



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

FACULTAD DE INGENIERÍAS

CARRERA DE INGENIERÍA ELECTRÓNICA

**PROYECTO TÉCNICO PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
INGENIERO ELECTRÓNICO**

TEMA:

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TRANSFERENCIAS DE DATOS A TRAVÉS DE LA RED ELÉCTRICA DE BAJA TENSIÓN CON LA INTERFAZ LABVIEW-ARDUINO EMPLEANDO LA TECNOLOGÍA POWER LINE COMMUNICATIONS (PLC)”

AUTORES:

DAVID PONCE LEÓN

SERGIO SANTILLÁN AGUIRRE

TUTOR:

ING. GABRIEL GARCÍA

GUAYAQUIL, ECUADOR, 2016

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

Nosotros, DAVID PONCE LEÓN y SERGIO SANTILLÁN AGUIRRE, alumnos de la Universidad Politécnica Salesiana, declaramos que los conceptos, análisis desarrollados y conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Guayaquil, 2016

David Ponce L.

C.I. 092639287-9

Sergio Santillán A.

C.I. 092338997-7

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Yo, DAVID JORGE PONCE LEÓN, con documento de identificación N° 0926392879, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado titulado: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TRANSFERENCIAS DE DATOS A TRAVÉS DE LA RED ELÉCTRICA DE BAJA TENSIÓN CON LA INTERFAZ LABVIEW-ARDUINO EMPLEANDO LA TECNOLOGÍA POWER LINE COMMUNICATIONS (PLC)”, mismo que ha sido desarrollado para optar por el título de: INGENIERO ELECTRONICO, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

.....

Nombre: David Ponce León

Cédula: 0926392879

Guayaquil, Septiembre 2016

Yo, SERGIO STEVEN SANTILLÁN AGUIRRE, con documento de identificación N° 0923389977, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado titulado: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TRANSFERENCIAS DE DATOS A TRAVÉS DE LA RED ELÉCTRICA DE BAJA TENSION CON LA INTERFAZ LABVIEW-ARDUINO EMPLEANDO LA TECNOLOGÍA POWER LINE COMMUNICATIONS (PLC)”, mismo que ha sido desarrollado para optar por el título de: INGENIERO ELECTRONICO, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

.....

Nombre: Sergio Santillán Aguirre

Cédula: 0923389977

Guayaquil, Septiembre 2016

DEDICATORIA

Dedico este trabajo principalmente a Dios, por haberme dado salud y permitirme haber alcanzado una meta más en mi vida profesional, por la fuerza que me da en momentos difíciles que me enseñan a levantarme para ser cada día mejor y por todos los triunfos y logros alcanzados a lo largo de mi vida.

A mi madre Marlene León y a mi padre Digno Ponce quienes me brindaron todo su apoyo y comprensión durante toda mi vida estudiantil, a mis hermanos quienes estuvieron hay para darme sus consejos y respaldo incondicional en cada momento a lo largo de mi trayectoria profesional, a todos mis familiares y amigos en general los cuales fueron pilar fundamentales para cumplir mis objetivos.

David Ponce L.

DEDICATORIA

A Dios, por haberme dado salud para llegar a esta altura de mi vida y cumplir mis metas a lo largo de mi vida estudiantil, y también por haberme dado la fuerza, sabiduría e inteligencia para la realización de este proyecto de titulación universitaria.

A mis Padres, Sergio Santillán y Petita Aguirre, por ser pilar fundamental en mi vida y en toda mi carrera estudiantil, por todos los valores inculcados desde mi infancia, por todo el cariño, apoyo, confianza y paciencia que me dieron durante la ejecución de este proyecto y a mis hermanas y hermano que también son parte importante en este logro que consigo.

Una mención especial para mi abuelito Marcos Aguirre, quien desde el cielo me inspiro y dio la fuerza para terminar esta etapa de mi vida.

Sergio Santillán A.

AGRADECIMIENTO

A Dios por dotarnos de sabiduría día a día para cumplir eficazmente los objetivos. A nuestros padres, por ser pilar fundamentales en nuestras vidas y enseñanza. Por los consejos durante la vida estudiantil.

Agradecemos a la Universidad Politécnica Salesiana, a los docentes de cada una de las materias que cursamos, por haber formado parte fundamental en nuestro desarrollo académico y profesional.

Agradecemos a su vez al ingeniero Gabriel García V. como tutor de nuestro proyecto técnico, quien nos guio, oriento, apoyo y corrigió en el desarrollo del proyecto a través de sus conocimientos, profesionalismo y motivación.

David Ponce L.

Sergio Santillán A.

RESUMEN

AÑO	ALUMNOS	TUTOR DEL PROYECTO	TEMA DEL PROYECTO
2016	David Ponce León Sergio Santillán Aguirre	Ing. Gabriel García	“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TRANSFERENCIAS DE DATOS A TRAVÉS DE LA RED ELÉCTRICA DE BAJA TENSIÓN CON LA INTERFAZ LABVIEW-ARDUINO EMPLEANDO LA TECNOLOGÍA POWER LINE COMMUNICATIONS (PLC)”

El presente proyecto técnico de titulación: “Diseño e Implementación de un Sistema de transferencias de datos a través de la Red eléctrica de Baja tensión con la interfaz LabVIEW-Arduino empleando la Tecnología Power Line Communications (PLC)”, se lo realizó para el laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil entre los años 2015 – 2016. Los medios de transmisión normalmente utilizados en la ejecución de prácticas o talleres propuestos en las distintas materias de especialización o seminarios en los laboratorios de clases de la universidad son cable coaxial, fibra óptica y cables UTP.

La propuesta del proyecto de titulación está basada en la implementación de un nuevo sistema de transferencia mediante tecnología Power Line Communications (PLC), Comunicación por la Red Eléctrica, que se fundamenta en la transmisión de información en diferentes puntos utilizando la red eléctrica ya existente de baja tensión en el Laboratorio de Telecomunicaciones, con la finalidad de contribuir con nuevas opciones de comunicación, de envío y recepción de información para aplicaciones tecnológicas en la formación académica hacia los estudiantes de las diversas materias técnicas de la carrera. Las características de esta tecnología permiten desplegar redes de control y acceso de manera eficiente en corto tiempo, además de evitar rediseños de infraestructura. Ofreciendo flexibilidad y movilidad a lo largo de todo el tendido de la red eléctrica.

Palabras claves: tecnología, red eléctrica, transferencia de datos, ahorrar recursos.

ABSTRACT

YEAR	STUDENTS	PROJECT TUTOR	PROJECT TITLE
2016	David Ponce León Sergio Santillán Aguirre	Ing. Gabriel García	"DESIGN AND IMPLEMENTATION OF A DATA TRANSFER SYSTEM THROUGH THE LOW VOLTAGE POWER GRID WITH LABVIEW-ARDUINO INTERFACE USING POWER LINE COMMUNICATIONS (PLC) TECHNOLOGY"

This technical titling project: "Design and Implementation of a System Data transfers through the Electric Low Voltage Network with the interface LabVIEW-Arduino using technology Power Line Communications (PLC)," it made for the laboratory Telecommunications of the Salesian Polytechnical University based Guayaquil in the year 2015-2016. The transmission means normally used in the implementation of practical or workshops proposed in different subjects of specialization or seminars in laboratories college classes are coaxial cable, fiber optic and UTP cables.

The project proposal graduation is based on the implementation of a new transfer system using technology Power Line Communications (PLC), Communication by the Power Grid, which is based on the transmission of information at different points using the existing power grid low voltage in the Telecommunications Laboratory, in order to contribute new communication options, sending and receiving of information technology applications in academic training to students of various technical matters of the career. The characteristics of this technology allow deploy access control networks and efficiently in a short time, and avoid redesigns infrastructure. Offering flexibility and mobility throughout the laying of the mains.

Keywords: technology, electricity grid, data transfer, save resources.

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA	I
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR	II
DEDICATORIA	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
RESUMEN.....	VII
ABSTRACT	VIII
ÍNDICE GENERAL.....	IX
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS	XIII
ÍNDICE DE ANEXOS.....	XIV
INTRODUCCIÓN	1
1. EL PROBLEMA.....	2
1.1 Antecedentes.	2
1.2 Importancia y Alcance.....	3
1.3 Delimitación.	3
2. OBJETIVOS	5
2.1 Objetivo General.	5
2.2 Objetivos Específicos.	5
3. FUNDAMENTOS TEÓRICOS.....	6
3.1 Tecnología PLC, Power Line Communications.....	6
3.1.1 Principio de trabajo.	7
3.1.2 Ventajas.....	7
3.1.3 Desventajas	7

3.1.4	Aspectos Técnicos y Legales de las Telecomunicaciones en Ecuador.	8
3.1.5	Proyectos Pilotos y Organismos Reguladores Internacionales.	10
3.2	Adaptador Nano PowerLine AV500 - TL-PA4010.....	11
3.2.1	Características:	11
3.3	Extensor WiFi Powerline	12
3.4	LABVIEW.....	14
3.4.1	¿Qué es LabVIEW?.....	14
3.4.2	Aplicaciones.....	15
3.5	ARDUINO.....	15
3.5.1	Lenguaje de programación.....	16
3.6	ARDUINO MEGA 2560.....	17
3.7	ARDUINO ETHERNET SHIELD.....	18
3.8	APP INVENTOR	19
3.9	MÓDULO BLUETOOTH ARDUINO HC-06	21
3.10	SENSORES	22
3.10.1	Sensor Ultrasónico HC-SR04.	22
3.10.2	Sensor de Temperatura LM35.....	23
3.11	SERVOMOTOR	23
3.11.1	Especificaciones:.....	24
4.	MARCO METODOLÓGICO.....	25
4.1	METODOLOGÍA	25
4.1.1	Método Inductivo	25
4.1.2	Método Deductivo.....	25
4.2	ANÁLISIS DE RESULTADOS	26
4.2.1	Práctica 1	26

4.2.2	Práctica 2.....	31
4.2.3	Práctica 3.....	36
4.2.4	Práctica 4.....	40
4.2.5	Práctica 5.....	45
4.2.6	Práctica 6.....	48
5.	IMPACTO	53
6.	FUNCIONALIDAD	54
	CONCLUSIONES	56
	RECOMENDACIONES.....	56
	REFERENCIAS.....	57
	ANEXOS	59

ÍNDICE DE FIGURAS

Figura 1: Red doméstica unificada a PowerLine.	8
Figura 2: Adaptador Nano Powerline AV500 - TL-PA4010.....	12
Figura 3: Extensor Powerline WiFi AV500 a 300 Mbps TL-WPA4220	13
Figura 4: Extensión de cobertura WiFi a través de PowerLine.	13
Figura 5: Botón de clonado Wi-Fi en el Extensor TL-WPA4220.	14
Figura 6: Logo LabVIEW 2013.....	15
Figura 7: Logo oficial de Arduino.	16
Figura 8: Tarjeta Arduino MEGA 2560.	17
Figura 9: Tarjeta Arduino Ethernet Shield.	19
Figura 10: Logo de la App Inventor	20
Figura 11: Módulo Bluetooth HC-06.	21
Figura 12: Sensor Ultrasónico HC-SR04.	22
Figura 13: Pines de conexión del sensor de temperatura LM35	23
Figura 14: Servomotor HS-311.....	24

ÍNDICE DE TABLAS

Tabla 1	12
Tabla 2	17
Tabla 3	27
Tabla 4	28
Tabla 5	32
Tabla 6	33
Tabla 7	37
Tabla 8	41
Tabla 9	46
Tabla 10	49

ÍNDICE DE ANEXOS

Anexo 1: Programación en Arduino de la Practica 1.	59
Anexo 2: Programación en Arduino de la Practica 2.	61
Anexo 3: Programación en Arduino de la Practica 3.	63
Anexo 4: Programación en Arduino de la Practica 4.	67
Anexo 5: Programación en Arduino de la Practica 5.	71
Anexo 6: Programación en Arduino de la Practica 6.	74
Anexo 7: Diseño de la interfaz, en App Inventor 2, para control de la Practica 6.	77
Anexo 8. Adaptador Powerline TL-PA4010, Características y Especificaciones técnicas.	79
Anexo 9: Extensor Powerline TL-WPA4220, características y especificaciones técnicas.	81
Anexo 10: Datasheet de la tarjeta Arduino MEGA 2560.....	83
Anexo 11: Datasheet del Módulo Bluetooth Arduino HC-06.....	84
Anexo 12: Datasheet del Sensor Ultrasónico HC-SR04.....	87
Anexo 13: Datasheet del Sensor de Temperatura LM-35.....	89
Anexo 14: Especificaciones del Servomotor Hitec HS-311.....	91
Anexo 15: Elaboración de módulos didácticos para las diferentes prácticas del Proyecto.	92
Anexo 16: GLOSARIO.....	98
Anexo 17: PRESUPUESTO.....	100

INTRODUCCIÓN

En el área de las comunicaciones, hay la necesidad de encontrar diversas alternativas para la transmisión de datos. Últimamente se ha desarrollado una tecnología que permite ahorrar y evitar rediseñar nuevas estructuras de cableado estructurado. La tecnología PLC (Power Line Communications, Comunicación por la Red Eléctrica) está basada en utilizar líneas de distribuciones eléctricas convencionales como puntos de acceso a la Red.

Implementar y dar a conocer la tecnología PLC en la Universidad Politécnica Salesiana es la meta de este proyecto, para que los estudiantes conozcan nuevas alternativas de transferencias de datos. En la práctica se aplican los conocimientos aprendidos en las diversas asignaturas de ámbito industrial y de telecomunicaciones.

Para el desarrollo del presente trabajo de titulación con esta tecnología, se emplea el software de programación e interfaz gráfica LabVIEW, el software libre App Inventor y la plataforma Arduino, para la debida adquisición y transferencia de datos a través de la red eléctrica de baja tensión, basados en el problema por cual se implementa este proyecto técnico en la Universidad Politécnica Salesiana, su importancia y el alcance.

Una fundamentación teórica breve sobre los equipos TP-LINK y las interfaces para controlar los dispositivos que conforman el proyecto técnico. Se dispone de equipos de comunicación PowerLine básicos, las tarjetas Arduino Mega 2560 y Ethernet Shield que son las que permiten la comunicación con los diferentes equipos a controlar tales como indicadores, sensores y servomotor.

Detallar las prácticas realizadas y mostrar los resultados obtenidos por medio de la implementación de la tecnología Power Line Communications, como sustento del proyecto técnico propuesto.

1. EL PROBLEMA

1.1 Antecedentes.

En nuestro país, se han realizados investigaciones, por parte de estudiantes universitarios, sobre la calidad y garantía de los recursos que se necesitarían para la implementación de la tecnología PLC, tales como el suministro eléctrico desde las centrales eléctricas hasta el cableado eléctrico residencial para un posible desarrollo de esta tecnología en el país. En el año 2007, en la ciudad de Quito se realizaron concursos para motivar a los estudiantes e investigadores, al desarrollo de esta tecnología ya que la Empresa Eléctrica de Quito (EEQ) daba las facilidades para su estudio y desarrollo de esta tecnología.

Dentro del proceso de las comunicaciones, existen diversos medios de transmisión guiados tales como cable coaxial, cables de fibra óptica y cables UTP, que en su mayoría son utilizados actualmente para el acceso a internet en oficinas, edificios, hogares y aulas de clases.

Actualmente en la Universidad Politécnica Salesiana de Guayaquil existen varios sistemas para las transferencias de datos a través de los diferentes medios de transmisión antes mencionados, gran parte de estos sistemas se han ido desarrollando por los estudiantes mediante proyectos técnicos e investigaciones, con el propósito de que se realicen pruebas en los laboratorios y para la interconexión entre las diferentes áreas de la universidad. Pero no poseen sistemas que utilicen la red eléctrica convencional como medio para la transmisión de datos entre dos puntos o más puntos de la misma red.

La propuesta del proyecto se basa en la implementación de un sistema de transferencia usando la tecnología PLC, la cual consiste en la transmisión de información en diferentes puntos de la misma red de distribución eléctrica de baja tensión ya existente en el laboratorio de Telecomunicaciones, con la finalidad de enriquecer el aprendizaje de los estudiantes en las diversas materias técnicas de la carrera, facilitando la interconexión entre diferentes puntos de manera más rápida de más sencilla y economizando recursos en la infraestructura.

1.2 Importancia y Alcance.

La importancia del presente trabajo de titulación es dar a conocer el uso de la tecnología PLC, la cual ayudará a efectuar de una manera más sencilla conexiones entre dos puntos sin la necesidad de ubicar nuevas instalaciones de cableado estructurado, procediendo a utilizar solo la red de distribución eléctrica en baja tensión (tomacorrientes) existente en nuestros hogares, aulas de clases, laboratorios y edificios que servirán como medio de transmisión guiado para envíos y recepción de datos.

El proyecto tiene como alcance desarrollar un sistema de transferencia de datos que permite controlar y monitorear dispositivos tales como sensores de temperatura, sensores ultrasónicos y servomotor, controlado por la interfaz LabVIEW-Arduino; del mismo modo se controlaran bombillos a través de un servidor web con una dirección IP específica y aplicaciones creadas en sistemas Android realizada por los autores, aplicación creada con software libre y de fácil instalación , utilizando la tecnología PLC la misma que permite mejor facilidad de ubicar dos equipos solo utilizando la red eléctrica de baja tensión.

Los principales o directos beneficiarios son los estudiantes de la carrera de ingeniería electrónica que se encuentran cursando las materias antes mencionadas, en las que le será de gran utilidad este proyecto. Como también los docentes, ya que por medio de este proyecto pueden realizar clase más didácticas con sus alumnos.

1.3 Delimitación.

El proyecto técnico de titulación está dirigido para el laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana, Sede Guayaquil, el cual será desarrollado en el año 2016. El sistema de transferencia de datos estará constituido de los siguientes elementos y equipos de comunicación, el kit de comunicación PowerLine Adaptador Nano TL-PA4010 y Extensor TL-WPA4220, para el enlace se usarán las tarjetas Arduino Mega 2560 y Ethernet Shield W5100, para las distintas practicas utilizaremos sensores de proximidad Ultrasónicos, de temperatura LM35, servomotor HS-311, como indicadores luces pilotos, módulo Bluetooth HC-06 de Arduino, módulos de relés de 5V, bombillas

120VAC, tomacorrientes convencionales. El software a utilizar es LabVIEW versión 2013 y el respectivo software del programador Arduino.

Se realizarán un total seis prácticas diferentes en nuestro sistema de transferencia de datos en el Laboratorio de Telecomunicación para lograr la comunicación y control de datos mediante la red eléctrica de baja tensión 120VAC usando la interfaz LabVIEW-Arduino, servidor web con una dirección IP específica y aplicaciones con sistemas Android utilizando la tecnología PLC (Power Line Communications).

2. OBJETIVOS

2.1 Objetivo General.

Diseñar e implementar un sistema de transferencia de datos por medio de la red eléctrica, de baja tensión, con la interfaz LabVIEW-Arduino usando tecnología Power Line Communications (PLC), en el laboratorio de Telecomunicaciones de la Universidad Politécnica Salesiana, Sede Guayaquil.

2.2 Objetivos Específicos.

Realizar prototipos de pruebas (módulos) con la interfaz LabVIEW-Arduino para verificar resultados con los dispositivos utilizados.

Utilizar las principales características de la tecnología Power Line Communications como transporte de datos entre las interfaces en nuestro sistema de módulos que nos permitirá mostrar el envío y recepción de datos a través de la Red eléctrica de Baja tensión.

Desarrollar tareas en el software Arduino capaz de cumplir el funcionamiento entre las plataformas y los componentes que actúen en el sistema.

Implementar y explicar de modo práctico-experimental la comunicación entre software de simulación LabVIEW (interfaz gráfica del usuario) y la plataforma Arduino.

Realizar una práctica demostrativa con el software libre App Inventor 2, aplicación que funciona con sistema Android.

Fortalecer por medio de los módulos el desarrollo académico de materias, tales como Sensores, Comunicaciones, Electrónica digital, Instrumentación.

3. FUNDAMENTOS TEÓRICOS

3.1 Tecnología PLC, Power Line Communications.

La tecnología PLC, Power Line Communications, ha sido creada para lograr transmitir voz y datos a través de la red eléctrica de un hogar, oficina o institución educativa. Mediante esta tecnología se busca también aprovechar la infraestructura o cableado ya existente para ahorrar recursos económicos.

Antes de todo, es factible detallar las diferencias entre las tres distintas redes eléctricas que son de alta, media y baja tensión. La red eléctrica de alta tensión es la energía distribuida desde las centrales eléctricas, ya que la energía que generan son del orden los cientos de kilovoltios, que es desde donde es transportada hasta las distintas regiones del país y/o industrias. En cada región del país o industria es muy probable que exista centros de transformación y subestación de distribución, por medio de los cuales se logra convertir la energía de alta tensión a valores inferiores de tensión y así se genera una segunda red, la cual es la red eléctrica de media tensión, que genera tensión con valores entre 15 y 20 kilovoltios. La última transformación que se produce y que es necesaria para el suministro a los domicilios, es la que se da en los centros de transformación localizados o instalados en las ciudades o bajo tierra y que es de 220V/110V el cual es el suministro eléctrico de baja tensión.

Ante el constante aumento de la demanda por conexiones a Internet, sobre todo en banda ancha, las empresas proveedoras de conexión tienen a su disposición una variedad de tecnologías que les permiten ofrecer el servicio de conexión a los clientes o abonados. Una de estas tecnologías es PowerLine Communications o PLC por sus siglas, la cual permite sacar provecho a una red tan masiva, que prácticamente está presente en todos los hogares: la red eléctrica. De esta manera, esta tecnología permite a las compañías eléctricas sacar provecho a un medio que no fue considerado para la transmisión de datos, pudiendo desplegarse como una tecnología de acceso a internet o como complementación a otras tecnologías de acceso ya desplegadas. (Riffo, 2009)

3.1.1 Principio de trabajo.

Con la tecnología PLC se puede transmitir datos a través de la red eléctrica, y por lo tanto se puede extender a una red de área local existente o compartir una conexión a Internet a través de los enchufes eléctricos con la instalación de unidades específicas.

El principio del PLC consiste en la superposición de una señal de alta frecuencia también llamada onda corta (de 1,6 a 30MHz) con bajos niveles de energía sobre la señal de la red eléctrica de 60 Hz (Ecuador). Esta segunda señal se transmite a través de la infraestructura de la red eléctrica y se puede recibir y decodificar de forma remota. Así, la señal PLC es recibida por cualquier receptor PLC que se encuentra en la misma red eléctrica. Un acoplador integrado en la entrada del equipo PLC receptor elimina las componentes de baja frecuencia antes de que la señal sea tratada. (Serna, 2011)

3.1.2 Ventajas

- Utiliza la infraestructura ya existente, es decir el cableado eléctrico, por lo que no es necesario ningún tipo de obra adicional.
- Cualquier enchufe en una casa es suficiente para estar conectado.
- Posibilidad de crear redes de datos domesticas utilizando el cableado existente.
- Proporciona una conexión permanente 24 horas al día.
- Por medio de microfiltros se evitan las posibles interferencias generadas por los electrodomésticos.
- Con el tiempo los costes se abaratarán. (victorgarcia.org)

3.1.3 Desventajas

- La producción de los equipos necesarios es todavía escasa.
- Ausencia de estándares tecnológicos para la interoperabilidad de equipos.
- La red eléctrica no fue diseñada para transmitir datos por lo que experimenta frecuentes caídas y la calidad de las llamadas telefónicas no es completamente satisfactoria.
- Interferencias durante la navegación a través de la web.
- Falta de seguridad.
- Oposición de las compañías telefónicas. (victorgarcia.org)

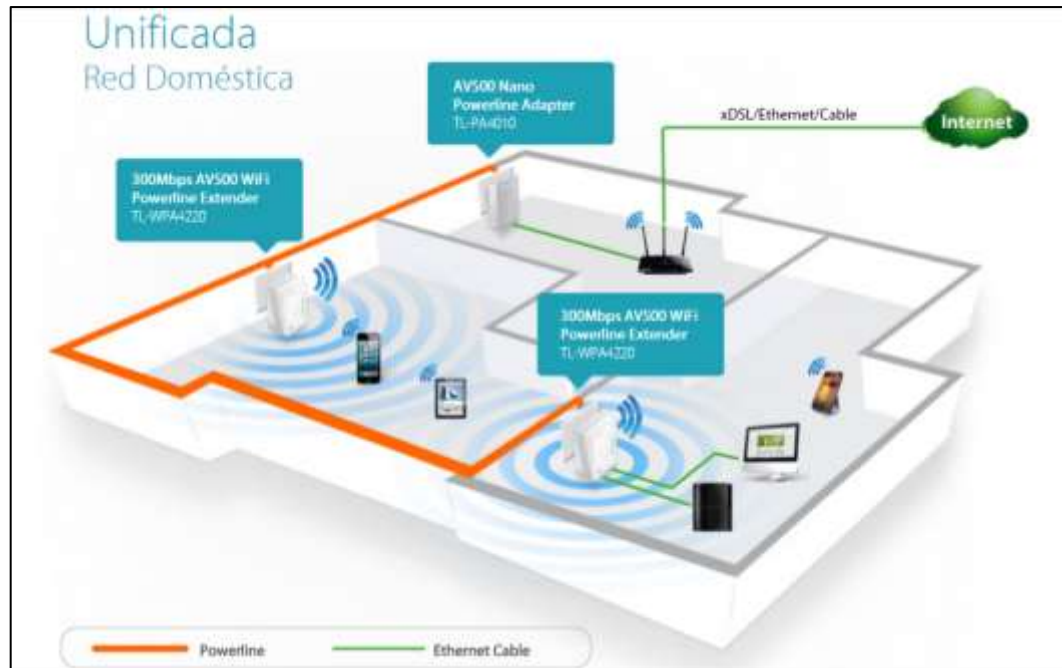


Figura 1: Red doméstica unificada a PowerLine.

Fuente: (TP-LINK, TP-LINK Technologies Co., Ltd., 2014), recuperado de:

<http://www.tp-link.es>

3.1.4 Aspectos Técnicos y Legales de las Telecomunicaciones en Ecuador.

El Ecuador cuenta con una estructura en lo que se refiere a los organismos regulatorios de las telecomunicaciones, los cuales están a cargo de vigilar, controlar el sector y cumplir funciones específicas buscando el desarrollo del país. Entre ellos tenemos, en orden de función:

- Ministerio de Telecomunicaciones y de la Sociedad de la Información (**MINTEL**)
- Consejo Nacional de Telecomunicaciones (**CONATEL**)
- Secretaría Nacional de Telecomunicaciones (**SENATEL**)
- Superintendencia de Telecomunicaciones (**SUPERTEL**)

El sector eléctrico Ecuatoriano está conformado por las siguientes instituciones:

- Ministerio de Electricidad y Energías Renovables (**MEER**)
- Consejo Nacional de Electricidad (**CONELEC**)
- Centro Nacional de Control de Energía (**CENACE**)

Estas instituciones mencionadas son las que están a cargo de dirigir, organizar y suministrar la generación, transmisión, distribución y consumo de la energía eléctrica a nivel nacional. Sistema que debe tener una alta calidad de servicios antes de implementar PLC.

En Ecuador no hay una normativa técnica para el uso de tecnología PLC, sin embargo tampoco se limita su implementación.

Las leyes y reglamentos en los cuales se analizó si existen impedimentos para una implementación de tecnología PLC son los siguientes:

- Ley Especial de Telecomunicaciones
- Ley Reformatoria a la Ley Especial de Telecomunicaciones
- Ley para la Transformación Económica del Ecuador
- Reglamento General a la Ley Especial de Telecomunicaciones Reformada
- Reglamento de concesiones de los Servicios de Telecomunicaciones
- Reglamento para la prestación de servicios finales de Telecomunicaciones a través de terminales de telecomunicaciones de uso Público.
- Reglamento para Homologación de Equipos de Telecomunicaciones
- Reglamento del fondo de Telecomunicaciones en Áreas Rurales

Luego de indagar y analizar los instrumentos legales mencionados, no se observa impedimento, cabe mencionar que el Estado regula servicios, más no tecnologías, es decir que mientras se cumpla los requerimientos legales es posible brindar servicios mediante un sistema con tecnología PLC, sin embargo se hace necesario una normativa técnica que permita conocer requisitos mínimos para acceder a servicios de calidad sin afectar a terceros.

3.1.5 Proyectos Pilotos y Organismos Reguladores Internacionales.

Son varios los países que han mostrado interés por el desarrollo de esta tecnología desde hace ya varios años, con el objetivo de evolucionar y garantizar la transmisión de datos a través de la red eléctrica. De los cuales citamos a algunos de ellos:

- PlcForum
- Asociación Universal de Línea de poder (UPA).
- HomePlug.
- Unión del Consejo de Líneas Eléctricas (UPLC)
- Alianza Europea de Investigación PLC (OPERA)
- PLC-J (PLC-Japón)
- Asociación de Empresas Propietarias de Infraestructura y Sistemas Privados de Telecomunicaciones (APTEL)

Los siguientes son organismos que han estudiado y dictado normas para el correcto funcionamiento de los sistemas PLC:

Entre los reguladores internacionales se encuentran:

- Unión Internacional de Telecomunicaciones (UIT)
- Instituto de Ingenieros Eléctricos y Electrónicos (IEEE)

Entre los Organismos de normalización europeo están:

- Instituto Europeo de Estándares Técnicos (ETSI)
- Comité Europeo para la Estandarización Electrotécnica (CENELEC).

Entre las organizaciones reguladoras en Estados Unidos están:

- Comisión Federal de Comunicaciones (FCC)
- Administración Nacional de Telecomunicaciones e Información (NTIA)

A nivel latinoamericano está:

- Comisión Interamericana de Telecomunicaciones (CITEL)

3.2 Adaptador Nano PowerLine AV500 - TL-PA4010.

El Adaptador Nano PowerLine de TP-LINK, TL-PA4010, convierte su Power Line existente en una red de alta velocidad sin necesidad de cables nuevos o de taladrar. No requiere configuración, simplemente conecte sus adaptadores a los tomacorriente y podrá establecer una infraestructura de red en instantes.

Con la tecnología avanzada de HomePlug AV, el TL-PA4010 puede proporcionar a los usuarios una velocidad de transferencia de datos estable y de alta velocidad de hasta 500Mbps en una extensión lineal de hasta 300 metros. Junto con su encriptación AES integrada, el TL-PA4010 es una gran elección para ser la solución para conectar todos los dispositivos compatibles con la red de un hogar completo desde computadoras y consolas de videojuegos.

El TL-PA4010, es parte de una nueva generación de adaptadores Powerline, con su práctico diseño y sofisticado modo de ahorro de energía consume menos comparado con los adaptadores de Powerline existentes. Cuando no hay transmisión o recepción de datos durante cierto periodo de tiempo, cambia de manera automática de su modo “en Funcionamiento” al modo de “Ahorro de Energía” que reduce su consumo de energía hasta en un 85%. (TP-LINK, TP-LINK Technologies Co., Ltd., 2014)

3.2.1 Características:

- En cumplimiento con el estándar HomePlug AV, la velocidad de transferencia de datos de alta velocidad de hasta 500Mbps, ideal para el streaming de video en alta definición o videos en 3D video y juegos en línea.
- Diseño en miniatura, más pequeño que la mayoría de los adaptadores Powerline en el mercado, se adapta indiscretamente en cualquier tomacorriente.
- Filtro de alimentación para un mejor desempeño de comunicación Power Line.
- No requiere cables nuevos, utiliza el cableado eléctrico existente.
- Rango de hasta 300 metros en el circuito eléctrico del hogar para un mayor desempeño a través de paredes o pisos.

Tabla 1

Especificaciones del Adaptador Nano Powerline AV500 - TL-PA4010.

Características del software	
Encriptación:	128-bits AES Encryption
Tecnología de modulación:	OFDM
Requisitos del Sistema:	Windows 8/7/Vista/XP/2000, Mac, Linux
Temperatura de operación:	0~40 °C
Temperatura de almacenamiento:	-40~70 °C

Fuente: <http://www.tp-link.ec>



Figura 2: Adaptador Nano Powerline AV500 - TL-PA4010

Fuente: <http://www.tp-link.ec>

3.3 Extensor WiFi Powerline

El TL-WPA4220 de TP-LINK lleva su conexión a Internet a cualquier área de la casa o edificio a través de la instalación eléctrica existente. Dispone de un botón de clonado Wi-Fi que permite una súper extensión de cobertura, copiando automáticamente el SSID y la contraseña de su router. De esta manera, el TL-WPA4220 simplifica su configuración Wi-Fi y le permite movilidad sin cortes dentro de su red doméstica.

Con tecnología HomePlug AV, proporciona tasas de transferencia de datos estables y de alta velocidad hasta 500Mbps sobre una línea de hasta 300 metros de distancia. Constituye una gran elección como solución doméstica integral para conectar todos los dispositivos de red, desde ordenadores y consolas de juegos hasta decodificadores para IPTV,

impresoras y unidades de almacenamiento NAS. (TP-LINK, TP-LINK Technologies Co., Ltd., 2015)



Figura 3: Extensor Powerline WiFi AV500 a 300 Mbps TL-WPA4220

Fuente: <http://www.tp-link.ec>

Con velocidades inalámbricas de hasta 300Mbps, podrá ampliar sin esfuerzo su red y WiFi a través del Power Line existente y retransmitir una señal inalámbrica para compartir con todos sus dispositivos WiFi en otras zonas de su casa u oficina.

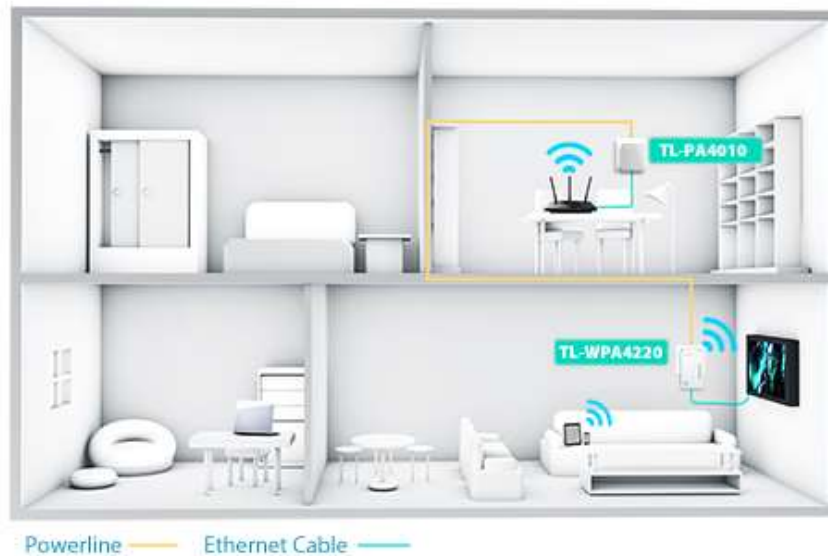


Figura 4: Extensión de cobertura WiFi a través de PowerLine.

Fuente: <http://www.tp-link.es>

Después de pulsar el botón, puede ubicar el adaptador Power Line allá donde desee acceso cableado o inalámbrico y disfrutará de Networking sin cortes por toda la casa u oficina. (TP-LINK, TP-LINK Technologies Co., Ltd., 2015)



Figura 5: Botón de clonado Wi-Fi en el Extensor TL-WPA4220.

Fuente: http://www.tp-link.es/products/details/cat-18_TL-WPA4220.html

3.4 LABVIEW

3.4.1 ¿Qué es LabVIEW?

LabVIEW es un entorno de desarrollo, con una sintaxis de programación gráfica que facilita visualizar, crear y codificar sistemas de ingeniería. Ayuda a ingenieros a convertir sus ideas en realidad, reducir tiempos de pruebas y ofrecer análisis de negocio basado en datos recolectados. Ofrece integración sin precedentes con todo el hardware de medidas, software legado existente e IP al aprovechar las últimas tecnologías de cómputo.

Desde desarrollar máquinas inteligentes hasta garantizar la calidad de los dispositivos conectados, LabVIEW ha sido la solución predilecta para crear, implementar y probar el Internet de las cosas por décadas. (Instruments, National Instruments Corporation, 2016).



Figura 6: Logo LabVIEW 2013.

Fuente: (Instruments, National Instruments Corporation, 2016) Recuperado de:

<http://fost.ws/other-soft/special-soft/10433-labview-2013-130-x64.html>

3.4.2 Aplicaciones

- **Adquirir Datos y Procesar Señales:** Mida cualquier sensor. Realice análisis y procesamiento de señales avanzados. Visualice datos en interfaces de usuario personalizadas. Registre datos y genere reportes.
- **Control de Instrumentos:** Automatice la colección de datos. Controle múltiples instrumentos. Analice y visualice señales.
- **Automatizar Sistemas de Pruebas y Validación:** Automatice las pruebas de validación y producción de su producto. Controle múltiples instrumentos. Analice y visualice resultados de pruebas con interfaces de usuario personalizadas.
- **Sistemas Embebidos de Monitoreo y Control:** Obtenga acceso a herramientas personalizadas para medicina, robótica y más.
- **Enseñanza Académica:** Utilice un enfoque práctico e interactivo de aprendizaje. Aumente el rendimiento de aplicaciones con procesamiento multinúcleo. (Instruments, National Instruments Corporation, 2016).

3.5 ARDUINO

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

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (bootloader) que corre en la placa. (BURUTEK, 2015).



Figura 7: Logo oficial de Arduino.

Fuente: <https://www.google.com.ec>

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador. Al ser open-hardware, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. (BURUTEK, 2015).

3.5.1 Lenguaje de programación.

El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el popular lenguaje de programación de alto nivel Processing. (BURUTEK, 2015).

3.6 ARDUINO MEGA 2560

El Arduino Mega es una placa electrónica basada en los ATmega2560. Lleva 54 Entradas /Salidas Digitales (14 pueden utilizarse para salidas PWM), 16 entradas analógicas, 4 UARTs (puertas seriales), un oscilador de 16MHz, una conexión USB, un conector de alimentación, un header ICSP y un botón de Reset. Contiene todo lo necesario para soporte del microcontrolador, simplemente como conectarlo a un ordenador con un cable USB, o alimentarlo con un adaptador de corriente AC a DC para empezar a utilizarlo. (S., 2013)

Tabla 2.

Características técnicas de tarjeta Arduino MEGA 2560

Micro Controlador	ATmega2560
Voltaje de Operación	5V
Voltaje (In) Recomendado	7-12V
Límite de Voltaje (In)	6-20V
Entradas/ Salidas Digitales	54 (14 como salidas PWM)
Entradas Análogas	16
Corriente por Pin In/Out	40 mA
Corriente Para el Pin 3.3V	50 mA
Memoria Flash	256 KB (Solo 8 KB son usados por la secuencia de arranque “bootloader”)
SRAM	8 KB (Dos bancos: 64 KB y 32KB)
EEPROM	4 KB
Velocidad de Reloj	16 MHz

Fuente: <http://pidelectronics.com/arduino-mega-2560-r3.html>



Figura 8: Tarjeta Arduino MEGA 2560.

Fuente: <http://tdrobotica.co/arduino-mega-2560/15.html>

3.7 ARDUINO ETHERNET SHIELD

La Arduino Ethernet Shield permite a una placa Arduino conectarse a internet. Está basada en el chip Ethernet Wiznet W5100, con funcionalidades de IP tanto para TCP como UDP.

Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para escribir programas que se conecten a internet usando la Shield. La Ethernet Shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino. (Interactivo, 2015)

Arduino Mega usa los pines digitales 50, 51 y 52 para comunicarse con el W5100 en la Ethernet Shield. Para deshabilitar el W5100 configura el pin 10 como una salida en HIGH.

La Shield provee un conector Ethernet estándar RJ45. El botón de Reset en la Shield resetea ambos, el W5100 y la placa Arduino.

La Shield contiene un número de leds para información:

- **PWR:** indica que la placa y la Shield están alimentadas
- **LINK:** indica la presencia de un enlace de red y parpadea cuando la Shield envía o recibe datos
- **FULLD:** indica que la conexión de red es full dúplex
- **100M:** indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
- **RX:** parpadea cuando la Shield recibe datos.
- **TX:** parpadea cuando la Shield envía datos.
- **COLL:** parpadea cuando se detectan colisiones en la red. (Interactivo, 2015)

Para usar la Ethernet Shield solo hay que montarla sobre la placa Arduino. Para cargar los sketches a la placa conectarla al ordenador mediante el cable USB como se hace normalmente. Una vez que el sketch ha sido cargado se puede desconectar la placa del ordenador y alimentarla desde una fuente externa.

Conectar la Ethernet Shield a un ordenador, a un switch o a un router utilizando un cable Ethernet standard (CAT5 ó CAT6 con conectores RJ45). (Interactivo, 2015)

El dispositivo será capaz de responder a una petición HTTP con su Ethernet Shield. Después de abrir un navegador web e ingresar la dirección IP asignada al Arduino Ethernet Shield, su Arduino responderá mostrando los valores del estado de su entrada analógica (A0) y su entrada digital (Pin 2). (Interactivo, 2015)



Figura 9: Tarjeta Arduino Ethernet Shield.

Fuente: <http://trobotica.co/shield-arduino-ethernet/52.html>

3.8 APP INVENTOR

App inventor es una aplicación de Google Labs para crear aplicaciones (valga la redundancia) de Android, estas apps pueden crearse desde cualquier smartphone con sistema operativo Android, además también puede ser utilizado desde una computadora que soporte el sistema operativo ya mencionado para PC.

Esta aplicación ha sido una de las grandes innovaciones de los últimos años, ya que ha permitido realizar sus propias aplicaciones con diferentes objetivos, ya sea para satisfacción y comodidad personal o incluso como una fuente de trabajo o negocio en Android Market. Gracias a App Inventor ahora es una realidad programar sus propias funcionalidades en su smartphone e incluso en su propia computadora gracias a Android para computadores. (Moviles, s.f.)



Figura 10: Logo de la App Inventor

Fuente: <http://apppmoviles.net/wp-content/uploads/app-inventor.jpg>

Para crear una aplicación con app inventor hay que realizar los siguientes pasos:

- El diseño de la aplicación, en la que se seleccionan los componentes para su aplicación.
- El editor de bloques, donde irás escogiendo los bloques que sean necesarios según la aplicación que tenga pensada hacer.

La aplicación aparecerá paso a paso en la pantalla del teléfono a medida que añada piezas a la misma. Cuando haya terminado, puede empaquetar la aplicación y producir una aplicación independiente para instalar. Si no tiene un celular Android, puede construir sus aplicaciones utilizando el emulador de Android, el software que se ejecuta en la computadora y se comporta como el teléfono.

El entorno de desarrollo de App Inventor es compatible con Mac y sistemas operativos de Windows, y varios modelos de celulares Android. Las aplicaciones creadas con App Inventor 2 se pueden instalar en cualquier celular Android.

Antes de poder utilizar App Inventor 2, es necesario configurar el ordenador e instalar los archivos de instalación de App Inventor en el equipo. (Núñez, s.f.)

3.9 MÓDULO BLUETOOTH ARDUINO HC-06

El HC-06 (Slave/Esclavo) es un dispositivo de fácil implementación y uso dentro de una sencilla aplicación para Android permite conectar inalámbricamente tus proyectos con un Smartphone, PC o Laptop. Es todo lo necesario para resolver sus conexiones Wireless.

Ventajas principales del módulo HC-06:

- Pequeño tamaño y sus buenas características de transmisión y recepción que le brindan un alcance muy amplio.
- Bajo consumo de corriente que posee tanto en funcionamiento, como en modo de espera, es decir, alimentado con energía, pero sin conexión o enlace a otro dispositivo.
- Una vez que ha realizado un enlace con otro dispositivo es capaz de recordarlo en su memoria y no solicita validación alguna, pero si se activa el pin (KEY) hacia la tensión de alimentación, esta información se elimina y el módulo HC-06 solicitará nuevamente la validación del enlace.
- Su voltaje de operación es: 3.3V / 5V.
- Su corriente de operación es: < 40 mA y en modo de espera: < 1mA.
- Posee un LED indicador de estado, cuando está energizado y no está conectado a otro dispositivo enciende en forma intermitente y al conectar y establecer vínculo con otro dispositivo el LED pasa a estar encendido permanentemente. El LED no se apaga en ninguna instancia del funcionamiento del HC-06. (NEOTEO, 2016)



Figura 11: Módulo Bluetooth HC-06.

Fuente: <http://www.naylampmechatronics.com/inalambrico/24-hc06-modulo-bluetooth.html>

3.10 SENSORES

3.10.1 Sensor Ultrasónico HC-SR04.

El HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. El HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio.



Figura 12: Sensor Ultrasónico HC-SR04.

Fuente: <http://www.geekfactory.mx/tienda/sensores/hc-sr04-sensor-de-distancia-ultrasonico/>

3.10.1.1 Características:

- Tensión de alimentación: 5 Vcc
- Frecuencia de trabajo: 40 KHz
- Rangos: 4.5 m (máx.) – 1.7 cm (mín.)
- Duración mínima del pulso de disparo (nivel TTL): 10 μ S.
- Duración del pulso eco de salida (nivel TTL): 100-25000 μ S.
- Tiempo mínimo de espera entre una medida y el inicio de otra: 20ms.
- Pines de conexión:
 - Vcc
 - Trig (Disparo del ultrasonido)
 - Echo (Recepción de ultrasonido)
 - Gnd. (Electronilab, s.f.)

3.10.2 Sensor de Temperatura LM35.

El sensor de temperatura LM35 es un integrado que posee un circuito de control, con el cual se genera un voltaje de salida proporcional a la temperatura. La temperatura medida es obtenida por su resistencia eléctrica.

El voltaje de salida es lineal con la temperatura, el cual por cada grado centígrado de la temperatura medida incrementa su valor a razón de 10mV.

El rango de medición de este sensor es de -55°C (-550mV) a 150°C (1500mV). Su precisión a temperatura ambiente es de $0,5^{\circ}\text{C}$. Además, a temperatura ambiente ofrece una precisión de $\pm 1.4^{\circ}\text{C}$.



Figura 13: Pines de conexión del sensor de temperatura LM35

Fuente: <http://www.luisllamas.es/2015/07/medir-temperatura-con-arduino-y-sensor-lm35/>

3.11 SERVOMOTOR

Un servomotor es un actuador mecánico, que está basado de un motor y un conjunto de engranajes, lo que permite multiplicar el torque o fuerza del sistema final, también consta de elementos de control para el monitoreo constante de la posición de un elemento mecánico que será el destino con el exterior. Al servomotor se le induce una acción electrónica para lograr obtener un control en el equipo mecánico final.

El HS-311 es el servo perfecto para aplicaciones de bajo coste, ya que el servo tiene muchas características que solo se encuentran en servos más caros. Una circuitería de calidad así como los engranes asegura la durabilidad del servo. Este servo puede operar a 180° cuando tiene un pulso de 600useg a 2400useg. (Labs, 2016)

3.11.1 Especificaciones:

- Velocidad: 0.19 seg/60° a 4.8V
- Torque: 42.00 oz.-in a 4.8V
- Velocidad: 0.15 seg/60° a 6.0V
- Torque: 51.00 oz.-in a 6.0V
- Largo: 1.60" (41mm)
- Ancho: 0.80" (20mm)
- Alto: 1.40" (37mm)
- Peso: 1.5oz (43g)



Figura 14: Servomotor HS-311.

Fuente: <http://www.robotshop.com/en/hitec-hs311-servo.html>

4. MARCO METODOLÓGICO

4.1 METODOLOGÍA

La metodología que se utiliza para el desarrollo de este trabajo de titulación, está conformada por el método inductivo y el método deductivo.

4.1.1 Método Inductivo

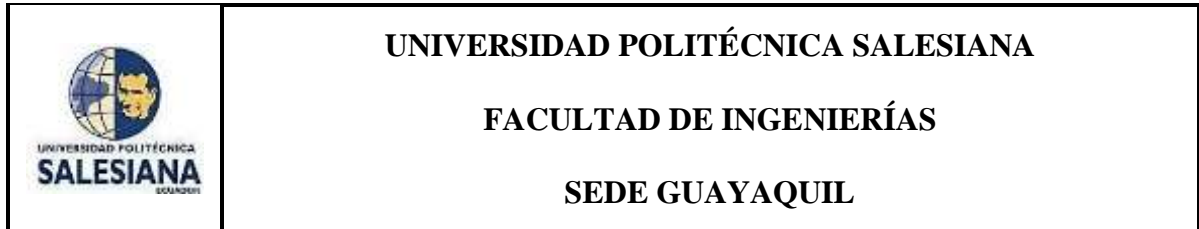
Se empleó este método ya que se refiere que va de las partes hacia el todo, es decir analiza los elementos del todo para llegar a un concepto, y nosotros en base a los conocimientos adquiridos a lo largo de la carrera universitaria fuimos capaces de diseñar, programar e implementar la transferencia de datos para poder lograr la aplicación de la tecnología PLC.

4.1.2 Método Deductivo

Por medio de este método se puede hacer un proceso sintético-analítico, y partimos desde el todo que es el sistema de transferencia de datos para llegar a las partes, es decir, a los resultados que se desean, los cuales los obtenemos aplicando los conocimientos académicos, técnicas de recopilación de información, con la asistencia de los software LabVIEW y Arduino los cuales son de adquisición de datos.

4.2 ANÁLISIS DE RESULTADOS

4.2.1 Práctica 1



CARRERA	PRÁCTICA N°	NOMBRE DE LA PRÁCTICA
Electrónica	1	Control para Activar y Desactivar Luces pilotos, mediante niveles de Temperatura a Través de la Interfaz LabVIEW-Arduino sobre comunicación PLC.

OBJETIVOS
<ul style="list-style-type: none">• Familiarizar al estudiante con la tecnología PLC.• Conocer las capacidades y alcances de los equipos PowerLine Communications.• Conocer al software LabVIEW como interface de la instrumentación virtual.• Lograr la comunicación entre los equipos para activar y desactivar las luces pilotos de acuerdo a la temperatura.

EQUIPOS A UTILIZAR	
<ul style="list-style-type: none">• Adaptador PowerLine TL-PA4010• Extensor PowerLine TL-WPA4220• Tarjeta Arduino Mega 2560• Tarjeta Arduino Ethernet Shield• Software LabVIEW y Arduino	<ul style="list-style-type: none">• Sensor de temperatura LM35• Luces piloto• Módulo de Relés de 12 Voltios• Laptop• Cables de conexión

INSTRUCCIONES
<ul style="list-style-type: none">• Diseñar, en el software LabVIEW, un sistema que indique los niveles de temperatura por medio de luces piloto.• Conectar los equipos Adaptador y Extensor PowerLine Communications, a tomacorrientes de 120V para su respectiva alimentación energética.• Programar las tarjetas Mega 2560 y Ethernet Shield con su respectivo software Arduino.• Declarar las variables necesarias para el enlace entre LabVIEW y Arduino.• Conectar, vía Ethernet, los equipos PLC con el módulo de las tarjetas Arduino.

- Conectar los módulos que intervienen en la ejecución de esta práctica.
- Ejecutar, primeramente, en el software LabVIEW la práctica a realizar, para verificar virtual y físicamente que las luces piloto están en correcto funcionamiento.
- Aplicar manualmente calor al sensor para mostrar la activación de las distintas luces piloto de acuerdo a los rangos de niveles de temperatura seteados para cada indicador y encendido del ventilador.

Tabla 3

Diagrama de conexiones de la Practica 1.

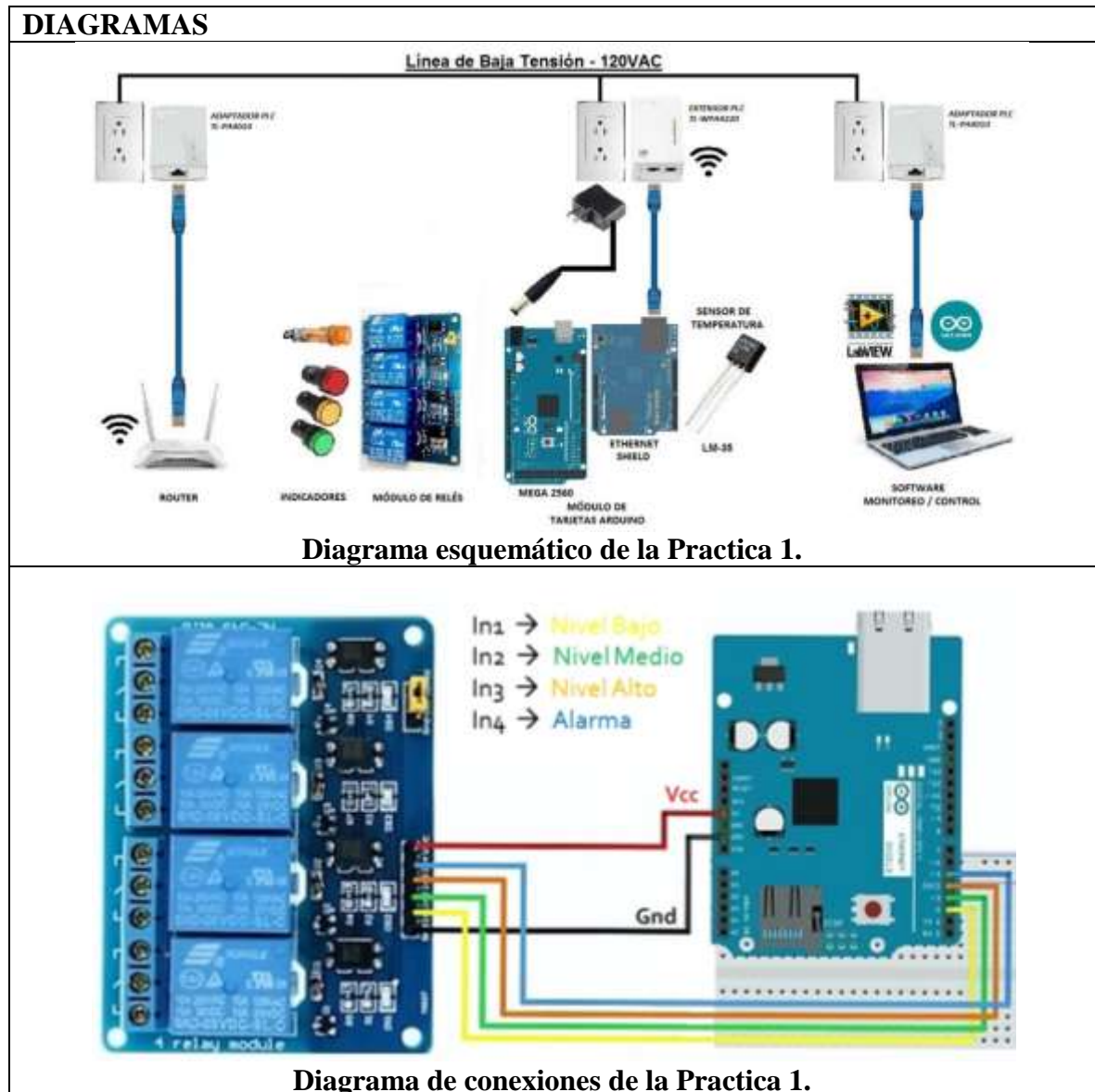


Tabla 4

Diseño de monitoreo y control en LabVIEW de la Practica 1.



Panel Frontal de la Práctica 1.

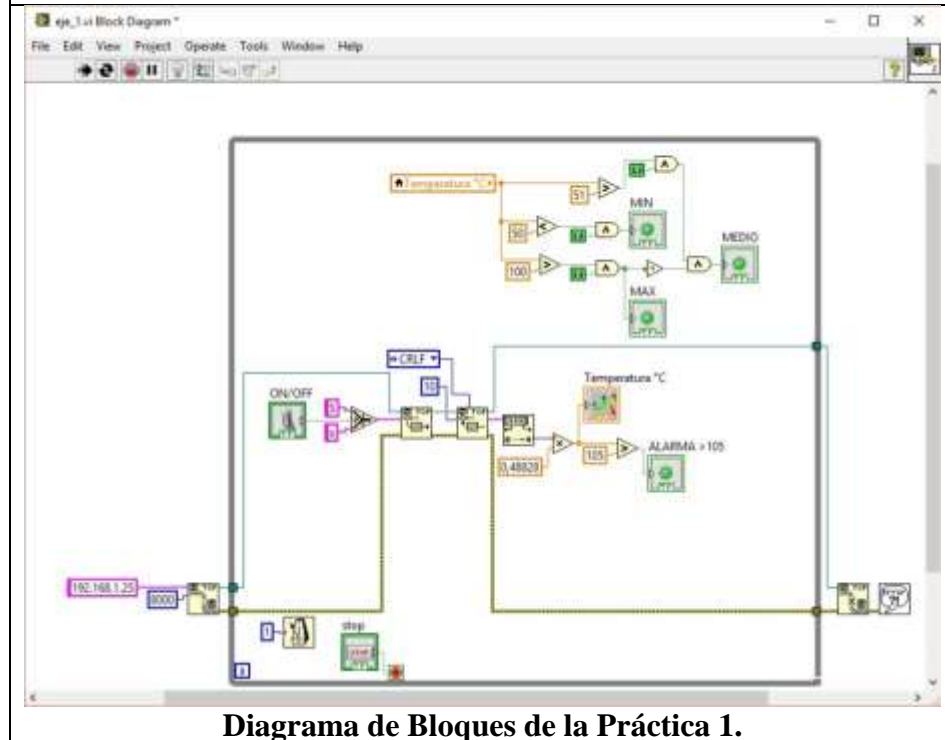


Diagrama de Bloques de la Práctica 1.

PROGRAMACIÓN ARDUINO

```
float temperatura = 0; // Núm. Real (con decimales) almacenado en 4 bytes (es decir 32 bits)

int INTERNAL=1;
int led1=2; //Define al Pin 2 como led1(Nivel Bajo)
int led2=3; //Pin 3 como led2(Nivel Medio)
int led3=4; //Pin 4 como led3(Nivel Alto)
int led4=5; //Pin 5 como led4(Alerta Alarma)

//FUNCION PRINCIPAL
void setup(){
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
  analogReference(INTERNAL);
}


//FUNCION CICLICA
void loop(){
  char c = client1.read();
  if (c == 's') digitalWrite(led4,HIGH);
  if (c == 'S') digitalWrite(led4,LOW);
  delay(10);

  //Calcula la temperatura usando como referencia 5V
  temperatura = (5.0 * analogRead(0)*100.0)/1024.0;
  delay (500); //tiempo para la siguiente medicion

  if (temperatura < 36){ //Enciende Led 1 Nivel Bajo
    digitalWrite (led1,LOW);
    digitalWrite (led2,HIGH);
    digitalWrite (led3,HIGH);
    delay (500);
  }
  if (temperatura >= 36 && temperatura <=45){ //Enciende Led 2 Nivel Medio
    digitalWrite (led2,LOW);
    digitalWrite (led1,HIGH);
    digitalWrite (led3,HIGH);
    delay (500);
  }
  if (temperatura > 46){ //Enciende Led 3 Nivel Alto
    digitalWrite (led3,LOW);
    digitalWrite (led1,HIGH);
```

```
digitalWrite (led2,HIGH);  
delay (500);  
}  
if (temperatura < 46){  
digitalWrite (led4,HIGH); //Led Alarma  
delay (3000);  
}  
}}
```

4.2.2 Práctica 2

	<p style="text-align: center;">UNIVERSIDAD POLITÉCNICA SALESIANA</p> <p style="text-align: center;">FACULTAD DE INGENIERÍAS</p> <p style="text-align: center;">SEDE GUAYAQUIL</p>
---	--

CARRERA	PRÁCTICA N°	NOMBRE DE LA PRÁCTICA
Electrónica	2	Registro de niveles de temperatura a través de la interfaz LabVIEW-Arduino y encendido de un ventilador auxiliar por medio de comunicación PLC.

OBJETIVOS
<ul style="list-style-type: none">• Analizar las características técnicas que la tecnología PLC brinda.• Implementar apropiadamente los equipos PowerLine Communications.• Establecer la comunicación entre LabVIEW y Arduino.• Generar, en LabVIEW, un registro en un documento de Excel de los datos obtenidos.

EQUIPOS A UTILIZAR	
<ul style="list-style-type: none">• Adaptador PowerLine TL-PA4010• Extensor PowerLine TL-WPA4220• Tarjeta Arduino Mega 2560• Tarjeta Arduino Ethernet Shield• Software LabVIEW y Arduino• Sensor de temperatura LM35	<ul style="list-style-type: none">• Luces piloto• Módulo de Relés 4 Canales• Laptop• Microsoft Excel• Cables de conexión• Ventilador

INSTRUCCIONES
<ul style="list-style-type: none">• Diseñar un sistema en el software LabVIEW que permita indicar los niveles de temperatura por medio de luces piloto.• Realizar la programación respectiva en el software Arduino para las tarjetas Mega 2560 y Ethernet Shield.• Declarar las variables útiles para el enlace entre LabVIEW y Arduino.• Conectar los equipos PLC, Adaptador y Extensor, a la línea de Baja Tensión (120V), y enlazarlos para su correcta comunicación.• Configurar en LabVIEW la herramienta para generar el registro de las temperaturas en un documento de Excel.• Realizar las conexiones, como se muestra en el diagrama, entre los módulos a usar para el desarrollo de la práctica.

- Conectar, vía Ethernet, los equipos PLC con el módulo de las tarjetas Arduino.
- Verificar todas las conexiones entre los módulos que intervienen en la práctica.
- Ejecutar desde el software LabVIEW la práctica a realizar, para verificar virtualmente que las luces piloto están en correcto funcionamiento.
- Generar en LabVIEW el archivo de Excel para que automáticamente se guarde la información de la temperatura obtenida en la ejecución de la práctica.
- Aplicar manualmente calor al sensor para la activación de las distintas luces piloto de acuerdo a los rangos de niveles de temperatura establecido y encendido del ventilador.

Tabla 5

Diagrama de conexiones de la Practica 2

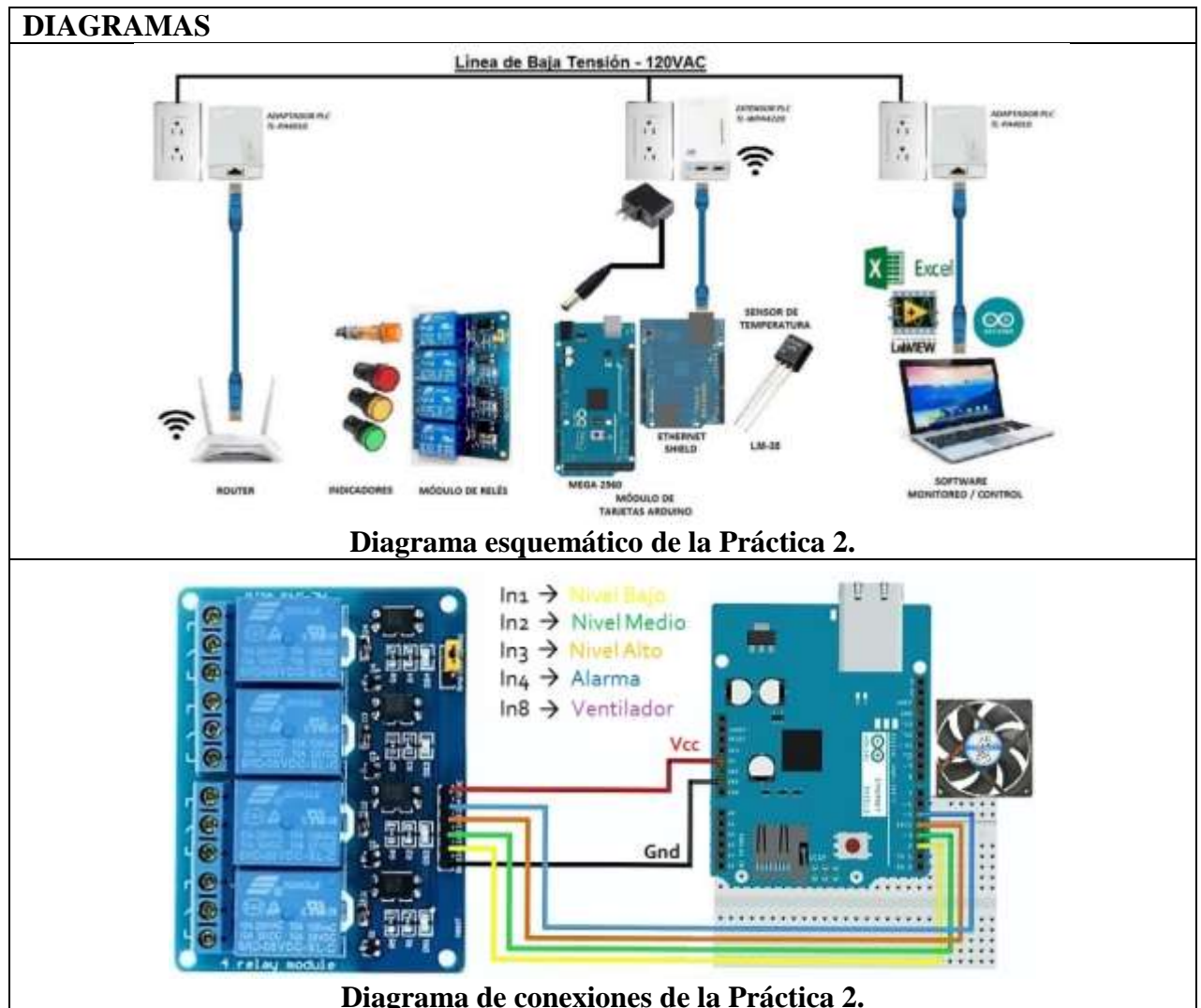


Tabla 6

Diseño de monitoreo y control en LabVIEW de la Practica 2.



Panel Frontal de la Práctica 2.

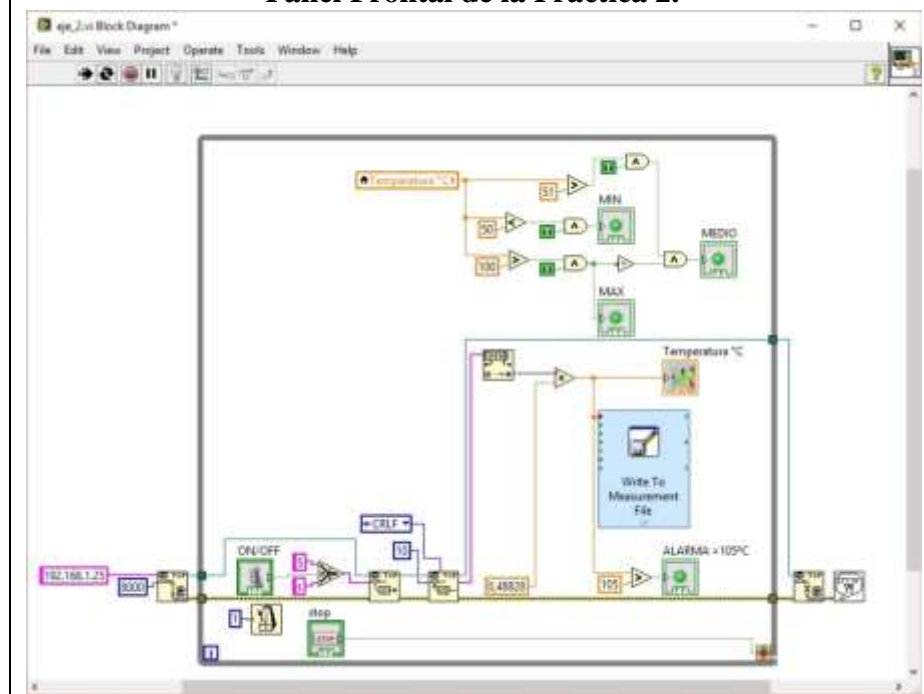


Diagrama de Bloques de la Práctica 2.

PROGRAMACIÓN ARDUINO

```
float temperatura = 0; // Núm. Real (con decimales) almacenado en 4 bytes (es decir 32 bits)
```

```
int INTERNAL=1;  
int led1=2; //Define al Pin 2 como led1(Nivel Bajo)  
int led2=3; //Pin 3 como led2(Nivel Medio)  
int led3=4; //Pin 4 como led3(Nivel Alto)  
int led4=5; //Pin 5 como led4(Alerta Alarma)  
int vent=8; //Pin 8 como Ventilador
```

```
//FUNCION PRINCIPAL
```

```
void setup()  
{  
  pinMode(led1,OUTPUT);  
  pinMode(led2,OUTPUT);  
  pinMode(led3,OUTPUT);  
  pinMode(led4,OUTPUT);  
  pinMode(vent, OUTPUT);  
  analogReference(INTERNAL);  
}
```

```
//FUNCION CICLICA
```

```
void loop(){  
{  
  char c = client1.read();  
  if (c == 's') digitalWrite(led4,HIGH);  
  if (c == 'S') digitalWrite(led4,LOW);  
  delay(10);
```

```
//Calcula la temperatura (como referencia 5V)  
temperatura = (5.0 * analogRead(0)*100.0)/1024.0;  
delay (500); //tiempo para la siguiente medicion
```

```
if (temperatura < 36){ //Enciende Led1 Nivel Bajo  
  digitalWrite (led1,LOW);  
  digitalWrite (led2,HIGH);  
  digitalWrite (led3,HIGH);  
  delay (500);  
}
```


```
if (temperatura >= 36 && temperatura <=45){ //Enciende Led2 Nivel Medio  
  digitalWrite (led2,LOW);  
  digitalWrite (led1,HIGH);  
  digitalWrite (led3,HIGH);  
  delay (500);  
}
```

```
if (temperatura > 46){ //Enciende Led3 Nivel Alto  
  digitalWrite (led3,LOW);
```

```
digitalWrite (led1,HIGH);
digitalWrite (led2,HIGH);
delay (500);
}
if (temperatura < 46){
digitalWrite (led4,HIGH); //Led Alarma
delay (3000);
}

int i = 0;          //Ventilador apagado ya que i=0
analogWrite(vent,i);
if(temperatura>=36) //Si Temperatura es mayor a 36°C se activa Ventilador al 60% aprox
{
int i = 200;
analogWrite(vent,i);
if(temperatura>45) //Si Temperatura es mayor a 48°C se activa Ventilador al 85% aprox
{
int i = 235;
analogWrite(vent,i); //FUNCION ANALOGWRITE TOMA VALORES ENTRE 0 y 255
}}
}}
```

4.2.3 Práctica 3

 <p>UNIVERSIDAD POLITÉCNICA SALESIANA GUAYAS</p>	<p style="text-align: center;">UNIVERSIDAD POLITÉCNICA SALESIANA</p> <p style="text-align: center;">FACULTAD DE INGENIERÍAS</p> <p style="text-align: center;">SEDE GUAYAQUIL</p>
---	--

CARRERA	PRÁCTICA Nº	NOMBRE DE LA PRÁCTICA
Electrónica	3	Control de giro de un Servomotor por medio de la interfaz Arduino a través de la Red eléctrica de Baja tensión con comunicación PLC.

OBJETIVOS
<ul style="list-style-type: none">• Entender el principio de aplicación y funcionamiento de la tecnología PLC• Establecer al software Arduino como interface de programación.• Lograr la comunicación con Arduino para la ejecución de la práctica.• Controlar el giro de un servomotor mediante la tecnología PLC.

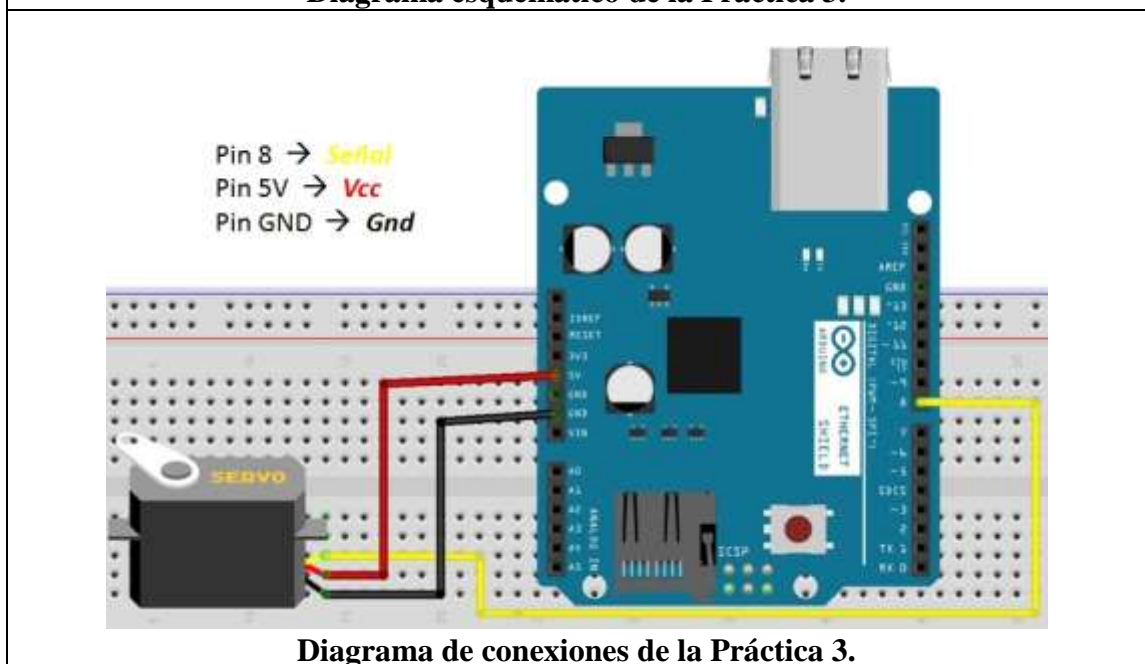
EQUIPOS A UTILIZAR	
<ul style="list-style-type: none">• Adaptador PowerLine TL-PA4010• Extensor PowerLine TL-WPA4220• Tarjeta Arduino Mega 2560• Tarjeta Arduino Ethernet Shield	<ul style="list-style-type: none">• Software Arduino• Servomotor• Laptop• Cables de comunicación

INSTRUCCIONES
<ul style="list-style-type: none">• Programar en el software Arduino las tarjetas Mega 2560 y Ethernet Shield las respectivas sentencias para que se ejecute correctamente el sistema.• Conectar energéticamente (120V) los equipos PLC (Adaptador y Extensor).• Declarar las variables necesarias para que se enlacen el software Arduino.• Hacer el respectivo cableado entre el servomotor y el módulo de las tarjetas Arduino (Mega 2560 y Ethernet Shield).• Establecer las respectivas variables en Arduino para que su ejecución de los resultados esperados en el control del servomotor.• Preparar los módulos con los equipos a utilizar, en este caso será el módulo de las tarjetas Arduino y el módulo del servomotor.• Conectar el módulo de las tarjetas Arduino vía Ethernet a los equipos PLC, y a su vez comprobar que exista la comunicación necesaria.

- Ejecutar la práctica, la cual por medio de la interfaz creada con Arduino se logre controlar el giro del servomotor.
- Demostrar los grados de giro que da el servomotor.

Tabla 7

Diagrama esquemático de equipos a usar en la Practica 3.



PROGRAMACIÓN ARDUINO

```
#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W510
#include <Ethernet.h> //Librería Ethernet para manejar el Shield
#include <Servo.h> //Librería para controlar el servomotor

Servo myservo; //Declaracion de variable tipo servo
int pos = 90; //Declaracion de variable tipo entero de la posición del servomotor

//FUNCION PRINCIPAL
void setup() {
  //Servomotor
  myservo.attach(8); //Pin8 para Servomotor
  myservo.write(pos); //Servomotor en la posición 'pos'
  delay(15);
}

//FUNCION CICLICA
while (client.connected()) {
  if (client.available()) {


    int posicion=cadena.indexOf("POS="); //Guardamos la posición de la instancia "POS="
    a la variable 'posicion'

    if(cadena.substring(posicion)== "POS=MAS")//Si 'posicion' hay "POS=MAS"
    {
      if (pos<180) pos+=5;
      myservo.write(pos); //Manda al servomotor a la posición de la variable 'pos'
    }
    if(cadena.substring(posicion)== "POS=MENOS")//Si 'posicion' hay "POS=MENOS"
    {
      if (pos>0) pos-=5;
      myservo.write(pos); //Manda al servomotor a la posición de la variable 'pos'
    }
    if(cadena.substring(posicion)== "POS=5")//Si 'posicion' hay "POS=CERO"
    {
      if (pos) pos=5;
      myservo.write(pos); //Manda al servomotor a la posición de la variable 'pos'
    }
    if(cadena.substring(posicion)== "POS=180")//Si 'posicion' hay "POS=CERO"
    {
      if (pos) pos=180;
      myservo.write(pos); //Manda al servomotor a la posición de la variable 'pos'
    }
    if(cadena.substring(posicion)== "POS=90")//Si 'posicion' hay "POS=CERO"
    {
      if (pos) pos=90;
```

```
myservo.write(pos);      //Manda al servomotor a la posicion de la variable 'pos'
}

//Cuando recibe una línea en blanco, la petición HTTP ha acabado y el servidor Web está
listo para enviar una respuesta
if (c == '\n' && currentLineIsBlank){
}
```

4.2.4 Práctica 4

	<p style="text-align: center;">UNIVERSIDAD POLITÉCNICA SALESIANA</p> <p style="text-align: center;">FACULTAD DE INGENIERÍAS</p> <p style="text-align: center;">SEDE GUAYAQUIL</p>
---	--

CARRERA	PRÁCTICA N°	NOMBRE DE LA PRÁCTICA
Electrónica	4	Monitoreo y simulación con Sensores de proximidad para abrir y cerrar puerta, usando Arduino y servomotores, con comunicación PLC por medio de la Red eléctrica de Baja tensión.

OBJETIVOS

- Comprender las características y utilidades de la tecnología PLC.
- Establecer al software Arduino como interface de programación.
- Indagar sobre las características técnicas de los sensores ultrasónicos.
- Establecer comunicación con el software Arduino mediante la tecnología PLC.

EQUIPOS A UTILIZAR

- | | |
|--|---|
| <ul style="list-style-type: none">• Adaptador Powerline TL-PA4010• Extensor Powerline TL-WPA4220• Tarjeta Arduino Mega 2560• Tarjeta Arduino Ethernet Shield• Software Arduino | <ul style="list-style-type: none">• Sensor Ultrasónico (2)• Servomotor (2)• Laptop• Microsoft Excel• Cables de comunicación |
|--|---|

INSTRUCCIONES

- Declarar la función que permita guardar las distancias obtenidas en un documento de Excel.
- Programar en el software Arduino las tarjetas Mega 2560 y Ethernet Shield las respectivas sentencias de acuerdo a las variables declaradas en LabVIEW.
- Realizar las conexiones respectivas a los pines declarados en el software Arduino, como se muestra en el diagrama, y revisar que los pines a utilizar estén correctamente conectados de acuerdo a lo programado.
- Activar los módulos de los equipos PLC y lograr que se enlacen entre ambos.
- Conectar vía Ethernet los módulos de los equipos PLC con el de las tarjetas Arduino.
- Acoplar el módulo de los sensores ultrasónico junto a los módulos de las tarjetas Arduino y de los equipos PLC.

- Realizar la correcta programación que enlace a los sensores de proximidad con los servomotores para la respectiva ejecución de la simulación.
- Mostrar las distancias medidas por los sensores de proximidad en una página web de acuerdo a una IP establecida.

Tabla 8

Diagrama esquemático de equipos a usar en la Práctica 4.

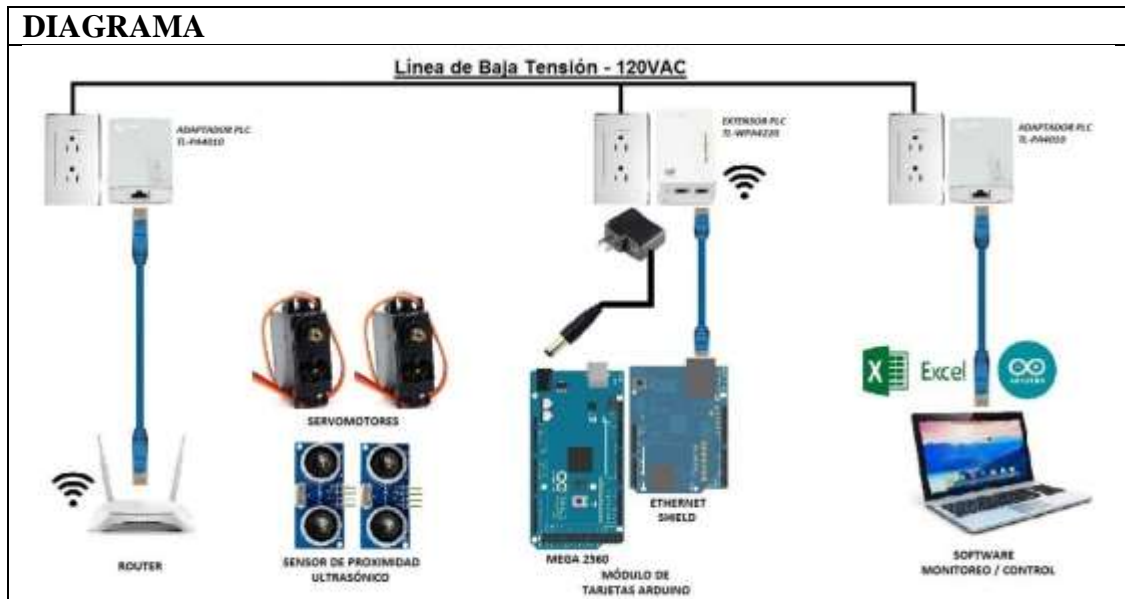


Diagrama esquemático de la Práctica 4.

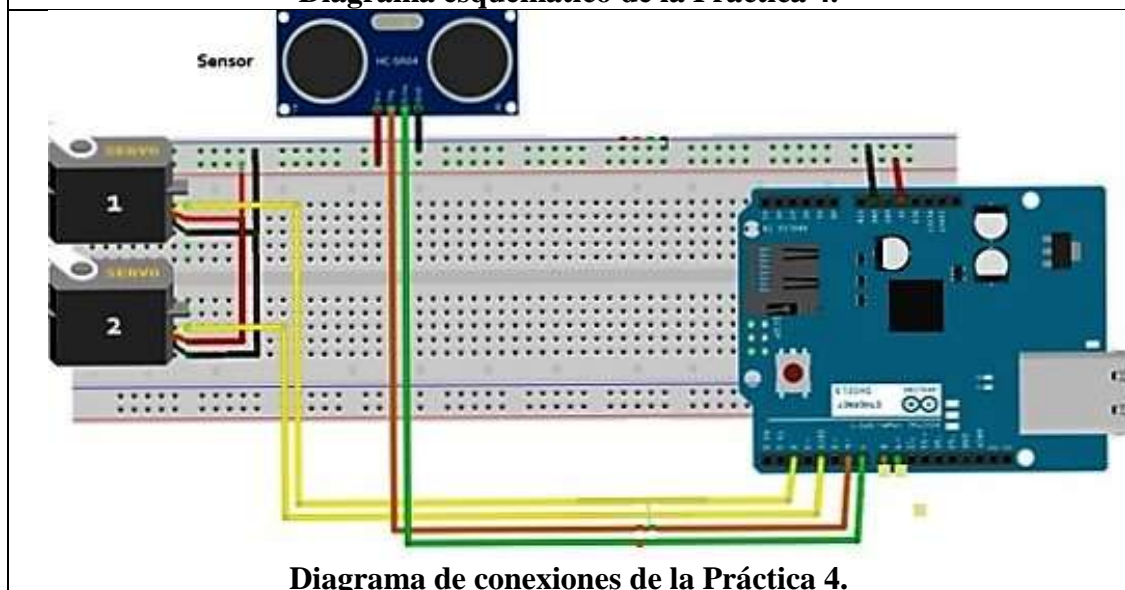


Diagrama de conexiones de la Práctica 4.

PROGRAMACIÓN ARDUINO

```
#include <Ultrasonic.h>
#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>

//Definir parametros del Sensor HC-SR04
#define TRIGGER 6 // Trigger del sensor
#define ECHO 7 // Echo del sensor

//Inicializacion del Sensor ultrasonico
Ultrasonic ultrasonico (TRIGGER,ECHO);

Servo servo1; //Declaracion de variable tipo servo para controlar el servomotor
Servo servo2;
int pos1 = 180; //Declaracion de variable tipo entero de la posición del servomotor
int pos2 = 180;

String estado; //estado para la puerta.

//Declaracion de IP, mascara de Red y Gateway
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress IP(192,168,1,20); //Asignacion de IP
IPAddress Gateway(192,168,1,1); //Asignacion de Gateway
IPAddress Subnet(255, 255, 255, 0); //Asignacion de Máscara de red

//Inicio del servidor puerto 80
EthernetServer server(80);

//FUNCION PRINCIPAL
void setup(){
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin(); // conexiones de entrantes
  Serial.println(Ethernet.localIP()); //Obtiene la dirección IP

  //Servomotor
  servo1.attach(2); //Conectar el servomotor al pin 2
  servo2.attach(4); //Conectar el servomotor al pin 4
  servo1.write(pos1); //Manda al servomotor a la posicion de la variable 'pos'
  servo2.write(pos2);
}

//FUNCION CICLICA
void loop(){
  float cmMsec; //Tipo de datos de números
  long microsec = ultrasonico.timing(); //Lectura del sensor
  cmMsec = ultrasonico.convert(microsec, Ultrasonic::CM); //Convertir la distancia en cm
```

```

EthernetClient client = server.available();// Esperar conexión de navegador
if (client){
  Serial.println("new client");
  boolean currentLineIsBlank = true; //petición HTTP acaba con línea en blanco
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      Serial.write(c);
      // Si ha llegado al final de la línea (recibido una nueva línea
      // carácter y la línea está en blanco, la petición HTTP ha terminado,
      // por lo que puede enviar una respuesta
      if (c == '\n' && currentLineIsBlank){

        //Enviar un encabezado de respuesta HTTP estándar
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        //La conexión se cierra después de la finalización de la respuesta
        client.println("Connection: close");
        client.println("Refresh:1"); //Actualizar pagina cada 1seg
        client.println();
        client.println("<!DOCTYPE HTML>");
        client.println("<html>");

        client.println("<body bgcolor=\"#FFFF99\">");
        client.println("<hr/>");
        client.println("<hr/>");
        client.println("<h4><center><img                                border=\"2\"
src=\"http://carlosordonezidrovo.info/wp-content/uploads/sites/389/2015/09/LOGO-
UPSp-10x3cm.jpg\" /></center></h4>");
        client.println("<hr/>");
        client.println("<hr/>");
        client.println("<title>UNIVERSIDAD</title>");
        client.println("</head>");
        client.println("<body width=150% height=150%>");
        client.println("<center>");
        client.println("<h1>PROYECTO DE TITULACION");
        client.print("<br>");

        client.print("<font color=#0000FF><b><u>");
        client.print("SENSOR ULTRASONICO 1");
        client.print("</u></b></font>");
        client.println("<br />");
        client.println("<br />");

        //Mostrar datos de medición del sensor
        client.print("Distancia Sensor Ultrasonico: ");

```

```

client.print("<b>");
client.print("<font color=#4B9C4A><b><u>");
client.print(cmMsec);
client.print("cm");
client.println("</b></html>");
client.println("<br />");
client.println("<br />");
client.print("<font color=#0000FF><b><u>");
client.print("PUERTA: ");
client.print("<b>");
client.print("<font color=#e70c23><b><u>");
client.print(estado);
client.println("<br />");
client.println("<br />");


if (cmMsec< 30) pos1 = 0 ;
if (cmMsec> 30) pos1 = 180 ;
  servo1.write(pos1); //Manda al servomotor a la posicion de la variable 'pos'
if (cmMsec< 30) pos2 = 180 ;
if (cmMsec> 30) pos2 = 0 ;
servo2.write(pos2); //Manda al servomotor a la posicion de la variable 'pos'

  if(cmMsec<30)
    estado ="OPEN" ;

  if (cmMsec>30)
    estado ="CLOSE" ;
}
}}

```

4.2.5 Práctica 5

	<p style="text-align: center;">UNIVERSIDAD POLITÉCNICA SALESIANA</p> <p style="text-align: center;">FACULTAD DE INGENIERÍAS</p> <p style="text-align: center;">SEDE GUAYAQUIL</p>
---	--

CARRERA	PRÁCTICA N°	NOMBRE DE LA PRÁCTICA
Electrónica	5	Comprobación del estado de Bombillos de luces a través de Internet (web server) utilizando un Smartphone.

OBJETIVOS
<ul style="list-style-type: none">• Establecer comunicación entre el software Arduino y los equipos PLC mediante la asignación de una dirección IP.• Conocer sobre las variables necesarias en la programación Arduino para el diseño de un servidor web (web server).• Aprender a diseñar una página web desde la programación Arduino.• Declarar variables y sentencias en el software Arduino para la comunicación entre las tarjetas Arduino Mega 2560 y Ethernet Shield y la red de internet.

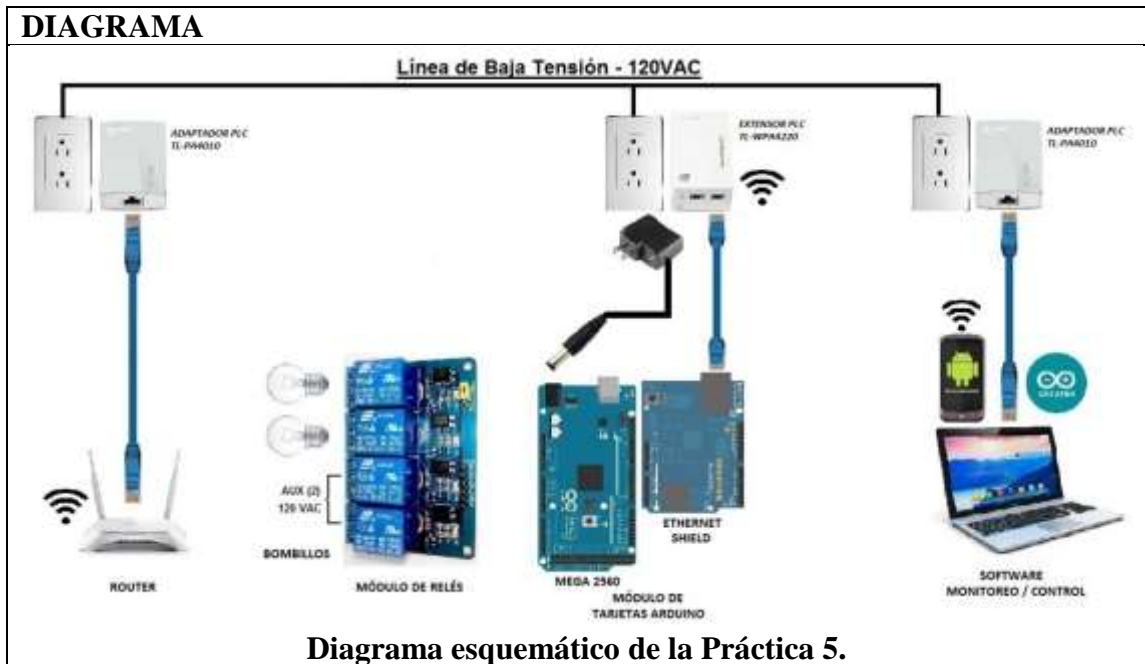
EQUIPOS A UTILIZAR	
<ul style="list-style-type: none">• Adaptador PowerLine TL-PA4010• Extensor PowerLine TL-WPA4220• Tarjeta Arduino Mega 2560• Tarjeta Arduino Ethernet Shield• Celular Smartphone	<ul style="list-style-type: none">• Bombillos (2)• Laptop• Software Arduino• Cables de comunicación

INSTRUCCIONES
<ul style="list-style-type: none">• Programar las tarjetas de comunicación en el software Arduino, declarar los pines a utilizar, incluir todas las librerías necesarias para la comunicación por internet.• Utilizar las variables y sentencias necesarias para el diseño de nuestro servidor web.• Declarar en el software Arduino las especificaciones que mostrara la página web para el control de las bombillas.• Comprobar que los equipos Power Line Communications estén enlazados y con conexión a internet.• Conectar el módulo de tarjetas Arduino al módulo de los bombillos, a los respectivos pines que se declararon en la programación, como se aprecia en el diagrama.

- Ingresar desde el smartphone a la dirección IP asignada previamente a las tarjetas de comunicación Arduino.
- Verificar que se ejecuten las sentencias programadas en las tarjetas de comunicación.

Tabla 9

Diagrama esquemático de equipos a usar en la Práctica 5.



PROGRAMACIÓN ARDUINO

```
#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W5100
#include <Ethernet.h> //Librería Ethernet para manejar el Shield
```

```
//Declaración de Pines a usar en la tarjeta Arduino
```

```
int PIN_FOC01=8;
int PIN_FOC02=7;
```

```
//Declaración de direcciones MAC e IP
```

```
IPAddress ip(192,168,1,20);
```

```
String readString=String(30); //Lee los caracteres de una secuencia en una cadena.
```

```
//Los strings se representan como arrays de caracteres (tipo char)
```

```
String state1;
```

```
String state2;
```

```

//FUNCION PRINCIPAL
void setup(){
pinMode(PIN_FOCO1,OUTPUT);
digitalWrite(PIN_FOCO1,HIGH);
pinMode(PIN_FOCO2,OUTPUT);
digitalWrite(PIN_FOCO2,HIGH);
}


//FUNCION CICLICA
void loop(){
//Almacenar los caracteres en la variable readString
if(readString.length()<30){
readString.concat(c);
}
{
int FOCO = readString.indexOf("LUZ=");
if(readString.substring(FOCO,FOCO+5)== "LUZ=A") {
digitalWrite(PIN_FOCO1,LOW);

state1="ON"; }
if(readString.substring(FOCO,FOCO+5)== "LUZ=C") {
digitalWrite(PIN_FOCO2,LOW);
state2="ON"; }
else if (readString.substring(FOCO,FOCO+5)== "LUZ=B") {
digitalWrite(PIN_FOCO1,HIGH);

state1="OFF";}
else if (readString.substring(FOCO,FOCO+5)== "LUZ=D") {
digitalWrite(PIN_FOCO2,HIGH);
state2="OFF";
}
}

```

4.2.6 Práctica 6

	<p style="text-align: center;">UNIVERSIDAD POLITÉCNICA SALESIANA</p> <p style="text-align: center;">FACULTAD DE INGENIERÍAS</p> <p style="text-align: center;">SEDE GUAYAQUIL</p>
---	--

CARRERA	PRÁCTICA N°	NOMBRE DE LA PRÁCTICA
Electrónica	6	Control del Arduino desde un dispositivo Android vía Bluetooth.

OBJETIVOS
<ul style="list-style-type: none">• Establecer comunicación entre el software Arduino y los equipos PLC mediante la asignación de una dirección IP.• Experimentar la herramienta APP Inventor 2 como plataforma para el diseño de la aplicación para controlar la tarjeta Arduino.• Diseñar una aplicación en el Smartphone acorde a los requerimientos de la práctica.• Declarar variables y sentencias en el software Arduino para la comunicación entre las tarjetas Mega 2560 y Ethernet Shield y la red de internet.• Conocer las características técnicas del módulo Bluetooth para su correcta aplicación.

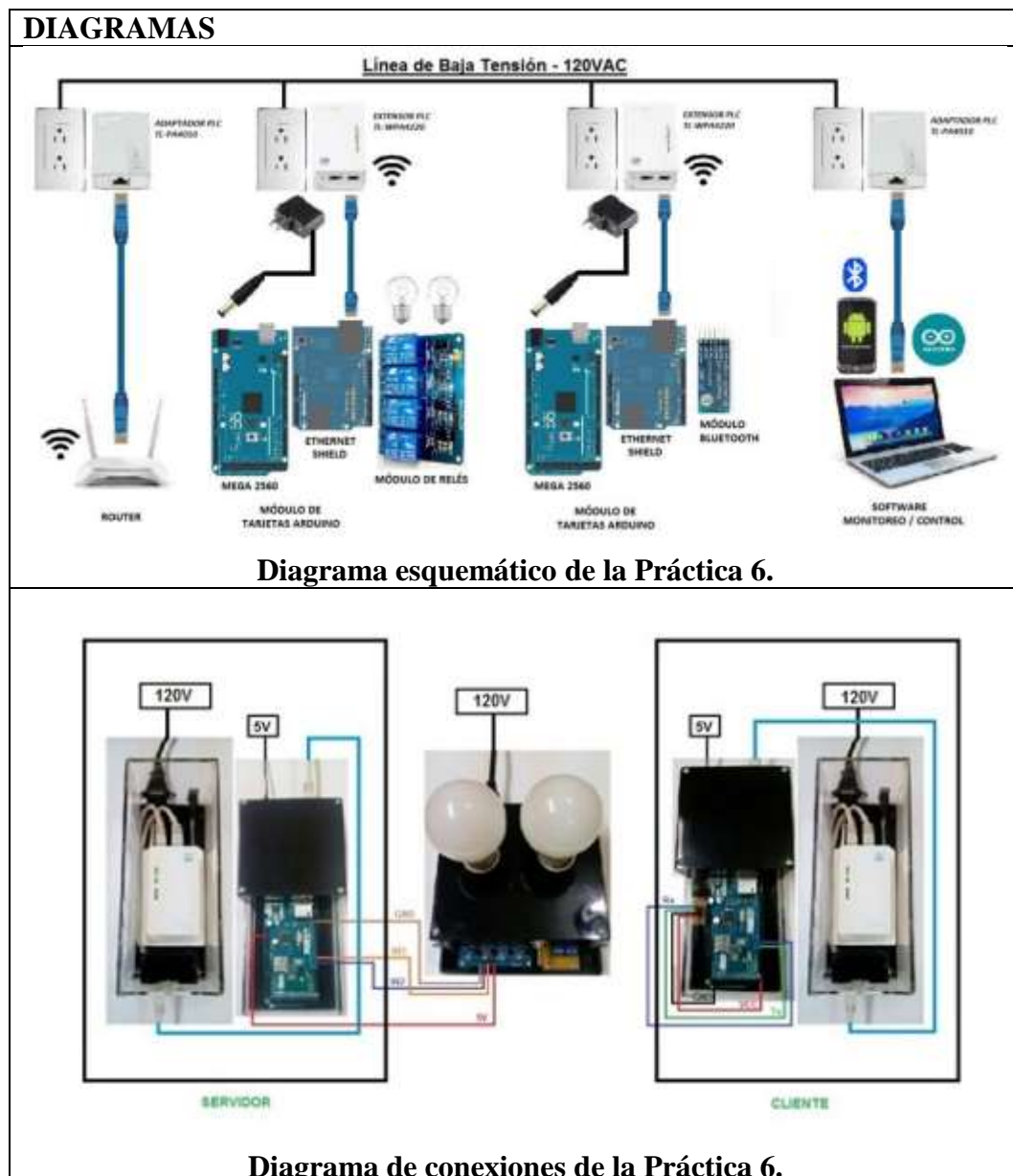
EQUIPOS A UTILIZAR	
<ul style="list-style-type: none">• Adaptador Powerline TL-PA4010• Extensor Powerline TL-WPA4220• Tarjeta Arduino Mega 2560• Tarjeta Arduino Ethernet Shield• Módulo Bluetooth Arduino HC-06	<ul style="list-style-type: none">• Celular Smartphone• Bombillo (2)• Laptop• Software Arduino• Cables de comunicación

INSTRUCCIONES
<ul style="list-style-type: none">• Diseñar en la plataforma App Inventor 2 la interfaz que me permita controlar las tarjetas Arduino.• Incluir, en la programación Arduino, las librerías de conexión a internet correspondientes y la asignación de la dirección IP para las tarjetas Arduino.• Declarar, en la programación del SERVIDOR, las constantes para la recepción de estados de pines de entrada correspondientes para cada bombilla.• Programar, en Arduino, al CLIENTE para la comunicación con su servidor mediante Bluetooth, también las constantes para envío de estados de pines de entrada.

- Asignar una IP única para lo que será el cliente y otra para el servidor.
- Conectar el módulo Bluetooth al módulo de las tarjetas Arduino, en los pines declarados en la programación respectiva.
- Verificar las direcciones IP asignadas a cada módulo de las tarjetas Arduino para saber cuál es nuestro Cliente y cual el Servidor.
- Por medio de internet, ejecutar desde el smartphone la aplicación creada para el control de las tarjetas Arduino.

Tabla 10

Diagramas de conexiones de la Practica 6.



PROGRAMACIÓN ARDUINO

```
//SERVIDOR

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W5100
#include <Ethernet.h> //Librería Ethernet para manejar el Shield

//Declaramos constantes para la recepción de estados de pines de entrada, deben de ser las
mismas que en la otra tarjeta
const byte BOMBILLA1_ON = 0x41;
const byte BOMBILLA1_OFF = 0x61;
const byte BOMBILLA2_ON = 0x42;
const byte BOMBILLA2_OFF = 0x62;

//Declaración de Señales Hardware
int BOMBILLA1 = 2, BOMBILLA2 = 3;

//Declaración de Parámetros de red
byte mac[] = {0x6E, 0xFD, 0xE5, 0x23, 0x0C, 0x9F};
IPAddress ipLocal(192,168,1,20);
IPAddress puertaEnlace (192,168,2,1);
IPAddress mascaraSubRed (255,255,255,0);

//Creamos un servidor y asignamos el puerto 23
EthernetServer servidor = EthernetServer(23);

//FUNCION PRINCIPAL
void setup() {
  // Configuración de Señales Hardware
  Serial.begin(9600);

  pinMode(BOMBILLA1, OUTPUT); //Se activara si se activa la entrada 1 de la otra
tarjeta
  pinMode(BOMBILLA2, OUTPUT); //Se activara si se activa la entrada 2 de la otra
tarjeta

  // Configuración de la red del dispositivo
  Ethernet.begin(mac, ipLocal, puertaEnlace, mascaraSubRed);

  // Iniciamos el servidor
  servidor.begin();
}

//FUNCION CICLICA
void loop() {
```

```

if (servidor.available()) {
  //Si hay datos disponibles los asignamos a un cliente
  EthernetClient cliente = servidor.available();

  //Leemos los datos
  byte dato = cliente.read();

  // Según dato recibido actuaran las salidas
  if (dato == BOMBILLA1_ON) digitalWrite(BOMBILLA1,HIGH);
  if (dato == BOMBILLA1_OFF) digitalWrite(BOMBILLA1,LOW);
  if (dato == BOMBILLA2_ON) digitalWrite(BOMBILLA2,HIGH);
  if (dato == BOMBILLA2_OFF) digitalWrite(BOMBILLA2,LOW);

  //Devolvemos un eco al cliente para vaciar
  cliente.write(dato);
}
}

//CLIENTE

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W5100
#include <Ethernet.h> //Librería Ethernet para manejar el Shield

//Declaramos constantes para envío de estados de pines de entrada
const byte BOMBILLA1_ON = 0x41;
const byte BOMBILLA1_OFF = 0x61;
const byte BOMBILLA2_ON = 0x42;
const byte BOMBILLA2_OFF = 0x62;

//Declaración de Señales Hardware
char val;

// Declaración de Parámetros de red
byte mac[] = {0xDE, 0xFD, 0xE5, 0xD3, 0x0C, 0x9A};
IPAddress ipLocal(192,168,1,25);
IPAddress puertaEnlace (192,168,2,1);
IPAddress mascaraSubRed (255,255,255,0);

// IP de la otra tarjeta
IPAddress ipArduServ(192,168,1,20);

//Creamos un cliente
EthernetClient cliente;

//FUNCION PRINCIPAL
void setup(){
  Serial.begin(9600); // Aquí establecemos la velocidad

```

```

// Configuración de la red del dispositivo
Ethernet.begin(mac, ipLocal, puertaEnlace, mascaraSubRed);
}

//FUNCION CICLICA
void loop(){
while (!cliente.connect(ipArduServ, 23))
{
//Se espera obtener conexión con IP de la otra tarjeta por el Puerto 23
if( Serial.available() )
val = Serial.read();

if( val == '0' )
cliente.write (BOMBILLA1_ON);
delay (200);
if( val == '1' )
cliente.write(BOMBILLA1_OFF);
delay (200);

if( val == '2' )
cliente.write (BOMBILLA2_ON);
delay (200);
if( val == '3' )
cliente.write(BOMBILLA2_OFF);
delay (200);
}
}
}

```

5. IMPACTO

Debido a las constantes actualizaciones en el ámbito de avances tecnológicos, se utiliza la comunicación PLC (Power Line Communications) para la transmisión de datos, empleando la red eléctrica convencional, en baja tensión, que permite comunicar dos o más puntos (tomacorrientes) de un mismo circuito eléctrico existente, sin la necesidad de rediseñar e implementar nuevos puntos de alimentación o cableado estructurado, facilitando la comunicación para el envío y recepción de datos en prácticas que empleen monitoreo y control, como arrancadores de motores, encendido y apagado de luces, estado de sensores, etc.

El presente proyecto de titulación culminado, sienta como precedente de innovación en la comunicación y transferencia de datos mediante la tecnología PLC, que actúa por medio de la red eléctrica instalada, con tomacorrientes convencionales de un mismo circuito.

En el desarrollo de los módulos para el laboratorio de Telecomunicaciones, con interfaz de comunicación y adquisición de datos LabVIEW-Arduino, se innovo el control de servomotores, luces piloto, sensores de temperatura, optimizando y facilitando recursos de conexión entre software y hardware. Además se diseñó una aplicación con el software libre App Inventor 2 para el control de estado de bombillos desde un smartphone. Demostramos el alcance y beneficios que podemos conseguir implementado esta tecnología, con los equipos PowerLine se alcanzará una mayor distribución de la señal, por red eléctrica o Wireless, ya que abarca ampliamente cada espacio o rincón en el laboratorio o aula de clases siempre y cuando los tomacorrientes a usar se encuentren en el mismo circuito de la red eléctrica.

Siendo importante que esta herramienta de software libre, App Inventor 2, permite generar infinidad de aplicaciones de acuerdo a las características de cada módulo implementado, así como también diseñar otras para el control de otros módulos existentes en los laboratorios de la universidad.

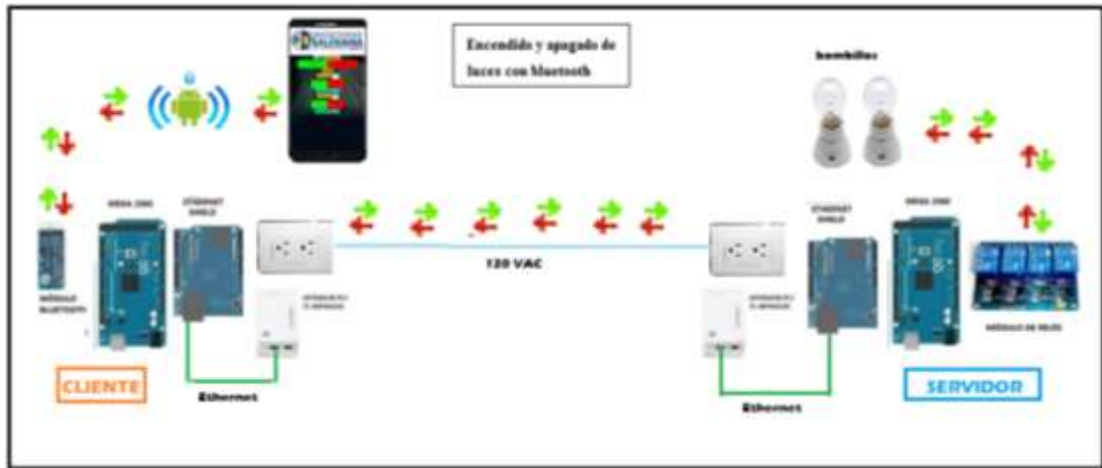
6. FUNCIONALIDAD

Mediante el presente proyecto técnico diseñado e implementado queremos dar a conocer más alternativas, opciones, innovaciones en lo que respecta a medios para la transmisión de datos entre software y hardware para el monitoreo y control de equipos o módulos que posee la universidad en sus respectivos laboratorios de enseñanza y aprendizaje de las materias de la carrera de ingeniería.

Se pudo establecer comunicación entre los módulos desarrollados y la interfaz LabVIEW-Arduino mediante lo cual demostramos el envío y recepción de datos a través de la red eléctrica de baja tensión implementando la tecnología Power Line Communications. Comprobamos que existe la comunicación, ya que los módulos didácticos diseñados respondieron a lo programado, tanto en las tarjetas de comunicación Arduino como también en el software LabVIEW en donde se diseñó, creo y codifico variables, sentencias para la adquisición y proceso de datos.

La funcionalidad del presente proyecto técnico de titulación se consiguió estudiando, analizando las características y ventajas que nos brinda la tecnología PLC, ya que hay nos basamos para la implementación del sistema de transferencia de datos presente, verificando resultados ejecutando las diferentes practicas propuestas como son el control on/off de luces pilotos y bombillos con la transmisión de datos por medio de la red eléctrica. Así mismo, se diseñó, creó una aplicación en la plataforma App Inventor 2, que es un software libre para la elaboración de aplicaciones móviles para sistemas Android, con las características y sentencias necesarias para el monitoreo y control de estados de bombillos, todo esto desde un smartphone.

Explicando todo la funcionalidad del proyecto de titulación presentado, también demostramos que se pudo ejecutar el sistema de transferencia de datos sin la necesidad del rediseño de la infraestructura existente en el laboratorio de clases, se consiguieron los resultados esperados basándonos en la red eléctrica instalada, en los tomacorrientes existentes, demostrando que hay la innovación en la comunicación para el envío de información por este medio evitando cableado extenso o trabajo en espacios reducidos.



Los módulos con esta tecnología están elaborados para el desarrollo de prácticas de distintas materias afines de la carrera, así como también sería muy útil en seminarios y también servirán para futuros estudios, análisis o investigación del desarrollo de esta tecnología en el mundo.

CONCLUSIONES

- Se logró establecer una nueva alternativa a nivel de comunicación de dispositivos y/o equipos mediante la tecnología PLC, usando la red eléctrica existente en el laboratorio como medio de transmisión de datos.
- Se visualizó que como punto importante de esta tecnología es la asignación de una dirección IP respectiva para la comunicación entre software para su correcto funcionamiento.
- Se estableció un web server y se creó una aplicación en App Inventor 2, software libre, para implementar un sistema de control de luces piloto utilizando la tecnología PLC, que es de fácil manejo e instalación en celulares smartphones.
- Por medio de una red eléctrica convencional se puede comunicar equipos IP utilizando la tecnología PLC para la transmisión de datos.
- Se diseñó una aplicación personal en la plataforma App Inventor 2 y se la enlazó con Arduino, por medio de la cual logramos controlar dispositivos.

RECOMENDACIONES

- Es recomendable que en los casos que se usen sensores, de temperatura o ultrasónicos, verificar que se asignen a pines en las tarjetas Arduino que cumplan y lean esas instrucciones.
- Comprobar el estado de la red eléctrica y tomacorrientes del área en donde se desee implementar la tecnología PLC para evitar contratiempos en el desarrollo del trabajo planeado.
- Al momento de diseñar una interfaz con las características necesarias para el desarrollo de la práctica deseada, es importante declarar la única dirección IP válida para la distribución de datos y acciones a realizar.
- Se recomienda utilizar la plataforma App Inventor 2 con el fin de conocer más opciones de control móvil de proyectos a nivel industriales diseñando aplicaciones con particularidades personales.

REFERENCIAS

- BURUTEK. (2015). *BURUTEK*. Retrieved from burutek.org:
<http://burutek.org/es/arduino/>
- Electronilab. (n.d.). *Electronilab* ®. Retrieved from Electronilab.co:
<http://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/>
- Hitek, P. (2014). *Panamahitek*. Retrieved from <http://panamahitek.com/que-es-arduino-y-para-que-se-utiliza/>
- Instruments, N. (2016). *National Instruments Corporation*. Retrieved from National Instruments Corporation: <http://www.ni.com/labview/esa/>
- Instruments, N. (2016). *National Instruments Corporation*. Retrieved from National Instruments Corporation: <http://www.ni.com/labview/applications/esa/>
- Interactivo, A. (2015). *Art Interactivo*. Retrieved from Art Interactivo:
<http://www.artinteractivo.com/arduino-ethernet>
- Labs, S. (2016). *Sual Labs* ©. Retrieved from Sual Labs: <http://www.suallabs.com/Servo-Estandar-Hitec-HS311>
- Moviles, A. (n.d.). *appmoviles.net*. Retrieved from appmoviles.net:
<http://appmoviles.net/que-es-el-app-inventor-para-que-sirve/>
- NEOTEO. (2016). *NeoTeo.com*. Retrieved from © NeoTeo:
<http://www.neoteo.com/modulo-bluetooth-hc-06-android>
- Núñez, J. L. (n.d.). *Tu AppInvetor*. Retrieved from Tu AppInvetor:
<http://www.tuappinvetorandroid.com/aprender/>
- Redeweb*. (n.d.). Retrieved from http://www.redeweb.com/_txt/676/62.pdf
- Riffo, R. (2009). *Cybertesis Universidad Austral deChile*. Retrieved from <http://cybertesis.uach.cl/tesis/uach/2009/bmfcir564t/doc/bmfcir564t.pdf>

- S., P. E. (2013). *PID Electronics S. de R.L. de C.V.* Retrieved from Soko Labs®:
<http://pidelectronics.com/arduino-mega-2560-r3.html>
- sadds. (2012). *adfsdfa*. gye: swefw.
- Serna, V. H. (2011, Marzo). *Redeweb*. Retrieved from
http://www.redeweb.com/_txt/676/62.pdf
- TP-LINK. (2014). *TP-LINK Technologies Co., Ltd.* Retrieved from <http://www.tp-link.ec/products/details/?model=TL-PA4010>
- TP-LINK. (2015). *TP-LINK Technologies Co., Ltd.* Retrieved from http://www.tp-link.es/products/details/cat-18_TL-WPA4220.html
- victorgarcia.org*. (n.d.). Retrieved from <http://www.victorgarcia.org/files/PLC-v2.0RC.pdf>

ANEXOS

Anexo 1: Programación en Arduino de la Practica 1.

```
//UNIVERSIDAD POLITÉCNICA SALESIANA
//INGENIERÍA ELECTRÓNICA
//PROYECTO TÉCNICO – PONCE-SANTILLÁN
//PRÁCTICA #1: NIVELES DE TEMPERATURA

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W510
#include <Ethernet.h> //Librería Ethernet para manejar el Shield

float temperatura = 0; //Núm. Real (con decimales) almacenado en 4 bytes (32 bits)
int INTERNAL=1;
int led1=2; //Define al Pin 2 como led1(Nivel Bajo)
int led2=3; //Define al Pin 3 como led2(Nivel Medio)
int led3=4; //Define al Pin 4 como led3(Nivel Alto)
int led4=5; //Define al Pin 5 como led4(Alerta Alarma)

//Declaración de direcciones MAC e IP
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1,20);
//Inicialización de la librería Ethernet puerto 8000
EthernetServer server(8000);

//FUNCION PRINCIPAL
void setup(){
  Serial.begin(9600); //Inicializamos la conexión Ethernet y Servidor
  Ethernet.begin(mac, ip); //Inicializamos con las direcciones asignadas

  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
  analogReference(INTERNAL);
}

//FUNCION CICLICA
void loop(){
  EthernetClient client1 = server.available();
  if (client1.available()){
    char c = client1.read();
    if (c == 's') digitalWrite(led4,HIGH); //Alarma Intermitente
    if (c == 'S') digitalWrite(led4,LOW); //Alarma Intermitente
    delay(10);
  }
}
```

```

//Calculo de temperatura (como referencia 5V)
temperatura = (5.0 * analogRead(0)*100.0)/1024.0;
delay (500); //Tiempo para siguiente medicion

if (temperatura < 36){ //Enciende Led 1: Nivel Bajo
  digitalWrite (led1,LOW);
  digitalWrite (led2,HIGH);
  digitalWrite (led3,HIGH);
  delay (500);
}
if (temperatura >= 36 && temperatura <=45){ //Enciende Led 2: Nivel Medio
  digitalWrite (led2,LOW);
  digitalWrite (led1,HIGH);
  digitalWrite (led3,HIGH);
  delay (500);
}
if (temperatura > 46){ //Enciende Led 3:: Nivel Alto
  digitalWrite (led3,LOW);
  digitalWrite (led1,HIGH);
  digitalWrite (led2,HIGH);
  delay (500);
}
if (temperatura < 46){
  digitalWrite (led4,HIGH); //Led 4: Alarma
  delay (3000);
}}

server.println(analogRead(0));
Serial.println(analogRead(0));
}

```

Anexo 2: Programación en Arduino de la Practica 2.

```
//UNIVERSIDAD POLITÉCNICA SALESIANA
//INGENIERÍA ELECTRÓNICA
//PROYECTO TÉCNICO - PONCE-SANTILLAN
//PRÁCTICA #2: NIVELES DE TEMPERATURA CON VENTILADOR

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W510
#include <Ethernet.h> //Librería Ethernet para manejar el Shield

float temperatura = 0; //Núm. Real (con decimales) almacenado en 4 bytes (32 bits)
int INTERNAL=1;
int led1=2; //Define al Pin 2 como led1(Nivel Bajo)
int led2=3; //Define al Pin 3 como led2(Nivel Medio)
int led3=4; //Define al Pin 4 como led3(Nivel Alto)
int led4=5; //Define al Pin 5 como led4(Alerta Alarma)
int vent=8; //Define al Pin 8 como (Ventilador)

//Declaración de direcciones MAC e IP
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1,20);
//Inicialización de la librería Ethernet puerto 8000
EthernetServer server(8000);

//FUNCION PRINCIPAL
void setup(){
  Serial.begin(9600); //Inicializamos la conexión Ethernet y Servidor
  Ethernet.begin(mac, ip); //Inicializamos con las direcciones asignadas

  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
  pinMode(vent, OUTPUT);
  analogReference(INTERNAL);
}

//FUNCION CICLICA
void loop(){
  EthernetClient client1 = server.available();
  if (client1.available()){
    char c = client1.read();
    if (c == 's') digitalWrite(led4,HIGH); //Alarma Intermitente
    if (c == 'S') digitalWrite(led4,LOW); //Alarma Intermitente
    delay(10);
  }
}
```

```

//Calculo de temperatura (como referencia 5V)
temperatura = (5.0 * analogRead(0)*100.0)/1024.0;
delay (500); //Tiempo para siguiente medicion

if (temperatura < 36){ //Enciende Led 1: Nivel Bajo
  digitalWrite (led1,LOW);
  digitalWrite (led2,HIGH);
  digitalWrite (led3,HIGH);
  delay (500);
}
if (temperatura >= 36 && temperatura <=45){ //Enciende Led 2: Nivel Medio
  digitalWrite (led2,LOW);
  digitalWrite (led1,HIGH);
  digitalWrite (led3,HIGH);
  delay (500);
}
if (temperatura > 46){ //Enciende Led 3: Nivel Alto
  digitalWrite (led3,LOW);
  digitalWrite (led1,HIGH);
  digitalWrite (led2,HIGH);
  delay (500);
}
if (temperatura < 46){
  digitalWrite (led4,HIGH); //Led 4: Alarma
  delay (3000);
}

int i = 0; //Ventilador apagado cuando i=0
analogWrite(vent,i);
if(temperatura>36) //Si Temperatura >36°C, se activa Ventilador al 60% aprox
{
  int i = 200;
  analogWrite(vent,i);

  if(temperatura>45) //Si Temperatura > 45°C, se activa Ventilador al 85% aprox
  {
    int i = 235;
    analogWrite(vent,i); //La funcion AnalogWrite toma valores entre 0 y 255.
  }
}

server.println(analogRead(0));
Serial.println(analogRead(0));
}
}

```

Anexo 3: Programación en Arduino de la Practica 3.

```
//UNIVERSIDAD POLITÉCNICA SALESIANA
//INGENIERÍA ELECTRÓNICA
//PROYECTO TÉCNICO - PONCE-SANTILLAN
//PRÁCTICA #3: CONTROL DE SERVOMOTOR

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W510
#include <Ethernet.h> //Librería Ethernet para manejar el Shield
#include <Servo.h> //Librería para controlar el servomotor

Servo myservo; //Declaracion de variable tipo servo
int pos = 90; //Declaracion de variable tipo entero de la posición del servomotor

//Declaración de direcciones MAC e IP
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1,20);

//Inicialización de la librería Ethernet puerto 80
EthernetServer server(80);

//FUNCION PRINCIPAL
void setup() {
  Ethernet.begin(mac, ip); //Inicializamos la conexión Ethernet y Servidor
  server.begin();

  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
  //servomotor
  myservo.attach(8); //Pin8 para Servomotor
  myservo.write(pos); //Servomotor en la posición 'pos'
  delay(15);
}

//FUNCION CICLICA
void loop() {
  int sensorReading;
  EthernetClient client = server.available(); //Creamos un cliente Web
  //Cuando detecte un cliente a través de una petición HTTP
  if (client) {

    boolean currentLineIsBlank = true; //Petición HTTP acaba con línea en blanco
    String cadena=""; //Creamos una cadena de caracteres vacía

    while (client.connected()) {
      if (client.available()) {
```

```

char c = client.read(); //Leemos la petición HTTP carácter por carácter
Serial.write(c); //Visualizamos la petición HTTP por el Monitor Serial
cadena.concat(c); //Unimos el String 'cadena' con la petición HTTP(c). Asi se
convierte la petición HTTP a String

int posicion=cadena.indexOf("POS="); //Guarda la posición "POS=" a la variable
'posicion'

if(cadena.substring(posicion)== "POS=MAS") //Si 'posicion' hay "POS=MAS"
{
if (pos<180) pos+=5;
myservo.write(pos); //Manda al servomotor a posicion de variable 'pos'
}
if(cadena.substring(posicion)== "POS=MENOS") //Si 'posicion' hay
"POS=MENOS"
{
if (pos>0) pos-=5;
myservo.write(pos); //Manda al servomotor a posicion de variable 'pos'
}
if(cadena.substring(posicion)== "POS=5") //Si 'posicion' hay "POS=CERO"
{
if (pos) pos=5;
myservo.write(pos); //Manda al servomotor a posicion de variable 'pos'
}
if(cadena.substring(posicion)== "POS=180")//Si 'posicion' hay "POS=CERO"
{
if (pos) pos=180;
myservo.write(pos); //Manda al servomotor a posicion de variable 'pos'
}
if(cadena.substring(posicion)== "POS=90")//Si 'posicion' hay "POS=CERO"
{
if (pos) pos=90;
myservo.write(pos); //Manda al servomotor a posicion de variable 'pos'
}

//Cuando recibe línea en blanco, la petición HTTP ha acabado y el servidor Web está
listo para enviar una respuesta
if (c == '\n' && currentLineIsBlank){

// Enviamos al cliente una respuesta HTTP
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();

//Página web en formato HTML
//Cabecera HTTP estándar

```

```

client.println("<body bgcolor=\"#FFFF99\">");
client.println("<hr/>");
client.println("<h1><center><img                                border=\"2\"
src=\"http://carlosordonezidrovo.info/wp-content/uploads/sites/389/2015/09/LOGO-
UPSp-10x3cm.jpg\" /></center></h1>");
client.println("<hr/>");

client.println("<center>");
client.println("<h1>PROYECTO DE TITULACION");
client.print("<br>");

client.println("<html>");
client.println("<head>");
client.println("</head>");
client.println("<body>");
client.println("<h1 align='center'>Control de Giro del Servomotor</h3>");

//Se crean botones
//Para enviar parametros a través de HTML se utiliza el metodo URL encode. Los
parámetros se envían a través del símbolo '?'
client.println("<div style='text-align:center;'>");
client.println("<button                                onClick=location.href='./?POS=MAS\'
style='margin:auto;background-color: #FF3300;color: snow;padding: 10px;border: 3px
solid #3F7CFF;width:65px;'>");
client.println("+ 5 Grados");
client.println("</button>");
client.println("<button                                onClick=location.href='./?POS=MENOS\'
style='margin:auto;background-color: #FF3300;color: snow;padding: 10px;border: 3px
solid #3F7CFF;width:65px;'>");
client.println("- 5 Grados");
client.println("</button>");
client.println("</button>");

client.println("<br /><br />");

client.println("<button                                onClick=location.href='./?POS=5\'
style='margin:auto;background-color: #CC0000;color: snow;padding: 15px;border:
3px solid #3F7CFF;width:75px;'>");
client.println("5 Grados");
client.println("</button>");
client.println("</button>");
client.println("</button>");
client.println("<button                                onClick=location.href='./?POS=90\'
style='margin:auto;background-color: #CC0000;color: snow;padding: 15px;border:
2px solid #3F7CFF;width:75px;'>");
client.println("90 Grados");
client.println("</button>");

```



```

        client.println("</button>");
        client.println("</button>");
        client.println("<button                                onClick=location.href='./?POS=180\'
style='margin:auto;background-color: #CC0000;color: snow;padding: 15px;border:
2px solid #3F7CFF;width:75px;'>");
        client.println("180 Grados");
        client.println("</button>");

        client.println("<br />");
        client.println("<body>");
        client.println("<h1 align='center'>Posicion actual = ");
        client.print(pos);
        client.println(" </b><br />");
        client.println("</b></body>");
        client.println("</html>");
        break;
    }}
}

//Tiempo al servidor para recibir datos
delay(1);
//Fin de la conexion:
client.stop();
}}
```

Anexo 4: Programación en Arduino de la Practica 4.

```
//UNIVERSIDAD POLITÉCNICA SALESIANA
//INGENIERÍA ELECTRÓNICA
//PROYECTO TÉCNICO – PONCE-SANTILLAN
//PRÁCTICA #4: MONITOREO DE SENSOR ULTRASONICO

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W5100
#include <Ethernet.h> //Librería Ethernet para manejar el Shield
#include <Ultrasonic.h>> //Librería para Sensor de Proximidad
#include <Servo.h> //Enlace con la libreria servo.h

//Definir parametros del Sensor HC-SR04
#define TRIGGER 6 // Trigger del sensor
#define ECHO 7 // Echo del sensor

//Inicializacion del Sensor ultrasonico
Ultrasonic ultrasonico (TRIGGER,ECHO);

Servo servo1; //Declaracion de variable tipo servo para controlar el servomotor
Servo servo2;
int pos1 = 180; //Declaracion de variable tipo entero de la posición del servomotor
int pos2 = 180;

String estado; //estado para la puerta.

//Declaracion de IP, mascara de Red y Gateway
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress IP(192,168,1,20); //Asignacion de IP
IPAddress Gateway(192,168,1,1); //Asignacion de Gateway
IPAddress Subnet(255, 255, 255, 0); //Asignacion de Máscara de red

//Inicio del servidor puerto 80
EthernetServer server(80);

//FUNCION PRINCIPAL
void setup(){
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin(); // conexiones de entrantes
  Serial.println(Ethernet.localIP()); //Obtiene la dirección IP

  //Servomotor
  servo1.attach(2); //Conectar el servomotor al pin 2
  servo2.attach(4); //Conectar el servomotor al pin 4
  servo1.write(pos1); //Manda al servomotor a la posicion de la variable 'pos'
  servo2.write(pos2);
```

```

}

//FUNCION CICLICA
void loop(){
float cmMsec;//Tipo de datos de números
long microsec = ultrasonico.timing();//Lectura del sensor
cmMsec = ultrasonico.convert(microsec, Ultrasonic::CM); //Convertir la distancia en
cm

EthernetClient client = server.available();// Esperar conexión de navegador
if (client){
  Serial.println("new client");
  boolean currentLineIsBlank = true; //petición HTTP acaba con línea en blanco
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      Serial.write(c);
      // Si ha llegado al final de la línea (recibido una nueva línea
      // carácter y la línea está en blanco, la petición HTTP ha terminado,
      // por lo que puede enviar una respuesta
      if (c == '\n' && currentLineIsBlank){

//Enviar un encabezado de respuesta HTTP estándar
      client.println("HTTP/1.1 200 OK");
      client.println("Content-Type: text/html");
//La conexión se cierra después de la finalización de la respuesta
      client.println("Connection: close");
      client.println("Refresh: 1"); //Actualizar pagina cada 1seg
      client.println();
      client.println("<!DOCTYPE HTML>");
      client.println("<html>");

      client.println("<body bgcolor=#FFFF99>");
      client.println("<hr/>");
      client.println("<hr/>");
      client.println("<h4><center><img
                                                                    border=2\"
src=http://carlosordonezidrovo.info/wp-content/uploads/sites/389/2015/09/LOGO-
UPSp-10x3cm.jpg\" /></center></h4>");
      client.println("<hr/>");
      client.println("<hr/>");
      client.println("<title>UNIVERSIDAD</title>");
      client.println("</head>");
      client.println("<body width=150% height=150%>");
      client.println("<center>");
      client.println("<h1>PROYECTO DE TITULACION");
      client.println("<br/>");

```

```

client.print("<font color=#0000FF><b><u>");
client.print("SENSOR ULTRASONICO 1");
client.print("</u></b></font>");
client.println("<br />");
client.println("<br />");

//Mostrar datos de medición del sensor
client.print("Distancia Sensor Ultrasonico: ");
client.print("<b>");
client.print("<font color=#4B9C4A><b><u>");
client.print(cmMsec);
client.print("cm");
client.println("</b></html>");
client.println("<br />");
client.println("<br />");
client.print("<font color=#0000FF><b><u>");
client.print("PUERTA: ");
client.print("<b>");
client.print("<font color=#e70c23><b><u>");
client.print(estado);
client.println("<br />");
client.println("<br />");

if (cmMsec< 30) pos1 = 0 ;
if (cmMsec> 30) pos1 = 180 ;
servo1.write(pos1); //Manda al servomotor a la posicion de la variable 'pos'
if (cmMsec< 30) pos2 = 180 ;
if (cmMsec> 30) pos2 = 0 ;
servo2.write(pos2); //Manda al servomotor a la posicion de la variable 'pos'

if(cmMsec<30)
estado ="OPEN" ;

if (cmMsec>30)
estado ="CLOSE" ;

client.println("</html>");
break;
}
if (c == '\n') {
// Vas a empezar una nueva línea
currentLineIsBlank = true;
} else if (c != '\r') {
// Que ha llegado a un false en la línea actual
currentLineIsBlank = false;
}
}
}

```

```
}  
//Tiempo al navegador para recibir datos  
delay(1);  
  
//Fin de la Conexion  
client.stop();  
Serial.println("client disconnected");  
}}
```

Anexo 5: Programación en Arduino de la Practica 5.

```
//UNIVERSIDAD POLITÉCNICA SALESIANA
//INGENIERÍA ELECTRÓNICA
//PROYECTO TÉCNICO
//PRÁCTICA #5 CONTROL DE BOMBILLAS

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W5100
#include <Ethernet.h> //Librería Ethernet para manejar el Shield

//Declaración de Pines a usar en la tarjeta Arduino
int PIN_FOCO1=8;
int PIN_FOCO2=7;

//Declaración de direcciones MAC e IP
byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED};
IPAddress ip(192,168,1,20);

//Inicialización de la librería Ethernet puerto 80
EthernetServer servidor(80);

String readString=String(30); //Lee los caracteres de una secuencia en una cadena.

//Los strings se representan como arrays de caracteres (tipo char)
String state1;
String state2;

//FUNCION PRINCIPAL
void setup(){
  Serial.begin(9600); //Inicializamos la conexión Ethernet y el servidor
  Ethernet.begin(mac, ip); //Inicializamos con las direcciones asignadas
  servidor.begin();

  pinMode(PIN_FOCO1,OUTPUT);
  digitalWrite(PIN_FOCO1,HIGH);

  pinMode(PIN_FOCO2,OUTPUT);
  digitalWrite(PIN_FOCO2,HIGH);
}

//FUNCION PRINCIPAL
void loop(){

//Se crea un cliente que se puede conectar a IP
EthernetClient cliente= servidor.available();
```

```

//Cliente conectado
if(cliente){
boolean lineaenblanco=true;
while(cliente.connected()) {
    if(cliente.available()) {
char c=cliente.read();
//Almacenar los caracteres en la variable readString
if(readString.length()<30){
readString.concat(c);
}

if(c=='\n' && lineaenblanco) //Si la petición HTTP ha finalizado
{
int FOCO = readString.indexOf("LUZ=");
if(readString.substring(FOCO,FOCO+5)=="LUZ=A") {
digitalWrite(PIN_FOCO1,LOW);

state1="ON"; }
if(readString.substring(FOCO,FOCO+5)=="LUZ=C") {
digitalWrite(PIN_FOCO2,LOW);
state2="ON"; }
else if (readString.substring(FOCO,FOCO+5)=="LUZ=B") {
digitalWrite(PIN_FOCO1,HIGH);

state1="OFF";}
else if (readString.substring(FOCO,FOCO+5)=="LUZ=D") {
digitalWrite(PIN_FOCO2,HIGH);
state2="OFF";}

//Cabecera HTTP estándar
cliente.println("<body bgcolor=#FFFF99>");
cliente.println("<center><img border=2 src=http://carlosordonezidrovo.info/wp-content/uploads/sites/389/2015/09/LOGO-UPSp-10x3cm.jpg /></center>");

cliente.println("<title>UNIVERSIDAD ON/OFF</title>");
cliente.println("</head>");
cliente.println("<body width=100% height=50%>");
cliente.println("<center>");
cliente.println("<h1>PROYECTO DE TITULACION");

cliente.println("<h1>Control de Estado de Bombillos</h1>");
cliente.print("<font color=red>");
cliente.print("BOMBILLO 1:");
cliente.print("<br>");
cliente.print("ESTADO:");
cliente.print(state1);
cliente.print("<br>");

```

```

cliente.println("<input type=submit value=ON style=width:300px;height:75px
onClick=location.href='./?LUZ=A\>");
cliente.println("<input type=submit value=OFF style=width:300px;height:75px
onClick=location.href='./?LUZ=B\>");
cliente.print("<br><br>");
cliente.print("<font color=blue>");
cliente.print("BOMBILLO 2: ");
cliente.print("<br>");
cliente.print("ESTADO: ");
cliente.print(state2);
cliente.print("<br>");
cliente.println("<input type=submit value=ON style=width:300px;height:75px
onClick=location.href='./?LUZ=C\>");
cliente.println("<input type=submit value=OFF style=width:300px;height:75px
onClick=location.href='./?LUZ=D\>");
cliente.println("</center>");
cliente.println("</body>");
cliente.println("</html>");

//Finalizacion Conexión con Cliente
cliente.stop();
readString="";
}}
}
}}

```


Anexo 6: Programación en Arduino de la Practica 6.

```
//UNIVERSIDAD POLITÉCNICA SALESIANA
//INGENIERÍA ELECTRÓNICA
//PROYECTO TÉCNICO – PONCE-SANTILLAN
//PRÁCTICA #6: BOMBILLAS-BLUETOOTH
//SERVIDOR

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W5100
#include <Ethernet.h> //Librería Ethernet para manejar el Shield

//Declaramos constantes para la recepción de estados de pines de entrada, deben de ser
las mismas que en la otra tarjeta
const byte BOMBILLA1_ON = 0x41;
const byte BOMBILLA1_OFF = 0x61;
const byte BOMBILLA2_ON = 0x42;
const byte BOMBILLA2_OFF = 0x62;

//Declaración de Señales Hardware
int BOMBILLA1 = 2, BOMBILLA2 = 3;

//Declaración de Parámetros de red
byte mac[] = {0x6E, 0xFD, 0xE5, 0x23, 0x0C, 0x9F};
IPAddress ipLocal(192,168,1,20);
IPAddress puertaEnlace (192,168,2,1);
IPAddress mascaraSubRed (255,255,255,0);

//Creamos un servidor y asignamos el puerto 23
EthernetServer servidor = EthernetServer(23);

//FUNCION PRINCIPAL
void setup() {
  // Configuración de Señales Hardware
  Serial.begin(9600);

  pinMode(BOMBILLA1, OUTPUT); //Se activara si se activa la entrada 1 de la otra
tarjeta
  pinMode(BOMBILLA2, OUTPUT); //Se activara si se activa la entrada 2 de la otra
tarjeta

  // Configuración de la red del dispositivo
  Ethernet.begin(mac, ipLocal, puertaEnlace, mascaraSubRed);

  // Iniciamos el servidor
  servidor.begin();
}
```

```

//FUNCION CICLICA
void loop() {
  if (servidor.available()) {
    //Si hay datos disponibles los asignamos a un cliente
    EthernetClient cliente = servidor.available();

    //Leemos los datos
    byte dato = cliente.read();

    // Según dato recibido actuaran las salidas
    if (dato == BOMBILLA1_ON) digitalWrite(BOMBILLA1,HIGH);
    if (dato == BOMBILLA1_OFF) digitalWrite(BOMBILLA1,LOW);
    if (dato == BOMBILLA2_ON) digitalWrite(BOMBILLA2,HIGH);
    if (dato == BOMBILLA2_OFF) digitalWrite(BOMBILLA2,LOW);

    //Devolvemos un eco al cliente para vaciar
    cliente.write(dato);
  }
}

```

```

//UNIVERSIDAD POLITÉCNICA SALESIANA
//INGENIERÍA ELECTRÓNICA
//PROYECTO TÉCNICO
//PRÁCTICA #6 BOMBILLAS-BLUETOOTH
//CLIENTE

#include <SPI.h> //Librería que comunica al Shield con el Microcontrolador W5100
#include <Ethernet.h> //Librería Ethernet para manejar el Shield

//Declaramos constantes para envío de estados de pines de entrada
const byte BOMBILLA1_ON = 0x41;
const byte BOMBILLA1_OFF = 0x61;
const byte BOMBILLA2_ON = 0x42;
const byte BOMBILLA2_OFF = 0x62;

//Declaración de Señales Hardware
char val;

// Declaración de Parámetros de red
byte mac[] = {0xDE, 0xFD, 0xE5, 0xD3, 0x0C, 0x9A };
IPAddress ipLocal(192,168,1,25);
IPAddress puertaEnlace (192,168,2,1);
IPAddress mascaraSubRed (255,255,255,0);

```

```

// IP de la otra tarjeta
IPAddress ipArduServ(192,168,1,20);

//Creamos un cliente
EthernetClient cliente;

//FUNCION PRINCIPAL
void setup(){
  Serial.begin(9600); // Aquí establecemos la velocidad

  // Configuración de la red del dispositivo
  Ethernet.begin(mac, ipLocal, puertaEnlace, mascaraSubRed);
}

//FUNCION CICLICA
void loop(){
  while (!cliente.connect(ipArduServ, 23))
  {
    //Se espera obtener conexión con IP de la otra tarjeta por el Puerto 23
    if( Serial.available() )
      val = Serial.read();

    if( val == '0' )
      cliente.write (BOMBILLA1_ON);
      delay (200);
    if( val == '1' )
      cliente.write(BOMBILLA1_OFF);
      delay (200);

    if( val == '2' )
      cliente.write (BOMBILLA2_ON);
      delay (200);
    if( val == '3' )
      cliente.write(BOMBILLA2_OFF);
      delay (200);
  }
}

```

Anexo 7: Diseño de la interfaz, en App Inventor 2, para control de la Practica 6.

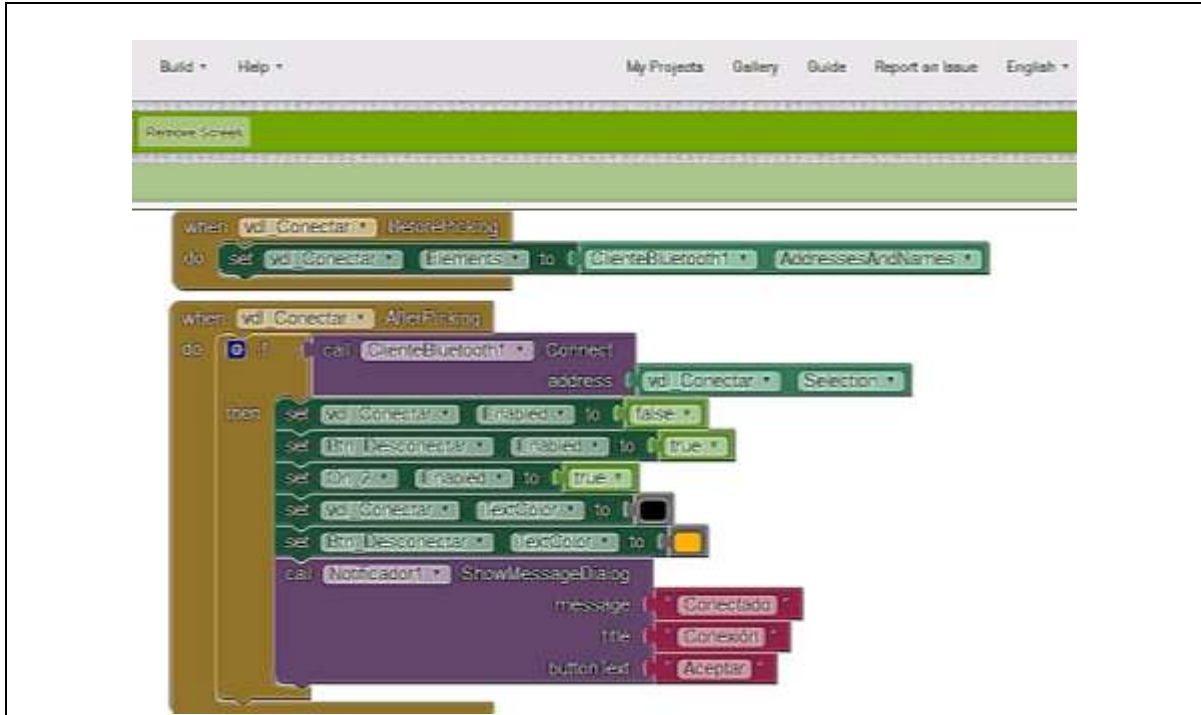


Diagrama de bloque para la “Conexión” al Bluetooth Arduino.

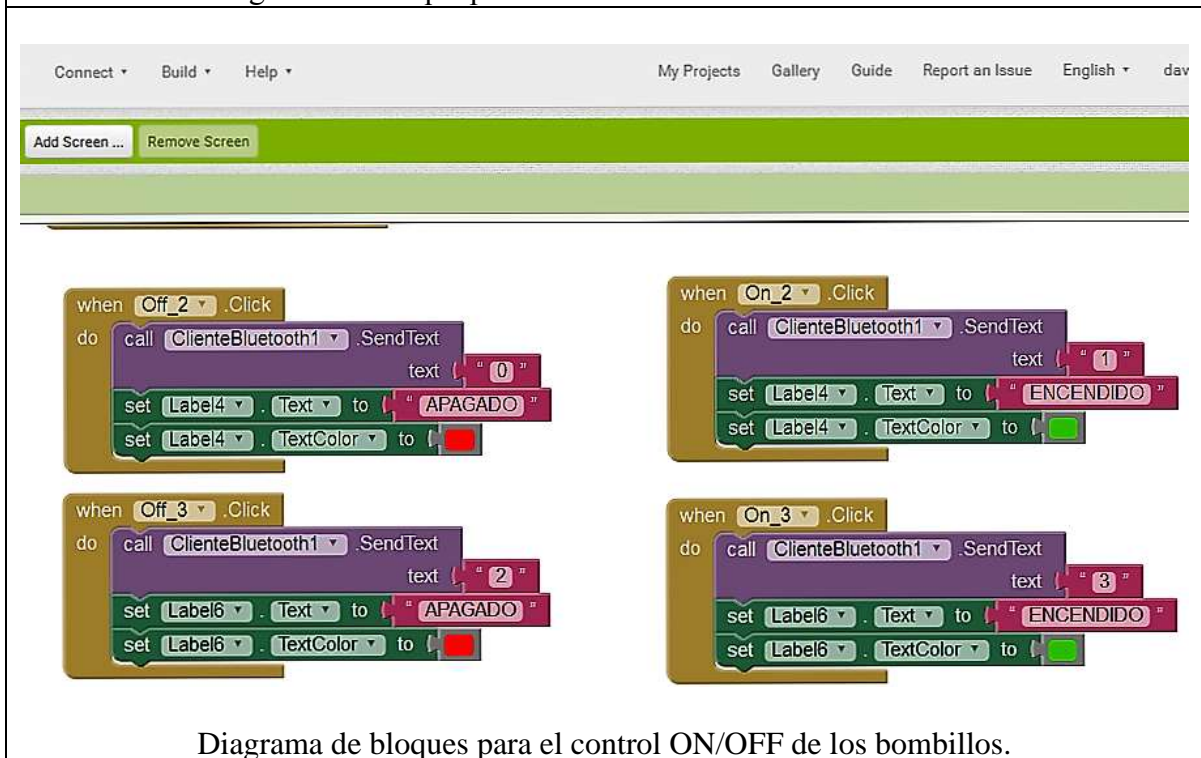


Diagrama de bloques para el control ON/OFF de los bombillos.



Diagrama de bloques para la “Desconexión” del Bluetooth Arduino.



Aplicación final diseñada para el control ON/OFF de los bombillos.

Anexo 8. Adaptador Powerline TL-PA4010, Características y Especificaciones técnicas.

TP-LINK®

AV500 Nano Powerline Adapter TL-PA4010

⦿ Features:

- Data transfer rate up to 500Mbps over electrical wires, ideal for HD video streaming
- Nano Size with a total new exterior
- No new wires or drilling required, just using existing electrical wires
- No setup required, simply plug and play
- Up to 300 meters range over the household power circuit for better performance through walls or across floors
- Power-Saving Mode automatically reduces power consumption by up to 85%*
- 128-bit AES encryption easily at a push of "Pair" Button
- QoS provides networking flexibility and stable IPTV streaming and online gaming
- A wide range of plug type including EU, UK, and US

* Actual data will vary because of the network conditions and environment factors.



⦿ Description:

TP-LINK's 500Mbps Nano Powerline Adapter, TL-PA4010 turns your existing powerline into a high speed network with no need for new wire or drilling. No configuration required, simply plug your adapters into your power sockets and you can establish networking infrastructure in a flash. With a speed of up to 500Mbps, TL-PA4010 is a great choice for an easy to build multimedia entertainment network.

www.tp-link.com

⦿ Specifications:

Hardware Features	
Standards and Protocols	HomePlug AV, IEEE802.3, IEEE802.3u
Interface	1* 10/100Mbps Ethernet Port
Plug Type	EU
Button	Pair Button
LED Indicator	PWR, PLC, ETH
Dimensions (W x D x H)	2.6 x 2.0 x 1.1in. (65 x 52 x 28.5mm)
Power Consumption	< 2.0W
Range	300meters in house
Software Features	
Modulation Technology	OFDM
Encryption	128-bit AES Encryption
Others	
Certifications	CE, FCC, RoHS
System Requirements	Windows 8/7/Vista/XP/2000, Mac, Linux *
Environment	Operating Temperature: 0°C~40°C (32°F~104°F)
	Storage Temperature: -40°C~70°C (-40°F~158°F)
	Operating Humidity: 10%~90% non-condensing
	Storage Humidity: 5%~90% non-condensing

* Utility supports Windows 7/Vista/XP/2000 operating system for the present.

⦿ Diagram:



Package:

- AV500 Nano Powerline Adapter TL-PA4010
- 2-meter RJ-45 Ethernet Cable
- Quick Installation Guide
- Resource CD

* Powerline adapters should be deployed on the same circuit and in set of two or more. The first, connected to the router and others located elsewhere to enable connectivity in those locations.

Anexo 9: Extensor Powerline TL-WPA4220, características y especificaciones técnicas.

TP-LINK®

Extensor de Rango Wi-Fi de 300Mbps, Edición Powerline AV500

TL-WPA4220KIT

⊙ Características:

- El Botón Wi-Fi Clone simplifica su configuración Wi-Fi y ayuda a crear una red doméstica unificada sin interrupciones
- La norma HomePlug AV que proporciona velocidades de transferencia de datos a alta velocidad de hasta 500Mbps a través del cableado eléctrico existente de su casa, ideal para el streaming de video en HD o 3D y juegos en línea sin retrasos
- Extiende las conexiones inalámbricas de 300Mbps a las áreas más difíciles de acceder de su casa y oficina
- Los dos puertos Ethernet le permiten conectar a Internet sus TVs, consolas de juegos o PCs
- Un rango de hasta 300 metros a través del circuito eléctrico del hogar
- Fácil de Instalar, sólo Conecte y Use (Plug and Play)
- Fácil Encriptación AES de 128-bit con sólo presionar el Botón "Pair"
- Soporte con la transmisión IP de multicast IGMP, que optimiza el streaming de IPTV



⊙ Descripción:

El Extensor de Rango Wi-Fi de 300Mbps, Edición Powerline AV500, el TL-WPA4220KIT de TP-LINK extiende su conexión de Internet a cada cuarto de la casa a través del circuito eléctrico existente de su casa. El TL-WPA4220 cuenta con el botón Wi-Fi Clone que activa la Súper Extensión de Rango lo cual significa que puede copiar automáticamente el SSID y contraseña de su router. De esta forma, el TL-WPA4220 simplifica la configuración de su Wi-Fi y permite el roaming sin interrupciones dentro de la red de su casa.

www.tp-link.com

Extensor de Rango Wi-Fi de 300Mbps, Edición Powerline AV500 TL-WPA4220KIT

● Especificaciones:

Características del Hardware	
Estándares y Protocolos	HomePlug AV, IEEE802.3, IEEE802.3u, IEEE802.11b/g/n
Interfaz	Puertos Ethernet de 10/100Mbps
Tipo de Enchufe	AP
Botón	Pair, Reset (Reinicio), Wi-Fi/Wi-Fi Clone
Indicador LED	PWR (Encendido), PLC, ETH, Wi-Fi/Wi-Fi Clone
Dimensiones (Largo x Ancho x Alto)	3.7 x 2.1 x 1.6 pulg. (94 x 54 x 40mm)
Consumo de Energía	<6W
Rango	300 metros sobre el circuito eléctrico
Características del Software	
Tecnología de Modulación	OFDM (PLC)
Encriptación	Seguridad Powerline: 128-bit AES Seguridad Inalámbrica: Encriptación WEP, WPA/WPA2, WPA-PSK/WPA2-PSK
Otras	
Certificaciones	CE, FCC, RoHS
Requisitos del Sistema	Windows 8/7/Vista/XP/2000, Mac OS, Linux*
Condiciones Ambientales	Temperatura de Operación: 0°C–40°C (32°F–104°F)
	Temperatura de Almacenamiento: –40°C–70°C (–40°F–158°F)
	Humedad de Funcionamiento: 10%–90% sin condensación Humedad de Almacenamiento: 5%–90% sin condensación

* La Utilidad Soporta el Sistema Operativo Windows 8/7/Vista/XP/2000 hasta el momento.

● Diagrama:

ICONECTE Y USE CON EL CLON WI-FI DE TP-LINK!



> Conecte el TL-WPA4220 cerca de su router.
> Presione el botón Wi-Fi en el TL-WPA4220.
> Presione el botón WPS en el router.
¡Ya está! ¡Ahora el TL-WPA4220 ha copiado el SSID y la Contraseña de su router y puede disfrutar de una red doméstica unificada!

RED DOMÉSTICA UNIFICADA



* El Extensor de Rango Wi-Fi Powerline necesita ser conectado en el mismo circuito eléctrico.

Paquete:

- Extensor de Rango WiFi de 300Mbps, Edición Powerline AV500 TL-WPA4220 & TL-PA4010
- Cable Ethernet RJ-45 de 2 metros * 2
- Guía de Rápida instalación
- CD de Recursos

Las especificaciones están sujetas a cambios sin previo aviso. © 2014 de una marca registrada de TP-LINK Technology Co., Ltd.
Este manual y producto no pueden ser usados, copiados o distribuidos sin el consentimiento explícito de la compañía productora.
Impreso en: abril de 2014 @ TP-LINK Technology Co., Ltd. Todos los derechos reservados.

www.tp-link.com

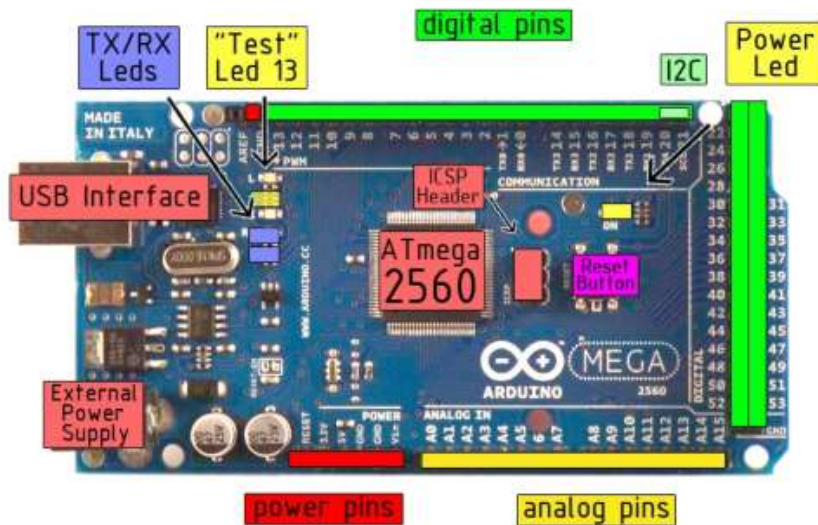
Anexo 10: Datasheet de la tarjeta Arduino MEGA 2560

Technical Specification

EAGLE files: [_arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary	
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Anexo 11: Datasheet del Módulo Bluetooth Arduino HC-06

Guangzhou HC Information Technology Co., Ltd.

1. Product's picture

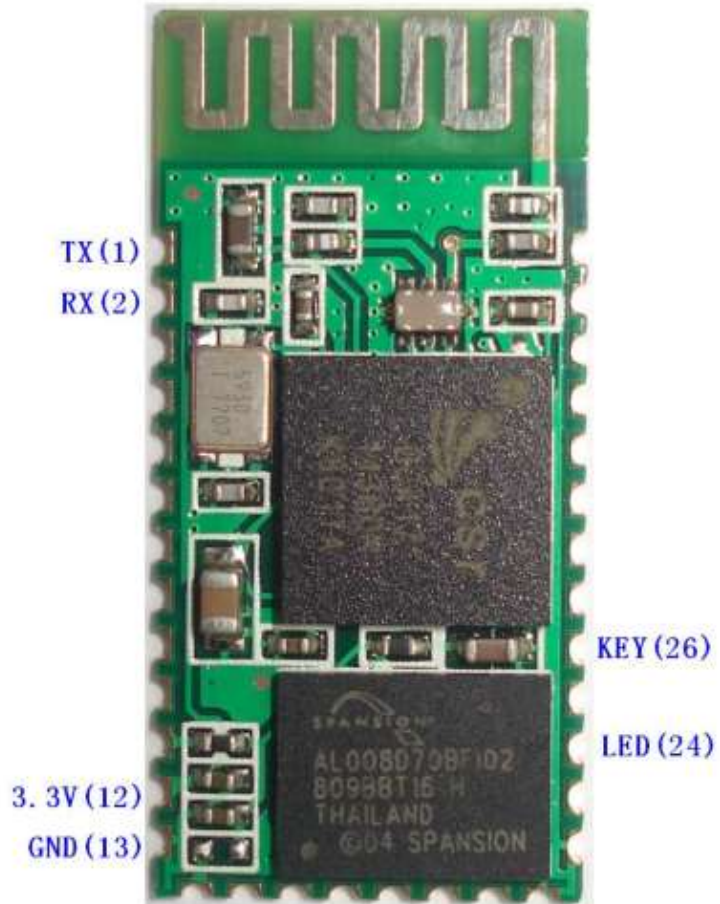


Figure 1 A Bluetooth module

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

- Application fields:
 - Bluetooth Car Handsfree Device
 - Bluetooth GPS
 - Bluetooth PCMCIA , USB Dongle
 - Bluetooth Data Transfer
- Software
 - CSR

3. PINs description

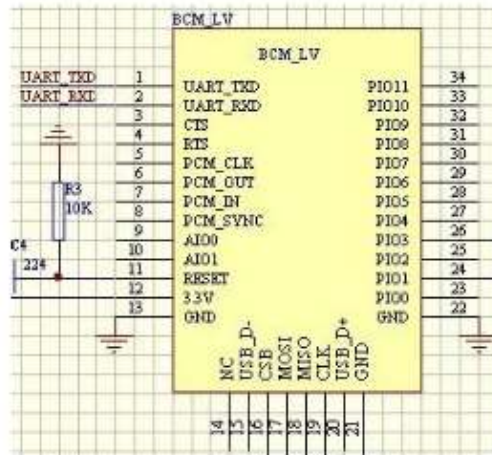


Figure 3 PIN configuration

The PINs at this block diagram is as same as the physical one.

PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
1V8	14	VDD	Integrated 1.8V (+) supply with On-chip linear regulator output within 1.7-1.9V	
VCC	12	3.3V		
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	CLK_REQ
PIO7	30	Bi-Directional	Programmable input/output line	CLK_OUT
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	
RESETB	11	CMOS Input with weak internal pull-down		
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
SPI_CSB	16	CMOS input with weak internal	Chip select for serial peripheral interface, active low	

Anexo 12: Datasheet del Sensor Ultrasónico HC-SR04



Tech Support: services@electfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
 - (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
 - (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
- Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

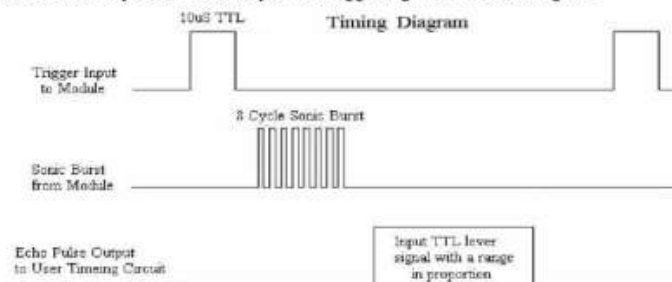
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Anexo 13: Datasheet del Sensor de Temperatura LM-35

LM35 Precision Centigrade Temperature Sensors

FEATURES

- Calibrated Directly in ° Celsius (Centigrade)
- Linear + 10 mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at +25°C)
- Rated for Full -55°C to +150°C Range
- Suitable for Remote Applications
- Low Cost Due to Wafer-Level Trimming
- Operates from 4 to 30 V
- Less than 60-µA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Nonlinearity Only ±¼°C Typical
- Low Impedance Output, 0.1 Ω for 1 mA Load

DESCRIPTION

The LM35 series are precision integrated-circuit temperature sensors, with an output voltage linearly proportional to the Centigrade temperature. Thus the LM35 has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±½°C over a full -55°C to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The low output impedance, linear output, and precise inherent calibration of the LM35 make interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 is rated to operate over a -55°C to +150°C temperature range, while the LM35C is rated for a -40°C to +110°C range (-10° with improved accuracy). The LM35 series is available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

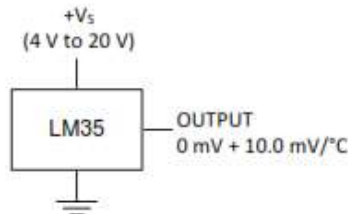
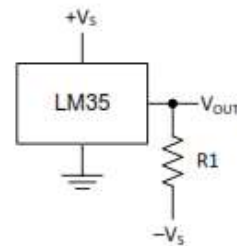



Figure 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)



Choose $R_1 = -V_S / 50 \mu\text{A}$
 $V_{OUT} = 1500 \text{ mV at } 150^\circ\text{C}$
 $V_{OUT} = 250 \text{ mV at } 25^\circ\text{C}$
 $V_{OUT} = -550 \text{ mV at } -55^\circ\text{C}$

Figure 2. Full-Range Centigrade Temperature Sensor

 Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet. All trademarks are the property of their respective owners.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

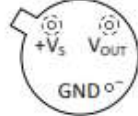
Copyright © 1989–2013, Texas Instruments Incorporated



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

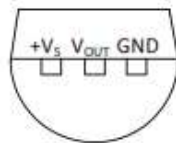
CONNECTION DIAGRAMS

METAL CAN PACKAGE TO (NDV)

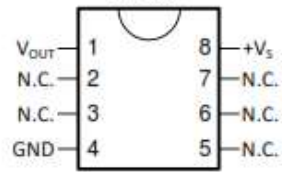


Case is connected to negative pin (GND)

PLASTIC PACKAGE TO-92 (LP) BOTTOM VIEW

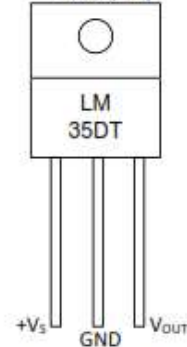


SMALL-OUTLINE MOLDED PACKAGE SOIC-8 (D) TOP VIEW



N.C. = No connection

PLASTIC PACKAGE TO-220 (NEB)



Tab is connected to the negative pin (GND).

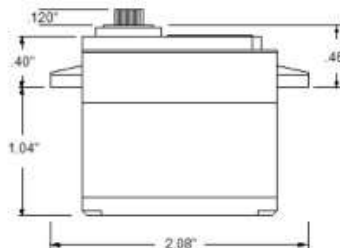
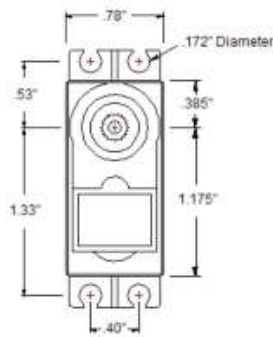
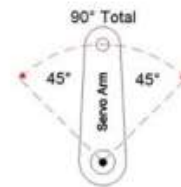
NOTE: The LM35DT pinout is different than the discontinued LM35OP

Anexo 14: Especificaciones del Servomotor Hitec HS-311

HS - 311 Standard

Detailed Specifications

Control System: +Pulse Width Control 1500usec Neutral
Required Pulse: 3-5 Volt Peak to Peak Square Wave
Operating Voltage: 4.8-6.0 Volts
Operating Temperature Range: -20 to +60 Degree C
Operating Speed (4.8V): 0.19sec/60° at no load
Operating Speed (6.0V): 0.15sec/60° at no load
Stall Torque (4.8V): 42 oz/in (3.0 kg/cm)
Stall Torque (6.0V): 49 oz/in (4.5 kg/cm)
Current Drain (4.8V): 7.4mA/idle, 160mA no load operating
Current Drain (6.0V): 7.7mA/idle, 180mA no load operating
Dead Band Width: 5usec
Operating Angle: 40° one side pulse traveling 400usec
Direction: Clockwise/Pulse Traveling 1500 to 1900usec
Motor Type: Cored Metal Brush
Potentiometer Drive: 4 Slider/Direct Drive
Bearing Type: Top/Resin Bushing
Gear Type: Nylon
360 Modifiable: [Yes](#)
Connector Wire Length: 11.81" (300mm)
Weight: 1.52oz (43g)



Anexo 15: Elaboración de módulos didácticos para las diferentes prácticas del Proyecto.

- Módulo de equipos Power Line Communications:
 - *Adaptador TL-PA4010*

 <p>Cableado interno de tomacorrientes.</p>	 <p>Vista superior.</p>
 <p>Vista delantera / trasera</p>	 <p>Vista lateral.</p>

○ Extensor TL-WPA4220



Vista superior.



Vista trasera.



Vista lateral.



Vista posterior.

- *Módulo de tarjetas Arduino: Mega2560 y Ethernet Shield*



Vista superior.



Vista trasera.



Vista lateral.



Vista trasera

- **Módulo de la Práctica 1 y Práctica 2.**

- *Consta de:*

- *Luces pilotos (3)*
- *Sensor de temperatura LM-35*
- *Módulo de relés de 12V*



Vista interna, conexiones.



Vista frontal.



Vista trasera.



- **Módulo de la práctica 3 y 4.**
 - *Consta de:*
 - *Servomotor*
 - *Sensor ultrasónico*



- **Módulo de la práctica 5 y práctica 6.**
 - *Conformado de:*
 - *Bombillas (2)*
 - *Módulo de relés de 12V.*



Vista superior, conexiones internas.



Vista frontal.



Vista frontal.



Mini placas (baquelitas) para conexión de 120VAC.

Anexo 16: GLOSARIO

ADSL: Asymmetric Digital Subscriber Line. Tecnología que transforma las líneas telefónicas normales en líneas digitales de alta velocidad aumentando considerablemente la velocidad de conexión a internet.

App Inventor: App Inventor te permite desarrollar aplicaciones para los teléfonos Android mediante un navegador web y, o bien un teléfono conectado o el emulador. Los servidores de App Inventor almacenan tu trabajo y te ayudarán a realizar el seguimiento de tus proyectos.

Comunicación: acción y resultado de comunicarse, transmitir señales (voz, datos, escritos) entre dos o más individuos.

Encriptación AES: El estándar de cifrado (encriptación) avanzado AES, Advanced Encryption Standard, es uno de los algoritmos más seguros y más utilizados hoy en día, disponible para uso público. El algoritmo se basa en varias sustituciones, permutaciones y transformaciones lineales, ejecutadas en bloques de datos de 16 bytes - por lo que se le llama blockcipher. Estas operaciones se repiten varias veces, llamadas "rondas". En cada ronda, un único "roundkey" se calcula de la clave de encriptación, y es incorporado en los cálculos.

FTTH: Fiber to the Home. Es una simple y segura solución que elimina los actuales atascamientos o colapsos en la última milla, ya que es la parte de la red más importante.

Head-End: es el equipo de donde se provee la señal de datos y comienza la red de distribución a la red de baja tensión, y actúa como maestro.

IPTV: Televisión por Protocolo de Internet, denominación más común para los sistemas de distribución por suscripción de señales de televisión o vídeo usando conexiones de banda ancha sobre el protocolo IP. No es TV por internet.

Latencia: se refiere a los retardos temporales que se registran en una red, los cuales se producen por la demora en la propagación y en la transmisión de los paquetes de datos. Al sumarse todos estos retardos, se obtiene la latencia de la red informática.

Microfiltro: El microfiltro es un dispositivo que evita el ruido o estática durante la

conversación del teléfono optimizando el desempeño de la conexión a Internet. Si no los instalas, es posible que tu conexión a internet sea inestable, con altas probabilidades de falla.

Repetidores: equipo que actúa como esclavos del Head-End y también actúan como maestros de otros repetidores.

Señal: a nivel de comunicación, es una onda proveniente de un indicio lumínico, sonoro o eléctrico que porta información, lo cual genera el mensaje.

SSID: (Service Set Identifier) es un nombre incluido en todos los paquetes de una red inalámbrica para identificarlos como parte de esa red. El código consiste en un máximo de 32 caracteres, que la mayoría de las veces son alfanuméricos. Todos los dispositivos inalámbricos que intentan comunicarse entre sí deben compartir el mismo SSID.

Streaming: es la distribución digital de multimedia a través de una red de computadoras, de manera que el usuario consume el producto (vídeo o audio) en paralelo mientras se descarga. También se refiere a una corriente continuada, que fluye sin interrupción.

TCP: Protocolo de Control de Transmisión, es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. TCP es un protocolo orientado a conexión, es decir, que permite que dos máquinas que están comunicadas controlen el estado de la transmisión.

Tecnología: es la transformación y desarrollo de recursos en nuevos productos, para resolver un determinado problema y lograr satisfacer una necesidad de un individuo.

UDP: Protocolo de datagrama de usuario, es un protocolo no orientado a conexión de la capa de transporte del modelo TCP/IP. Este protocolo es muy simple ya que no proporciona detección de errores.

Unidad de almacenamiento NAS: Un dispositivo de almacenamiento conectado en red (NAS) es un dispositivo de almacenamiento conectado a una red que permite almacenar y ubicar los datos en un punto centralizado para usuarios autorizados de la red y multiplicidad de clientes. Los dispositivos NAS son flexibles y expansibles. Un dispositivo NAS es como tener una nube privada en la oficina.

Anexo 17: PRESUPUESTO

EQUIPO	CANTIDAD	PRECIO UNITARIO	TOTAL
Modem TP-Link	1	\$150	\$150
Kit TI-WPA4220 300MBPS Av500 Wifi Powerline Extender Starter	2	\$125	\$250
Arduino Ethernet shield w5100	2	\$65	\$130
Tarjeta Arduino MEGA 2560	2	\$70	\$140
Sensor ultrasónico	2	\$18	\$36
Sensor Temperatura	1	\$8	\$8
Servomotor	1	\$20	\$20
Arduino bluetooth HC-06	1	\$25	\$25
Cable	20 metros	\$1	\$20
Cable UTP	25 metros	\$0.75	\$18.75
RJ-45	8	\$0.50	\$4
Puerto RJ45 (hembra)	8	\$7	\$56
Cajas didácticas	9	\$30	\$270
VARIOS			\$200
TOTAL			\$1307.75