



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN

CARRERA: ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA: Diseño e Implementación de prototipo para la simulación de presencia humana mediante el control automático de la iluminación residencial.

AUTOR: Adrián Luis León Arroba

TUTOR: Lic. René Cañete Mg.PhD.

2015

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Graduación certifico:

Que el trabajo de graduación **“DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO PARA LA SIMULACIÓN DE PRESENCIA HUMANA MEDIANTE EL CONTROL AUTOMÁTICO DE LA ILUMINACIÓN RESIDENCIAL”**, presentado por el Sr. Adrián Luis León Arroba, estudiante de la carrera de Electrónica y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe para su correspondiente estudio y calificación.

Quito D.M. Febrero de 2015

TUTOR

.....

PhD. René Cañete

UNIVERSIDAD TECNOLÓGICA ISRAEL

AUTORÍA DE TRABAJO DE TITULACIÓN

El abajo firmante, en calidad de estudiante de la Carrera de Electrónica y Telecomunicaciones, declaro que los contenidos de este Trabajo de Titulación, requisito previo a la obtención del Grado de Ingeniería en Electrónica y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito D.M Febrero 2015

.....

Adrián Luis León Arroba

C.C.: 0603059932

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado, aprueban el Proyecto Integrador de Carrera de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

Quito D.M., Febrero del 2015

Para constancia firman:

TRIBUNAL DE GRADO

.....

PRESIDENTE

.....

MIEMBRO 1

.....

MIEMBRO 2

AGRADECIMIENTO

Agradezco a Dios por la vida, por darme fortaleza y la salud para no desmayar en la culminación de mi carrera.

A la Universidad Tecnológica Israel por brindarme la oportunidad de continuar de crecer profesionalmente.

Agradezco a mi tutor el PhD. René Cañete, por su guía en el desarrollo del proyecto y a todos los profesores por compartir sus conocimientos y experiencia.

Y a todas aquellas personas de una u otra manera colaboraron en la realización de este proyecto.

Gracias a todos

DEDICATORIA

El presente proyecto lo quiero dedicar a mi familia, a los cuales en ciertos momentos no les pude brindar mi apoyo y cuidado por ausentarme debido a mis estudios.

A mi esposa Marisol por brindar sus cuidados y abnegación hacia nuestros hijos.

A Joel y Felipe, mis hijos, por ser mi motivación e inspiración en mi vida.

A mis padres Eugenia y Luis, por los valores inculcados en mí, permitiéndome culminar mi carrera profesional.

ÍNDICE GENERAL

TRABAJO DE TITULACIÓN.....	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA DE TRABAJO DE TITULACIÓN	iii
APROBACIÓN DEL TRIBUNAL DE GRADO	iv
AGRADECIMIENTO.....	v
DEDICATORIA.....	vi
ÍNDICE GENERAL	vii
ÍNDICE DE GRÁFICOS.....	ix
ÍNDICE DE TABLAS	xi
INTRODUCCIÓN	xii
CAPÍTULO I	1
FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO PARA LA SIMULACIÓN DE PRESENCIA HUMANA.....	1
1.1 Introducción	1
1.2 Descripción general de los componentes del sistema.....	2
1.3 Comunicación Inalámbrica.....	3
1.4 Sistema Global para comunicaciones Móviles (GSM).....	5
1.5 Tecnología Arduino.....	5
1.6 SHIELDS	9
1.7 Descripción general del funcionamiento del sistema.....	9
1.8 Alcance del proyecto.....	11
1.9 Productos similares existentes en el mercado	12
CAPÍTULO 2	16
DESCRIPCIÓN DEL PROCESO INVESTIGATIVO Y METODOLOGÍA	16
2.1. Problema Principal.....	16

2.2. Descripción de los Objetivos	16
2.3. Hipótesis	16
2.3.1 Variable Independiente.....	16
2.3.2 Variable dependiente.....	17
2.4 Descripción de las teorías de fundamentación del Proyecto	17
2.5 Metodología Utilizada	17
CAPÍTULO 3	18
PRESENTACIÓN Y DESCRIPCIÓN DEL PROTOTIPO PARA LA SIMULACIÓN DE PRESENCIA HUMANA	18
3.1 Requerimientos planteados	18
3.2 Diseño Electrónico	18
3.2.1 Diseño de hardware	19
3.2.2 Diseño de software.....	32
3.3. Instalación en la maqueta.	51
3.4 Pruebas y Resultados	53
3.5 Costos	56
CONCLUSIONES Y RECOMENDACIONES	58
Conclusiones	58
Recomendaciones	59
BIBLIOGRAFÍA	60
ANEXO A1: Manual de Usuario.....	61
ANEXO A.2: Programa para la simulación de presencia	68

ÍNDICE DE GRÁFICOS

Figura 1. Uso de TIC's en los hogares del Ecuador	4
Figura 2. Servicios de las redes inalámbricas de telecomunicaciones.....	4
Figura 3. Red GSM.	5
Figura 4. Descripción de pines de la tarjeta Arduino Uno	8
Figura 5. Partes de una tarjeta Arduino Uno.....	8
Figura 6. SHIELDS para Arduino Uno	9
Figura 7. Componentes de hardware del sistema.....	11
Figura 8. Diagrama HDL Automatización Inteligente	13
Figura 9. Arquitectura sistema LonWork.....	14
Figura 10. Arquitectura KNX.....	14
Figura 11. Sistema de Seguridad House Lock.....	15
Figura 12. Diagrama de bloques del sistema.....	19
Figura 13. Composición de un Relé y circuito de disparo.	20
Figura 14. Conexión del Sistema de Relés a un circuito de iluminación manual residencial.....	21
Figura 15. Circuito de opto acoplamiento del relé y módulo de relés de 8 canales de 5VDC.....	22
Figura 16. Conexión de la parte eléctrica y a la tarjeta Arduino uno del módulo de Relés	22
Figura 17. Módulo SIM900 Vista frontal, posterior y descripción de pines.	23
Figura 18. SIMCOM SIM900 GSM. Vista frontal, posterior e identificación de partes.	24
Figura 19. Configuración de jumper para comunicación serial a través de los pines D0 y D1. ..	25
Figura 20. Puente de suelda que activa el jumper para control de activación del Módulo GSM.	25
Figura 21. Conexión del módulo GSM a la tarjeta Arduino, zócalo de inserción de chip GSM y plug antena	26
Figura 22. Distribución de pines y circuito típico de utilización del chip DS1302.....	27
Figura 23. Módulo DS1302.....	27
Figura 24. Conexión del Módulo RTC a la tarjeta Arduino.	28
Figura 25. Circuito integrado receptor FM Philips TEA5767 y circuito de utilización típico.....	29
Figura 26. Módulo Receptor FM, conexión al módulo Arduino Uno, vista frontal y posterior.....	30
Figura 27. Placas de conexión elaboradas y cables.	31
Figura 28. Esquema de conexión del proyecto.....	32
Figura 29. Diagrama de flujo propuesto.....	33
Figura 30. Diagrama de flujo del modo automático.....	40

Figura 31. Código y descripción del proceso de extracción del número y mensaje de texto	45
Figura 32. Diagrama de Flujo de Modo Remoto	47
Figura 33. Plano y foto de la maqueta realizada.....	52
Figura 34. Esquema y foto de conexión y distribución eléctrica en la maqueta	53
Figura 35. Pruebas de Relés y módulo GSM.....	54

ÍNDICE DE TABLAS

Tabla 1. Características de algunos de los modelos Arduino.	7
Tabla 2. Características del Módulo 5v 76-108mhz Tea5767 Fm Stereo Radio para Arduino. ...	28
Tabla 3. Mapa de utilización de memoria EEPROM del Microcontrolador ATmega328P.....	34
Tabla 4. Almacenamiento de números celulares en la memoria EEPROM.....	35
Tabla 5. Comandos para la comunicación con el módulo RTC utilizados de la librería virtuabotixRTC.h.....	36
Tabla 6. Horas de encendido y apagado de ambientes en modo automático	37
Tabla 7. Asignación de horarios de encendido apagado en la memoria EEPROM.....	38
Tabla 8. Comandos AT Básicos.	41
Tabla 9. Descripción comandos AT para recepción de mensajes SMS utilizados.	42
Tabla 10. Ejemplo de comandos SMS.....	43
Tabla 11. Comandos establecidos para el control a través de mensajes de texto en modo remoto.....	44
Tabla 12. Descripción de la conexión y asignación de Ambientes entre Tarjeta Arduino y Módulo de Relés.....	48
Tabla 13. Ejemplo de programación de prendido y apagado de la Sala en modo remoto y automático.....	49
Tabla 14. Tabla de conversión de tiempos de 24 horas a 12 minutos	55
Tabla 15. Costos parte eléctrica y electrónica de etapa de control	56
Tabla 16. Costos del prototipo, maqueta y sistema eléctrico de iluminación.....	57

INTRODUCCIÓN

En la actualidad el país vive un nivel de seguridad mínimo, debido a los índices de delincuencia, robos, atracos, etc. Muchas personas temen de sufrir este tipo de siniestros sean en la calle, trabajo, o en las mismas casas.

Uno de este tipo de atracos son los que se perpetran a los domicilios en horas nocturnas, en los momentos que las residencias no están habitadas, ya sea porque los miembros del hogar están de vacaciones, compromisos sociales, o están fuera de la ciudad, etc. Para la ejecución del atraco los delincuentes previamente realizan un seguimiento a la casa para asegurarse que la vivienda esté completamente deshabitada.

Sin embargo para controlar este tipo de robos la ciudadanía, toma diferentes medidas de control con el afán de disminuir el riesgo de hurto en sus domicilios, con acciones como requerimiento de patrullajes policiales, solicitar a vecinos o familiares que resguarden su vivienda, desamparan sus domicilios por pocas horas o en el peor de los casos algún miembro del hogar opta por no abandonar su vivienda, por temor a que sufra algún tipo de atraco.

Existen otro tipo de resguardo, el cual recurre a sistemas electrónicos, existiendo en el mercado varias opciones de seguridad como las alarmas residenciales, monitoreo y patrullaje residencial, guardianía privada, circuitos cerrados de video, cercas eléctricas, entre otros, que si bien es cierto todas estas opciones dan un mayor nivel de seguridad, sin embargo tienen un costo más elevado lo que lo pone fuera del alcance de hogares de clase media o inferior, e incluso la delincuencia ya es experta en evadir a este tipo de protecciones, algunas de ellas quitan privacidad o pueden generar riesgos para las mismas personas que habitan en la residencia.

Todo esto es derivado por falta de un sistema de simulación de presencia en los domicilios con el objetivo de disuadir de la intrusión de personas ajenas, hacia el interior del domicilio.

Varias de las opciones anteriormente citadas se aplican cuando la delincuencia ya ha ingresado al domicilio, sin embargo siempre es mejor prevenir que ocurra el siniestro, una de estas formas de prevención es la simulación de habitabilidad de la casa, lo que da la impresión de que el interior del domicilio existe presencia humana.

En la actualidad en el ámbito residencial, la utilización de sistemas de simulación tiene una gran auge, debido a que permiten a través del control remoto por medio de diferentes medios del encendido y apagado de las luces interiores y exteriores de una residencia lo que genera para las personas ajenas la sensación de presencia dentro del hogar lo que reduce la posibilidad de ingreso de delincuentes.

OBJETIVOS

Objetivo general:

Construir e implementar un prototipo para la simulación de presencia humana, mediante un control automático de la iluminación residencial, con accionamiento remoto, que sea de bajo costo.

Objetivos específicos:

- Determinar las etapas y los dispositivos electrónicos que se utilizara en cada fase del proceso de simulación, mediante el análisis y tipos de señales con las que se van a trabajar.
- Comprobar el principio de funcionamiento y programación de módulos GSM para el envío y recepción de mensajes de texto.
- Programar la secuencia de activación de encendido, apagado y temporización del circuito de iluminación, que pueda ser controlado remotamente.
- Implementar el prototipo con el circuito electrónico de control automático y activación remota de simulación de presencia a través del sistema de iluminación.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO PARA LA SIMULACIÓN DE PRESENCIA HUMANA

1.1 Introducción

Aspectos como seguridad y bienestar, son conceptos que están enmarcados en el Plan Nacional del Buen Vivir 2013-2017 donde se plantea: “Consolidar la transformación de la justicia y fortalecer la seguridad integral, en estricto respeto a los derechos humanos.” (SENPLADES, 2013. Pg. 17) dentro de lo cual se plantean aspecto de seguridad ciudadana y de su habitad, que se promoverán ambientes de seguridad y bienestar dentro del hábitat en el cual el ciudadano realiza sus actividades, dentro de lo cual su hogar es uno de sus principales hábitats, lo que justifica la necesidad de implementar este tipo de proyecto que tiene como propósito el generar un ambiente seguro para los bienes y personas que habitan en un hogar, por medio de la simulación de su presencia, en el mismo a través de un sistema que puede funcionar en un modo automático, en el cual se tiene la posibilidad de controlar el encendido y apagado de diferentes artefactos en un horario previamente establecido, y un modo remoto en el cual el propietario puede desde su celular controlar el encendido o apagado de los aparatos, lo cual tiende a mejorar la calidad de vida de la población y adicionalmente tiene implícito el tránsito hacia la sociedad de la información y el conocimiento; considerando el uso de las Tecnologías de Información (TIC) que también es parte de Plan Nacional del Buen Vivir.

Con respecto a la investigación se establece que la misma se realiza en las universidades y debe transformarse en uno de los principales puntales de la transformación de la economía primaria exportadora, lo que ratifica el hecho de aplicar conceptos de innovación en proyectos desarrollados dentro de las universidades y que estos apunten a satisfacer una necesidad de la sociedad en busca de su progreso y desarrollo.

El presente trabajo de titulación por lo tanto se alinea con las estrategias y planes planteados por el Gobierno Nacional y se orienta a plantear una opción que permita incrementar el nivel de seguridad de un hogar encaminado a resolver un problema de seguridad residencial a

través de un concepto de disuasión más que de acción ofreciendo un medio de comunicación y control basado en el uso de tecnologías de la comunicación.

1.2 Descripción general de los componentes del sistema

El proyecto será elaborado con elementos diseñados con tecnologías actuales y disponibles en el mercado, lo que garantiza su disponibilidad en caso de ser requerido, adicionalmente, son programables lo que permite ofrecer un producto adaptable y modificable a las necesidades de cada aplicación y residencia, se emplea un concepto modular, lo que permite ofrecer un concepto de escalabilidad permitiendo futuras evoluciones e inserción de nuevos componentes que le den un mayor nivel de funcionalidad, finalmente se pretende ofrecer un producto que brinda características similares a productos ya existentes en el mercado, pero a un precio mucho menor, debido a que se está empleando para su desarrollo herramientas de hardware y software que permiten ofrecer un producto de calidad con mano de obra e intelecto nacional, lo cual lo diferencia de productos similares existentes en el mercado que incrementan su costo por ser sistemas desarrollados en el exterior y que vienen en la mayoría de los casos listos para su configuración e instalación.

Se combinarán componentes y tecnologías diversas dentro de un solo sistema como son:

- Módulo Global System for Mobile Communications o Sistema Global para Comunicaciones Móviles (GSM), encargado de la comunicación a través de sistema de mensajería SMS.
- Módulo Real Time Clock o Reloj en Tiempo Real (RTC), a cargo del manejo de temporizaciones del sistema.
- Módulo de Relés, empleado para la conexión con los elementos de potencia, en su mayoría luces.
- Módulo Arduino, que se constituye en el cerebro del sistema y quien maneja todo el sistema.
- Módulo Radio FM, empleado para generar un sistema de audio como parte adicional del sistema que complementa la simulación de presencia.
- Fuente de Poder, que alimenta a todos los componentes del sistema con 5VDC.

Todos los componentes con los que va a ser implementado el sistema, cumplirá normas de fabricación técnicas y ambientales, y al emplear componentes enteramente electrónicos que no consumen combustibles, ni generar ningún tipo de emisiones tóxicas, el sistema puede

considerarse como amigable con el medio ambiente, además al no tener como propósito del mismo el activar ningún tipo de alarma sonora, el mismo no genera contaminación auditiva, lo que lo constituye en un sistema completamente amigable con el hábitat en el cual sea instalado, en cuanto a su consumo será muy bajo, por la energía disipada por la parte electrónica y en general el resto dependerá el tipo de sistemas que sean controlados a partir del mismo.

1.3 Comunicación Inalámbrica

El desarrollo de aplicaciones y uso de tecnologías móviles dentro del campo de las telecomunicaciones han provocado una completa revolución en el ámbito de las comunicaciones, pasando el teléfono celular a constituirse en un medio de acceso permanente a un conjunto de servicios y aplicaciones y lo han convertido en algunos casos en el medio más utilizado por el usuario para comunicarse con otras personas, acceder a información y otros servicios.

“El 16,9% (1'261.944) de las personas de cinco años y más que tienen celular poseen un teléfono inteligente (Smartphone), lo que representa un crecimiento de 141% frente al 2011, según los últimos datos de la Encuesta de Tecnologías de la Información y la Comunicación (TIC) del Instituto Nacional de Estadística y Censos (INEC).

El estudio, que se realizó en diciembre de 2013, se hizo en 21.768 hogares a personas de 5 años y más, a nivel nacional, regional, provincial, de nivel urbano y rural.

Según la encuesta, el 51,3% de la población de 5 años y más tiene por lo menos un celular activado, en el 2011 ese porcentaje era del 46,6%.

Por edades, el grupo etario con mayor uso de teléfono celular activado es la población que se encuentra entre 25 y 34 años con el 76,5%, seguido de los de 35 a 44 años con el 76%.” (INEC, 2014)

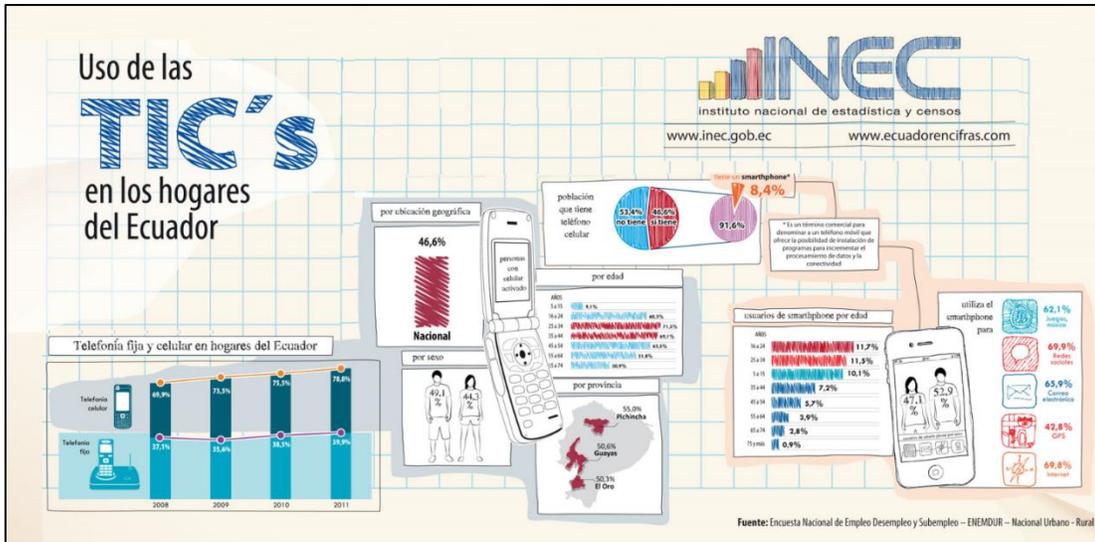


Figura 1. Uso de TIC's en los hogares del Ecuador.

Fuente: El Autor

Dentro de la cada vez más grande lista de servicios que ofrecen las comunicaciones móviles en la actualidad, se destacan: Enviar y recibir llamadas de voz, Short Message Service (SMS) Servicio de mensajes cortos o simples, navegación por internet, acceso a correos electrónicos y redes sociales, videoconferencias, acceso remoto a equipos, bases de datos, acceso a múltiples aplicaciones y servicios complementarios, entre otros.



Figura 2. Servicios de las redes inalámbricas de telecomunicaciones.

Fuente: El Autor

Existen varias tecnologías desarrolladas y aplicadas a nivel mundial en este aspecto, pero, en el país la que se emplea de forma exclusiva la tecnología Global System for Mobile

communications o Sistema Global para comunicaciones Móviles (GSM), la cual es utilizada por las tres operadoras móviles existentes en el país actualmente, como son: Claro, Movistar y CNT.

1.4 Sistema Global para comunicaciones Móviles (GSM)

El Sistema Global para las comunicaciones Móviles, GSM, comprende la segunda generación de telefonía celular, formado por un conjunto de dispositivos de red, que brindan una mayor velocidad de transmisión que la primera generación, las bandas de frecuencia en las cuales puede operar son 850, 900, 1800 y 1900 Mhz.

En el estándar GSM, se puede transmitir mensajería tipo correo electrónico, video, audio, mensajería SMS, así como también el servicio de roaming internacional.

Su operación se realiza mediante un chip o SIM, en el cual está contenida toda la información celular, usuarios y claves de identificación para acceder a la red móvil, número, contactos, etc.

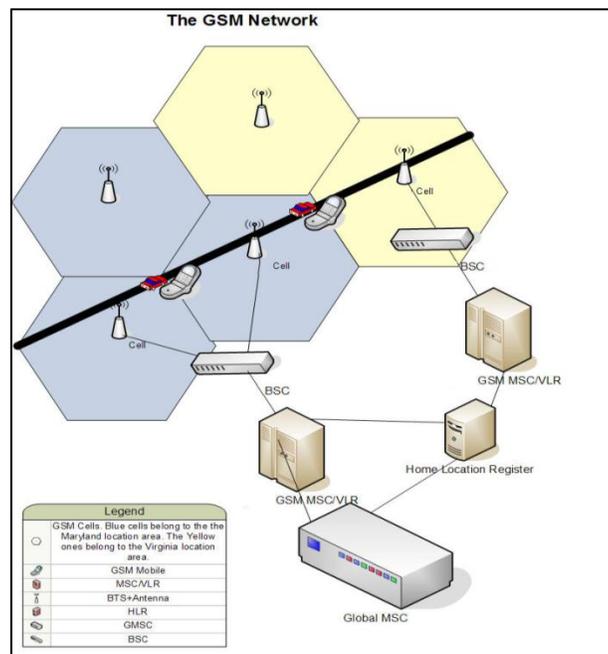


Figura 3. Red GSM.

Fuente: Comunidad Movistar – Funcionamiento Red GSM (2013)

1.5 Tecnología Arduino

Arduino está compuesto de tres elementos:

“Una placa hardware libre que incorpora un microcontrolador reprogramable y una serie de pines-hembra (los cuales están unidos internamente a las patillas de E/S del microcontrolador) que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores y actuadores.” (Torrente, 2013. Pág. 63)

Entendiendo por placa hardware a una tarjeta de circuito impreso las cuales existen en varios modelos concretos dentro de la tecnología Arduino y utilizan una misma familia de microcontroladores AVR de la marca Atmel. Esta placa apareció en el año 2003 como un proyecto personal de Hernando Barragán, estudiante del Instituto de Diseño de Ivrea, lugar donde surgió en 2005 la placa Arduino.

“Un software (más en concreto, un “entorno de desarrollo”) gratis, libre y multiplataforma (ya que funciona en Linux, MacOS y Windows) que debemos instalar en nuestro ordenador y que nos permite escribir, verificar y guardar (“cargar”) en la memoria del microcontrolador de la placa Arduino el conjunto de instrucciones que deseamos que este empiece a ejecutar. Es decir: nos permite programarlo. La manera estándar de conectar nuestro computador con la placa Arduino para poder enviarle y grabarle dichas instrucciones es mediante un simple cable USB, gracias a que la mayoría de placas Arduino incorporan un conector de este tipo.” (Torrente, 2013. Pág. 64)

“Un lenguaje de programación libre. Por “lenguaje de programación” se entiende cualquier idioma artificial diseñado para expresar instrucciones (siguiendo unas determinadas reglas sintácticas) que pueden ser llevadas a cabo por máquinas. Concretamente dentro del lenguaje Arduino, encontramos elementos parecidos a muchos otros lenguajes de programación existentes (como los bloques condicionales, los bloques repetitivos, las variables, etc.), así como también diferentes comandos, así mismo llamados “órdenes” o “funciones” que nos permiten especificar de una forma coherente y sin errores las instrucciones exactas que queremos programar en el microcontrolador de la placa. Estos comandos los escribimos mediante el entorno de desarrollo Arduino.” (Torrente, 2013. Pág. 65).

A partir de lo establecido dentro del entorno del desarrollo Arduino, estos presentan las siguientes ventajas:

- Arduino es libre y extensible
- Arduino tiene una gran comunidad

- Su entorno de programación es multiplataforma
- Su entorno y el lenguaje de programación son simples y claros.
- Las placas Arduino son baratas.
- Las placas Arduino son reutilizables y versátiles.

En el caso de la tarjeta Arduino Uno, utilizada en este proyecto, utiliza como microcontrolador el ATmega328P caracterizados por un bajo consumo de corriente, debido a ser fabricados con tecnología Picopower de Atmel. A su vez las características técnicas de algunos de los modelos Arduino se resumen a continuación en la tabla.

Tabla 1. Características de algunos de los modelos Arduino. Fuente: <http://openhardware.pe/que-modelo-de-arduino-debo-comprar-para-mis-proyectos> (2013)

Modelo	Arduino UNO	Arduino Leonardo	Arduino Mega 2560	Arduino DUE
<i>Microcontrolador</i>	ATmega328	ATmega32u4	ATmega2560	AT91SAM3X8E
<i>Voltaje de Operación</i>	5V	5V	5V	3.3V
<i>Voltaje de entrada</i>	7-12V	7-12V	7-12V	7-12V
<i>Voltaje de entrada (limites)</i>	6-20V	6-20V	6-20V	6-20V
<i>Digital I/O Pines</i>	14	20	54	54
<i>Digital I/O Pines</i>	6	7	15	12
<i>Salidas PWM</i>				
<i>Entradas Analógicas Pines</i>	6	12	16	12
<i>Salidas Analógicas Pines</i>				2 (DAC)
<i>Flash Memory</i>	32 KB	32 KB	256 KB	512 KB available
	0.5 KB bootloader	4 KB bootloader	8 KB bootloader	
<i>SRAM</i>	2 KB (ATmega328)	2.5 KB	8 KB	96 KB Dos bancos: 64KB y 32KB
<i>EEPROM</i>	1 KB (ATmega328)	1 KB	4 KB	
<i>Frecuencia de Reloj</i>	16 MHz	16 MHz	16 MHz	84 MHz
<i>Tipo de USB</i>	Estandar	Mini	Estándar	Mini

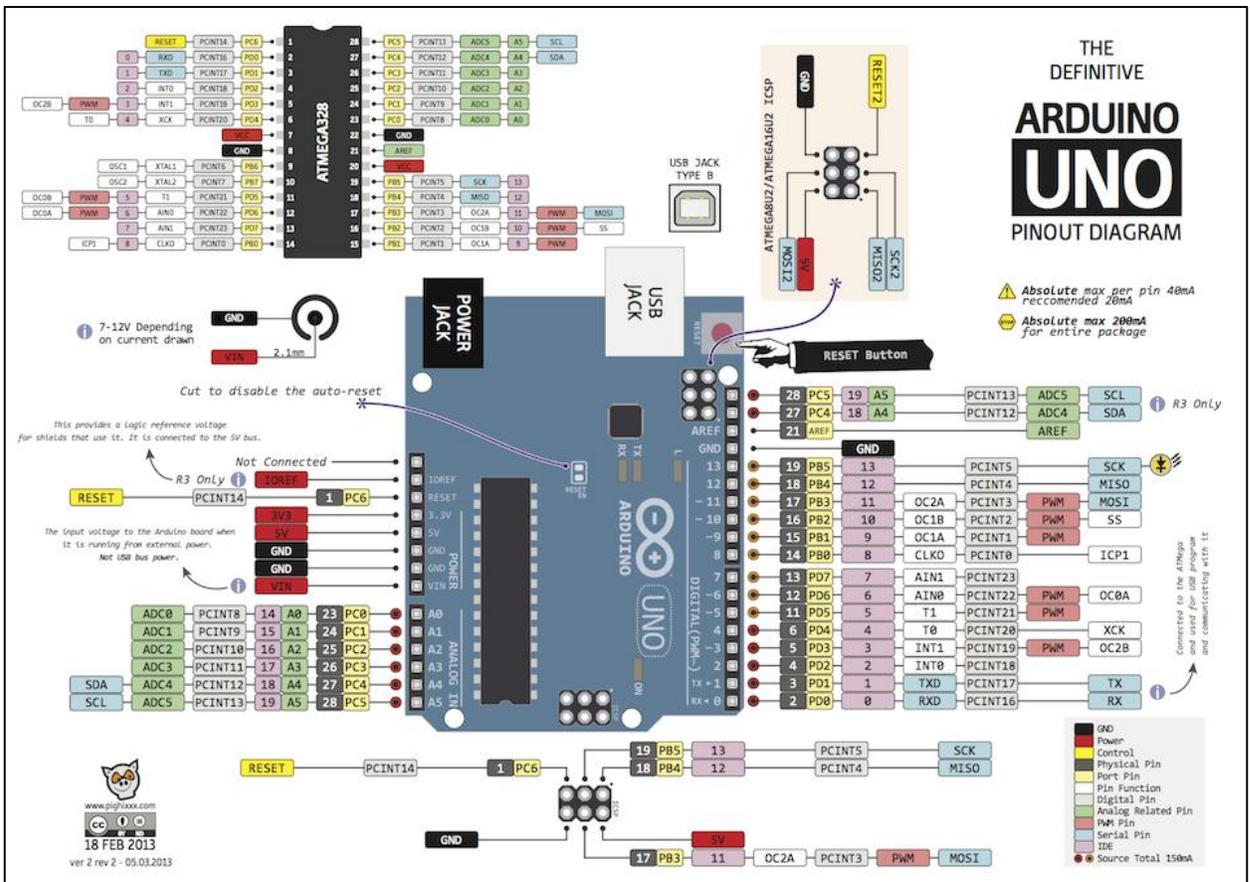


Figura 4. Descripción de pines de la tarjeta Arduino Uno
Fuente: www.pighixx.com – Pinout Diagrama (2013)

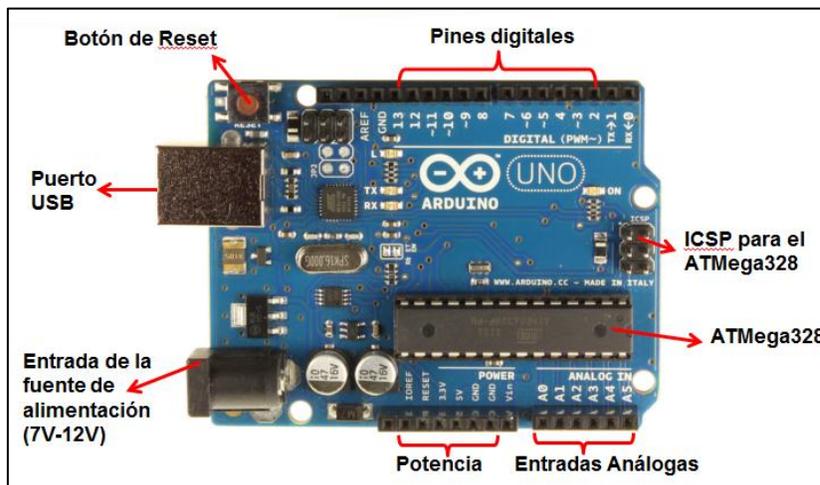


Figura 5. Partes de una tarjeta Arduino Uno
Fuente: El autor

1.6 SHIELDS

“Además de las placas Arduino propiamente dichas, también existen los llamados “shields”. Un “shield” (en inglés significa “escudo”) no es más que una placa de circuito impreso que se coloca en la parte superior de una placa Arduino y se conecta a ella mediante el acoplamiento de sus pines sin necesidad de ningún cable. Su función es actuar como placas supletorias, ampliando las capacidades y complementando la funcionalidad de la placa Arduino base de una forma más compacta y estable.” (Torrente, 2013. Pág. 117)

Las denominadas placas "SHIELD" son extensiones para placa Arduino (generalmente Arduino UNO o Arduino Pro 328) con las que se puede expandir las posibilidades de la placa, incrementar sus funcionalidades y adicionar opciones de programación basados en las características de la tarjeta Arduino original. Hay SHIELDS de todo tipo, entre los cuales se destacan: para módulos GSM, relés, comunicación con XBee, Bluetooth, GPS entre otros, con un nivel cada vez más mayor de integración.

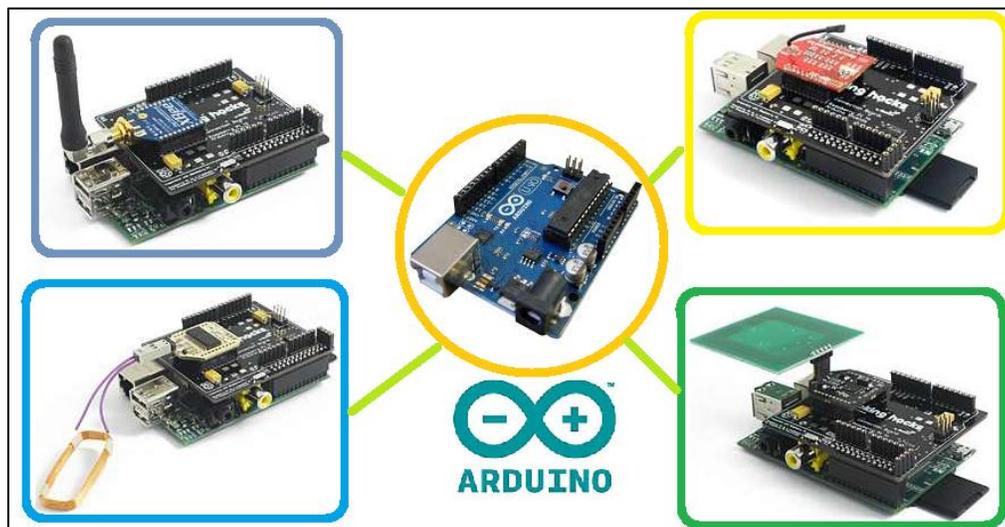


Figura 6. SHIELDS para Arduino Uno
Fuente: El autor

1.7 Descripción general del funcionamiento del sistema

El sistema propuesto plantea ofrecer un producto con las siguientes características de funcionamiento:

- El sistema cuenta con dos formas de funcionamiento: una temporizada y otra remota.
- En modo temporizado el sistema basa su funcionamiento un sistema de reloj en tiempo real el cual informa la hora del día, la misma que será empleada para encender o apagar los equipos en base a su comparación con horas previamente establecidas y almacenadas en la memoria EEPROM del microcontrolador de la tarjeta Arduino para cada dispositivo.
- En modo remoto el control, activación y desactivación de equipos y componentes eléctricos de una casa conectados al sistema se realiza a través de un sistema de comunicación basado en mensajes de texto simple (SMS), los mismos que contendrán comandos previamente establecidos que permitan encender o apagar los aparatos de una casa y cambiar el modo de funcionamiento de temporizado a remoto y viceversa, los mensajes aceptados para esto deben provenir de teléfonos celulares previamente registrados y almacenados en la memoria EEPROM del microcontrolador de la tarjeta Arduino.
- El número de equipos conectados al sistema en esta primera versión es de 8 equipos eléctricos, todos los cuales son de tipo ON/OFF, cuyas características de consumo inicialmente son de hasta 10 A con una alimentación de 120V, dato que es tomado de las características de la tarjeta SHIELD de relés del sistema.
- Adicionalmente, el sistema contara con un sistema de audio operado a partir de un receptor FM que permite incorporar al sistema la posibilidad de simulación de presencia no solo por la parte visual, que sería a partir de control de iluminación sino también a nivel de audio.

La siguiente figura describe los componentes de hardware del sistema, los mismos que serán controlados por la tarjeta Arduino uno, que tendrá un algoritmo de funcionamiento desarrollada en la plataforma de software Arduino.

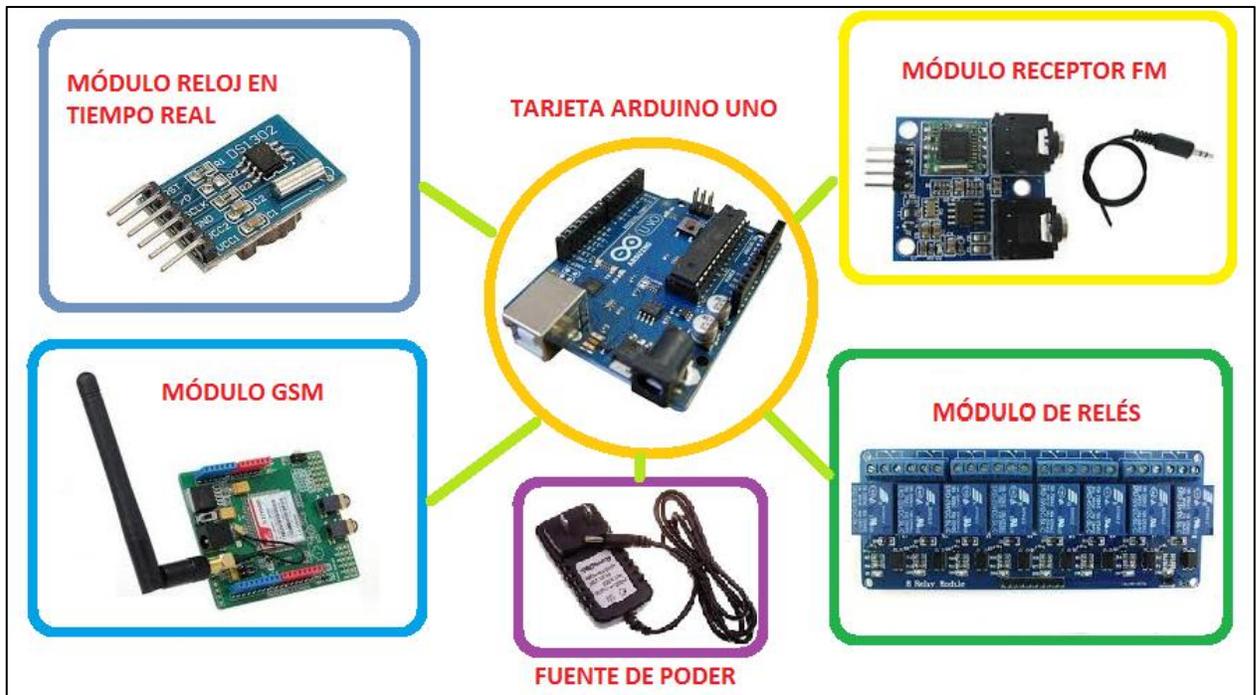


Figura 7. Componentes de hardware del sistema
Fuente: El autor

1.8 Alcance del proyecto

En base la descripción del funcionamiento del sistema planteado en la sección anterior, se establecen los siguientes alcances para el proyecto:

- Se implementará en un prototipo, instalado en una maqueta a escala de dimensiones 100cm X 65cm X 17cm aproximadamente, constituido de 8 áreas.
- Se emplearan lámparas incandescentes de 120V/7W AC para representar la iluminación de cada área.
- Se colocará un panel de control en un extremo de la maqueta donde se instalara la parte electrónica del sistema y se realizaran cableados a cada lámpara y el interruptor correspondiente.
- El tiempo establecido para el modo automático será establecido en minutos y segundos para facilitar una rápida representación del funcionamiento, un día sería simulado en 24 minutos.

- Se implementara un sistema de audio a partir de un receptor FM que permitirá simular la presencia también a partir de generación de un sistema de sonido, a partir de 4 emisoras diferentes.
- El sistema no cuenta con sensores que realimenten el estado de los aparatos, por lo que de ser activados a partir de los interruptores internos no se podrá ejercer ningún tipo de control sobre los mismos.
- Al utilizarse SHIELDS el sistema es completamente modular y queda abierto para futuras mejoras e incremento de funcionalidades.

1.9 Productos similares existentes en el mercado

Como resultado de la investigación realizada se han encontrado varias empresas en el país que ofrecen equipos con prestaciones similares a los establecidos para este proyecto, la mayoría de ellos basados en tecnologías y equipos importados, los cuales son instalados y configurados, cuyos precios son bastante altos, dependiendo principalmente del país de origen de los equipos, siendo los de procedencia China los más económicos en este tipo, pero siendo a su vez de menor calidad y tiempo de garantía y presentado aún costos relativamente altos dentro del mercado local, lo cual los constituyen en sistemas de lujo dentro de hogares y residencias, obviamente estos equipos complementan su funcionamiento con una variedad amplia de servicios adicionales que van orientados hacia aplicaciones Domóticas, que si bien es cierto tienen un auge en el país aún no han captado un mercado amplio y se aplican para residencias de clase alta en su totalidad. Dentro de esto se destacan los sistemas ofrecidos por empresas como:

Soluciones Domóticas y Electrónicas (SODEL) que ofrecen equipos y soluciones domóticas a partir de equipos marca HDL para sistemas residenciales y comerciales, en esquemas modulares.

“El control del proyecto automatizado es flexible, el usuario puede seleccionar el panel LCD, el panel con botón táctil, interruptores normales, pantalla táctil a color, mando a distancia, lógica automática, programación temporizada, sistema telefónico y otras formas de controlar los diferentes ambientes con la combinación de uno o varios dispositivos. Los métodos específicos de control van de acuerdo con las necesidades reales de los usuarios. Los usuarios pueden elegir el diseño funcional.”(<http://hdlecuador.com>)

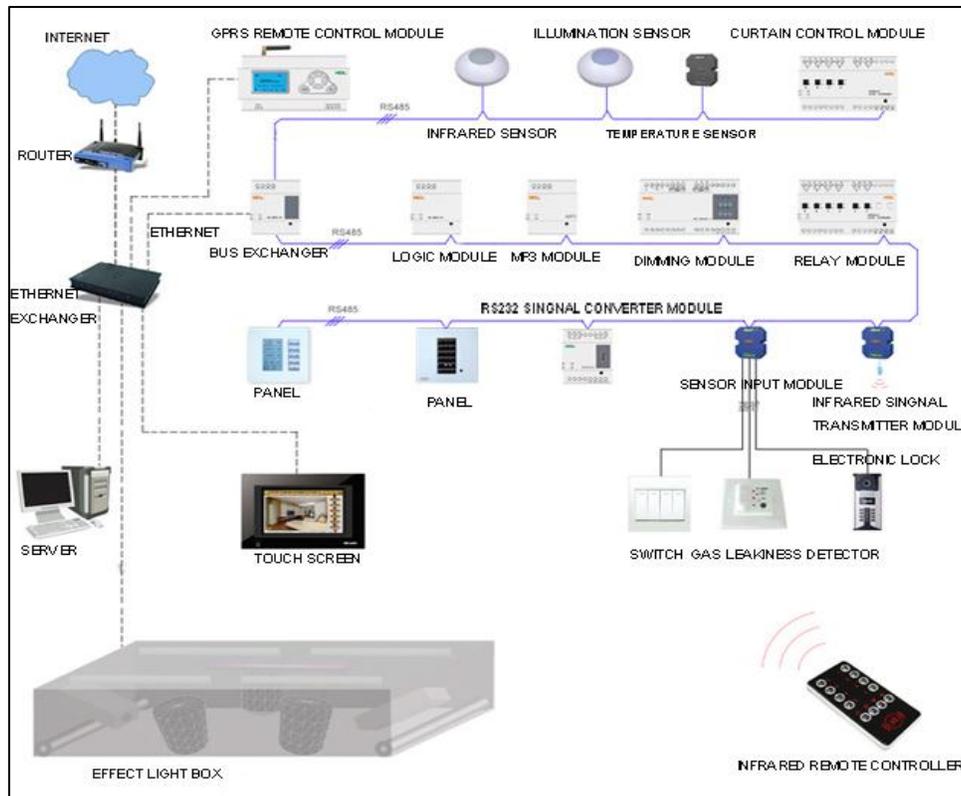


Figura 8. Diagrama HDL Automatización Inteligente
Fuente: HDL Ecuador – Soluciones (2013)

“ISDE ECUADOR es una empresa fundada en el año 2005 como filial para el Ecuador de ISDE España, con el objetivo primordial de importar y distribuir equipos electrónicos de control, destinados a la automatización de viviendas, edificios y alumbrado público, así como aportar soluciones avanzadas en redes de control”. (<http://www.isde-ecuador.com>).

La empresa ofrece sistemas domóticos a partir de equipos y tecnología basada en redes LonWork, un sistema de comunicación abierta con más de 1000 empresas fabricantes.



Figura 9. Arquitectura sistema LonWork
 Fuente: ISDE-Ecuador/Domotica/Domotica Centralizada (2013)

“Smart Controls es un estudio de Ingeniería Domótica ubicado en Quito - Ecuador. Realizamos los estudios, el diseño, ingeniería, instalación y mantenimiento de instalaciones domóticas, inmóticas y urbóticas. Instalamos sistemas de control de iluminación, seguridad contra intrusos, control de persianas, control de accesos, control de climatización, alumbrado público inteligente, medición eléctrica, ahorro energético.” (<http://www.smart-controls.com/>). Trabajan con equipos con tecnología LonWork y KNX.



Figura 10. Arquitectura KNX
 Fuente: <http://www.smart-controls.com> (2013)

Otro tipo de empresas en el país que ofrecen servicios similares a los planteados son las de seguridad como:

“CGB diseña e instala sistemas de seguridad electrónica modernos y confiables de acuerdo a las necesidades de su residencia o empresa. Si bien la seguridad física es muy importante, los altos costos laborales han encarecido los servicios de guardianía. Los sistemas de seguridad electrónica que diseñamos le permitirán reducir la necesidad de personal de seguridad física, el ahorro generado podrá cubrir la inversión en sus sistemas de seguridad electrónica en pocos meses.”(<http://cgbseguridad.com/>)

Dentro de la gama de servicios que ofrece la empresa están: Envío automático de los eventos principales generados por su sistema de alarma vía SMS y supervisión de conexiones y desconexiones con identificación individual de cada usuario.

House Lock sistema de seguridad que ofrece el control de aparatos y notificación de estado de alarma a través de sistema de mensajes SMS, como encender la alarma, encender o apagar aparatos eléctricos, comunicación con un sistema de monitoreo, el sistema es desarrollado con tecnología local y es el que más se asemeja el planteado.



Figura 11. Sistema de Seguridad House Lock
Fuente: <http://www.houselock.ec> (2013)

CAPÍTULO 2

DESCRIPCIÓN DEL PROCESO INVESTIGATIVO Y METODOLOGÍA

2.1. Problema Principal

Todas las personas en algún momento se han ausentado de sus viviendas ó tienen el anhelo de salir todos los miembros del hogar, por razones distintas, sin embargo en ausencia de los miembros del hogar existe el riesgo de que sus domicilios sean allanados por la delincuencia, y es en esos momentos que la gente anhela simular que sus hogares están habitados, dando la impresión que el domicilio existe presencia humana, con el propósito de disuadir y evitar que personas ajenas, intenten asaltar la vivienda. En ciertas ocasiones las personas prefieren evitar abandonar sus domicilios por el mismo temor; que si tuvieran un mecanismo que simule que la vivienda existe presencia humana, estos podrían ausentarse.

2.2. Descripción de los Objetivos

Los objetivos están enmarcados en la demostración de un sistema de simulación de presencia mediante la iluminación residencial, el mismo que pueda trabajar bajo una secuencia predeterminada ó bajo disposición del usuario, ambos de forma remota. Esto dará al usuario una disminución del riesgo de intrusión de personas ajenas a su domicilio en ausencia del propietario.

Por tanto los objetivos específicos están enmarcados con el propósito de alcanzar el objetivo principal, como es la determinación de etapas y dispositivos a utilizar, programación de dispositivos, validación del producto para su posterior implantación

2.3. Hipótesis

En la implementación del prototipo demostrativo de simulación de presencia humana mediante la iluminación residencial, se logrará reducir el riesgo de atraco al domicilio, siendo un sistema de bajo costo referencial en el mercado de la seguridad electrónica, de bajo consumo eléctrico ya que la tecnología a utilizar es de bajo nivel de voltaje y corriente; y económico en la operación por la activación mediante mensajería celular que tiene un valor muy económico.

2.3.1 Variable Independiente

La variable independiente es el sistema electrónico de simulación automática y remota, de presencia humana mediante el control de la iluminación residencial.

2.3.2 Variable dependiente

Disminuir el riesgo de robos a domicilios, activando la secuencia de encendido-apagado de las lámparas de la vivienda, con el propósito de disuadir a personas ajenas a la vivienda que intente ingresar al domicilio

2.4 Descripción de las teorías de fundamentación del Proyecto

El sistema de control electrónico de iluminación será la tecnología ARDUINO, siendo de última tecnología, la cual posee ventajas, como la eficiencia, economía y confiabilidad. Así también sus componentes externos como la comunicación móvil, etapa de control de potencia, reproducción de audio, y sincronización en tiempo real.

2.5 Metodología Utilizada

- En la selección y determinación de las etapas y dispositivos a utilizarse se requiere del método de análisis y síntesis, ya que se analizan los requerimientos y parámetros de funcionamiento del sistema, desde el punto de vista del usuario, para posteriormente pasar a un diseño inicial.
- Para la ésta fase del proyecto se recurre a la etapa de modelación, realizando bosquejo previo, y del diseño definitivo, reconociendo en cada etapa del diseño del proyecto sus características requeridas para el cumplimiento de los parámetros funcionales del proyecto en su conjunto.
- En la siguiente etapa se recurre al método de observación y experimentación, aplicando cada etapa del proyecto de forma secuencial e individual, para llegar a acoplar todas las etapas del sistema y experimentar la funcionalidad del mismo, siendo en esta etapa la verificación y validación del producto terminado.

CAPÍTULO 3

PRESENTACIÓN Y DESCRIPCIÓN DEL PROTOTIPO PARA LA SIMULACIÓN DE PRESENCIA HUMANA

3.1 Requerimientos planteados

A partir de los parámetros de funcionamiento y alcance del proyecto presentado en el capítulo anterior se establecen los requerimientos tanto de hardware como de software del sistema, estableciendo como principales premisas para la selección de componentes, el hecho de que el sistema debe ser modular, planteando la necesidad de la utilización de SHIELDS que cumplan diferentes funciones.

En cuanto al lenguaje de programación, se plantea el uso de la plataforma propia de Arduino, empleando sus librerías e instalando en el caso de ser necesario las que se requieran según los requerimientos de los SHIELDS seleccionados, el código será lo más óptimo posible y cubrirá las necesidades de funcionamiento planteado donde se destacan los modos de funcionamiento, la comunicación con el módulo GSM para la lectura de mensajes, el manejo del reloj en tiempo real para el registro y control de la hora del día, la activación y desactivación de los relés que controlan las lámparas e iluminación y el control del sistema receptor FM que maneja el sistema de audio incorporado. Todos estos aspectos serán descritos, diseñados e implementados en los siguientes ítems.

Considerando todo lo planteado para la parte de software y hardware se procede a analizar y seleccionar las diferentes opciones para la selección de los componentes tanto de hardware y software que serán utilizadas en la implementación del mismo, tomándose como aspectos fundamentales en el orden de selección y la compatibilidad con la tarjeta Arduino, la disponibilidad del módulo en el mercado, información disponible para su conexión, requerimientos de alimentación, utilización de librerías, recursos de software disponibles y su precio.

3.2 Diseño Electrónico

Para comprender la función que cada componente se realiza previamente al diseño del sistema una descripción de las características de cada componente y se justifica su selección, para lo cual se procede a esquematizar el sistema a partir de un diagrama de

bloques, estableciendo sus requerimientos de conexión hacia la tarjeta Arduino como cerebro del sistema estableciendo los pines, conexiones requeridas para su comunicación y adicionalmente se consideran sus requisitos de alimentación que en este caso para todos los módulos son de 5VDC aunque en corriente el módulo que más corriente requiere por obvias razones es el de relés.

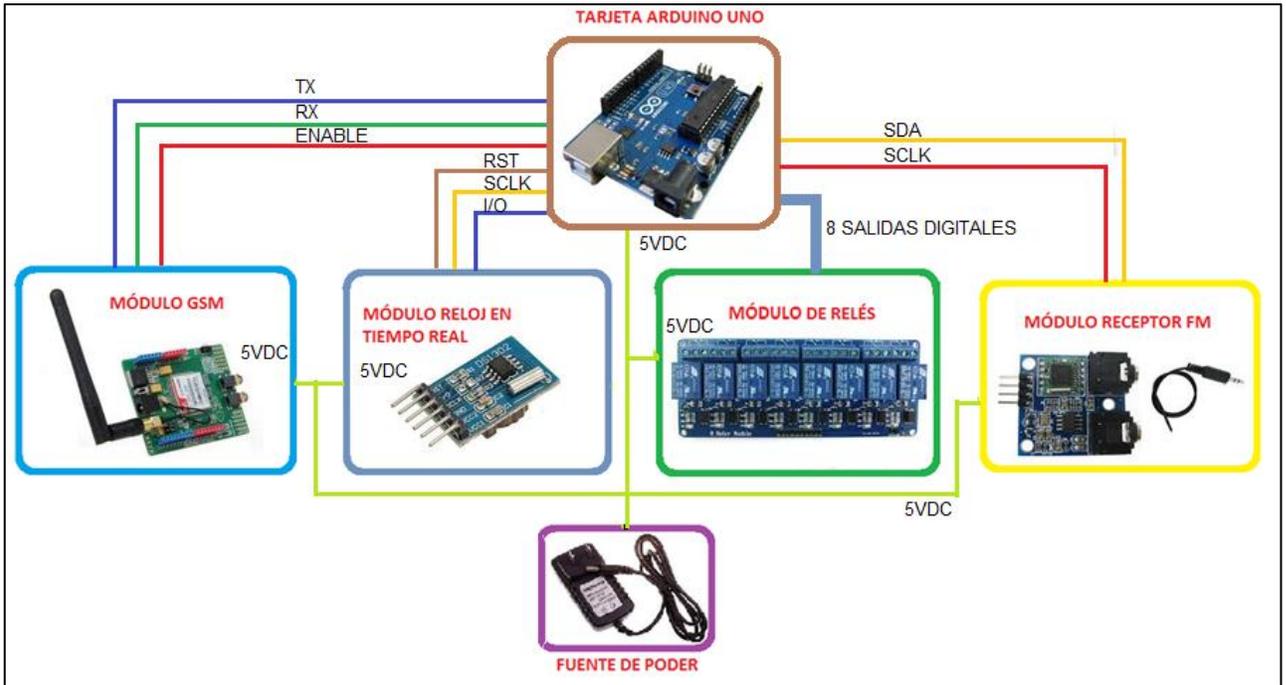


Figura 12. Diagrama de bloques del sistema
Fuente: El autor

3.2.1 Diseño de hardware

Circuito de Potencia

Considerando que la iluminación residencial funciona a partir de una alimentación de 120VAC se plantea la selección del mecanismo de control, donde desde el punto de vista eléctrico – electrónico se tienen dos opciones, la primera es la manejo de TRIAC y la segunda la utilización de Relés.

Los TRIACs ofrecen altas velocidades de conmutación aunque las potencias que manejan pueden ser de bajas a medias, posibilidad de dimerización y control regulable, no planteados en esta aplicación, su costo varía dependiendo de la potencia que se desee manejar, no

requiere mantenimiento y su vida útil depende de los factores ambientales, limpieza y ventilación del sitio de instalación.

Los relés son dispositivos electromecánicos compuestos por una bobina que a través de un campo electromagnético activa o desactiva sus contactos en los cuales son conectados los elementos a controlar, la corriente y voltaje del elemento controlado puede ser alterna o continua y depende de las características del relé, provee un aislamiento eléctrico entre la parte de control y la de potencia, su costo es bajo, existe en una gran variedad de voltajes de disparo, y voltajes y corrientes de trabajo, debido a no existir en el proyecto proyecciones de control proporcional, ni altas velocidades de conmutación se optó por esta opción para la implementación utilizando relés de 5VDC de disparo y con un circuito de disparo a partir de un transistor NPN, como el que se muestra en la figura.



Figura 13. Composición de un Relé y circuito de disparo.
Fuente: El autor

Posteriormente a su comprobación a partir de su implementación en un protoboard, se analizó como se lo conectará en un circuito de iluminación dentro de una instalación residencial, viendo como la opción más adecuada la de colocar al interruptor manual del circuito de iluminación en paralelo con el contacto del relé como se muestra en la figura.

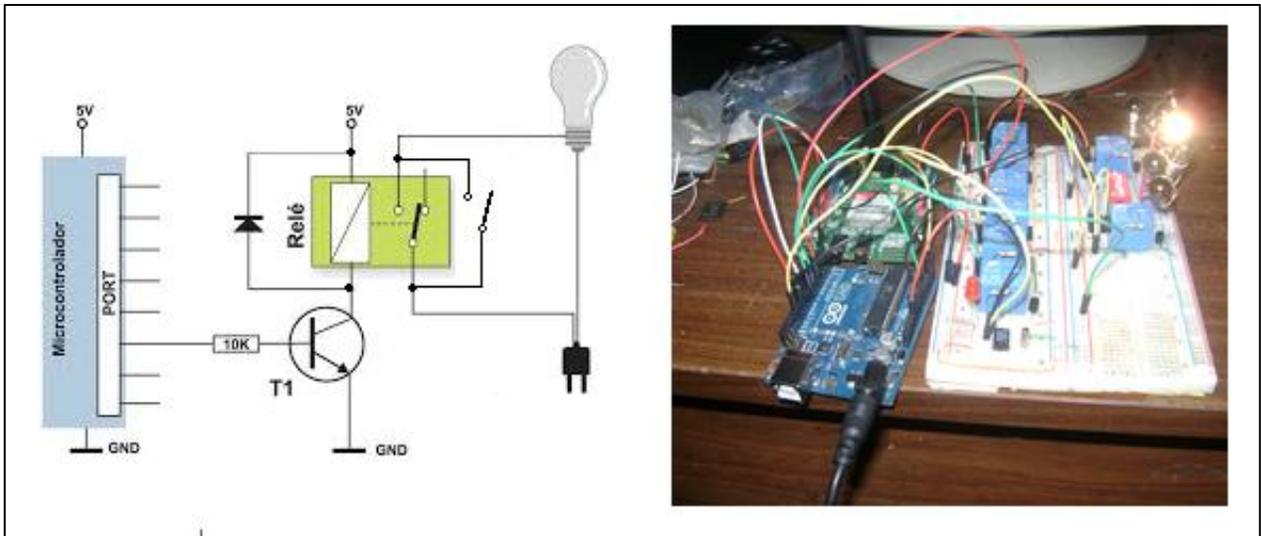


Figura 14. Conexión del Sistema de Relés a un circuito de iluminación manual residencial
Fuente: El autor

Finalmente, al seleccionar un SHIELD de relés se consideró que tenga el número de canales necesarios para su funcionamiento, en este caso un módulo de 8 canales, el voltaje de disparo del relé sea de 5VDC para no requerir fuentes adicionales, voltaje y corriente de trabajo de 120VCA y de hasta 10Amp. Respectivamente, y que provea seguridades adicionales como optoacopladores para dar una mayor aislamiento entre la parte de potencia y control, seleccionando por lo tanto un módulo de relés de 8 canales con optoacopladores y LEDs indicadores de activación, cuyo circuito se muestra en la figura.

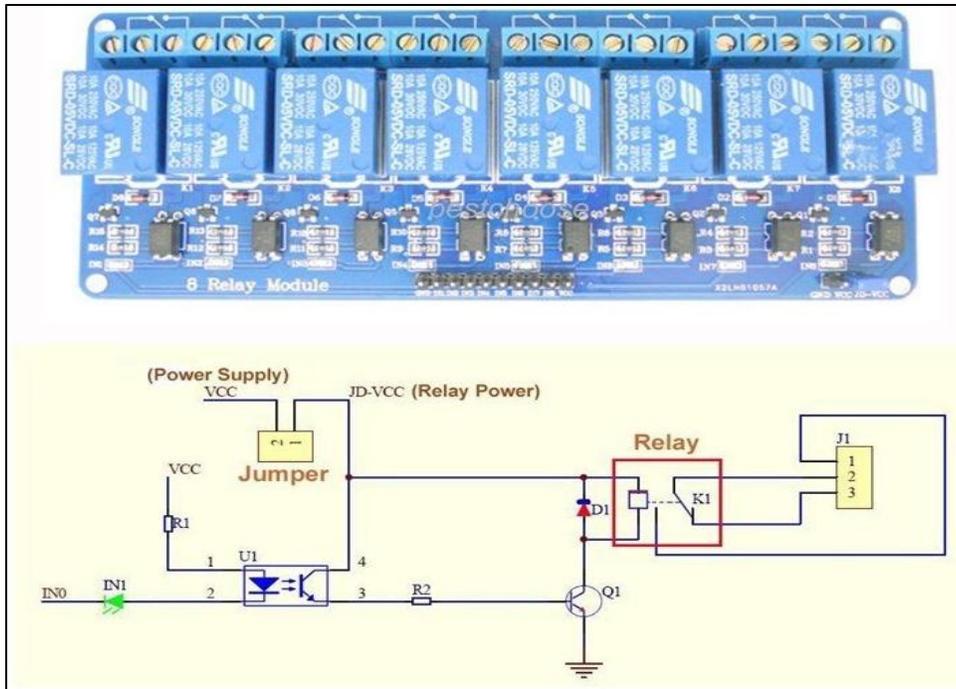


Figura 15. Circuito de opto acoplamiento del relé y módulo de relés de 8 canales de 5VDC
Fuente: ARDUINO/ Forum (<http://forum.arduino.cc/index.php?topic=300482.0>)

Su implementación se puede visualizar en la figura, que muestra el esquema de conexión tanto de la parte eléctrica como a la tarjeta Arduino y una vista del circuito implementado inicialmente para las pruebas de funcionamiento previo al montaje en la maqueta realizada.

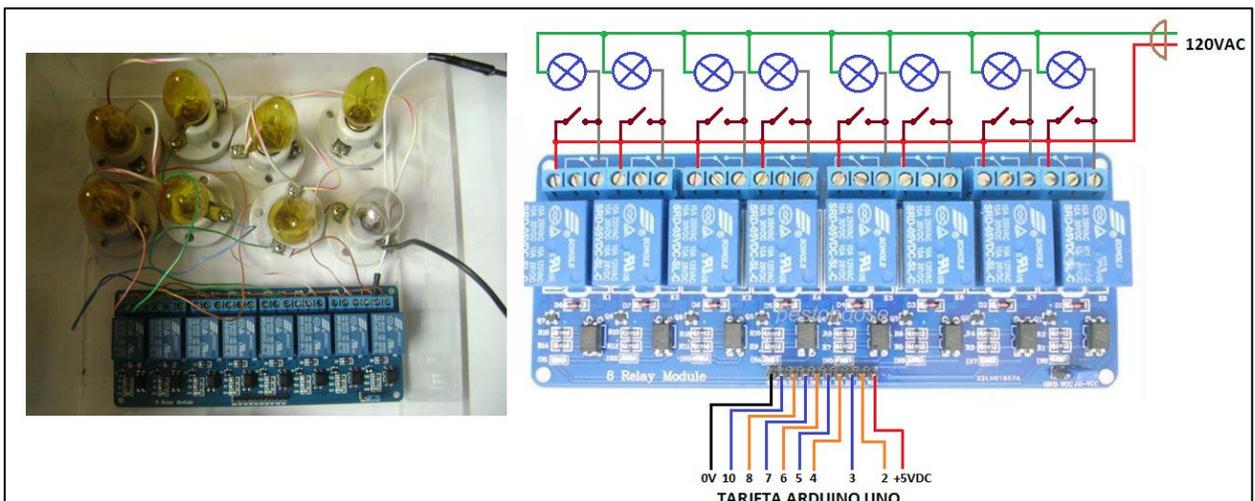


Figura 16. Conexión de la parte eléctrica y a la tarjeta Arduino uno del módulo de Relés
Fuente: El autor

Módulo GSM

Los GSM disponibles en el mercado local, están basados en el circuito integrado SIM900, que es un integrado ampliamente usado dentro de los sistemas GSM, en este caso se va a utilizar el módulo SHIELD SIMCOM SIM900 GSM, el cual tiene un hardware y funcionamiento muy similar a otros SHIELDS basados en el SIM900, por lo que se cuenta con una gran cantidad de información sobre sus componentes de hardware y su programación.

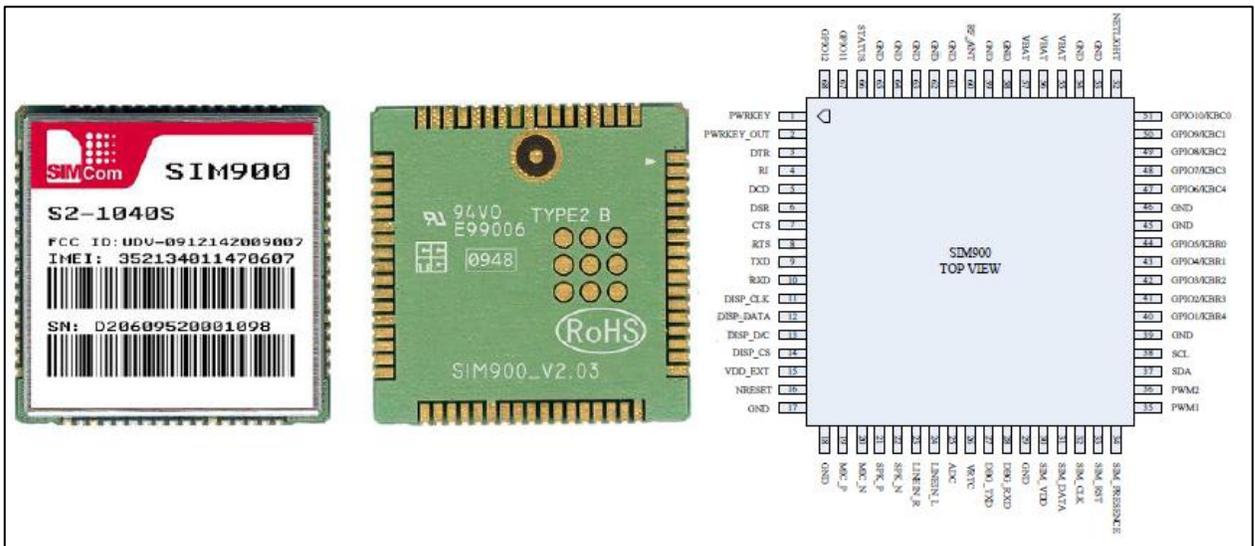


Figura 17. Módulo SIM900 Vista frontal, posterior y descripción de pines.

Fuente; Hardware Desing SIM900_HD_V1.01

El módulo GSM cuenta con las siguientes características generales:

- Requiere un Chip GSM de algún proveedor de telefonía móvil.
- Cobertura de red en las frecuencias GSM 850 MHz, GSM 900 MHz, DCS 1800 MHz o 1900 MHz PCS red, esto significa GSM 2G - no 3G, 4G o LTE.
- Una entrada de alimentación externa de alta corriente, requiere una alimentación interna a través de pines de la Tarjeta Arduino o externa de 5VDC seleccionable por medio de un interruptor.
- Antena incluida
- Posibilidad de comunicación con tarjeta Arduino a través de pines D7 y D8, denominado Hardware serial o pines D0 y D1, denominado software serial, para su control, configurable a través de un par de jumpers.

- Entradas y salidas de Audio.
- Indicadores luminosos de estado y red.
- Posibilidad de activar o desactivar el módulo a través del pin D9 de la tarjeta Arduino previa configuración del jumper correspondiente.



Figura 18. SIMCOM SIM900 GSM. Vista frontal, posterior e identificación de partes.
Fuente: El autor

Para su utilización en el proyecto, se aplicará en la comunicación entre el módulo GSM y la tarjeta Arduino UNO, la configuración de los pines D0 y D1, debido a que estos no pueden ser usados como salidas digitales y son destinados a la comunicación serial, dejando liberados los pines D7 y D8 para utilizarlos como salidas digitales, configurando mediante un par de jumpers como se ilustra en la figura.

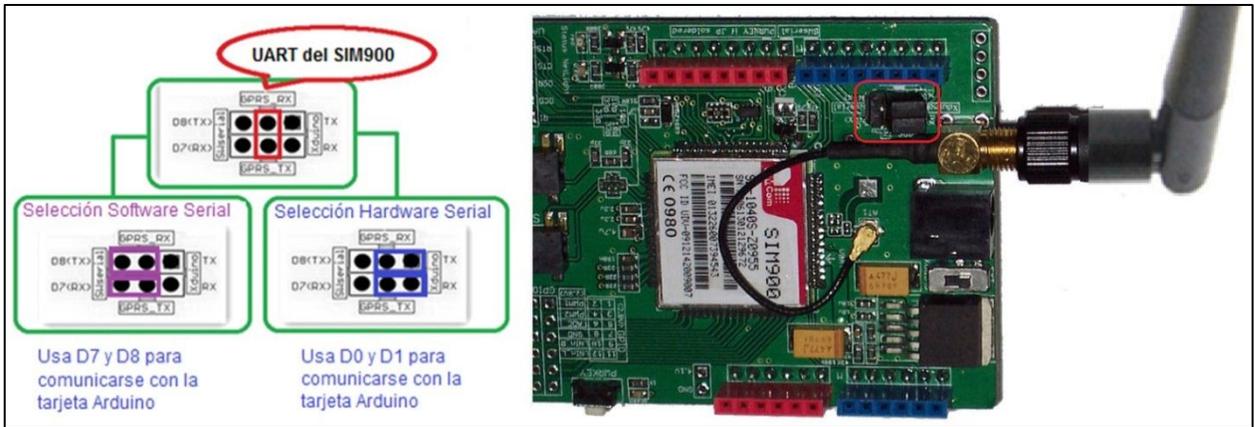


Figura 19. Configuración de jumper para comunicación serial a través de los pines D0 y D1.
Fuente: El autor

Con la finalidad de controlar la activación y desactivación del módulo GSM a partir de la tarjeta Arduino, se selecciona el pin D9 para manipular la activación del módulo, para lo cual se debe configurar el jumper correspondiente, uniendo a través de una suelda con caufín que una los terminales correspondientes.

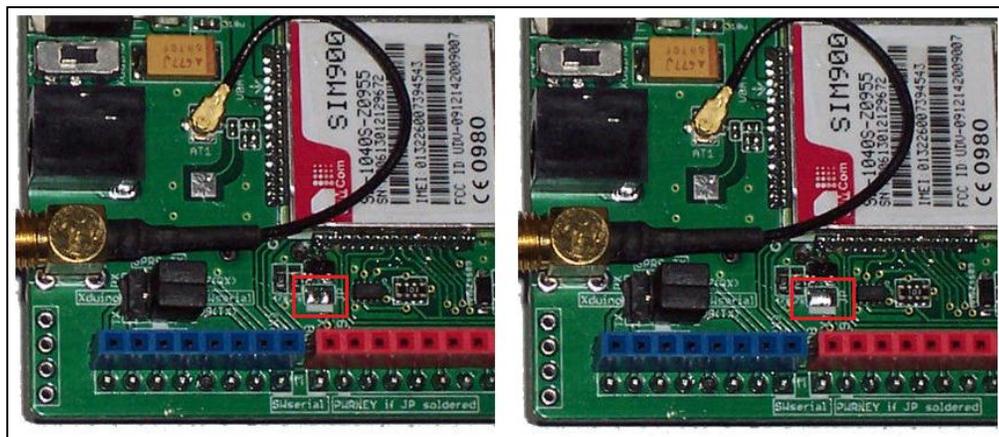


Figura 20. Puente de suelda que activa el jumper para control de activación del Módulo GSM
Fuente: El autor

Finalmente, se muestra los pines a ser conectados entre el módulo GSM y la tarjeta Arduino Uno, el zócalo donde se debe insertar el chip GSM ubicado en la parte posterior del módulo y el conector donde debe verificarse la conexión de la antena.

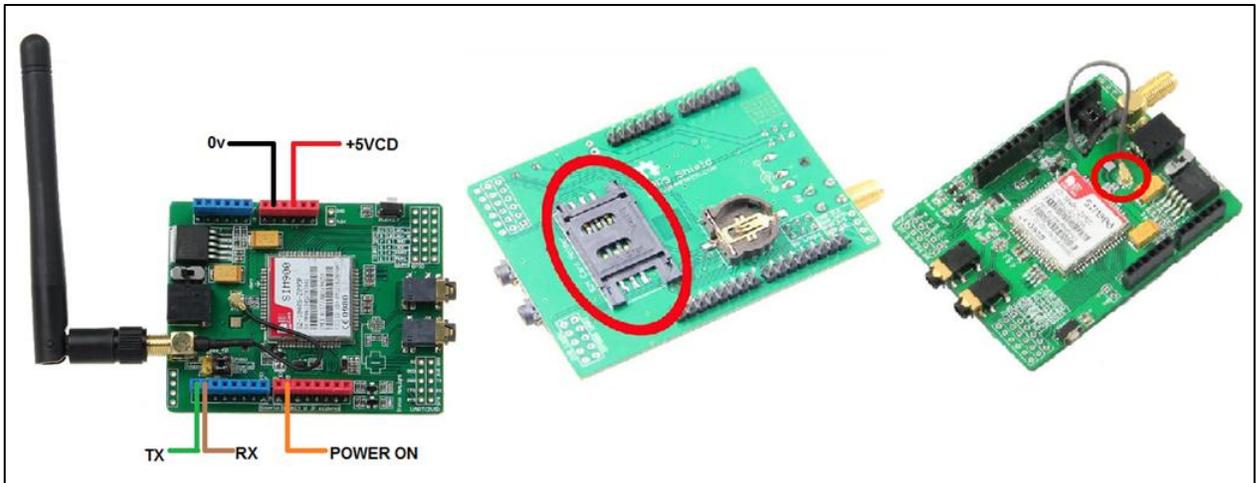


Figura 21. Conexión del módulo GSM a la tarjeta Arduino, zócalo de inserción de chip GSM y plug antena
Fuente: El autor

Para su selección se consideró el hecho de cumplir con los parámetros de funcionamiento requeridos, el poseer una antena externa que le da más potencia, su disponibilidad en el mercado, ya que es el mayor demanda, disponibilidad actualmente y su costo, que es relativamente bajo, respecto a otras opciones.

Módulo Reloj en Tiempo Real

Para la administración y registro de la hora, día, fecha, asistida para el modo automático, propuesto anteriormente, se seleccionó un módulo reloj en tiempo real, basado en un integrado DS1302, el cual tiene la distribución de pines y circuito típico de utilización mostrado en la siguiente figura, además de presentar características de funcionamiento que lo hacen ideal para este tipo de aplicaciones, como el hecho de contar con una pila que permite mantener el conteo de tiempo, e inclusive sin el suministro principal de energía, teniendo una exactitud en la medición y registro del tiempo bastante alta, al ser fabricado para este propósito, con una comunicación serial a través de tres pines con el módulo Arduino.

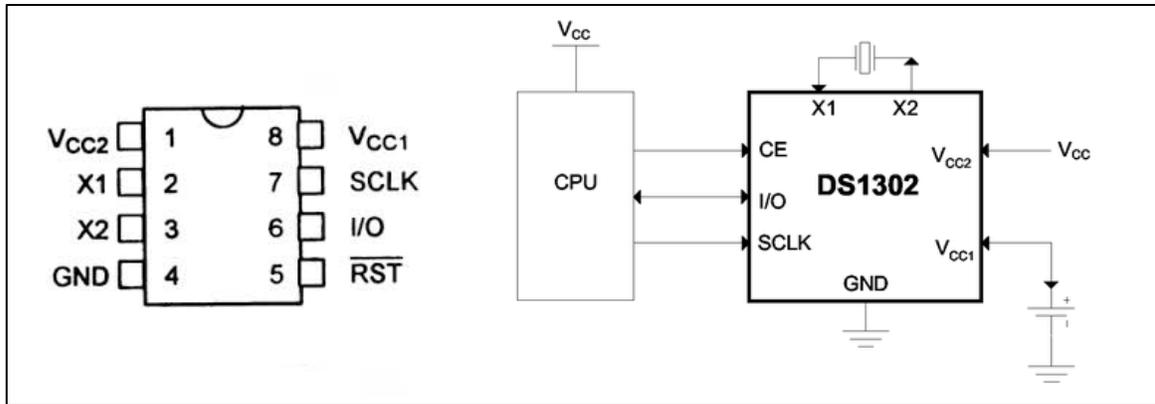


Figura 22. Distribución de pines y circuito típico de utilización del chip DS1302
Fuente: Datasheet DS1302 Trickle-Charge Timekeeping Chip, Maxim Integrated

En el mercado local existen varios módulos RTC basados en el chip DS1302 y en el DS1307, de los cuales se seleccionó el mostrado en la figura debido a su tamaño compacto, facilidad de conexión y bajo costo.

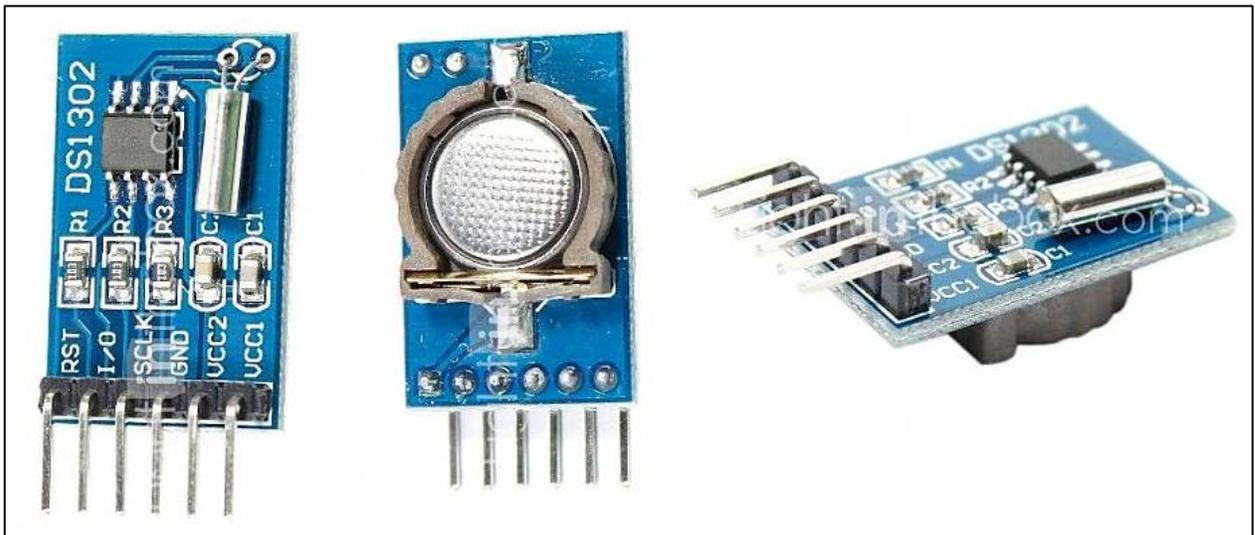


Figura 23. Módulo DS1302
Fuente: El autor

Para la conexión, se aplicara de acuerdo a las recomendaciones del fabricante, siendo los pines de alimentación de 5VDC del módulo Arduino y para la comunicación los pines D11, D12 y D13 del Arduino, conectados a los pines RST, I/O y SCLK del módulo RTC respectivamente, como se ilustra en la figura.

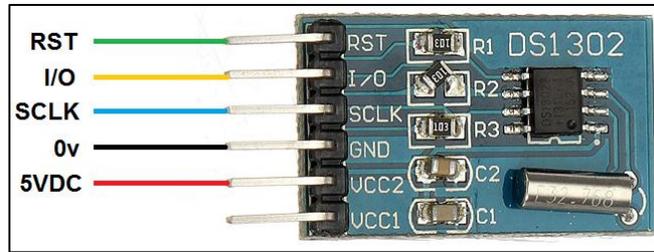


Figura 24. Conexión del Módulo RTC a la tarjeta Arduino.
Fuente: El autor

Receptor FM

El módulo receptor FM está diseñado en base al circuito integrado receptor FM de Philips TEA5767, ampliamente manejados en este tipo de aplicaciones, la comunicación con cualquier microcontrolador se realiza a través de comunicación Inter Integrated Comunication (I2C), requiriendo únicamente la alimentación de 5VDC y la conexión de comunicación para funcionar. Adicionalmente tiene las características descritas en la tabla.

Tabla 2. Características del Módulo 5v 76-108mhz Tea5767 Fm Stereo Radio para Arduino.
Fuente: El autor

CARACTERÍSTICA	DESCRIPCIÓN
Fuente de alimentación	5 V
Circuito tamaño del módulo	31mm x 30mm (largo x ancho)
Rango de frecuencia	76-108 MHz
Protección	De alimentación inversa por medio de diodo Con inductancia de filtro de salida
Indicador	Filtro combinado capacitancia múltiple LED azul de alimentación
Conector audio	3.5 mm estéreo
Conector antena	3.5 mm
Comunicación	bus I2C
Amplificación de audio	Philips TDA1308
Receptor FM	Philips TEA5767

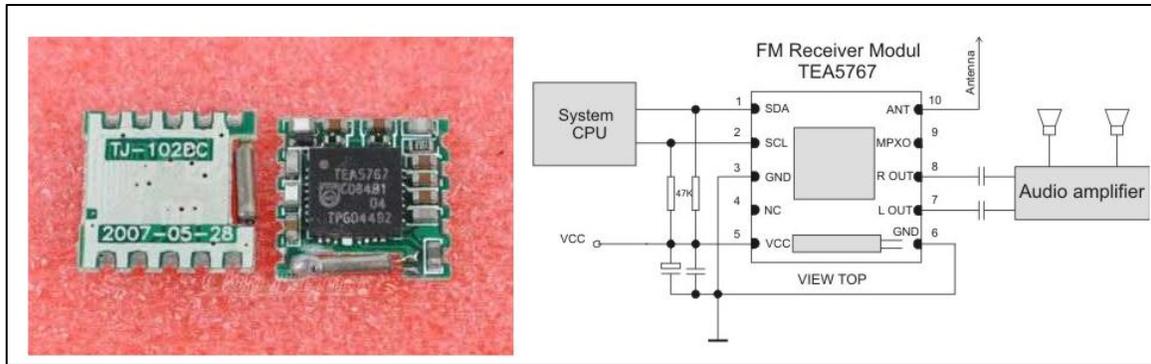


Figura 25. Circuito integrado receptor FM Philips TEA5767 y circuito de utilización típico
 FUENTE: Hoja técnica TEA5767HN Low-power FM stereo radio for handheld applications

El módulo es conectado con la tarjeta Arduino Uno a través de dos pines, I2C, por los terminales análogos A5 y A4, que corresponden a las señales I2C SCL y SDA respectivamente, adicionalmente de la conexión a la fuente de alimentación de 5VDC, tomadas de los pines de alimentación de la misma tarjeta Arduino, el módulo utilizado incluye un pequeño amplificador que lo hace útil en el caso de utilizar audífonos, en este caso para alcanzar un volumen lo suficientemente alto fue necesario el incluir un sistema amplificador tomado de parlantes de computadora.

Se seleccionó este módulo debido a su fácil programación, funcionalidad, tamaño reducido, se ajusta a los requerimientos planteados, es de precio accesible y está disponible en el mercado.

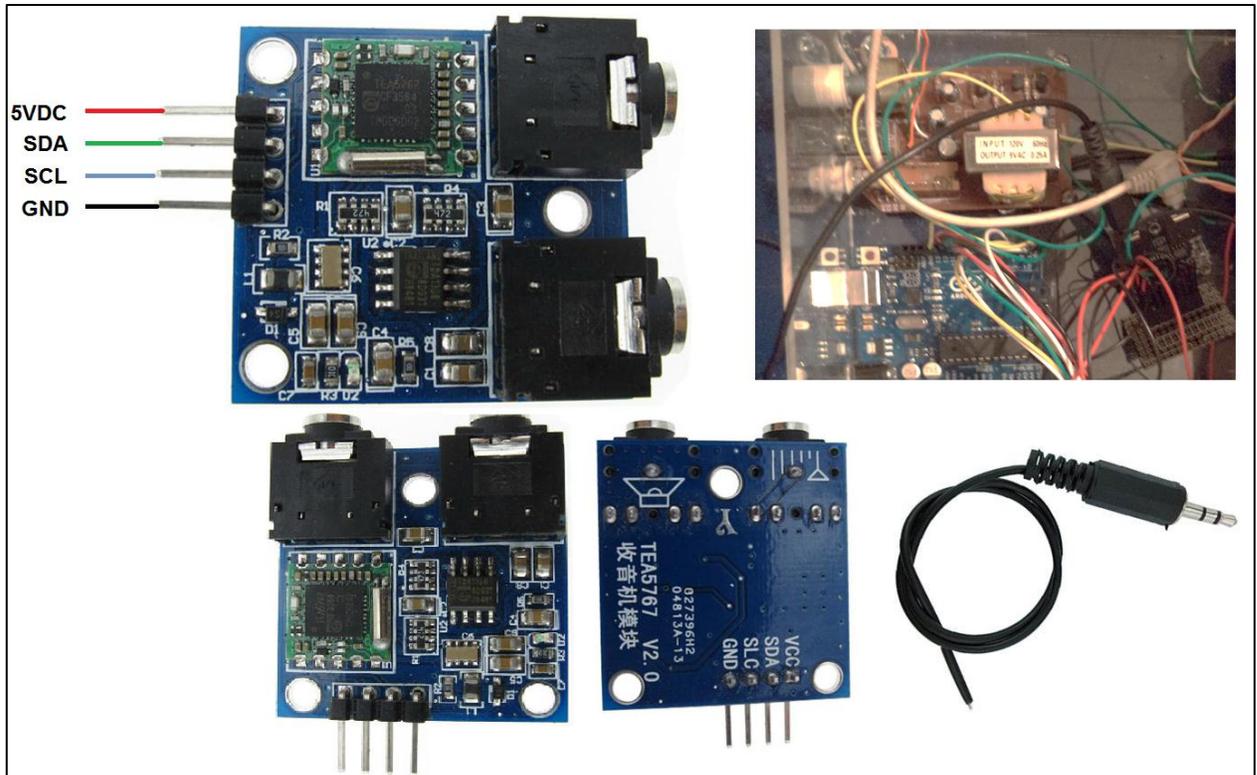


Figura 26. Módulo Receptor FM, conexión al módulo Arduino Uno, vista frontal y posterior.
Fuente: El autor

Tarjeta Arduino Uno

La tarjeta Arduino Uno empleada como cerebro del trabajo realizado basado en el microcontrolador ATmega328P, cuyas características ya se describieron, es seleccionada por aspectos como, que cuenta con el número de terminales de salida requeridos, los puertos de comunicación como serial e I2C que como se planteó en la descripción de los módulos, son requeridos para su control y funcionamiento, tiene un bajo consumo de corriente, es de fácil programación al usar el software de desarrollo propio de Arduino.

Fuente de Alimentación

Para alimentar al sistema se proyecta el manejo de una fuente regulada conmutada de 5VDC de 1.5 Amp., con lo cual se suministra a todo el sistema, pese a que en el análisis respectivo se planteó un consumo inferior al 1 Amp., siendo el de mayor consumo el módulo de relés, el cual consume 800mA aproximadamente, pero debido al consumo pico de disparo de los relés se optó por incrementar la corriente de la fuente seleccionada para dar un mayor margen de seguridad dentro de la alimentación del sistema.

Conexión del sistema

Determinando de que todos los componentes del sistema son modulares y dado el hecho de que este sistema está concebido con fines didácticos, se proyecta la conexión a través de cables flexibles para facilitar su modificación y poder comprobar diferentes tipos de conexión, por lo que se hicieron pequeñas tarjetas de conexión a partir de placas impresas universales para la conexión de los módulos de relés y RTC en una sola placa y otra para el módulo receptor FM, adicionalmente en las pruebas iniciales se aplicó el uso de una placa con un dip-switch para el encendido manual de las luces.

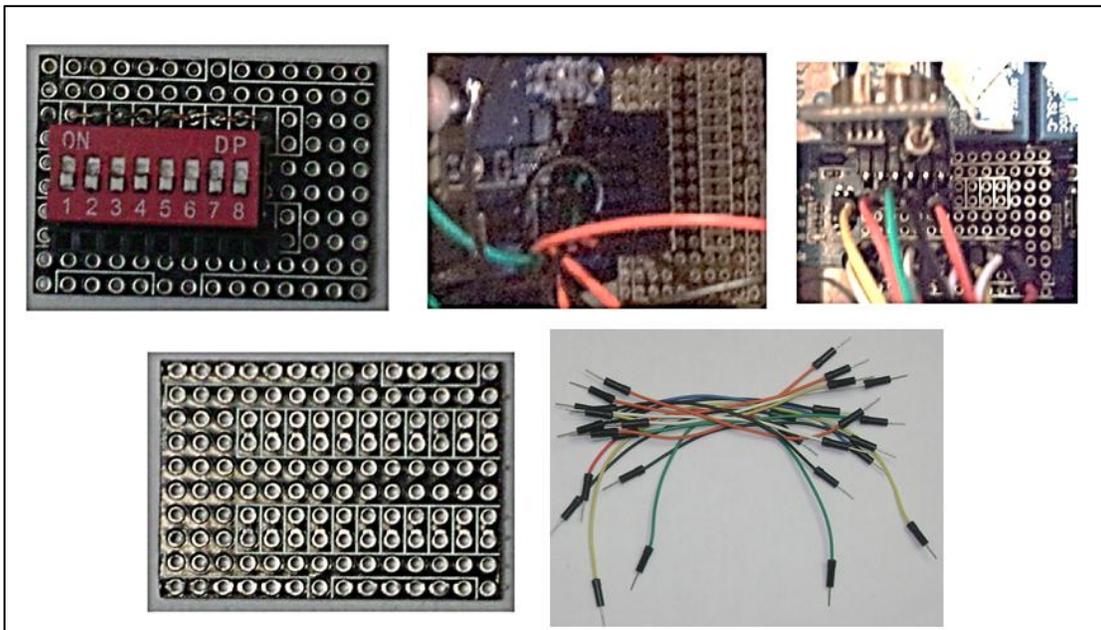


Figura 27. Placas de conexión elaboradas y cables.

Fuente: El autor

Finalmente, se muestra el esquema de conexión de todo el sistema planteado en la figura en la cual se evidencia la conexión en la tarjeta Arduino Uno en los diferentes módulos empleados, a partir del cual se desarrolló el software que establece el cumplimiento de los parámetros de funcionamiento planteados.

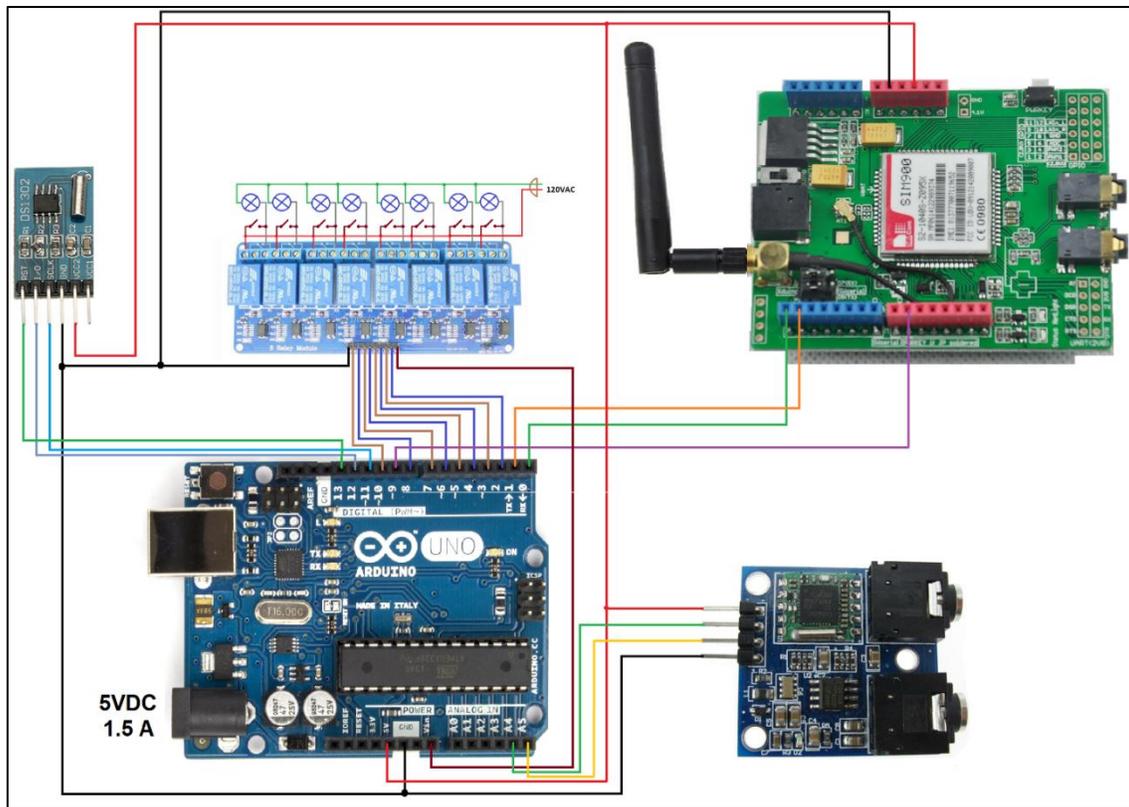


Figura 28. Esquema de conexión del proyecto
Fuente: El autor

3.2.2 Diseño de software

Para el desarrollo del algoritmo de funcionamiento del sistema se planteó el uso del software propio de Arduino que permite el desarrollo, compilación y transferencia del código desarrollado en su versión 1.0.6.

En base a los parámetros de funcionamiento propuestos en el capítulo anterior, se establece el diagrama de flujo mostrado en la figura, en el cual se describe su implementación, logrando verificar la forma de ejecutar cada modo, en base a la validación del número celular como primer paso, seguido de la validación del comando y la ejecución del mismo, dependiendo del dispositivo o localización especificada en el comando, para el caso de modo remoto y para el modo automático se establece en base a la validación de la hora configurada para el dispositivo, la acción inmediata de encendido o apagado para el área de la vivienda asignadas según programación.

Para almacenar los datos de los números celulares y las horas configuradas, se establece el almacenamiento de estos datos en la memoria EEPROM, memoria no volátil del microcontrolador, a partir de un mapa de memoria, donde se desprende que existen ocupados 500 de los 1024 bytes disponibles para ampliaciones.

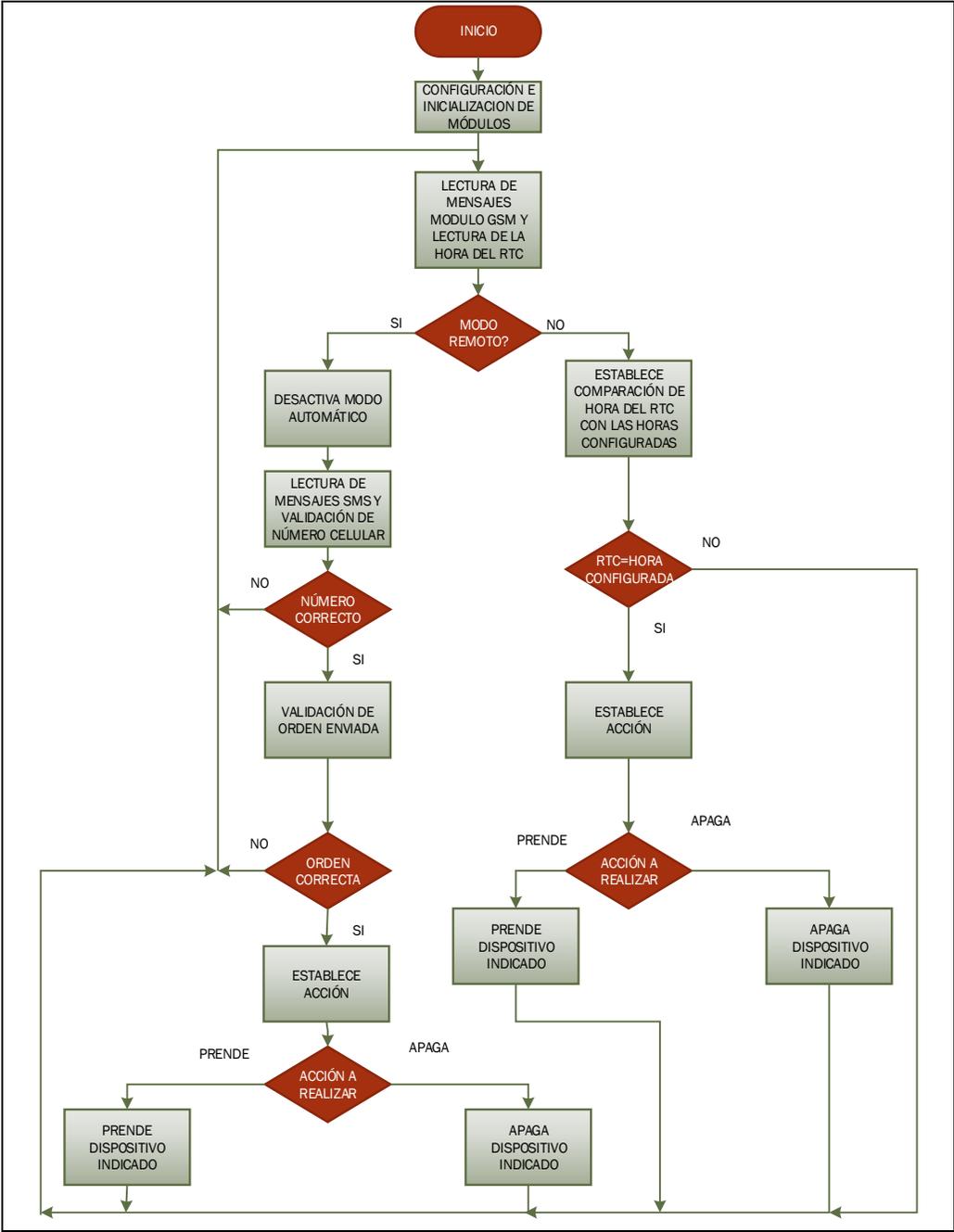


Figura 29. Diagrama de flujo propuesto
Fuente: El autor

Mapa de memoria EEPROM

Tabla 3. Mapa de utilización de memoria EEPROM del Microcontrolador ATmega328P

Fuente: El autor

DIRECCIÓN	UTILIZACIÓN	COMENTARIO
0000	Números Celulares	Se establece la posibilidad de almacenar 10 números celulares de 9 dígitos cada uno, sin contar el 0 inicial, por ejemplo 997288957
.	Registrados	
.		
0099		Se asignan 10 espacios para almacenar la hora de encendido de los aparatos, pudiendo configurar hasta 10 dispositivos
0100	Hora de encendido de	
.	dispositivos	
.		Se asignan 10 espacios para almacenar la hora de apagado de los aparatos, pudiendo configurar hasta 10 dispositivos
0199	Hora de apagado de	
0200	dispositivos	
.		Se asignan 10 espacios para almacenar el minuto de encendido de los aparatos, pudiendo configurar hasta 10 dispositivos
0299	Minuto de encendido de	
0300	dispositivos	
.		Se asignan 10 espacios para almacenar el minuto de apagado de los aparatos, pudiendo configurar hasta 10 dispositivos
0399	Minuto de apagado de	
0400	dispositivos	
.		Se utiliza un byte que almacena el modo configurado por SMS aun en caso de corte de energía
0499	Modo Remoto o Automático	
500		

A cuanto al registro de los números celulares, se establece el almacenamiento de los números celulares se plantea el almacenamiento de 10 números de 9 dígitos, los dígitos deben almacenado en formato ASCII, a través del comando:

```
EEPROM.write(30, '9');
```

Donde 30 es la dirección y el 9 es el digito a guardar, el cual para almacenarlo en formato ASCII debe ser puesto entre comillas simples, almacenando en esta versión solo 4 números celulares, pero, quedando libre el espacio para 6 números adicionales, el almacenamiento de

los mismos se muestra en la tabla, evidenciando que entre números existe una dirección libre.

Tabla 4. Almacenamiento de números celulares en la memoria EEPROM
Fuente: El autor

No CELULAR	DIRECCIÓN	Digito	No CELULAR	DIRECCIÓN	Digito
0997288957	0000	9	0984979025	0020	9
	0001	9		0021	8
	0002	7		0022	4
	0003	2		0023	9
	0004	8		0024	7
	0005	8		0025	9
	0006	9		0026	0
	0007	5		0027	2
	0008	7		0028	5
	0009	LIBRE		0029	LIBRE
0992912776	0010	9	0981806875	0030	9
	0011	9		0031	8
	0012	2		0032	1
	0013	9		0033	8
	0014	1		0034	0
	0015	2		0035	6
	0016	7		0036	8
	0017	7		0037	7
	0018	6		0038	5
	0019	LIBRE		0039	LIBRE

Modo Automático

Para la comunicación con el RTC se utiliza una librería libre `virtuabotixRTC.h` descargada de la página <https://www.virtuabotix.com/virtuabotix-ds1302-real-time-clock-module-pin-out-coding-guide/> la cual permite una fácil comunicación con el módulo RTC, basada en los comandos que se describen en la tabla.

Tabla 5. Comandos para la comunicación con el módulo RTC utilizados de la librería `virtuabotixRTC.h`
Fuente: El autor

COMANDO	FUNCIÓN	EJEMPLO
<code>virtuabotixRTC myRTC(x, y, z);</code>	Configura los pines con los cuales se va a realizar la comunicación, siendo: X: SCLK Y: I/O Z: CE	<code>virtuabotixRTC myRTC(11, 12, 13);</code> Que configura los pines tal como se describe en el esquema del sistema.
<code>myRTC.setDS1302Time(seg, min, hor, dias, día, mes, año);</code>	Setea la hora del sistema, este comando debe ser usado solo cuando se requiera igualar la hora del sistema, ya que la pila el modulo la mantiene pese a cortes de energía.	<code>myRTC.setDS1302Time(30, 8, 13, 6, 3, 2, 2015);</code>
<code>myRTC.updateTime();</code>	Actualiza la hora, leyendo la misma del módulo RTC.	<code>myRTC.updateTime();</code>

Los datos de RTC se devuelven en las siguientes variables:

`myRTC.seconds`

`myRTC.minutes`

`myRTC.hours`

Siendo estos datos los utilizados para realizar comparaciones con los datos de horas de encendido y apagado configurados. Para el registro de la hora se plantea una maqueta en la cual se identifican 8 ambientes para el control de iluminación y un sistema de audio, a su vez

se identifican las horas y minutos de encendido o pagado en modo automático, los cuales se plantearon en base a un registro de horas de encendido y apagado de los diferentes ambientes que se plasmaron en la tabla que se muestra en continuación y en la tabla siguiente se muestra el ejemplo de almacenado de los datos de la hora y minuto de encendido y apagado dentro de la memoria EEPROM, la cual se replica para el resto de ambientes. Adicionalmente, nótese que si bien es cierto existen asignados 10 espacios, el que más horas tiene es el baño ocupando 6 espacios, lo que significa que quedan espacios libres para futuras ampliaciones y cambios en los horarios, adicionalmente al estar dentro de la memoria EEPROM de datos, los cambios de los mismos no involucran cambios en la programación.

Tabla 6. Horas de encendido y apagado de ambientes en modo automático
Fuente: El autor

No	AMBIENTE	HORAS ENCENDIDO	HORAS APAGADO
1	Baño	19:43 21:13 23:26 0:01 3:35 5:30	19:48 21:23 23:30 0:03 3:38 5:45
2	Cocina	19:23 21:34 3:39 5:40	20:31 21:44 3:43 6:00
3	Sala	18:00 4:36	22:00 6:00
4	Dormitorio Master	20:18 5:00	23:00 5:45
5	Dormitorio 1	18:25 21:30	20:09 22:54
6	Dormitorio 2	18:13 20:57	19:41 23:34
7	Entrada	19:00 5:30	22:12 6:24
8	Parte Externa	17:00 5:49 13:00	19:00 7:30 16:09
9	Audio	18:00 4:36 12:00	21:00 8:00 15:24

Tabla 7. Asignación de horarios de encendido apagado en la memoria EEPROM

Fuente: El autor

No	AMBIENTE	DIR HORA ENC	DATO	DIR MIN ENC	DATO	DIR H APAG	DATO	DIR M APAG	DATO
1	Baño	0100	19	0300	43	0200	19	0400	48
		0101	21	0301	13	0201	21	0401	23
		0102	23	0302	26	0202	23	0402	30
		0103	0	0303	1	0203	0	0403	3
		0104	3	0304	35	0204	3	0404	38
		0105	5	0305	30	0205	5	0405	45
	
2	Cocina	0110	19	0310	23	0210	20	0410	31
	
3	Sala	0120	18	0320	0	0220	22	0420	0
	
4	Dormitorio Master	0130	20	0330	18	0230	23	0430	0
	
5	Dormitorio 1	0140	18	0340	25	0240	20	0440	9
	
6	Dormitorio 2	0150	18	0350	13	0250	19	0450	41
	
7	Entrada	0160	19	0360	0	0260	22	0460	12
	
8	Parte Externa	0170	17	0370	0	0270	19	0470	0
	
9	Audio	0180	18	0380	0	0280	21	0480	0
	

Cabe mencionar que estos horarios se leen y comparan en pares de hora y minuto de encendido o apagado es decir se lee y compara por ejemplo la dirección 0101 con la hora del RTC y la dirección 0301 con el minuto del RTC y si ambos son iguales se procede a encender la luz del baño en este caso por estar en la primera década de la serie, cosa similar pero con las direcciones 0201 y 0402 respectivamente para el horario de apagado de

la misma, nótese adicionalmente que las horas no están colocadas en orden ascendente ni descendente ya al realizar la comparación se hace un barrido de todo el bloque de horas para cada ambiente y se realiza de forma repetitiva para cada ambiente, hasta completar los 10 existentes, sin importar si esta cargadas horas o no, esto se implementó a través de un ciclo for anidado en otro ciclo for, una vez completado el ciclo de encendido se inicia un ciclo similar pero con direcciones de hora y minuto de apagado para el ciclo de apagado automático, como describe el diagrama de flujo de la figura

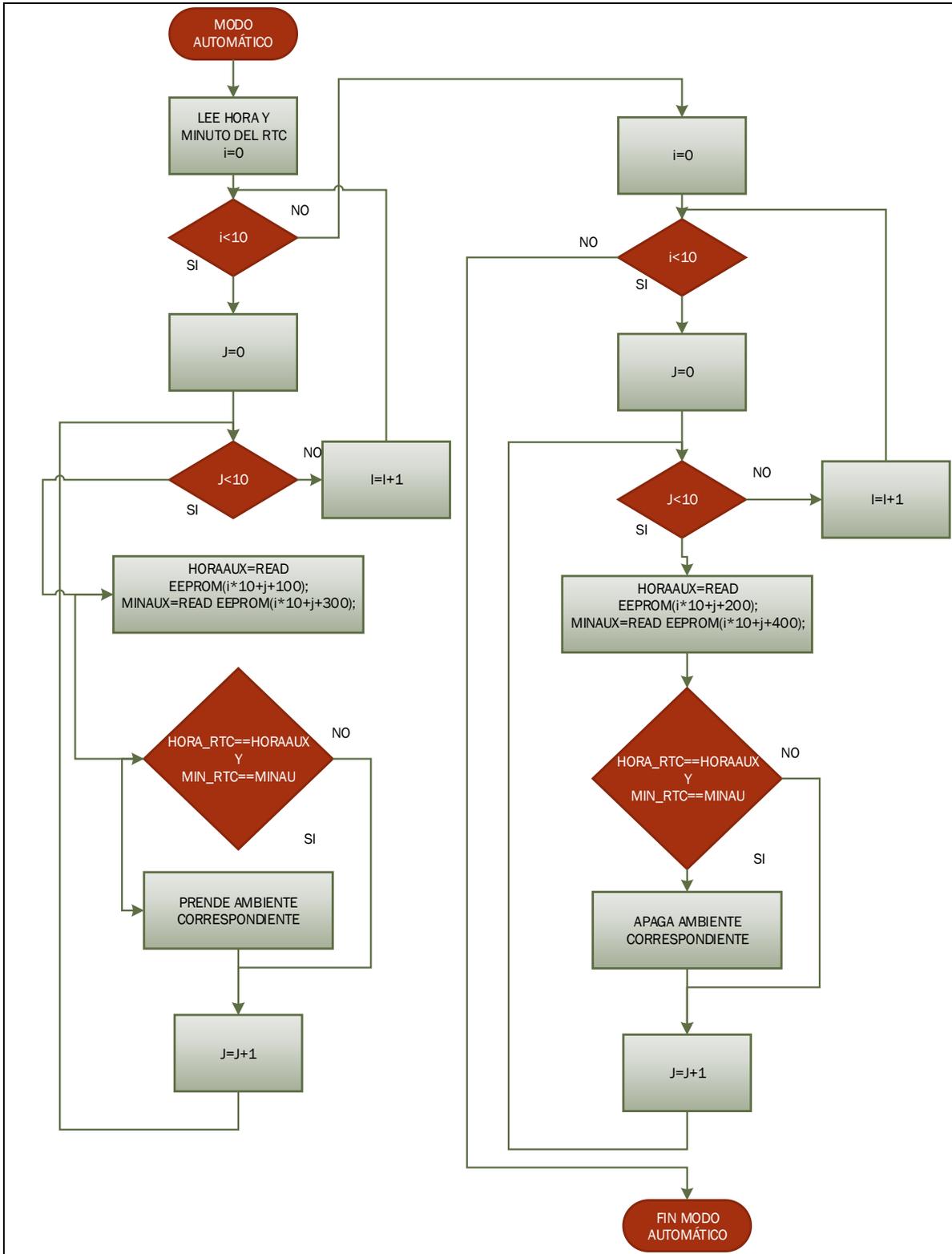


Figura 30. Diagrama de flujo del modo automático
Fuente: El autor

Modo remoto

Para comprender el funcionamiento del modo remoto, primero se debe entender cómo se realiza la comunicación entre la tarjeta Arduino Uno y el módulo GSM, encargado de recibir los mensajes de texto que permiten controlar el funcionamiento en este modo.

La comunicación con el módulo se realiza a través de comunicación serial, con una velocidad de 19200kbps por defecto, aunque es configurable, como ya se explicó en la parte de hardware para la comunicación se estableció a través de jumpers los pines D0 (Rx) y D1 (Tx) en el módulo GSM, y el pin D9 para el encendido del módulo, para esto se empleó la librería `SoftwareSerial.h` incluida en la cabecera del programa.

Comandos AT

El control y comunicación entre la tarjeta Arduino Uno y el módulo GSM se realiza a través de comandos AT, los cuales son denominados así por la abreviatura Attention Command, el cual es un lenguaje de comunicación entre equipos terminales modem de comunicación y computadores, el cual se ha extendido como un estándar de comunicación entre diferentes tipos de equipos de telecomunicaciones, en este caso entre la tarjeta Arduino uno y el módulo GSM.

Los comandos AT utilizan líneas de comandos de texto empleado caracteres alfanuméricos, que indican al módulo GSM las funciones que debe realizar, todos los comandos AT inician con el prefijo "AT" acompañadas del comando específico. La tabla muestra un resumen de los comandos AT básicos.

Tabla 8. Comandos AT Básicos.
Fuente: SIM900_AT Comand Manual_V1.03

COMANDO	FUNCIÓN	EJEMPLO
AT	Comprueba la disponibilidad del dispositivo, es una manera de comprobar que existe comunicación	AT OK
ATA	Para contestar una llamada, también se configura en respuesta automática. Si no hay ninguna llamada entrante responde.	ATA NO CARRIER
ATE	Activación/Desactivación del eco del módem ATE0 desactiva el eco del módem ATE1 activa el eco del módem	ATE0 OK

Para este caso se utilizaron los comandos AT específicos para la lectura de mensajes de texto SMS, los cuales son especificados en la siguiente tabla:

Tabla 9. Descripción comandos AT para recepción de mensajes SMS utilizados.
Fuente: El autor

COMANDO	FUNCIÓN	EJEMPLO
AT+CMGF	Establece el modo en que se van a recibir y leer los mensajes de texto, en este caso se configuro con el valor de 1, que corresponde a codificación de texto en formato ASCII.	AT+CMGF=1
AT+CNMI	<p>Lectura de nuevos mensajes de texto recibidos, en este caso se establece con parámetros de configuración:</p> <p>Modo 2, establece que los datos cuando son recibidos reemplacen a los previamente existentes.</p> <p>Almacenamiento 2, configura el sistema para que los mensajes una vez recibidos no sean almacenados en la memoria de chip sino que una vez transmitidos sean borrados.</p> <p>Estatus 0, esto permite que no se reporte el estado de los mensajes recibidos.</p> <p>Buffer 0, establece que en el caso de no existir mensajes recibidos el sistema indique esto a través de un OK</p>	<p>AT+CNMI=2,2,0,0,0</p> <p>En el caso de existir mensajes recibidos:</p> <pre>+CMT: "+5939972889 57", "", "14/12/11, 00:49:08-20"</pre> <p>hola</p> <p>En caso de no existir mensajes:</p> <p>OK</p>

Para la inicialización del módulo y la comunicación se utiliza el comando `Serial.begin(19200);` que inicializa y configura la velocidad del puerto serial, lo cual se esquematiza y se ilustra a partir del siguiente ejemplo.

Tabla 10. Ejemplo de comandos SMS
Fuente: El autor

MENSAJE	MODO	ACCIONES
+CMT:"+593997288957", "", "14/12 /11,00:49:08-20" Aut0 Ba 1 Sa 1 D1 1	Remoto	Cambia a modo remoto, enciende las luces del baño, sala y dormitorio1.
+CMT:"+593997288957", "", "14/12 /11,00:49:08-20" Ba 0 Mu 1 D1 0	Remoto	Si está en modo remoto, apaga la luz del baño y dormitorio 1 y enciende la radio, caso contrario no hace nada.
+CMT:"+593997288957", "", "14/12 /11,00:49:08-20" Aut1	Automático	Cambia a modo Automático y apaga todas las luces, el sistema las encenderá según el horario configurado.

En la configuración del módulo GSM se envía:

```
Serial.print("ATE0\r");
Serial.println();
Serial.print("AT+CMGF=1\r");
Serial.println();
```

Adicionalmente, basado en la información obtenida en los foros oficiales de Arduino y en información de la ayuda del mismo, se establece que para el encendido del módulo se a través de la salida digital del módulo Arduino se requiere un pulso en alto de al menos 2 segundos, por lo que se utiliza una subrutina para este propósito.

```
void SIM900power(){
digitalWrite(9, HIGH);
delay(4000);
digitalWrite(9, LOW);
delay(1000);
}
```

Para el control de la iluminación a través de los mensajes de esto se establecen códigos que identifican a cada ambiente a ser controlado, considerando que siempre la primera letra debe ser mayúscula y asignando a cada código las 2 primeras letras de cada ambiente un espacio en blanco y el número 1 que identifica al encendido o el número 0 que identifica al apagado,

adicionalmente se establece para el modo el comando Aut1 que coloca el sistema en modo automático y Aut0 que configura el sistema en modo remoto.

Finalmente cabe mencionar que si el sistema está en modo automático los SMS son ignorados a no ser que contengan el comando Aut0 y a su vez las horas de encendido son ignoradas en modo remoto a no ser que se reciba un comando Aut1 y existe la posibilidad de prender o pagar varios aparatos en un mismo mensaje de texto, todo esto está ilustrado en la tabla.

Tabla 11. Comandos establecidos para el control a través de mensajes de texto en modo remoto
Fuente: El autor

AMBIENTE	CÓDIGO HABITACIÓN	DATO DE ENCENDIDO APAGADO	EJEMPLO
Baño	Ba	1	Ba 1 Enciende la luz del baño
		0	Ba 0 Apaga la luz del baño
Cocina	Co	1	Co 1 Enciende la luz de la cocina
		0	Co 0 Apaga la luz de la cocina
Sala	Sa	1	Sa 1 Enciende la luz de la sala
		0	Sa 0 Apaga la luz de la sala
Dormitorio Master	Ma	1	Dm 1 Enciende la luz del dormitorio master
		0	Dm 0 Apaga la luz del dormitorio master
Dormitorio 1	D1	1	D1 1 Enciende la luz del dormitorio 1
		0	D1 0 Apaga la luz del dormitorio 1
Dormitorio 2	D2	1	D2 1 Enciende la luz del dormitorio 2
		0	D2 0 Apaga la luz del dormitorio 2
Entrada	En	1	En 1 Enciende la luz del Pasillo de entrada
		0	En 0 Apaga la luz del Pasillo de entrada
Parte Externa	Pe	1	Pe 1 Enciende la luz de la parte externa
		0	Pe 0 Apaga la luz de la parte externa
Audio	Mu	1	Mu 1 Enciende el radio
		0	Mu 0 Apaga el radio
Modo	Aut	1	Aut1 Modo Automático en función de las horas configuradas.
		0	Aut0 Modo remoto en función de los comandos recibidos en el mensaje de texto.

Esta verificación al igual que la extracción del número celular se hace a partir del mensaje recibido y que es guardado en un espacio de memoria RAM de la tarjeta Arduino Uno a partir del esquema y código presentado en la figura, en la cual se puede observar en el momento

en que se recibe un mensaje de texto el módulo GSM, inicialmente existe un contador que cuenta los caracteres recibidos, a partir de lo cual se identifica un formato de envío que contiene cuatro partes, la antesala en la cual se establece que es una respuesta a un comando de mensajes de texto, el número celular, el cual viene entre comillas, por lo que se utiliza la comilla de fin del número para establecer a partir del mismo que los 9 caracteres anteriores corresponden al número celular que envió el SMS, luego de lo cual se provee información de fecha y hora en la que se recibió el mensaje, lo cual ocupa en conjunto con lo demás 64 caracteres, por lo que a partir del carácter 57 se debe proceder a guardar el mensaje de texto como tal, calculado como espacio máximo para este tipo de mensajes 45 caracteres por lo que de existir más estos no serán almacenados para posteriormente validar el comando y ejecutarlo.

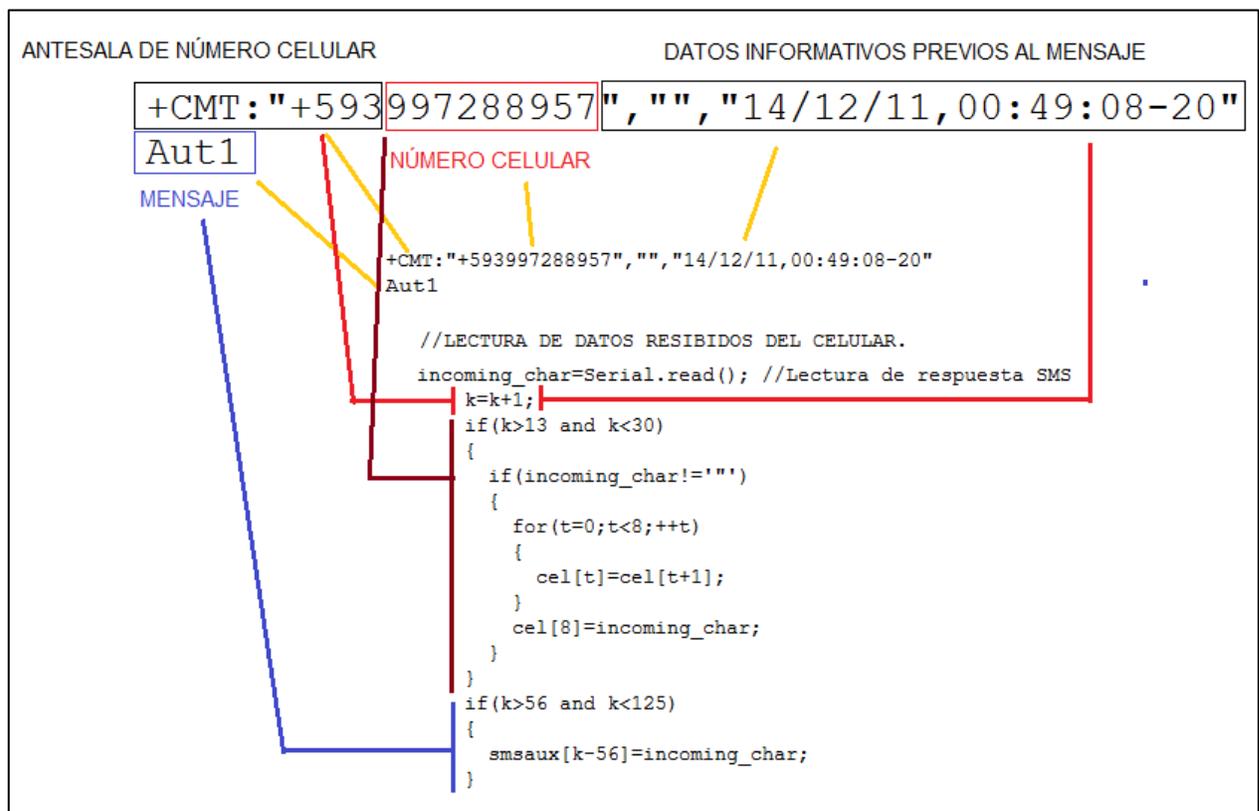


Figura 31. Código y descripción del proceso de extracción del número y mensaje de texto
Fuente: El autor

Para este ejemplo si el numero celular es una de los registrados en la memoria EEPROM de la tarjeta Arduino se valida y al verificar el mensaje de texto se encendería la cambiaria a modo automático. La descripción del funcionamiento en modo remoto se ilustra en el

siguiente diagrama de flujo, en el cual luego de verificar la existencia de un mensaje de texto a partir del valor del contador k de bytes recibidos, se inicia la identificación de la validez del número celular dentro de un ciclo FOR que lee de la memoria EEPROM los 10 números celulares guardados uno por uno y en caso de existir uno correcto continúa con la verificación de comandos a través de otro bloque FOR que lo busca en el mensaje guardado comparándolo con los comandos y ordenes descritos anteriormente, de encontrar alguna coincidencia, ejecuta la acción correspondiente, ya sea de encendido o apagado y continúa buscando dentro del texto algún otro comando hasta completar todo el mensaje. Todo este ciclo lo ejecuta 2 veces por minuto validado a partir de un IF que compara el segundo del sistema, detectando los 5 o 35 segundos para ejecutar la secuencia.

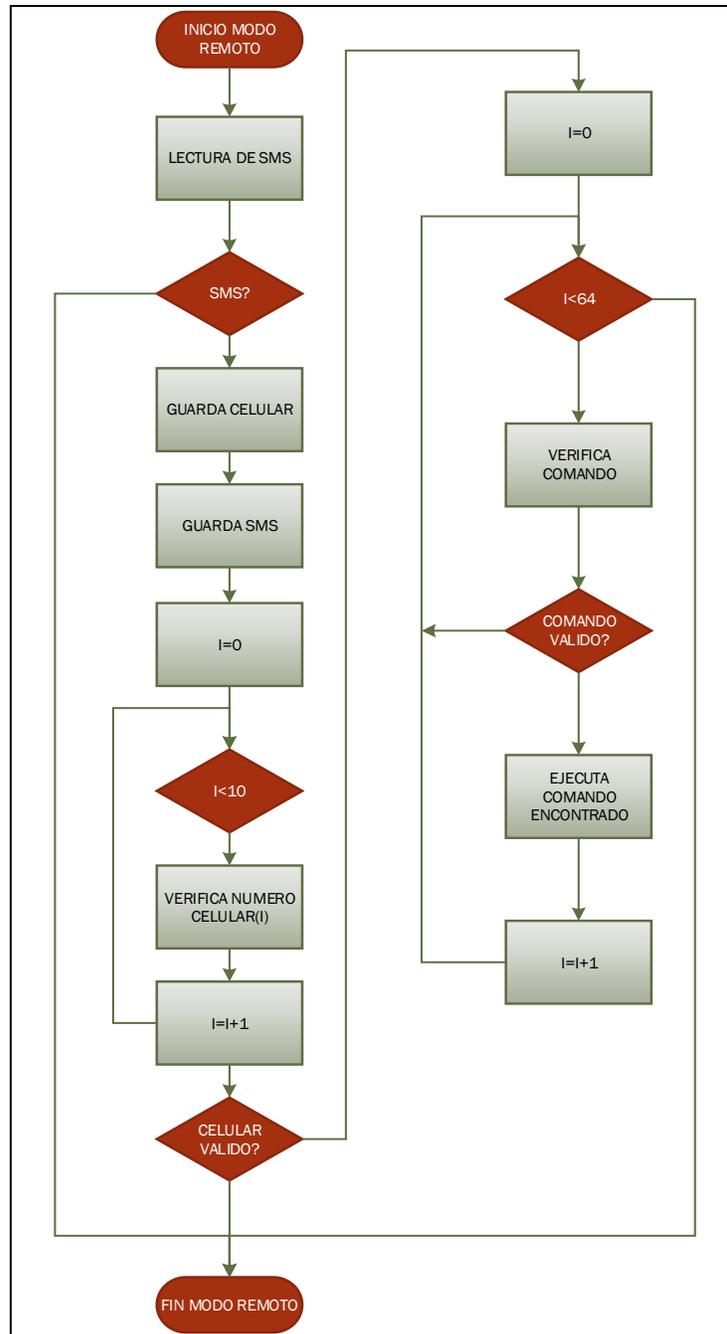


Figura 32. Diagrama de Flujo de Modo Remoto
Fuente: El autor

Para el encendido y apagado de los aparatos se utiliza el comando `digitalWrite(5, LOW)`; para el encendido y el comando `digitalWrite(5, HIGH)`; para le apagado, en todos 8 ambientes a controlar la iluminación donde el primer dato identifica el bit de salida que activa el relé del módulo correspondiente que activa la luz, los cuales están conectados

través del módulo relés a la tarjeta Arduino según se indica en la tabla.

Tabla 12. Descripción de la conexión y asignación de Ambientes entre Tarjeta Arduino y Módulo de Relés
Fuente: El autor

AMBIENTE	SALIDA DIGITAL MÓDULO ARDUINO	SALIDA DEL MÓDULO DE RELÉS
Baño	2	8
Cocina	3	7
Sala	4	6
Dormitorio Master	5	5
Dormitorio 1	6	4
Dormitorio 2	7	3
Entrada	8	2
Parte Externa	10	1

Como ejemplo de programación para el encendido y apagado de la sala se coloca en la tabla el código utilizado en cada caso tanto en modo automático como remoto, recordando que estos están metidos dentro de un ciclo FOR, por lo que para este caso $i=2$.

Tabla 13. Ejemplo de programación de prendido y apagado de la Sala en modo remoto y automático
Fuente: El autor

AMBIENTE	MODO	PROGRAMACIÓN
COCINA	AUTOMÁTICO	<p>PRENDIDO:</p> <pre>if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==2) { digitalWrite(4, LOW); }</pre> <p>APAGADO</p> <pre>if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==2) { digitalWrite(4, HIGH); }</pre>
COCINA	REMOTO	<p>PRENDIDO</p> <pre>if(smsaux[j]=='S' and smsaux[j+1]=='a' and smsaux[j+2]==' ' and smsaux[j+3]=='1') { digitalWrite(4, LOW); }</pre> <p>APAGADO</p> <pre>if(smsaux[j]=='S' and smsaux[j+1]=='a' and smsaux[j+2]==' ' and smsaux[j+3]=='0') { digitalWrite(4, HIGH); }</pre>

Para el funcionamiento del radio receptor FM se requiere la configuración de la comunicación I2C a través de la librería `Wire.h` propia de Arduino y maneja tres variables 2 tipo char y una tipo int, para la inicialización se utiliza el comando `Wire.begin()`; que inicializa la comunicación.

Para el control del funcionamiento se desarrollan 2 subrutinas, una de encendido denominada `radio_on(104.5)`; que sintoniza la emisora de la frecuencia enviada, por ejemplo, en este caso sintoniza la emisora “Radio América”, para la ciudad de Quito, a continuación se coloca el código de la subrutina de encendido, tomada a partir de tutoriales e información obtenida en foros:

```
void radio_on(double frequency)
{
    frequencyB=4*(frequency*1000000+225000)/32768; //calculating PLL word
    frequencyH=frequencyB>>8;
    frequencyL=frequencyB&0XFF;
    delay(100);
    Wire.beginTransmission(0x60); //Dirección TEA5767
    Wire.write(frequencyH);
    Wire.write(frequencyL);
    Wire.write(0xB0);
    Wire.write(0x10);
    Wire.write(0x00);
    Wire.endTransmission();
}
```

Para el apagado del sistema, se tiene no un control de encendido sino más bien se coloca en un estado de mute a partir de la subrutina `radio_mute(102.5)` cuyo código se indica a continuación:

```
void radio_mute(double frequency)
{
    frequencyB=4*(frequency*1000000+225000)/32768; //calculating PLL word
    frequencyH=frequencyB>>8;
    frequencyL=frequencyB&0XFF;
    delay(100);
    Wire.beginTransmission(0x60); //Dirección TEA5767
    Wire.write(frequencyH);
    Wire.write(frequencyL);
    Wire.write(0xB6);
    Wire.write(0x10);
    Wire.write(0x00);
    Wire.endTransmission();
}
```

3.3. Instalación en la maqueta.

La implementación se realizó en un prototipo montado sobre una maqueta de dimensiones 110 X 65 cm que representa una casa tipo de una planta con ambientes como: baño, sala, cocina, 3 dormitorios, pasillo y exterior, en el cual se colocaron las luces e interruptores de control manual de las luces para cada ambiente. En la figura se muestra el plano de la maqueta.



Figura 33. Plano y foto de la maqueta realizada.
Fuente: El autor

A partir de la maqueta construida se realizó el cableado y colocación de los focos e interruptores para cada uno de los ambientes a controlar y también se instaló el sistema de audio colocando dos parlantes conectados al Módulo receptor FM a través de un amplificador de audio pequeño, el esquema de conexión se ilustra en la figura

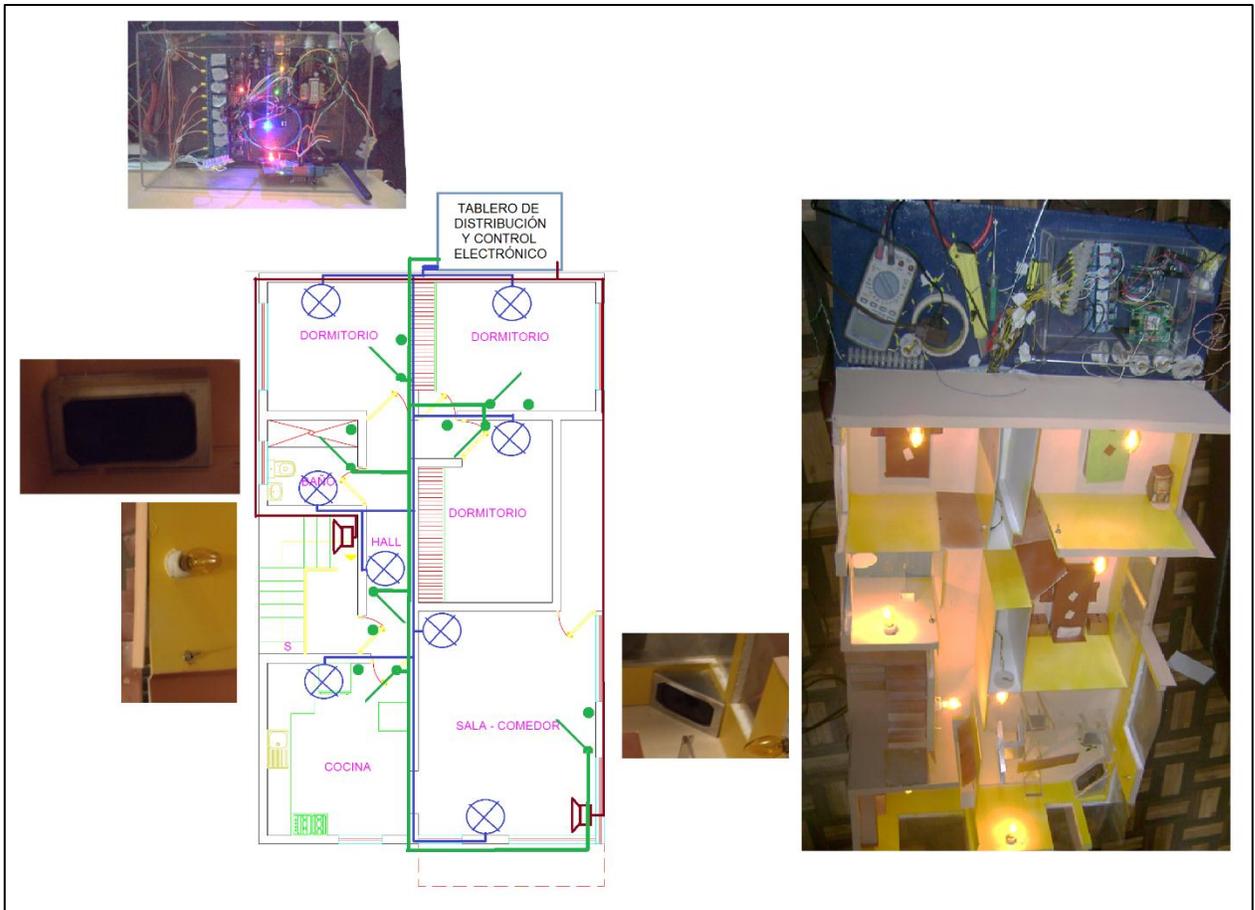


Figura 34. Esquema y foto de conexión y distribución eléctrica en la maqueta
Fuente: El autor

3.4 Pruebas y Resultados

Dentro las pruebas y resultados obtenidos se evaluaron individualmente cada segmento, donde se comprobó cada etapa, existieron partes como la de relés que se implementaron y probaron previamente en protoboard, antes de la adquisición del módulo correspondiente, donde se realizaron pruebas, se comprobaron y midieron aspectos como consumo de corriente de los relés y se evidencio la necesidad de una fuente de corriente superior a 1Amp.

Dentro del módulo GSM se comprobó la recepción de mensajes y se realizaron varias pruebas de métodos de extracción del número celular y el mensaje, siendo el implementado el que presento la mejor respuesta tanto en eficiencia como en pérdida de datos, también se vio la necesidad de separar el módulo GSM y no implementarlo sobrepuesto sobre la tarjeta

Arduino debido a que no permitía el trabajo con el módulo receptor FM ya que bloqueaba los canales analógicos.

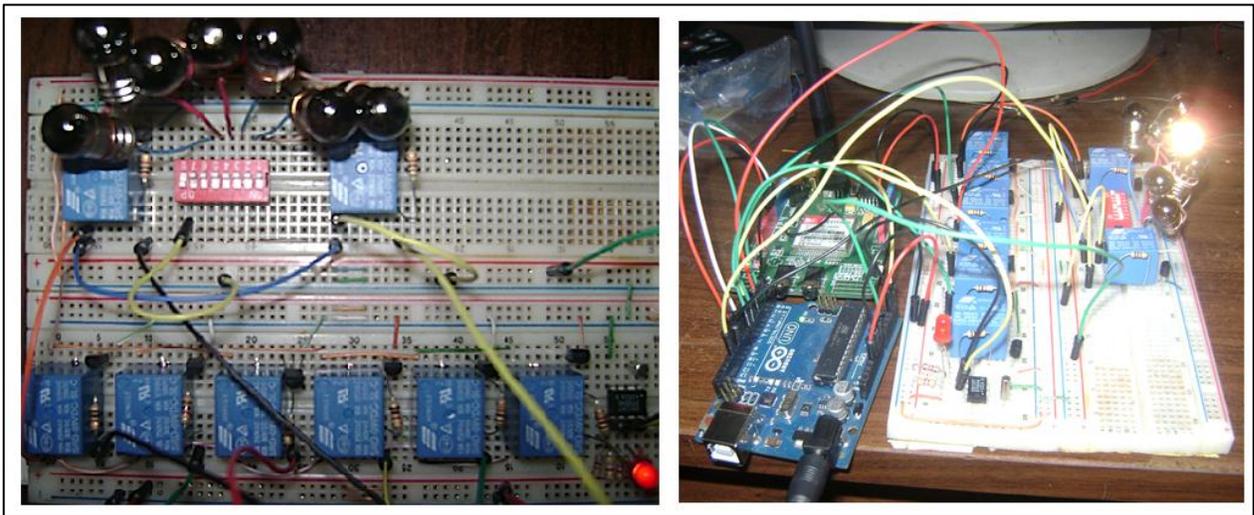


Figura 35 Pruebas de Relés y módulo GSM

Fuente: El autor

De igual forma se comprobaron los elementos antes de instalarlos en la maqueta y dentro de ella, de las pruebas realizadas se desprende que el sistema funciona de forma adecuada sin presentar mayores errores en su funcionamiento.

Finalmente, para fines demostrativos, se planteó y recalcularon los tiempos de tal forma que para aplicación didáctica las 24 horas del día son simuladas en 12 minutos de funcionamiento a través de la tabla que se muestra. Adicionalmente se incrementó un comando que hace que cada vez que los minutos de RTC lleguen a 12 se enceren simulado el cambio del día y se cambiaron las comparaciones de horas por minutos y las de minutos por segundos.

De la siguiente manera:

```

if (myRTC.minutes==12)
{
    myRTC.setDS1302Time (0, 0, 13, 6, 3, 2, 2015);
}

```

Tabla 14. Tabla de conversión de tiempos de 24 horas a 12 minutos
Fuente: El autor

No	AMBIENTE	HORAS ENCENDIDO	HORAS APAGADO	TIEMPO ENCENDIDO	TIEMPO APAGADO
1	Baño	19:43	19:48	9:51	9:54
		21:13	21:23	10:36	10:41
		23:26	23:30	11:43	11:45
		0:01	0:03	0:00	0:01
		3:35	3:38	1:47	1:49
		5:30	5:45	2:45	2:52
2	Cocina	19:23	20:31	9:41	10:15
		21:34	21:44	10:47	10:52
		3:39	3:43	1:49	1:51
		5:40	6:00	2:50	3:00
3	Sala	18:00	22:00	9:00	11:00
		4:36	6:00	2:18	3:00
4	Dormitorio Master	20:18	23:00	10:09	11:30
		5:00	5:45	2:30	2:52
5	Dormitorio 1	18:25	20:09	9:12	10:04
		21:30	22:54	10:45	11:27
6	Dormitorio 2	18:13	19:41	9:06	9:50
		20:57	23:34	10:28	11:47
7	Entrada	19:00	22:12	9:30	11:06
		5:30	6:24	2:45	3:12
8	Parte Externa	17:00	19:00	8:30	9:30
		5:49	7:30	2:54	3:45
		13:00	16:09	6:30	8:04
9	Audio	18:00	21:00	9:00	10:30
		4:36	8:00	2:18	4:00
		12:00	15:24	6:00	7:42

3.5 Costos

Con respecto a los costos generales del sistema se plantean los analices de costos de la parte eléctrica y electrónica y los cuales se reflejan en las tablas.

Tabla 15. Costos parte eléctrica y electrónica de etapa de control
Fuente: El autor

MATERIALES	COSTO
Tarjeta Arduino Uno	33,00
Módulo SHIELD GSM SIM900	80,00
Módulo de Relés	25,00
Módulo RTC	8,00
Módulo receptor FM	25,00
Amplificador de Audio	8,00
Fuente conmutada de 5VDC 1.5 A	7,00
Puentes eléctricos	6,00
Placa Universal	3,00
Chip Movistar	10,00
Caja de Acrílico	25,00
Conectores, borneras y terminales	10,00
Etiquetas	2,00
Varios	10,00
Mano de obra (100 horas a \$5,00)	510,00
TOTAL	752,00

Tabla 16. Costos del prototipo, maqueta y sistema eléctrico de iluminación.
Fuente: El autor

MATERIALES	COSTO
Materiales elaboración maqueta	100,00
Material Eléctrico (Boquillas, cables y focos)	20,00
Mano de obra construcción maqueta	150,00
TOTAL	270,00

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Se determinó las etapas y dispositivos del proyecto como suministro eléctrico, el procesamiento de datos con el Módulo Arduino UNO, el control de potencia a través del módulo de relés de 8 canales, sincronización en tiempo real (RTC), la comunicación inalámbrica con la tarjeta SIM900 y la reproducción de audio implementando un sistema modular de fácil manipulación y abierto a posibles cambios y mejoras del sistema.
- Se utilizó un sistema de desarrollo del equipo basado en la integración de diferentes tecnologías las cuales fueron previamente comprobadas por separado por luego ser unidas en un solo sistema, lo cual facilitó la implementación de un protocolo de pruebas y corrección de errores.
- Se logró la recepción y manipulación del texto de mensajes SMS, a partir de la programación y manipulación de módulo shield SIM900 a partir de comandos AT, lo que permitió el manejo y control del sistema de forma remota.
- La programación de la secuencia automática de encendido-apagado de luces en una temporización, se estableció de acuerdo a un previo estudio de accionamientos rutinarios de iluminación en cada habitación de la vivienda tipo, lo cual se implementó a través de su almacenaje en la memoria EEPROM del Arduino y su comparación con los datos de tiempo arrojados por el reloj en tiempo real, a partir de lo que se obtuvo un control de sistema temporizado.
- La implantación del circuito eléctrico de iluminación en el prototipo se ejecutó según características eléctricas similares respecto a la de una vivienda a escala real (110VCA), se colocaron interruptores manuales para el control local de la iluminación en paralelo a relés controlados a partir del módulo Arduino, obteniéndose un circuito de control compacto de fácil montaje y conexión con las líneas eléctricas domiciliarias.
- Se evidencia que el desarrollo de este tipo de aplicaciones puede ser de gran ayuda y permitir la domotización de sistemas residenciales de seguridad a muy bajos costos, además de mantener la privacidad interna de la residencia.

Recomendaciones

- Se podría ampliar la capacidad de control de periféricos del sistema, reemplazando el módulo Arduino Uno, por otro con mayor número de entradas, salidas y de mayor capacidad de almacenamiento.
- Se lograría acoplar a sistemas residenciales, industriales, urbanísticos, con facilidad, previo un análisis de los requerimientos de corrientes y voltajes de los mismos.
- Se puede acoplar un sistema de retroalimentación de SMS, hacia el usuario, en la cual se notifique la recepción de SMS, o información que algún dispositivo/s se ha activado.
- Acoplamiento de sensores, para reconocer el estado de los dispositivos controlados para un mejor confort y seguridad del usuario., estableciendo esto en una vivienda domótica
- A escala real habrá que cablear un hilo eléctrico adicional a los de las instalaciones convencionales desde el relé hacia el punto eléctrico entre el foco-interruptor.
- Apoyar e incentivar el desarrollo de sistemas similares y generar productos tecnológicos que brinden confort y seguridad a las personas y sus bienes, ya que esto aporta a mejorar su calidad de vida y al desarrollo tecnológico del país.

BIBLIOGRAFÍA

HDL Ecuador. (2014).

http://hdlecuador.com/4/index.php?option=com_k2&view=item&id=9:sistema-de-solución-de-automatización-para-el-hogar&Itemid=132

INEC. (2014). 1,2 millones de ecuatorianos tienen un teléfono inteligente (Smartphone).

<http://www.ecuadorencifras.gob.ec/12-millones-de-ecuatorianos-tienen-un-telefono-inteligente-smartphone>.

Murphy, T. (2014). VIRTUABOTIX. Virtuabotix DS1302 Real Time Clock Module Pin-out and coding guide <https://www.virtuabotix.com/virtuabotix-ds1302-real-time-clock-module-pin-out-coding-guide/>

Philips. (2007). TEA5767HN_5. Low-power FM stereo radio for handheld applications.

Senplades. (2013). Plan Nacional para el Buen Vivir 2013-2017. Buen Vivir PLAN NACIONAL 2013-2017. Quito: Senplades.

Shanghai SIMCom wireless solutions Ltd. (2010). SIM900 AT Command Manual. SIM900_AT Command Manual_V1.03. Shanghai:SIMCom

Shanghai SIMCom wireless solutions Ltd. (2009). SIM900 Hardware Design. Hardware Design SIM900_HD_V1.01. Shanghai:SIMCom

Simon, M. (2012) 30 PROYECTOS CON ARDUINO. Madrid: Estribor.

Torrente, O.(2013).ARDUINO Curso práctico de formación. Mexico DF:Alfaomega.

<http://openhardware.pe> (2013).

Movistar.(2013). <http://comunidad.movistar.es/t5/Soporte-Técnico-Terminales/GSM-Cómo-funciona-la-telefonía-móvil/td-p/1383296>

ANEXO A.1:

MANUAL DE USUARIO

SISTEMA DE CONTROL AUTOMATICO Y REMOTO DE ENCENDIDO APAGADO DE ILUMINACIÓN RESIDENCIAL

1.0



1. GENERALIDADES

INTRODUCCIÓN

El sistema de control automático y remoto del sistema de iluminación residencial surgió de la necesidad de colaborar en control de la seguridad de los ciudadanos, mediante la simulación de presencia humana en las viviendas con el propósito de disuadir a posibles hurtos/robos. El presente manual pretende transmitir la estructura, conceptos e información básica de la operación del sistema, con la finalidad de que conociendo su funcionamiento los usuarios lo puedan utilizar de manera efectiva. Por lo mencionado, es de suma importancia leer el manual que se detalla a continuación antes y durante la aplicación del sistema.

OBJETIVO

El principal objetivo del presente manual, es ayudar al cliente, a que conozca los requisitos y operación del sistema, así como una idea clara del alcance que tiene a través de la explicación ilustrada de cada una de las opciones que lo conforman.

A QUIEN VA DIRIGIDO.

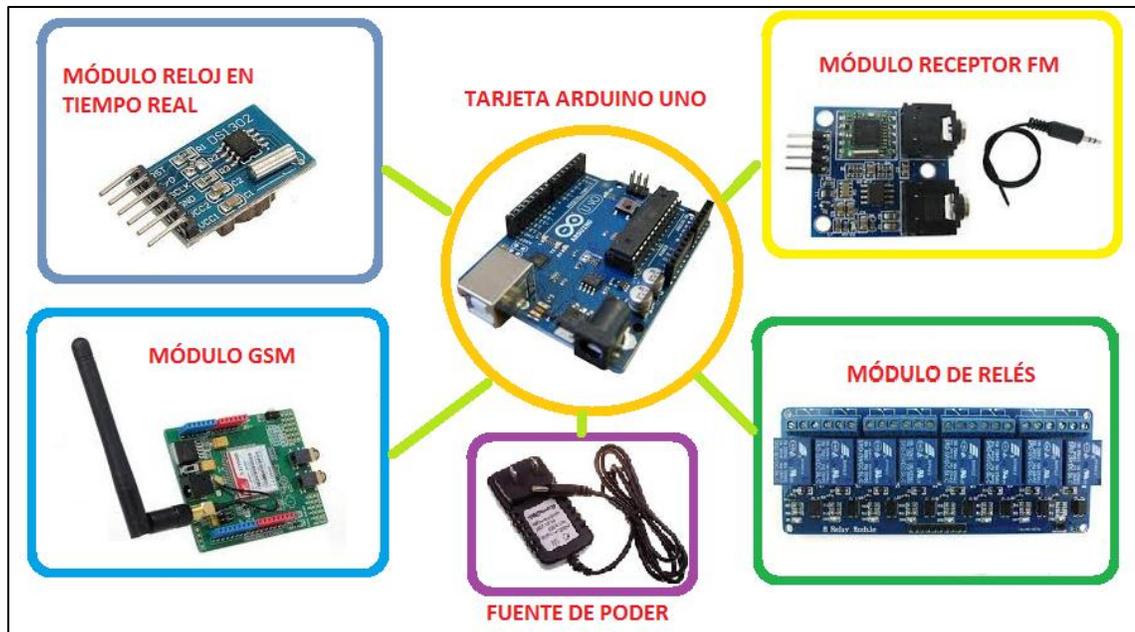
Este manual va dirigido a los clientes finales involucrados en la operación del sistema.

2. INTRODUCCIÓN DEL SISTEMA.

En esta etapa se definirá los requerimientos recomendados de hardware y software para la correcta operación y funcionamiento del sistema. Además, se detallan los elementos del entorno con el que el usuario debe relacionarse para su operación. Los requerimientos se fundamentan principalmente, en la rutina de encendido/apagado de la iluminación residencial. Así como los números celulares desde los cuales se realizará el control. En definitiva los requerimientos de la aplicación son los siguientes:

ELEMENTOS

A continuación se describe brevemente las partes constitutivas del sistema de control de iluminación residencial, con sus respectivos dispositivos



REQUERIMIENTOS DE HARDWARE Y SOFTWARE

Los requisitos recomendados de hardware y software son:

Hardware:

- ✓ Suministro de energía 110VCA, para la alimentación del sistema de control.
- ✓ La disponibilidad a la caja de distribución (Caja de Breakers/Caja de dispersión de iluminación), así como la conexión de la red (110VCA) del sistema de iluminación, con el sistema de control, estará a cargo del servicio técnico del fabricante/vendedor, previo a una visita técnica y análisis de los requerimientos técnicos de conexión, bajo acuerdo de las partes vendedor y comprador.
- ✓ Tarjeta SIM, de cualquier operadora celular, y que se mantenga activa durante la operación del sistema.

Software:

- ✓ Disponibilidad de SMS desde el celular/es que van a controlar.
- ✓ El cliente proporcionará la secuencia de encendido-apagado de las habitaciones en función del tiempo, para poder programar en el sistema y aplicar la simulación en modo automático.

Ejemplo:

No	AMBIENTE	HORAS ENCENDIDO	HORAS APAGADO
1	Baño	19:43 21:13 23:26 0:01 3:35 5:30	19:48 21:23 23:30 0:03 3:38 5:45
2	Cocina	19:23 21:34 3:39 5:40	20:31 21:44 3:43 6:00
3	Sala	18:00 4:36	22:00 6:00
4	Dormitorio Master	20:18 5:00	23:00 5:45
5	Dormitorio 1	18:25 21:30	20:09 22:54
6	Dormitorio 2	18:13 20:57	19:41 23:34
7	Entrada	19:00 5:30	22:12 6:24
8	Parte Externa	17:00 5:49 13:00	19:00 7:30 16:09
9	Audio	18:00 4:36 12:00	21:00 8:00 15:24

- ✓ La asignación de códigos y dato de encendido o apagado será proporcionado por el usuario. La extensión del código del ambiente o habitación podrá ser 1 a 10 caracteres, solo letras, y para el dato que enciende o apaga la habitación podrá ser de 1 a 4 caracteres (1/0;ON/OFF)

En el ejemplo se establece

AMBIENTE	CODIGO	EJEMPLO
Baño	Ba	Ba 1 Enciende la luz del baño Ba 0 Apaga la luz del baño
Cocina	Co	Co 1 Enciende la luz de la cocina Co 0 Apaga la luz de la cocina
Sala	Sa	Sa 1 Enciende la luz de la sala Sa 0 Apaga la luz de la sala
Dormitorio Master	Ma	Dm 1 Enciende la luz del dormitorio master Dm 0 Apaga la luz del dormitorio master
Dormitorio 1	D1	D1 1 Enciende la luz del dormitorio 1 D1 0 Apaga la luz del dormitorio 1
Dormitorio 2	D2	D2 1 Enciende la luz del dormitorio 2 D2 0 Apaga la luz del dormitorio 2
Entrada	En	En 1 Enciende la luz del Pasillo de entrada En 0 Apaga la luz del Pasillo de entrada
Parte Externa	Pe	Pe 1 Enciende la luz de la parte externa Pe 0 Apaga la luz de la parte externa
Audio	Mu	Mu 1 Enciende el radio Mu 0 Apaga el radio
Modo	Aut	Aut1 Modo Automático en función de las horas configuradas. Aut0 Modo remoto en función de los comandos recibidos en el mensaje de texto.

OPERACIÓN DEL SISTEMA

- ✓ El control del sistema cuenta con un esquema general del proceso realizado por el usuario desde su celular:

SMS: Aut1 + 09... (# Cel Del Sistema) + **SEND**

Activa la secuencia de encendido-apagado de las lámparas de toda la vivienda de forma programada y bajo secuencia de tiempo.

SMS: Aut0 + 09... (# Cel Del Sistema) + **SEND**

Desactiva la secuencia de encendido-apagado de las lámparas de toda la vivienda de forma programada y bajo secuencia de tiempo.

SMS: Mu 1En 1Ma 1Ba 1 + 09... (# Cel Del Sistema) + **SEND**

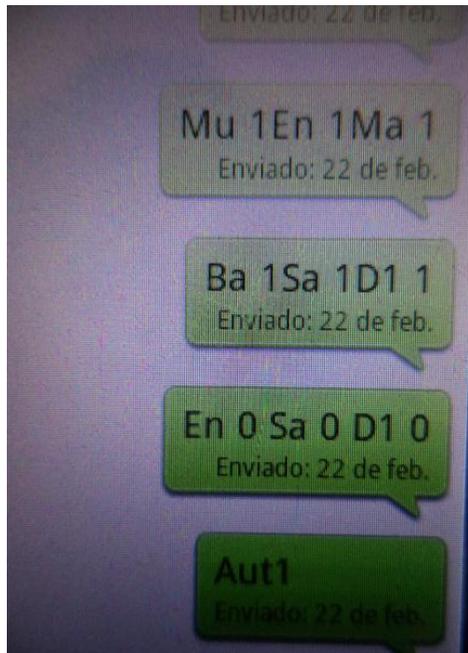
Enciende el sistema de audio, enciende la iluminación de entrada, dormitorio master y baño.

SMS: Mu 0En 1Ma 0Ba 1 + 09... (# Cel Del Sistema) + **SEND**

Apaga el sistema de audio, enciende o mantiene encendido la entrada y baño, y apaga la luz del dormitorio master.

Nota.-De acuerdo a la extensión o disponibilidad de caracteres máximo por mensaje enviado, se podrá digitar todas las áreas de la vivienda con sus respectivos mandos en un solo SMS; así como también se podrá enviar una solo área en el SMS, esto por efectos de ahorro de dinero.

Esquema ejemplificado de los envíos de los SMS con los códigos, para la activación de operación automática ó remota y viceversa.



RESETEO

- Es posible que el sistema bajo condiciones normales sufra algún inconveniente, bajo las cuales se recomienda desconectar el suministro eléctrico de alimentación del sistema de control y esperar un tiempo de 10seg., volver a conectarlo verificar su funcionamiento y si continua el desperfecto comunicarse con servicio técnico del fabricante/vendedor.

GARANTÍA

- La garantía sobre defectos de fabricación será de seis meses, tiempo en el cual se brindará dos visitas técnicas, para mantenimiento, cambios en la programación, conexión de red de iluminación, cambio de tarjeta SIM
- Posterior a los seis meses que dure la garantía, cualquier requerimiento del cliente tendrá su costo, previamente pactado con el cliente.

PREGUNTAS FRECUENTES

¿Qué hacer en caso de corte de suministro eléctrico?

El sistema de control de la iluminación al igual que el sistema iluminación convencional estará deshabilitado de la simulación. Sin embargo en conteo de la hora estará continuando, ya que el módulo de control de tiempo real (RTC) posee una batería interna propia del sistema, para en caso de retorno del suministro eléctrico retome su estado según sea en modo automático o remoto.

CONTACTOS

Para soluciones y soporte técnico comunicarse a:
(02)3068339-0992912776 Servicio Técnico
Quito - Ecuador

ANEXO A.2: Programa para la simulación de presencia

```
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <virtuabotixRTC.h>
#include <Wire.h>

unsigned char frequencyH=0;
unsigned char frequencyL=0;
unsigned int frequencyB;

int radio=0;
char incoming_char=0;
int t=0;
char celaux;
char cel[9];
int contacel;
char smsaux[45];
int contasms;
int prueba=0;
int segaux=0;
int horaux;
int miniaux;
int i;
int j;
int k=0;
int aut;
virtuabotixRTC myRTC(11, 12, 13);

void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(10, OUTPUT);
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(8, HIGH);
  digitalWrite(10, HIGH);

  pinMode(9, OUTPUT); //PRENDE MODULO GSM

  myRTC.setDS1302Time(30, 8, 13, 6, 3, 2, 2015);
  Wire.begin();
  Serial.begin(19200); // for serial monitor
  SIM900power(); // turn on shield
```

```

delay(2000); // give time to log on to network.
Serial.print("ATE0\r");
Serial.println();
Serial.print("AT+CMGF=1\r"); // set SMS mode to text
Serial.println();
delay(100);
radio_on(104.5);
digitalWrite(2, LOW);
digitalWrite(4, LOW);
digitalWrite(6, LOW);
digitalWrite(8,LOW);
digitalWrite(3, HIGH);
digitalWrite(5, HIGH);
digitalWrite(7, HIGH);
digitalWrite(10,HIGH);
delay(3000);
digitalWrite(2, HIGH);
digitalWrite(4, HIGH);
digitalWrite(6, HIGH);
digitalWrite(8, HIGH);
digitalWrite(3, LOW);
digitalWrite(5, LOW);
digitalWrite(7, LOW);
digitalWrite(10,LOW);
delay(3000);
apaga();
delay(3000);
aut=EEPROM.read(500);
/*
//No celular 1
EEPROM.write(0, '9');
EEPROM.write(1, '9');
EEPROM.write(2, '7');
EEPROM.write(3, '2');
EEPROM.write(4, '8');
EEPROM.write(5, '8');
EEPROM.write(6, '9');
EEPROM.write(7, '5');
EEPROM.write(8, '7');

//No celular 2
EEPROM.write(10, '9');
EEPROM.write(11, '9');
EEPROM.write(12, '2');
EEPROM.write(13, '9');
EEPROM.write(14, '1');
EEPROM.write(15, '2');
EEPROM.write(16, '7');
EEPROM.write(17, '7');
EEPROM.write(18, '6');

//No celular 3
EEPROM.write(20, '9');

```

```
EEPROM.write(21,'8');
EEPROM.write(22,'4');
EEPROM.write(23,'9');
EEPROM.write(24,'7');
EEPROM.write(25,'9');
EEPROM.write(26,'0');
EEPROM.write(27,'2');
EEPROM.write(28,'5');

//No celular 4
EEPROM.write(30,'9');
EEPROM.write(31,'8');
EEPROM.write(32,'1');
EEPROM.write(33,'8');
EEPROM.write(34,'0');
EEPROM.write(35,'6');
EEPROM.write(36,'8');
EEPROM.write(37,'7');
EEPROM.write(38,'5');

//Baño HORAS PRENDIDO
EEPROM.write(100,9);
EEPROM.write(101,10);
EEPROM.write(102,11);
EEPROM.write(103,0);
EEPROM.write(104,1);
EEPROM.write(105,2);

//Baño MINUTOS PRENDIDO
EEPROM.write(300,51);
EEPROM.write(301,36);
EEPROM.write(302,43);
EEPROM.write(303,1);
EEPROM.write(304,47);
EEPROM.write(305,45);

//Baño HORAS APAGADO
EEPROM.write(200,9);
EEPROM.write(201,10);
EEPROM.write(202,11);
EEPROM.write(203,0);
EEPROM.write(204,1);
EEPROM.write(205,2);

//Baño MINUTOS APAGADO
EEPROM.write(400,54);
EEPROM.write(401,41);
EEPROM.write(402,45);
EEPROM.write(403,3);
EEPROM.write(404,49);
EEPROM.write(405,52);

//Cocina HORAS PRENDIDO
EEPROM.write(110,9);
```

```
EEPROM.write(111,10);
EEPROM.write(112,1);
EEPROM.write(113,2);

//Cocina MINUTOS PRENDIDO
EEPROM.write(310,41);
EEPROM.write(311,47);
EEPROM.write(312,49);
EEPROM.write(313,50);

//Cocina HORAS APAGADO
EEPROM.write(210,10);
EEPROM.write(211,10);
EEPROM.write(212,1);
EEPROM.write(213,3);

//Cocina MINUTOS APAGADO
EEPROM.write(410,15);
EEPROM.write(411,52);
EEPROM.write(412,51);
EEPROM.write(413,0);

//Sala HORAS PRENDIDO
EEPROM.write(120,9);
EEPROM.write(121,2);

//Sala MINUTOS PRENDIDO
EEPROM.write(320,0);
EEPROM.write(321,18);

//Sala HORAS APAGADO
EEPROM.write(220,11);
EEPROM.write(221,3);

//Sala MINUTOS APAGADO
EEPROM.write(420,0);
EEPROM.write(421,1);

//Master HORAS PRENDIDO
EEPROM.write(130,10);
EEPROM.write(131,2);

//Master MINUTOS PRENDIDO
EEPROM.write(330,9);
EEPROM.write(331,30);

//Master HORAS APAGADO
EEPROM.write(230,11);
EEPROM.write(231,2);

//Master MINUTOS APAGADO
EEPROM.write(430,30);
EEPROM.write(431,52);
```

```
//Dorm1 HORAS PRENDIDO
EEPROM.write(140,9);
EEPROM.write(141,10);

//Dorm1 MINUTOS PRENDIDO
EEPROM.write(340,12);
EEPROM.write(341,45);

//Dorm1 HORAS APAGADO
EEPROM.write(240,10);
EEPROM.write(241,11);

//Dorm1 MINUTOS APAGADO
EEPROM.write(440,4);
EEPROM.write(441,27);

//Dorm2 HORAS PRENDIDO
EEPROM.write(150,9);
EEPROM.write(151,10);

//Dorm2 MINUTOS PRENDIDO
EEPROM.write(350,6);
EEPROM.write(351,28);

//Dorm2 HORAS APAGADO
EEPROM.write(250,9);
EEPROM.write(251,11);

//Dorm2 MINUTOS APAGADO
EEPROM.write(450,50);
EEPROM.write(451,47);

//Entrada HORAS PRENDIDO
EEPROM.write(160,9);
EEPROM.write(161,2);

//Entrada MINUTOS PRENDIDO
EEPROM.write(360,30);
EEPROM.write(361,45);

//Entrada HORAS APAGADO
EEPROM.write(260,11);
EEPROM.write(261,3);

//Entrada MINUTOS APAGADO
EEPROM.write(460,6);
EEPROM.write(461,12);

//Musica HORAS PRENDIDO
EEPROM.write(170,9);
EEPROM.write(171,2);
EEPROM.write(172,6);

//Musica MINUTOS PRENDIDO
```

```

EEPROM.write(370,0);
EEPROM.write(371,18);
EEPROM.write(372,0);

//Musica HORAS APAGADO
EEPROM.write(270,10);
EEPROM.write(271,4);
EEPROM.write(272,7);

//Musica MINUTOS APAGADO
EEPROM.write(470,30);
EEPROM.write(471,0);
EEPROM.write(472,42);

//Persiana HORAS PRENDIDO
EEPROM.write(180,8);
EEPROM.write(181,2);
EEPROM.write(182,6);

//Persiana MINUTOS PRENDIDO
EEPROM.write(380,30);
EEPROM.write(381,54);
EEPROM.write(382,30);

//Persiana HORAS APAGADO
EEPROM.write(280,9);
EEPROM.write(281,3);
EEPROM.write(282,8);

//Persiana MINUTOS APAGADO
EEPROM.write(480,30);
EEPROM.write(481,45);
EEPROM.write(482,4);

/  */
}

void SIM900power()
{
  digitalWrite(9, HIGH);
  delay(4000);
  digitalWrite(9, LOW);
  delay(1000);
}

void radio_on(double frequency)
{
  //frequency=92.9; //starting frequency
  frequencyB=4*(frequency*1000000+225000)/32768; //calculating PLL word
  frequencyH=frequencyB>>8;
  frequencyL=frequencyB&0XFF;
  delay(100);
  Wire.beginTransmission(0x60); //writing TEA5767

```

```

Wire.write(frequencyH);
Wire.write(frequencyL);
Wire.write(0xB0); //B6
Wire.write(0x10);
Wire.write(0x00);
Wire.endTransmission();
}
void radio_mute(double frequency)
{
//frequency=92.9; //starting frequency
frequencyB=4*(frequency*1000000+225000)/32768; //calculating PLL word
frequencyH=frequencyB>>8;
frequencyL=frequencyB&0XFF;
delay(100);
Wire.beginTransaction(0x60); //writing TEA5767
Wire.write(frequencyH);
Wire.write(frequencyL);
Wire.write(0xB6); //B6
Wire.write(0x10);
Wire.write(0x00);
Wire.endTransmission();
}
void apaga()
{
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
digitalWrite(5, HIGH);
digitalWrite(6, HIGH);
digitalWrite(7, HIGH);
digitalWrite(8, HIGH);
digitalWrite(10, HIGH);
radio_mute(100.5);
}

void loop()
{
if(myRTC.minutes==12)
{
myRTC.setDS1302Time(0, 0, 13, 6, 3, 2, 2015); //SEG MIN HOR DIS DIA
MES AÑO
}
if(Serial.available()>0)
{
//+CMT: "+593997288957","", "15/01/18,17:14:53-20"
//El numero +593997288957 ha solicitado que lo llame.
//+CMT: "+593997288957","", "14/12/11,00:49:08-20"
//hola

incoming_char=Serial.read(); //Get the character from the cellular
serial port.
k=k+1;
if(k>13 and k<30)
{

```

```

    if(incoming_char!='')
    {
        for(t=0;t<8;++t)
        {
            cel[t]=cel[t+1];
        }
        cel[8]=incoming_char;
    }
}
if(k>56 and k<125)
{
    smsaux[k-56]=incoming_char;
}

}
else
{
    myRTC.updateTime(); // Obtiene la fecha y hora del RTC
}
if(prueba==0 and (myRTC.seconds==5 or myRTC.seconds==35))
{
    prueba=1;
    k=0;
    Serial.print("AT+CNMI=2,2,0,0,0\r");
    Serial.println();
}
if(myRTC.seconds!=segaux)
{
    segaux=myRTC.seconds;
    if (aut==1)
    {
        for (i=0;i<9;++i)
        {
            for (j=0;j<10;++j)
            {
                horaaux=EEPROM.read(100+(i*10)+j);
                minuaux=EEPROM.read(300+(i*10)+j);
                if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==0)
                {
                    digitalWrite(2, LOW);
                }
                if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==1)
                {
                    digitalWrite(3, LOW);
                }
                if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==2)
                {
                    digitalWrite(4, LOW);
                }
                if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==3)
                {
                    digitalWrite(5, LOW);
                }
                if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==4)

```

```

{
    digitalWrite(6, LOW);
}
if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==5)
{
    digitalWrite(7, LOW);
}
if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==6)
{
    digitalWrite(8, LOW);
}
if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==7)
{
    if (radio==0)
    {
        radio_on(102.5);
    }
    if (radio==1)
    {
        radio_on(104.5);
    }
    if (radio==3)
    {
        radio_on(92.9);
    }
    if (radio==4)
    {
        radio_on(98.9);
    }
    radio=radio+1;
    if (radio>=5)
    {
        radio=0;
    }
}
if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==8)
{
    digitalWrite(10, LOW);
}
horaaux=EEPROM.read(200+i*10+j);
minuaux=EEPROM.read(400+i*10+j);
if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==0)
{
    digitalWrite(2, HIGH);
}
if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==1)
{
    digitalWrite(3, HIGH);
}
if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==2)
{
    digitalWrite(4, HIGH);
}
if(myRTC.minutes==horaaux and myRTC.seconds==minuaux and i==3)

```



```

        }
        if(smsaux[j]=='C' and smsaux[j+1]=='o' and smsaux[j+2]==' '
and smsaux[j+3]=='1')
        {
            digitalWrite(3, LOW);
        }
        if(smsaux[j]=='S' and smsaux[j+1]=='a' and smsaux[j+2]==' '
and smsaux[j+3]=='1')
        {
            digitalWrite(4, LOW);
        }
        if(smsaux[j]=='M' and smsaux[j+1]=='a' and smsaux[j+2]==' '
and smsaux[j+3]=='1')
        {
            digitalWrite(5, LOW);
        }
        if(smsaux[j]=='D' and smsaux[j+1]=='1' and smsaux[j+2]==' '
and smsaux[j+3]=='1')
        {
            digitalWrite(6, LOW);
        }
        if(smsaux[j]=='D' and smsaux[j+1]=='2' and smsaux[j+2]==' '
and smsaux[j+3]=='1')
        {
            digitalWrite(7, LOW);
        }
        if(smsaux[j]=='E' and smsaux[j+1]=='n' and smsaux[j+2]==' '
and smsaux[j+3]=='1')
        {
            digitalWrite(8, LOW);
        }
        if(smsaux[j]=='P' and smsaux[j+1]=='e' and smsaux[j+2]==' '
and smsaux[j+3]=='1')
        {
            digitalWrite(10, LOW);
        }
        if(smsaux[j]=='M' and smsaux[j+1]=='u' and smsaux[j+2]==' '
and smsaux[j+3]=='1')
        {
            if (radio==0)
            {
                radio_on(102.5);
            }
            if (radio==1)
            {
                radio_on(104.5);
            }
            if (radio==3)
            {
                radio_on(92.9);
            }
            if (radio==4)
            {
                radio_on(98.9);
            }
        }

```

```

    }
    radio=radio+1;
    if (radio>=5)
    {
        radio=0;
    }
}

if(smsaux[j]=='B' and smsaux[j+1]=='a' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    digitalWrite(2, HIGH);
}
if(smsaux[j]=='C' and smsaux[j+1]=='o' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    digitalWrite(3, HIGH);
}
if(smsaux[j]=='S' and smsaux[j+1]=='a' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    digitalWrite(4, HIGH);
}
if(smsaux[j]=='M' and smsaux[j+1]=='a' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    digitalWrite(5, HIGH);
}
if(smsaux[j]=='D' and smsaux[j+1]=='1' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    digitalWrite(6, HIGH);
}
if(smsaux[j]=='D' and smsaux[j+1]=='2' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    digitalWrite(7, HIGH);
}
if(smsaux[j]=='E' and smsaux[j+1]=='n' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    digitalWrite(8, HIGH);
}
if(smsaux[j]=='P' and smsaux[j+1]=='e' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    digitalWrite(10, HIGH);
}
if(smsaux[j]=='M' and smsaux[j+1]=='u' and smsaux[j+2]==' '
and smsaux[j+3]=='0')
{
    radio_mute(100.5);
}
}

```

```

        if(smsaux[j]=='A' and smsaux[j+1]=='u' and smsaux[j+2]=='t'
and smsaux[j+3]=='1')
        {
            aut=1;
            EEPROM.write(500,1);
            apaga();
        }
        Serial.println();
    }

}
if(myRTC.seconds==0 or myRTC.seconds==30)
{
    Serial.println();
    Serial.print(myRTC.hours, DEC); // Horas
    Serial.print(':');
    Serial.print(myRTC.minutes, DEC); // Minutos
    Serial.print(':');
    Serial.print(myRTC.seconds, DEC); // Segundos
    Serial.println();
    for(t=0;t<9;++t)
    {
        Serial.print(CEL[t]);
        CEL[t]=0;
    }
    for(t=0;t<45;++t)
    {
        Serial.print(smsaux[t]);
        smsaux[t]=0;
    }
    Serial.println();
    prueba=0;
}
}
}

```