



**UNIVERSIDAD NACIONAL
PEDRO RUIZ GALLO**



FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

**ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN TIMBRE INTELIGENTE
BASADO EN EL INTERNET DE LAS COSAS (IOT) PARA
FORTALECER LA SEGURIDAD CONTRA ROBOS EN
VIVIENDAS SOCIALES”**

TESIS

**PARA OPTAR EL TÍTULO DE
INGENIERO ELECTRÓNICO**

AUTORES:

**Bach. ARNOLD IVAN BELLASMIL FARROÑAN
Bach. JOSE LUIS ZUÑIGA PEÑA**

ASESOR:

Ing. OSCAR UCHELLY ROMERO CORTEZ

**LAMBAYEQUE – PERÚ
2018**



**UNIVERSIDAD NACIONAL
PEDRO RUIZ GALLO**



FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

**ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN TIMBRE INTELIGENTE
BASADO EN EL INTERNET DE LAS COSAS (IOT) PARA
FORTALECER LA SEGURIDAD CONTRA ROBOS EN
VIVIENDAS SOCIALES”**

TESIS

**PARA OPTAR EL TÍTULO DE
INGENIERO ELECTRÓNICO**

PRESENTADO POR:

Bach. ARNOLD IVAN BELLASMIL FARROÑAN
AUTOR

Bach. JOSE LUIS ZUÑIGA PEÑA
AUTOR

ASESORADO POR:

Ing. OSCAR UCHELLY ROMERO CORTEZ
ASESOR



**UNIVERSIDAD NACIONAL
PEDRO RUIZ GALLO**



FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

**ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN TIMBRE INTELIGENTE
BASADO EN EL INTERNET DE LAS COSAS (IOT) PARA
FORTALECER LA SEGURIDAD CONTRA ROBOS EN
VIVIENDAS SOCIALES”**

TESIS

**PARA OPTAR EL TÍTULO DE
INGENIERO ELECTRÓNICO**

APROBADO POR EL SIGUIENTE JURADO:

**Ing. MANUEL RAMÍREZ CASTRO
PRESIDENTE**

**Ing. CARLOS OBLITAS VERA
SECRETARIO**

**Ing. MARTIN NOMBERRA LOSSIO
VOCAL**

AGRADECIMIENTO

En primer lugar quiero agradecer a **Dios** por todas las bendiciones que nos ha dado a lo largo de nuestra carrera, por ser nuestra fortaleza en los momentos complicados.

Un agradecimiento muy especial a nuestros **padres** que siempre confiaron en nosotros, por todos los valores y enseñanzas que nos brindaron y por confiar en nosotros en todo momento.

Gracias a nuestro asesor de tesis el **Ing. OSCAR UCHELLY ROMERO CORTEZ** por el apoyo que nos ha brindado para la elaboración de nuestra tesis.

Gracias a nuestro jurado de tesis; el **Ing. MANUEL RAMÍREZ CASTRO, CARLOS OBLITAS VERA y MARTIN NOMBERRA LOSSIO**, por habernos brindado su apoyo para que este proyecto de tesis salga de la mejor manera.

Arnold Ivan Bellasmil Farroñan
Jose Luis Zuñiga Peña

DEDICATORIA

A:

Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mis **padres** por ser el pilar fundamental en todo lo que soy, en mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo. Todo este trabajo ha sido posible gracias a ellos.

Arnold Ivan Bellasmil Farroñan

A **Dios** por su infinita bondad, y por haber estado conmigo en los momentos que más lo necesitaba.

A mis abuelos, **Bertha** y **Gilberto** por el amor y educación que me han brindado, por su cariño y comprensión, por estar conmigo en todo momento.

A mis padres, **Cecilia** y **Pepe**, todo su apoyo fue esencial en mi formación, todos sus consejos y cariños brindados me han servido para seguir adelante.

A mis tíos, **Carmen** y **Beto**, por todo el apoyo y cariño, siempre brindándome las ganas e impulso para conseguir mis objetivos

A mi novia, **Stefany**, por su apoyo incondicional, por todo el ánimo, felicidad y amor que me brinda para poder alcanzar mis metas, su compañía ha sido demasiado importante en este proceso.

A todo el resto de mi familia y amistades que me brindaron mucho ánimo y aliento.

Todo este trabajo ha sido posible gracias a ellos.

Jose Luis Zuñiga Peña



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DECANATO

Ciudad Universitaria - Lambayeque



ACTA DE SUSTENTACIÓN N° 060-2018-D/FACFyM

(Sustentación Autorizada por Resolución N° 1296-2018-D/FACFyM)

En la ciudad de Lambayeque, siendo las 11:00 AM del día 09 de setiembre de 2018 se reunieron en sala de sustentación de Ingeniería Electrónica los miembros del Jurado designados mediante Resolución N° 597-2018-D/FACFyM, los docentes:

Ing. Manuel Javier Ramírez Castro Presidente
Ing. Carlos Leonardo Oblitas Vera Secretario
Mg. Ing. Martín Augusto Nombera Lossio Vocal

Para recibir la tesis titulada:

Diseño e Implementación de un timbre inteligente basado en el internet de las cosas (IoT) para fortalecer la seguridad contra Robos en viviendas sociales.

desarrollada por el Bachiller en Ingeniería Electrónica, **Bellasmil Farroñan Arnold Ivan** y el **Egresado** de la Escuela Profesional de Ingeniería Electrónica, **Zúñiga Peña José Luis**, (Art. 277° Reglamento General de la UNPRG)

Después de escuchar la exposición y las respuestas a las preguntas formuladas por los miembros del Jurado, se acordó APROBAR el trabajo por UNANIMIDAD con el calificativo de BUENO

En consecuencia, el Bachiller y el Egresado en referencia quedan aptos para recibir el Título Profesional de **Ingeniero Electrónico** de acuerdo a la Ley Universitaria, el Estatuto y Reglamento de la Universidad Nacional Pedro Ruiz Gallo de Lambayeque.

Observaciones:

Para constancia del hecho firman.

Ing. Carlos Leonardo Oblitas Vera
Secretario

Ing. Manuel Javier Ramírez Castro
Presidente

Mg. Ing. Martín Augusto Nombera Lossio
Vocal

CERTIFICO: Que, es copia fiel del original
Fecha: 23.11.2018

Dr. Armando José Moreno Heredia
SECRETARIO DOCENTE - FACFyM
VÁLIDO PARA TRÁMITES INTERNOS DE LA UNPRG

DECLARACIÓN JURADA DE ORIGINALIDAD

Yo, **Arnold Ivan Bellasmil Farroñan** y **Jose Luis Zuñiga Peña** investigadores principales y **Oscar Uchelly Romero Cortez** asesor del trabajo de investigación **“DISEÑO E IMPLEMENTACIÓN DE UN TIMBRE INTELIGENTE BASADO EN EL INTERNET DE LAS COSAS (IoT) PARA FORTALECER LA SEGURIDAD CONTRA ROBOS EN VIVIENDAS SOCIALES”** declaramos bajo juramento que este trabajo no ha sido plagiado, ni contiene datos falsos. En caso se demostrara lo contrario, asumo responsablemente la anulación de este informe y por ende el proceso administrativo a que hubiera lugar.

Lambayeque, 20 de noviembre del 2018

Investigadores:

Bach. Arnold Ivan Bellasmil Farroñan

Bach. Jose Luis Zuñiga Peña

Asesor:

Ing. Oscar Uchelly Romero Cortez

RESUMEN

El presente proyecto tiene como finalidad el Diseño e Implementación de un Timbre Inteligente basado en el Internet de las Cosas (IoT) para fortalecer la seguridad contra robos en viviendas sociales, como cualquier timbre estará presente un botón de llamado y el cable de señal hacia el timbre, para que este dispositivo este basado en IoT debe estar conectado a una red y a la internet, es aquí donde un microcontrolador o microprocesador, con interface de conexión a internet interviene.

Como parte de la seguridad a nuestra vivienda, y saber quién está llamando nuestra puerta, se contará con una cámara web de alta resolución; como intercomunicador se tendrá el soporte de un módulo GPRS/GSM, el cual no solo podrá recibir llamadas de parte nuestra, si no también poder enviar mensajes de texto de aviso cuando se toque el botón del timbre.

El diseño e implementación del timbre inteligente se llevó a cabo mediante el estudio de internet de las cosas y los diferentes dispositivos que se pueden conectar a está, utilizando la Raspberry pi modelo 3B+, el módulo sim800L, una cámara web, un convertidor de USB a TTL, el lenguaje de programación es Python, requiriendo para su implementación también otros dispositivos electrónicos tales como resistores, capacitores entre otros, para el diseño de una tarjeta adicional.

Se concluye; Diseño e Implementación de un Timbre Inteligente basado en el Internet de las Cosas (IoT) para fortalecer la seguridad contra robos en viviendas sociales, ayudará a reducir el porcentaje de robos en viviendas, permitiendo a los usuarios tener un monitoreo en tiempo real a través de un dispositivo móvil.

ABSTRACT

The purpose of this project is the Design and Implementation of an Intelligent Ring based on the Internet of Things (IoT) to strengthen security against thefts in social housing, as any bell will be present a call button and the signal cable to the timbre, so that this device is based on IoT must be connected to a network and the internet, this is where a microcontroller or microprocessor, with internet connection interface intervenes.

As part of the security of our home, and knowing who is calling our door, we will have a high resolution webcam; as an intercom, you will have the support of a GPRS / GSM module, which will not only be able to receive calls from us, but also to be able to send warning text messages when the bell button is touched.

The design and implementation of the smart timbre was carried out by studying the internet of things and the different devices that can be connected to it, using the Raspberry pi model 3B, the sim800L module, a webcam, a USB converter to TTL, the programming language is Python, requiring for its implementation also other electronic devices such as resistors, capacitors, among others, for the design of an additional card.

It concludes; Design and Implementation of an Intelligent Ring based on the Internet of Things (IoT) to strengthen security against theft in social housing, will help reduce the percentage of home thefts, allowing users to have real-time monitoring through of a mobile device.

ÍNDICE

AGRADECIMIENTO.....	3
DEDICATORIA... ..	4
RESUMEN... ..	8
ÍNDICE.....	10
ÍNDICE DE FIGURAS	13
ÍNDICE DE TABLAS	14
INTRODUCCIÓN.....	15
CAPÍTULO I: ASPECTOS DE LA INVESTIGACIÓN	18
1.1. Aspecto informativo.....	18
1.1.1. Título	18
1.1.2. Personal Investigador	18
1.1.3. Área de Investigación	18
1.1.4. Lugar de ejecución	18
1.1.5. Duración	18
1.2. Aspecto de la investigación	19
1.2.1. Situación problemática	19
1.2.2. Antecedentes bibliográficos	20
1.2.3. Formulación del problema científico.....	22
1.2.4. Objetivos.....	23
1.2.5. Justificación e importancia de la investigación	23
1.2.6. Hipótesis	23
CAPÍTULO II: MARCO TEÓRICO	24
2.1. Timbre	25
2.2. Tipos de Timbres	25
2.2.1. Dos Hilos	25
2.2.2. Inalámbrico.....	27
2.3. Raspberry Pi.....	29
2.3.1. Modelos	29
<input type="checkbox"/> Raspberry Pi 1 Modelo A	29
<input type="checkbox"/> Raspberry Pi 1 Modelo B y B+	30
<input type="checkbox"/> Raspberry Pi 2 Modelo B	31
<input type="checkbox"/> Raspberry Pi 3 Modelo B	32

□ Raspberry Pi 3 Modelo B+	33
2.4. Generación de Telefonía Móvil.....	33
2.4.1. 1G	34
2.4.2. 2G	35
2.4.3. 2.5G	35
2.4.4. 3G	36
2.4.5. 4G	37
2.4.6. 5G	38
2.5. Modelo OSI.....	40
2.5.1. Capa Física	40
2.5.2. Capa de Enlace de Datos	40
2.5.3. Capa de Red.....	41
2.5.4. Capa de Transporte.....	41
2.5.5. Capa de Sesión.....	41
2.5.6. Capa de Presentación.....	41
2.5.7. Capa de Aplicación.....	42
2.6. Streaming.....	42
2.6.1. Tipo de Streaming	44
2.6.2. Elementos Fundamentales para realizar Streaming.....	44
2.6.3. Enfoque de servidor.....	45
2.6.4. Enfoque de transporte.....	47
2.6.5. Enfoque de cliente	48
2.7. Herramienta para Transmitir Contenido Multimedia	49
2.7.1. FFmpeg.....	49
2.7.2. Icecast	50
2.8. Lenguajes de Programación	51
2.8.1. Python.....	51
2.8.2. Java	52
2.9. Software de Diseño de PCB.....	53
2.9.1. Eagle	53
2.9.2. Proteus ISIS	54
CAPÍTULO III: DISEÑO Y FUNCIONAMIENTO	56
3.1. Introducción	57

3.2. Descripción del Sistema.....	57
3.3. Lista de Materiales	57
3.3.1. Lista de materiales para la Raspberry Pi 3B+.....	58
3.3.2. Listado de Materiales para la tarjeta electrónica	59
3.3.3. Funcionamiento	59
3.4. Diseño de PCB	68
3.5. Armado de PCB	69
3.6. Conexiones.....	70
3.7. Especificación de la Raspberry Pi Modelo 3B	71
3.7.1. Características Técnicas	71
3.8. Especificaciones del módulo GSM	72
3.8.1. SIM800L.....	72
3.9. Especificación del conversor USB a TTL	73
3.9.1. CP2102	73
3.10. Especificaciones del SO Raspbian	75
3.11. Suministro Eléctrico	76
3.12. Diagrama de Flujo	77
3.13. Lista de Comandos AT utilizados	78
CAPÍTULO IV: EVALUACIÓN ECONÓMICA	79
4.1. Costos de Hardware y Software	80
4.2. Costos Totales.....	81
4.3. Justificación Económica	81
CONCLUSIONES	82
RECOMENDACIONES	83
REFERENCIA BIBLIOGRÁFICA	84
ANEXOS... ..	86

ÍNDICE DE FIGURAS

Figura 1: Partes del Timbre.....	26
Figura 2: Funcionamiento del Timbre.....	27
Figura 3: Vista interior de un Timbre Inalambrico	28
Figura 4: Funcionamiento del Timbre Inalambrico	29
Figura 5: Raspberry Pi 1 Modelo A	30
Figura 6: Raspberry Pi 1 Modelo B+	31
Figura 7: Raspberry Pi 2 Modelo B	32
Figura 8: Raspberry Pi 3 Modelo B	32
Figura 9: Raspberry Pi 3 Modelo B+	33
Figura 10: Evolucion de la Tecnologia Movil	34
Figura 11: Modelo de Referencia OSI	42
Figura 12: Streaming a través de Internet.	43
Figura 13: Código de Python en Raspberry Pi	51
Figura 14: Código de Java en Raspberry Pi.	53
Figura 15: Puertos y conexiones de la Raspberry pi 3	58
Figura 16: Ejemplo de script automático	67
Figura 17: Board de la tarjeta electrónica	68
Figura 18: Esquemático de la tarjeta electrónica	69
Figura 19: Conexiones	70
Figura 20: Vista Superior de la Raspberry Pi.....	71
Figura 21: Sim800L	72
Figura 22: Módulo CP2102.....	73
Figura 23: Escritorio Raspbian.....	76

ÍNDICE DE TABLAS

Tabla 1: Lista de complementos para la Raspberry pi.....	58
Tabla 2: Lista Materiales para la tarjeta	59
Tabla 3: Costos de Hardware	80
Tabla 4: Costos Totales.	81

INTRODUCCIÓN

El presente proyecto tiene como finalidad el diseño e implementación de un timbre inteligente basado en el internet de las cosas (IoT) para fortalecer la seguridad contra robos en viviendas sociales, como cualquier timbre estará presente su botón de llamado y el cable de señal hacia el timbre, para que este dispositivo esté basado en IoT debe estar conectado a una red y a la internet, es aquí donde la Raspberry Pi interviene como parte fundamental para controlar el aviso y la transmisión de video, por ello como parte de la seguridad de quién está llamando nuestra puerta el timbre inteligente contará con una cámara web de 12Mp para la visualización de quién o quienes quieren contactarse con nosotros, como intercomunicador se tendrá un módulo de llamadas, además dicho módulo también podrá enviar mensajes de texto.

Luego de un análisis de la situación, se lleva a cabo la selección para diseñar el timbre cumpliendo los requisitos pre establecido:

En el **Capítulo I: ASPECTOS DE LA INVESTIGACIÓN:** Se presentan los hechos preliminares de la tesis tales como la situación problemática, justificación e importancia de la investigación, objetivos, formulación de la hipótesis y el diseño metodológico.

En el **Capítulo II: MARCO TEÓRICO:** De acuerdo con la revisión bibliográfica, se define la base teórica que servirá para el desarrollo y sustento de la investigación, definiendo los modelos de las Raspberry Pi, las capas del modelo OSI, MQTT, Streaming, lenguajes de programación para la Raspberry Pi y software de diseño de PCB .

En el **Capítulo III: DISEÑO Y FUNCIONAMIENTO:** Se realiza el diseño y funcionamiento del timbre inteligente con la Raspberry Pi Modelo 3B+ utilizando el lenguaje de programación Python, también se seleccionara los diferentes dispositivos y módulos necesarios para el correcto funcionamiento del timbre inteligente.

En el **Capítulo IV: JUSTIFICACION ECONOMICA:** Mostraremos el balance general económico de todos los elementos y/o dispositivos que intervienen para poder realizar este proyecto de tesis.

CONCLUSIONES Y RECOMENDACIONES: Mostraremos las razones fundamentales por las que el desarrollo de nuestro proyecto de investigación es viable y correcto; a su vez, daremos a conocer algunas recomendaciones para su futura implementación. Finalmente se puede concluir que el trabajo de investigación permite aplicar por parte del alumno el correcto aprendizaje teórico y procedimientos prácticos; tales como, afianzar su conocimiento en el desarrollo del diseño y simulación de un sistema automático de control.

ASPECTOS DE LA INVESTIGACIÓN

CAPÍTULO I

CAPÍTULO I: ASPECTOS DE LA INVESTIGACIÓN

1.1. Aspecto informativo

1.1.1. Título

Diseño e implementación de un timbre inteligente basado en el internet de las cosas (IoT) para fortalecer la seguridad contra robos en viviendas sociales.

1.1.2. Personal Investigador

1.1.2.1. Autor

Nombre : Arnold Ivan Bellasmil Farroñan
Dirección : Mz. C Lt. 8 Int. B Fanny Abanto Calle,
Chiclayo
E – mail : ivan_bellasmil_29@hotmail.com
Teléfono : 949854650

1.1.2.2. Autor

Nombre : José Luis Zúñiga Peña
Dirección : Andrés Razuri #420 - Lambayeque
E – mail : Luchoz.2302@hotmail.com
Teléfono : 943658782

1.1.2.3. Asesor

Nombre : Ing. Oscar Uchelly Romero Cortez
E – mail : oromero@unprg.edu.pe
Teléfono : 941329254

1.1.3. Área de Investigación

Ingeniería Electrónica – Control y Automatización

1.1.4. Lugar de ejecución

Facultad de Ciencias Físicas y Matemáticas de la Universidad Nacional Pedro Ruíz Gallo.

1.1.5. Duración

04 meses

1.2. Aspecto de la investigación

1.2.1.Situación problemática

En el mundo existe cada vez más inseguridad, y todos tarde o temprano suelen verse envueltos en algún tipo de incidente de seguridad, sea en la calle o bien en nuestro hogar, cerca de nuestra familia.

De acuerdo con el reporte de las Estadísticas de Seguridad Ciudadana del INEI (Instituto Nacional de Estadística e Informática) solo en el primer semestre del 2015 (enero a junio), el 11.1% de viviendas a nivel nacional urbano fueron víctimas de algún robo o intento de robo.

Específicamente, en ciudades con más de 20 mil habitantes el promedio de viviendas afectadas es de 10.8%, mientras que en ciudades de 2 mil a 20 mil habitantes, el porcentaje es de 11.9%.

Según el artículo del diario gestión publicado el 30 de enero del 2016, en nuestro país, El 90% de robos a viviendas se lleva a cabo cuando no están sus dueños.

Gustavo Velasco, gerente general de la empresa de seguridad Verisure, Asegura que los ladrones eligen la madrugada porque hay pocas posibilidades de ser visualizados. Sin embargo, la incidencia es mayor entre 10:30 am y 12:00 pm, pues las casas quedan deshabitadas debido a que las familias salen a realizar actividades cotidianas.

En cualquiera de los casos, la principal desventaja es no contar con un sistema de monitoreo continuo de fácil acceso como en un dispositivo móvil, que alerte a los propietarios en caso de actividad sospechosa o de un intento de allanamiento, así mismo que cuente con la opción de dar aviso a las autoridades correspondientes.

1.2.2. Antecedentes bibliográficos

- **METAMODELO PARA LA INTEGRACIÓN DEL INTERNET DE LAS COSAS Y REDES SOCIALES**

José Ignacio Rodríguez Molano (2017)

ABSTRACT

El Internet de las cosas (IoT) está cambiando el mundo, y las aplicaciones desarrolladas en la industria son consideradas como la nueva revolución industrial generando nuevas oportunidades en la economía y la sociedad. La adopción del internet de las cosas al desarrollo de la industria 4.0 cambia los procesos y sistemas de fabricación planteando muchos desafíos. La información de estas aplicaciones está relacionada con el estado de la producción, el comportamiento de consumo de energía de máquinas, el movimiento de materiales, órdenes de clientes, datos de proveedores y en general, cualquier tipo de datos asociados a elementos inteligentes implementados en los procesos.

En este documento se hace una descripción de internet de las cosas y sus aplicaciones a la industria, los principios en los que se basa, los elementos y tecnologías disponibles para lograr la comunicación entre personas, objetos y aplicaciones que se han desarrollado en esta área.

Describe una propuesta de arquitectura de la internet de cosas aplicada a la industria, un metamodelo de integración (Internet de las Cosas, Redes Sociales, La Nube e Industria 4.0) para la generación de aplicaciones para la Industria 4.0 y un prototipo de monitoreo implementado a través del microordenador Raspberry Pi, un servidor de almacenamiento en la nube y un dispositivo móvil para el control de un proceso de producción en línea.

Con el aumento del IoT y el desarrollo de aplicaciones cloud computing, los problemas tecnológicos relacionados disminuyen y se encuentra una ampliación de servicios en un contexto industrial. En este artículo se

expone la necesidad de integrar el internet de las cosas, los sensores, los actuadores, las redes sociales y la computación en la nube que permitan realizar el internet de las cosas industrial (IoT).

La IoT requiere reconocer los objetos inteligentes y mantener un flujo de mensajes constante entre los diferentes objetos. No obstante, cada implementación de diferentes redes de sensores puede presentar diferentes problemas y cada aplicación proporciona una solución diferente. Aunque se han adelantado trabajos para una mejor interconexión en la IoT y una solución en este caso es realizar una arquitectura que soporte el paso de mensajes de los diferentes tipos de dispositivo y sea capaz de responderles. Una de las aproximaciones de este planteamiento es la arquitectura dirigida por modelos “Model-Driven Architecture” (MDA) la cual establece una serie de tecnologías a utilizar en la construcción de software bajo el esquema de la Ingeniería Dirigida por Modelos (MDE), con la cual se propone la creación de aplicaciones que permitan interconectar objetos heterogéneos, siendo estas la base en la generación de un Metamodelo que integre los elementos requeridos (sensores, Cloud y Web 2.0) para el avance de la IoT.

- **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD INALÁMBRICO CON TECNOLOGÍA BLUETOOTH PARA VIVIENDAS**

Fernando Wilfredo Ramírez Marocho (2012)

ABSTRACT

El incremento cuantitativo y cualitativo de la inseguridad y violencia en el Perú, expresada en diferentes formas y resaltando la que tiene un mayor índice de repetitividad, el hurto y robo a las viviendas, ha hecho que la población se encuentre en la búsqueda permanente de medios para protegerse de estas amenazas. Ante esta situación, la presente tesis busca una solución para resguardar el patrimonio de las personas y brindar seguridad a los inmuebles, es por ello que esta tesis lleva como título “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE

SEGURIDAD INALÁMBRICO CON TECNOLOGÍA BLUETOOTH PARA VIVIENDAS”.

- **IMPLEMENTACIÓN DE UN SISTEMA DE VIDEOVIGILANCIA UTILIZANDO UNA WEB CAM, ASTERISK, MOTION Y CHAN MOBILE**

Rodrigo Daniel Guayaquil Loor, José Luis Silva Palma

ABSTRACT

Actualmente en el Ecuador aún se encuentran muchas dificultades al momento de adquirir un Sistema de video vigilancia, ya sea por el alto costo de los equipos y/o por la complejidad del funcionamiento de los mismos, esto se debe en gran parte a que todo este tipo de soluciones de vigilancia se distribuyen bajo licencias.

El presente trabajo ofrece una alternativa a esto, al implementar un Sistema de video vigilancia con software de código abierto implementado totalmente bajo una plataforma de Linux.

El procedimiento utiliza las bondades de un servidor ASTERISK vinculado con un programa de control de cámaras web, llamado MOTION, y un módulo incorporado en Asterisk llamado CHAN-MOBILE, el cual permite establecer comunicación con dispositivos móviles utilizando celulares como puertas de salida hacia la red GSM a través de un enlace Bluetooth.

El sistema detecta movimiento a través de una cámara web, luego ejecuta un sonido de ALARMA y realiza una llamada a un teléfono CELULAR, donde comunica la presencia de algún intruso y brinda la opción de apagar el sonido de alarma a través del celular.

1.2.3. Formulación del problema científico

¿Cómo diseñar e Implementar un Timbre Inteligente basado en el Internet de las Cosas (IoT) para fortalecer la seguridad en viviendas sociales?

1.2.4. Objetivos

1.2.1.1 Objetivo general

- Diseño e Implementación de un Timbre Inteligente basado en el Internet de las Cosas (IoT) para fortalecer la seguridad contra robos en viviendas sociales.

1.2.1.2 Objetivos específicos

- Estudiar y clasificar los diferentes sistemas de monitoreo para seguridad contra robos en viviendas sociales.
- Diseñar la plataforma de interconexión entre timbre y teléfono celular.
- Seleccionar los equipos e instrumentos adecuados para la implementación del sistema.
- Realizar la implementación y pruebas de funcionamiento del sistema.

1.2.5. Justificación e importancia de la investigación

La tasa de robos a domicilio sigue incrementándose a pesar de las medidas tomadas por las autoridades y los propietarios. Esto promueve la búsqueda de alternativas de seguridad que se ajusten al estilo de vida actual, esto es en la mayoría de los casos, que todos los miembros del hogar estén fuera, por motivos laborales.

Los timbres inteligentes reducirían considerablemente el porcentaje de robos en domicilios, ya que el o los usuarios podrán monitorear de forma sencilla desde un dispositivo móvil cuando no se encuentren en el inmueble, ya sea por motivos de trabajo, que es lo más frecuente, o por motivos más particulares como un viaje familiar.

1.2.6. Hipótesis

Con el diseño e Implementación de un Timbre Inteligente basado en el Internet de las Cosas (IoT) fortaleceremos la seguridad contra robos en viviendas sociales.

MARCO TEÓRICO

CAPÍTULO II

CAPÍTULO: MARCO TEÓRICO

2.1. Timbre

Antes de que existiera el timbre, existía el llamador de la puerta. Un visitante levantaría un pomo de hierro o bronce en la puerta y golpearía una placa de metal para indicar su presencia. Más tarde, una pequeña campana fue instalada en el interior de la casa; el visitante tiraría de la cuerda para hacer sonar la campana.

Sin embargo, todo cambió durante 1831 cuando Joseph Henry diseñó el primer timbre eléctrico, modelo que se fue ajustando con el fin de cambiar su molesto sonido que en varias ocasiones se quedaba pegado por tiempo indefinido.

Más tarde, en la década de 1920 el modelo ya se había mejorado y los molestos sonidos pasaron a ser más amables. Ya en la década de 1930 los timbres ya habían causado sensación y se solía regalar en navidad modelos con diferentes características.

Cabe señalar que uno de los hechos más curiosos durante la masificación de este invento fue que algunas señoras consultaban al doctor aquejadas de crisis nerviosas producto de la contaminación auditiva. El trastorno llegó a denominarse como “door-bell-nerves”.(«El timbre» 2014)

2.2. Tipos de Timbres

En cuanto a su evolución de este objeto solo se ha valido de su tipo de tecnología y las variaciones de sonidos.

2.2.1. Dos Hilos

El primer timbre fue un timbre eléctrico fuerte. A principios de la década de 1930, los campaneos musicales con temas atractivos como "Westminster Quarters" llegaron a estar disponibles. Los timbres inalámbricos modernos utilizan ondas de radio para que suene el timbre. Los sordos utilizan dispositivos visuales como las luces para hacer destacar una alarma de timbre.

Un timbre eléctrico es un dispositivo capaz de producir una señal sonora al pulsar un interruptor. Su funcionamiento se basa en fenómenos electromagnéticos.

Consiste en un circuito eléctrico compuesto por un generador, un interruptor y un electroimán. La armadura del electroimán está unida a una pieza metálica llamada martillo, que puede golpear una campana pequeña.

- **Partes**

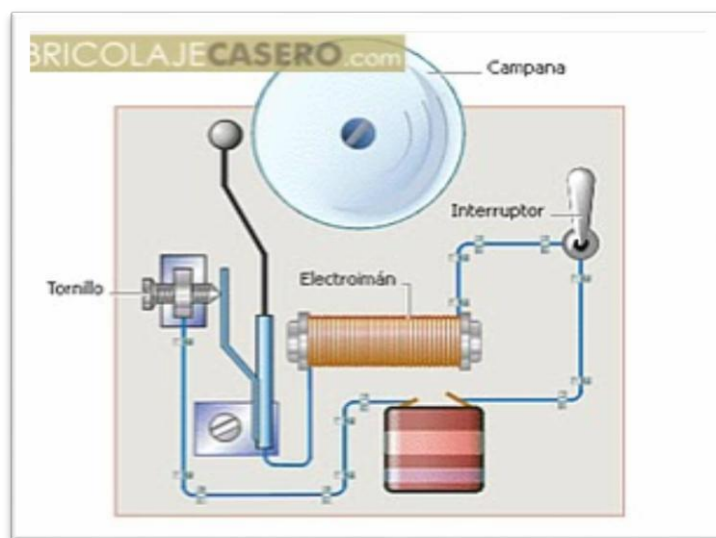


Figura 1: Partes del timbre

- **Funcionamiento**

Al cerrar el pulsador, la electricidad circula por el enrollamiento del electroimán y este crea un campo magnético en su núcleo y atrae la armadura. El martillo, soldado a la armadura, golpea la campana produciendo el sonido. Al abrir el interruptor cesan la corriente y el campo magnético del electroimán, y un resorte devuelve la armadura a su posición original para interrumpir el sonido.

Para conseguir que el martillo golpee la campana repetidamente mientras el interruptor esté cerrado, y no una sola vez, se sitúa un contacto eléctrico en la armadura que actúa como un interruptor.

Así, cuando la armadura es atraída por el electroimán, se interrumpe el contacto, cesa la corriente en el electroimán y la armadura retrocede a su posición original.

Allí vuelve a establecerse el contacto eléctrico, con lo que el electroimán vuelve a atraer a la armadura, y así sucesivamente.

Modernamente, muchos timbres no tienen interruptor, basándose en golpear la campana al doble de la frecuencia de la red. Tienen la ventaja de ser más fiables y más duraderos, ya que no se ensucian ni se desgastan los contactos del interruptor. Algunos no tienen ni campana, bastando la vibración de los contactos transmitida a la caja del timbre. A veces se llama zumbadores a estos timbres sin campana, porque el sonido que producen es un zumbido. («Timbre eléctrico» 2018)

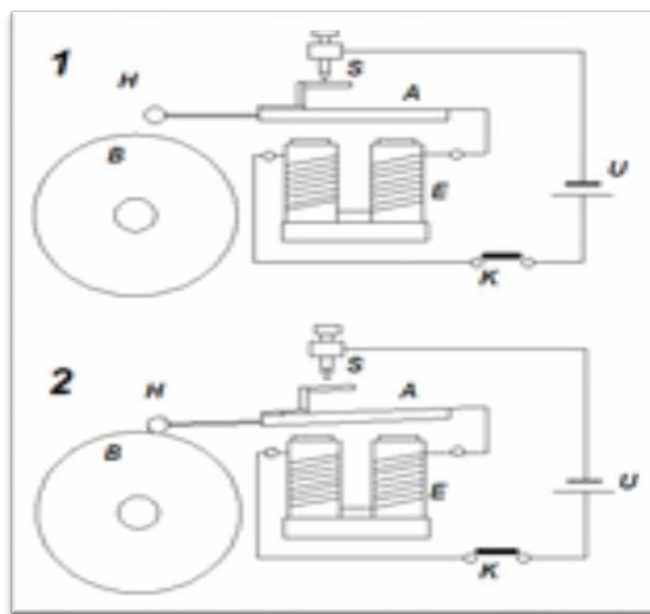


Figura 2: Funcionamiento del timbre

2.2.2. Inalámbrico

Los timbres inalámbricos son montados en superficie y no requieren cableado, por lo que son una opción excelente para las casas de apartamentos con múltiples unidades, o en los casos en que los cables de una puerta vieja no funcionan y el propietario no tiene los medios o la habilidad para reparar

el cableado. Los timbres inalámbricos pueden ser instalados en lugar de un timbre de edad con cable, o en un espacio donde no se ha instalado timbre de la puerta en primer lugar.



Figura 3: Vista interior de un timbre inalámbrico

- **Funcionamiento**

Cada botón de timbre inalámbrico contiene un transmisor que emite una frecuencia reconocida por un receptor situado dentro de la caja del timbre. Cuando se pulsa el botón, suena la campana. La mayoría de los modelos cuentan con varios tipos diferentes de timbres, y varias opciones de frecuencia. Si notas que tu timbre está sonando por ninguna razón aparente y tienes un vecino cercano, puede que tengas que cambiar la frecuencia. El receptor puede recoger un envío de una caja idéntica cercana. («Preble» 2018)

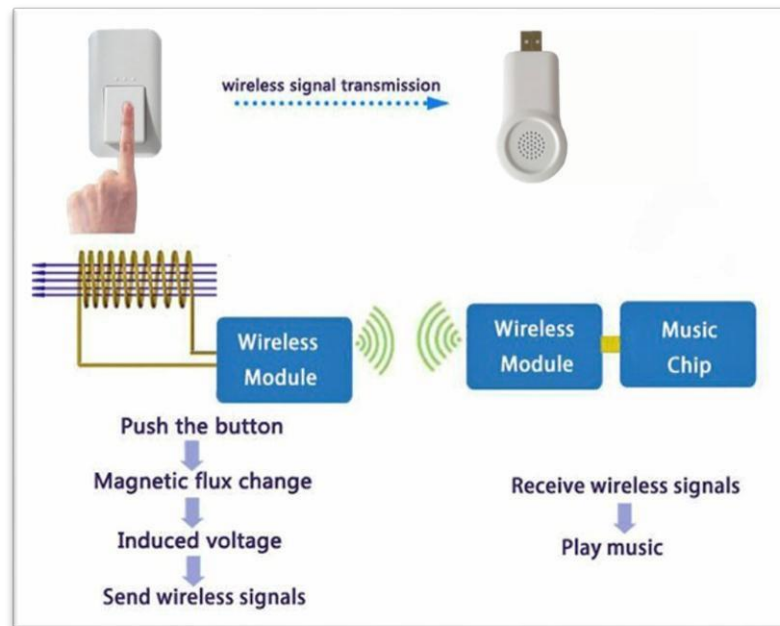


Figura 4: Funcionamiento del Timbre Inalámbrico

2.3. Raspberry Pi

2.3.1. Modelos

Los esquemas del modelo A y el modelo B fueron lanzados el 20 de abril del 2012 por la fundación.

- **Raspberry Pi 1 Modelo A**

Este fue el primer modelo de Raspberry, sus ventas comenzaron en el año 2012. Carecía de puerto Ethernet, por lo que para su conexión a Internet requería de un adaptador Wi-Fi por USB. Poseía 26 conectores GPIO, salida de vídeo vía HDMI y Video RCA, un conector Jack de 3.5 milímetros, un único conector USB, Micro USB (De alimentación) y un conector de cámara. Su procesador fue un Broadcom BCM2835, Single-Core a 700MHz. También tuvo 256 MB de RAM y una gráfica Broadcom VideoCore IV. Requería de una fuente de alimentación de 5 voltios y 2 amperios, elemento común al resto de versiones. Tuvo un coste inicial de 40 euros. («Zúñiga» 2015)



Figura 5: Raspberry Pi Modelo A

- **Raspberry Pi 1 Modelo B y B+**

También del año 2012, es una variante del Modelo A, trajo consigo diversas mejoras, la inclusión del doble de memoria RAM, pasando de 256MB a 512MB. Trajo consigo un puerto USB más y, por fin, un conector Ethernet (RJ-45), se mantuvo tanto su tamaño como su coste. No hubo variaciones ni en el procesador ni en la parte gráfica. Tiempo después se lanzó el Modelo B+, que incluyó 4 puertos USB y pasó de usar una SD a una MicroSD. («Zúñiga» 2015)

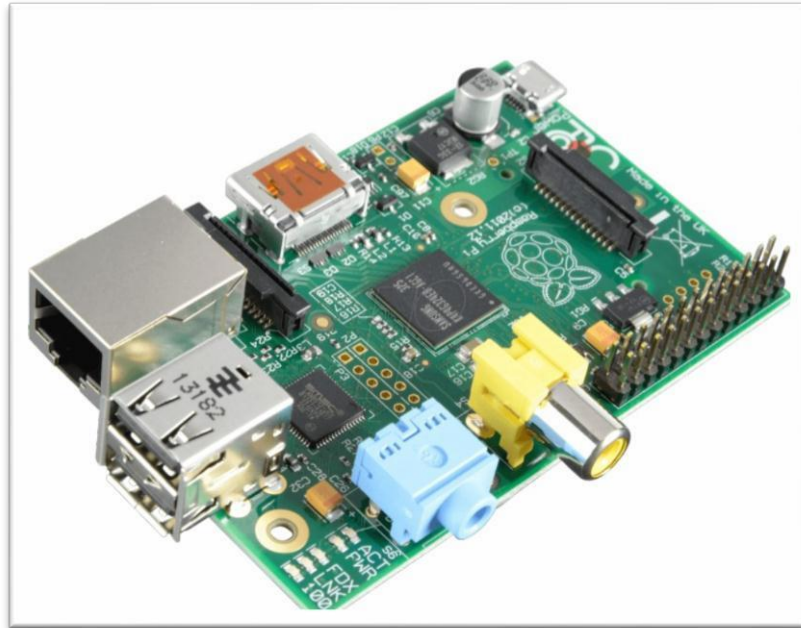


Figura 6: Raspberry Pi 1 Modelo B+

- **Raspberry Pi 2 Modelo B**

Lanzada en 2014 es el primer modelo que no incluye el mismo procesador usado en los tres anteriores: se sustituye por uno de la misma marca, pero de modelo BCM2836. Pasa de ser de un núcleo a cuatro, y de 700MHz a 900MHz. No obstante emplea la misma gráfica, la VideoCore IV. Dobra la cantidad de memoria RAM, pasando de 512MB a 1GB (Algo menos en realidad) esta memoria está compartida con la gráfica. También incluye 40 pines GPIO, y mantiene los cuatro puertos USB. Suprime la conexión RCA. («Zúñiga» 2015)

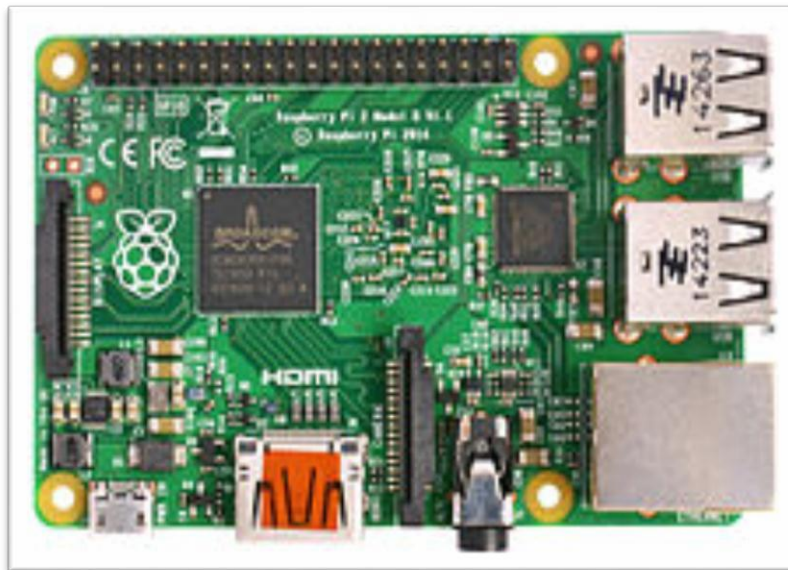


Figura 7: Raspberry Pi 2 Modelo B

- **Raspberry Pi 3 Modelo B**

Sacada a la luz en el año 2016, renueva procesador, una vez más de la compañía Broadcom, una vez más un Quad-Core, pero pasa de 900MHz a 1.20GHz. Mantiene la RAM en 1GB. Su mayor novedad fue la inclusión de Wi-Fi y Bluetooth (4.1 Low Energy) sin necesidad de adaptadores. («Zúñiga» 2018)



Figura 8: Raspberry Pi 3 Modelo B

- **Raspberry Pi 3 Modelo B+**

La Raspberry Pi 3 B+ apareció en marzo del 2018 para actualizar el modelo anterior la Raspberry Pi 3 Model B y entre sus mejoras cuenta con un nuevo procesador y mejor conectividad, así que pasa de tener 1.2Ghz a tener 1.4Ghz y en cuanto a la conectividad inalámbrica ahora incorpora doble banda a 2,4GHz y 5GHz, y su nuevo puerto Ethernet se triplica, pasa de 100 Mb/s en el modelo anterior a 300 Mb/s en el nuevo modelo, también cuenta con Bluetooth 4.2. («Zúñiga» 2018)



Figura 9: Raspberry Pi 3 Modelo B+

2.4. Generación de Telefonía Móvil

Cuando vemos la evolución de la red de comunicación móvil, la G de generación inalámbrica móvil generalmente nos indica que ha habido un cambio en la naturaleza del sistema, la velocidad, la tecnología y la frecuencia. Cada generación tiene algunos estándares, capacidades técnicas y nuevas características que la diferencian de la anterior.

(«Evolución de la red de comunicación móvil, del 1G al 5G | VIU» [sin fecha])

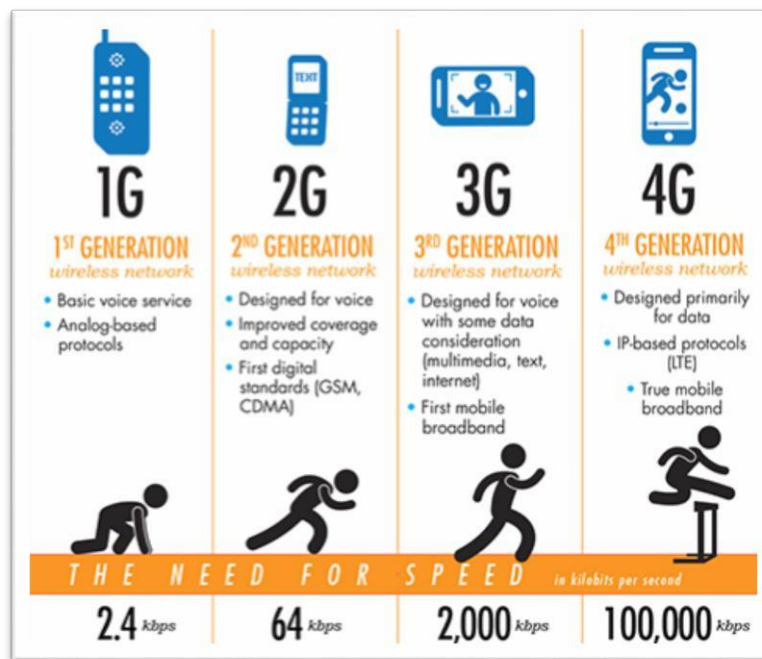


Figura 10: Evolución de la Tecnología Móvil

2411 1G

La 1ª red de comunicación móvil automatizada comercial fue lanzada por NTT en Japón en 1979, seguida por el lanzamiento del sistema de Telefonía Móvil Nórdica (NMT) en Dinamarca, Finlandia, Noruega y Suecia, en 1981.

- Año 1970 - 1980
- Estándares - AMPS (Advanced Mobile Phone System).
- Servicios - Sólo voz
- Tecnología - analógica
- Velocidad - 1kbps a 2,4 kbps
- Multiplexación - FDMA
- Conmutación - conmutación de circuitos
- Core Network - PSTN
- Frecuencia - 800- 900 MHz
- Ancho de banda de RF - 30 kHz. La banda tiene capacidad para 832 canales dúplex, entre los cuales 21 están reservadas para el establecimiento de llamada, y el resto para la comunicación de voz
- Malas comunicaciones de voz y ninguna seguridad ya que las llamadas de voz se reproducen en las torres de radio.

La tecnología GSM fue la primera en facilitar voz y datos digitales, así como roaming internacional permitiendo al cliente ir de un lugar a otro.

- Año - 1980 -1990
- Tecnologías - Digital
- Velocidad - 14kbps a 64 Kbps
- Banda de frecuencia - 850 - 1900 MHz (GSM) y 825 - 849 MHz (CDMA)
- Ancho de banda / canal - GSM divide cada canal de 200 kHz en bloques de 25 kHz El canal CDMA es nominalmente de 1,23 MHz
- Multiplexación / Tecnología de acceso - TDMA y CDMA.
- Conmutación - Conmutación de circuitos
- Estándares - GSM (Sistema Global para Comunicaciones Móviles), IS-95 (CDMA) - utilizado en América y partes de Asia), JDC (Celular Digital Japonés) (basado en TDMA), utilizado en Japón, iDEN (basado en TDMA), red de comunicación propietaria utilizado por Nextel en los Estados Unidos.
- Servicios: Voz Digital, SMS, roaming internacional, conferencia, llamada en espera, retención de llamada, transferencia de llamadas, bloqueo de llamadas, número de identificación de llamadas, grupos cerrados de usuarios (CUG), servicios USSD, autenticación, facturación basada en los servicios prestados a sus clientes, por ejemplo, cargos basados en llamadas locales, llamadas de larga distancia, llamadas con descuento, en tiempo real de facturación.

Introducción de la red de paquetes para proporcionar transferencia e internet de alta velocidad de datos.

- Año - 2000- 2003
- Estándares - Servicio General de Paquetes de Radio (GPRS) y EDGE (Velocidades de datos mejoradas en GSM)
- Frecuencia: 850 -1900 MHz

- Velocidad - 115kpbs (GPRS) / 384 kbps (EDGE)
- Conmutación - Conmutación de paquetes para la transferencia de datos
- Multiplexación - desplazamiento mínimo gaussiano keying-GMSK (GPRS) y EDGE (8-PSK)
- Servicios - pulsar para hablar, multimedia, información basada en la web de entretenimiento, soporte WAP, MMS, SMS juegos móviles, búsqueda y directorio, acceso a correo electrónico, videoconferencia.

2443 3G

El objetivo de los sistemas 3G fue ofrecer aumento de las tasas de datos, facilitar el crecimiento, mayor capacidad de voz y datos, soporte a diversas aplicaciones y alta transmisión de datos a bajo coste. Los datos se envían a través de la tecnología de una tecnología llamada Packet Switching. Las llamadas de voz se traducen mediante conmutación de circuitos.

- Año - 2000
- Estándares:
 - UMTS (WCDMA) basado en GSM (Global Systems for Mobile) infraestructura del sistema 2G, estandarizado por el 3GPP.
 - CDMA 2000 basado en la tecnología CDMA (IS-95) estándar 2G, estandarizada por 3GPP2.
 - interfaz de radio TD-SCDMA que se comercializó en 2009 y sólo se ofrece en China
- Velocidad: 384KBPS 2Mbps
- Frecuencia: aproximadamente 8 a 2,5 GHz
- Ancho de banda: de 5 a 20 MHz
- Tecnologías de multiplexación y acceso
- Interfaz de radio llamada WCDMA (Wideband Code División Multiple Access)
- HSPA es un actualización de W-CDMA que ofrece velocidades de 14,4 Mbit / s de bajada y 5,76 Mbit / s de subida.

- HSPA + puede proporcionar velocidades de datos pico teóricas de hasta 168 Mbit / s de bajada y 22 Mbit / s de subida.
- CDMA2000 1X: Puede soportar tanto servicios de voz como de datos. La máxima velocidad de datos puede llegar a 153 kbps
- Servicios - telefonía móvil de voz, acceso a Internet de alta velocidad, acceso fijo inalámbrico a Internet, llamadas de video, chat y conferencias, televisión móvil, vídeo a la carta, servicios basados en la localización, telemedicina, navegación por Internet, correo electrónico, buscapersonas, fax y mapas de navegación, juegos, música móvil, servicios multimedia, como fotos digitales y películas. servicios localizados para acceder a las actualizaciones de tráfico y clima, servicios móviles de oficina, como la banca virtual.

2.4.5. 4G

El sistema móvil de cuarta generación está basado totalmente en IP. El objetivo principal de la tecnología 4G es proporcionar alta velocidad, alta calidad, alta capacidad, seguridad y servicios de bajo coste para servicios de voz y datos, multimedia e internet a través de IP. Para usar la red de comunicación móvil 4G, los terminales de los usuarios deben ser capaces de seleccionar el sistema inalámbrico de destino. Para proporcionar servicios inalámbricos en cualquier momento y en cualquier lugar, la movilidad del terminal es un factor clave en 4G.

- Inicio - años de 2010. En 2008, la UIT-R especifica los requisitos para los sistemas 4G
- Estándares - Long-Term Evolution Time-Division Dúplex (LTE-TDD y LTE-FDD) estándar WiMAX móvil (802.16m estandarizado por el IEEE)
- Velocidad - 100 Mbps en movimiento y 1 Gbps cuando se permanece inmóvil.
- Telefonía IP
- Nuevas frecuencias, ancho de banda de canal de frecuencia más amplia.

- Tecnologías de multiplexación / acceso - OFDM, MC-CDMA, CDMA y LAS-Red-LMDS
- Ancho de Banda - 5-20 MHz, opcionalmente hasta 40 MHz
- Bandas de frecuencia: - LTE cubre una gama de diferentes bandas. En América del Norte se utilizan 700, 750, 800, 850, 1900, 1700/2100 (AWS), 2300 (WCS) 2500 y 2600 MHz (bandas 2, 4, 5, 7, 12, 13, 17, 25, 26, 30, 41); 2500 MHz en América del Sur; 700, 800, 900, 1800, 2600 MHz en Europa (bandas 3, 7, 20); 800, 1800 y 2600 MHz en Asia (bandas 1, 3, 5, 7, 8, 11, 13, 40) 1800 MHz y 2300 MHz en Australia y Nueva Zelanda (bandas 3, 40).
- Servicios - acceso móvil web, telefonía IP, servicios de juegos, TV móvil de alta definición, videoconferencia, televisión 3D, computación en la nube, gestión de flujos múltiples de difusión y movimientos rápidos de teléfonos móviles, Digital Video Broadcasting (DVB), acceso a información dinámica, dispositivos portátiles.

2465 5G

La capa física y de enlace de datos define la tecnología inalámbrica 5G indicando que es una tecnología Open Wireless Architecture (OWA).

Para realizar esto, la capa de red está subdividida en dos capas; capa de red superior para el terminal móvil y un menor nivel de red para la interfaz. Aquí todo el enrutamiento se basa en direcciones IP que serían diferentes en cada red IP en todo el mundo.

En la tecnología 5G la pérdida de velocidad de bits se supera mediante el Protocolo de Transporte Abierta (OTP). El OTP es soportado por transporte y capa de sesión. La capa de aplicación es para la calidad de la gestión de servicio a través de varios tipos de redes. 5G adelanta un verdadero mundo inalámbrico Wireless-World Wide Web (WWWW).

- Inicio año - 2015
- Velocidad - 1 a 10 Gbps.
- Ancho de Banda - 1.000x ancho de banda por unidad de superficie.
- Frecuencia - 3 a 300 GHz

- Tecnologías de multiplexación / Access - CDMA y BDMA
- Estándares - banda ancha IP LAN / W AN / PAN & WWW
- Características: rendimiento de tiempo real - de respuesta rápida, de baja fluctuación, latencia y retardo
- Muy alta velocidad de banda ancha - velocidades de datos Gigabit, cobertura de alta calidad, multi espectro
- Infraestructura virtualizada - Software de red definido, sistema de costes escalable y bajo.
- Soporta Internet de las Cosas y M2M - 100 veces más dispositivos conectados, Cobertura en interiores y eficiencia de señalización
- Reducción de alrededor del 90% en el consumo de energía a la red.
- Su tecnología de radio facilitará versión diferente de las tecnologías de radio para compartir el mismo espectro de manera eficiente.
- Servicios: - Algunas de las aplicaciones son importantes - personas y dispositivos conectados en cualquier lugar en cualquier momento. Su aplicación hará que el mundo real sea una zona Wi Fi. Dirección IP para móviles asignada de acuerdo con la red conectada y la posición geográfica. Señal de radio también a mayor altitud. Múltiples servicios paralelos, con los que se puede saber el tiempo meteorológico y en tu posición geográfica mientras hablas. La educación será más fácil. Un estudiante que se sienta en cualquier parte del mundo puede asistir a la clase. El diagnóstico remoto es una gran característica de 5G. Un Médico puede tratar al paciente situado en la parte remota del mundo. El seguimiento será más fácil, una organización gubernamental y otros investigadores pueden monitorear cualquier parte del mundo. Se hace posible reducir la tasa de criminalidad. La visualización del universo, galaxias y planetas serán posibles. Posible también detectar más rápidamente desastres naturales incluyendo tsunamis, terremotos, etc. («Evolución de la red de comunicación móvil, del 1G al 5G | VIU» [sin fecha])

2.5. Modelo OSI

Este modelo se encarga de subdividir los sistemas de comunicaciones en partes más pequeñas, a las que se les denomina capas. Las funciones que son similares, se agrupan en capas lógicas; cada una provee servicios hacia la capa superior, mientras recibe servicios de la capa de un nivel inferior. El modelo OSI, dicho en otras palabras, es un estándar para la definición de arquitecturas de interconexión de sistemas de comunicaciones; describe desde el inicio de transmisión de datos (bits), hasta llegar a las aplicaciones que se utilizan cotidianamente (correo, navegador de Internet, etc.). El modelo OSI consta de 7 capas, las cuales se describirán a continuación y que ayudará a comprender de mejor forma cómo se transmite la información a través de una red, o bien, a través de la red de redes: Internet. (Arrecis, s. f.)

2.5.1. Capa Física

Esta capa define todo lo que se refiera a componentes eléctricos y físicos. En particular define la relación entre un dispositivo y el medio de transmisión, por ejemplo: un cable UTP, transmisión inalámbrica, RS-232, etc. La unidad de datos que se maneja en ésta capa es el bit. (Arrecis, s. f.)

2.5.2. Capa de Enlace de Datos

Esta capa es la encargada del enrutamiento a nivel físico, a nivel de topología de red. Para ello hace uso de la dirección MAC, la cual provee acceso al medio; ésta dirección es única en cada dispositivo de red, ninguna se repite. La tarea principal de ésta capa es tomar la información tal cual la envía la capa física y transformarla en una abstracción que no contenga errores, para luego enviarla a la capa de red; cada una de éstas abstracciones de datos se denomina trama, nombre que recibe la unidad de datos de ésta capa. (Arrecis, s. f.)

2.5.3. Capa de Red

Esta capa es la encargada del enrutamiento entre una o varias redes. En ésta capa la unidad de datos que se maneja es denominada: paquete. La capa de red realiza las funciones de enrutamiento, así también, unifica los fragmentos, y reporta errores de envío. Los enrutadores operan en ésta capa, enviando información a través de la red, lo cual hace posible la Internet.

Es decir, el objetivo de la capa 3 es hacer que los paquetes lleguen desde el origen al destino, incluso si éstos no están conectados directamente. (Arrecis, s. f.)

2.5.4. Capa de Transporte

Esta capa provee una transferencia de datos transparente entre usuarios y de servicios de transferencia confiables hacia las capas superiores. A la unidad de datos de la capa 4 se le denomina segmento o datagrama, esto dependiendo si corresponde al protocolo TCP o a UDP. El protocolo TCP está orientado a la conexión, mientras que UDP no lo está. Ambos trabajan con puertos lógicos para la comunicación, y al unir la capa de transporte con la capa de red se obtienen los sockets, que no es más que la unión de una dirección IP y un puerto. (Arrecis, s. f.)

2.5.5. Capa de Sesión

Esta capa se encarga de controlar las conexiones entre computadoras. Establece, maneja y termina las conexiones entre la aplicación local y remota. Permite cifrar los datos y también comprimirlos, por lo tanto, puede decirse que actúa también, como un traductor. (Arrecis, s. f.)

2.5.6. Capa de Presentación

Esta capa provee independencia de representación de datos a través de la traducción entre formatos de aplicación y de red. Esta capa transforma datos de manera que la aplicación los acepte. Es la primera capa que se acerca más

al contenido de la comunicación en lugar de ver cómo se establece ésta.
(Arrecis, s. f.)

2.5.7. Capa de Aplicación

Esta capa es la más cercana al usuario final, lo que significa que la capa de aplicación y el usuario interactúan directamente con las aplicaciones de software que utilizan algún componente para comunicación. Por ejemplo: correo electrónico, servidores de ficheros, gestores de bases de datos, entre otros. (Arrecis, s. f.)

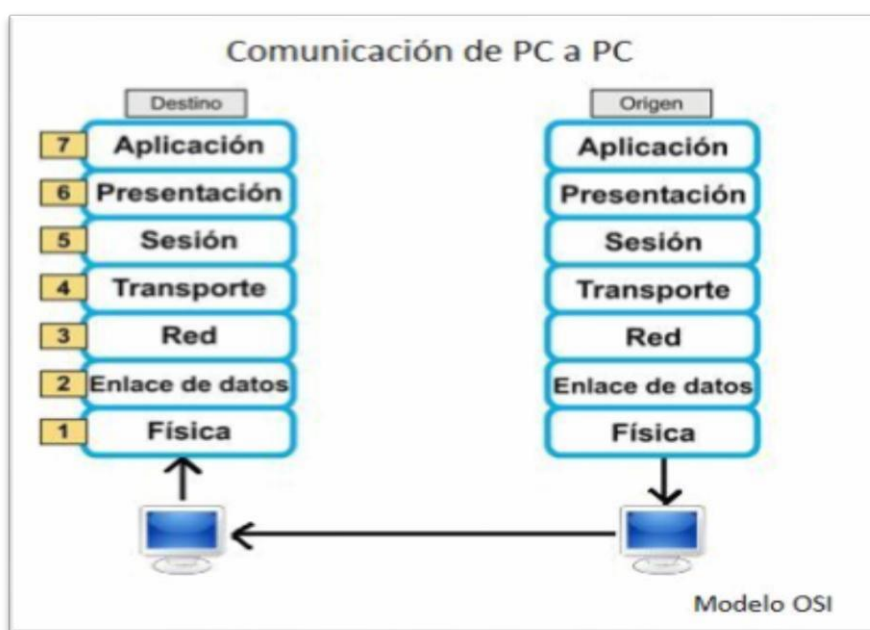


Figura 11: Modelo de referencia OSI

2.6. Streaming

Se conoce como Streaming a la transmisión continua e ininterrumpida de audio y/o vídeo a través de Internet, donde, esta se transmite desde una máquina que funge como servidor, y hay una o varias máquinas que actúan como clientes; es decir, José por mencionar algún nombre, puede ser el servidor y realizar Streaming de vídeo a través de Internet, mientras que en cualquier lugar del mundo con acceso a Internet pueden estar uno o varios amigos de José recibiendo el Stream de vídeo que les está enviando desde su computadora. Se puede observar que se mencionan dos conceptos nuevos,

Streaming y Stream; se puede definir como Stream a la información que se envía del origen a uno o varios destinos; mientras que Streaming sería la acción de transmitir esa información.



Figura 12: Streaming a través de Internet

El Streaming nació en 1995 con el lanzamiento de real audio, una aplicación para transmisión de audio en tiempo real. Antes de que apareciera el Streaming, si alguien quería escuchar una canción o ver un vídeo, era estrictamente necesario descargar un archivo completo para luego poder reproducirlo en la computadora, mientras que con la aparición del Streaming todo esto quedó atrás, ahora es posible reproducir contenido multimedia desde Internet, sin necesidad de que éste descargue por completo, es decir, se puede reproducir el Stream conforme éste está siendo transmitido hacia la computadora.

Una analogía que define de una forma fácil el concepto de Streaming, y que vale la pena mencionar es la siguiente: se tiene una botella con leche, un vaso y muchas ganas de beber la leche. Sin Streaming (descarga): se debe tomar la botella de leche y servirla en el vaso hasta que éste se llene; una vez lleno puede empezar a consumirse.

Con Streaming: se toma la botella de leche, pero en lugar de llenar el vaso completamente, éste se hace a un lado y se consume la leche directamente desde la caja. (Arrecis, s. f.).

2.6.1. Tipo de Streaming

Existen dos tipos de Streaming, por demanda y en directo, a continuación se describe cada uno de ellos y cómo funciona:

- **Por demanda**

Éste se refiere a la transmisión de contenido multimedia almacenado en un servidor y que puede ser visualizada en cualquier momento. El mejor ejemplo de este tipo es de nuevo YouTube, ya que en él se pueden encontrar millones de vídeos, que han sido subidos por gente de todo el mundo, los cuales se pueden visualizar cuando es requerido sin importar la hora.

- **En directo**

A diferencia del Streaming por demanda que puede ser visto en cualquier momento, el Streaming en directo puede ser consumido únicamente en el momento en que se está realizando la transmisión; viene siendo similar a la televisión convencional, donde un televisor sintoniza señales que están siendo transmitidas en tiempo real, no importando si son pregrabados o no, la señal está siendo transmitida en ese momento, de tal forma que al encender el televisor será sintonizado lo que la televisora esté enviando en ese preciso instante. (Arrecis, s. f.).

2.6.2. Elementos Fundamentales para realizar Streaming

Para que todo esto funcione son fundamentales tres partes, comenzando desde dónde se transmite la información, es decir, el servidor; a quién se le enviará la información, es decir, los clientes; y cómo transmitir la información. A continuación, se describe brevemente cada una de estas tres partes:

- **Servidor**

Este es el encargado de distribuir la información a partir de una fuente, a través de él se genera datagramas que se envían a través de la red. Al hablar de servidor se hace referencia tanto a hardware, como al posible software que éste incluya.

- **Cliente**

Es cada una de las máquinas receptoras de la información que se transmite, como mínimo se necesita tener un cliente, sino no tendría sentido la transmisión.

- **Medio**

Es a través de dónde se enviará la información, grosso modo puede decirse que es la Internet, pero si se profundiza un poco más y se llega al nivel en donde esto se realiza, se estaría hablando de la capa 4 del modelo OSI, es decir, la capa de transporte; para realizar Streaming se utilizan datagramas (UDP), que son paquetes que se envían sin esperar confirmación de entrega al destinatario, esto permite que la transmisión sea más rápida y fluida. (Arrecis, s. f.).

2.6.3. Enfoque de servidor

No es estrictamente necesario contar con un servidor especial para colocar contenido multimedia, ya que un servidor normal puede enviar información, siendo el cliente el que se encargue de procesarla y mostrarla a medida que está siendo recibida.

Sin embargo, para montar un verdadero servidor de Streaming es ideal el uso de herramientas de software que provean de funcionalidades y optimizaciones para la realización del objetivo, pudiendo con ellas modificar la calidad del archivo que se envía, para así optimizar el ancho de banda del que se dispone.

En el caso del Streaming en directo, es imprescindible el uso de estas herramientas de software, la cual al momento de recibir la información del emisor, la enviará a través del servidor hacia los clientes que estén sintonizando en ese momento. En este capítulo se mencionarán las herramientas más conocidas en el medio, sin entrar en mayor detalle sobre ellas, ya que serán profundizadas en el siguiente capítulo.

- **Flash media server**

Es un sistema propiedad de Adobe Systems. Este servidor funciona como un distribuidor, se basa en aplicaciones flash para conectar los clientes al distribuidor mediante el protocolo de envío de mensajes en tiempo real, RTMP.

- **Wowza media server**

Este sistema fue desarrollado por Wowza Media Systems. Es utilizado para realizar Streaming de audio o vídeo, por demanda o en directo. Puede ser utilizado sobre varios sistemas operativos, ya que es una aplicación java.

- **Red5 media server**

Es un sistema para la distribución de Streaming que provee funcionalidad de multi-usuario basado en tecnología de Adobe Flash Player.

Está basado en Java, red5 puede ser utilizado en empresas de cualquier ámbito, sin importar su tamaño. La principal característica de éste es que es código abierto, por esto y por el poder que brinda, será el servidor que se utilizará para el desarrollo de la plataforma. (Arrecis, s. f.).

2.6.4. Enfoque de transporte

Para realizar Streaming a través de Internet es fundamental la velocidad de transmisión, ya que es necesario que la tasa de llegada de datos sea superior a la tasa de visualización que mantiene el cliente.

Es por ello que para este tipo de transmisiones se utilizan protocolos ligeros, siendo éstos: UDP y RTSP, los cuales se desarrollan en la capa 3 del modelo OSI.

- **User Datagram Protocol (UDP)**

En inglés, User Datagram Protocol, es uno de los miembros núcleo de los protocolos de Internet. Con UDP las aplicaciones de computadora pueden enviar mensajes a través de Internet, sin necesidad de establecer canales especiales de transmisión o rutas de envío. Este protocolo, a diferencia del conocido TCP, no está orientado a la conexión, por lo que, a éste no le interesa saber si los datagramas (paquetes) han sido recibidos por los clientes de forma correcta o no. De ésta forma se logra una transmisión fluida de información a través de Internet, ya que no se pierde tiempo verificando si la información está llegando de forma correcta. Lograr una transmisión fluida ayuda a que el Streaming se produzca de forma continua y sin interrupciones, algo que es indispensable.

- **Real time streaming protocol (RTSP)**

Por sus siglas en inglés, Real Time Streaming Protocol. Es un protocolo no orientado a la conexión, el servidor mantiene sesiones asociadas a un cliente por medio de un identificador. RTSP es un protocolo que puede usar tanto UDP como TCP, dependiendo de la situación; utiliza TCP para los datos de control del reproductor, y utiliza UDP para la transmisión de audio y vídeo, que como se explicó en UDP, esto contribuye a una transmisión más fluida. Este protocolo es el que utiliza YouTube. (Arrecis, s. f.).

2.6.5. Enfoque de cliente

No serviría de mucho tener un servidor funcionando en perfectas condiciones, ni tampoco la optimización en el transporte de la información, si del lado del cliente no se tienen correctamente configuradas las herramientas necesarias para una reproducción óptima del Stream de datos que se envía, es por ello que se deben de tener claros algunos conceptos que serán manejados del lado del cliente.

- **Precarga de datos**

Cuando se transmite la información desde el servidor hacia los clientes, ésta está sujeta a demoras o retrasos, esto ocurre cuando los datos se escasean, ya sea por interrupciones en la conexión, o bien por el poco ancho de banda de cualquiera de las partes. Es por ello que los reproductores multimedia utilizan los conocidos buffer, un buffer funciona como una memoria, va almacenando los datos que va recibiendo para así disponer de una reservación el objetivo de evitar que el Stream se congele. Como se mencionaba antes, esto ocurre en los reproductores multimedia, es decir, del lado del cliente.

- **Codecs**

Éstos funcionan como programas residentes en memoria de la computadora del cliente, están encargados de descifrar o interpretar la información que llega desde el servidor, ésta es descifrada basada en el formato de vídeo que se utilizó para cifrarla, una vez hecho esto se envía al reproductor para que éste lo reproduzca al usuario final.

Hay dos tipos de Codecs, éstos se dividen según la forma en que realizan su función:

- **Codecs transformadores:** Estos basan su funcionamiento en el ordenamiento eficiente de la información, para así poder ahorrar espacio.
- **Codecs predictores:** Estos operan con base en la eliminación de información innecesaria para la codificación.
- **Reproductor**

Éste es la interfaz que está entre el usuario y el contenido o Stream que se recibe, toma como soporte los codecs que hayan instalados en la computadora para así descifrar los datos y mostrarlos como una imagen o bien como audio, dependiendo de lo que se esté transmitiendo.

Existen reproductores para la computadora, como también reproductores para sitios web, en este caso se utilizará FlowPlayer, que es un reproductor web, a base de javascript, a este tipo de reproductores basta con decirles qué tipo de información deben leer y cómo mostrarla. (Arrecis, s. f.).

2.7. Herramienta para Transmitir Contenido Multimedia

2.7.1.FFmpeg

FFmpeg es una colección de software libre que puede grabar, convertir (transcodificar) y hacer streaming de audio y vídeo. Incluye libavcodec, una biblioteca de codecs. FFmpeg está desarrollado en GNU/Linux, pero puede ser compilado en la mayoría de los sistemas operativos, incluyendo Windows. El proyecto comenzó por Gerard Lantau, un seudónimo de Fabrice Bellard, y ahora es mantenido por Michael Niedermayer. Es destacable que la mayoría de los desarrolladores de FFmpeg lo sean también del proyecto MPlayer (más un miembro del proyecto Xine), y que FFmpeg esté hospedado en el servidor del proyecto MPlayer.(«FFmpeg» 2018)

2.7.2.Icecast

Es una herramienta para realizar Streaming de medios, se puede utilizar para crear estaciones de radio por Internet o para el uso que se quiera dar, siempre basado en Streaming. Es bastante versátil, ya que se pueden agregar nuevos formatos fácilmente, como también soporta estándares abiertos para comunicación e interacción.

Actualmente Icecast soporta Stream de varios tipos:

- MP3
- Ogg FLAC
- Ogg Theora
- AAC
- Ogg Vorbis
- Ogg Speex

Icecast nació como una herramienta para Streaming de audio, desde su versión 2.2.0 y es capaz de soportar Streaming de vídeo.

Aunque, actualmente existe una versión para Microsoft Windows, Icecast nació como una herramienta para GNU/Linux, es por ello que su configuración se realiza desde archivos de texto plano, quizás un poco rústica, pero está bien organizado y no importando la plataforma la configuración es estándar.

Icecast funciona con puntos de montaje, o mountpoints, los cuales conforman cada una de las líneas de distribución que éste tiene en un momento específico. Para enviar datos hacia el servidor Icecastse necesita de alguna herramienta que permita enviar un Stream hacia él, mediante hypertext transfer protocol, más conocido como http.

Como se mencionó anteriormente, la configuración es fácil, a través de archivos de texto, pero se debe tener mucho cuidado en algo, y esto es el cambio de la contraseña que tiene por default, ya que si no se cambia ésta, cualquier persona que sepa del famoso Admin→ podrá acceder al servidor y modificar lo que desee a su antojo. (Arrecis, s. f.)

2.8. Lenguajes de Programación

2.8.1. Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Python es un lenguaje de programación muy popular, hoy en día entre los programadores, sobre todo para aquellos que se dedican a la programación de Raspberry Pi porque es software y hardware libre aprobada por OSI. (Admin, 50 ebooks gratis en Phytion, 2014).

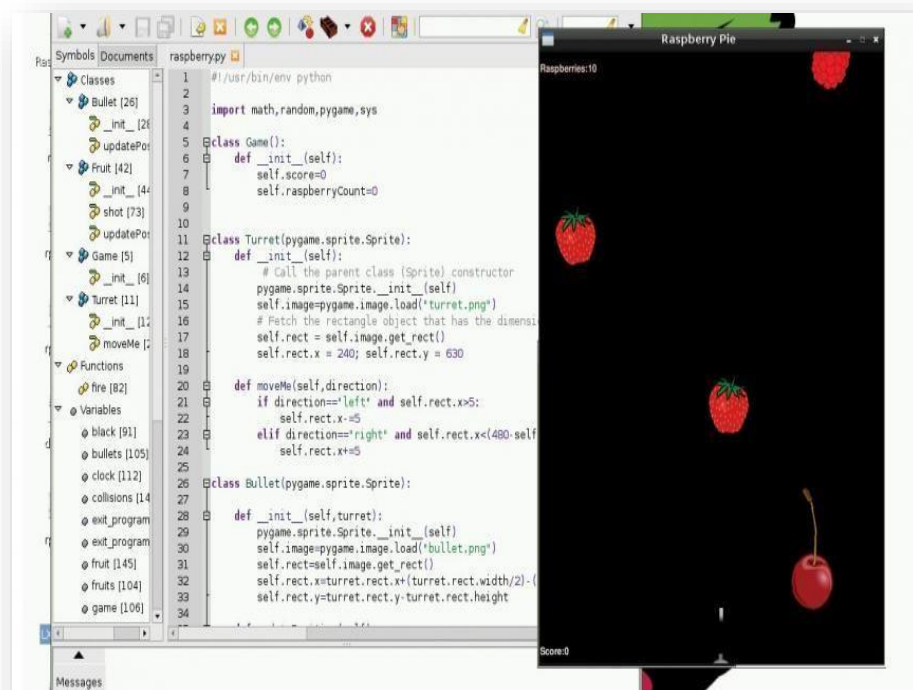


Figura7: Código de Python en Raspberry Pi

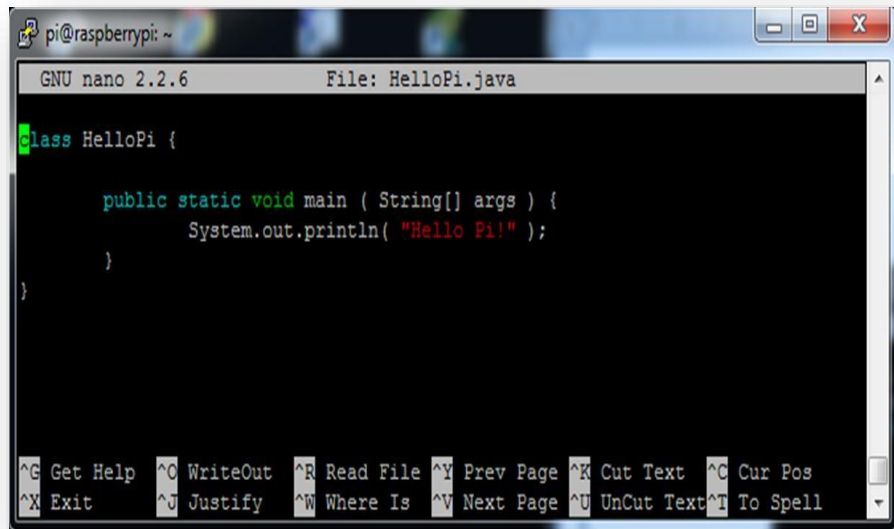
2.8.2. Java

Java es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana, como veremos en el punto 4.2. También posee otras características muy importantes:

- Es un lenguaje que es compilado, generando ficheros de clases compilados, pero estas clases compiladas son en realidad interpretadas por la máquina virtual java. Siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando.
- Es un lenguaje multiplataforma: El mismo código java que funciona en un sistema operativo funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java.
- Es un lenguaje seguro: La máquina virtual al ejecutar el código java realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.
- Gracias al API de java podemos ampliar el lenguaje para que sea capaz de, por ejemplo, comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas, crear aplicaciones visuales al estilo Windows

Para poder trabajar con java es necesario emplear un software que permita desarrollar en java, existen varias alternativas comerciales en el mercado: JBuilder, Visual Age, Visual Café, y un conjunto de herramientas shareware e incluso freeware que permiten trabajar con java. Pero todas estas herramientas en realidad se basan en el uso de una herramienta proporcionada por Sun, el creador de java, que es el Java Development Kit (JDK). Nosotros nos centraremos en el uso de dicha herramienta.

Existen diversas versiones del JDK, siendo posible obtener cualquiera de dichas versiones desde la propia página de sun: <http://java.sun.com> siendo la más reciente JSDK 1.4 (Exes [sin fecha])



```
pi@raspberrypi: ~  
GNU nano 2.2.6 File: HelloPi.java  
  
class HelloPi {  
  
    public static void main ( String[] args ) {  
        System.out.println( "Hello Pi!" );  
    }  
}  
  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura14: Código de Java en Raspberry Pi

2.9. Software de Diseño de PCB

2.9.1.Eagle

Autodesk EAGLE es un software de automatización de diseño electrónico (EDA). Permitir a los diseñadores de placas de circuito impreso (PCB) que conecten sin problemas diagramas esquemáticos, colocación de componentes, enrutamiento de PCB y contenido completo de la biblioteca.

El estándar EAGLE incluye 99 hojas de esquemas, 4 capas de señales y un área de placa de 160 cm². EAGLE Premium incluye 999 hojas de esquemas, 16 capas de señal y áreas de tablero ilimitadas.

Requisitos del sistema

- Se requiere Microsoft® Windows® 7 o posterior.
- EAGLE de 64 bits requiere un sistema operativo de 64 bits.
(«EAGLE | PCB Design Software | Autodesk»)

2.9.2. Proteus ISIS

Proteus Design Suite es una solución de software completa para simulación de circuitos y diseño de PCB. Comprende varios módulos para la captura esquemática, el firmware IDE y el diseño de PCB que aparecen como pestañas dentro de una única aplicación integrada. Esto proporciona un flujo de trabajo ágil de AGILE para el ingeniero de diseño y ayuda a que los productos lleguen al mercado más rápido.

Los productos Proteus PCB Design incluyen tanto la captura esquemática como los módulos de diseño de PCB y están diseñados para ser fáciles de usar y potentes. Las características tales como un autorouter basado en la forma de clase mundial, visualización 3D, ajuste de red automático, fragmentos de diseño y variantes de montaje le ahorran tiempo durante el diseño del producto. Mientras tanto, un poderoso sistema de reglas de diseño impone las reglas y autorizaciones que pueda necesitar para su PCB. El enrutamiento de las pistas es totalmente consciente de las reglas de diseño y la comprobación de la habilitación en tiempo real facilita la localización y corrección de cualquier infracción.

Requisitos del sistema

- El rendimiento en el software Proteus generalmente es escalable de acuerdo con las especificaciones de su máquina. Sin embargo, los siguientes pueden considerarse los requisitos mínimos (correctos a partir de otoño de 2017).
- Procesador Intel de 2 GHz o más rápido (procesadores AMD finos pero menos optimizados). Tarjeta gráfica compatible con OpenGL versión 2.0 o superior y / o Direct 3D y multi-sampling (MSAA). 3GB de RAM (recomendado 8GB). Microsoft Windows XP o posterior (recomendado Microsoft Windows 7 de 64 bits o posterior).
- En particular; tenga en cuenta que si su tarjeta gráfica no cumple con los requisitos anteriores, el software se ejecutará en modo Windows GDI. Esto significa que la visualización de la pantalla es manejada por

Windows y no por su hardware de gráficos y, por lo tanto, algunas funciones del software no estarán disponibles y el rendimiento podría verse afectado negativamente. Como regla general, las tarjetas gráficas discretas tales como las de AMD (ATI) y NVIDIA satisfarán estos requisitos, mientras que los gráficos de chipset como los suministrados por Intel no lo harán.

(«Questions - find out about common ordering questions and answers - Proteus Design Suite»).

DISEÑO Y FUNCIONAMIENTO

CAPÍTULO III

CAPÍTULO III: DISEÑO Y FUNCIONAMIENTO

3.1. Introducción

A continuación, se describirá el funcionamiento de nuestro de timbre inalámbrico inteligente con vídeo, y los materiales necesarios para poder lograr el objetivo. Además, se ilustrarán: Diagrama de conexión, Funcionamiento, Diagrama esquemático, Diagrama de ubicación de los diferentes dispositivos, Detalles de montaje, también se realizará una descripción de los diferentes códigos a usar y como estos trabajan en conjunto.

3.2. Descripción del Sistema

En el siguiente proyecto se utilizará una Raspberry Pi 3B+, para el control de las llamadas telefónicas y envío de mensajes de texto que se harán al módulo GPRS/GSM sim800L, y la inicialización de la transmisión de video a través de la cámara web utilizando el lenguaje de programación Python, además contará con tarjeta electrónica (PCB), la cual integrará la tarjeta SIM800L, una batería de 3.7voltios, entre otros dispositivos electrónicos (capacitores, resistores, reguladores, leds, switch), y para la comunicación entre ambas se utilizará el conversor de USB a TTL y no los pines del puerto serie del GPIO de la Raspberry Pi Modelo 3B.

3.3. Lista de Materiales

La lista de materiales para el timbre inteligente lo hemos separado en dos cuadros:

3.3.1. Lista de materiales para la Raspberry Pi 3B+

Instrumento	Ubicación
Fuente de la alimentación conmutada	Conectado a la entrada de alimentación de la Raspberry Pi (Micro USB Power input)
Cámara Web	Conectado a un puerto USB a la Raspberry Pi 3B
Ventilador 5V	Conectado a los pines de voltaje de la Raspberry Pi (GPIO)
Cable Ethernet	Puerto Ethernet de la Raspberry pi (10/100 LAN port)

Tabla 1: Lista de complementos para la Raspberry pi.

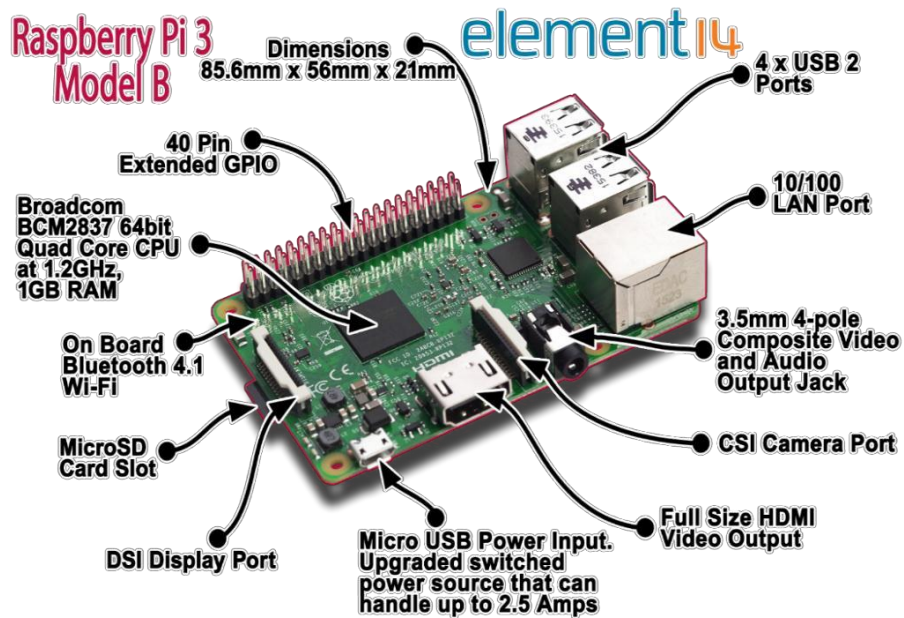


Figura 15: Puertos y conexiones de la Raspberry pi 3

3.3.2. Listado de Materiales para la tarjeta electrónica

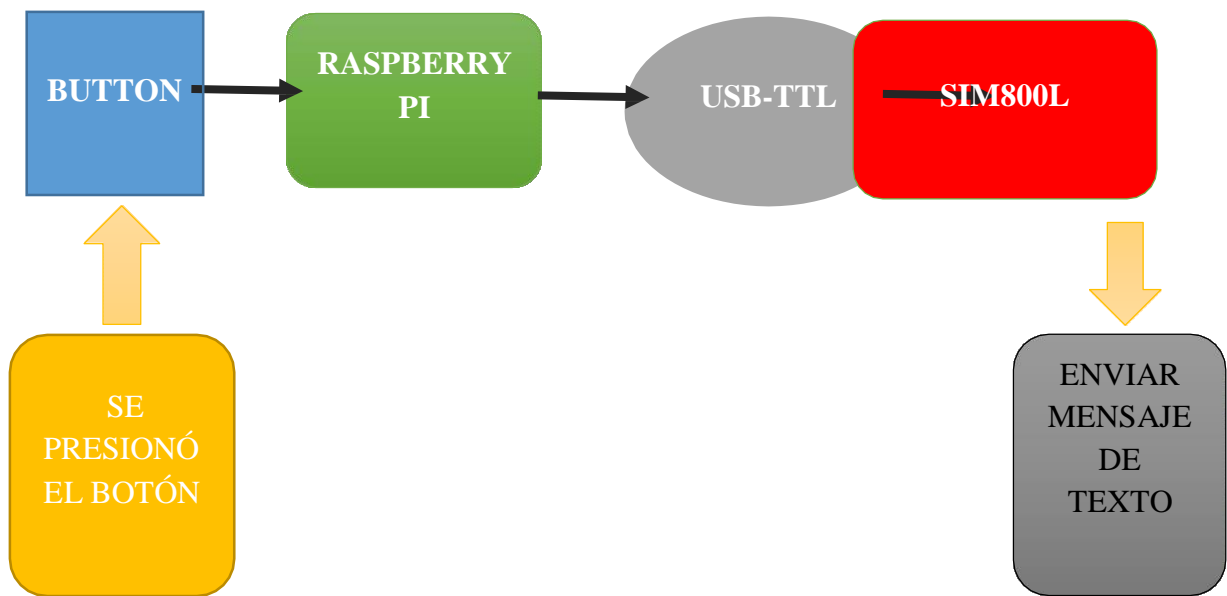
MATERIAL	Tipo	CANTIDAD	VALOR	Encapsulado
Capacitor	Cerámico	10	10pF	SMD 0808
Capacitor	Cerámico	10	33pF	SMD 0808
Capacitor	Cerámico	2	1uF	SMD 0808
Capacitor	electrolítico	2	100uF 6.3V	TH
Capacitor	electrolítico	2	100uF	TH
Resistor		1	420	SMD 0808
Resistor		1	4.7K	SMD 0808
Switch	Dos Toques	1		
Regulador	variable	1	Lm317	TO220
Regulador		1	5V	TO220
Regulador		1	3.3V	SOT220
PCB	FR4	1		

Tabla 2: Lista de Materiales para la tarjeta.

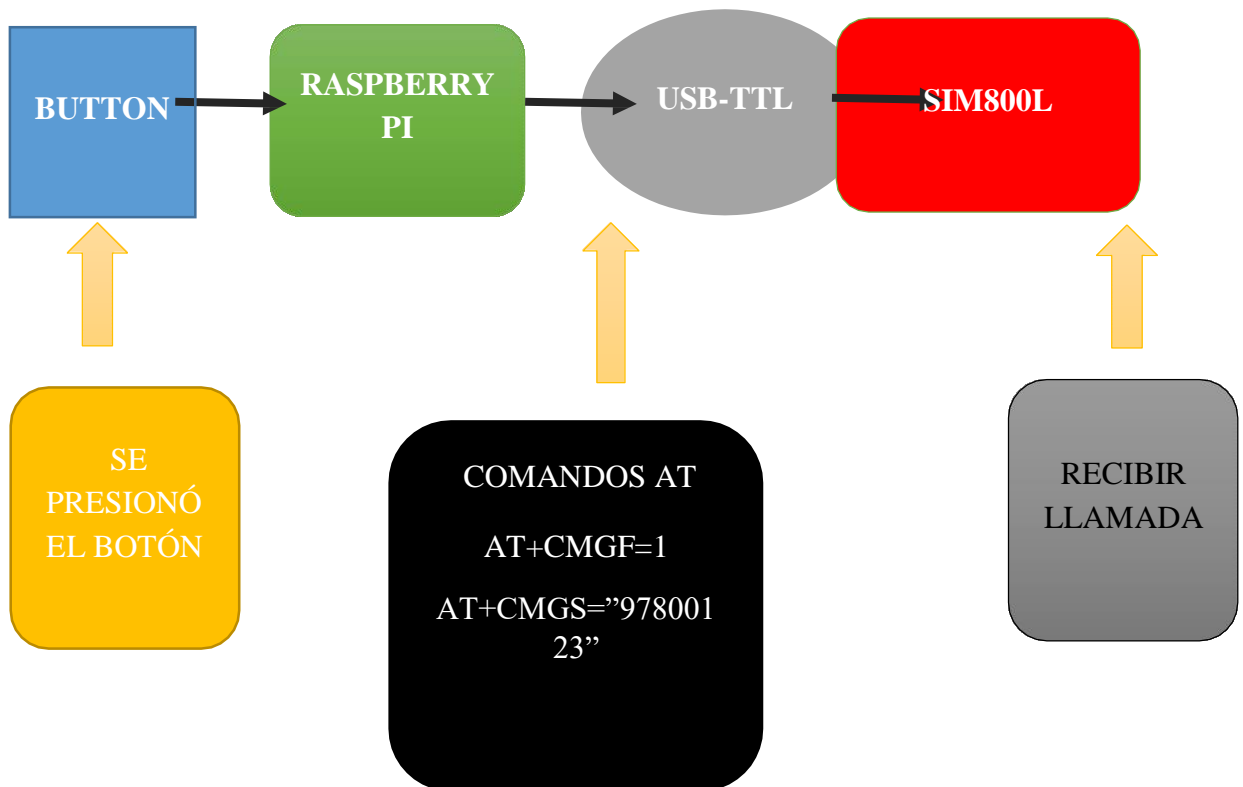
3.3.3. Funcionamiento

3.3.3.1. Raspberry Pi

Cuando se presione el botón del timbre inteligente, este dispositivo se encargará de enviar los diferentes comandos AT, y estos se enviarán a través del puerto USB serial utilizando la biblioteca del mismo nombre “serial”, con el apoyo del conversor de USB a TTL. Cuando se envíe el mensaje de “llaman a la puerta”, el usuario podrá hacer una llamada al número de la tarjeta SIM800, el cual contestará automáticamente después del segundo timbrado, una vez que conteste la Raspberry Pi activará el Video – Streaming y el usuario podrá ver desde la pantalla de su celular quién está tocando la puerta, y así poder establecer la comunicación con la persona que tocó el timbre.



Para que envíe el mensaje de texto, la Raspberry Pi envía los siguientes comandos AT a través del puerto USB hacia el conversor de USB a TTL (CP2102).



3.3.3.2. Código en Python para enviar mensaje

Se define como variables al número de teléfono móvil y el mensaje a enviar, “recipient” es la variable que guarda el número de teléfono y “message” es la variable que guarda el mensaje de Texto “Tocan Timbre”.

```
class TextMessage:  
  
def __init__(self, recipient="", message=""):  
  
self.recipient = recipient  
  
self.content = message  
  
def sendMessage(self):  
  
self.ser.write('AT+CMGF=1r')  
  
time.sleep(1)  
  
self.ser.write(''AT+CMGS="" + self.recipient + ""r'')  
  
time.sleep(1)  
  
self.ser.write(self.content + 'r')  
  
time.sleep(1)  
  
self.ser.write(chr(26))  
  
time.sleep(1)
```

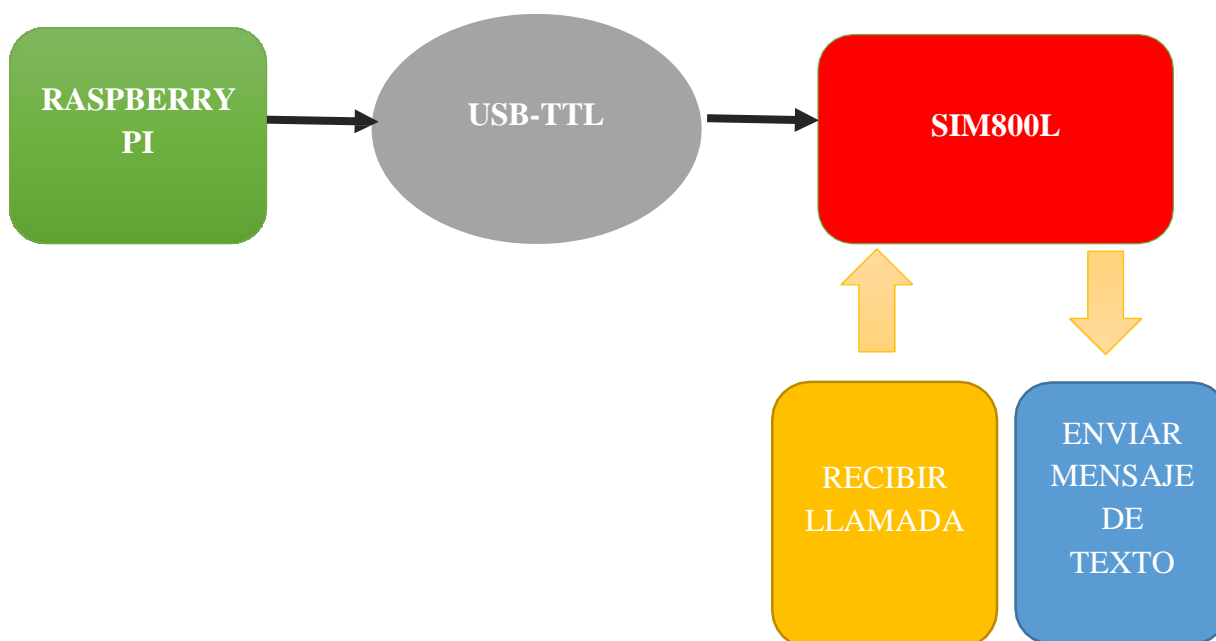
En esta línea de código definimos las variables ya mencionada anteriormente que son “recipient” y “message”.

```
sms = TextMessage('97800123', 'Tocan Timbre!')
```

3.3.3.3. Sim800L

Este módulo es quien recibirá los comandos AT enviado por la Raspberry Pi, debidamente ya configurado, este módulo GPRS/GSM podrá enviar mensajes de texto y recibir llamadas.

Para su correcto funcionamiento este módulo debe alimentarse con un voltaje entre los 3.4v- 4.4v.



3.3.3.4. Conversor de USB a TTL

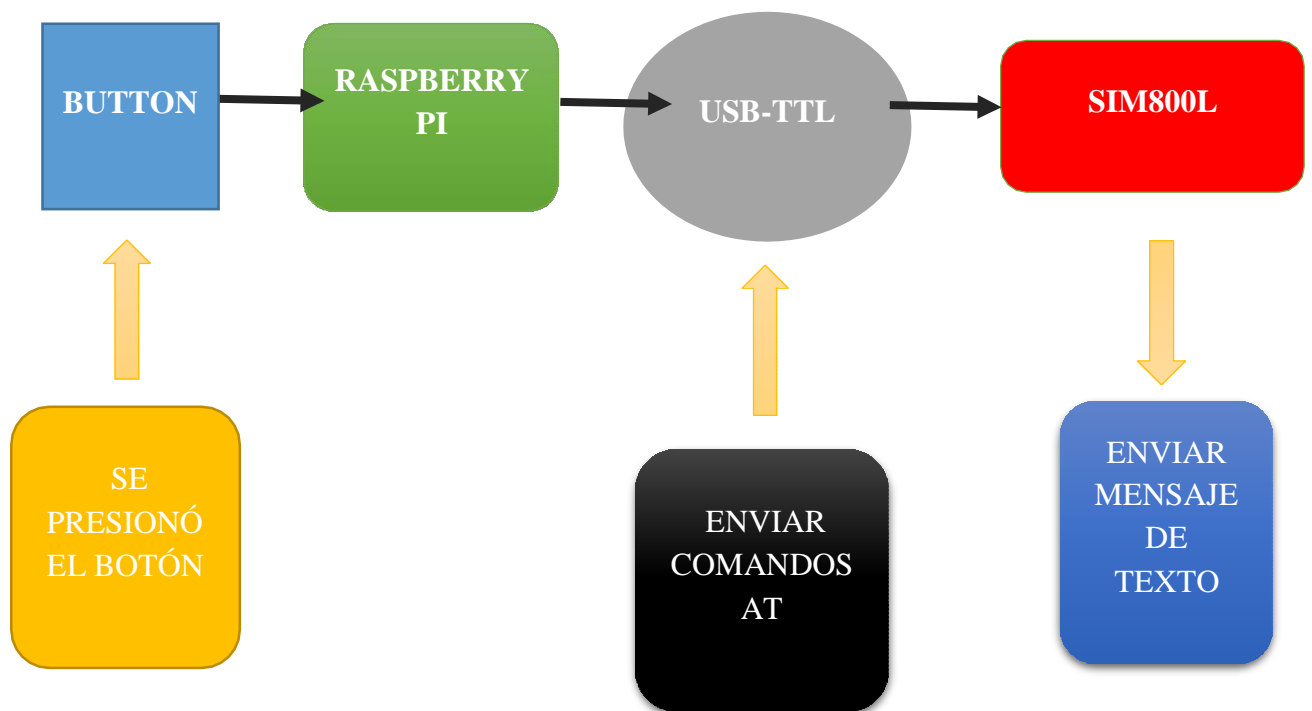
Este módulo juega un papel importante, ya que no se va utilizar los pines GPIO de la Raspberry Pi Modelo 3B+, este módulo ayudará en la fluidez y velocidad en la que se puede comunicar la Raspberry Pi y el Módulo Sim800L.

Es importante resaltar que, dentro del código en Python el conversor de USB a TTL CP2102 se tiene que configurar el puerto de salida *dev/ttyUSB0*, la velocidad de transmisión o baudrate **115200**.

```
def connectPhone(self):
```

```
self.ser = serial.Serial('/dev/ttyUSB0', 115200, timeout=5)
```

```
time.sleep(1)
```



3.3.3.5. Streaming

Para el Streaming se ha utilizado la biblioteca FFmpeg, y un extenso código en Python el cual se activará cuando el usuario haga la llamada al módulo sim800L

El Streaming se desarrolla a través de la plataforma de YouTube en la que se puede acceder al video en vivo y con una cuenta privada de YouTube del mismo usuario, cabe resaltar que no se necesita un servidor de Streaming privado para poder llevar a cabo este proyecto.

Se declaran las bibliotecas a utilizar

#!/usr/bin/env python3

Esta biblioteca habilita todas las funciones de rutinas o subprocessos que se puedan llevar dentro del código de programación en Python utilizando uno o hasta los 4 núcleos del ARM de la Raspberry pi.

import subprocess

Esta biblioteca habilita todas las funciones de la cámara que se instala en la Raspberry pi, tal como su inicialización, toma de fotos, grabar un video y cierre de la cámara.

import picamera

Esta biblioteca habilita todas las funciones de tiempo de espera o de fecha y hora de la Raspberry Pi.

import time

En esta línea de código se declara

YOUTUBE="rtmp://a.rtmp.youtube.com/live2/"

En esta línea de código se declara la clave de usuario que te brinda YouTube para la transmisión de video en vivo a través de sus servidores.

KEY= 3eju-s91p-bd66-7aus

En esta línea de código se declara las especificaciones de cómo deseas transmitir tu video, el formato, si transmites video con audio:

stream_cmd = 'ffmpeg -f h264 -r 25 -i - -itsoffset 5.5 -fflags nobuffer -f alsa -ac 1 -i hw:1,0 -vcodec copy -acodec aac -ac 1 -ar 8000 -ab 32k -map 0:0 -map 1:0 -strict experimental -f flv ' + YOUTUBE + KEY

stream_pipe = subprocess.Popen(stream_cmd, shell=True, stdin=subprocess.PIPE)

En esta línea del código se declara la resolución en la que se va transmitir el video

camera = picamera.PiCamera(resolution=(640, 480), framerate=25)

try:

now = time.strftime('%Y-%m-%d-%H:%M:%S')

camera.framerate = 25

camera.vflip = True

camera.hflip = True

En esta línea del código se declara el formato de transmisión de video junto a los bitrate

camera.start_recording(stream.stdin, format='h264', bitrate = 2000000)

En esta línea del código se declara el formato de transmisión de video junto a los bitrate

while True:

camera.wait_recording(1)

En estas líneas del código se cancela la transmisión de video con la interrupción de fin de la llamada

except commandAT==ATH:

camera.stop_recording()

En esta línea de código se declara la desactivación de la cámara y también la espera para una nueva transmisión de video.

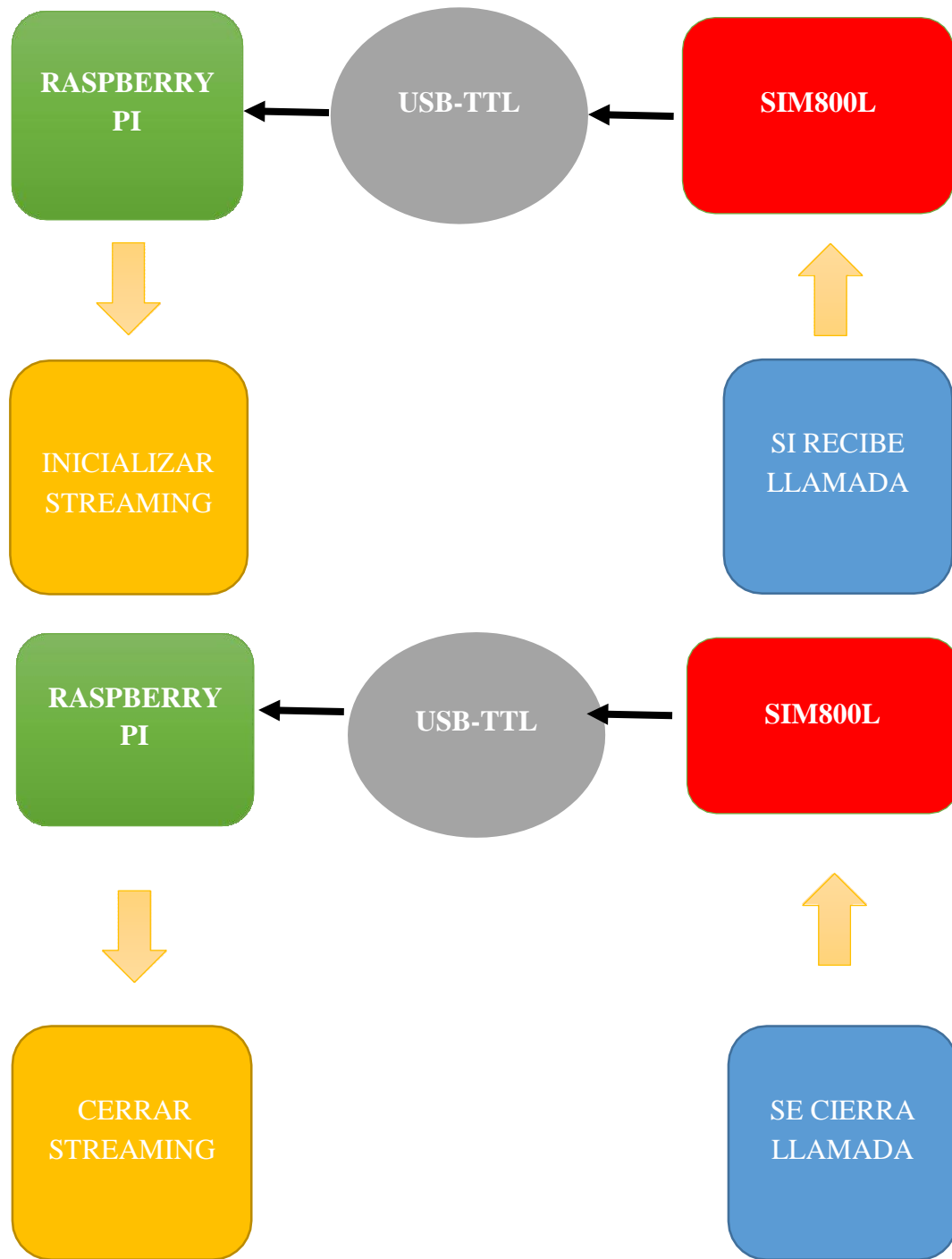
finally:

camera.close()

stream.stdin.close()

stream.wait()

print("Good bye")



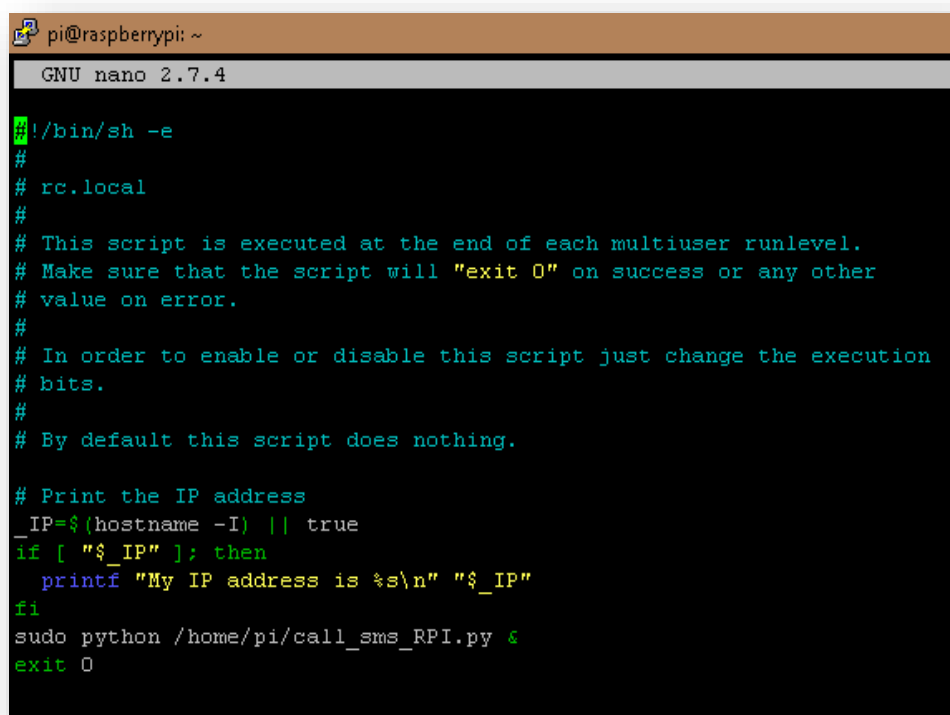
3.3.3.6. Ejecución Automática

Una vez llevado acabo todos los resultados, lo último que se debe hacer es que el script sea ejecutable y agregarlo al archivo rc.local para que se inicie automáticamente cada vez que se inicie el Pi.

- `sudo chmod +x youtube_stream.py`
- `sudo nano /etc/rc.local`

Hacia la parte inferior del archivo rc.local, justo antes de "exit 0", agregue esta línea:

- `sudo python /home/pi/call_sms_RPI.py &`



```
pi@raspberrypi: ~  
GNU nano 2.7.4  
#!/bin/sh -e  
#  
# rc.local  
#  
# This script is executed at the end of each multiuser runlevel.  
# Make sure that the script will "exit 0" on success or any other  
# value on error.  
#  
# In order to enable or disable this script just change the execution  
# bits.  
#  
# By default this script does nothing.  
  
# Print the IP address  
_IP=$(hostname -I) || true  
if [ "$_IP" ]; then  
    printf "My IP address is %s\n" "$_IP"  
fi  
sudo python /home/pi/call_sms_RPI.py &  
exit 0
```

Figura 16: Ejemplo de script automático

Guardar y reiniciar el Pi. Una vez que haya terminado de reiniciar, simplemente el programa estará a la espera de las instrucciones que son el presionado del botón y la recepción de la llamada.

3.4. Diseño de PCB

Para el diseño de PCB se tomará en cuenta lo siguiente:

- Se utilizarán dos tipos de voltaje, uno de 3.3V y otro de 3.7V.
- Se utilizarán dispositivos con encapsulado de montaje superficial (smd) y de agujero pasante (TH).
- Se utilizará un buffer para las comunicaciones entre el sim800L y el módulo CP2102.

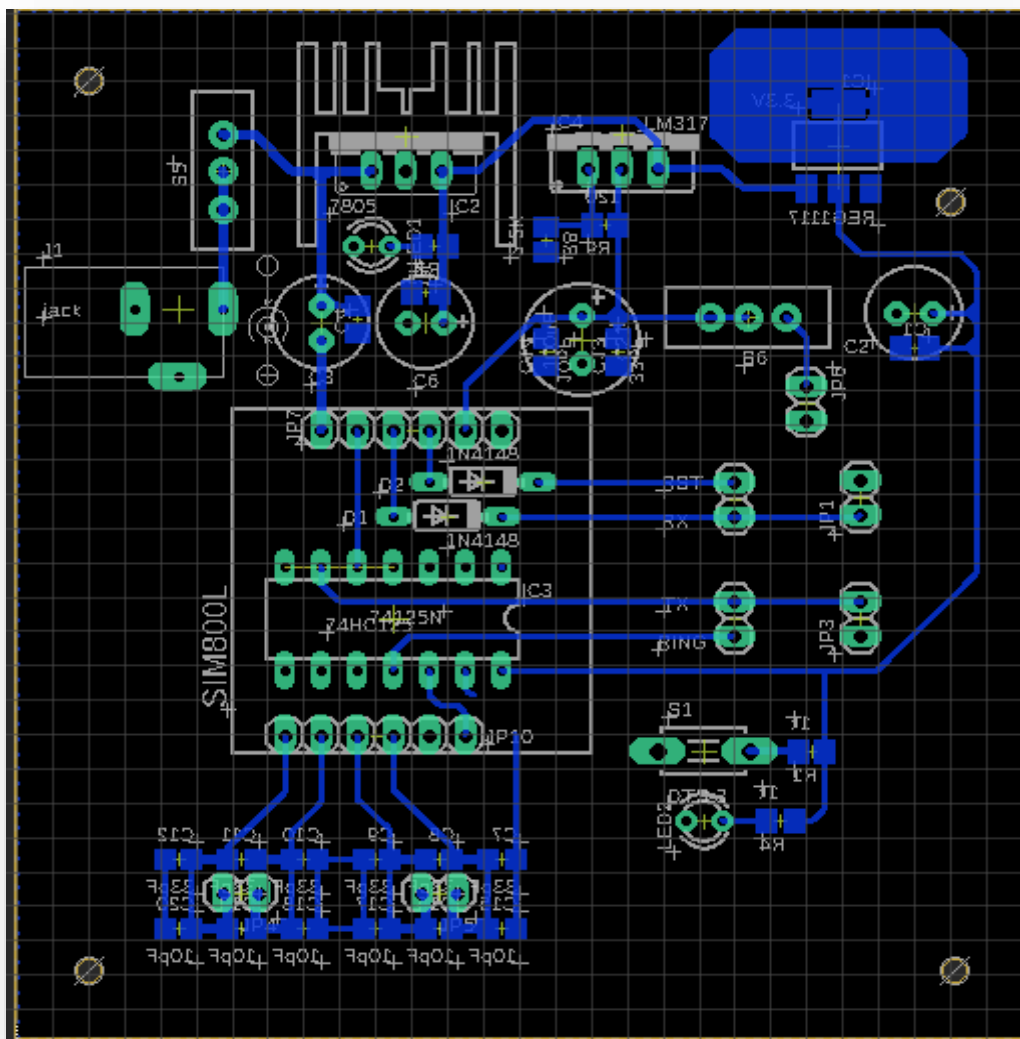


Figura 17: Board de la tarjeta electrónica

3.5. Armado de PCB

En el Esquemático se puede observar los diferentes dispositivos que se utilizaron para desarrollar la tarjeta electrónica, como son los circuitos integrados del buffer, los reguladores de voltaje, los resistores y capacitores.

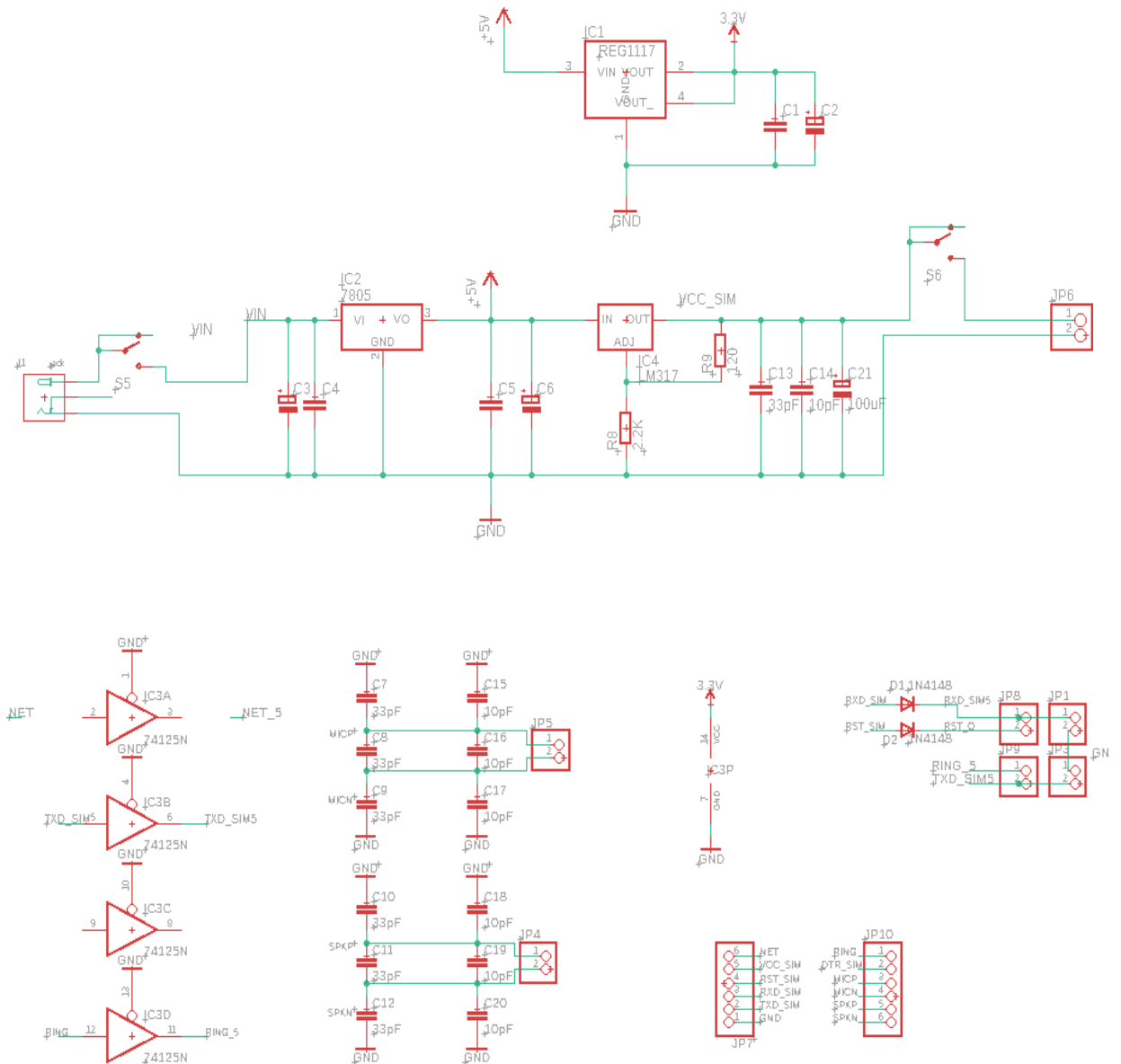


Figura 18: Esquemático de la tarjeta electrónica

3.6. Conexiones

A continuación se detallará en un diagrama de Flujo las conexiones del timbre inteligente:

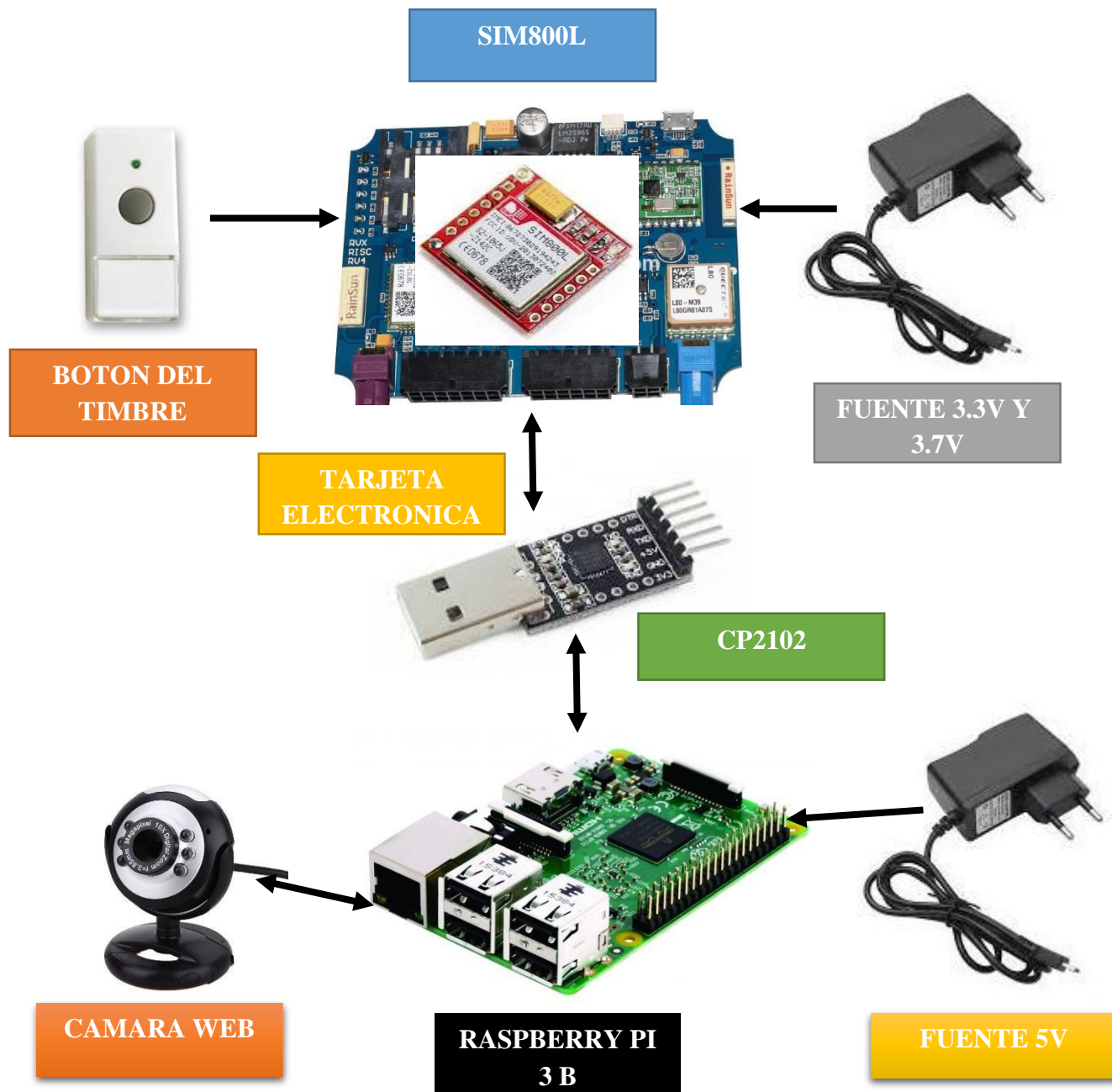


Figura19. Conexiones

3.7. Especificación de la Raspberry Pi Modelo 3B



Figura 20: Vista superior de la Raspberry pi

3.7.1. Características Técnicas

A continuación veras la lista completa de las características de la nueva Raspberry Pi 3 B+:

- **Procesador:** Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz.
- **Memoria RAM:** 1GB LPDDR2 SDRAM.
- **Wi-Fi:** Doble-Banda 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac.
- **Bluetooth:** 4.2, BLE.
- **Conexión LAN:** puerto Gigabit Ethernet over USB 2.0 (rendimiento máximo de 300 Mbps).
- **GPIO:** 40 pines.
- **HDMI.**
 - 4 puertos **USB 2.0**.
 - Puerto de cámara CSI para conectar una cámara Raspberry Pi.
 - Puerto de pantalla DSI para conectar una pantalla táctil.
- Salida estéreo de 4 polos y puerto de vídeo compuesto.

- Puerto Micro SD para cargar su sistema operativo y almacenar datos.
- Entrada de alimentación de 5V / 2.5A DC.
- Soporte PoE (viene por separado).

3.8. Especificaciones del módulo GSM

3.8.1.SIM800L



Figura 21: SIM800L

3.8.1.1. Descripción

SIM800C es una solución completa de banda cuádruple GSM / GPRS en tipo SMT, que puede integrarse en las aplicaciones del cliente.

SIM800C es compatible con Quad-band 850/900/1800 / 1900MHz, puede transmitir voz, SMS e información de datos con bajo consumo de energía. Con un pequeño tamaño de 17.6 * 15.7 * 2.3 mm, puede adaptarse sin problemas a las demandas del diseño del cliente, esbelto y compacto.

3.8.1.2. Características y Especificaciones

- Protección Contra Golpes y Vibraciones ETSI300-019-1.4
- Cuatro bandas 850/900/1800 / 1900MHz
- GPRS multi-slot clase 12/10
- Estación móvil GPRS clase B
- Compatible con GSM fase
 - 2/2+-Clase 4 (2 W @ 850 / 900MHz)
 - -Clase 1 (1 W @ 1800 / 1900MHz)
- Dimensiones: 17.6 * 15.7 * 2.3mm
- Peso: 1.3g
- Control a través de comandos AT
 - (3GPP TS 27.007, 27.005 y Comandos AT mejorados por SIMCom)
- Rango de voltaje de alimentación 3.4 ~ 4.4V
- Bajo consumo de energía
- Temperatura de funcionamiento: -40 °C ~ 85 °C(«SIM800C» [sin fecha])

3.9. Especificación del conversor USB a TTL

3.9.1. CP2102



Figura 22: Modulo CP2102

3.9.1.1. Descripción

El CP2102 es un controlador de puente USB a UART altamente integrado que proporciona una solución simple para actualizar RS-232 diseños a USB utilizando un mínimo de componentes y espacio de PCB. El CP2102 incluye un USB 2.0 de alta velocidad controlador de función, transceptor USB, oscilador, EEPROM y bus de datos en serie asíncrono (UART) con señales de control de módem en un paquete compacto MLP-28 de 5 x 5 mm. No se requieren otros componentes USB externos.

El EEPROM integrado en el chip se puede utilizar para personalizar la identificación del proveedor de USB, la identificación del producto, la cadena de descripción del producto.

Descriptor de alimentación, número de liberación del dispositivo y número de serie del dispositivo según lo deseado para las aplicaciones OEM. Los EEPROM se programan a bordo a través del USB permitiendo que el paso de programación se integre fácilmente en el proceso de fabricación y prueba del producto.

3.9.1.2. Características técnicas

Transceptor USB integrado; sin resistencias externas necesario

- Reloj integrado; no se requiere cristal externo
- EEPROM integrado de 1024 bytes para la identificación del proveedor, producto
- ID, número de serie, descriptor de potencia, número de versión, y cadenas de descripción del producto
- Circuito de reinicio de encendido en el chip
- Regulador de voltaje en chip: salida de 3.3 V
- Pin 100% y software compatible con CP2101

3.10. Especificaciones del SO Raspbian

Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Stretch(Debian 9.4) para la placa computadora (SBC) Raspberry Pi, orientado a la enseñanza de informática. El lanzamiento inicial fue en junio de 2012.

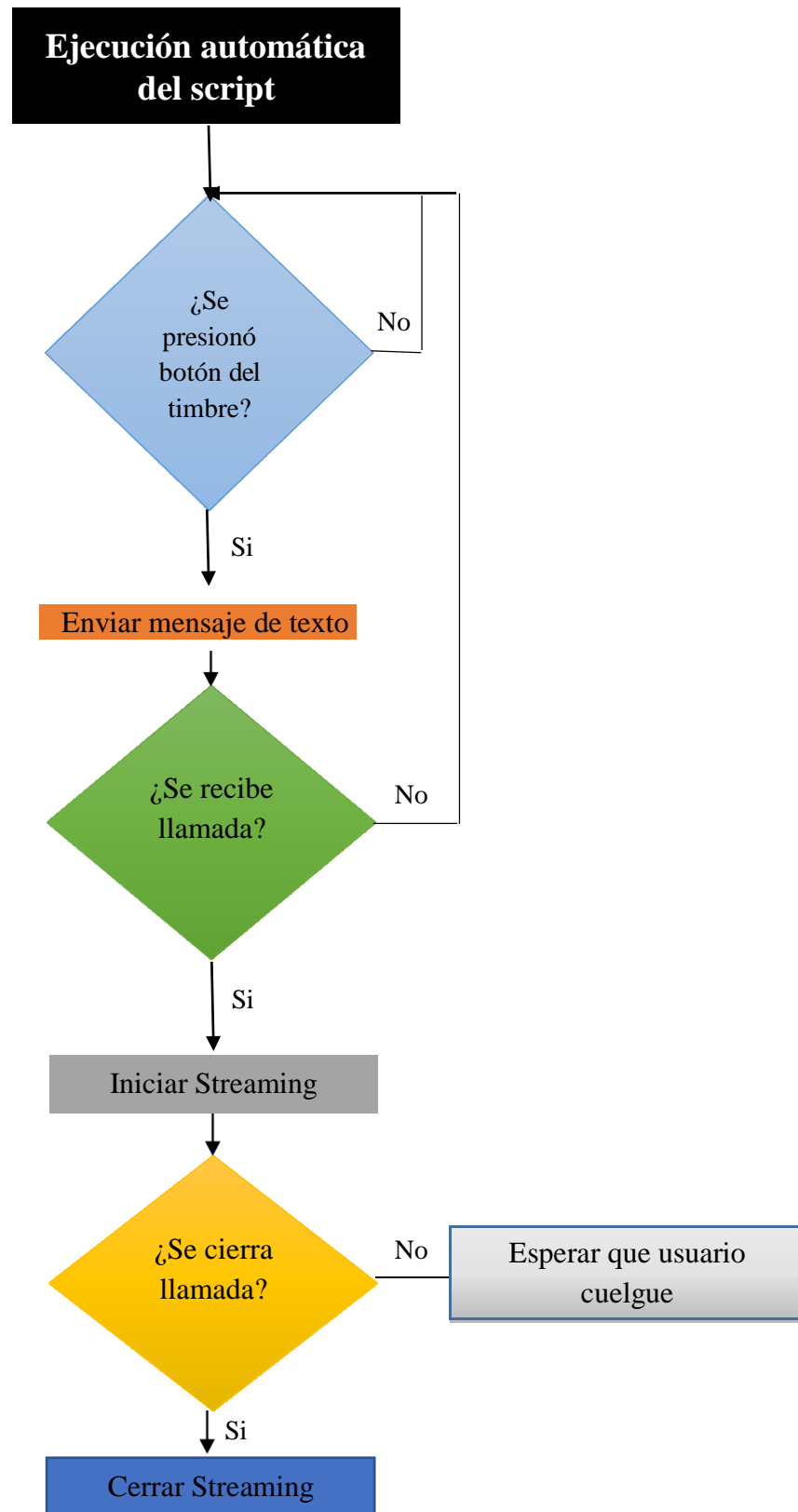
Técnicamente el sistema operativo es un port no oficial de Debian armhf para el procesador (CPU) de Raspberry Pi, con soporte optimizado para cálculos en coma flotante por hardware, lo que permite dar más rendimiento en según qué casos. El port fue necesario al no haber versión Debian armhf para la CPU ARMv6 que contiene el Raspberry PI.

La distribución usa LXDE como escritorio y Chromium como navegador web. Además contiene herramientas de desarrollo como IDLE para el lenguaje de programación Python o Scratch, y diferentes ejemplos de juegos usando los módulos Pygame.

Destaca también el menú "raspi-config" que permite configurar el sistema operativo sin tener que modificar archivos de configuración manualmente. Entre sus funciones, permite expandir la partición root para que ocupe toda la tarjeta de memoria, configurar el teclado, aplicar overclock, etc.

El 17 de diciembre de 2012, junto a la versión 2012-12-16-wheezy-raspbian de Raspbian, se lanzó la tienda de aplicaciones "Pi Store", que en el momento de salida incluía desde aplicaciones como LibreOffice o Asterisk a juegos como Freeciv o OpenTTD. En esta plataforma se puede poner a disposición de todos los usuarios de Raspbian, mediante moderación y posterior lanzamiento, contenidos gratuitos o de pago, como archivos binarios, código python, imágenes, audio o vídeo. Además se quiere incluir documentación acerca del Raspberry Pi como la revista MagPi y tutoriales de proyectos. Actualmente, la tienda ya no se incluye en el sistema pero se puede instalar ejecutando el comando "sudo apt-get update && sudo apt-get install pystore"

3.12. Diagrama de Flujo



3.13. Lista de Comandos AT utilizados

AT+CPIN?

Comando para verificar si el módulo Sim8000L está listo para funcionar.

AT+CGREG?

Comando para verificar si el módulo ya está conectado a una red móvil.

AT+COPS?

Comando que nos brinda información de la red a la q está conectada el módulo Sim800L.

AT+CSQ

Comando que nos brinda información del valor de la señal de la red a la que está conectada el módulo Sim800L.

AT+CLIP

Comando que nos brinda información del usuario del módulo Sim800L.

AT+CMGF

Comando para seleccionar el tipo de formato de mensaje 0-PDU mode, 1-text mode.

AT+CMGS

Comando para enviar mensaje.

EVALUACIÓN ECONÓMICA

CAPÍTULO IV

CAPÍTULO IV: EVALUACIÓN ECONÓMICA

4.1. Costos de Hardware y Software

CANTIDAD	MATERIALES	PRECIO UNITARIO	PRECIO TOTAL S/.
1	Raspberry Pi 3 B	250	250.00
1	Sim800L	14	14.00
8	Cap. 10pF smd	1	8.00
8	Cap 33pF smd	1	8.00
1	Bateria 3.7V 1000mA/H	30	30.00
2	Cap 1000uF 6.3V	2	4.00
1	Resistor 120 smd	1	1.00
1	Resistor 4.7K smd	1	1.00
1	Mini Parlante de 8 ohmios – 1W	10	10.00
1	Micrófono Eléctrico	5	5.00
1	Cámara web	40	40.00
1	Conector plug	1	1.00
1	Fuente Switching 5V	40	40.00
1	Fuente Switching 9V	30	30.00
2	Led 3mm	0.5	1.00
1	PCB 8cmx10cm	15	15.00
		SUB TOTAL	S/. 458.00
		IGV	S/. 82.44
		TOTAL	S/. 540.44

Tabla 3: Costos de Hardware

4.2. Costos Totales

DESCRIPCION	PRECIO TOTAL
Costo de Hardware	S/. 540.44
Costo de Software	S/. 0.00
Costo de Ingeniería	S/. 300.00
TOTAL	S/. 840.44

Tabla 4: Costos Totales.

4.3. Justificación Económica

Una vez terminado, los recursos económicos no serán grandes, más bien mínimos, por lo que ya se dispone del software y hardware necesarios. Se tomó en cuenta adquirir algunos de los materiales a través de compras online para reducción de costos.

La elaboración de los documentos que acompañan al software creado, y los gastos que ello ocasionare por fotocopiado, anillado, empastado y/o otros serán asumidos por el investigador.

CONCLUSIONES

Al terminar el proyecto de tesis, se llegó a las siguientes conclusiones:

1. Se estudiaron los diferentes sistemas de monitoreo como cámaras directamente conectadas a un VDR, cámaras con conexión a internet y cámaras de video vigilancia supervisadas por empresas públicas o privadas.
2. Se diseñó y se logró interconectar con interfaz directa el timbre y la Raspberry por medio del pin 23, luego a través del conversor de USB a TTL utilizando el protocolo RS-232 conectamos la Raspberry con el Sim800L. Luego se logró conectar el teléfono móvil con nuestro sistema, a través del aplicativo AndroindStreamingMediaPlayer.
3. Seleccionamos la cámara web raspberry pi 3B+, el módulo GSM Sim800L, el conversor serial CP2102, ya que son el hardware adecuado para trabajar con la Raspberry de acuerdo al datasheet del fabricante.
4. Realizando pruebas en diferentes escenarios a diferentes distancias, se logró que el sistema funcione, logrando transmitir video por streaming.

RECOMENDACIONES

1. Para el buen funcionamiento del sistema es necesario que exista una señal ininterrumpida de internet, con 2Mbps de UPLOAD, para así poder tener una transmisión de video, sin retardo y con buena calidad de imagen.
2. El celular debe ser de media gama alta por las prestaciones de fabricación. cuente con una conexión a internet móvil de buena calidad, y memoria RAM mayor a los 2GB.
3. Comprar fuentes Switching certificadas para garantizar el buen funcionamiento del sistema.
4. Añadir una batería adicional para nuestro sim800L, ya que este tiene picos de consumo de corriente de hasta 2A.

.

REFERENCIA BIBLIOGRÁFICA

- Admin. (2014). 50 ebooks gratis en Phyton. Retrieved from <http://www.raspberrystore.es/wp/50-ebooks-gratis-en-python-para-utilizar-con-la-raspberry-pi/>
- Admin. (2014). Adafruit actualiza sus librerías en Python para la Raspberry Pi. Retrieved from <http://www.raspberrystore.es/wp/adafruit-actualiza-sus-librerias-en-python-para-la-raspberry-pi/>
- Canonical LTDA. (2014). Imagen Writer. Retrieved from <https://launchpad.net/win32-image-writer/+download>
- EAGLE | PCB Design Software | Autodesk. [en línea], [sin fecha]. [Consulta: 5 agosto 2018]. Disponible en: <https://www.autodesk.com/products/eagle/overview>.
- El timbre: Un invento útil y con curiosos orígenes | Guioteca.com. *Guioteca.com / Curiosidades* [en línea], 2014. [Consulta: 9 junio 2018]. Disponible en: <https://www.guioteca.com/curiosidades/el-timbre-un-invento-util-y-con-curiosos-origenes/>.
- Evolución de la red de comunicación móvil, del 1G al 5G | VIU. [en línea], [sin fecha]. [Consulta: 5 agosto 2018]. Disponible en: <https://www.universidadviu.com/evolucion-la-red-comunicacion-movil-del-1g-al-5g/>.
- EXES, [sin fecha]. Cursos oracle, cursos java, master oracle, master java, formación en informática. [en línea]. [Consulta: 6 agosto 2018]. Disponible en: http://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java_inicial_4_1.html.
- Famatech. (2014). Advanced IP Scanner. Retrieved from <http://www.advanced-ipscanner.com/es/>

FFmpeg. En: Page Version ID: 108250392, *Wikipedia, la enciclopedia libre* [en línea], 2018. [Consulta: 9 junio 2018]. Disponible en:
<https://es.wikipedia.org/w/index.php?title=FFmpeg&oldid=108250392>.

La historia del timbre. *Geniolandia* [en línea], [sin fecha]. [Consulta: 9 junio 2018].
Disponible en: <https://www.geniolandia.com/13125389/la-historia-del-timbre>.

PREBLE, A., [sin fecha]. Cómo funcionan los timbres inalámbricos. [en línea].
[Consulta: 9 junio 2018]. Disponible en:
http://www.ehowenespanol.com/funcionan-timbres-inalambricos-info_207715/.

Questions - find out about common ordering questions and answers - Proteus Design Suite. [en línea], [sin fecha]. [Consulta: 6 agosto 2018]. Disponible en:
<https://www.labcenter.com/faq/>.

SIM800C. [en línea], [sin fecha]. [Consulta: 13 agosto 2018]. Disponible en:
<http://simcomm2m.com/En/module/detail.aspx?id=74>.

Timbre eléctrico. En: Page Version ID: 108198235, *Wikipedia, la enciclopedia libre* [en línea], 2018. [Consulta: 9 junio 2018]. Disponible en:
https://es.wikipedia.org/w/index.php?title=Timbre_el%C3%A9ctrico&oldid=108198235.

ZÚNIGA, +César, 2015. Raspberry PI 2 | Mini PC que soporta Windows 10. *Proyecto Byte* [en línea]. [Consulta: 5 agosto 2018]. Disponible en:
<https://www.proyectobyte.com/android/raspberry-pi-2-mini-pc-que-soporta-windows-10>.

ZÚNIGA, +César, 2018. Raspberry Pi 3 Model B+ | Novedades. *Proyecto Byte* [en línea]. [Consulta: 5 agosto 2018]. Disponible en:
<https://www.proyectobyte.com/android/raspberry-pi-3-model-b-plus-novedades>.

ANEXOS

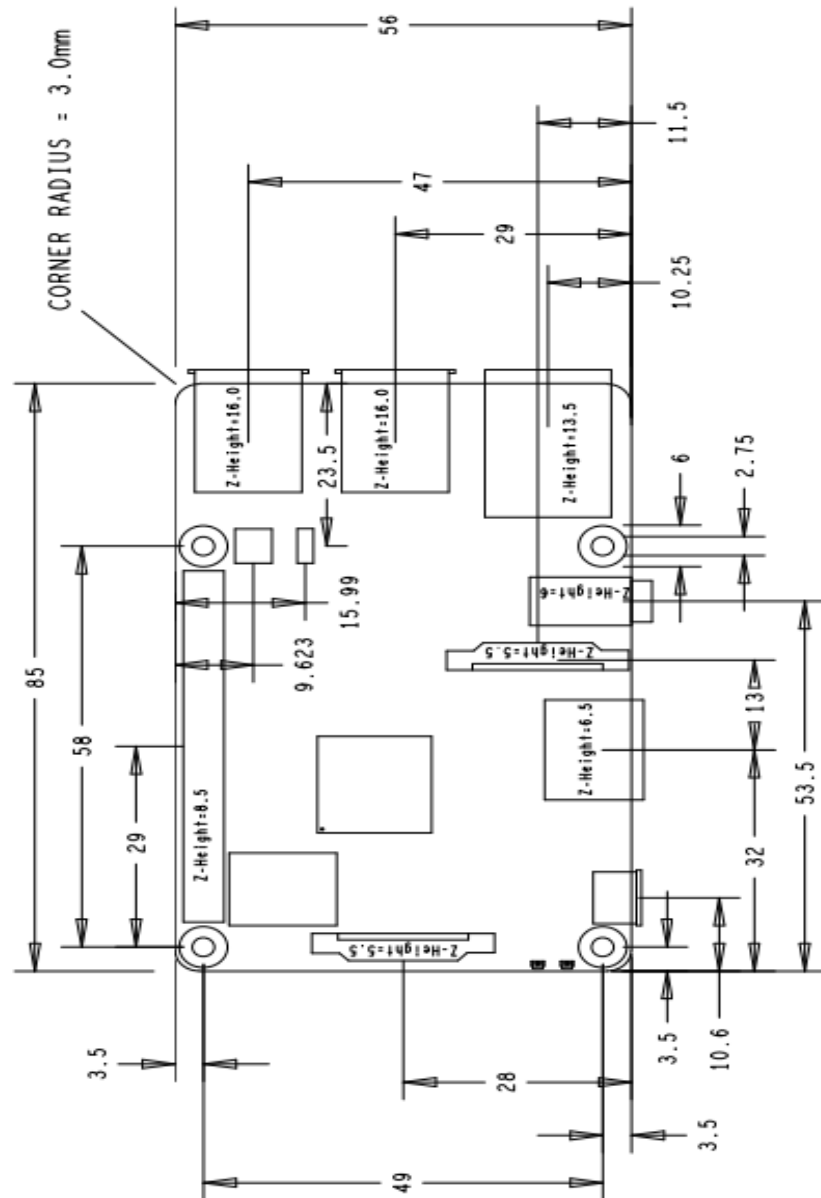
A. RASPBERRY PI 3B+

Specifications

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4 GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	<ul style="list-style-type: none">■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	<ul style="list-style-type: none">■ 1 × full size HDMI■ MIPI DSI display port■ MIPI CSI camera port■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	<ul style="list-style-type: none">■ 5V/2.5A DC via micro USB connector■ 5V DC via GPIO header■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50°C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.

A.1. Especificaciones Físicas

Physical specifications



A.2. Diferencias entre los tipos de Raspberry

	Raspberry Pi 3 Model B	Raspberry Pi 2 Model B	Model B+	Model A+	Model A	CMDK
						
Processor Chipset	Broadcom BCM2837 64Bit ARMv7 Quad Core Processor powered Single Board Computer running at 1250MHz	Broadcom BCM2836 32bit ARMv7 Quad Core Processor powered Single Board Computer running at 900MHz	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor	Broadcom BCM2835 32bit ARMv6 SoC full HD multimedia applications processor
GPU	Videocore IV	Videocore IV	Videocore IV	Videocore IV	Videocore IV	Videocore IV
Processor Speed	QUAD Core @1250 MHz	QUAD Core @900 MHz	Single Core @700 MHz	Single Core @700 MHz	Single Core @700 MHz	Single Core @700 MHz
RAM	1GB SDRAM @ 400 MHz	1GB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz	256 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz
Storage	MicroSD	MicroSD	MicroSD	MicroSD	SDCard	4GB eMMC
USB 2.0	4x USB Ports	4x USB Ports	4x USB Ports	1x USB Port	1x USB Port	1x USB Port
Power Draw / voltage	2.5A @ 5V	1.8A @ 5V	1.8A @ 5V	1.8A @ 5V	1.2A @ 5V	1.8A @ 5V
GPIO	40 pin	40 pin	40 pin	40 pin	26 pin	120 pin
Ethernet Port	Yes	Yes	Yes	No	No	No
Wi-Fi	Built in	No	No	No	No	No
Bluetooth LE	Built in	No	No	No	No	No

B. MÓDULO SIM800L

MÓDULO GSM SIM800L

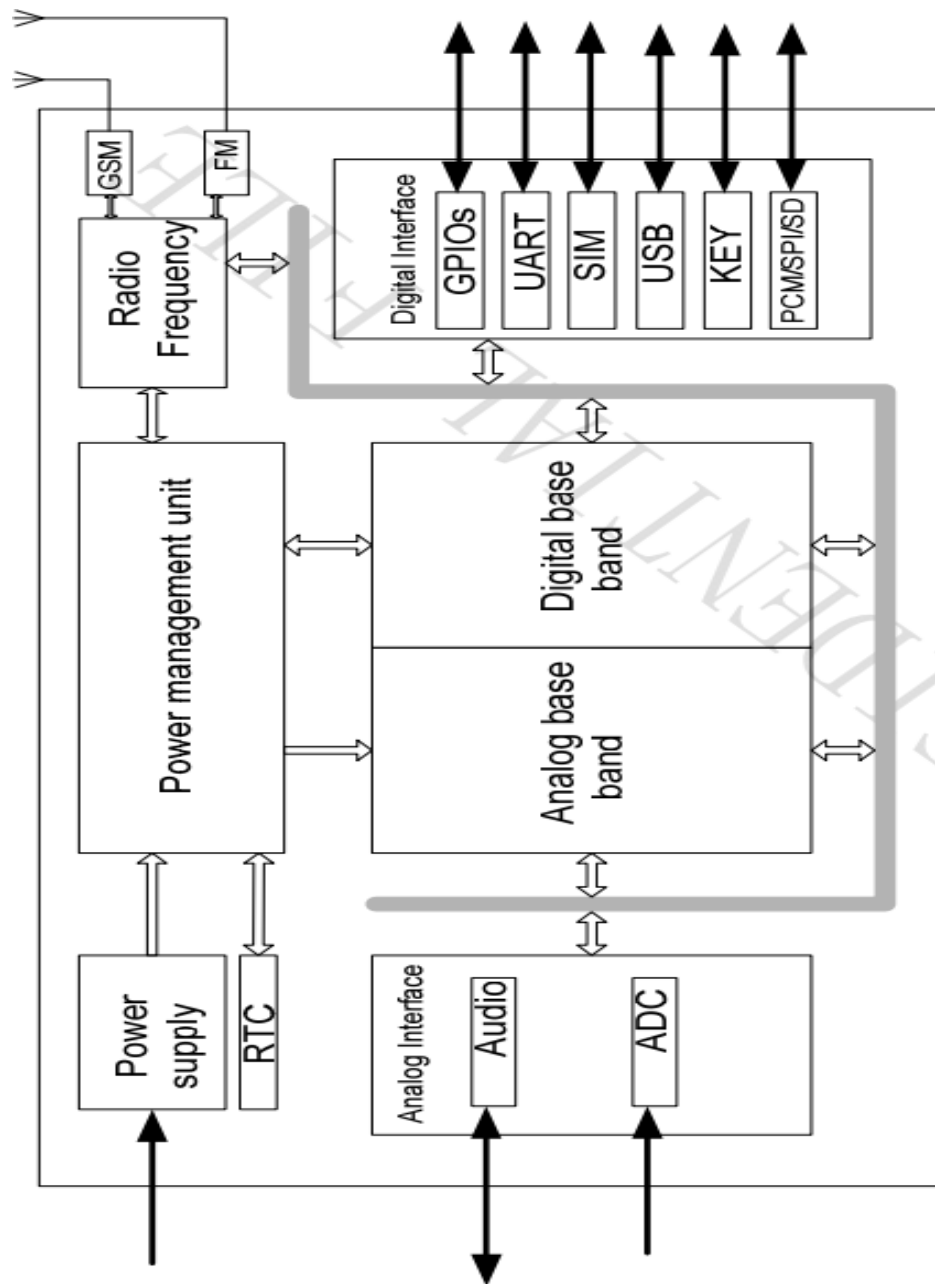
Este módulo de telefonía celular que te permite añadir voz, texto, datos y SMS a tu proyecto.

Se requiere un microcontrolador para controlarlo, usamos un Arduino pero cualquier microcontrolador 3-5V con una UART puede enviar y recibir comandos a través de los pines RX/TX.

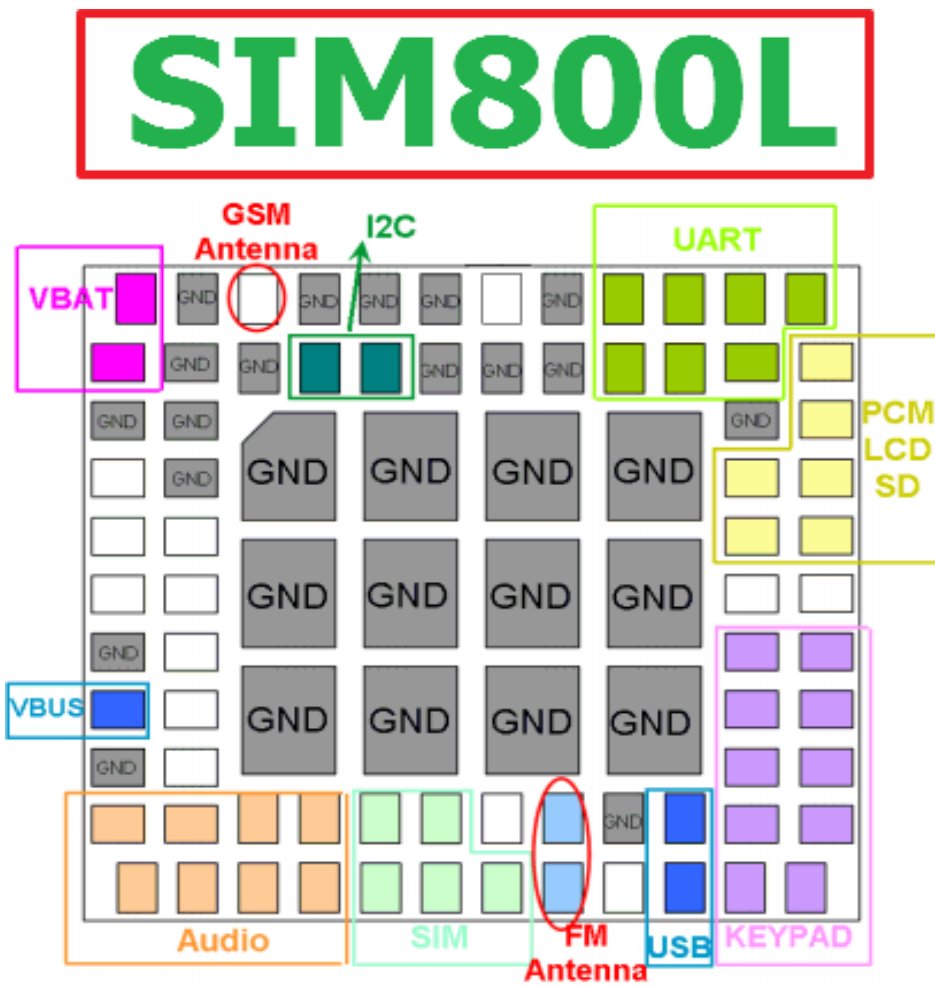
ESPECIFICACIONES TÉCNICAS

- Voltaje de Operación: 3.4V - 4.4V DC
- Nivel Lógico de 3V a 5V
- Consumo de corriente (máx): 500 mA
- Consumo de corriente (modo de reposo): 0.7 mA
- Interfaz: Serial UART
- Quad-band 850/900/1800/1900MHz – se conectan a cualquier red mundial GSM con cualquier SIM 2G
- Trabaja solo con tecnología 2G
- Hacer y recibir llamadas de voz usando un auricular o un altavoz de 8Ω externo + micrófono electret.
- Enviar y recibir mensajes SMS
- Enviar y recibir datos GPRS (TCP/IP, HTTP, etc)
- Escanear y recibir emisiones de radio FM
- Controlado por Comandos AT
- Interfaz de comandos AT con detección “automática” de velocidad de transmisión
- Soporta A-GPS
- Datos GPRS:
 - Velocidad máxima de transmisión 85.6 Kbps
 - Protocolo TCP/IP en chip
 - Codificación: CS-1, CS-2, CS-3 y CS-4
 - Soporta USSD
- Soporta reloj en tiempo real (RTC)
- Velocidades de transmisión serial desde 1200bps hasta 115 200 bps
- Tamaño de la SIM: Micro SIM

B.1. Diagrama de funciones



B.2. Pines del Sim800L



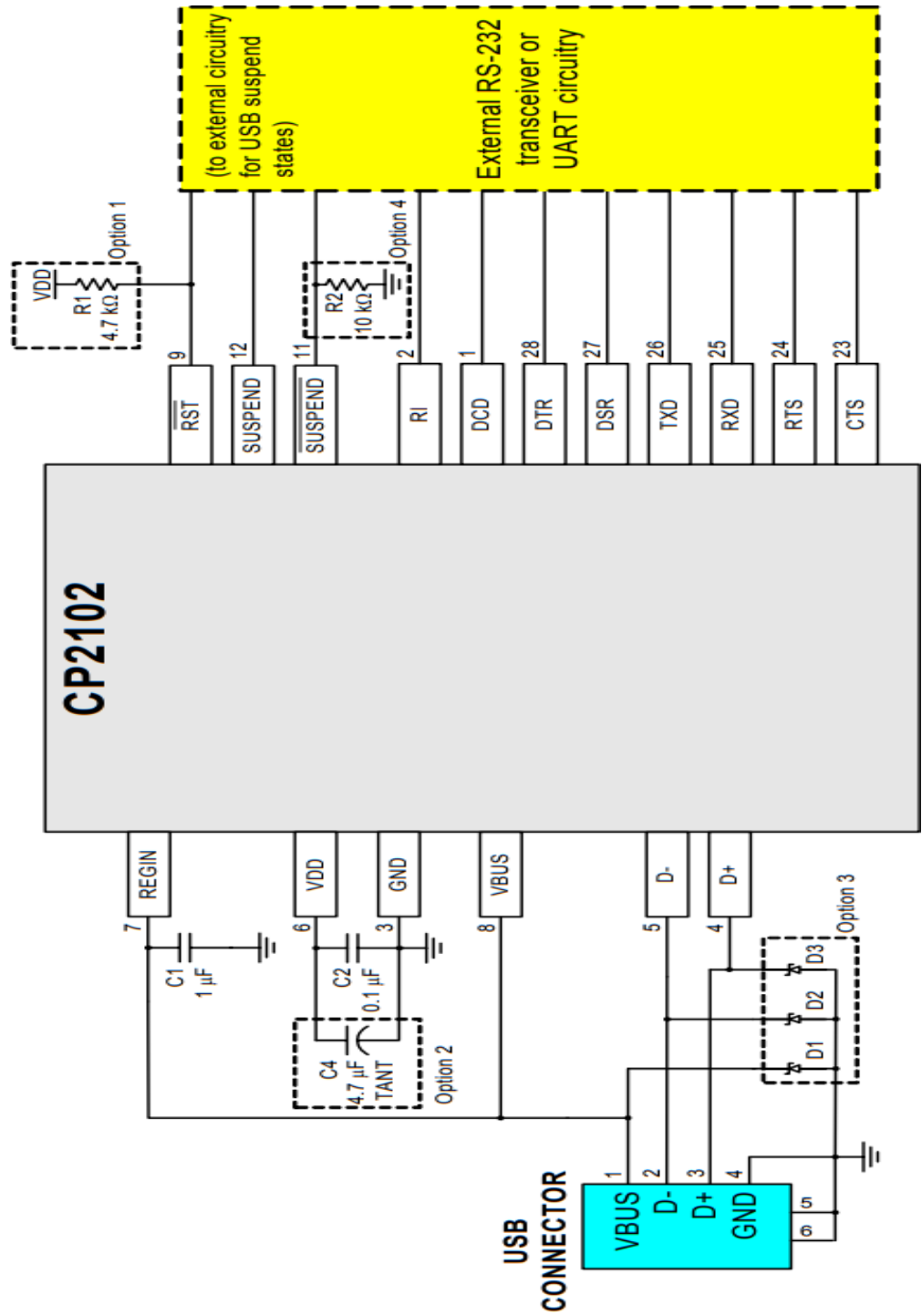
C. CONVERTOR DE USB A TTL (CP2102)

El CP2102 es un controlador de puente USB-UART altamente integrado que proporciona una solución simple para actualizar RS-. Diseña a USB utilizando un mínimo de componentes y espacio PCB. El CP2102 incluye un USB 2.0 de máxima velocidad. Controlador de funciones, transceptor USB, oscilador, EEPROM y bus de datos serie asíncrono (UART) con total.

C.1. Pines CP2102

Name	Pin #	Type	Description
V _{DD}	6	Power In	3.0–3.6 V Power Supply Voltage Input.
		Power Out	3.3 V Voltage Regulator Output. See <PDF link>Section 10.
GND	3		Ground
RST	9	D I/O	Device Reset. Open-drain output of internal POR or V _{DD} monitor. An external source can initiate a system reset by driving this pin low for at least 15 μ s.
REGIN	7	Power In	5 V Regulator Input. This pin is the input to the on-chip voltage regulator.
VBUS	8	D In	VBUS Sense Input. This pin should be connected to the VBUS signal of a USB network. A 5 V signal on this pin indicates a USB network connection.
D+	4	D I/O	USB D+
D–	5	D I/O	USB D–
TXD	26	D Out	Asynchronous data output (UART Transmit)
RXD	25	D In	Asynchronous data input (UART Receive)
CTS	23*	D In	Clear To Send control input (active low)
RTS	24*	D Out	Ready to Send control output (active low)
DSR	27*	D in	Data Set Ready control input (active low)
DTR	28*	D Out	Data Terminal Ready control output (active low)
DCD	1*	D In	Data Carrier Detect control input (active low)
RI	2*	D In	Ring Indicator control input (active low)
SUSPEND	12*	D Out	This pin is driven high when the CP2102 enters the USB suspend state.
$\overline{\text{SUSPEND}}$	11*	D Out	This pin is driven low when the CP2102 enters the USB suspend state.
NC	10, 13–22		These pins should be left unconnected or tied to V _{DD} .

C.2. Diagrama de conexión



D. BÚFER (74HC125)

Sirve para cambiar el umbral de los valores lógicos de comunicación de salida, donde el 3.7v pasa a ser 3.3v (1 lógico) y q el voltaje de 0 volts (0 lógico).

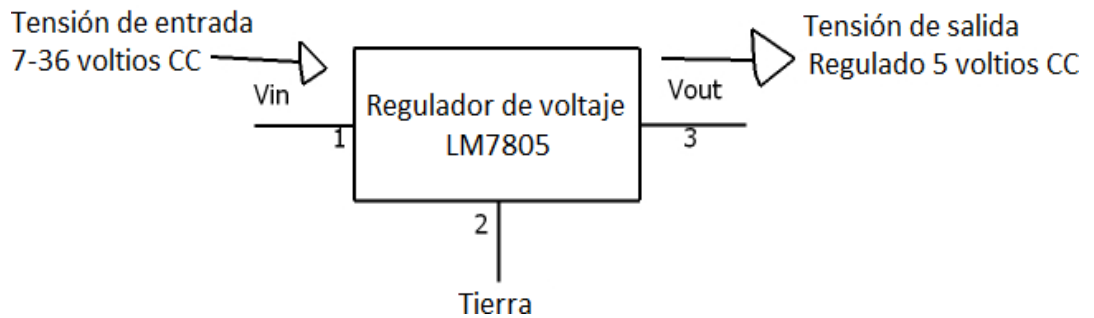
AC CHARACTERISTICS FOR 74HC

GND = 0 V; $t_r = t_f = 6$ ns; $C_L = 50$ pF

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} (V)	WAVEFORMS
		+25			−40 to +85		−40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} / t _{PLH}	propagation delay nA to nY		30 11 9	100 20 17		125 25 21		150 30 26	ns	2.0 4.5 6.0	Fig.6	
t _{PZH} / t _{PZL}	3-state output enable time nOE to nY		41 15 12	125 25 21		155 31 26		190 38 32	ns	2.0 4.5 6.0	Fig.7	
t _{PHZ} / t _{PLZ}	3-state output disable time nOE to nY		41 15 12	125 25 21		155 31 26		190 38 32	ns	2.0 4.5 6.0	Fig.7	
t _{THL} / t _{TLH}	output transition time		14 5 4	60 12 10		75 15 13		90 18 15	ns	2.0 4.5 6.0	Fig.6	

E. REGULADOR (LM7805)

Un regulador de voltaje LM7805 es un regulador de voltaje que genera +5 voltios.



E.1. Diferencias de voltajes de entradas y salidas de los distintos tipos de reguladores

	7833	7805	7806	7808	7809	7810	7812	7815	7818	7824
V_{out}	3,3V	5V	6V	8V	9V	10V	12V	15V	18V	24V
$V_{in} - V_{out}$		2,2V - 30V					2,5V - 23V	2,6V - 20V		
Temperatura de operación	0 - 125 °C	0 - 125 °C		0 - 125 °C	0 - 125 °C			0 - 125 °C		
I_{max} de salida	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A

E.2. Características eléctricas

Electrical Characteristics (LM7805)

Refer to the test circuits. $-40^{\circ}\text{C} < T_J < 125^{\circ}\text{C}$, $I_O = 500\text{mA}$, $V_I = 10\text{V}$, $C_I = 0.1\mu\text{F}$, unless otherwise specified.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_O	Output Voltage	$T_J = +25^{\circ}\text{C}$	4.8	5.0	5.2	V
		$5\text{mA} \leq I_O \leq 1\text{A}$, $P_O \leq 15\text{W}$, $V_I = 7\text{V to } 20\text{V}$	4.75	5.0	5.25	
Regline	Line Regulation ⁽¹⁾	$T_J = +25^{\circ}\text{C}$ $V_O = 7\text{V to } 25\text{V}$	–	4.0	100	mV
		$V_I = 8\text{V to } 12\text{V}$	–	1.6	50.0	
Regload	Load Regulation ⁽¹⁾	$T_J = +25^{\circ}\text{C}$ $I_O = 5\text{mA to } 1.5\text{A}$	–	9.0	100	mV
		$I_O = 250\text{mA to } 750\text{mA}$	–	4.0	50.0	
I_Q	Quiescent Current	$T_J = +25^{\circ}\text{C}$	–	5.0	8.0	mA
ΔI_Q	Quiescent Current Change	$I_O = 5\text{mA to } 1\text{A}$	–	0.03	0.5	mA
		$V_I = 7\text{V to } 25\text{V}$	–	0.3	1.3	
$\Delta V_O/\Delta T$	Output Voltage Drift ⁽²⁾	$I_O = 5\text{mA}$	–	-0.8	–	mV/ $^{\circ}\text{C}$
V_N	Output Noise Voltage	$f = 10\text{Hz to } 100\text{kHz}$, $T_A = +25^{\circ}\text{C}$	–	42.0	–	$\mu\text{V}/V_O$
RR	Ripple Rejection ⁽²⁾	$f = 120\text{Hz}$, $V_O = 8\text{V to } 18\text{V}$	62.0	73.0	–	dB
V_{DROP}	Dropout Voltage	$I_O = 1\text{A}$, $T_J = +25^{\circ}\text{C}$	–	2.0	–	V
r_O	Output Resistance ⁽²⁾	$f = 1\text{kHz}$	–	15.0	–	m Ω
I_{SC}	Short Circuit Current	$V_I = 35\text{V}$, $T_A = +25^{\circ}\text{C}$	–	230	–	mA
I_{PK}	Peak Current ⁽²⁾	$T_J = +25^{\circ}\text{C}$	–	2.2	–	A

F. REGULADOR (LM317)

F.1. Características eléctricas

ELECTRICAL CHARACTERISTICS ($V_I - V_O = 5.0$ V; $I_O = 0.5$ A for D2T and T packages; $T_J = T_{low}$ to T_{high} [Note 1]; I_{max} and P_{max} [Note 2]; unless otherwise noted.)

Characteristics	Figure	Symbol	Min	Typ	Max	Unit
Line Regulation (Note 3), $T_A = +25^\circ\text{C}$, $3.0\text{ V} \leq V_I - V_O \leq 40\text{ V}$	1	Reg_{line}	—	0.01	0.04	%/V
Load Regulation (Note 3), $T_A = +25^\circ\text{C}$, $10\text{ mA} \leq I_O \leq I_{max}$ $V_O \leq 5.0\text{ V}$ $V_O \geq 5.0\text{ V}$	2	Reg_{load}	— —	5.0 0.1	25 0.5	mV % V_O
Thermal Regulation, $T_A = +25^\circ\text{C}$ (Note 6), 20 ms Pulse		Reg_{therm}	—	0.03	0.07	% V_O /W
Adjustment Pin Current	3	I_{Adj}	—	50	100	μA
Adjustment Pin Current Change, $2.5\text{ V} \leq V_I - V_O \leq 40\text{ V}$, $10\text{ mA} \leq I_L \leq I_{max}$, $P_D \leq P_{max}$	1, 2	ΔI_{Adj}	—	0.2	5.0	μA
Reference Voltage, $3.0\text{ V} \leq V_I - V_O \leq 40\text{ V}$, $10\text{ mA} \leq I_O \leq I_{max}$, $P_D \leq P_{max}$	3	V_{ref}	1.2	1.25	1.3	V
Line Regulation (Note 3), $3.0\text{ V} \leq V_I - V_O \leq 40\text{ V}$	1	Reg_{line}	—	0.02	0.07	% V
Load Regulation (Note 3), $10\text{ mA} \leq I_O \leq I_{max}$ $V_O \leq 5.0\text{ V}$ $V_O \geq 5.0\text{ V}$	2	Reg_{load}	— —	20 0.3	70 1.5	mV % V_O
Temperature Stability ($T_{low} \leq T_J \leq T_{high}$)	3	T_S	—	0.7	—	% V_O
Minimum Load Current to Maintain Regulation ($V_I - V_O = 40\text{ V}$)	3	I_{Lmin}	—	3.5	10	mA
Maximum Output Current $V_I - V_O \leq 15\text{ V}$, $P_D \leq P_{max}$, T Package $V_I - V_O = 40\text{ V}$, $P_D \leq P_{max}$, $T_A = +25^\circ\text{C}$, T Package	3	I_{max}	1.5 0.15	2.2 0.4	— —	A
RMS Noise, % of V_O , $T_A = +25^\circ\text{C}$, $10\text{ Hz} \leq f \leq 10\text{ kHz}$		N	—	0.003	—	% V_O
Ripple Rejection, $V_O = 10\text{ V}$, $f = 120\text{ Hz}$ (Note 4) Without C_{Adj} $C_{Adj} = 10\text{ }\mu\text{F}$	4	RR	— 66	65 80	— —	dB
Long-Term Stability, $T_J = T_{high}$ (Note 5), $T_A = +25^\circ\text{C}$ for Endpoint Measurements	3	S	—	0.3	1.0	%/1.0 k Hrs.
Thermal Resistance Junction to Case, T Package		$R_{\theta JC}$	—	5.0	—	$^\circ\text{C/W}$

G. REGULADOR (LM1117)

LM1117 es una serie de Reguladores Bajo-Salida lineales con una deserción de 3.3V a 800mA de corriente de carga.

G.1. Características Físicas

Categorías	Circuitos integrados PMIC - reguladores de voltaje, lineal
Fabricante	Texas Instruments
Serie	LM1117-3.3
Empaquetado ?	Cinta cortada (CT) ?
Estado de la pieza	Activo
Configuración de salidas	Positivo
Tipo de salida	Fija
Cantidad de reguladores	1
Voltaje - Entrada (máx.)	15 V
Voltaje - salida (mín/fijo)	3.3 V
Voltaje - Salida (máx.)	-
Caída del voltaje (máx.)	1.2 V a 800 mA
Salida de corriente	800 mA
Corriente - Reposo (Iq)	5mA
Corriente - Alimentación (máx.)	10mA
PSRR	75 dB (120 Hz)
Características de control	-
Características de protección	Sobrecorriente, sobretemperatura
Temperatura de operación	-40 °C ~ 125 °C
Tipo de montaje	Montaje en superficie
Paquete / Caja (carcasa)	TO-261-4, TO-261AA
Paquete del dispositivo del proveedor	SOT-223-4
Número de pieza base	LM1117

H. CÓDIGO EN PYTHON

H.1. Código en Python para el envío de SMS

```
import serial
import time

class TextMessage:
    def __init__(self, recipient="", message=""):
        self.recipient = recipient
        self.content = message

    def setRecipient(self, number):
        self.recipient = number

    def setContent(self, message):
        self.content = message

    def connectPhone(self):
        self.ser = serial.Serial('/dev/ttyAMA0', 115200, timeout=5)
        time.sleep(1)

    def sendMessage(self):
        self.ser.write('ATZr')
        time.sleep(1)
        self.ser.write('AT+CMGF=1r')
        time.sleep(1)
        self.ser.write('AT+CMGS="" + self.recipient + ""r')
        time.sleep(1)
        self.ser.write(self.content + "r")
        time.sleep(1)
        self.ser.write(chr(26))
        time.sleep(1)

    def disconnectPhone(self):
        self.ser.close()
```

H.1. Código en Python para las llamadas

```
import RPi.GPIO as GPIO
import subprocess
import serial
import time, sys
import datetime

def setup():
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
    #puerto serial
    ser= serial.Serial('/dev/ttyUSB0', 115200, timeout=5)
    #inicializar
    setup()
    ser.write('AT+CMGF=1\r') # modo texto
    time.sleep(1)
    ser.write('AT+CMGDA="DEL ALL"\r') # borrar all SMS
    time.sleep(1)
    sudo modprobe bcm2835-v4l2
    v4l2-ctl -V

try:
    while True:
        reply = ser.read(ser.inWaiting())
        if GPIO.input(23):
            print "esperando tocar timbre o llamada"
            time.sleep(1)
            # if ser.read() == RING
            #ser.write('AT+CMGS="+51970292787"\r')
            raspivid -o - -t 0 -hf -w 600 -h 300 -fps 24 /cvlc -vvv
            stream:///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8080}'
            :demux=h264

        else:
            print "se tocó timbre"
            #ser.write('AT+CMGS="+51970292787"\r')
            #time.sleep(1)
            ser.write('tocan timbre')
            time.sleep(1)
            ser.write((chr(26)))
            time.sleep(.500)
            #ser.write('AT+CMGDA="DEL ALL"\r')
            time.sleep(.500)
            ser.read(ser.inWaiting()) # Clear buffer
            time.sleep(.500)
    finally:
        GPIO.cleanup()
```