiento es la retención de los datos para su posterior uso, y la comunicación es la transferencia de los datos. Cada uno de estos aspectos debe ser implementado. Se usan procesadores para la transformación, memoria para el almacenamiento, y buses para la comunicación. En esta sección se describe el elemento memoria.

2.1.9 Organización de memoria:

El elemento básico de una memoria es la celda. Aunque se utilizan diversas tecnologías electrónicas, todas las celdas de memoria comparten ciertas propiedades:

- Presentan dos estados estables, que pueden emplearse para representar el 1 y 0 binarios.
- Puede escribirse en ellas (al menos una vez) para fijar su estado.
- Pueden leerse para detectar su estado.

Lo más común es que la celda tenga tres terminales para transportar señales eléctricas. El terminal de selección selecciona la celda, para que pueda realizarse una operación de escritura o de lectura. El terminal de control indica si la operación se trata de una lectura o una escritura. Para la escritura el tercer terminal proporciona la señal que fija el estado de la celda a 1 ó 0. En la lectura el tercer terminal se utiliza como salida del estado de la celda.

2.1.10 Tipos de memoria

A continuación se describen los dos tipos de memoria (RAM y ROM), junto con algunas de sus variantes.

2.1.11 Memoria RAM:

Una de las características distintivas de las RAM (Random Access Memory) es la posibilidad de leer datos, como escribirlos rápidamente. La otra característica distintiva es que una RAM es volátil. Una RAM debe estar siempre alimentada por corriente eléctrica, si se interrumpe la alimentación se pierden los datos.

Las dos formas básicas de memoria de acceso aleatorio son la RAM dinámica (DRAM – Dynamic RAM) y la RAM estática (SRAM – Static RAM). Una DRAM está hecha con celdas que almacenan los datos como cargas eléctricas en condensadores. La presencia o ausencia de carga en un condensador se interpreta como el 1 ó 0 binarios. Ya que los condensadores tienen una tendencia natural a descargarse, las DRAM requieren refrescos periódicos para mantener memorizados los datos. Una SRAM es un dispositivo digital, basado en los mismos elementos que se usan en el procesador. En una SRAM los valores binarios se almacena utilizando flip-flops, los cuales mantienen sus datos en tanto se mantengan alimentados. Otras formas de RAM son la RAM pseudo estática (PSRAM – PseudoStaticRAM) y la RAM no volátil (NVRAM – NonVolatile RAM). La PSRAM es una DRAM con un controlador refrescador de memoria embebido. La NVRAM es una variación especial de la RAM que es capaz de mantener datos, incluso luego de que se remueve la alimentación de poder. Esto se logra almacenando los valores en una memoria EEPROM (explicada en la siguiente sección), justo antes de perder la alimentación.

2.1.12 Memoria ROM:

Una memoria de sólo lectura (ROM –Read Only Memory) contiene un patrón permanente de datos que no puede alterarse. Una ROM es no volátil. Además suele ser utilizada para almacenar subrutinas de bibliotecas para funciones de uso frecuente, programas del sistema, etc.

Existen varios tipos de memoria ROM, entre los cuales se encuentran:

- ROM programada por máscara. Las conexiones son programadas cuando el chip es fabricado, creando un conjunto apropiado de máscaras.
- OTP ROM (One Time Programmable ROM). Esta puede ser programada por un diseñador en un laboratorio, después de la manufactura del chip. La OTP ROM más básica usa un fusible para conexiones programables, dichos fusibles son fundidos para la representación del valor 0 en binario.
- EPROM (Erasable Programmable ROM). Este dispositivo usa un transistor como componente programable. Esta se lee y escribe electrónicamente, pero para el borrado de la misma es necesario exponerla a radiación ultravioleta para reasignar carga negativa a todas las celdas.
- EEPROM (Electrically Erasable Programmable ROM). En esta memoria se puede escribir en cualquier momento sin borrar todo el contenido anterior, ya que permite actualizaciones a nivel de byte. El borrado de los datos se realiza mediante un voltaje específico a los bytes de memoria direccionados.

- Flash. Esta memoria se introdujo a finales de los 80s como una mejora de las memorias EEPROM. Su funcionamiento es similar a la de su antecesora, con una mejora significativa en la velocidad de lectura y escritura. (Pérez, 2009)

DSC: Dispositivos mixtos microcontrolador/DSP que algunos fabricante ofrecen dentro de su catálogo de productos.

Probablemente, el microcontrolador 8051 (8 bits) desarrollado por Intel en 1980, marca el inicio en la carrera hacia el desarrollo de productos específicos para aplicaciones embebidas. Éste es probablemente el microcontrolador más popular, pues aunque se lleva hablando mucho tiempo de que estaba condenado a la desaparición, sus continuas mejoras le aseguran una larga vida. En este aspecto, comentar que los núcleos 8051 se usan en más de 100 microcontroladores de más de 20 fabricantes independientes como Atmel, Dallas Semiconductor, Philips, Winbond, entre otros. La denominación oficial de Intel para familia de μ Cs 8051 es MCS 51.

2.1.13 Comunicaciones:

Los sistemas de comunicaciones adquieren, en el diseño de sistemas embebidos, cada vez mayor importancia. Lo normal es que el SE pueda comunicarse mediante interfaces estándar de comunicaciones por cable o inalámbricas. Así un SE normalmente incorpora puertos de comunicaciones bajo los estándares más extendidos, bien aquellos que necesitan de un cableado físico o se trate de comunicaciones inalámbricas. Podemos citar:

RS-232, RS485, SPI, CAN, USB

2.1.14 Presentación:

El subsistema presentación típico suele ser una pantalla gráfica, LCD Alfanumérico, diodos LED, etc. Por lo general forma parte del interfaz hombre máquina del sistema, si es que lo lleva. El uso de pantallas gráficas del tipo táctil suele ser una solución muy aceptada, aunque conlleva mayor complejidad en el software a desarrollar y mayor potencia de cálculo de la CPU seleccionada.



En la figura 2.1 se muestra un típico display LCD alfanumérico de dos líneas por 20 caracteres por línea.

2.1.15 Actuadores:

Se denominan actuadores a los posibles elementos encargados de llevar a cabo las acciones indicadas por la CPU. Entre éstos disponemos de drivers de corriente, controladores de motores eléctricos, conmutadores, relés, etc.

2.1.16 Pines de E/S analógicos y digitales:

El módulo de Entrada/Salida (I/O) se encarga de hacer llegar o enviar las señales analógicas y digitales a los diferentes circuitos encargados de su generación y procesamiento. Tal es el caso de la conversión A/D para el procesamiento digital de señales analógicas procedentes de sensores, activación de actuadores mediante circuitos ‘driver’, reconocimiento del estado abierto cerrado de un conmutador o pulsador, encendido de diodos LED, etc.

2.1.17 Reloj:

El módulo de reloj es el encargado de generar las diferentes señales de reloj necesarias para la temporización de los circuitos digitales. Habitualmente se parte de un único oscilador principal, cuyas características son de vital importancia en determinadas aplicaciones.

2.1.18 Módulo de alimentación (Power):

El módulo de energía (power) se encarga de generar las diferentes tensiones y corrientes necesarias para alimentar los componentes activos que forman el SE. Lo normal es el empleo de baterías para los dispositivos portátiles y fuentes de alimentación (convertor AC/DC) para los sistemas que disponen de acceso a la red de energía eléctrica. (Miñarro, 2009)

2.1.19 Arduino.

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.



Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software.

Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, así pues eres libre de adaptarlos a tus necesidades.

Arduino recibió una Mención Honorífica en la sección *Digital Communities* de la edición del 2006 del *ArsElectronicaPrix*. El equipo Arduino(Arduinoteam) es: MassimoBanzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis. (Arduino, 2014)

2.1.20 ¿Qué es Wiring?

El Wiring es un Framework de programación de código abierto para los microcontroladores

El Wiring permite la escritura de software multiplataforma para controlar los dispositivos conectados a una amplia gama de placas de microcontrolador para crear todo tipo de codificación creativa, objetos interactivos, espacios o experiencias físicas. El Framework es creado cuidadosamente con diseñadores y artistas en mente para fomentar una comunidad donde los principiantes a través de expertos de todo el mundo para compartir ideas, conocimientos y su experiencia colectiva. Hay miles de estudiantes, artistas, diseñadores, investigadores y aficionados que utilizan cableado para el aprendizaje, la creación de prototipos y la producción de un trabajo profesional terminado. (Wiring, 2004).

2.1.21 ¿Qué es Processing?

Processing es un lenguaje de programación, entorno de desarrollo, y la comunidad en línea. Desde el año 2001, producción ha promovido la alfabetización de software dentro de las artes visuales y la cultura visual dentro de la tecnología. Inicialmente creado para servir como un cuaderno de bocetos de software y para enseñar los fundamentos de programación de computadoras dentro de un contexto visual, producción evolucionó hasta convertirse en una herramienta de desarrollo para los profesionales. Hoy en día, hay decenas de miles de estudiantes, artistas, diseñadores, investigadores y aficionados

que utilizan el procesamiento para el aprendizaje, la creación de prototipos y producción. (Processing, 2004)

2.1.22 Sistema Android.

Android es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware (Software que reside entre una aplicación y el Sistema operativo para realizar interacciones entre ellos. Utilizado para soportar aplicaciones distribuidas)



(Chévez) está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google. Este sistema por lo general maneja aplicaciones como Market (Marketing) o su actualización, Play Store.

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. Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43,6% en el tercer trimestre. A nivel mundial alcanzó una cuota de mercado del 50,9% durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (ios de iPhone) con más cuota.

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 Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java. No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ello es descargado de sitios de terceros.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto. (Yerika, 2012).

2.1.23 Las primeras versiones Android:

A continuación se muestran las versiones Android, desde la primera hasta la actual.

2.1.24 Android 1.0 Nivel de API 1 (septiembre 2008)

Primera versión de Android. Nunca se utilizó comercialmente, por lo que no tiene mucho sentido desarrollar para esta plataforma.

2.1.25 Android 1.1 Nivel de API 2 (febrero 2009)

No se añadieron apenas funcionalidades simplemente se fijaron algunos errores de la versión anterior. Es la opción a escoger si queremos desarrollar una aplicación compatible con todos los dispositivos Android. No obstante apenas existen usuarios con esta versión.

2.1.26 Cup cake

Android 1.5 Nivel de API 3 (abril 2009)



En la figura 2.2 se representa el logo de la primera versión Android

Es la primera versión con algún usuario, aunque en la actualidad apenas quedan. Como novedades, se incorpora la posibilidad de teclado en pantalla con predicción de texto, los terminales ya no tienen que tener un teclado físico, así como la capacidad de grabación avanzada de audio y vídeo. También aparecen los widgets de escritorio y live folders.

2.1.27 Donut

Android 1.6 Nivel de API 4 (septiembre 2009)



En la figura 2.3 se representa el logo de la segunda versión Android

Permite capacidades de búsqueda avanzada en todo el dispositivo. También se incorpora gestores y multi-touch. Permite la síntesis de texto a voz. También se facilita que una aplicación pueda trabajar con diferentes densidades de pantalla. Soporte para resolución de pantallas WVGA. Aparece un nuevo atributo XML, onClick, que puede especificarse en una vista. Play Store antes, AndroidMarket se mejora permitiendo una búsqueda más sencilla de aplicaciones.

2.1.28 Éclair

Android 2.0 Nivel de API 5 (octubre 2009)



En la figura 2.4 se representa el logo de la tercera versión Android

Esta versión de API apenas cuenta con usuarios, dado que la mayoría de fabricantes pasaron directamente de la versión 1.6 a la 2.1. Como novedades cabría destacar que incorpora un API para manejar el bluetooth 2.1. Nueva funcionalidad que permite sincronizar adaptadores para conectarlo a cualquier dispositivo.

Android 2.1 Nivel de API 7 (enero 2010)

Se considera una actualización menor, por lo que le siguieron llamando Éclair. Destacamos el reconocimiento de voz que permite introducir un campo de texto dictando sin necesidad de utilizar el teclado.

2.1.29 Froyo

Android 2.2 Nivel de API 8 (mayo 2010)



En la figura 2.5 se representa el logo de la cuarta versión Android

Como característica más destacada se puede indicar la mejora de velocidad de ejecución de las aplicaciones (ejecución del código de la CPU de 2 a 5 veces más rápido que en la versión 2.1 de acuerdo a varios benchmarks). Esto se consigue con la introducción de un nuevo compilador JIT de la máquina Dalvik.

Se añaden varias mejoras relacionadas con el navegador Web, como el soporte de Adobe Flash 10.1 y la incorporación del motor JavaScript V8 utilizado en Chrome.

Android 2.3 Nivel de API 9 (diciembre 2010)



En la figura 2.6 se representa el logo de la actualización de la versión froyo 2.2 a 2.3.

Debido al éxito de Android en las nuevas tabletas ahora soporta mayores tamaños de pantalla y resoluciones (WXGA y superiores).

Incorpora un nuevo interfaz de usuario con un diseño actualizado. Dentro de las mejoras de la interfaz de usuario destacamos la mejora de la funcionalidad de “cortar, copiar y pegar” y un teclado en pantalla con capacidad multitáctil.

2.1.30 Honeycomb

Android 3.0 Nivel de API 11 (febrero 2011)



En la figura 2.7 se representa el logo de la Quinta versión Android

Para mejorar la experiencia de Android en las nuevas tabletas se lanza la versión 3.0 optimizada para dispositivos con pantallas grandes. La nueva interfaz de usuario ha sido completamente rediseñada con paradigmas nuevos para la interacción y navegación. Entre las novedades introducidas destacan: Los fragmentos, con los que podemos diseñar diferentes elementos del interfaz de usuario

Android 3.1 Nivel de API 12 (mayo 2011)

Se permite manejar dispositivos conectados por USB (tanto host como dispositivo). Protocolo de transferencia de fotos y vídeo (PTP/MTP) y de tiempo real (RTP).

Android 3.2 Nivel de API 13 (julio 2011)

Optimizaciones para distintos tipos de tableta. Zoom compatible para aplicaciones de tamaño fijo. Sincronización multimedia desde SD.

2.1.31 IceCreamSandwich

Android 4.0 Nivel de API 14 (octubre 2011)



En la figura 2.8 se representa el logo de la sexta versión Android

La característica más importante es que se unifican las dos versiones anteriores (2.x para teléfonos y 3.x para tabletas) en una sola compatible con cualquier tipo de dispositivo. Entre las características más interesantes destacamos:

Se introduce un nuevo interfaz de usuario totalmente renovado.

Android 4.0.3 Nivel de API 15 (diciembre 2011)

Se introducen ligeras mejoras en algunas API incluyendo el de redes sociales, calendario, revisor ortográfico, texto a voz y bases de datos entre otros. En marzo de 2012 aparece la actualización 4.0.4.

2.1.32 JellyBean

Android 4.1 Nivel de API 16 (julio 2012)



En la figura 2.9 se representa el logo de la septima versión Android

En esta versión se hace hincapié en mejorar un punto débil de Android: la fluidez del interfaz de usuario. Con este propósito se incorporan varias técnicas, como: sincronismo vertical, triple búfer y aumentar la velocidad del procesador al tocar la pantalla.

Se mejoran las notificaciones con un sistema de información expandible personalizada. Los Widgets de escritorio pueden ajustar su tamaño y hacerse sitio de forma automática al situarlos en el escritorio. El dictado por voz puede realizarse sin conexión a Internet (de momento en inglés).

Parciales de aplicaciones.

Android 4.2 Nivel de API 17 (noviembre 2012)

Una de las novedades más importantes es que podemos crear varias cuentas de usuario en el mismo dispositivo. Aunque, esta característica solo está disponible en tabletas. Cada cuenta tendrá sus propias aplicaciones y configuración.

Los Widgets de escritorio pueden aparecer en la pantalla desbloqueo.

Android 4.3 Nivel de API 18 (julio 2013)

Esta versión introduce mejoras en múltiples áreas. Entre ellas los perfiles restringidos (disponible sólo en tabletas) que permiten controlar los derechos de los usuarios para ejecutar aplicaciones específicas y para tener acceso a datos específicos. Igualmente, los programadores pueden definir restricciones en las apps, que los propietarios puedan activar si quieren.

2.1.33 Kit Kat



En la figura 2.10 se representa el logo de la ultima versión Android

Android 4.4 Nivel de API 19 (octubre 2013)

Un objetivo principal de la versión 4.4 es hacer Android disponible en una gama aún más amplia de dispositivos, incluyendo aquellos con tamaños de memoria

RAM de sólo 512 MB. Para ello, todos los componentes principales de Android han sido recortados para reducir sus requerimientos de memoria, y se ha creado una nueva API que permite adaptar el comportamiento de la aplicación en dispositivos con poca memoria.

Más visibles son algunas nuevas características de la interfaz de usuario. El modo de inmersión en pantalla completa oculta todas las interfaces del sistema (barras de navegación y de estado) de tal manera que una aplicación puede aprovechar el tamaño de la pantalla completa.

2.1.34 Comunicación entre Android y Arduino:

Con el uso de la programación en Android, la tarjeta Arduino, su sketch, el dispositivo móvil, y los medios de conectividad, integrados se logra dar origen a una solución que permite el control del encendido y apagado de los bombillos de una casa de condiciones mínimas, contribuye a la sostenibilidad del medio ambiente y a las condiciones económicas de quienes la habitan por el ahorro de energía. (Ferley Medina Rojas, 2013).

2.1.35 Domótica:

Se entiende por domótica el conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y

exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado. (Acosta, 2012).

2.2 Marco Teórico de Solución.

La solución presentada para el “Complejo Educativo Marcos Ochoa” es un temporizador de actividades el cual automáticamente ya programado podrá hacer sonar el timbre a la hora establecida según programada por el director del complejo educativo; las dos partes que constituyen el proyecto son:

1-Circuito armado dentro de una caja de acrílico

En esta primera parte de nuestro proyecto estará constituida por un microcontrolador Atmega328 el cual ya está incorporado en la plataforma Arduino uno, también está constituido por diferentes módulos los cuales son el módulo de bluetooth, el módulo de relé y el RTC. Para visualizar la hora tendrá una LCD la cual mostrara la fecha y hora actual.



Figura 2.11 Imagen frontal del chasis del circuito armado en la caja de acrílico con los principales módulos para su correcto funcionamiento.

2-Un Smartphone con la APK ya instalada y funcional.



En la figura 2.12 se da a conocer el móvil con la aplicación Android

El temporizador del “Complejo Educativo Marcos Ochoa” está dividido en: Parte frontal, parte trasera y parte lateral. A continuación se describirán cada parte del proyecto.

Parte Frontal.

En la parte frontal de nuestro circuito podemos visualizar los diferentes módulos y los demás componentes.

Arduino Uno: Componente principal del circuito ya que este se podría decir que es el cerebro de todo el circuito ya que en él está el programa con el cual podrá controlar todos los módulos.

Pantalla LCD: La función de esta pantalla es mostrar mensajes del Complejo Educativo ya programados, la fecha y hora actual del circuito para que así puedan estar verificando la hora y la fecha.

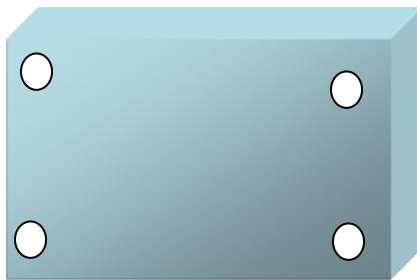
Módulo de Bluetooth: Este es uno de los módulos importantes ya que la función principal es ser un puente de comunicación entre el Arduino y el Smartphone y así lograr que desde el Smartphone se pueda programar el Arduino y establecer los diferentes horarios en los cuales se desea que el timbre se active.

Módulo de Relé: La función de este módulo es como de la un interruptor que a la hora de recibir un pulso del Arduino se activa y manda voltaje al timbre para que emita el sonido.

Módulo RTC: Este módulo es el que nos proporciona la hora y fecha actual este mandara la información correspondiente al Arduino para ser leídos y según las horas establecidas en el programa las comparara con la fecha y hora actual y si los dos datos son iguales el Arduino de inmediato mandara un impulso al relé para que se active el timbre.

Parte Trasera

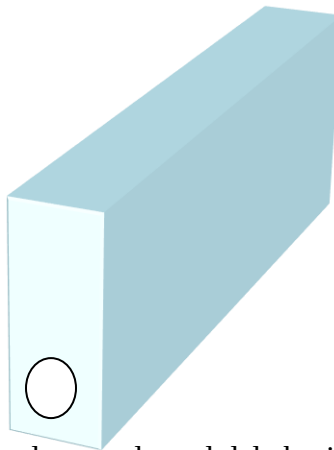
La función de la parte trasera del chasis del circuito es para sujetar el circuito adonde estará instalado el chasis. Este tendrá cuatro orificios con cuatro anclas para uno en cada esquina para sujetarlo en la pared ya que este circuito estará instalado en la pared de la dirección.



En la figura 2.13 se da a conocer la parte trasera del acrílico donde estará el circuito.

Parte Lateral

La parte lateral del chasis del circuito tendrá un orificio en la parte inferior y servirá para que el cable de poder puedan entrar fácilmente y así suministrar el circuito.



En la figura 2.14 se da a conocer la parte lateral del chasis.

2.3 Marco Teórico Conceptual.

A continuación se presentan una serie de conceptos con sus respectivos significados.

Actuador: dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.



Figura 2.15 Ejemplo de actuador.

Arduino: es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.



Figura 2.16 Logo de la plataforma Arduino.

Automatización: Sistema tecnológico basado en la ingeniería y la informática, que proporciona una optimización de los procesos productivos mediante la regulación automática (autorreguladores).



Figura 2.17 Ejemplo de Automatización.

Byte: Es una unidad de información utilizada como un múltiplo del bit. Generalmente equivale a 8 bits.



Figura 2.18 Referente a un Byte

Cable: conductor (generalmente cobre) o conjunto de ellos generalmente recubierto de un material aislante o protector, si bien también se usa el nombre de cable para transmisores de luz (cable de fibra óptica) o esfuerzo mecánico (cable mecánico).



Figura 2.19 Tipos de cable

Circuito: red eléctrica (interconexión de dos o más componentes, tales como resistencias, inductores, condensadores, fuentes, interruptores y semiconductores) que contiene al menos una trayectoria cerrada.

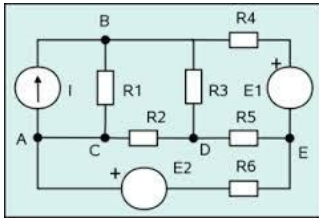


Figura 2.20 Ejemplo de circuito

Circuito integrado: también conocido como chip o microchip, es una pastilla pequeña de material semiconductor, de algunos milímetros cuadrados de área, sobre la que se fabrican circuitos electrónicos generalmente mediante fotolitografía y que está protegida dentro de un encapsulado de plástico o cerámica.

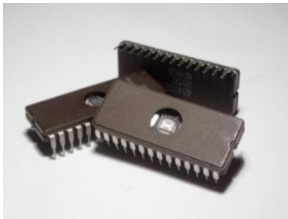


Figura 2.21 Circuitos integrados

Comunicación: Es proceso mediante el cual se puede transmitir información de una entidad a otra.



Figura 2.22 Imagen de Ejemplo de Comunicación.

Comunicación inalámbrica: Es aquella en la que extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio.



Figura 2.23 Logo de comunicación Inalámbrica

Conexión: Acción de conectar un aparato, una máquina u otro sistema eléctrico a su fuente de energía o dos aparatos o partes de un sistema eléctrico entre sí.



Figura 2.24 Ejemplo de conexión.

Dato: Es un valor o referente que recibe el computador por diferentes medios, los datos representan la información que el programador manipula en la construcción de una solución o en el desarrollo de un algoritmo.



Figura 2.25 Imagen referente a un Dato.

Domótica: Por Domótica entendemos la incorporación al equipamiento de nuestras viviendas y edificios de una sencilla tecnología que permita gestionar de forma energéticamente eficiente, segura y confortable para el usuario, los distintos aparatos e

instalaciones domésticas tradicionales que conforman una vivienda (la calefacción, la lavadora, la iluminación).



Figura 2.26 Ejemplo de Sistema Domótico

Electrónica: es la rama de la física y especialización de la ingeniería, que estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el control del flujo de los electrones u otras partículas cargadas eléctricamente.



Figura 2.27 Imagen Electrónica

Ethernet: Es un estándar de redes de área local para computadores con acceso al medio por contienda (CSMA/CD). Su nombre viene del concepto físico de Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.



Figura 2.28 conexión Ethernet

Parámetro: Es una variable que puede ser recibida por una rutina o subrutina. Una subrutina usa los valores asignados a sus argumentos para alterar su comportamiento en tiempo de ejecución. La mayor parte de los lenguajes de programación pueden definir subrutinas que aceptan cero o más argumentos.

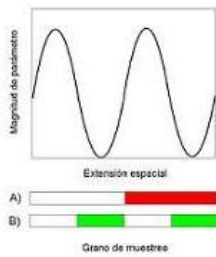


Figura 2.32 Imagen de un parámetro.

Pin: palabra inglesa que significa “clavija”, terminal o patilla a cada uno de los contactos metálicos de un conector o de un componente fabricado de un material conductor de la electricidad. Estos se utilizan para conectar componentes sin necesidad de soldar nada, de esta manera se logra transferir electricidad e información.



Figura 2.33 Imagen de un Pin.

Protocolo: Uno o un conjunto de procedimientos destinados a estandarizar un comportamiento humano u sistémico artificial frente a una situación específica.

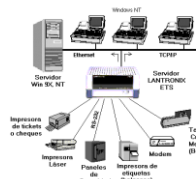


Figura 2.34 Imagen de Ejemplo de Protocolo.

Sensor: es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.



Figura 2.35 Ejemplo de Sensor.

Sistema embebido: sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas frecuentemente en un sistema de computación en tiempo real.

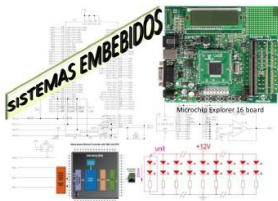


Figura 2.36 Ejemplo de Sistema Embebido

Sistema empotrado: es un sistema informático (hardware + software) de tiempo real integrado en un sistema de ingeniería más general, en el que realiza funciones de control, procesamiento y/o monitorización.

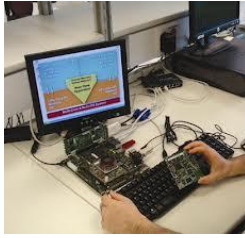


Figura 2.37 Ejemplo de Sistema Empotrado

Software: sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.



Figura 2.38 Logos de Diferentes Software

Timbre: Dispositivo electromagnético o mecánico que emite un sonido que sirve de llamada o de aviso: el timbre indica el comienzo de cada clase.



Figura 2.39 Timbre Electrónico

2.4 Documentación Técnica.

ARDUINO UNO

COMPONENTE	DESCRIPCIÓN
	Microcontrolador: ATmega328 Tensión de funcionamiento: 5V Voltaje de entrada (recomendado): 7-12V Voltaje de entrada (límites): 6-20V Digital I / O Pins: 14 (de los cuales 6 proporcionan PWM) Pines de entrada analógica: 6 Corriente continua para las E / S Pin: 40 mA Corriente de la CC para Pin 3.3V: 50 mA Memoria Flash: 32KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque: Tensión de funcionamiento: 5V Voltaje de entrada (recomendado): 7-12V Voltaje de entrada (límites): 6-20V Digital I / O Pins: 14 (de los cuales 6 proporcionan PWM) Pines de entrada analógica: 6 Corriente continua para las E / S Pin: 40 mA Corriente de la CC para Pin 3.3V: 50 mA Memoria Flash: 32 KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque SRAM: 2 KB (ATmega328) EEPROM: 1 KB (ATmega328) Velocidad del reloj: 16 MHz

MODULO BLUETOOTH

COMPONENTE	DESCRIPCIÓN
	Puede ser usado como un puerto serial Bluetooth, reemplazo directo de una conexión serial cableada, el uso es transparente. Ideal para proyectos embebidos con microcontroladores, módulos GPS y otros dispositivos de comunicación mediante UART. Chip CSR Bluetooth, compatible con el protocolo Bluetooth V2.0 Voltaje de operación 3.3V BaudRateajustable: 200400, 4800, 9600,19200, 38400, 57600, 115200 Tamaño: 28mm x 15 mm x 2.35mm Corriente de operación 40mA Corriente en modo de descanso< 1mA

MODULO DE RELÉ

COMPONENTES	DESCRIPCIÓN
	El control de todos los canales de este módulo se realiza a través de los pines E / S digitales 5, 6, 7 y 8 de la placa Arduino sobre la que va montado este shield. De todos modos, es necesaria una fuente de alimentación externa de 12 V CC para alimentar correctamente a las bobinas de cada relé. Esta tensión de 12VDC viene señalizada por el encendido del LED PWR rojo situado junto al bloque de conexiones o clemas correspondiente.

PANTALLA LCD

COMPONENTES	DESCRIPCÓN																																																												
	El LCD 16x2 (LCD 2x16) tiene en total 16 pines (tome en cuenta que la posición correcta del displayes con los pines en la parte superior, aunque existen modelos en los que la posición correcta es con los pines en la parte inferior).																																																												
	<table><tr><th>Pin No</th><th>Name</th><th>Function</th><th>Description</th></tr><tr><td>1</td><td>Vss</td><td>Power</td><td>GND</td></tr><tr><td>2</td><td>Vdd</td><td>Power</td><td>+ 5 V</td></tr><tr><td>3</td><td>Vee</td><td>Contrast Adj.</td><td>(-) 0 - 5 V</td></tr><tr><td>4</td><td>RS</td><td>Command</td><td>Register Select</td></tr><tr><td>5</td><td>R/W</td><td>Command</td><td>Read / Write</td></tr><tr><td>6</td><td>E</td><td>Command</td><td>Enable (Strobe)</td></tr><tr><td>7</td><td>D0</td><td>I/O</td><td>Data LSB</td></tr><tr><td>8</td><td>D1</td><td>I/O</td><td>Data</td></tr><tr><td>9</td><td>D2</td><td>I/O</td><td>Data</td></tr><tr><td>10</td><td>D3</td><td>I/O</td><td>Data</td></tr><tr><td>11</td><td>D4</td><td>I/O</td><td>Data</td></tr><tr><td>12</td><td>D5</td><td>I/O</td><td>Data</td></tr><tr><td>13</td><td>D6</td><td>I/O</td><td>Data</td></tr><tr><td>14</td><td>D7</td><td>I/O</td><td>Data MSB</td></tr></table>	Pin No	Name	Function	Description	1	Vss	Power	GND	2	Vdd	Power	+ 5 V	3	Vee	Contrast Adj.	(-) 0 - 5 V	4	RS	Command	Register Select	5	R/W	Command	Read / Write	6	E	Command	Enable (Strobe)	7	D0	I/O	Data LSB	8	D1	I/O	Data	9	D2	I/O	Data	10	D3	I/O	Data	11	D4	I/O	Data	12	D5	I/O	Data	13	D6	I/O	Data	14	D7	I/O	Data MSB
Pin No	Name	Function	Description																																																										
1	Vss	Power	GND																																																										
2	Vdd	Power	+ 5 V																																																										
3	Vee	Contrast Adj.	(-) 0 - 5 V																																																										
4	RS	Command	Register Select																																																										
5	R/W	Command	Read / Write																																																										
6	E	Command	Enable (Strobe)																																																										
7	D0	I/O	Data LSB																																																										
8	D1	I/O	Data																																																										
9	D2	I/O	Data																																																										
10	D3	I/O	Data																																																										
11	D4	I/O	Data																																																										
12	D5	I/O	Data																																																										
13	D6	I/O	Data																																																										
14	D7	I/O	Data MSB																																																										

RELOJ RTC DS1307:

COMPONENTE	DESCRIPCÓN
	Ds1307 i2c el reloj de tiempo real chip (rtc) 24 c32 32 k i2c la memoria eeprom 1024 la batería de litio recargable, y hacerse cargo del circuito ds1307 solución con la batería de nuevo- hasta no pueden leer o escribir. Lleno de electricidad, puede proporcionar la sincronización ds1307 1 año. Diseño compacto, mm 27*28* mm mm 8.4 sacar ds1307 pin de reloj, proporcionar la señal del reloj para el microordenador chip único. Puede conectar en cascada el otro i2c dispositivos.

TIMBRE:

COMPONENTE	DESCRIPCIÓN
	Timbre eléctrico con martillo, Trabaja a 110v. 10cm de tamaño, posee gran sonoridad.

Capítulo III Desarrollo de la Solución.

3.1 Propuesta de Solución.

Después de haber estudiado a profundidad la situación problemática el “Complejo Educativo Marcos Ochoa”, se identificó que carece de un sistema de timbrado que cumpla con exactitud, eficiencia y puntualidad la activación del timbre para las actividades diarias de dicha institución. Se logró identificar que para evitar todos los inconvenientes que tiene el Director y secretaria encargados de tocar la campana para cada actividad (recreos, reuniones e ingreso a clases), la solución más viable para que se eviten las problemáticas presentadas, es de realizar el proyecto denominado “Implementación de un sistema Domótico con tecnología Android para el temporizado de las actividades del Complejo Educativo Marcos Ochoa”.

Este proyecto consiste en la implementación de un sistema para el temporizado de las actividades diarias, teniendo como producto principal un circuito que controla la hora y fecha en tiempo real para así comparar con las horas enviadas vía aplicación Android, instalada en cualquier dispositivo móvil con sistema Android.

El sistema está compuesto por: Arduino Uno, Pantalla LCD SHIELD, Modulo Bluetooth, Reloj RTC, y Modulo Relé.

El funcionamiento estará controlado por la placa Arduino Uno, la cual será programada para el funcionamiento correcto de los demás componentes y así leer el sensor Bluetooth para programar las alarmas, luego leer Reloj RTC el cual tiene la hora y fecha actual, el programa verificara si la hora actual es igual a la hora de la alarma para activar el timbre y si no es así retorna a leer hora actual. El sistema también tendrá la facilidad de recibir nuevos horarios de alarmas por medio del módulo Bluetooth, por lo que se creó una aplicación para sistema Android la cual permitirá crear horarios de alarmas y enviarlas al sistema, siendo posible al momento de aparear los dos dispositivos Bluetooth (El del sistema y el dispositivo móvil).

Esto fue posible implementarlo gracias a los conocimientos adquiridos en la carrera Técnico en Ingeniería de Hardware, en la cual se impartieron asignaturas como Electrónica Analógica, Electrónica Digital, Microcontroladores, Robótica y Domótica, desarrollando proyectos similares en cada una de ellas.

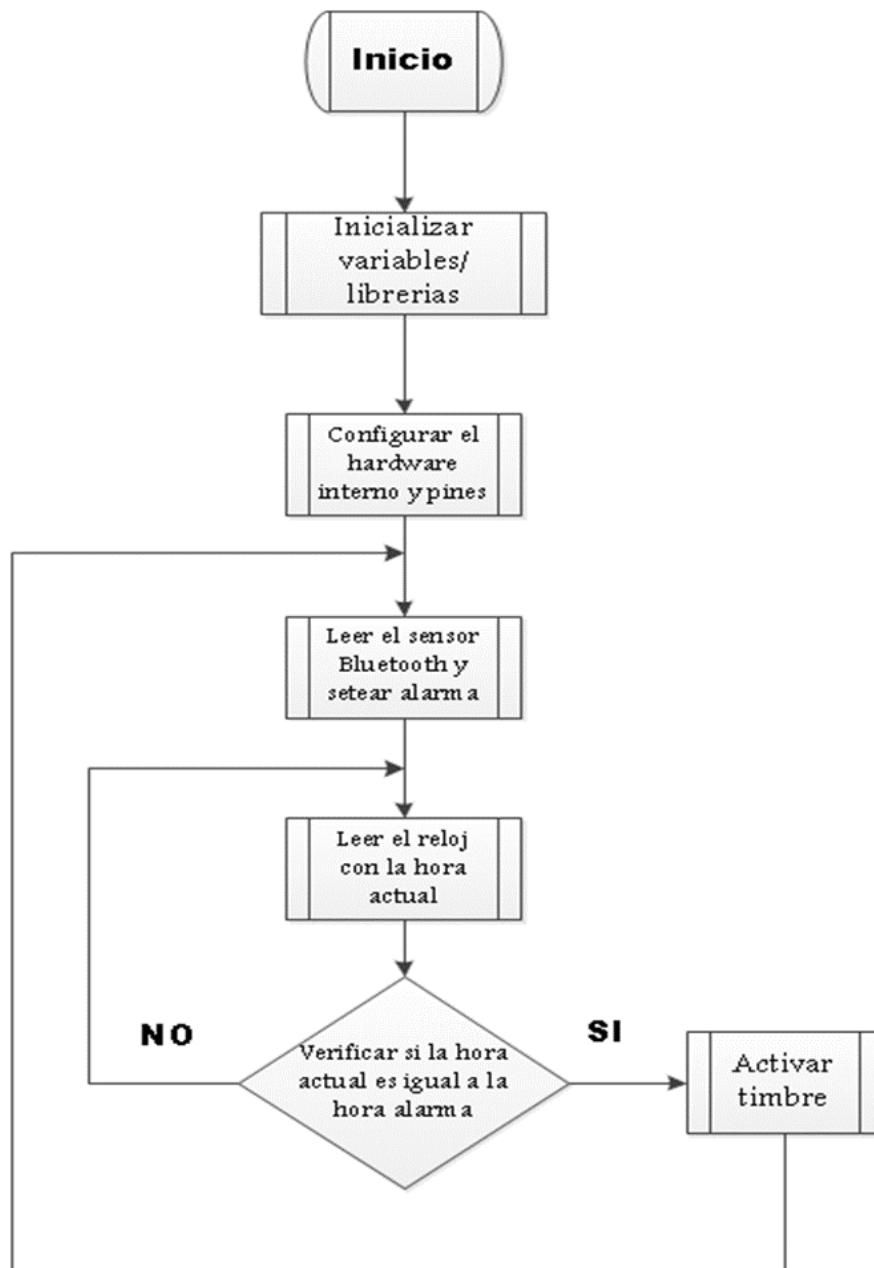
Para una posible implementación de este sistema se debe tomar en cuenta conocimientos en cuanto a la electrónica, sistemas embebidos y sistemas domóticos implementados con la plataforma Arduino. El conocimiento sobre creación de aplicaciones para sistema Android.

3.1.1 Firmware de control general del sistema.

Como todo circuito electrónico que utiliza un microcontrolador, siempre es necesario crear su firmware que lleve las instrucciones necesarias para el buen funcionamiento de los diferentes componentes que se utilizaran; es por ello que, para que nuestro proyecto funcione a la perfección, fue necesario crearlo en el software Arduino para así permitir que el sistema detecte cada componente y compare los horarios de alarma con la hora de tiempo real y activar el timbre.

A continuación se presenta el flujograma del funcionamiento del firmware de control.

3.1.2 Flujograma de funcionamiento de sistema de temporizado.



```

#include <Wire.h>

#include "RTCLib.h"

#include <LiquidCrystal.h>

#include <SoftwareSerial.h>


#define LOOP_DELAY 3000


LiquidCrystal lcd(8, 13, 9, 4, 5, 6, 7);

RTC_DS1307 RTC;

boolean bBlink = true;

const char *appVer = "C.E. Marcos Ochoa";

const char *modTxt = "Pres Select.....";

char rec;

SoftwareSerial blue(1,0);

// declara los horarios

const int alarmas[] = {7, 30, 9, 0, 9, 20, 10, 50, 11, 05, 11, 50, 13, 00, 14, 30, 14, 45, 15, 30, 15, 40, 17, 15};


// Global variables for time setup/displaying purposes.

uint16_t set_Year;

uint8_t set_Month;

uint8_t set_Day;

uint8_t set_Hour;

```

```

uint8_t set_Minute;

/*
 * Keypad shield uses resistors array and single analog input.
 * The values in adc_key_val array help to determine which
 * key on the shield was pressed by checking the analog input
 * read value.
 */
const int adc_key_val[5] = {50, 200, 400, 600, 800 };
int adc_key_in;

// keypad keys definitions
enum KP
{
    KEY_RIGHT = 0,
    KEY_UP,
    KEY_DOWN,
    KEY_LEFT,
    KEY_SELECT,
    KEY_NUMKEYS, // mark the end of key definitions
    KEY_NONE // definition of none of the keys pressed
};

// Finite Machine States
enum FMS
{
    RUN = 0,

```

```
SETCLOCK,  
SETYEAR,  
SETMONTH,  
SETDAY,  
SETHOUR,  
SETMINUTE  
} ClockState;
```

```
// Finite Machine State Transitions Table.
```

```
// Defines the flow of the application modes from one to another.
```

```
enum FMS StateMachine[] =  
{  
    /* RUN    -> */ SETCLOCK,  
    /* SETCLOCK -> */ SETYEAR,  
    /* SETYEAR -> */ SETMONTH,  
    /* SETMONTH -> */ SETDAY,  
    /* SETDAY  -> */ SETHOUR,  
    /* SETHOUR -> */ SETMINUTE,  
    /* SETMINUTE -> */ RUN  
};
```

```
enum KP key = KEY_NONE;
```

```
// Array of the numbers of month days.
```

```
const unsigned int month_days [] =  
{31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
const char *daysOfWeek[] =
```

```

{ "DOM", "LUN", "MAR", "MIE", "JUE", "VIE", "SAB" };

int nDelay = 0; // controlling loops/key press latency

void setup (){
  pinMode(13,OUTPUT);
  blue.begin(9600);
  blue.println("Conectado");
}
{
  pinMode(53, OUTPUT);

  nDelay = 0;
  key = KEY_NONE;
  // initialize LCD keypad module
  lcd.clear();
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.print(appVer);
  // power to i2c_ds1307_at24c32 module provided via A2, A3 pins
  pinMode(A3, OUTPUT);
  digitalWrite(A3, HIGH);
  pinMode(A2, OUTPUT);
  digitalWrite(A2, LOW);
  // start communication, I2C and RTC
  Wire.begin();
  RTC.begin();

```

```

    ClockState = RUN;

    /*
    readKey();
    // if key SELECT is held at RESET/Start up procedure
    if (key == KEY_SELECT)
    {
        setClock(RTC.isrunning());
    }
    else if (key == KEY_LEFT) // if key LEFT is held at RESET
    {
        // following line sets the RTC to the date & time this sketch
        // was compiled
        RTC.adjust(DateTime(__DATE__, __TIME__));
    }
    else if (! RTC.isrunning()) // if battery was changed
    {
        setClock(false);
    }
    */

    analogWrite(10, 128);
} //fin del SETup

//Recibir datos modulo Bluetooth

void loop (){
    if(blue.available()){
        rec=blue.read();
    }
}

```

```

switch(rec){
  case '0':
    digitalWrite(13,LOW);
    delay(500);
    blue.println("Led off");
    break;
  case '1':
    digitalWrite(13,HIGH);
    delay(500);
    blue.println("Led on");
    break;
  default:
    delay(500);
    blue.print(rec);
    blue.println(" no es una orden valida. Introduzca 0 o 1");
}
}
}
{
  readKey();
  if (key == KEY_SELECT)
  {
    ClockState = StateMachine[ClockState]; // switch mode
    delay(330);
  }
}

switch (ClockState)

```

```

{
case RUN:
    if (nDelay <= 0)
    {
        DateTime now = RTC.now();
        dispTime(now.year(),
                now.month(),
                now.day(),
                now.hour(),
                now.minute(),
                now.dayOfWeek());
        bBlink = ((bBlink) ? false : true);
    }

    break;

case SETCLOCK:

    nDelay = 0;
    lcd.setCursor(0,0);
    lcd.print(modTxt);
    setClock(RTC.isrunning());
    lcd.setCursor(0,0);
    lcd.print(appVer);
    break;

default: break;

```

```

    }

    if (nDelay <= 0)
        nDelay = LOOP_DELAY;
    else
        nDelay--;
}

/*
 * Functions to aid displaying the date/time.
 */
void dispYear(uint16_t yr)
{
    if (ClockState == SETYEAR)
    {
        if (bBlink)
            lcd.print(yr, DEC);
        else
            lcd.print("  ");
    }
    else
        lcd.print(yr, DEC);
}

void dispMonth(uint8_t mo)
{
    if (ClockState == SETMONTH)

```

```

{
    if (bBlink)
        lcd.print(mo, DEC);
    else
    {
        lcd.print(' ');
        if (mo >= 10)
            lcd.print(' ');
    }
}
else
    lcd.print(mo, DEC);
}

```

```

void dispDay(uint8_t dy)
{
    if (ClockState == SETDAY)
    {
        if (bBlink)
            lcd.print(dy, DEC);
        else
        {
            lcd.print(' ');
            if (dy >= 10)
                lcd.print(' ');
        }
    }
}

```

```

else
    lcd.print(dy, DEC);
}

void dispHour(uint8_t hr)
{
    if (hr < 10)
    {
        if (ClockState == SETHOUR)
        {
            if (bBlink)
                lcd.print('0');
            else
                lcd.print(' ');
        }
        else
            lcd.print('0');
    }
    if (ClockState == SETHOUR)
    {
        if (bBlink)
            lcd.print(hr, DEC);
        else
        {
            lcd.print(' ');
            if (hr >= 10)
                lcd.print(' ');
        }
    }
}

```

```

    }
}
else
    lcd.print(hr, DEC);
}

void dispMinute(uint8_t mn)
{
    if (mn < 10)
    {
        if (ClockState == SETMINUTE)
        {
            if (bBlink)
                lcd.print('0');
            else
                lcd.print(' ');
        }
        else
            lcd.print('0');
    }
    if (ClockState == SETMINUTE)
    {
        if (bBlink)
            lcd.print(mn, DEC);
        else
        {
            lcd.print(' ');

```

```

        if (mn >= 10)
            lcd.print(' ');
        }
    }
    else
        lcd.print(mn, DEC);
}

////////////////////////////////////
/// funcion para desplegar la fecha y hora
////////////////////////////////////

void dispTime(uint16_t yr,
              uint8_t mo,
              uint8_t dy,
              uint8_t hr,
              uint8_t mn,
              uint8_t dow)
{
    lcd.setCursor(0,1);
    dispHour(hr);
    lcd.print(':');
    dispMinute(mn);
    lcd.setCursor (13, 1);
    lcd.print(daysOfWeek[dow]);

    ///// comparar las alarmas
    lcd.setCursor (5, 1);
    lcd.print(':');

```

```

DateTime now = RTC.now();

int horas = now.hour();

    int minutos = now.minute();

    int segundos = now.second();

if (segundos<10)

lcd.print('0');

lcd.print(segundos, DEC);

for (int indice=0; indice<33; indice=indice+2)

{

    //DateTime now = RTC.now();

    if (horas==alarmas[indice] && minutos==alarmas[indice + 1] && segundos < 1)

    {

        digitalWrite(53, HIGH); delay(2000); digitalWrite(53, LOW);

    }

}

/// fin comparar alarmas

}

/*

* Functions to aid setting date/time.

*/

void setYear(boolean incdec)

{

    if (incdec)

```

```
{  
    if (set_Year < 2100)  
        set_Year++;  
}  
else  
{  
    if (set_Year > 2000)  
        set_Year--;  
}  
}
```

```
void setMonth(boolean incdec)
```

```
{  
    if (incdec)  
    {  
        if (set_Month < 12)  
            set_Month++;  
    }  
    else  
    {  
        if (set_Month > 1)  
            set_Month--;  
    }  
}
```

```
void setDay(boolean incdec)
```

```
{
```

```

if (incdec)
{
    if ((set_Month != 2 && set_Day < month_days[set_Month])
        ||
        (set_Month == 2 && set_Day < 28)
        ||
        (set_Month == 2 && set_Day == 28 && isLeapYear(set_Year))
    )
        set_Day++;
}
else
{
    if (set_Day > 1)
        set_Day--;
}
}

```

```

void setHour(boolean incdec)
{
    if (incdec)
    {
        if (set_Hour < 23)
            set_Hour++;
        else
            set_Hour = 0;
    }
    else

```

```

{
    if (set_Hour > 1)
        set_Hour--;
    else
        set_Hour = 23;
}
}

```

```

void setMinute(boolean incdec)

```

```

{
    if (incdec)
    {
        if (set_Minute < 59)
            set_Minute++;
        else
            set_Minute = 0;
    }
    else
    {
        if (set_Minute > 1)
            set_Minute--;
        else
            set_Minute = 59;
    }
}
}

```

```

void setDateTime(boolean incdec)

```

```
{
  if (incdec)
  {
    switch (ClockState)
    {
      case SETYEAR:

        setYear(true); // increment year
        break;

      case SETMONTH:

        setMonth(true); // increment month
        break;

      case SETDAY:

        setDay(true); // increment day
        break;

      case SETHOUR:

        setHour(true); // increment hour
        break;

      case SETMINUTE:
```

```
        setMinute(true); // increment minute
        break;

    default: break;

    }
}
else
{
    switch (ClockState)
    {
        case SETYEAR:

            setYear(false); // decrement year
            break;

        case SETMONTH:

            setMonth(false); // decrement month
            break;

        case SETDAY:

            setDay(false); // decrement day
            break;

        case SETHOUR:
```

```

        setHour(false); // decrement hour
        break;

    case SETMINUTE:

        setMinute(false); // decrement minute

        default: break;

    }
}
}

void setClock(boolean readrtc)
{
    DateTime now = DateTime(2013,1,1,0,0,0);

    if (readrtc)
        now = RTC.now();

    delay(500);

    set_Year = now.year();
    set_Month = now.month();
    set_Day = now.day();
    set_Hour = now.hour();
    set_Minute = now.minute();

```

```

ClockState = SETYEAR;

while (ClockState >= SETCLOCK)
{
    if (nDelay <= 0)
    {
        dispTime(set_Year,
                set_Month,
                set_Day,
                set_Hour,
                set_Minute,
                now.dayOfWeek());

        bBlink = ((bBlink) ? false : true);
    }

    readKey();
    if (key == KEY_UP || key == KEY_DOWN)
        bBlink = true;
    if (key == KEY_SELECT)
    {
        ClockState = StateMachine[ClockState];
        delay(330);
    }
    else
    {
        if (nDelay <= 0)

```

```

{
  if (key == KEY_UP)
  {
    setDateTime(true); // increment
  }
  else if (key == KEY_DOWN)
  {
    setDateTime(false); // decrement
  }
  else if (key == KEY_LEFT)
  {
    ClockState = RUN; // exit set clock mode
  }
  nDelay = LOOP_DELAY;
}
}
if (nDelay <= 0)
  nDelay = LOOP_DELAY;
else
  nDelay--;
//delay(330);
}

```

```

RTC.adjust(DateTime(set_Year,
                    set_Month,
                    set_Day,
                    set_Hour,

```

```

        set_Minute,
        0));
    }

// Determine the leap year.
boolean isLeapYear(uint16_t yr)
{
    if ((yr%400)==0)
        return true;
    else if ((yr%100)==0)
        return false;
    else if ((yr%4)==0)
        return true;

    return false;
}

// Get key code from analog input.
unsigned int get_key(unsigned int input)
{
    unsigned int k;
    for (k = KEY_RIGHT; k < KEY_NUMKEYS; k++)
    {
        if (input < adc_key_val[k])
        {
            return k;
        }
    }
}

```

```

    }

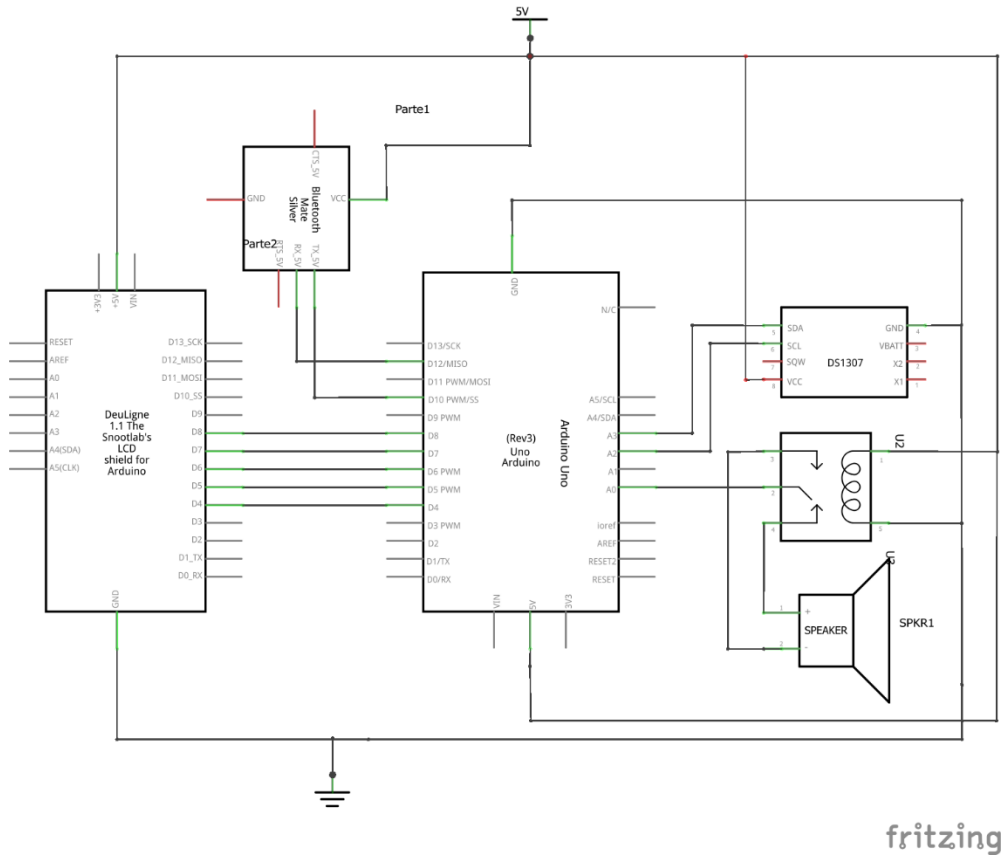
    if (k >= KEY_NUMKEYS) k = KEY_NONE; // No valid key pressed

    return k;
}

// Read analog input 0 to obtain key code in global variable: key
void readKey(void)
{
    key = KEY_NONE;
    adc_key_in = analogRead(0); // read the value from the sensor
    // convert into key press
    key = (enum KP) get_key(adc_key_in);
}

```

3.1.3 Diagrama Electrónico de Temporizador.



3.1.4 Aplicación Móvil (APK)

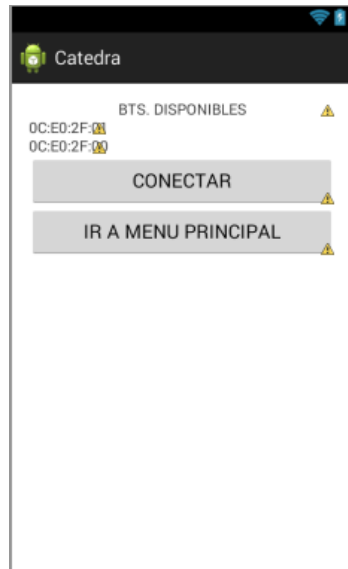
APP para administrar los horarios de timbrado automático.

La APP que se desarrolló para el proyecto está basada en una aplicación para el manejo de un sistema de timbrado del “Complejo Educativo Marcos Ochoa”, basado en la plataforma ARDUINO; la idea es hacer dinámico el proceso de configuración de horarios en los que se desea que el timbre suene a través de una aplicación Android que estará instalada en un Smartphone, de la persona

encargada en el complejo educativo, lo que se pretende es que en determinado momento el usuario final pueda consultar el horario actual y además poderlos modificar sin necesidad de asistencia técnica o manipulación directa del timbre como tal.

A continuación se muestra unas pantallas de la aplicación móvil.

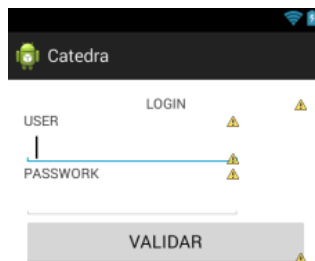
PANTALLA DE INICIO	
 Pantalla 1	El usuario, tendrá como bienvenida la Pantalla 1 En la cual se le mostrará tres botones para realizar diferentes acciones. • CONECTAR BT. Esta botón desplegará una Pantalla 2 • CONFI. HORARIO. Este botón desplegará una Pantalla 3, que se pedirá una validación de usuario para poder hacer una modificación a los horarios evitando que otra persona que no esté registrada acceda al sistema. • VER HORARIO. Este botón desplegará una pantalla informativa del último estado que se envió al timbre básicamente es un log. desplegará una Pantalla 5



Pantalla 2

La **pantalla 2**, es vital para poder parear el móvil con el BLUETOOTH del sistema del timbrado, el usuario seleccionara el dispositivo ARDUINO asociado.

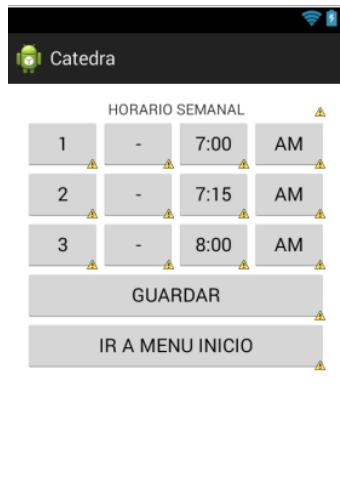
- **CONECTAR.** Una vez presionado este botón el teléfono quedara pareado con el BLUETOOTH del timbre.
- **IR A MENU PRINCIPAL.** Este botón retornara a la **Pantalla 1** de inicio.



Pantalla 3

Esta **pantalla 3**, permite hacer una validación del usuario, para evitar personas ajenas a la aplicación.

- **VALIDAR.** Este botón desplegará una **Pantalla 4** en el caso que el usuario exista, caso contrario no pasara a la siguiente pantalla para poder realizar los cambios al horario.



Pantalla 4

Esta **pantalla 4**, permite configurar las horas en las que se activara el timbre.

- **GUARDAR.** Este botón guardará los cambios en el horario.
- **IR A MENU INICIO.** Este botón retornara a la **Pantalla 1** de inicio.



Pantalla 5

Esta **pantalla 5**, solo es u

- **IR A MENU INICIO.** Este botón retornara a la **Pantalla 1** de inicio. Y básicamente en esta pantalla solo es una vista nada más de lo que hay actualmente cargado en el sistema del timbre.

3.1.5 Modelo ADDIE

Circuito Electrónico

Análisis.

En la actualidad una de las necesidades básicas de los Centros Escolares es la necesidad de tener un buen sistema de timbrado que anuncie los toques de entrada, salida, cambios de clases y otras actividades por esta razón surge la necesidad de realizar la implementación de un Sistema Domótico con tecnología Android para el temporizado de las actividades del “Complejo Educativo Marcos Ochoa”. Este servirá para que el director que es el encargado de hacer sonar la campana, ya no lo haga ya que el circuito electrónico será el encargado de realizar la actividad de temporizado del timbre , esto es posible gracias a las nuevas plataformas de circuitos electrónicos que han surgido desde el 2005 que son la plataforma Arduino. Esta es una placa electrónica capaz de controlar otros dispositivos electrónicos conectados a ella y gracias a que esta plataforma es de hardware libre y con el tiempo se han desarrollado módulos compatibles con Arduino los cuales faciliten la comunicación entre el circuito y la aplicación, el activar el timbre a la hora establecida, llevar la hora correcta, gracias a estos módulos y a la compatibilidad que tienen con Arduino es posible armar un circuito electrónico capaz de realizar las actividades de temporizado del “Complejo Educativo Marcos Ochoa”.

Diseño.

El diseño del circuito electrónico del sistema de temporizado estará compuesto por varios módulos compatibles con Arduino. Se optó por elegir diseñar el circuito electrónico con los módulos elaborados especialmente para Arduino ya que estos módulos son fabricados con tarjetas especiales, las cuales duran más que las tarjetas cobreadas comunes, las soldaduras son más resistentes ya que están son realizadas con máquinas especializadas; también las pistas no tienden a levantarse como normalmente lo hacen anteriores diseños de los circuitos. Cabe resaltar que estos módulos son diseñados a la medida de la plataforma Arduino.

Desarrollo.

Para el desarrollo del circuito electrónico se tomó de base de las cátedras impartidas de la carrera Técnico en Ingeniería de Hardware que son: Microcontroladores, Robótica y Domótica.

Para elegir cada módulo se tomaron los conocimientos de las cátedras mencionadas anteriormente, ya teniendo los conocimientos necesarios para poder elegir que modulos poder utilizar para el desarrollo por lo cual se realizaron dos evaluaciones la primera es la evaluación técnica en la cual se analizaron las características técnicas de cada módulo a ocupar que son el modulo del bluetooth, módulo de relé, módulo RTC y una LCD SHIELD. Luego de haber

realizado la evaluación técnica y de haber seleccionado los mejores módulos con las mejores características y los que más se acoplaban a la necesidad para realizar un circuito electrónico completo y funcional se realizó la evaluación económica ya que estos módulos son fabricados fuera del país, era necesario hacer un estudio en la internet de los diferentes proveedores y precios de cada de uno de los módulos se eligió por un proveedor que se ajuste a nuestras necesidades.

Luego de haber elegido el proveedor se procedió a realizar pruebas de cada de uno de los módulos individualmente y así mismo se comprobó la compatibilidad entre todos los módulos y la placa Arduino.

Implementación.

El circuito electrónico antes mencionado queda implementado y funcional en el Complejo Educativo Marcos Ochoa. El cual se encarga de realizar las actividades de temporizado del complejo educativo este quedará instalado en lugar fresco accesible al responsable y fuera del alcance de los estudiantes del complejo.

Evaluación.

Aspectos a evaluar del circuito electrónico son:

- Circuitos de primera calidad.
- Durabilidad.

- Funcionalidad de cada uno de los módulos.
- Compatibilidad entre todos los módulos.
- Tamaño.

Software

Análisis.

El software es uno de los elementos más importantes de este proyecto ya que este controla a todos los dispositivos electrónicos y así mismo manda las ordenes a cada uno de ellos para el correcto funcionamiento del circuito.

Para desarrollar el software se tomó de base las cátedras de microcontroladores, domótica, robótica y otras fuentes para así poder desarrollar el programa y realizar el archivo .INO

El software se realizó con el objetivo de que este controle a todos los dispositivos para cumplir con las labores de temporizado del complejo educativo para así darle un mayor confort al encargado de hacer tocar la campana.

Diseño.

El programa del circuito se realizó en el Software Arduino IDE el cual es el software original de la placa Arduino este facilita la programación ya que se le pueden añadir librerías de los diferentes elementos o módulos.

El programa estará compuesto por librerías específicas las cuales permiten controlar cada módulo de una manera eficiente y fácil ya que los proveedores de estos módulos desarrollan estas librerías para que al usuario se le facilite a la hora de utilizarlos y programarlos con Arduino. También tendrá variables de entrada y salida las cuales permiten guardar valores como la hora y la fecha y una de las partes más importantes que componen el programa son dos estructuras las cuales son el void setup() y el void loop() en estas dos estructuras se colocaran las configuraciones de pines, estructuras lógicas etc.

Desarrollo.

Para la creación del programa se necesitaron conocimientos básicos de programación en el lenguaje C para así tener una base de como programar declarar variables y comprender como estructurar una función lógica correctamente. para poder programar el Arduino se necesita un software llamado Arduino IDE en este software se crean los programas para la plataforma Arduino en este programa se desarrollaron las diferentes variables condiciones y acciones. Para poder desarrollar correctamente el programa se necesitan instrucciones

necesarias que debe tener el programa para funcionar correctamente las cuales son las siguientes:

Librerías - Las cuales nos ayudan a controlar fácilmente los diferentes módulos.

Variables de E/S – Estas variables son necesarias ya que en ellas almacenaremos los datos necesarios ya sea de entrada o salida

Declaración de pines – La declaración de pines es muy importante ya que en estas líneas de códigos se definen los pines de salida entrada y así se facilita la programación y la conexión de los dispositivos electrónicos.

Funciones Void - Estas funciones son las principales y necesarias para realizar un programa en el Arduino IDE ya que en el void setup() se colocan las instrucciones que se ejecutaran una vez al inicializar el programa y la void loop() se colocan todas las instrucciones que se repetirán infinitamente en el programa.

Funciones lógicas – las funciones lógicas son necesarias para tomar decisiones como la de comparar la hora establecida en el programa con la hora actual si esta condición encuentra que son iguales hace sonar el timbre.

Implementación.

El programa quedará ejecutándose en el Arduino para que este controle todos los dispositivos electrónicos y cumplir con la labor de timbrado del Complejo

Educativo, siendo uno de los grandes beneficios que dará mayor confort al encargado .

Evaluación.

Aspectos a evaluar del programa son:

- Uso correcto de las librerías de cada módulo.
- Buena estructura.
- Utilización de funciones lógicas.
- Ningún error lógico.
- Fácil de comprender.

Proyecto final

Análisis.

El último producto, es uno de los más importantes en este trabajo ya que ha sido desarrollado con un fin específico el cual es cumplir con las labores de timbrado del Complejo Educativo y así mismo dar mayor confort y tranquilidad para el encargado de hacer sonar el timbre.

Otro de los beneficios importantes de este proyecto es que se utilizaron elementos de última generación como son la plataforma Arduino los módulos

compatibles con este y la tecnología Android con lo cual se desarrolló un aplicación móvil APK.

Para poder realizar el programa funcional compilado y la aplicación APK, se investigó como poder realizarlos los más sencillamente y eficazmente posible ya habiendo realizado la investigación se optó por utilizar el Arduino IDE para poder realizar el programa funcional compilado ya que este es el software original para programar la placa Arduino. Para realizar la aplicación APK se tomó de base el libro de texto: “El gran libro de Andorid” el cual nos facilita el cómo realizar una aplicación APK desde cero, ya que en su contenido encontramos una guía para realizar aplicaciones móviles ya sea que eres un usuario principiante o experto.

Este proyecto funcional tendrá dos productos importantes: un circuito armado dentro de una caja de acrílico y una aplicación APK instalada en un SmartPhone.

Diseño.

El diseño de la aplicación será muy amistosa y fácil de programar para el usuario, tendrá tres botones en la pantalla principal con los cuales podrá modificar los horarios, ver los horarios y hacer la conectividad con el dispositivo de Bluetooth.

Desarrollo.

Para realizar estos últimos productos como siempre es necesario tener unas bases de estudio y saber que conocimientos tomar para así poder realizar cada uno de los productos como anteriormente se ha mencionado para poder realizar un programa funcional para la plataforma Arduino se necesitan tener conocimientos básicos sobre el lenguaje C y sobre cómo utilizar el Arduino IDE y saber declarar variables y estructurar correctamente las expresiones lógicas.

Para el desarrollo de la aplicación móvil funcional en Android se necesita el conocimientos de programación para poder realizar una aplicación estable y funcional, se tomó de base el libro de texto “El gran libro de Android” ya que este es un libro muy completo este es como una guía para comenzar a introducirse al desarrollo de aplicaciones Android es muy recomendado para principiantes y expertos en el desarrollo de aplicaciones Android.

Implementación.

El programa y la aplicación serán utilizado por el encargado del Complejo Educativo. Se concluye el trabajo dando así un buen servicio a la institución y así comprobar su funcionalidad, rapidez y exactitud del programa y la aplicación móvil a la hora de ingresar los horarios en los que desea que suene el timbre, también poder modificarlos y comprobar la exactitud de este.

Los beneficiados de este proyecto ya no tendrán que preocuparse por estar pendiente de tocar el timbre a la hora establecida ya que este lo hará automáticamente luego de haberle configurado las horas en las que deseen que suene el timbre.

Evaluación

Aspectos a evaluar del proyecto funcional

- Funcionalidad
- Rapidez.
- Exactitud.
- Buena presentación.
- Interfaz amigable de la aplicación.

3.2 Conclusiones.

Luego de finalizar el proyecto denominado: “Implementación de un sistema Domótico con tecnología Android para el temporizado de las actividades del Complejo Educativo Marcos Ochoa”, se concluye que la implementación de dicho sistema ha sido de gran importancia ya que ayuda a mejorar las labores de timbrado en el complejo educativo utilizando tecnología avanzada programable libre.

Se logró implementar luego de estudiar a detalle cada componente electrónico para elegir los más adecuados para un buen funcionamiento y una apariencia agradable del dispositivo, a su vez se realizó lo que es el firmware para que el sistema funcione correctamente utilizando el Lenguaje de programación Arduino.

También se concluye que los objetivos propuestos al inicio del proyecto fueron cumplidos, ya que se implementó el sistema Domótico, se elaboró el circuito electrónico, el firmware que permite que el sistema funcione y al final se integraron ambas partes.

De igual manera se brinda un aporte tecnológico al complejo educativo el cual es de mucha ayuda, beneficiando a los alumnos, maestros, y Director, dando un mayor confort para dedicarse a las labores estudiantiles sin preocupación alguna.

Como grupo fue de mucho agrado lograr contribuir con los conocimientos adquiridos en la carrera Técnico en Ingeniería de Hardware, ya que debido a eso

el Complejo Educativo Marcos Ochoa cuenta con un buen sistema de temporizado de timbrado.

3.3 Recomendaciones.

Como grupo se plantean diferentes recomendaciones para el uso adecuado del sistema y así evitar posibles daños, o mal funcionamiento.

- Mantener el dispositivo en lugar fresco y seco para evitar daños en los componentes.
- Verificar que el cable de alimentación se mantenga siempre conectado para que el dispositivo este siempre funcionando.
- En ningún momento retirar la batería del reloj RTC, ya que causaría perdida de horario y fecha.
- Al utilizar la aplicación Android verificar que los dispositivos Bluetooth estén apareados para que haya conexión y el sistema reciba los datos enviados vía teléfono o tablet.
- Mantener fuera del alcance de los alumnos.
- No conectar en toma corrientes con una regleta sobrecargados para evitar un corto circuito.
- Para dar un mayor alcance de la conectividad entre los dispositivos se recomienda usar una red Wifi.

3.4 Referencias

- Acosta, L. (2012). *Slideshare*. Recuperado de <http://www.slideshare.net/cinthia12345/que-es-domotica>
- Arduino. (2014). *Arduino*. Recuperado de: <http://www.arduino.cc/es/>
- Boxall, J. (2013). *Arduino Workshop*.
- Chávez, A. *Servidor de aplicaciones Jboss*.
- Ferley Medina Rojas, Y. J. (2013). *Arduino y Android una pareja para Aplicaciones de Ubicuidad*. Mexico: LACCEI.
- Michael Margolis, N. W. (2012). *Arduino Cookbook* (2da. Ed.). United States of America: O'REILLY.
- Miñarro, B. Ú. (2009). *Sistemas embebidos*.
- Monk, S. (2012). *Arduino + Android for the Evil Genius*. United States of America: Mc Graw Hill.
- Oto. (2011). *Sistemas Embebidos*. San Salvador: Universidad Tecnológica de El Salvador.
- Pérez, P. D. (2009). *Sistemas Embebidos y Sistemas Operativos Embebidos*.
- Processing. (2004). *Processing*. Recuperado de: <http://www.processing.org/>
- Wiring. (2004). *Wiring*. Recuperado de: <http://wiring.org.co/>
- Yerika, E. A. (2012). *Tecnología Android*. Recuperado de <http://tecnologiasandroid.blogspot.com/search?updated-min=2012-01-01T00:00:00-08:00&updated-max=2013-01-01T00:00:00-08:00&max-results=4>

Anexo

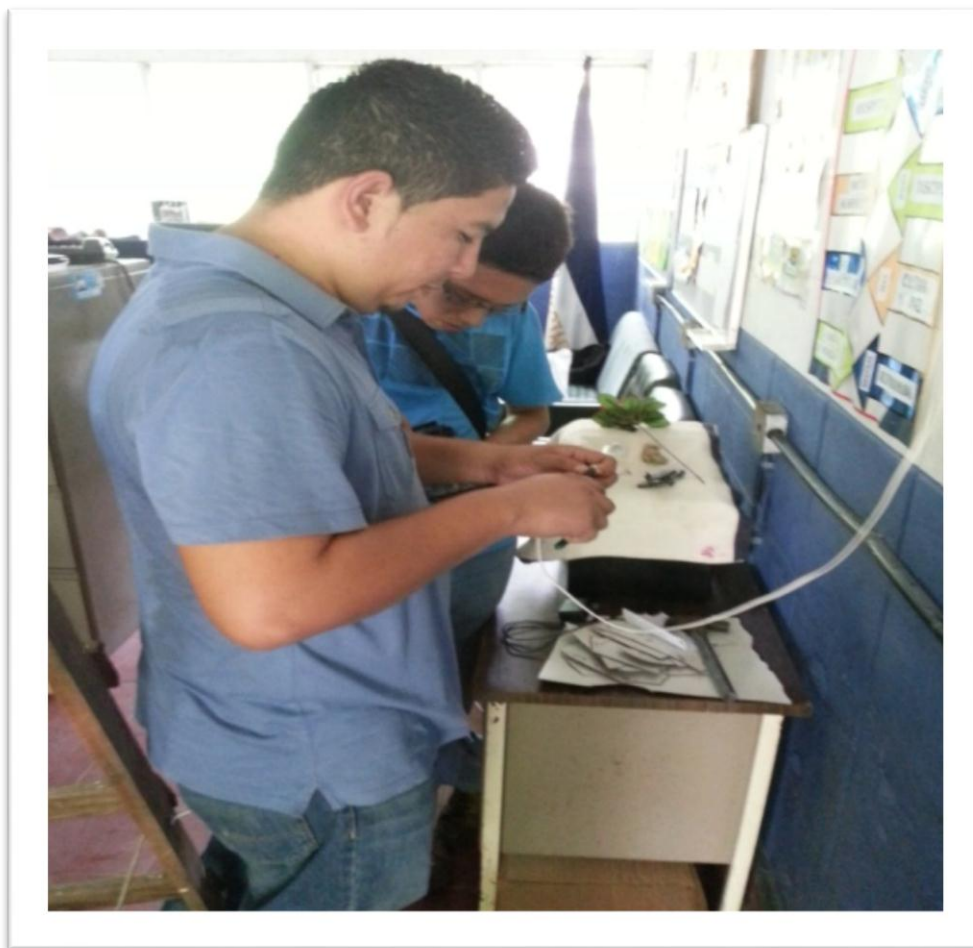
A1. Matriz de Congruencia

MATRIZ DE CONGRUENCIA		
TEMA: "Implementación de un Sistema Domótico con tecnología Android para el temporizado de las actividades del Complejo Educativo Marcos Ochoa".		
ENUNCIADO DEL PROBLEMA:		
OBJETIVO GENERAL: Implementar un Sistema Domótico con tecnología Android para el temporizado de las actividades del Complejo Educativo Marcos Ochoa y así brindar un mayor confort		
OBJ.ESPECIFICO 1:	OBJ.ESPECIFICO 2:	OBJ.ESPECIFICO 3:
Elaborar un circuito electrónico que sea capaz de llevar el temporizado de las actividades del Complejo Educativo Marcos Ochoa y a la misma vez active el timbre según las horas establecidas.	Elaborar un firmware o programa que permita que el circuito funcione adecuadamente y a la misma pueda recibir información enviada por medio de una aplicación de sistema android, instalada en un teléfono móvil o Tablet con dicho sistema.	Integrar en un solo prototipo el Hardware y Software para la realización del proyecto final y funcional.
ALCANCE 1:	ALCANCE 2:	ALCANCE 3:
Diseñar un dispositivo electrónico automatizado el cual controle el temporizado de las actividades del Complejo Educativo Marcos Ochoa	Diseñar un software el cual controle el sistema electrónico que llevara el temporizado de las actividades del Complejo Educativo	Integrar un sistema Domótico que adapte el circuito electrónico con el software y aplicación Android la cual enviara información vía bluetooth para su correcto funcionamiento
PRODUCTO 1:	PRODUCTO 2:	PRODUCTO 3:
Un sistema electrónico capaz de llevar el temporizado de las actividades del complejo educativo.	Software compilado funcional el cual controle todos los dispositivos electrónicos para que puedan cumplir su función en el sistema electrónico	Proyecto funcional con el programa compilado y una aplicación android con la cual podrán modificar los horarios.
DOCUMENTACIÓN TÉCNICA:		
ARDUINO UNO: Microcontrolador: ATmega328 Tensión de funcionamiento: 5V Voltaje de entrada (recomendado): 7-12V Voltaje de entrada (límites): 6-20V Digital I/O Pins: 14 (de los cuales 6 proporcionan PWM) Pines de entrada analógica: 6 Corriente continua para las E/S Pin: 40 mA Corriente de la CC para Pin 3.3V: 50 mA Memoria Flash: 32KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque: Tensión de funcionamiento: 5V Voltaje de entrada (recomendado): 7-12V Voltaje de entrada (límites): 6-20V Digital I/O Pins: 14 (de los cuales 6 proporcionan PWM) Pines de entrada analógica: 6 Corriente continua para las E / S Pin: 40 mA Corriente de la CC para Pin 3.3V: 50 mA Memoria Flash: 32 KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque SRAM: 2 KB (ATmega328) EEPROM: 1 KB (ATmega328) Velocidad del reloj: 16 MHz MODULO BLUETOOTH: Puede ser usado como un puerto serial Bluetooth, reemplazo directo de una conexión serial cableada, el uso es transparente. Ideal para proyectos embebidos con microcontroladores, módulos GPS y otros dispositivos de comunicación mediante UART. Chip CSR Bluetooth, compatible con el protocolo Bluetooth V2.0 Voltaje de operación 3.3V BaudRateajustable: 200400, 4800, 9600,19200, 38400, 57600, 115200 - Tamaño: 28mm x 15 mm x 2.35mm - Corriente de operación 40mA - Corriente en modo de descanso< 1Ma MODULO RELÉ: El control de todos los canales de este módulo se realiza a través de los pines E / S digitales 5, 6, 7 y 8 de la placa Arduinosobre la que va montado este shield. De todos modos, es necesaria una fuente de alimentación externa de 12 V CC para alimentar correctamente a las bobinas de cada relé. Esta tensión de 12VDC viene señalizada por el encendido del LED PWR rojo situado junto al bloque de conexiones o clemas correspondiente. PANTALLA LCD: El LCD 16x2 (LCD 2x16) tiene en total 16 pines (tome en cuenta que la posición correcta del displays con lospines en la parte superior, aunque existen modelos en los que la posición correcta es con los pines en la parte inferior) . RELOJ RTC DS1307: Ds1307 i2c el reloj de tiempo real chip(rtc) 24 c32 32 k i2c la memoria eeprom lir2032 la batería de litio recargable, y hacerse cargo del circuito ds1307 solución con la batería de nuevo- hasta no pueden leer o escribir. Lleno de electricidad, puede proporcionar la sincronización ds1307 1 año. Diseño compacto, mm 27* 28* mm mm 8.4 sacar ds1307 pin de reloj, proporcionar la señal del reloj para el microordenador chip único. Puede conectar en cascada el otro i2c dispositivos. TIMBRE: Timbre eléctrico con martillo, 230v. 10cm. gran sonoridad.		
PROYECTO TEMÁTICO: Implementación de un sistema Domótico con tecnología Android para el temporizado de las actividades del Complejo Educativo Marcos Ochoa.		
DETALLE DEL PRESUPUESTO PROYECTADO:		
ELEMENTOS PRINCIPALES		
MICROCONTROLADOR	RELOJ RTC DS1307	MODULO BLUETOOTH
\$3.36	\$7.00	\$15.00
OFERTA ECONÓMICA:		
ELEMENTOS PRINCIPALES		
MICROCONTROLADOR	RELOJ RTC DS1307	MODULO BLUETOOTH
\$3.36	\$7.00	\$15.00

A2. A continuación se exponen imágenes de la instalación del proyecto.



En la imagen se observa cómo se está trabajando para la instalación del Sistema



En esta figura Se muestra como se prepara el cableado para la Instalación.



En esta fotografía se observa como se está terminando de instalar el sistema.



En esta figura se muestra ya finalizada la instalación del sistema.