



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

TEMA:

**Estudio de un sistema de acoplamiento híbrido inteligente para energía solar
fotovoltaica y la red eléctrica de una vivienda**

AUTOR:

Peñañiel Caicedo, Joseph Adrián

Trabajo de titulación previo a la obtención del título de
INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO

TUTOR:

Mendoza Merchán, Eduardo Vicente

Guayaquil, Ecuador

12 de marzo de 2018



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación fue realizado en su totalidad por Sr. **Peñañiel Caicedo, Joseph Adrián** como requerimiento para la obtención del Título de **INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO**

TUTOR

Mendoza Merchán, Eduardo Vicente

DIRECTOR DE CARRERA

Heras Sánchez, Miguel Armando

Guayaquil, a los 12 días del mes de marzo del año 2018



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Peñañiel Caicedo, Joseph Adrián**

DECLARO QUE:

El Trabajo de Titulación, **Estudio de un sistema de acoplamiento híbrido inteligente para energía solar fotovoltaica y la red eléctrica de una vivienda** previo a la obtención del Título de **Ingeniero Electrónico en Control y Automatismo**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 12 días del mes de marzo del año 2018

EL AUTOR

PEÑAÑIEL CAICEDO, JOSEPH ADRIÁN



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

AUTORIZACIÓN

Yo, Peñafiel Caicedo, Joseph Adrián

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Estudio de un sistema de acoplamiento híbrido inteligente para energía solar fotovoltaica y la red eléctrica de una vivienda**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 12 días del mes de marzo del año 2018

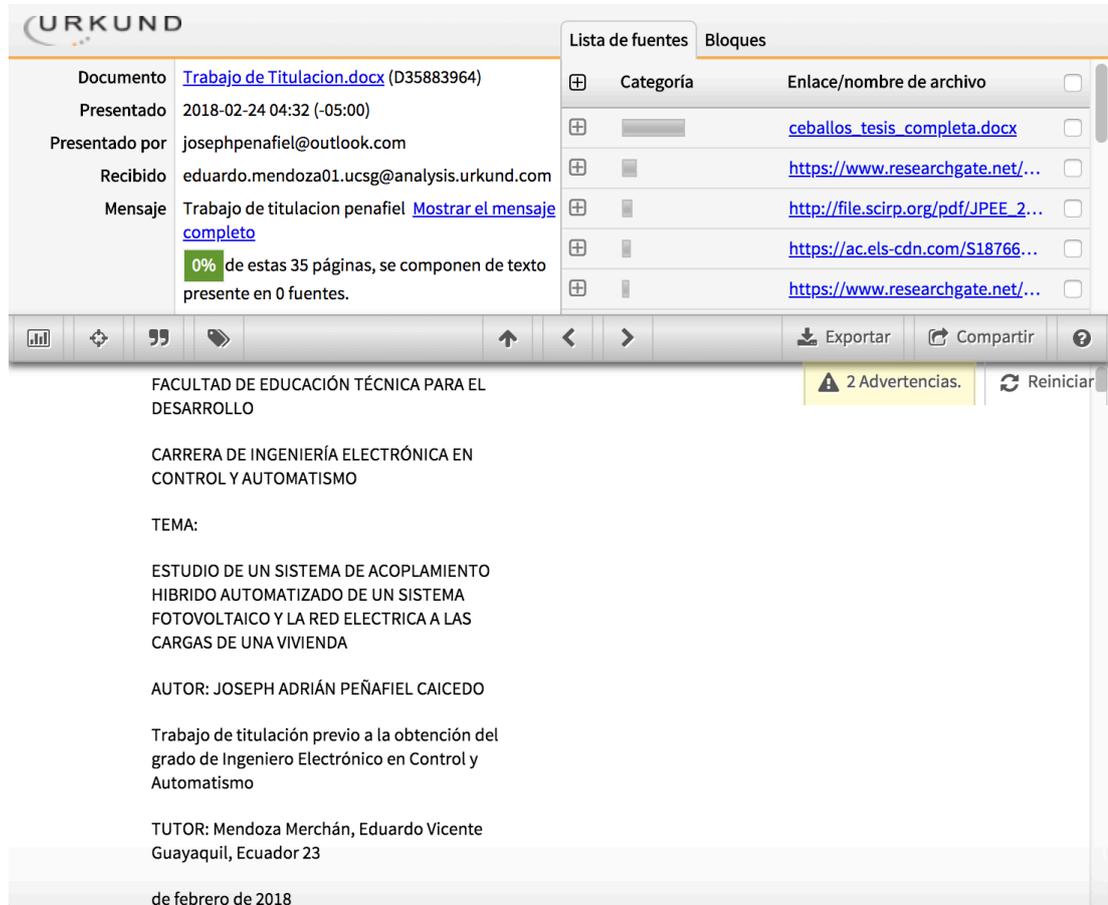
EL AUTOR

PEÑAFIEL CAICEDO, JOSEPH ADRIÁN

REPORTE DE URKUND

TÍTULO: “Estudio de un sistema de acoplamiento híbrido inteligente para energía solar fotovoltaica y la red eléctrica de una vivienda”.

AUTOR: Peñafiel Caicedo, Joseph Adrián



The screenshot displays the URKUND interface. On the left, document details are shown: 'Documento: Trabajo de Titulacion.docx (D35883964)', 'Presentado: 2018-02-24 04:32 (-05:00)', 'Presentado por: josephpenafiel@outlook.com', 'Recibido: eduardo.mendoza01.ucsg@analysis.urkund.com', and 'Mensaje: Trabajo de titulacion penafiel. Mostrando el mensaje completo. 0% de estas 35 páginas, se componen de texto presente en 0 fuentes.' On the right, a 'Lista de fuentes' table lists several sources with checkboxes. Below the table is a toolbar with icons for navigation and actions like 'Exportar' and 'Compartir'. The main content area shows the text of the document, including the faculty name 'FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO', the career 'CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y AUTOMATISMO', the topic 'ESTUDIO DE UN SISTEMA DE ACOPLAMIENTO HÍBRIDO AUTOMATIZADO DE UN SISTEMA FOTOVOLTAICO Y LA RED ELÉCTRICA A LAS CARGAS DE UNA VIVIENDA', the author 'AUTOR: JOSEPH ADRIÁN PEÑAFIEL CAICEDO', the tutor 'TUTOR: Mendoza Merchán, Eduardo Vicente', and the date 'de febrero de 2018'. A yellow warning box indicates '2 Advertencias'.

Lista de fuentes	Bloques
Categoría	Enlace/nombre de archivo
	ceballos_tesis_completa.docx
	https://www.researchgate.net/...
	http://file.scirp.org/pdf/JPEE_2...
	https://ac.els-cdn.com/S18766...
	https://www.researchgate.net/...

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y AUTOMATISMO

TEMA:

ESTUDIO DE UN SISTEMA DE ACOPLAMIENTO HÍBRIDO AUTOMATIZADO DE UN SISTEMA FOTOVOLTAICO Y LA RED ELÉCTRICA A LAS CARGAS DE UNA VIVIENDA

AUTOR: JOSEPH ADRIÁN PEÑAFIEL CAICEDO

Trabajo de titulación previo a la obtención del grado de Ingeniero Electrónico en Control y Automatismo

TUTOR: Mendoza Merchán, Eduardo Vicente
Guayaquil, Ecuador 23

de febrero de 2018

El trabajo de titulación del estudiante: PEÑAFIEL CAICEDO JOSEPH ADRIÁN, observa un porcentaje del 0% de coincidencias con otros documentos encontrados en internet.

Ing. Eduardo Mendoza Merchán, Mgs
DOCENTE-TUTOR

AGRADECIMIENTO

Quisiera empezar agradeciendo a los pilares fundamentales de mi vida que son mi padre Ángel Peñafiel Villareal, y mi madre Janet Caicedo Pérez. Ellos han sido los que me han impulsado e inspirado para alcanzar el nivel académico obtenido con este trabajo de titulación. Además de los logros obtenidos, lo que más agradezco de ellos es que nunca dejaron de confiar en mí, aun cuando decidí retirarme de la carrera de medicina y empezar una carrera nueva.

Agradezco a Pablo Fuentes Espinoza por mostrar su apoyo incondicional durante el transcurso de esta carrera, por compartir sus conocimientos, y permitir ser testigo de lo que se puede obtener con dedicación y constancia. Agradezco también a sus padres, por abrirme las puertas de su hogar.

Agradezco a Giovanni Borja, por mostrarme que la vida universitaria no se resume en tener altas calificaciones sino más bien, se resume en crear una noción de liderazgo y compañerismo.

Agradezco a Nicole Vivar Estrella por ayudarme a salir de mi zona de confort y aprender más allá de los apuntes de clases.

Agradezco a mis amigos más cercanos, por mostrar interés en los proyectos realizados en el transcurso de la carrera. Esto sirvió de inspiración para realizar proyectos que iban más allá de lo que se aprendía en clases.

Finalmente, agradezco a mis compañeros de curso por el trabajo en equipo que se dio durante toda la carrera, y los momentos de gozo que compartimos.

DEDICATORIA

Este trabajo se lo dedico a mi familia; mis padres, mis sobrinos, mis hermanos, y especialmente a mi bisabuela, la Sra. Lucinda Pareja que en paz descansa. Me hubiera gustado que esté presente en este momento, pero sé que nos acompaña en espíritu y estaría contenta de verme haber logrado llegar a este punto.

Joseph Peñafiel Caicedo



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

TRIBUNAL DE SUSTENTACIÓN

ROMERO PAZ, MANUEL DE JESÚS
DECANO

CÓRDOBA RIVADENEIRA, LUIS SILVIO
COORDINADOR DEL ÁREA

SUAREZ MURILLO, EFRAÍN
OPONENTE

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS.....	XIV
Resumen.....	XV
Abstract.....	XVI
INTRODUCCIÓN.....	2
CAPÍTULO I.....	4
1.1. Planteamiento del Problema.....	4
1.2. Justificación.....	4
1.3. Objetivos.....	5
1.3.1. Objetivo General.....	5
1.3.2. Objetivos Específicos.....	5
1.4. Hipótesis.....	6
1.5. Tipo de Investigación y Metodología de Investigación.....	6
CAPÍTULO II.....	8
2.1. Estado del Arte.....	8
2.2. Fuentes de Generación Centralizadas y Distribuidas.....	10
2.3. Energía Solar y Sistemas Fotovoltaicos.....	12
2.3.1. Células Fotovoltaicas.....	12
2.3.2. Parámetros de las Células Solares.....	14
2.4. Tipos de Sistemas Fotovoltaicos.....	16
2.4.1. Sistemas Fotovoltaicos conectados a la Red.....	16
2.4.2. Sistemas Fotovoltaicos Independientes.....	17
2.5. Inversores de Potencia.....	17
2.5.1. Configuración de Inversores.....	20
2.6. Controladores de Carga.....	21
2.6.1. Características de los Controladores de Carga.....	22
2.6.2. Tipos de Controladores de Carga.....	22
2.7. Sensores para Energía Eléctrica.....	24
2.7.1 Sensor de corriente de efecto Hall.....	25
2.8. Sistemas basados en microcontroladores.....	26
2.8.1. Arquitecturas de Microcontroladores.....	26
2.8.2. Familia ARM Cortex.....	27
2.8.3. Microcontroladores en los Sistemas Fotovoltaicos.....	28
CAPÍTULO III.....	29

3.1. Consideraciones del Sistema.....	29
3.1.1. Descripción del Sistema.....	29
3.1.2. Características	30
3.1.3. Funciones del Sistema.....	31
3.2. Consideraciones Iniciales de Diseño	32
3.2.1. Análisis del Entorno y Espacio Físico	32
3.2.2. Análisis de los Componentes del Sistema de Acoplamiento Automatizado.....	32
3.3. Diseño Hardware del Controlador de Acoplamiento.....	35
3.3.1. Circuito de Control	35
3.3.2. Adquisición de Datos: Sensores y Controlador de Carga.....	37
3.4. Diseño Software del sistema de Acoplamiento	42
3.4.1. Firmware de Control	43
3.4.2. Interfaz de Usuario.....	48
3.5. Pruebas Experimentales del Sistema	54
3.5.1. Pruebas del Sistema Fotovoltaico	54
3.5.2. Pruebas del sensor de Corriente.....	55
3.5.3. Pruebas del controlador de Carga	56
3.5.4. Pruebas de Acoplamiento	56
3.5.5. Pruebas de la Interfaz de Usuario	59
3.6. Cálculo de Costos	60
CAPÍTULO IV.....	61
4.1. Resultado del diseño y análisis del prototipo desarrollado.....	61
4.1.1. Sistema Fotovoltaico.....	61
4.1.2. Sistema de Control.....	61
4.1.3. Sistema de Acoplamiento Automatizado.....	62
4.1.4. Interfaz de Usuario Remota	62
4.2. Discusión de Resultados	62
4.2.1. Análisis de las funciones del sistema.....	62
4.3. Cumplimiento de los Objetivos	63
4.3.1. Objetivos Específicos.....	63
4.3.2. Objetivo General.....	63
4.4. Análisis de la metodología.....	63
4.5. Análisis de la hipótesis.....	64
4.6. Análisis de la solución del problema	64
CAPÍTULO V	65
5.1. Conclusiones.....	65

5.2. Recomendaciones	65
5.3. Trabajo Futuro	65
Bibliografía y Referencias	66
ANEXOS	70
ANEXO A: Código Fuente del Controlador.....	70
ANEXO B: Instrucciones de configuración del Raspberry Pi para el sistema de monitoreo	76
ANEXO C: Instrucciones de Instalación de NodeRED en el Raspberry Pi	86
ANEXO D: Hoja de datos del sensor de corriente	87
ANEXO E: Protocolo de Comunicación del controlador de carga EPSolar Tracer 1210m	88

ÍNDICE DE FIGURAS

Figura 1 Esquema de Generación Centralizada	11
Figura 2 Esquema de Generación de energía distribuida.....	12
Figura 3 Unión P-N de una célula fotovoltaica	13
Figura 4 Flujo de electrones debido a los fotones.....	13
Figura 5 Ilustración de célula, panel, y arreglo solar. Generalmente los paneles solares se los puede encontrar en configuraciones de 12 y 24 voltios.	14
Figura 6 Curva Característica de los Paneles Solares.....	15
Figura 7 Esquema de un sistema fotovoltaico Grid-Tie.	16
Figura 8 Esquema de un sistema fotovoltaico independiente.....	17
Figura 9 Ilustración de la función del inversor.	18
Figura 10 Esquema de un inversor.....	18
Figura 11 Voltaje de Salida del Puente-H.....	19
Figura 12 Ilustración de configuración de inversor en cadena.	20
Figura 13 Ilustración de configuración de micro inversores.	21
Figura 14 Ilustración de la diferencia de eficiencia de las configuraciones de inversores. 21	21
Figura 15 Esquema de básico de un controlador de carga MPPT	23
Figura 16 Principio del efecto Hall, en presencia de un campo magnético.	25
Figura 17 Transformador de Corriente.	26
Figura 18 Esquema de la arquitectura Von-Neumann.....	27
Figura 19 Esquema de la arquitectura Harvard.....	27
Figura 20 Diagrama de bloques del sistema	33
Figura 21 Maqueta con los componentes del sistema.....	33
Figura 22 Esquema de conexiones del panel AC.....	34
Figura 23 Esquema de conexiones del panel DC.....	34
Figura 24 Componentes del Controlador de Acoplamiento	35
Figura 25 Esquema de conexiones del microcontrolador	36
Figura 26 PCB del controlador de acoplamiento.....	36
Figura 27 Sensor de Corriente	37
Figura 28 Circuito equivalente del sensor de corriente utilizado	37
Figura 29 Interfaz del sensor con el microcontrolador	39
Figura 30 Onda de Entrada al pin analógico del microcontrolador	39
Figura 31 Esquema simplificado del sensor de estado de la red eléctrica	40
Figura 32 Esquema de conexión del controlador de carga y el microcontrolador.....	40
Figura 33 Conexión física del conector RJ45 al controlador de carga	41
Figura 34 Módulos de Relés utilizados.....	42
figura 35 Esquema de conexión de los modulos de relé al microcontrolador	42
Figura 36 Diagrama de flujo para la recepción de datos en el microcontrolador	43
Figura 37 Diagrama de flujo para la recepción de datos del controlador de carga.....	44
Figura 38 Trama de datos utilizados en la comunicación con el controlador de carga.....	45
Figura 39 Diagrama de flujo del bucle principal	47
Figura 40 Rutina del microcontrolador para el acoplamiento de las cargas hacia el inversor	47
Figura 41 Rutina del microcontrolador para el acoplamiento de las cargas hacia la red	48

Figura 42 Diagrama de acceso al sistema operativo del Raspberry Pi	48
Figura 43 Ventana de configuración de conexión SSH mediante TCP/IP utilizando TeraTerm.....	49
Figura 44 Ventana de la interfaz de línea de comando de Linux.....	49
Figura 45 Ventana principal del programa Node-RED	50
Figura 46 Interfaz de programación del programa Node-RED 51	
Figura 47 Interfaz de usuario del sistema de acoplamiento	51
Figura 48 Nodos de programación para la recepción de datos del microcontrolador	52
Figura 49 Nodos utilizados para el control del sistema de acoplamiento.....	53
Figura 50 Set de prueba de las características del panel solar, controlador de carga e inversor	54
Figura 51 Datos obtenidos de las pruebas del sistema fotovoltaico y sensor de corriente de la red.....	55
Figura 52 Gráfico de calibración. El eje x pertenece a la constante de calibración mientras que el eje y, pertenece a la potencia leída.	56
Figura 53 Respuesta recibida del controlador de carga mostrado en el monitor serial del computador.....	57
Figura 54 Pruebas de conmutación de la carga hacia las fuentes de energía.....	58
Figura 55 Fuente corriente variable utilizado para las pruebas del sistema	58
Figura 56 Pruebas del bucle principal del microcontrolador	59
Figura 57 Interfaz de usuario del sistema de acoplamiento.....	59

ÍNDICE DE TABLAS

Tabla 1	Tabla de estados lógicos de las compuertas de los transistores del puente H.....	19
Tabla 2	Tabla de posición de los relés del sistema de acoplamiento.....	35
Tabla 3	Tabla de Comandos del microcontrolador	44
Tabla 4	Tabla de información recibida por el controlador de carga.....	46
Tabla 5	Tabla de Comandos del Sistema.....	53
Tabla 6	Cálculo de costos del proyecto	60

Resumen

El trabajo realizado en esta investigación se orientó al diseño de un sistema de acoplamiento automatizado de cargas eléctricas a un sistema fotovoltaico y la red eléctrica. Se realizó un prototipo que demuestre el funcionamiento del sistema utilizando microcontroladores para la lectura de los sensores de las variables del sistema y el control del proceso que conlleva a la selección de la fuente a la que las cargas se deben acoplar. Junto a esto se desarrolló un sistema de visualización remota utilizando un ordenador de placa reducida con sistema operativo Linux. Utilizando metodologías de investigación de tipo experimentales y bibliográficas se pudo desarrollar dicho sistema de acoplamiento automatizado. El resultado de la investigación es un sistema que monitorea la potencia de producción solar y la potencia de consumo para poder decidir el acoplamiento de las cargas al sistema fotovoltaico o la red eléctrica, además de una interfaz remota de control y visualización del sistema, mediante una conexión de red LAN. Al finalizar la lectura de esta investigación, se puede llegar a la conclusión que el uso de microcontroladores es de gran ayuda para el desarrollo de cualquier tipo de sistemas que requieran control automatizado.

Palabras Clave: sistemas fotovoltaicos, medidores de energía, microcontroladores, acoplamiento, controlador de carga, red eléctrica, inversores de potencia, acumuladores.

Abstract

The work carried out in this research aims to create a system of automated coupling of electrical loads to a photovoltaic system and the electrical grid. A prototype was made to demonstrate the operation of the system using microcontrollers for getting sensor readings of the system variables and the control of the process that leads to the selection of the source to which the loads must be coupled. Along with this, a remote viewing system was developed using a small board computer with a Linux operating system. Using experimental and bibliographic research methodologies, this automated coupling system could be developed. The result of the investigation is a system that monitors the power of solar production and the power of consumption to be able to decide the coupling of the loads to the photovoltaic system or the electrical grid, as well as a remote control and visualization interface of the system through a LAN network connection. At the end of the reading of this research, it can be concluded that the use of microcontrollers is of great help for the development of any type of systems that require automated control.

Key Words: Microcontrollers, control systems, energy meters, coupling, charge controllers, electric grid, power inverters, accumulators.

INTRODUCCIÓN

Los sistemas fotovoltaicos para uso residencial son cada vez más comunes debido a una gran reducción de los costos los paneles solares utilizados en estos sistemas. Gracias a esto se han desarrollado servicios por parte de las empresas de servicio eléctrico como el de medida de energía neto. Este servicio permite entregar el exceso de generación de potencia solar a la red eléctrica a aquellos usuarios que dispongan de un sistema fotovoltaico en sus hogares. Al hacer esto las empresas de servicio eléctrico crean un crédito en la factura. Es decir, los cargos generados a los usuarios es la diferencia entre el consumo por parte de la red eléctrica y el exceso de generación solar que se haya registrado en el medidor (Castejón & Santamaría, 2010). Al haber fallas en la red eléctrica, lo más lógico sería poder utilizar al sistema fotovoltaico como fuente de alimentación de emergencia, pero esto no es así. Cuando se tiene un sistema fotovoltaico interactivo con la red eléctrica con un medidor de energía neto y la red eléctrica falla, el sistema fotovoltaico se apagará. Esto sucede debido a que hay un acoplamiento directo de un generador de energía (el sistema fotovoltaico) a las líneas de baja tensión de una red eléctrica. Por lo tanto, para prevenir peligros potenciales al personal de mantenimiento de la red eléctrica, estos sistemas están diseñados para que se apaguen cuando la red eléctrica no se encuentre disponible.

El uso de sistemas fotovoltaicos en el país se encuentra limitado a la alimentación de pequeñas cargas en áreas remotas donde es difícil acceder a la red eléctrica. Con el desarrollo de este trabajo de investigación se busca explorar el uso de sistemas fotovoltaicos con acoplamiento a la red para uso doméstico sin la necesidad de requerir permisos o servicios por parte de las empresas de servicio eléctrico. Para alcanzar los objetivos planteados en esta investigación se decidió desarrollar una maqueta del prototipo del sistema, utilizando varias herramientas de desarrollo de tecnologías como microcontroladores, sensores de corriente, y ordenadores de placa reducida, que permitan acoplar las cargas eléctricas de una fuente a otra dependiendo de las condiciones en las que se encuentre. En este documento se describirá cada uno de los procesos que intervienen para la creación de la maqueta del sistema de acoplamiento automatizado, se dará un detalle de las pruebas realizadas, y por último se dará un análisis de los resultados obtenidos al culminar el trabajo de investigación.

Este documento se divide en 5 capítulos, comenzando con los aspectos generales de la investigación; el marco teórico que indica el contexto de la investigación; el análisis del diseño de la maqueta; la presentación de los resultados; y por último las conclusiones y recomendaciones.

CAPÍTULO I

ASPECTOS GENERALES

1.1. Planteamiento del Problema

Una de las principales razones por la que se instalan sistemas fotovoltaicos de uso doméstico es para depender menos de la red, y tener una fuente de generación de energía renovable limpia. Por las razones descritas y los problemas descritos en la introducción de esta investigación, se ha decidido estudiar un sistema fotovoltaico híbrido, que permita acoplar las cargas eléctricas al sistema fotovoltaico o a la red eléctrica sin necesidad de tener un medidor de energía neto, y que permita utilizar el sistema fotovoltaico cuando la red no se encuentre disponible.

De todo lo analizado se formula la siguiente pregunta de investigación:

¿Cuán eficazmente trabajaría un acoplamiento híbrido inteligente en la red eléctrica de una vivienda, trabajando de forma autónoma con energía renovable, en caso de un corte de la red pública?

1.2. Justificación

Con las crecientes preocupaciones sobre el cambio climático y el calentamiento global, y ante el crecimiento del desarrollo de sistemas fotovoltaicos, es pertinente desarrollar tecnologías para aprovechar al máximo la energía solar. Las herramientas disponibles hoy en día, permiten explorar alternativas en los diseños de los sistemas fotovoltaicos. Con el sistema a desarrollarse se busca fomentar el estudio de sistemas fotovoltaicos a nivel doméstico, con el fin de cubrir en totalidad o parcialmente la demanda eléctrica con energía solar, y depender menos de la red eléctrica. La combinación de los sistemas fotovoltaicos, y la automatización, permiten crear un sistema automatizado, que permite a los usuarios aprovechar de una mejor manera la energía solar generada. La problemática que se ha determinado en el planteamiento del problema genera soluciones que aún no están al alcance de la población en general, de la ciudad de Guayaquil y de hecho la información que existe sobre esta temática es muy escasa. Esta situación amerita, que el tema sea desarrollado principalmente por dos razones. La primera para solucionar una problemática existente en el acoplamiento de los sistemas fotovoltaicos y la red pública, debido a la falta del servicio de medida de energía neta y el segundo, que está encaminado a generar literatura científica en este

tema, que favorezca a estudiantes de las carreras eléctricas que cursen asignaturas con este contenido o que simplemente inicien en esta línea de investigación.

1.2. Delimitación

El presente proyecto de investigación se orienta a solucionar problemáticas de tipo tecnológico puntuales, donde es requerida una fundamentación bastante sólida en cuestiones de generación fotovoltaica y redes eléctricas, para proponer soluciones adecuadas tanto en costo, eficiencia y robustez. El estudio se delimitará a un análisis teórico basado en fuentes bibliográficas con rigor académico y científico. La experimentación con ayuda de una maqueta de un sistema fotovoltaico, basado en elementos, componentes y equipos de generación reducida, que no va más allá de 100 vatios de potencia global; y el monitoreo de las variables principales de sistema a través de un ordenador con ayuda de programas desarrollados por el autor.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema de acoplamiento automatizado capaz de acoplar las cargas de una vivienda a un sistema fotovoltaico o a la red eléctrica según su demanda, utilizando un prototipo basado en dispositivos embebidos para el control y supervisión de los procesos.

1.3.2. Objetivos Específicos

- Diseñar un prototipo de maqueta de un sistema fotovoltaico utilizando elementos y componentes de generación reducida.
- Implementar un sistema de acople y desacople automatizado de las cargas hacia sistema fotovoltaico o la red, utilizando microcontroladores como unidad de procesamiento.

- Diseñar un sistema de supervisión y control basado en una plataforma de desarrollo de ordenadores de placa reducida.

1.4. Hipótesis

El sistema desarrollado trabaja en tiempo real, es decir, que el tiempo de acople o desacople del dispositivo, no genera interferencia en el normal funcionamiento de la vivienda. En este caso la hipótesis se plantea de la siguiente forma:

El tiempo de respuesta del sistema de acoplamiento inteligente es de al menos 100 milisegundos.

1.5. Tipo de Investigación y Metodología de Investigación

El estudio de un sistema de acoplamiento automatizado involucra algunos factores como la potencia que se consume, la potencia que se genera, y rapidez de conmutación. Estos parámetros definen que el tipo de investigación a utilizar en este trabajo sea del tipo correlacional con enfoque cualitativo, puesto que se quiere determinar la relación de estos factores o variables dentro del sistema automatizado a través de mediciones numéricas experimentales con la ayuda de un prototipo. Para lograr lo mencionado se emplea métodos bibliográficos y experimentales. El método bibliográfico se lleva a cabo mediante el análisis de investigaciones y trabajos previos referentes al tema, mientras que la parte experimental se desarrollará en el transcurso del desarrollo del prototipo. Con todo esto en mente, a continuación, se describen algunos de los pasos necesarios para el desarrollo del proyecto:

- Explorar los fundamentos y los tipos de instalaciones fotovoltaicas que se aplican actualmente.
- Investigación de los componentes requeridos para el desarrollo de los sistemas fotovoltaicos.
- Análisis de los tipos de acoplamientos que se pueden aplicar en un sistema fotovoltaico.
- Seleccionar los componentes que forman parte del sistema de acoplamiento.
- Determinar las funciones del sistema y los varios escenarios posibles.

- Desarrollar un algoritmo de programación para el microcontrolador basado en diagramas de máquina de estados finitos.
- Calibrar los sensores de corriente para obtener mediciones correctas.
- Recolectar datos en las pruebas de los componentes para el entendimiento de su funcionamiento.
- Seleccionar elementos de protección, como disyuntores y fusibles para las fuentes de energía.
- Diseño de tarjeta electrónica.
- Desarrollo de una maqueta del sistema fotovoltaico.
- Pruebas de funcionamiento del programa del microcontrolador, en cada escenario posible.

CAPÍTULO II MARCO TEÓRICO

2.1. Estado del Arte

Los sistemas fotovoltaicos son cada vez más populares debido a las reducciones del costo de los paneles solares y el incremento de la eficiencia de estos. Esto ha permitido el desarrollo de nuevas tecnologías para aprovechar el máximo la energía solar, así como también la creación de topologías dependiendo del tipo de uso de los sistemas fotovoltaicos, ya sea residencial o industrial. A continuación, se hablará sobre las últimas tendencias en tecnologías de sistemas fotovoltaicos.

Un estudio realizado en la ciudad de New Borg El Arab city en Egipto, determinó la factibilidad de crear una fábrica alimentada por un sistema híbrido de fuentes renovables de energía de distintos tipos. Estas fuentes se basan en energía solar, energía eólica, un banco de baterías, y un generador alimentado por diésel. Para este estudio, los autores hicieron un perfil de consumo de potencia residencial e industrial, un promedio de irradiación solar, y un promedio de velocidad de viento. Utilizando el software “HOMER”, y los datos recolectados pudieron determinar el dimensionado óptimo de potencia para las diferentes fuentes de energía (Diab, Lan, Zhan, & Ali, 2015).

En el trabajo desarrollado por (Mohammad B. & Robert S., 2014), se desarrolló un algoritmo para la optimización de la generación y almacenamiento de energía para una fuente híbrida de energía solar y eólica. El objetivo de este trabajo es poder determinar un sistema que sea óptimo considerando varios parámetros, y así obtener un sistema relativamente económico, ya que se evitaría sobredimensionamientos.

Debido a la baja eficiencia de conversión que sufren los sistemas fotovoltaicos, se han desarrollado varias técnicas para seguimiento de punto máximo de potencia (MPPT) de los paneles solares. En el estudio realizado por (Reisi, Moradi, & Jamasb, 2012), se hace una clasificación y comparación de las técnicas empleadas en el desarrollo de MPPT para sistemas fotovoltaicos. En el artículo se hacen varios modelos matemáticos con la finalidad de demostrar cada una de las técnicas. También desarrollaron un

modelo de un sistema fotovoltaico utilizando el software Simulink para poder visualizar los modelos matemáticos antes mencionados.

Un estudio hecho por alumnos de la Universidad de Roma (Bruni, y otros, 2013), describe la influencia de las estrategias de control sobre la eficiencia de los sistemas generación de energía híbrida (Solar-Baterías). Estos mencionan varios métodos de control para mantener una alimentación estable durante cualquier condición utilizando las diferentes fuentes de generación de energía. Entre esos métodos se encuentran los basados en reglas, lógica difusa, métodos basados en redes neuronales, y modelos de control predictivos. Para demostrar estos sistemas de control utilizaron modelos creados en Matlab y Simulink.

El artículo desarrollado por (Kuoro, Leon, Vinnikov, & Franquelo, s.f.) describe un resumen de los sistemas de conversión de energía de sistemas fotovoltaicos, enfocándose en las diferentes configuraciones del sistema y topologías de conversión que se usan en los sistemas PV conectados a la red eléctrica (Grid-Connected Systems). Entre estas topologías se mencionan algunas como las de cadena, multi cadena, y sistemas centrales.

En un estudio realizado por (Anantwar, Lakshmikantha, & Sundar, 2017) se implementó un regulador de voltaje de un inversor para un sistema fotovoltaico híbrido utilizando un controlador Proporcional-Integral (PI) auto programable mediante lógica difusa. Esta implementación se hizo para tener un mejor manejo de la potencia reactiva de las cargas y poder tener una calidad de voltaje óptimo constante. Esta implementación fue realizada utilizando Matlab y Simulink utilizando modelos matemáticos de los sistemas fotovoltaicos.

En el artículo de (Jouda, Elyes, Rabhi, & Abdelkader, 2016) se describe un controlador de punto de potencia máximo (MPPT) utilizando lógica difusa y optimización de enjambre de partículas (Particle Swarm Optimization), que es un tipo de algoritmo que trabaja con una nube o enjambre de soluciones. El objetivo del estudio es mejorar la respuesta dinámica de los sistemas fotovoltaicos ante cambios bruscos de irradiación y temperatura. Para el desarrollo de este estudio los autores utilizaron el software

Matlab/Simulink para simular el sistema y el MPPT y así como también ilustrar los resultados obtenidos.

En el artículo desarrollado por (Dharmaian, Nanjappa, & Gounden, 2014) se desarrolló una técnica que provee de un punto de operación de potencia constante para sistemas fotovoltaicos acoplados a la red eléctrica. Para la interface entre los paneles solares y la red eléctrica, los autores desarrollaron un convertidor DC-DC de tipo boost y un inversor de línea conmutado (LCI, Line Commutated Inverter). Los autores del artículo desarrollaron experimentos prácticos, y también mediante simulaciones creadas en Matlab y Simulink, obteniendo resultados similares.

En los sistemas de generación de energía con fuentes renovables, especialmente en aquellas instaladas en residencias generalmente se puede generar un exceso de potencia. En el artículo realizado por (Ismail, Moghavvemi, Mahlia, Muttaqi, & Moghavvemi, 2014) se hacen varias sugerencias para la utilización de este exceso de potencia como, inyectar el exceso de potencia a la red eléctrica nacional, producir hidrógeno, calentamiento de agua, desalinización de agua, entre otros.

La topología híbrida de los sistemas fotovoltaicos en los cuales aparte de la energía proveniente de los paneles solares, se utiliza también otras fuentes como la de la red eléctrica nacional, y bancos de baterías. En el artículo creado por (Traore, Taylor, Zohdy, & Peng, 2017) se modela y se simula un sistema híbrido donde se tiene un sistema fotovoltaico, sistemas de energía de almacenamiento como baterías y super capacitores, y la red pública nacional. En este artículo también se hace un contraste entre los sistemas “micro-grid” AC y DC.

2.2. Fuentes de Generación Centralizadas y Distribuidas

La alimentación de energía sin importar su uso final normalmente proviene de centrales de generación de energía. Es decir, una fuente ya sea térmica, hidroeléctrica, o nuclear, que produce grandes cantidades de energía y se encuentran en un solo lugar. Esta energía recorre por varias etapas de transmisión y distribución hasta llegar al consumidor final. Cada una de estas etapas opera en diferentes rangos de voltajes y se dividen en generación, transporte y distribución.

La etapa de generación, como su nombre indica, es donde se produce la energía y su voltaje puede ser alrededor de 130 a 400 Kilovoltios. En la etapa de transporte, el voltaje generado se reduce con la ayuda de transformadores a un rango entre 30 y 66 kilovoltios. En esta etapa la energía viaja desde el generador hasta las subestaciones para continuar con la etapa de distribución. En la distribución, el voltaje se reduce nuevamente a un rango de 3 a 20 kilovoltios para ser dirigidas al consumidor final por medio de las líneas de baja tensión, logrando obtener un voltaje final de 120 o 240 voltios (Monroy, 2002). La figura 1 ilustra las etapas de una fuente de generación centralizada.

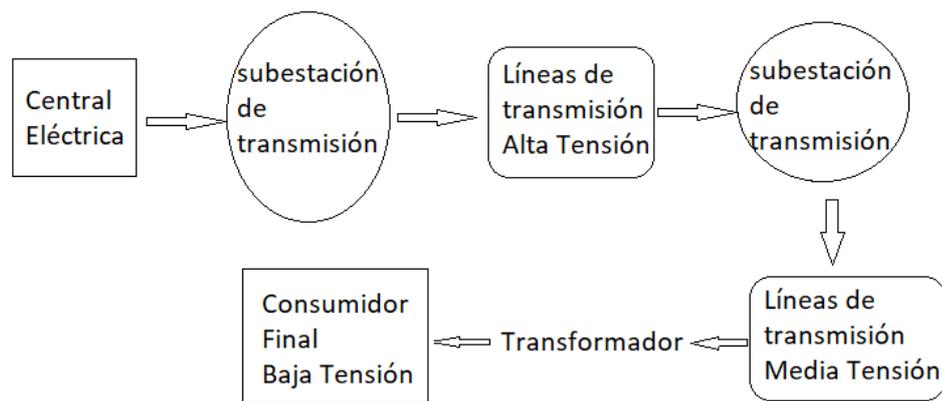


Figura 1 Esquema de Generación Centralizada
Fuente: (Valencia Quintero, 2008)

Las fuentes de generación distribuidas, también llamadas fuentes de generación descentralizadas son fuentes de producción de energía que se pueden dar en diferentes puntos de un país y se conectan a la red eléctrica a través de las líneas de baja tensión. Estas fuentes de energía generalmente son de tipo renovables, como la energía solar o eólica, e incluso generadores de combustión de diésel, estas fuentes pueden provenir de residencias o de micro redes (Environmental Protection Agency, 2017). Las fuentes de generación descentralizadas están sujetas a regulaciones y normas locales, puesto que, para acoplar un generador a una red eléctrica, el voltaje de ellas debe tener la misma frecuencia, fase y amplitud, entre otro tipo de regulaciones y medidas de seguridad (McDonald, Wojszczyk, Flynn, & Voloh, 2013). La figura x muestra un esquema simplificado de las fuentes de generación distribuida.

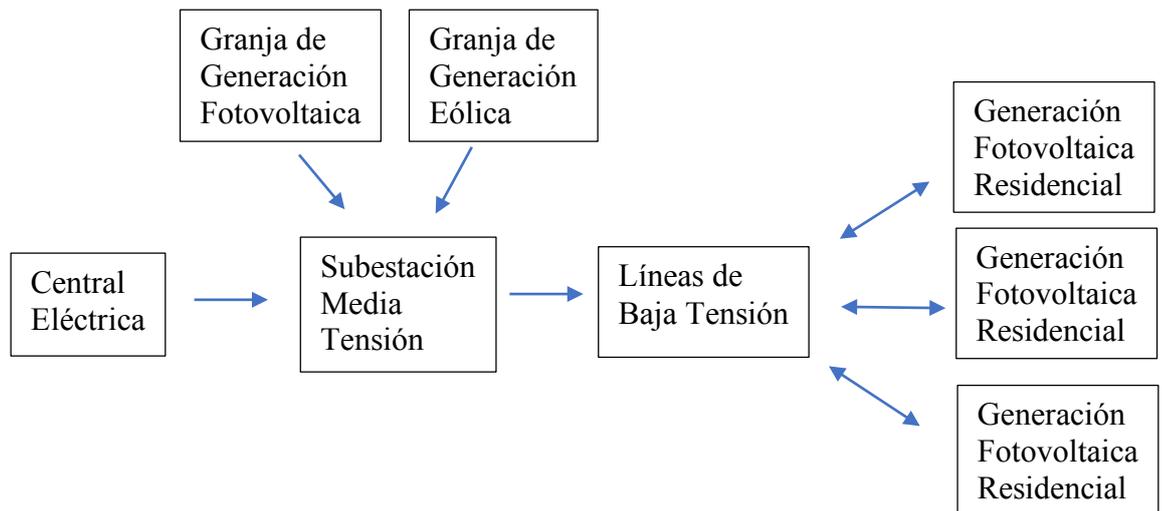


Figura 2 Esquema de Generación de energía distribuida
 Fuente: (McDonald, Wojszczyk, Flynn, & Voloh, 2013)

2.3. Energía Solar y Sistemas Fotovoltaicos

La energía solar es un tipo de energía renovable que se obtiene a través de la radiación electromagnética que emite el sol. Se considera tipo de energía limpia ya que es abundante y virtualmente inagotable, y la utilización de esta no produce cambios en el equilibrio de la tierra (Méndez & Cuervo, 2012). La energía solar se aprovecha transformándola en otros tipos de energía como energía eléctrica y energía térmica por medio de sistemas fotovoltaicos y colectores térmicos, respectivamente.

Los sistemas fotovoltaicos son un conjunto de componentes y dispositivos electrónicos que captan la energía del sol para luego ser transformada en energía eléctrica. Estos sistemas están compuestos por paneles solares, que a su vez están compuestos por células fotovoltaicas. Estos paneles solares generan corriente continua, que debe ser procesada por un inversor, el cual convierte la corriente continua en corriente alterna. En algunos casos se utilizan bancos de baterías para almacenar el exceso de energía generado por los paneles solares y poder ser utilizada cuando la luz solar ya no esté disponible.

2.3.1. Células Fotovoltaicas

En este dispositivo es donde ocurre la transformación de la radiación solar en corriente eléctrica. Está compuesto por una fina placa de material semiconductor, en la mayoría de los casos silicio, el cual es dopado con otros elementos para que se pueda generar el efecto fotovoltaico.

Como en los transistores, las células fotovoltaicas están compuestas por regiones P y N. Cuando los fotones inciden sobre la región P-N de las células fotovoltaicas, hace que los electrones libres de la región N se desplacen hacia la región P de la célula, la cual se encuentra con huecos libres generando un par electrón-hueco, produciendo una corriente eléctrica y, por lo tanto, un voltaje en sus terminales a causa del campo eléctrico creado (Castejón & Santamaría, 2010).

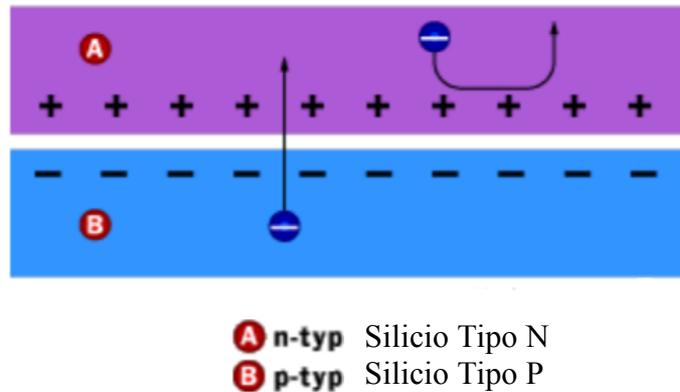


Figura 3 Unión P-N de una célula fotovoltaica
 fuente: (Castejón & Santamaría, 2010)

La energía que es posible utilizar depende de las características del material de fabricación de la célula, por ejemplo, las células fabricadas con silicio monocristalino tienen alrededor de 13% y 17% de eficiencia, por otro lado, las células fabricadas con componentes especiales pueden alcanzar hasta un 32.5% de eficiencia (Enerpoint, 2017).

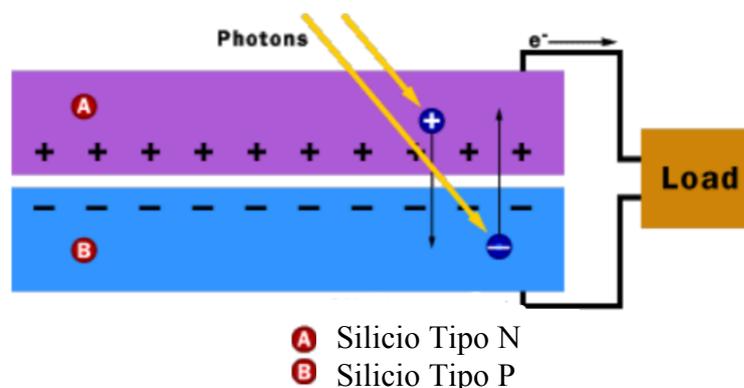


Figura 4 Flujo de electrones debido a los fotones.
 Fuente: (Castejón & Santamaría, 2010)

Las células solares, se conectan en serie, paralelo, o una combinación serie-paralelo, para formar los famosos paneles solares. Estos paneles luego se los junta de igual manera, en serie, paralelo, o ambos para formar un arreglo de paneles solares y así lograr obtener voltaje y corriente óptimos que se ajuste a las especificaciones de los inversores. La figura 5 ilustra los arreglos de células para formar los paneles solares, también los arreglos de paneles.

2.3.2. Parámetros de las Células Solares

Las células solares entregan potencia mientras se encuentran iluminadas y tienen una carga conectada a ellas. La potencia que las células solares entregan está sometida a variables como la temperatura, la irradiancia que es la magnitud que describe la intensidad de iluminación solar y sus medidas se dan en W/m^2 (vatios/metro cuadrado), y la carga que se alimenta de la célula (Castejón & Santamaría, 2010).

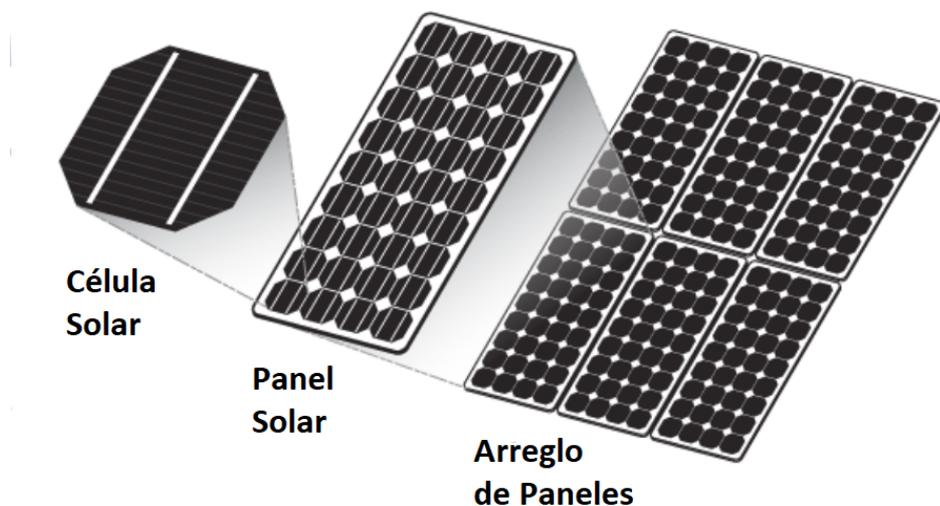


Figura 5 Ilustración de célula, panel, y arreglo solar.
Fuente: (Castejón & Santamaría, 2010)

Si conectamos a la célula solar una carga variable, se podrá obtener una curva de Voltaje/Corriente y Voltaje/Potencia, y a partir de estas curvas se obtienen los siguientes parámetros:

- Corriente de Corto Circuito (I_{sc}): Es la corriente que entrega la célula solar cuando se encuentra a una carga con 0 ohmios de resistencia. En este punto de la curva el voltaje de los paneles es de 0 voltios.

- Voltaje de Circuito abierto (V_{oc}): Esta medida es el máximo potencial al que se encuentra la célula solar. En este punto de la curva la corriente es de 0 amperios.
- Punto Máximo de Potencia (MPP): Este es el punto donde la célula solar entrega su máxima potencia. Este valor está definido por los valores de V_{mp} e I_{mp} (Voltaje y Corriente de máxima potencia, respectivamente).
- Factor de forma (Fill Factor): Esta es la relación entre la potencia máxima de la célula y la multiplicación entre V_{oc} e I_{sc} . Mientras más alto es su valor, mejor es la célula.

Cabe recalcar que estos parámetros normalmente se dan en condiciones de medida estándar, las cuales están definidas en 1000 W/m^2 de irradiancia y 25 grados centígrados (Sunfields Europe, 2016). La figura 6 muestra la curva característica de las células solares.

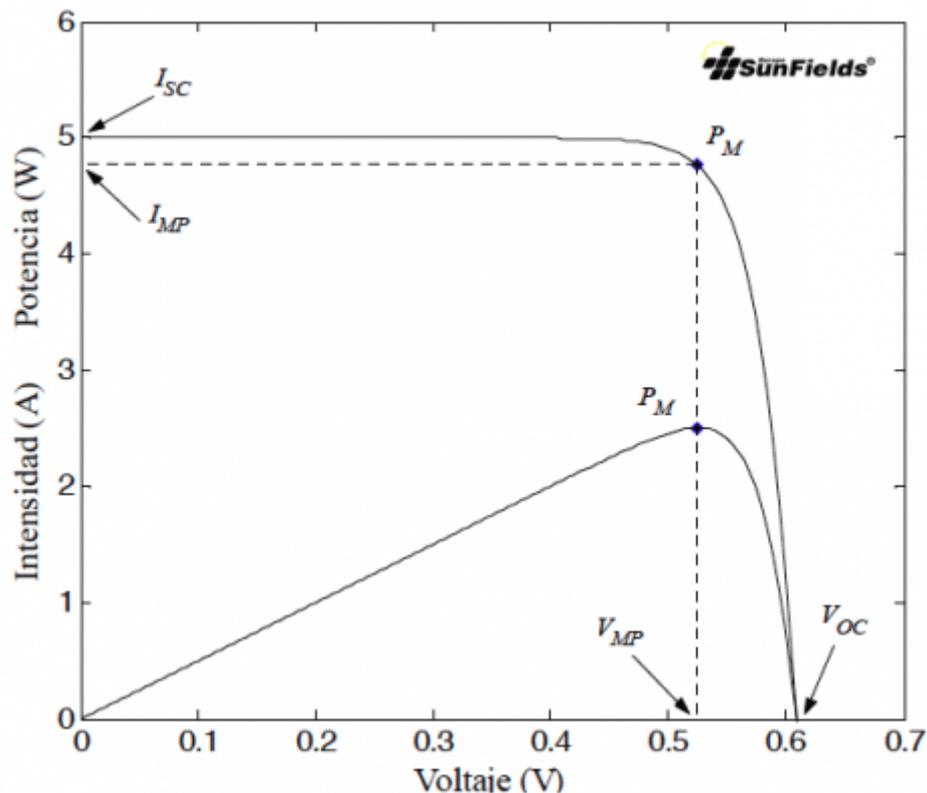


Figura 6 Curva Característica de los Paneles Solares
fuente: (Castejón & Santamaría, 2010)

2.4. Tipos de Sistemas Fotovoltaicos

Los sistemas fotovoltaicos se pueden dividir en dos grupos, siendo estos los Sistemas Fotovoltaicos conectados a la red, y los Sistemas Fotovoltaicos Independientes.

2.4.1. Sistemas Fotovoltaicos conectados a la Red

En los sistemas conectados a la red pública, la corriente generada por los paneles solares se acopla directamente a la red eléctrica nacional a través de un inversor de tipo “grid-tie”. Este inversor pone prioridad a la potencia que los paneles solares entregan, y el exceso de potencia generado se lo vende a la red pública por medio de un medidor inteligente instalado por la compañía nacional de electricidad. En el caso que la carga requiera más potencia de la que lo paneles puedan entregar, la potencia se comparte con la de la red. En las noches, cuando no hay luz solar, la potencia que consume la carga viene directamente de la red.

Cuando existen cortes de energía, todo el sistema deja de funcionar, ya que el inversor necesita de la onda de referencia de la red pública para su operación, y cuentan con un sistema de seguridad llamado anti-isla, que desconecta al sistema fotovoltaico de la red, para prevenir que la red siga teniendo voltaje cuando exista un corte y evitar accidentes al momento de su mantenimiento.

Estos sistemas son los más populares debido a su sencillez, y a su reducido costo en comparación con los otros tipos de sistemas (SineTech, 2017). El usuario puede ajustar la potencia del sistema de acuerdo con su presupuesto, logrando obtener reducciones de gastos en la factura de consumo eléctrico. En la figura 10 se muestra un esquema básico de un sistema fotovoltaico conectado a la red.

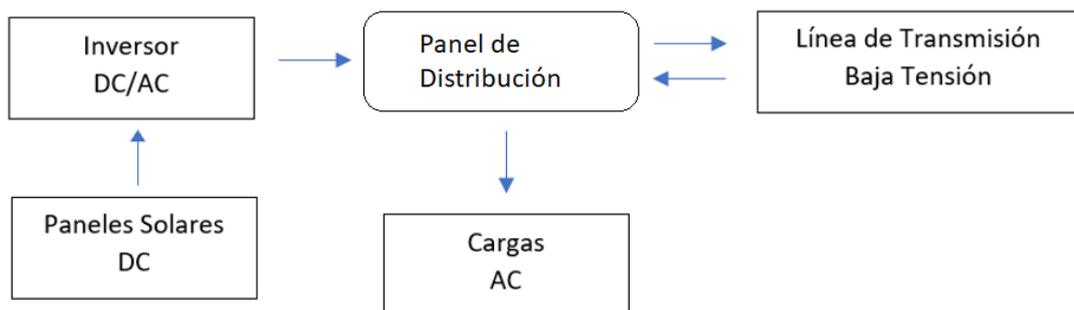


Figura 7 Esquema de un sistema fotovoltaico Grid-Tie.
Fuente: (Méndez & Cuervo, 2012)

2.4.2. Sistemas Fotovoltaicos Independientes

Estos sistemas también conocidos como “Off-Grid”, funcionan independientemente de la red eléctrica pública. En estos sistemas el inversor tiene un generador de onda AC, y adicionalmente se añade un banco de baterías de ciclo profundo que se encargaran de entregar energía cuando los paneles no la estén generando. Estos sistemas son más costosos debido a que se debe dimensionar el sistema de acuerdo con la máxima potencia que requiera la carga y adicional a esto, se suma el valor del banco de baterías, o algún otro tipo de generación de energía como, por ejemplo, un generador a diésel, o un sistema eólico (SineTech, 2017).

En el día cuando la potencia generada por los paneles solares es mayor que la potencia consumida por la carga, el exceso de esta es direccionada hacia las baterías a través de un controlador de carga. Cuando la potencia consumida es mayor a la que los paneles solares pueden entregar, el controlador de carga hace que las baterías dejen de consumir la potencia de los paneles solares para que estas entreguen la energía almacenada y entren en modo de descarga.

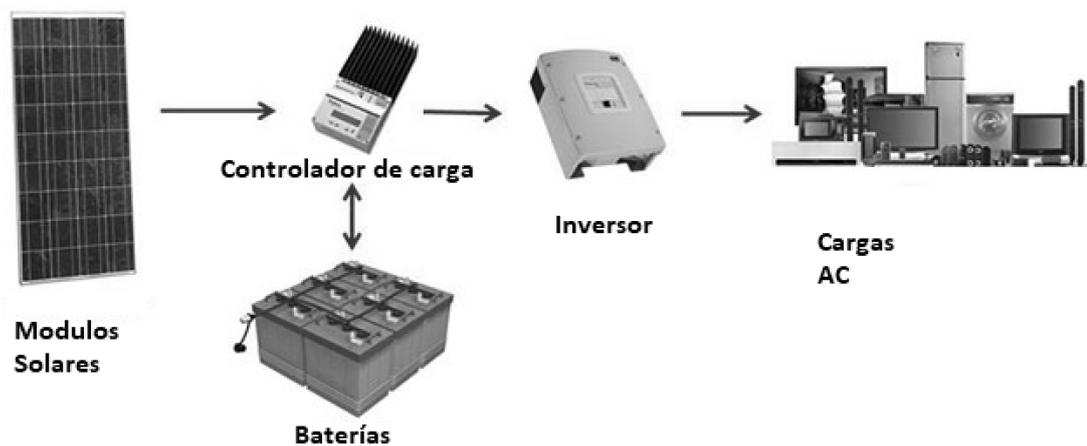


Figura 8 Esquema de un sistema fotovoltaico independiente.
Fuente: (Ditrolic Solar, 2017)

2.5. Inversores de Potencia

Los inversores de potencia son dispositivos que convierten la energía eléctrica de corriente continua (DC) en energía de corriente alterna (AC), como se ilustra en la figura 9.

En los sistemas fotovoltaicos, la energía proveniente de los arreglos de paneles solares es de corriente continua, por lo tanto, para poder adaptar la energía proveniente de estos, se utilizan estos dispositivos para obtener un voltaje de corriente alterna, en los casos residenciales, 120 voltios AC o 240 voltios AC

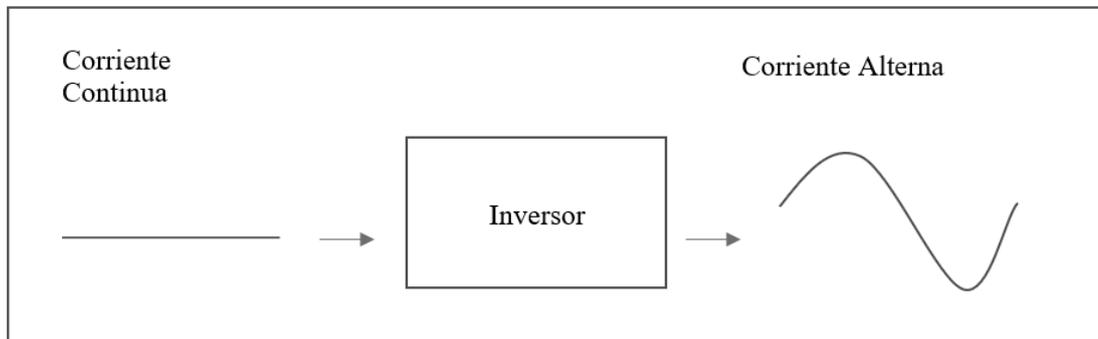


Figura 9 Ilustración de la función del inversor.
Fuente: (Zipp, 2016)

En los inversores solares, el voltaje entregado por los paneles solares es elevado por un convertidor Boost DC-DC. La transformación de corriente continua a corriente alterna se lo hace por medio de un circuito de transistores llamado Puente H. En este circuito, dependiendo del orden de conmutación de los transistores, se obtiene un voltaje positivo y un voltaje negativo en forma de onda cuadrada. Para obtener la forma sinusoidal de la corriente alterna, los transistores del puente H, se activan mediante una señal PWM, que es controlada por un circuito que produce una señal de referencia sinusoidal. El circuito de un inversor se muestra en la figura 10, y la Tabla 1 muestra los estados de conmutación para generar el voltaje positivo y negativo.

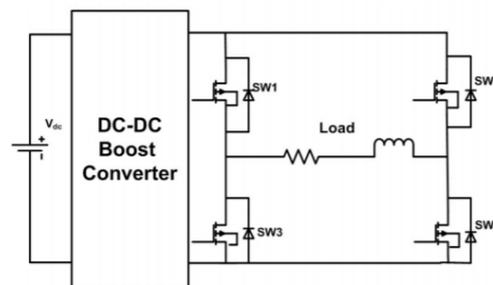


Figura 10 Esquema de un inversor.
Fuente: (Abdel & Orabi, 2011)

SW1	SW2	SW3	SW4	VLoad
1	0	0	1	Voltaje +
0	1	1	0	Voltaje -
1	1	0	0	0 V
0	0	1	1	0 V

Tabla 1 Tabla de estados lógicos de las compuertas de los transistores del puente H.
Fuente: (Zipp, 2016)

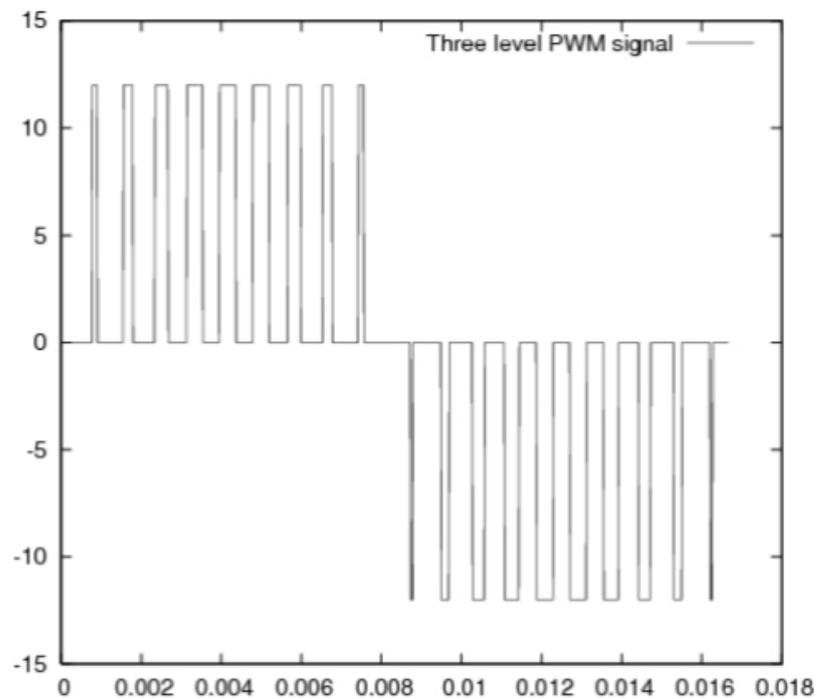


Figura 11 Voltaje de Salida del Puente-H.
Fuente: (Doucet, Eggleston, & Shaw, 2006)

En la figura 11, se puede observar el voltaje de salida del puente H. El ciclo de trabajo de este voltaje va a depender de la señal PWM de las compuertas del puente H, que a su vez este depende del voltaje de referencia de la onda a generar. Estas ondas cuadradas pasan a través de un filtro que suaviza la onda haciéndola una onda sinusoidal pura.

2.5.1. Configuración de Inversores

Existen dos tipos de configuración para los inversores en los sistemas fotovoltaicos para uso residencial, siendo estos, la configuración en cadena o String, y la configuración de micro inversores. La configuración que se elija dependerá del presupuesto del proyecto, o de la cantidad de irradiación que tendrán los paneles solares.

2.5.1.1. Configuración en Cadena (String)

En la configuración de inversor en cadena como se muestra en la figura 12, los arreglos de paneles solares se conectan a un solo inversor. La salida de voltaje continuo total de los arreglos de paneles es procesada por el inversor para transformarla en voltaje alterno. Esta configuración es popular debido a que ofrece mejor costo por kilowatt del sistema fotovoltaico, comparado con la configuración de micro inversores. Una desventaja de esta configuración es la reducción de rendimiento de todo el sistema a causa de algún arreglo que sea afectado por sombra (Zipp, 2016).

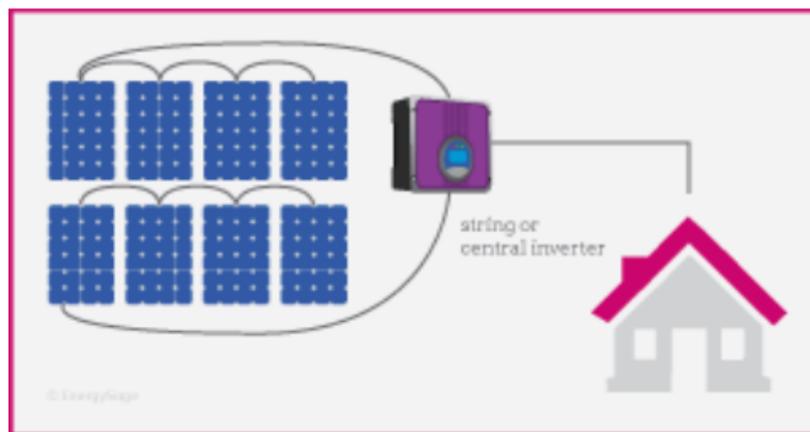


Figura 12 Ilustración de configuración de inversor en cadena.
Fuente: (Energysage, 2017)

2.5.1.2. Configuración Micro Inversor

Los micro inversores, son inversores que se conectan individualmente a cada panel solar. Esto ofrece modularidad, ya que se puede añadir los paneles que sean necesarios dependiendo de la potencia que se quiera obtener de ellos, y también ofrece la mayor eficiencia al momento de entregar la potencia, ya que cada panel entrega su potencia individualmente y no se ve afectado por algún panel que se encuentre tapado por la

sombra de una nube (Zipp, 2016). Estos micro inversores se utilizan en aplicaciones residenciales. Una de sus desventajas es su elevado costo por kilowatt, comparado con otras configuraciones de inversores. La figura 13 y 14 muestra este tipo de configuración.



Figura 13 Ilustración de configuración de micro inversores.
Fuente: (Energysage, 2017)

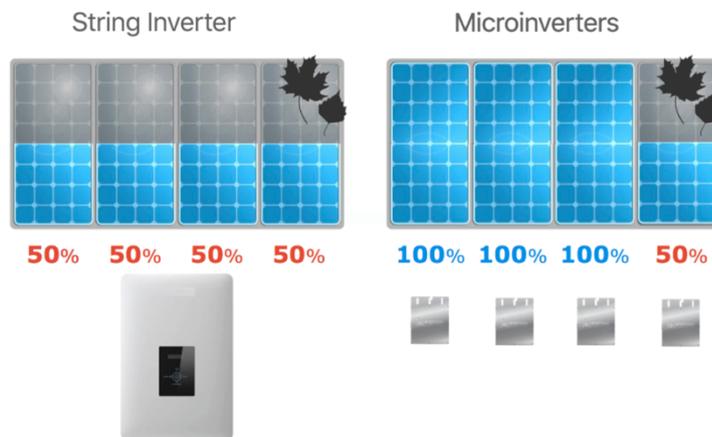


Figura 14 Ilustración de la diferencia de eficiencia de inversores.
Fuente: (Tindo Solar, 2017)

2.6. Controladores de Carga

Los controladores de carga son los dispositivos que administran la potencia que se dirige al banco de baterías desde los paneles solares. También, se encargan de que las baterías no se sobrecarguen durante las horas de máxima producción de energía de los paneles solares, y se aseguran de que la corriente no fluya desde las baterías hacia los paneles solares. Adicionalmente, los fabricantes de los controladores de carga añaden funciones adicionales como, ajuste de carga según el tipo de batería, protección contra

inversión de polaridad, protección contra corto circuitos, medición de temperatura de las baterías, protección de sobre voltaje y corriente, y por último protección contra descarga excesiva. Todas estas funciones permiten un buen mantenimiento de las baterías (Castejón & Santamaría, 2010).

2.6.1. Características de los Controladores de Carga

Para el correcto dimensionamiento de un controlador de carga es importante saber dos parámetros del sistema fotovoltaico, Voltaje nominal y corriente nominal. El voltaje nominal del sistema fotovoltaico debe ser el mismo que el del banco de baterías. Estos valores generalmente son 12 voltios, 24 voltios y 48 voltios. La intensidad nominal del controlador de carga debe ser seleccionado en función a la corriente de corto circuito del arreglo de paneles solares del sistema (Castejón & Santamaría, 2010).

2.6.2. Tipos de Controladores de Carga

Los controladores de carga determinan el algoritmo de carga de las baterías midiendo el voltaje de estas. Una vez hecho esto, el controlador de carga ajusta el voltaje a uno adecuado para la carga de las baterías. Entre los métodos que se utilizan para realizar esta función están los de tipo PWM y MPPT (Castejón & Santamaría, 2010).

2.6.2.1. Controladores de Carga PWM

Los controladores de carga PWM, utilizan la técnica de modulación de ancho de pulso para traducir el voltaje producido por los paneles, a un voltaje que se ajuste al de las baterías. La desventaja de estos controladores es que se aplica la técnica de PWM directamente al voltaje que entregan los paneles solares, resultando en una entrega ineficiente de potencia o pérdidas.

2.6.2.2. Controladores de Carga MPPT

Los controladores de carga de tipo MPPT (Maximum Power Point Tracking), están diseñados para obtener la máxima potencia de los paneles solares. La máxima potencia que estos entregan depende de varios parámetros, como irradiación solar, temperatura del panel y la carga. El algoritmo más utilizado para encontrar este punto es llamado Método de Perturbación y Observación (Microchip, Sergiu, & Rosu-Hamzescu, 2013).

En este método el voltaje de los paneles solares es perturbado en cierta dirección; si la potencia extraída de los paneles solares incrementa, se indica que aún no se ha alcanzado el punto máximo y se sigue incrementando hacia la misma dirección el voltaje de perturbación. Por otro lado, si la potencia extraída de los paneles solares disminuye, se indica que el punto máximo de operación se encuentra por debajo del voltaje de perturbación (Harish & Prasad, 2013). Los controladores de carga de este tipo monitorean estos parámetros y los regula de tal manera que, sin importar las condiciones de operación o la carga, estos siempre entregarán la máxima potencia posible hacia las baterías. Esta regulación se da por medio de un convertidor DC-DC llamado Buck-Boost (Microchip, Sergiu, & Rosu-Hamzescu, 2013), mostrado en la figura 15.

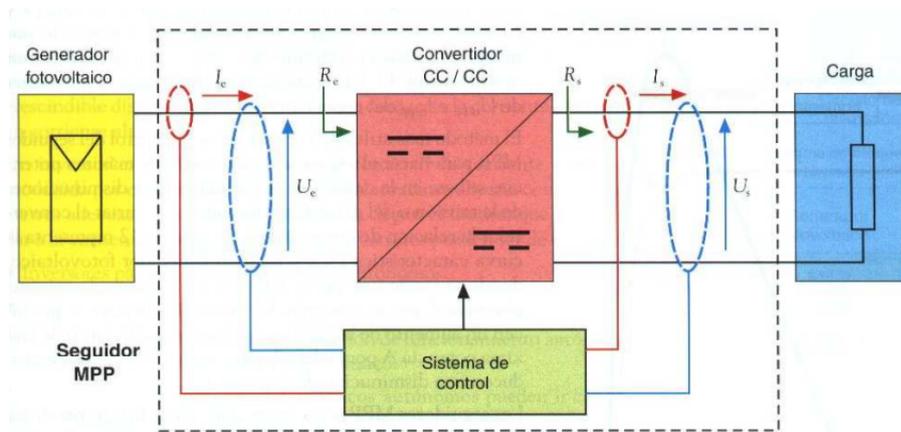


Figura 15 Esquema de básico de un controlador de carga MPPT
fuente: (Castejón & Santamaría, 2010)

Para realizar el algoritmo se miden los voltajes y corriente de entrada y salida como muestra la figura. En un convertidor DC/DC se cumple lo siguiente:

$$(1) d = \frac{U_e}{U_s} = \frac{I_s}{I_e}$$

En la ecuación (1), d es la relación de conversión que el sistema de control del MPPT puede variar durante su funcionamiento. Esta variable dicta el ciclo de trabajo de la unidad PWM a la cual operará el convertidor Buck/Boost.

El parámetro R_e es la resistencia que ve el generador y al mismo tiempo la resistencia de entrada del convertidor DC-DC. Esta resistencia es la que fija el punto de potencia de la curva característica de los paneles. Esta resistencia viene dada por:

$$(2) R_e = \frac{U_e}{I_e}$$

La resistencia R_s es la resistencia de la carga, la cual viene dada por:

$$(3) R_s = \frac{U_s}{I_s}$$

De las expresiones (1), (2), y (3) se obtiene:

$$(5) R_e = d^2 * R_s$$

La expresión (5), demuestra que se puede variar la variable d para que la resistencia R_e coincida con la carga y por lo tanto haya la máxima transferencia de potencia (Castejón & Santamaría, 2010).

2.7. Sensores para Energía Eléctrica

En los sistemas de generación distribuidos, es importante saber la cantidad de energía que los generadores producen, esto se hace mediante sensores de corriente y de voltaje. Con estos se podrá obtener una medida de la potencia que se genera.

Los sensores de corriente son dispositivos que miden la corriente la corriente eléctrica en un conductor y generan una señal proporcional a la magnitud medida. Esta señal puede ser dada en voltaje o en corriente. Con entrada de corriente alterna se puede obtener tres tipos de salidas de los sensores de corriente. La primera es una salida analógica que replica la onda sinusoidal de la corriente en la señal de medición, la segunda es de salida bipolar que entrega la señal de igual manera que la anterior, y la última de salida unipolar, que entrega una medida promedio RMS de la corriente medida (Honeywell, 2017).

Con entrada de corriente continua, la señal de salida que entregan los sensores puede ser unipolar o de salida digital. La primera duplica la forma de la corriente medida, y la segunda da un valor digital cuando la corriente excede cierto valor predeterminado. Entre los varios métodos de medida de corriente, el más popular es el sensor de corriente de efecto Hall.

2.7.1 Sensor de corriente de efecto Hall

El efecto Hall ocurre cuando circula corriente a través de una lámina conductora y esta se coloca en un campo magnético. Las cargas que circulan experimentan una fuerza magnética, y son desplazadas hacia uno de los bordes del conductor. Por consecuencia de esto, se genera un exceso de carga negativa en uno de los bordes del conductor, y un exceso de carga positiva en el otro extremo, provocando la aparición de un campo eléctrico. A partir de esto se obtiene un voltaje de salida proporcional a la fuerza del campo magnético (Honeywell, 2017) como muestra la figura 16.

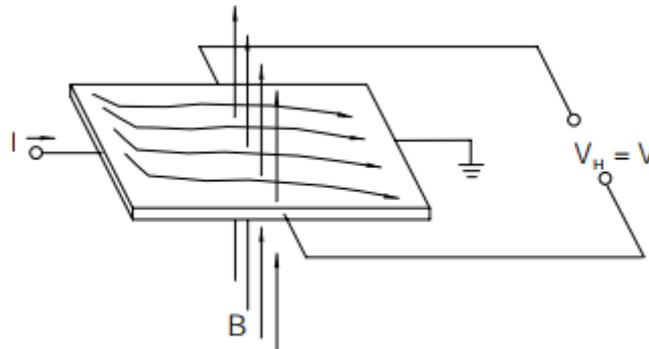


Figura 16 Principio del efecto Hall, en presencia de un campo magnético.

Fuente: (Honeywell, 2017)

El voltaje entregado en estos terminales es procesado y amplificado para obtener una señal que pueda ser leída por un convertidor analógico-digital, o un microcontrolador.

2.7.2. Transformadores de Corriente

Los transformadores de corriente son dispositivos que se utilizan como sensor de corriente alterna. Como su nombre indica, estos dispositivos constan de una bobina primaria y una bobina secundaria. La línea de alimentación de la fuente que quiera ser medida actúa como el primario del dispositivo (ver figura 17), mientras que el secundario lleva la señal proporcional al instrumento de medida que se vaya a utilizar, como por ejemplo un ADC. La señal proporcional se puede calcular utilizando una resistencia de carga (burden resistor) utilizando la siguiente fórmula (Open Energy Monitor, 2015).

$$R_b = \frac{V_{ref} \times Vueltas_{secundario}}{2\sqrt{2} \times CorrienteMax_{primario}} \quad (1)$$



Figura 17 Transformador de Corriente.
Fuente: (Open Energy Monitor, 2015)

2.8. Sistemas basados en microcontroladores

Los microcontroladores son dispositivos de procesamiento que incluyen todos los componentes encontrados en un computador. Entre estos componentes se encuentran, el procesador, memoria de programa y memoria de datos (ROM y RAM), las interfaces de entrada y salida, las cuales pueden ser digitales o analógicas y por último también cuentan con protocolos de comunicación como UART, I2C, SPI, entre otros. Comúnmente son llamados sistemas en chip (SoC). Estos son utilizados para controlar procesos, como dispositivos de adquisición de datos, procesamiento de señales entre otros. Debido a que se utilizan para cumplir una función específica, los sistemas que se crean basados en microcontroladores se los denomina sistemas embebidos (Ali Mazidi, Naimi, & Naimi, 2011).

Los microcontroladores se clasifican en arquitecturas de 8, 16, y hasta 32bits, siendo la última las más eficientes y con mayor desempeño.

2.8.1. Arquitecturas de Microcontroladores

Los microcontroladores utilizan el programa almacenado en su memoria para realizar acciones. Estas acciones las realiza la unidad de procesamiento del microcontrolador. La manera que el CPU accede a la memoria del sistema se puede realizar de una de dos maneras que definirán su arquitectura (Ali Mazidi, Naimi, & Naimi, 2011).

2.8.1.1. Arquitectura Von-Neumann

Considerada como la arquitectura tradicional de los microcontroladores. En esta arquitectura la unidad central de proceso se conecta a la memoria que contiene las

instrucciones del programa de datos mediante un solo bus. La velocidad de transmisión dependerá del ancho de banda de este bus la cual hace que presente limitaciones a la hora de acceder a datos más grandes que el bus. La figura 18 muestra un esquema de esta arquitectura.

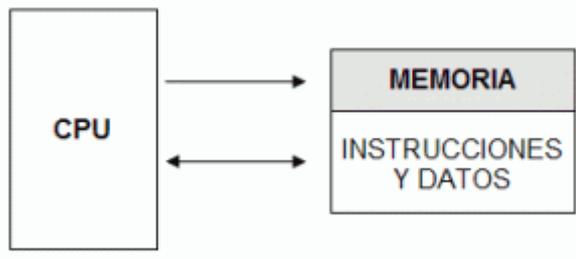


Figura 18 Esquema de la arquitectura Von-Neumann.

Fuente: (Ali Mazidi, Naimi, & Naimi, 2011)

2.8.1.2. Arquitectura Harvard

En esta arquitectura la memoria de programa y la memoria de datos se encuentra separada, a diferencia de la arquitectura Von-Neumann. En la memoria de programa se encuentran las instrucciones de programa, y en la memoria de datos se almacenan los datos del programa. La figura 19 ilustra esta arquitectura.

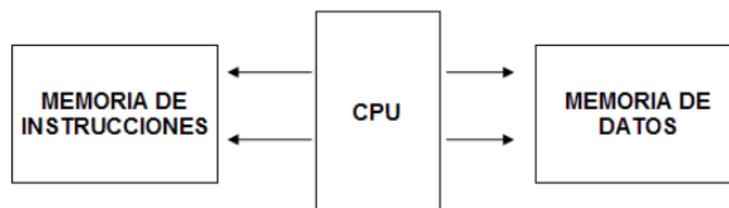


Figura 19 Esquema de la arquitectura Harvard.

Fuente: (Ali Mazidi, Naimi, & Naimi, 2011)

2.8.2. Familia ARM Cortex

Fueron introducidos en el año 2004 por la compañía británica ARM Holdings. Se dividen en tres familias; Cortex-A, los cuales son procesadores capaces de ser manejados por un sistema operativo ligero como algunas distribuciones de Linux; los Cortex-R, que son microcontroladores para uso en aplicaciones de tiempo real, como en sistemas ABS de automóviles; y finalmente la línea de microcontroladores orientados al desarrollo de sistemas embebidos, los Cortex-M (Yiu, 2014).

La familia Cortex-M, se divide en 4 subfamilias que varían en rendimiento y aplicación. Estos son los Cortex M0, M0+, M3, M4, M7. La subfamilia M0/M0+, fueron desarrollados para competir directamente con los microcontroladores de 8 y 16 bits, ofreciendo menor costo, menor consumo de energía, y mayor rendimiento. La subfamilia M3 fue desarrollado para ser un microcontrolador de uso general, con buen rendimiento, precio y consumo de energía. Las subfamilias M4 y M7, son las de mayor rendimiento. Estos incluyen DSP o procesadores de señales digitales y controladores de audio y video en los Cortex M7.

La velocidad de reloj de estos microcontroladores varía desde los 30MHz hasta 200MHz en los Cortex M7 (STMicroelectronics, 2017).

2.8.3. Microcontroladores en los Sistemas Fotovoltaicos

Los microcontroladores pueden encontrarse hoy en día desde los dispositivos más sencillos, como un reloj digital, hasta en los más complejos como en los componentes de los sistemas fotovoltaicos. Los dispositivos que utilizan microcontroladores en los sistemas fotovoltaicos son, los inversores y los controladores de carga.

Los inversores necesitan un sistema de generación de onda la cual los microcontroladores pueden proveer fácilmente con la ayuda del convertidor digital-analógico. Para la generación de la onda sinusoidal de salida, se utiliza los módulos PWM que la mayoría de los microcontroladores tienen embebidos, para la conmutación del puente H del inversor.

En los controladores de carga, los microcontroladores se utilizan para la medición de los parámetros de las baterías, mediante las entradas analógicas de estos. Los controladores de carga también hacen uso de los módulos PWM de los microcontroladores para la regulación del voltaje de los paneles solares.

Los microcontroladores también son muy útiles para crear interfaces de comunicación entre los componentes del sistema fotovoltaico.

CAPÍTULO III

CARACTERIZACIÓN DEL SISTEMA FOTOVOLTAICO CON ACOPLAMIENTO AUTOMATIZADO A LA RED ELÉCTRICA

En este capítulo se hará una descripción general del sistema y su funcionamiento. Luego se analizará en detalle la función de cada uno de los componentes del sistema, el software del microcontrolador y el programa de visualización, para finalmente integrar en un sistema que cumplan con los requerimientos para ser una solución al problema de investigación planteado.

3.1. Consideraciones del Sistema

En este apartado se desarrollará un análisis general del sistema, que permita describirlo de forma general, para luego ser abordado en detalle para su análisis y diseño.

3.1.1. Descripción del Sistema

En esta sección se describirá de forma general el funcionamiento del sistema, lo que permitirá determinar de forma rápida pero completa una solución al problema planteado; más adelante en las funciones del sistema se detallará con más precisión lo que realizará exactamente el dispositivo de control.

Los componentes fundamentales del sistema son el panel solar, el controlador de carga, la batería, el inversor, el controlador de acoplamiento, y por último la red eléctrica. Los paneles solares serán los que conviertan la radiación solar en energía eléctrica. Estos se conectarán al controlador de carga, que asegura que el panel solar opere en su punto máximo de potencia en las condiciones en que se encuentre. El controlador de carga también se encarga de regular la corriente de carga de la batería, y desacoplarlas del panel fotovoltaico cuando se encuentre completamente cargada. El controlador de acoplamiento será el encargado de acoplar las cargas a la red o al inversor y opera en dos modos; automático y en conexión a la red. En modo automático, se comparan las medidas de producción (potencia solar) y consumo de potencia (potencia de la carga).

Una vez hecha esta comparación el controlador toma la decisión de acoplar las cargas a la red o al inversor. Estas decisiones se basan en las siguientes condiciones:

- El consumo de potencia es mayor al de la producción de potencia solar.
- La potencia producida por los paneles es mayor a la potencia de consumo.

En la primera condición, se verifica que la carga de la batería se encuentre por encima del 50%. De ser así, el controlador acoplará las cargas al inversor del sistema fotovoltaico, caso contrario, las cargas se acoplarán a la red eléctrica hasta que la batería se cargue hasta el 90% de su capacidad utilizando energía solar. En caso de que la red falle, el sistema acoplará las cargas al inversor.

3.1.2. Características

En este ítem se ha elaborado una lista de las principales características de los componentes con las que será analizado y diseñado el sistema de acoplamiento automatizado.

- El panel solar entrega 80 vatios de potencia pico. Sus celdas son de tecnología monocristalina. Tiene un voltaje de circuito abierto de 21.5 voltios y una corriente de corto circuito de 6.25 amperios. Los valores voltaje y corriente de máxima potencia son 17.5 voltios y 5.7 amperios respectivamente.
- El inversor utilizado, convierte 12 voltios de corriente directa en 120 voltios de corriente alterna. La onda entregada por este dispositivo es senoidal modificada. La potencia máxima del inversor es de 400 vatios.
- El controlador de carga es de tipo MPPT, diseñado para paneles y baterías y de 12 ó 24 voltios. La corriente máxima de carga es de 10 amperios, y tiene una función de selección de tipo baterías. Además de esto cuenta con un puerto de comunicación serial UART para la recepción de las variables del sistema.
- El microcontrolador STM32L432KC es un microcontrolador de 32 bits con arquitectura ARM. Posee una frecuencia máxima de CPU de 80MHz, 256 KB de memoria flash para ROM, y 64KB de memoria SRAM. Cuenta con 8 pines analógicos con resolución de 12 bits, 11 pines digitales de los cuales 5 son

PWM. A pesar de tener características de muy alto rendimiento, el consumo de corriente de este dispositivo se encuentra en el orden de los miliamperios, el cual lo hace adecuado para este tipo de aplicaciones. Este será el componente principal del controlador de acople.

- La batería del sistema es una batería de plomo, de 12 voltios con una capacidad de carga de 2Ah 20H.
- Para las medidas de corriente se usará el sensor SCT013, con capacidad de medir hasta 30 amperios. Sus valores se toman a través de las entradas analógicas del microcontrolador. Este es un sensor no invasivo, es decir que se coloca alrededor de una de las líneas de la fuente de alimentación para su funcionamiento.
- La conexión o desconexión de la red al sistema fotovoltaico de acuerdo con las condiciones especificadas, se lo hará mediante relés que serán activados por el microcontrolador.
- Un cargador de baterías de 12 voltios será utilizado para recargar las baterías desde la red eléctrica.
- El sistema de supervisión se hará utilizando un Raspberry Pi. Este ordenador recibirá todos los datos recolectados por el microcontrolador a través de sus puertos USB, y los acomodará en una interfaz gráfica para la supervisión y control del sistema.

3.1.3. Funciones del Sistema

De la descripción del sistema y considerando el problema de investigación, se ha determinado que el sistema propuesto deberá de disponer de al menos las siguientes funciones para que cumpla con el propósito fundamental de este trabajo:

- Monitoreo y visualización de potencia de la carga, potencia de generación solar, y potencia de la red a través de una interfaz gráfica de usuario accedida por medio de un explorador web.
- Acoplamiento y desacoplamiento automático de las cargas hacia la red eléctrica o al sistema fotovoltaico.
- Detección de desconexión de la red eléctrica

3.2. Consideraciones Iniciales de Diseño

Dentro de las consideraciones iniciales de diseño, se prevé analizar y establecer las condiciones fundamentales sobre las que actuará el sistema, y que permitirán definir los detalles finales de los diferentes componentes del dispositivo de acoplamiento automatizado.

3.2.1. Análisis del Entorno y Espacio Físico

El estudio realizado se hará en el sector norte de la ciudad de Guayaquil. Según las estadísticas del Atlas Solar Global, la irradiación solar diaria en la ciudad es de 4.566 KWh/m². Para poner en perspectiva, con estas condiciones un sistema fotovoltaico de 1KW de potencia generaría 3.312 KWh por día.

El prototipo por desarrollarse está destinado a ser utilizado para uso doméstico, y dado el correcto dimensionamiento del sistema, este puede cubrir cargas de cualquier tipo, ya sea para iluminación, electrodomésticos, climatización, etc. El dimensionamiento del sistema se encuentra ligado al poder de adquisitivo del usuario, ya que mientras más potencia entregue el sistema fotovoltaico, mayor será el costo de implementación.

3.2.2. Análisis de los Componentes del Sistema de Acoplamiento Automatizado

La maqueta desarrollada para la demostración del sistema está limitada a una potencia de 100 watts. Las cargas por utilizarse serán netamente resistivas, mediante el uso de focos. La tensión del sistema es de 120 voltios en corriente alterna.

La maqueta tiene dos paneles eléctricos, uno en el lado de corriente alterna y otro en lado de corriente directa. El panel AC acopla las líneas de alimentación de la red, y del inversor hacia el enchufe de las cargas a través del controlador de acoplamiento. A las líneas de la red se encuentran conectados, el cargador de la batería, y un convertidor AC-DC que dará una señal de 3.3V al microcontrolador cuando la red se encuentre disponible. El panel DC, acopla las líneas de los paneles solares, las baterías, la entrada DC del inversor, y la salida DC del cargador de la batería hacia el controlador de carga del sistema fotovoltaico. La figura 20 el diagrama de bloques del sistema y la figura 21 muestra los dispositivos del sistema.

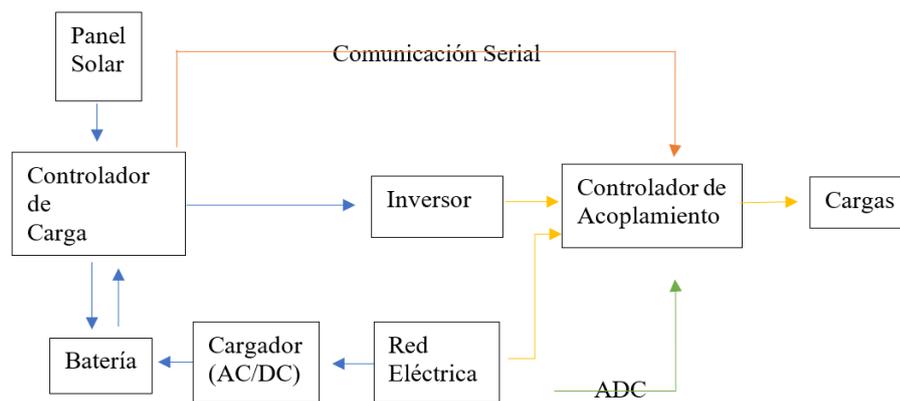


Figura 20 Diagrama de bloques del sistema
Fuente: Elaboración Propia

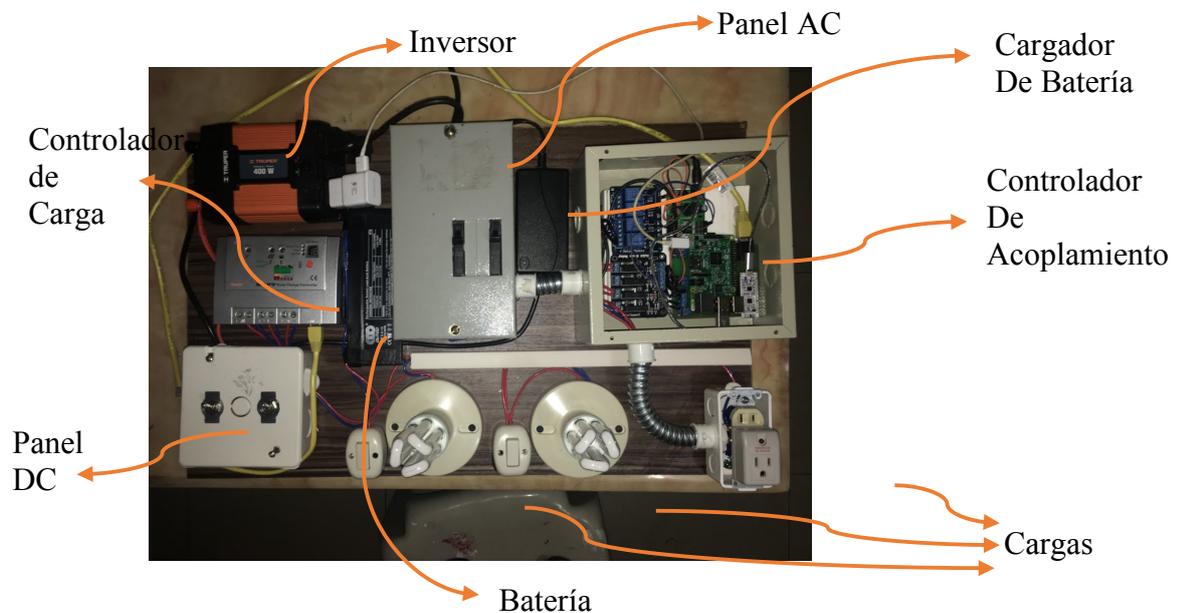


Figura 21 Maqueta con los componentes del sistema
Fuente: Elaboración Propia

El controlador de acoplamiento consiste en el microcontrolador, los módulos de relés de estado sólido y mecánicos, el sensor de corriente, el convertidor AC-DC y el Raspberry Pi. Los relés denominados SSR1, SSR2, SSR3 y RLE1 en la figura 21, serán controlados por medio de las salidas digitales del microcontrolador, y la conmutación de estos se dará dependiendo del estado en el que se encuentre el sistema como se indicó en la sección 3.1.1. Los disyuntores de la figura 21 denominados D1 y D2 sirven de protección ante sobre corriente y corto circuitos. Los interruptores denominados SW1, y SW2 en la figura 22, servirán para la desconexión de la batería o el panel solar

en caso de ser necesario. La entrada del inversor y la batería tendrán un fusible de 10 amperios para protección contra sobre cargas o cortocircuitos. El controlador recibirá las variables del sistema a través de un sensor de corriente y el puerto de comunicación serial del controlador de carga. Con estas variables se podrán obtener las potencias de las fuentes y la potencia de la carga las cuales servirán para definir el acoplamiento de las cargas hacia una de las fuentes. Estos datos son enviados vía USB al sistema de supervisión instalado en el Raspberry Pi, el cual podrá ser accedido desde un explorador web utilizando la dirección IP del equipo.

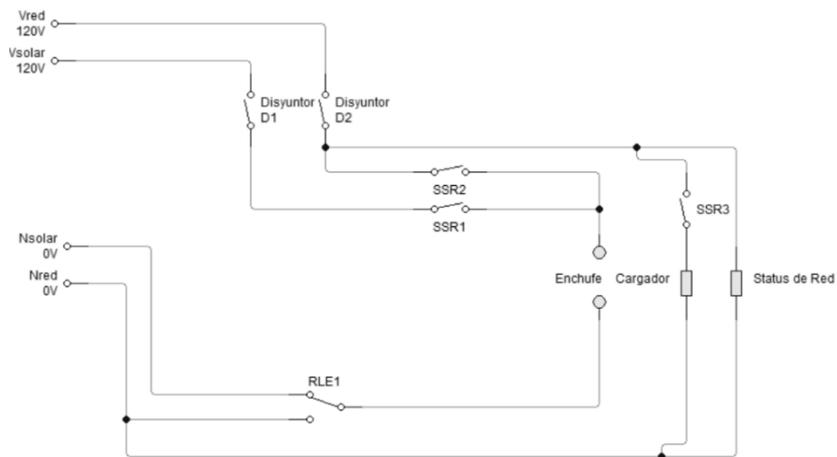


Figura 22 Esquema de conexiones del panel AC
Fuente: Elaboración Propia

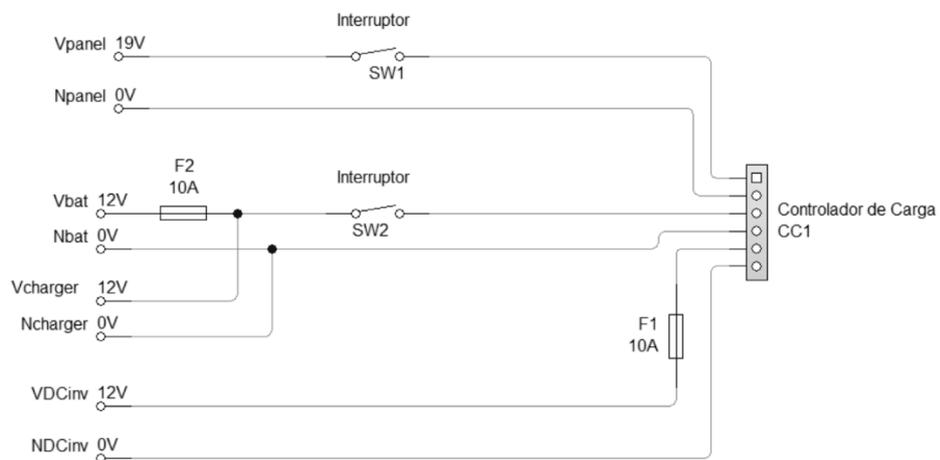


Figura 23 Esquema de conexiones del panel DC
Fuente: Elaboración Propia

Tabla 2
Tabla de posición de los relés del sistema de acoplamiento

Estado	SSR1	SSR2	SSR3	RLE1
Apagado	Abierto	Abierto	Abierto	NC
Conexión a Red	Abierto	Cerrado	Definido por el usuario	NO
Conexión a Inversor	Cerrado	Abierto	Abierto	NC

Fuente: Elaboración Propia

3.3. Diseño Hardware del Controlador de Acoplamiento

En este apartado se analiza los componentes del sistema desde un enfoque hardware, es decir los componentes, conexiones y alimentación. El hardware del sistema de acoplamiento está compuesto por el microcontrolador, el Raspberry Pi, el sensor de corriente, el convertidor AC-DC utilizado para verificar el estado de la red y los relés. La figura 23 muestra los componentes del sistema de acoplamiento.

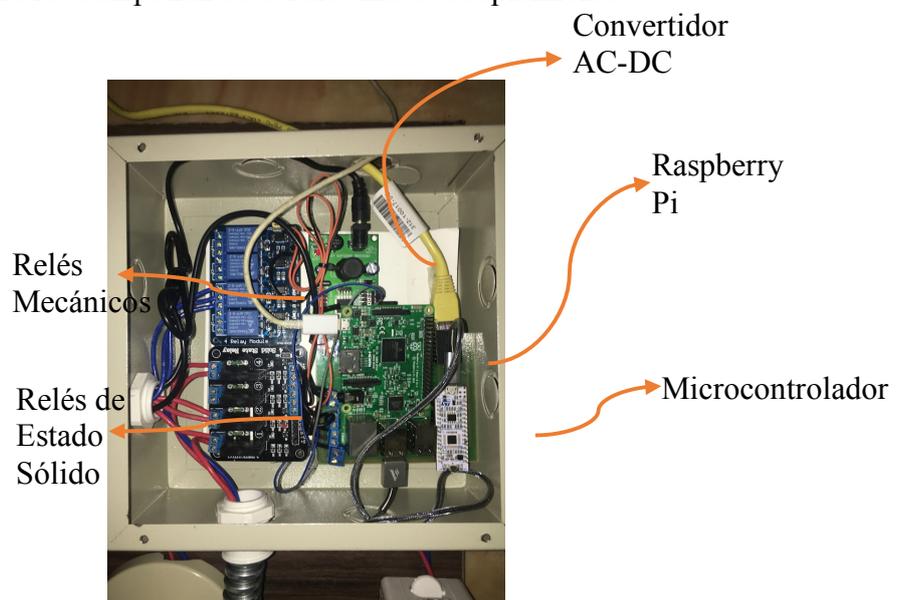


Figura 24 Componentes del Controlador de Acoplamiento
Fuente: Elaboración Propia

3.3.1. Circuito de Control

El componente principal del circuito de control es el microcontrolador. Este dispositivo es el cerebro del sistema, es decir controlará la conmutación de los relés, la lectura del sensor de corriente, y la comunicación con el controlador de carga y el Raspberry Pi. La alimentación del microcontrolador se dará a través del puerto USB del Raspberry Pi.

Para el control de los relés se utilizaron 4 pines digitales configurados como salida, las cuales entregan una señal de 3.3 voltios o 0 voltios. El estado de la red se verifica utilizando un convertidor AC-DC el cual entrega una señal de 3.3 voltios que será leída por uno de los pines digitales del microcontrolador configurado como entrada. Para la lectura del sensor de corriente se utilizó uno de sus pines analógicos, con una resolución de 12 bits. La comunicación con el Raspberry Pi, y el controlador de carga se da por medio de las interfaces UART que contiene el microcontrolador. La figura 24 muestra el esquemático del circuito de control, y la figura 25 muestra la PCB desarrollada.

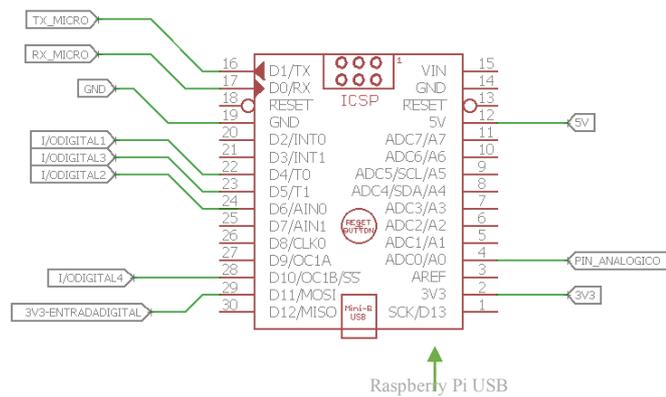


Figura 25 Esquema de conexiones del microcontrolador
Fuente: Elaboración Propia

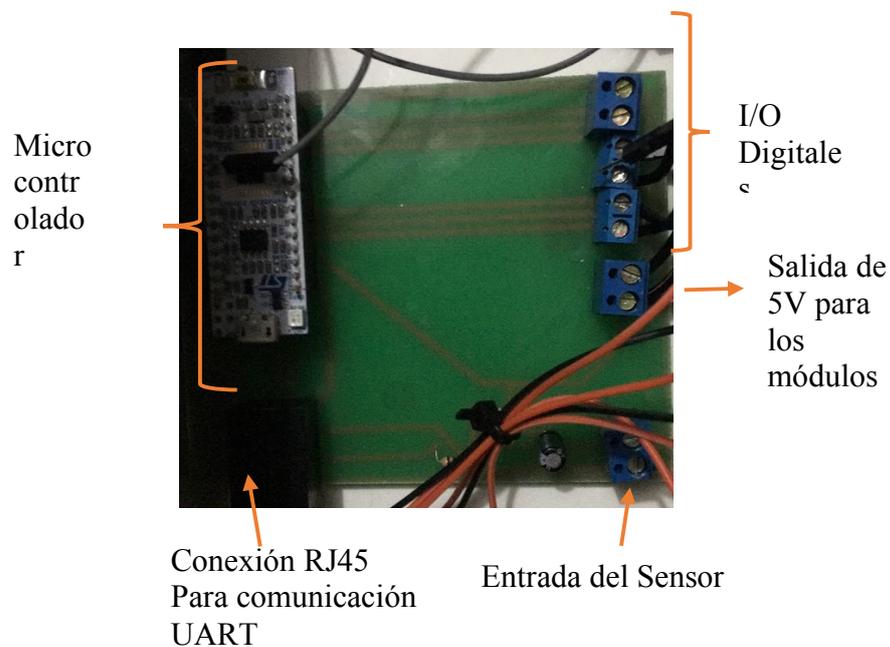


Figura 26 PCB del controlador de acoplamiento
Fuente: Elaboración Propia

3.3.2. Adquisición de Datos: Sensores y Controlador de Carga

En esta sección se describirán las conexiones con los dispositivos utilizados para obtener las variables del sistema.

3.3.2.1. Sensor de Corriente

Para obtener la medida de corriente de la red, se utilizó el sensor SCT-013. Este sensor es común en proyectos de este tipo debido a la disponibilidad de recursos y documentación para su uso. El sensor es un transformador de corriente que cuenta con un lado primario y un secundario. Una de las líneas de alimentación de la fuente de energía alterna hace la función de primario del transformador, mientras que, en el secundario a través de un resistor de carga (Burden Resistor) se obtiene un voltaje proporcional a la corriente del primario. A través de una interfaz de resistores y capacitores, esta señal proporcional puede ser leída por una entrada analógica del microcontrolador. La figura 26 muestra el sensor de corriente SCT-013 y sus terminales, y la figura 27 el circuito equivalente del sensor.



Figura 27 Sensor de Corriente
Fuente: Elaboración Propia

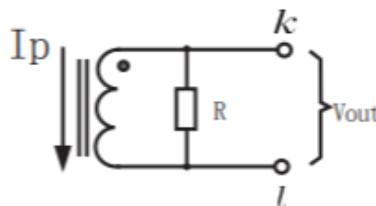


Figura 28 Circuito equivalente del sensor de corriente utilizado
Fuente: (Seed Studio, 2017)

El resistor de carga denominado R en la figura es utilizado para obtener la señal de voltaje proporcional y se calcula de la siguiente manera:

- I. Se escoge un valor de corriente máxima a ser leída por el microcontrolador.
- II. La corriente máxima escogida se multiplica por raíz de 2 para obtener su valor pico, ya que se está asumiendo un valor RMS.

$$I_{\text{primario}_{\text{pico}}} = I_{\text{rms}} * \sqrt{2} \quad (1)$$

- III. Se divide la corriente pico a medirse para el número de vueltas del secundario del transformador, para obtener el valor de corriente pico del secundario.

$$I_{\text{secund}_{\text{pico}}} = \frac{I_{\text{primario}_{\text{pico}}}}{n_{\text{vueltas}_{\text{secundario}}}} \quad (2)$$

- IV. Finalmente, para obtener el valor de resistor de carga ideal, se divide la mitad del voltaje de referencia del convertidor analógico a digital para la corriente pico del secundario. Esto se hace debido a que la señal de entrada tiene que ser desplazada de tal manera que su oscilación no tenga valores negativos ya que el microcontrolador no tiene voltaje de referencia negativo. Al hacer los cálculos utilizando el voltaje de referencia dividido para 2, se previene que el pico de la señal sobrepase al voltaje de referencia cuando esta sea desplazada.

$$R_{\text{burden}} = \frac{\frac{V_{\text{ref}}}{2}}{I_{\text{secund}_{\text{pico}}}} \quad (3)$$

Para el cálculo del valor del resistor de carga se asumió una corriente I_{rms} de 2 amperios, resultando en un valor de resistor de 1 K Ω . La figura 28 muestra la interfaz entre el sensor de corriente y el microcontrolador.

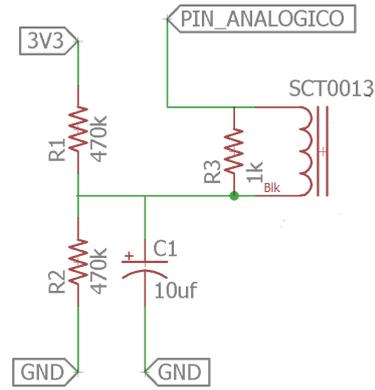


Figura 29 Interfaz del sensor con el microcontrolador
Fuente: Elaboración Propia

Las resistencias R1 y R2 de la figura x se utilizaron para crear un divisor del voltaje de referencia del microcontrolador, mientras que R3, es el resistor de carga del transformador. Al conectar este divisor de voltaje a la señal del transformador de corriente, la salida se verá desplazada 1.65 voltios hacia arriba del eje de coordenadas, resultando en un voltaje que oscila entre 3.3 voltios y 0 voltios. Esto se ilustra en la figura 29.

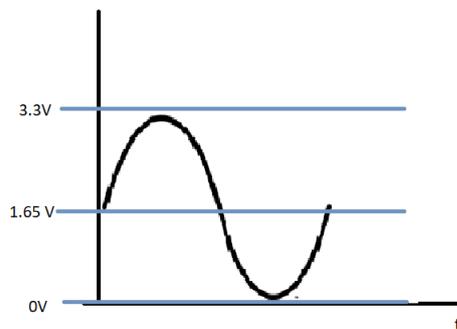


Figura 30 Onda de Entrada al pin analógico del microcontrolador
Fuente: Elaboración Propia

3.3.2.2. Sensor de Estado de Red Eléctrica

Para saber el estado de la red eléctrica, simplemente se utilizó, un convertidor AC-DC de 120 voltios a 3.3 voltios. El terminal de 3.3 voltios del convertidor se conecta a la entrada digital D10 del microcontrolador mientras que el terminal neutro se conecta a la tierra del microcontrolador. La figura 30 muestra el esquema de conexión entre el convertidor AC-DC y el microcontrolador.

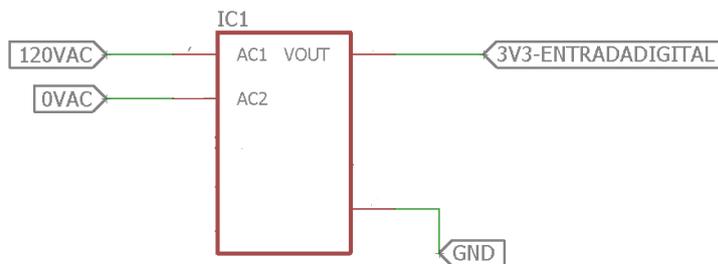


Figura 31 Esquema simplificado del sensor de estado de la red eléctrica
Fuente: Elaboración Propia

3.3.2.3. Controlador de Carga

El controlador de carga envía información de los parámetros del panel solar y la batería a través de un puerto de comunicación serial UART. Este dispositivo usa el conector RJ45 para el bus de comunicación, el cual tiene 8 pines codificados en colores y normalmente es utilizado para comunicación Ethernet. El controlador de carga sólo utiliza los pines 4, 5, y 6 del conector RJ45 que corresponden a GND, TX, y RX respectivamente. Convenientemente las líneas de transmisión de datos del dispositivo funcionan a niveles TTL, es decir 3.3 voltios para una señal de 1 lógico, y 0 voltios para una señal de 0 lógico, simplificando la interfaz con el microcontrolador, ya que se conectarían directamente sin ningún problema. La figura 31 muestra el esquema de conexión entre el microcontrolador y el controlador de carga. La figura 32 ilustra el controlador de carga. Las etiquetas RX_MICRO y TX_MICRO pertenecen a los pines D0 y D1 del microcontrolador respectivamente.

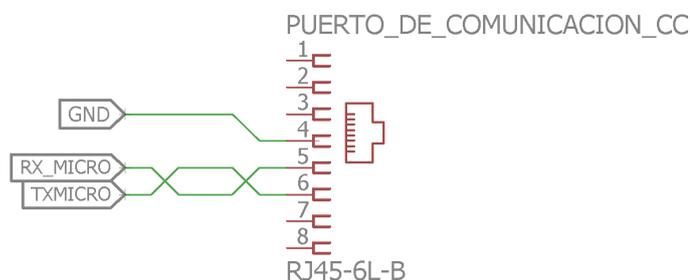


Figura 32 Esquema de conexión del controlador de carga y el microcontrolador
Fuente: Elaboración Propia



Figura 33 Conexión física del conector RJ45 al controlador de carga
Fuente: Elaboración Propia

3.3.2.4. Relés de Conmutación

Para la conmutación de la conexión de las cargas hacia la red o hacia el inversor, se utilizaron módulos de relés. Uno de estos módulos ofrece 4 canales de relés en estado sólido, mientras que el segundo ofrece 4 canales de relés mecánicos. La figura 33 muestra los módulos utilizados. Cada relé del módulo es activado con una señal de 3.3 o 5 voltios. La alimentación de los módulos es de 3.3 o 5 voltios provenientes del microcontrolador. El sistema utiliza 3 canales del módulo de relés en estado sólido, y un canal del módulo de relés mecánicos. Los relés de estado sólido son utilizados para las líneas vivas del sistema, uno para la red y otro para el inversor; y el tercero para la activación o desactivación del cargador de batería.

El relé mecánico acopla las líneas neutras de las fuentes como se ilustró en la sección 3.2.2. El pin común del relé se encuentra conectado a la línea neutra de las cargas, mientras que los contactos normalmente abierto, y normalmente cerrados se encuentran conectados en los neutros de la red y del inversor respectivamente. La figura 35 muestra el esquema de conexión de los relés.

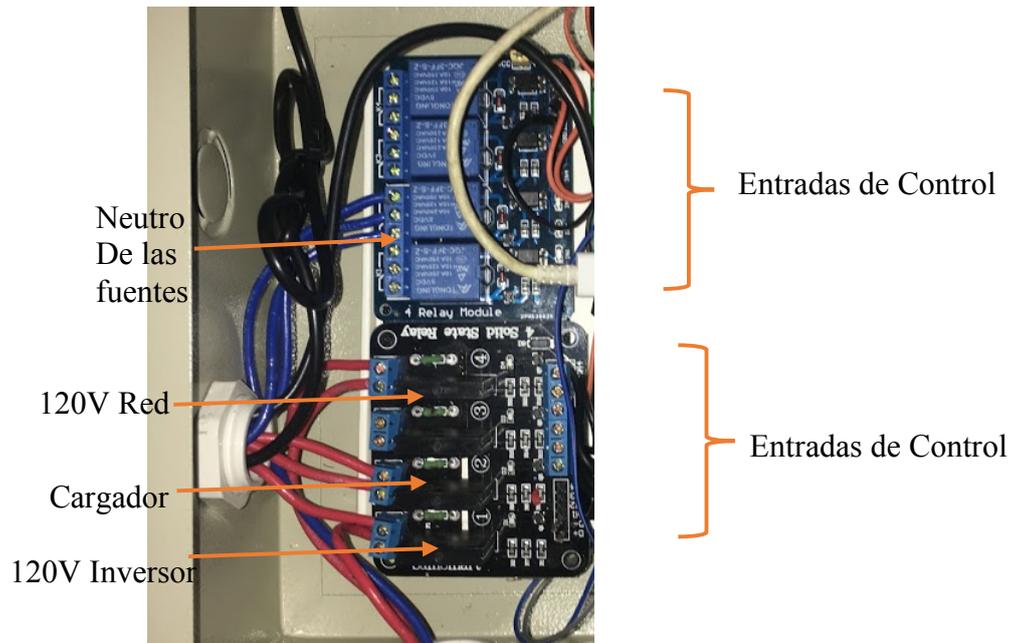


Figura 34 Módulos de Relés utilizados
Fuente: Elaboración Propia

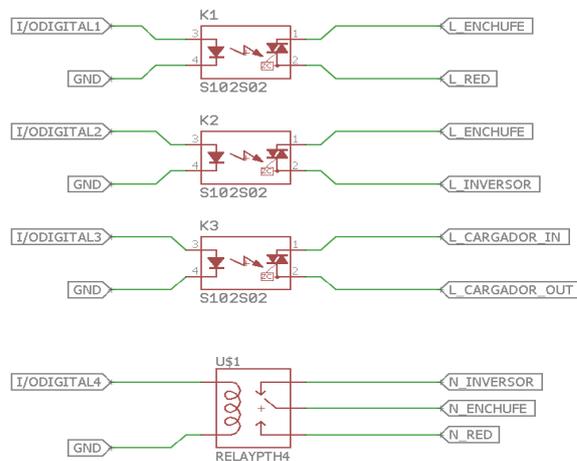


figura 35 Esquema de conexión de los relés al microcontrolador
fuente: Elaboración Propia

3.4. Diseño Software del sistema de Acoplamiento

En este apartado se realizará el análisis del programa del microcontrolador, el protocolo utilizado para la comunicación con el controlador de carga y la interfaz de usuario del Raspberry Pi.

3.4.1. Firmware de Control

Para un mejor entendimiento del firmware de control se ha decidido subdividirlo en 5 secciones, que son:

- Rutina de Recepción de comandos del Raspberry Pi
- Rutina de Recepción de datos del controlador de carga
- Bucle Principal o Lógica de Control
- Rutinas para conmutación de Relés
- Envío de Datos al Raspberry Pi

3.4.1.1. Recepción de Comandos

El microcontrolador cuenta con un esquema de interrupciones muy versátil que facilita la recepción de comandos sin interrumpir el flujo de ejecución de las instrucciones del algoritmo. Debido a esto la rutina de la recepción de datos se puede considerar una subdivisión del firmware de control. El diagrama de flujo de la figura 36 ilustra el proceso de recepción de comandos.

El bloque de configuración de la figura inicializa el periférico UART 1 a una velocidad de 115200 BAUD. Al recibir un comando, el microcontrolador genera una interrupción y procesa lo recibido. La descripción de los comandos se muestra en la tabla 3

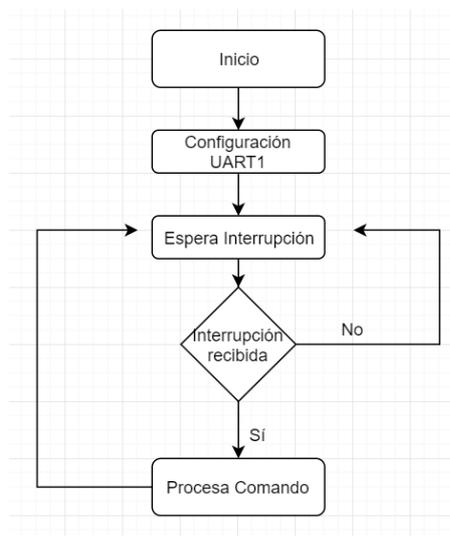


Figura 36 Diagrama de flujo para la recepción de datos
Fuente: Elaboración Propia

Tabla 3
Tabla de Comandos del sistema embebido

<i>Comando Enviado</i>	<i>Acción</i>
0x65	El sistema inicia en modo automático
0x4F	Se abren todos los relés
0x68	Activa modo Automático
0x72	Modo conexión a red
0x63	Enciende el cargador de la batería
0x6E	Apaga el cargador de la batería

Fuente: Elaboración Propia

3.4.1.2. Recepción de Datos del controlador de Carga y Sensor de Corriente

Para la recepción de datos del controlador de carga se utilizó un temporizador del microcontrolador para obtener los datos cada segundo. La figura 37 muestra el diagrama de flujo que ilustra el proceso.

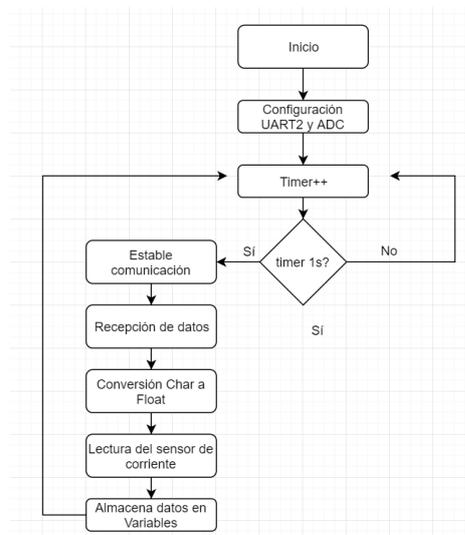


Figura 37 Diagrama de flujo para la recepción de datos del controlador de carga
Fuente: Elaboración Propia

En el bloque de configuración de la figura 36, se inicializa el segundo periférico UART a una velocidad de 9600 BAUD determinado por el controlador de carga y se configura el sensor de corriente utilizando la librería Emonlib que provee de funciones para inicialización, calibración y lectura de datos. Una vez configurada la comunicación UART en el microcontrolador, para iniciar la comunicación con el controlador de carga se envía un código de inicio de 6 bytes, y luego un código adicional de 6 bytes, para la sincronización de la comunicación. Una vez iniciada la comunicación y los dispositivos en sincronía, el microcontrolador envía un código de comando para recibir los datos del controlador de carga. El código recibido es una cadena de 24 bytes, de los cuales sólo se utilizarán los bytes que contienen la información de voltaje de la batería, voltaje de los paneles solares, corriente de la carga, y corriente de los paneles solares. La figura 38 muestra todo el paquete de datos utilizados en la comunicación entre estos dispositivos.



Figura 38 Trama de datos utilizados en la comunicación con el controlador de carga
Fuente: Elaboración Propia

De los 24 bytes recibidos, 17 de estos bytes corresponden a los datos del sistema fotovoltaicos, de los cuales los utilizados para el desarrollo del sistema se muestran en la tabla 4.

Los datos se reciben en código hexadecimal, los cuales deben ser convertidos a un valor numérico flotante. Por ejemplo, una lectura de 12.5 voltios del byte número 1 que corresponde al voltaje de la batería es recibida en hexadecimal como 0x04E2. Este valor hexadecimal corresponde a 1250 y para ser transformado a valor flotante simplemente se divide para 100.

Tabla 4
Tabla de información recibida por el controlador de carga

<i>No.</i>	<i>Info</i>	<i>Cantidad de bytes</i>
1	Voltaje de la batería	2 bytes
2	Voltaje del Panel Solar	2 bytes
4	Corriente consumida	2 bytes
13	Indicador de batería llena	1 byte
16	Corriente de carga de batería	2 bytes

Fuente: Elaboración Propia

3.4.1.3. Bucle Principal

El bucle principal es el que determinará la rutina de conmutación de los relés. En esta sección del programa del microcontrolador se hará uso de las variables obtenidas por el sensor de corriente y el controlador de carga. El diagrama de flujo del bucle se muestra en la figura 39.

Después del bloque de configuración que configura los pines digitales del microcontrolador, se verifica el estado el modo de operación del sistema que por defecto se iniciará en modo automático. En este modo se verifica si la potencia de la carga es mayor a la potencia solar. De ser esto verdadero, se procede a verificar el estado de la red; si esta se encuentra disponible, es decir, no ha habido apagones, el algoritmo continúa con la observación del nivel de la batería. Si esta se encuentra en un nivel mayor al 50%, la carga seguirá acoplada al inversor del sistema fotovoltaico. Caso contrario, la carga se acoplará a la red hasta que el nivel de la batería se haya recargado hasta en un 90% utilizando la energía solar. En el segundo modo de operación, las cargas se acoplan a la red sin importar las condiciones del sistema fotovoltaico o la batería.

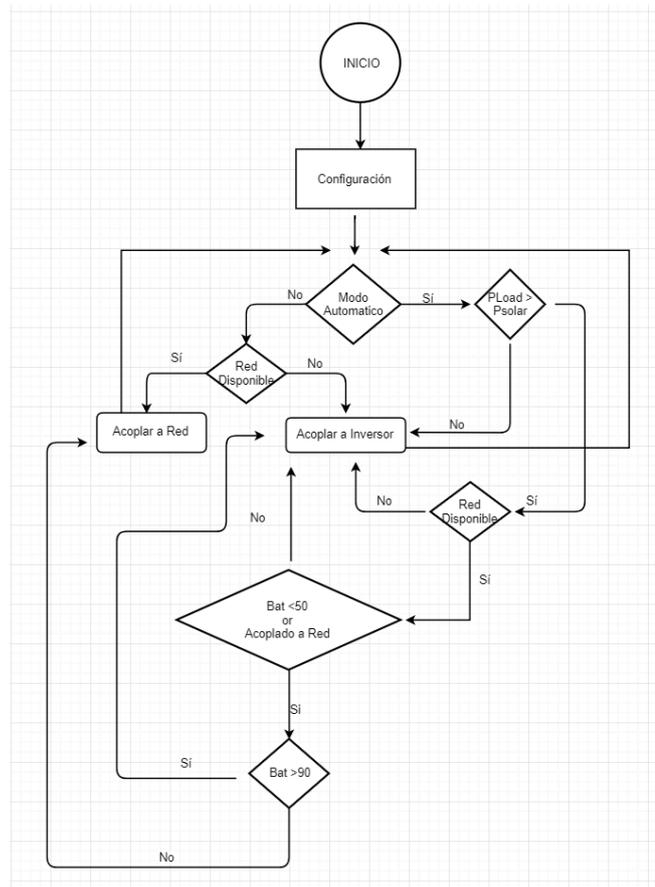


Figura 39 Diagrama de flujo del bucle principal
Fuente: Elaboración Propia

3.4.1.4. Conmutación de Relés

La sección del programa que determina la conmutación de los relés se ilustra en las figuras 40 y 41 respectivamente. El bucle principal es el que determina si la carga se debe acoplar a la red o al inversor.

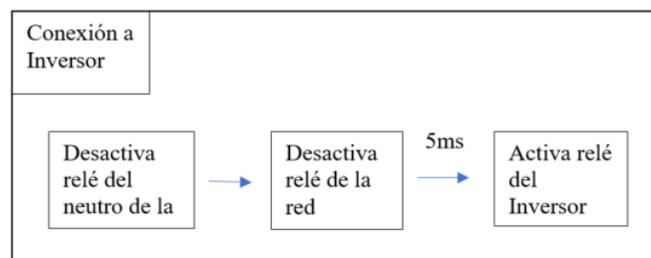


Figura 40 Rutina del microcontrolador para el acoplamiento hacia el inversor
Fuente: Elaboración Propia

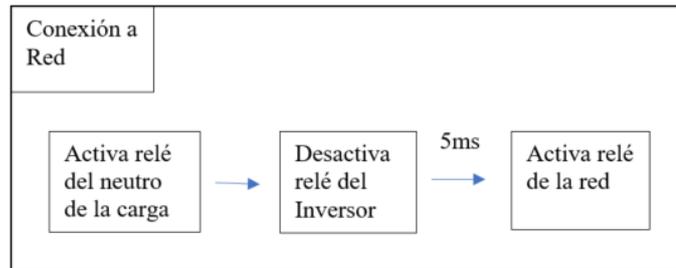


Figura 41 Rutina del microcontrolador para el acoplamiento hacia la red
Fuente: Elaboración Propia

El proceso descrito en las figuras asegura que no haya conflicto entre las fuentes al conectarse a la carga, previniendo daños en la misma. El tiempo de acople de las cargas a las fuentes es de 5 milisegundos. Con esto se evitará causar interrupciones de energía a la carga.

3.4.1.5. Envío de Datos

El envío de datos al Raspberry Pi se hace mediante el puerto serial UART 1. Los datos se han ordenado de tal manera que sean separados con una coma, para facilitar la recepción en el ordenador. Los datos enviados al ordenador se disponen de la siguiente manera: potencia solar, potencia de la red, potencia de consumo, porcentaje de batería, indicador de fuente de energía, indicador de estado de la red eléctrica.

3.4.2. Interfaz de Usuario

La interfaz de usuario del sistema se hace gracias al Raspberry Pi. Este dispositivo cuenta con el sistema operativo Linux. Sus dimensiones reducidas y sus opciones de comunicación como Wifi, y Ethernet lo hacen ideal para el desarrollo de interfaces que puedan ser accedidas a través de cualquier dispositivo con un explorador web. La figura 42 muestra un diagrama de bloque para acceder al ordenador.

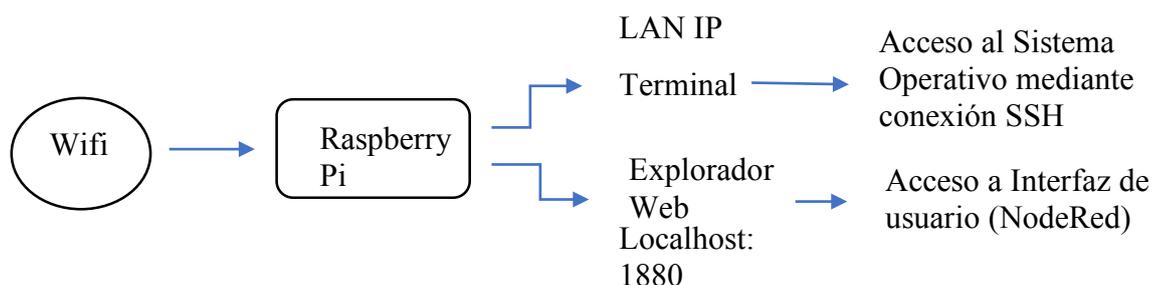


Figura 42 Diagrama de acceso al sistema operativo del Raspberry Pi
Fuente: Elaboración Propia

Para acceder tener acceso al Raspberry Pi, este tiene que ser conectado por medio de Ethernet o WIFI a la red LAN. El acceso al ordenador se lo hace a través de SSH (Secure-Shell), por medio de la consola o terminal de otro computador que se encuentre conectada a la misma red LAN del Raspberry Pi, el cual se ilustra en las figuras 43 y 44. Esta es la forma más sencilla de acceder al sistema operativo del ordenador. En este caso se ha utilizado Tera Term.

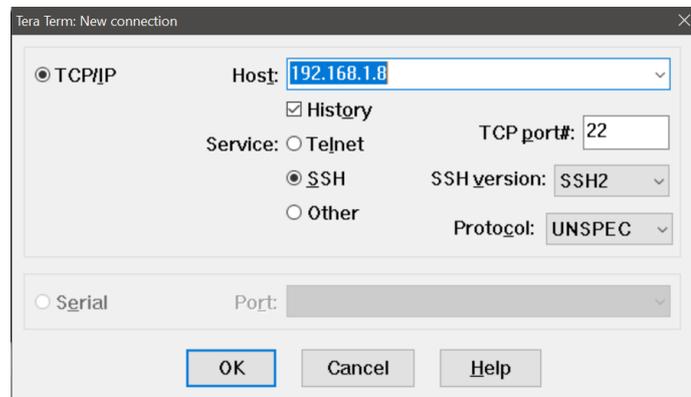


Figura 43 Ventana de configuración de conexión SSH mediante TCP/IP
Fuente: Elaboración propia

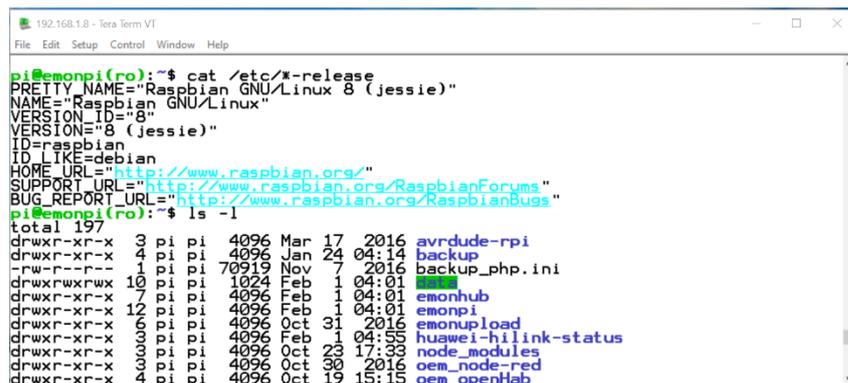


Figura 44 Ventana de la interfaz de línea de comando de Linux
Fuente: Elaboración Propia

Utilizando la conexión SSH con el Raspberry Pi, se podrá instalar los programas utilizados para la creación de la interfaz de usuario.

Para obtener los datos en el ordenador, se utilizó un programa llamado Node-RED. Este programa se accede mediante el puerto 1880 de la dirección IP del Raspberry Pi, utilizando un explorador web. Cabe recalcar que el computador con el que se accede al puerto y el Raspberry Pi deben estar conectados a la misma red LAN. La figura 45 muestra la ventana de inicio del programa Node-RED.

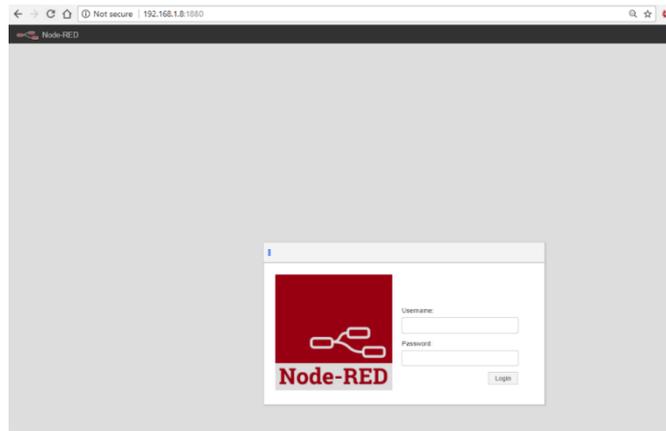


Figura 45 Ventana principal del programa Node-RED
Fuente: Elaboración Propia

La programación de este programa se basa en nodos, los cuales cumplen una función específica. Con estos nodos se pueden hacer flujos (Flows), que describirán el comportamiento de la interfaz que se quiera crear. Para la comunicación entre el microcontrolador y el programa se utiliza un nodo llamado Serial, el cual permite establecer la comunicación serial entre los dispositivos. La cadena de datos que se envía al programa se lo hace en formato CSV (comma separated values). Este formato delimita cada variable enviada por medio de comas. Por ejemplo, si se quiere enviar tres variables en formato CSV, se haría de la siguiente manera: Var1, Var2, Var3. La imagen 45 muestra el entorno de programación de Node-RED. Este programa también contiene herramientas que permiten crear interfaces de usuario llamados Dashboards para la visualización y control de datos. La figura 46 muestra la interfaz gráfica de usuario del programa utilizado, y la figura 47 muestra la interfaz de visualización creada para el sistema desarrollado. En la figura 47 se pueden observar 3 secciones llamadas Control, Status, e Historial. La sección de control contiene los switches para el control del sistema, etiquetados I/O, Modo, y Cargador. El switch I/O sirve para apagar o encender el sistema; el switch Modo selecciona el modo de operación que pueden ser Auto, o Red.

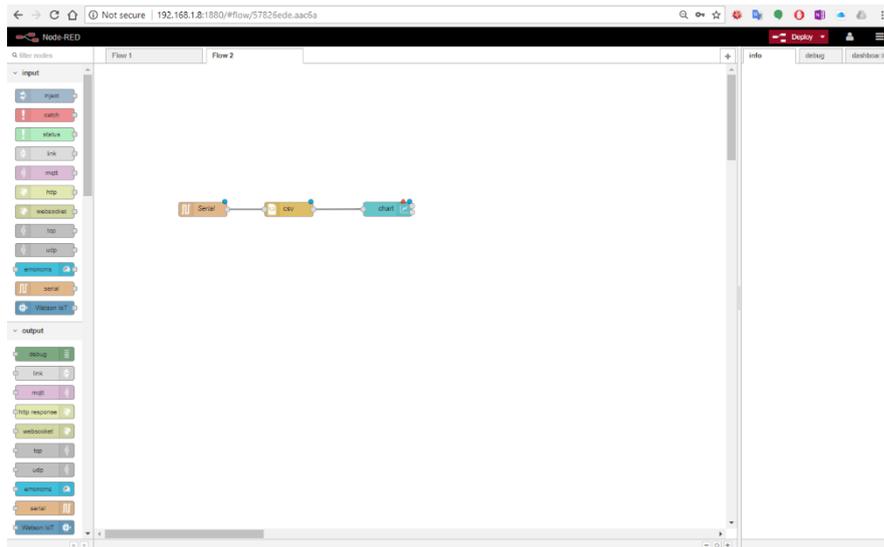


Figura 46 Interfaz de programación del programa Node-RED
Fuente: Elaboración Propia

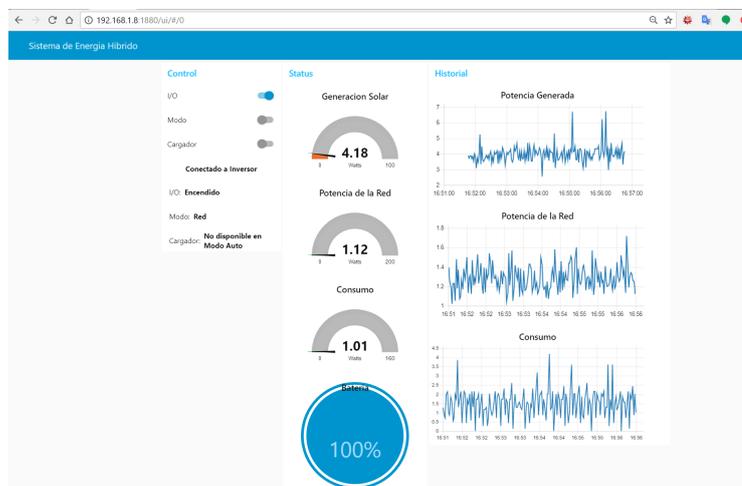


Figura 47 Interfaz de usuario del sistema de acoplamiento
Fuente: Elaboración Propia

El switch de cargador sirve para encender o apagar el cargador de la batería. Cada switch tiene un indicador, que indica el estado que corresponde a cada posición, es decir si el sistema se encuentra encendido o apagado, el modo en el que se encuentra, y si el cargador se encuentra encendido o no. La sección etiquetada Status, muestra la potencia de generación, potencia consumida de la Red, el consumo general, y el porcentaje de batería. La sección llamada Historial, muestra un gráfico en el tiempo de cada una de las potencias del sistema.

Los datos mostrados son enviados desde el microcontrolador cada segundo, mientras que los switches envían comandos al microcontrolador cada vez que estos sean accionados. El microcontrolador genera una interrupción con cada comando recibido desde el programa, y realiza las acciones correspondientes.

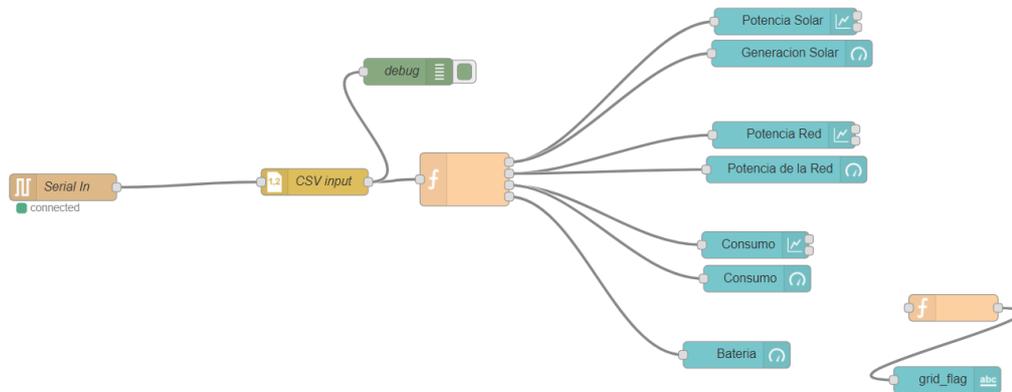


Figura 48 Nodos de programación para la recepción de datos del microcontrolador
Fuente: Elaboración propia

La figura 48 muestra los nodos utilizados para la recepción de datos del microcontrolador. El bloque Serial In se encarga de la recepción de datos del puerto serial para ser enviados al nodo etiquetado CSV input. Este bloque se encarga de separar en variables individuales la cadena de datos en formato CSV para la utilización de estas en el programa. Luego de esto, el bloque de función separa las variables para ser enviadas a los nodos de las secciones Status, e Historial representados en color celeste.

La imagen 49 muestra los nodos utilizados para el envío de comandos desde el Raspberry Pi hacia el microcontrolador. Los nodos de color celeste representan los switches de la interfaz de usuario. Estos retornan un valor verdadero o falso dependiendo de la posición en que se encuentren. Los bloques de función reciben estos datos booleanos y deciden que comando enviar a los siguientes bloques según la posición de los switches. Una vez determinado el comando estos se envían a los nodos de texto de la interfaz representados en color celeste, y a los nodos denominados Serial Out. Los comandos enviados al microcontrolador se especifican en la tabla 5.

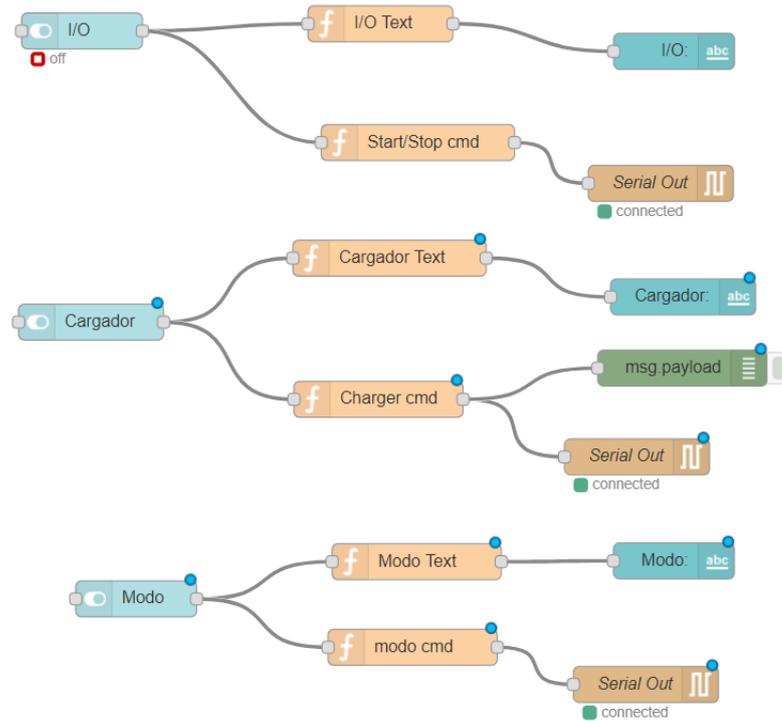


Figura 49 Nodos utilizados para el control del sistema de acoplamiento
Fuente: Elaboración Propia

Tabla 5
Tabla de Comandos del Sistema

<i>Switches</i>	<i>Posición</i>	<i>Comando Enviado</i>	<i>Acción</i>
I/O	Encendido	0x65	El sistema inicia en modo automático
	Apagado	0x4F	Se abren todos los relés
Modo	Encendido	0x68	Activa modo Automático
	Apagado	0x72	Modo conexión a red
Cargador	Encendido	0x63	Enciende el cargador de la batería
	Apagado	0x6E	Apaga el cargador de la batería

Fuente: Elaboración Propia

3.5. Pruebas Experimentales del Sistema

Las pruebas experimentales del sistema permiten determinar si el prototipo cumple con la descripción y funciones asignadas al inicio del capítulo. Las funciones permiten definir el comportamiento del sistema y constituyen un conjunto de requisitos que el sistema debe cumplir para resolver el problema que se ha planteado en este trabajo de investigación. En este apartado se realizan diferentes pruebas o ensayos que permiten determinar el cumplimiento real o experimental de las funciones básicas del sistema.

3.5.1. Pruebas del Sistema Fotovoltaico

Para facilitar el desarrollo del algoritmo de programación del microcontrolador, se puso a prueba el sistema para observar el comportamiento de sus elementos en diferentes condiciones de cargas posibles. Gracias a esto se pudo determinar la carga máxima que soporta el sistema, el tiempo de descarga de la batería, y por último el comportamiento del sistema al acoplar la red a las baterías utilizando un convertidor DC/AC. En la figura 50, se observan el panel solar, la batería, el controlador de carga, el inversor y el protoboard que contiene al microcontrolador.



Figura 50 Set de prueba de las características del sistema fotovoltaico
Fuente: Elaboración Propia

Utilizando una fuente de corriente variable, se pudo variar la impedancia de entrada del inversor. Con los datos recogidos se crearon gráficos utilizando Excel, mostrados en la figura 51

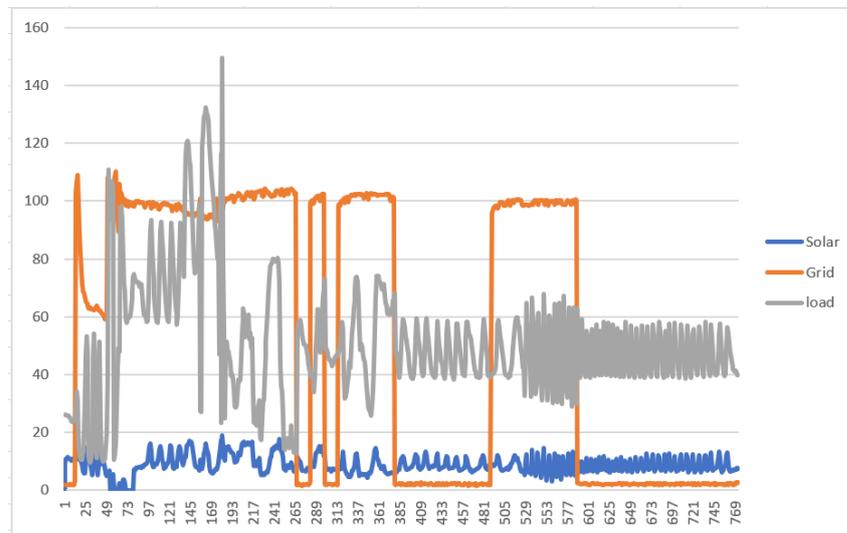


Figura 51 Datos obtenidos de las pruebas del sistema fotovoltaico y sensor de corriente de la red
Fuente: Elaboración Propia

La leyenda del gráfico de la figura 51 indica los colores que representa cada gráfico. Los picos que se observan en el gráfico naranja fueron generados al conectar el cargador de batería, mientras se encontraba alimentando a la fuente de corriente a través del inversor. Se puede notar que estos picos son mayores al de lo que se está consumiendo, indicando que no sólo suple el déficit de generación solar marcado de azul, sino que también carga las baterías. El principio del sistema era que las baterías idealmente sean cargadas con energía solar y en caso de que se necesite más energía de lo que los paneles solares producen, el convertidor AC-DC, sólo entregue la diferencia del consumo y la producción. Al observar un comportamiento diferente al esperado se pudo realizar un mejor algoritmo de programación para el sistema de acoplamiento.

3.5.2. Pruebas del sensor de Corriente

Objetivo de la prueba: Calibrar el sensor hasta obtener una medida precisa.

Variables que intervienen: Constante de calibración, corriente de carga.

Una vez calculado el valor de resistor de carga del sensor para obtener medidas de hasta 2 amperios, la calibración del sensor se hace mediante software. Utilizando una librería para el sensor, se ajustó la constante de calibración a base de prueba y error

utilizando una carga de 10 vatios a 120 voltios, hasta llegar a la medida deseada. La figura 52 muestra un gráfico con las lecturas de potencia utilizando diferentes constantes de calibración.

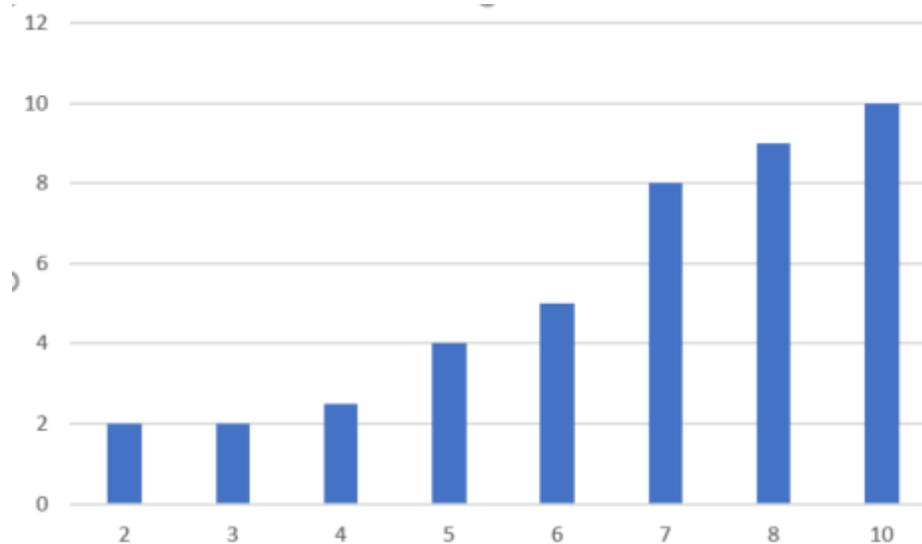


Figura 52 Gráfico de calibración. El eje x pertenece a la constante de calibración mientras que el eje y, pertenece a la potencia leída.
Fuente: Elaboración Propia

3.5.3. Pruebas del controlador de Carga

Las pruebas del controlador de carga se hicieron netamente para explorar el protocolo de comunicación del dispositivo. Leyendo el manual del dispositivo, se pudo crear la sección de programa del microcontrolador que cumpla esta función. En general no hubo problemas en cuanto a la comunicación del dispositivo con el microcontrolador. La figura 53 muestra la respuesta del controlador de carga mostradas en el monitor serial del ordenador.

3.5.4. Pruebas de Acoplamiento

Para las pruebas de acople se realizó un programa distinto al principal que conmutaba entre los relés entre la red y el inversor. Originalmente se utilizó un inversor de 160 vatios de potencia con una carga inductiva. Esto resultó en la sobrecarga del inversor debido a los picos de corriente generados por la carga al variar su alimentación entre la red y el inversor.

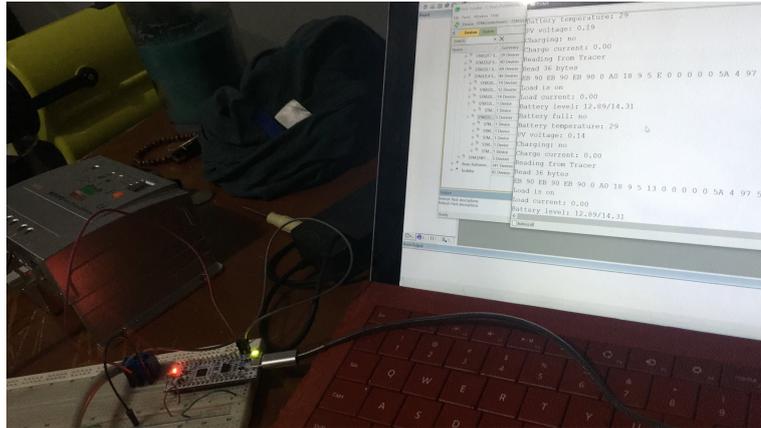


Figura 53 Respuesta recibida del controlador de carga
Fuente: Elaboración Propia

A consecuencia de esto, se utilizó un inversor de más capacidad y las pruebas se realizaron sin cargas para prevenir los incidentes similares al ocurrido. Una vez comprobado que los relés funcionen adecuadamente, se realizaron las pruebas con una carga resistiva. Se utilizaron diferentes periodos de conmutación para determinar el mejor tiempo de desacople y acople de las fuentes utilizadas. Con un periodo de 20ms se observó, que la carga interrumpía notablemente su funcionamiento. A un periodo de 1ms, ocurrían corto circuitos. Finalmente se pudo determinar un tiempo de acople de una fuente a otra de 5 milisegundos, resultando en una conmutación sin interrupciones. Con la verificación de este tiempo de conmutación, se procedió a realizar la prueba del programa principal del sistema.

Los resultados de las pruebas del programa principal fueron satisfactorios. El sistema de acoplamiento se conectaba al inversor al desconectar la red, tal y como se esperaba. Utilizando la fuente de corriente variable, se pudo determinar que el sistema se acople a la red o al inversor dependiendo de la potencia que esta consuma y del estado de la batería.

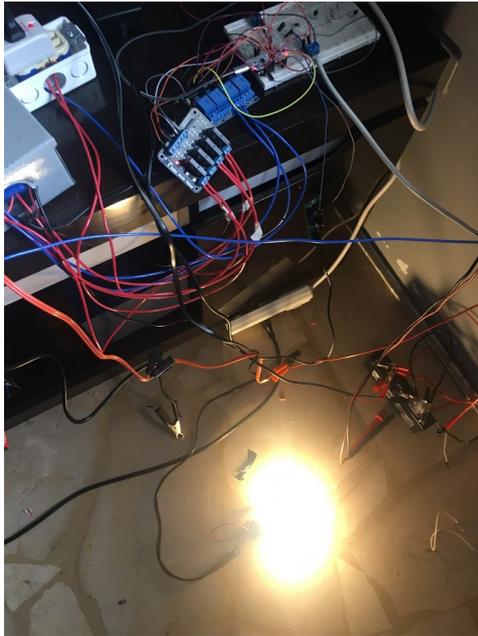


Figura 54 Pruebas de conmutación de la carga hacia las fuentes de energía
Fuente: Elaboración Propia



Figura 55 Fuente corriente variable utilizado para las pruebas del sistema
Fuente: Elaboración Propia

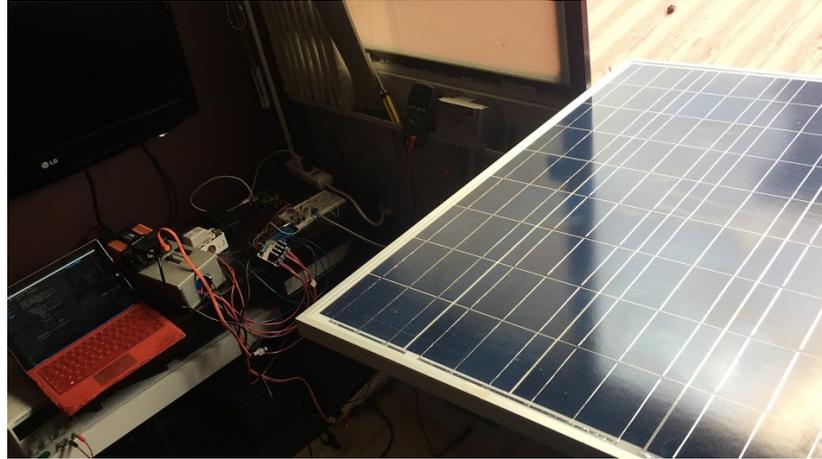


Figura 56 Pruebas del bucle principal del microcontrolador
Fuente: Elaboración Propia

3.5.5. Pruebas de la Interfaz de Usuario

La interfaz de usuario resultó eficaz y no ocasionó problemas en cuanto a su funcionamiento. El acceso a la interfaz es estable y confiable, ya que no existe ninguna pérdida de información al momento de recibir los datos. Las pruebas de envío de datos se realizaron enviando comandos al microcontrolador una inmediatamente después de la otra con el fin de determinar pérdidas de información al momento de enviar los comandos.

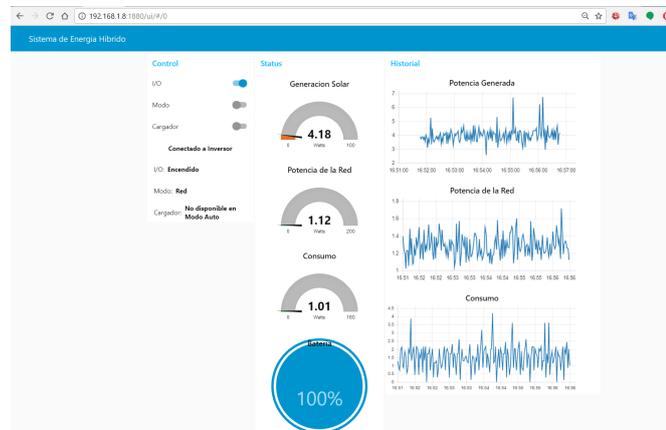


Figura 57 Interfaz de usuario del sistema de acoplamiento
Fuente: Elaboración Propia

3.6. Cálculo de Costos

En esta sección se mostrará una lista de los componentes utilizados para el desarrollo de la maqueta junto con el costo correspondiente.

Tabla 6
Cálculo de costos del proyecto

Descripción	Precio
Inversor de Potencia	\$120
Controlador de Carga Tracer 1210rn	\$120
Panel Solar 100W	\$156.80
Módulos de Relés	\$30
STM321432KC	\$15
Sensor de Corriente SCT-001	\$16
Batería 12V plomo	\$16
Componentes eléctricos (Cables, Paneles, disyuntores, etc	\$80.61
Raspberry Pi 3	\$60
Total	\$614.41

Fuente: Elaboración Propia

CAPÍTULO IV PRESENTACIÓN DE RESULTADOS, ANÁLISIS DE RESULTADOS Y DISCUSIÓN

En este capítulo se presentará los resultados obtenidos del sistema automatizado de acoplamiento, y se realizará un análisis del diseño, el sistema embebido, la interfaz de usuario y el funcionamiento del sistema en general.

4.1. Resultado del diseño y análisis del prototipo desarrollado

En esta sección del capítulo se realizará la presentación del resultado de investigación, la cual se orienta a realizar un resumen de los logros alcanzados en este trabajo de investigación, enfocándose en mostrar detalles generales constructivos, descriptivos o de funcionamiento de los resultados mostrados.

4.1.1. Sistema Fotovoltaico

El sistema de acoplamiento general consta del sistema fotovoltaico como fuente y principal y la red eléctrica como la fuente auxiliar. Para la implementación del sistema fotovoltaico se utilizó un panel solar de 100 vatios de potencia, un inversor de potencia de 400 vatios, un controlador de carga de 10 amperios y una batería de plomo de 30 amperios-hora de carga. Las limitaciones del sistema fotovoltaico están dadas por su capacidad de producción de energía, mas no por el tipo de carga que se utilice. Para la realización del prototipo, se utilizaron cargas resistivas.

4.1.2. Sistema de Control

Culminado el desarrollo del prototipo, se obtuvo un sistema de control que permite obtener las variables del sistema con precisión y rapidez gracias a las características del microcontrolador utilizado. Estas características son necesarias para obtener un sistema que pueda reaccionar de manera instantánea antes las circunstancias que ocurran durante el uso del prototipo. Es decir, reaccionar a los cambios de la cantidad de potencia consumida y potencia producida.

4.1.3. Sistema de Acoplamiento Automatizado

Se logró desarrollar un sistema capaz de acoplar las cargas a la red o al sistema fotovoltaico en un tiempo de 5 milisegundos dependiendo de las condiciones descritas en la programación del sistema de control. Este bajo tiempo de acoplamiento, resulta en un cambio de fuente de alimentación sin interrupción alguna del funcionamiento de las cargas.

4.1.4. Interfaz de Usuario Remota

Con la interfaz remota, la visualización y control de las variables del sistema se pudo lograr de manera sencilla y rápida. Al utilizar una interfaz remota con acceso mediante un explorador web, se evita la utilización de displays o pantallas en el sistema físico. Puede decirse que se agrega un nivel de confort a los usuarios del sistema.

4.2. Discusión de Resultados

El análisis por describirse en esta sección permite entregar una idea más clara en cuanto a los alcances del sistema. En esta sección se desarrollará un breve análisis de los resultados obtenidos en este trabajo de investigación, confrontándolo con el planteamiento del problema, la hipótesis, los objetivos y la metodología empleada, de tal forma que permita determinar si los resultados obtenidos son coherentes con la realidad del problema planteado, y si se ha obtenido un aporte significativo para la comunidad académica, que valide y justifique su desarrollo.

4.2.1. Análisis de las funciones del sistema

Las funciones del sistema son la descripción detallada de las acciones que debe realizar el prototipo. En esta sección se analizará el cumplimiento de estas, considerando las pruebas desarrolladas en el Capítulo III.

En el capítulo anterior se describieron las siguientes funciones:

- Monitoreo y visualización de las potencias del sistema.
- Acoplamiento y Desacoplamiento automático de las cargas

- Detección de fallo de la red eléctrica.

El monitoreo y visualización de las variables se dieron gracias al controlador de carga, el sensor de corriente y el Raspberry Pi. El acoplamiento de la carga hacia las fuentes de alimentación junto con la detección de fallo de la red eléctrica se pudo realizar gracias al microcontrolador utilizado para el sistema de control.

4.3. Cumplimiento de los Objetivos

4.3.1. Objetivos Específicos

Dentro de los objetivos específicos para la creación del sistema automatizado se planteó el diseño de un prototipo de maqueta de un sistema fotovoltaico utilizando elementos y componentes de generación reducida; la implementación un sistema de acople y desacople automatizado utilizando microcontroladores; y el diseño de un sistema de supervisión y control basado en un ordenador de placa reducida. El cumplimiento de estos objetivos se encuentra demostrado en el capítulo anterior.

4.3.2. Objetivo General

El objetivo general de la investigación planteaba el desarrollo de un sistema capaz de acoplar las cargas de una vivienda a un sistema fotovoltaico o a la red eléctrica dependiendo de la demanda de esta. El cumplimiento del objetivo se alcanzó siguiendo los objetivos específicos planteados en la sección anterior.

4.4. Análisis de la metodología

La metodología utilizada para la elaboración de esta investigación fue fundamental para la comprensión y cumplimiento de los objetivos. Los recursos bibliográficos fueron de gran ayuda para obtener los fundamentos sólidos requeridos para el desarrollo de la investigación. La experimentación con cada uno de los componentes del sistema permitió desarrollar algoritmos eficaces para el firmware de control.

4.5. Análisis de la hipótesis

En la hipótesis se planteó un tiempo de por lo menos 100 milisegundos para el acoplamiento de la carga del sistema hacia una de las fuentes sin producir interrupciones en el funcionamiento de esta. El tiempo de acople de la carga hacia las fuentes desarrollado en esta investigación es de 5 milisegundos, dando por verdadera la hipótesis planteada.

4.6. Análisis de la solución del problema

El problema planteado en la investigación describía la inhabilidad de los sistemas fotovoltaicos conectados a la red de continuar su operación cuando la red eléctrica falle. Otro de los problemas descritos, era la falta de servicio de medida de energía neta en el país la cual hace que los sistemas fotovoltaicos acoplados a la red eléctrica sean inconvenientes. Con el desarrollo del prototipo planteado en la investigación permite acoplar la red eléctrica a un sistema fotovoltaico de tal manera que no se necesite un medidor de energía neto, ni componentes especiales como inversores con sincronizadores de onda para que el sistema fotovoltaico sea interactivo con la red. La utilización de acumuladores permite usar la energía solar almacenada en casos de emergencia cuando la red eléctrica falle, logrando solucionar otro de los problemas descritos.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Al término de la presente investigación, luego de haber hecho un análisis de los resultados se puede concluir con lo siguiente:

- El uso de microcontroladores de alto rendimiento permite desarrollar sistemas con un alto nivel de desempeño en cuanto a nivel de procesamiento, resolución de las lecturas de sensores y tiempo de respuesta ante los estímulos a los que debe reaccionar.
- Un tiempo de acople/desacople de al menos 20ms en la red eléctrica de una vivienda, no genera perturbaciones en los electrodomésticos.
- Tener un sistema de visualización de energía permite a los usuarios tener una mejor perspectiva de los patrones del consumo de esta, resultando en un mejor manejo del uso de energía. Con la disponibilidad de ordenadores de placa reducida con capacidad operar bajo un sistema operativo como Linux, el desarrollo de estos sistemas de visualización se puede hacer de manera remota bajo una red LAN.

5.2. Recomendaciones

- Es altamente recomendado el uso de protecciones contra cortocircuitos o sobre cargas como disyuntores y fusibles.
- Antes de conectar una carga a un sistema asegurarse que su consumo de potencia no exceda las especificaciones de los componentes del sistema.
- Se recomienda no dejar cables expuestos al realizar las conexiones de los componentes.

5.3. Trabajo Futuro

El sistema desarrollado puede ser mejorado considerablemente mediante el uso de técnicas de inteligencia artificial para el control de acoplamiento. Con la inteligencia artificial se puede realizar patrones de consumo que podrán servir para predecir el momento de acoplamiento de una fuente a otra, o predecir la disponibilidad de radiación solar con anticipación.

REFERENCIAS BIBLIOGRÁFICAS

- Abdel, O., & Orabi, M. (2011). High Gain Single-Stage Inverter for Photovoltaic AC Modules. Aswan City, Aswan, Egipto. Obtenido de https://www.researchgate.net/profile/Omar_Abdel-Rahim/publication/234080597_High_gain_single-stage_inverter_for_photovoltaic_AC_modules/links/0912f50eed5f21eb84000/High-gain-single-stage-inverter-for-photovoltaic-AC-modules.pdf
- Ali Mazidi, M., Naimi, S., & Naimi, S. (2011). the avr microcontroller and embedded system. New Jersey, New York, EEUU.
- AltE Store. (2017). What is a solar charge controller. Obtenido de AltE Store: <https://www.altestore.com/store/info/solar-charge-controller/>
- Anantwar, H., Lakshmikantha, B. R., & Sundar, S. (2017). Fuzzy self tuning PI controller based inverter control for voltage regulation in off-grid hybrid power system . Obtenido de Science Direct: https://ac.els-cdn.com/S187661021732386X/1-s2.0-S187661021732386X-main.pdf?_tid=7815bdd8-be98-11e7-aa29-00000aacb362&acdnat=1509494952_f5244c8aab28b436432d9524c95755cc
- Baskys, A. (2012). Microcontrollers. Obtenido de Vilnius Gediminas Technical University: http://dSPACE.vgtu.lt/bitstream/1/1381/1/1339_Baskys_Microcontrllers_Micro_controllers_web.pdf
- Bruni, G., Cordiner, S., Galeotti, M., Mulone, V., Nobile, M., & Rocco, V. (2013). Control Strategy Influence on the Efficiency of a Hybrid Photovoltaic-Battery-Fuel. Obtenido de Science Direct: https://ac.els-cdn.com/S1876610214000277/1-s2.0-S1876610214000277-main.pdf?_tid=dfaf9be8-bdc7-11e7-9963-00000aacb35e&acdnat=1509405361_4257c6b884ec7bbab7224000a51f67b5
- Castejón, A., & Santamaría, G. (2010). Instalaciones Solares Fotovoltaicas. Editex.
- Climate Scope. (2017). Ecuador. Obtenido de Climate Scope: <http://global-climatescope.org/en/country/ecuador/#/enabling-framework>
- Dharmaian, B., Nanjappa, A., & Gounden, V. (2014). Hybrid power electronic controller for combined operation of constant power and maximum power

- point tracking for single-phase grid-tied photovoltaic systems. Obtenido de IET Journals: <http://dl.matlabproject.ir/form/files/236875.pdf>
- Diab, F., Lan, H., Zhan, L., & Ali, S. (2015). An environmentally friendly factory in Egypt based on hybrid photovoltaic/wind/diesel/battery system. Obtenido de Science Direct: <http://www.sciencedirect.com/science/article/pii/S0959652615008884>
- Doucet, J., Eggleston, D., & Shaw, J. (2006). DC/AC Pure Sine Wave Inverter. Obtenido de Worcester Polytechnic Institute: https://web.wpi.edu/Pubs/E-project/Available/E-project-042507-092653/unrestricted/MQP_D_1_2.pdf
- Enerpoint. (2017). La célula fotovoltaica. Obtenido de Enerpoint: http://www.enerpoint.es/fotovoltaic_technology_2.php
- Environmental Protection Agency. (2017). Energy and the Environment. Obtenido de Distributed Generation of Electricity and its Environmental Impacts: <https://www.epa.gov/energy/distributed-generation-electricity-and-its-environmental-impacts>
- Harish, A., & Prasad, M. (2013). Microcontroller Based Photovoltaic MPPT Charge Controller. Obtenido de IJETT Journal: <http://ijettjournal.org/volume-4/issue-4/IJETT-V4I4P304.pdf>
- Honeywell. (2017). Hall Effect Sensing and Application. Obtenido de Honeywell: <https://sensing.honeywell.com/hallbook.pdf>
- Ismail, M., Moghavvemi, M., Mahlia, T., Muttaqi, K., & Moghavvemi, S. (2014). Effective utilization of excess energy in standalone hybrid renewable. Obtenido de Science Direct: <http://ro.uow.edu.au/cgi/viewcontent.cgi?article=4401&context=eispapers>
- Jouda, A., Elyes, F., Rabhi, A., & Abdelkader, M. (2016). Optimization of Scaling Factors of Fuzzy–MPPT Controller for Stand-alone. Obtenido de Science Direct: https://ac.els-cdn.com/S1876610217302916/1-s2.0-S1876610217302916-main.pdf?_tid=a92e3826-be9f-11e7-83d0-00000aacb362&acdnat=1509498040_dab6ecdc23640e3ed4f34106c0c2d76e
- Kuoro, S., Leon, J. I., Vinnikov, D., & Franquelo, L. G. (s.f.). Grid-Connected Photovoltaic Systems: An Overview of Recent Research and Emerging PV Converter Technology. Recuperado el 31 de Octubre de 2017, de Research Gate: https://www.researchgate.net/profile/Leopoldo_Franquelo/publication/27588

9741_Grid-

Connected_Photovoltaic_Systems_An_Overview_of_Recent_Research_and_Emerging_PV_Converter_Technology/links/5548c9170cf27c50006708ee/Grid-Connected-Photovoltaic-Systems-An-Ove

Mcdonald, J., Wojszczyk, B., Flynn, B., & Voloh, I. (2013). Obtenido de Distribution Systems, Substations and Integration of Distributed Generation: file:///C:/Users/josep/Desktop/New%20folder/9781461458296-c1.pdf

Méndez, M. J., & Cuervo, G. R. (2012). Energía Solar Fotovoltaica. Madrid: Fundación Confemetal.

Microchip, Sergiu, O., & Rosu-Hamzescu, M. (2013). Practical Guide to Implementing Solar Panel MPPT Algorithms. Obtenido de Microchip: <http://ww1.microchip.com/downloads/en/AppNotes/00001521A.pdf>

Mohammad B., S., & Robert S., B. (2014). Multi-Objective Optimization and Design of Photovoltaic-Wind Hybrid System for Community Smart DC Microgrid. Obtenido de IEEE: <https://pdfs.semanticscholar.org/9c4f/caf2ec0d869f09128b61044d2b699d7b0107.pdf>

Monroy, I. (2002). La generación de energía eléctrica y el ambiente. México: Gaceta Ecológica.

Pukhrem, S. (2013). How Solar Cells Work - Components & Operation of Solar Cells. Obtenido de Solar Love: <http://solarlove.org/how-solar-cells-work-components-operation-of-solar-cells/>

Reisi, A. R., Moradi, M. H., & Jamasb, S. (2012). Classification and Comparison of Maximum Power Point Tracking techniques for Photovoltaic Systems: A review. Obtenido de Elsevier: https://s3.amazonaws.com/academia.edu.documents/39234029/Classification_and_comparison_of_maximum_power_point_tracking_techniques_for_photovoltaic_system_A_review.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1509401060&Signature=YZI4jKIjGD7Nus07lC0m4ze

SineTech. (2017). PV Solar Systems. Obtenido de Sinetech: <http://www.sinetech.co.za/solar-systems.html>

Sunfields Europe. (2016). Módulo Fotovoltaico: Listado de Parámetros fundamentales. Obtenido de Sunfields Europe: <https://www.sfe-solar.com/noticias/articulos/modulo-fotovoltaico-parametros-fundamentales/>

- Traore, A., Taylor, A., Zohdy, M., & Peng, F. (2017). Modeling and Simulation of a Hybrid Energy Storage System for Residential Grid-Tied Solar Microgrid Systems. Obtenido de Scientific Research Publishing: http://file.scirp.org/pdf/JPEE_2017052614520528.pdf
- Valencia Quintero, J. P. (2008). Generación Distribuida: Democratización de la Energía Eléctrica. Bogotá, Colombia.
- Yiu, J. (2014). The definitive guide to ARM Cortex M3 and Cortex M4 procesors. Oxford: Elsevier.
- Zipp, K. (2016). What are the different types of solar inverters? Obtenido de Solar Power World: <https://www.solarpowerworldonline.com/2016/05/different-types-solar-inverters/>

ANEXOS

ANEXO A: Código Fuente del Controlador

```
/*
Coupling Controller
Autor: Joseph Peñafiel
Microcontrolador: STM32L4K32C

    Copyright 2018 Joseph Adrián Peñafiel Caicedo. Licensed under the
    Educational
    Community License, Version 2.0 (the "License"); you may not use this
    file
    except in compliance with the License. You may obtain a copy of the
    License at

    http://opensource.org/licenses/ECL-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT
    WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
    the
    License for the specific language governing permissions and limitations
    under
    the License.

*/

#include "EmonLib.h"
EnergyMonitor ct;

#define V_AC          122 // Voltaje de la red
#define BAT_MIN      11.6 // voltaje minimo de bateria
#define CHARGER_SSR  3   // ssr del cargador (channel 2)
#define GRID_SSR     2   // ssr de la red (channel 4)
#define SOLAR_SSR    4   // ssr del inversor (channel 1)
#define N_RELAY      9   // transfer relay HIGH -> NC ; LOW -> NO
#define GRID_PIN     10  // input estado de red 1 -> on; 0 -> off
#define INTERVAL     1500
#define SOLAR        0
#define GRID         1

// ===== Variables ===== //

// Variables para el Controlador de Carga //
uint8_t start[] = { 0xAA, 0x55, 0xAA, 0x55, 0xAA, 0x55,
                   0xEB, 0x90, 0xEB, 0x90, 0xEB, 0x90
                 };
uint8_t id      = 0x16;
uint8_t cmd[]  = { 0xA0, 0x00, 0xB1, 0xA7, 0x7F }; // comandos del controlador
de carga
uint8_t buff1[128]; // almacena los datos recibidos
```

```

// variables para HMI //
int  incoming      = 0; // datos recibidos del hmi
float solar_power  = 0.00; // potencia solar
float load_power   = 0.00; // potencia consumida
float grid_power   = 0.00; // potencia de la red
int   bat_percent  = 0;
bool  auto_mode    = true; // el programa empieza en modo hibrido
bool  system_on    = false;
char  buff2[200]; // datos enviados al HMI

// variables para calculos //

int   grid_status  = 0;
float I_rms        = 0.00;
float V_batt       = 0.00;
float V_solar      = 0.00;
float charge_current = 0.00;
float load_current  = 0.00;
float battery_max   = 0.00;
uint8_t battery_full;
int   grid_flag    = 0;
int   state        = 0;
bool  use_grid     = false;
bool  use_solar    = false;
unsigned long previous_millis = 0;

// =====//

void setup() {
  Serial.begin(115200); // usb
  Serial1.begin(9600); // controlador de carga
  analogReadResolution(12);
  ct.current(A0, 8) ; // Current: p_in, calibr.
  system_on = false;
  pinMode(CHARGER_SSR, OUTPUT);
  pinMode(GRID_SSR, OUTPUT);
  pinMode(SOLAR_SSR, OUTPUT);
  pinMode(N_RELAY, OUTPUT);
  pinMode(GRID_PIN, INPUT);

  digitalWrite(CHARGER_SSR, LOW);
  digitalWrite(GRID_SSR, LOW);
  digitalWrite(SOLAR_SSR, LOW);
  digitalWrite(N_RELAY, HIGH); // NC conectado a 0V de inversor
}

//--- Prototipos de Funciones ---//
float to_float(uint8_t* buffer, int offset) {

```

```

    unsigned short full = buffer[offset + 1] << 8 | buff1[offset];
    return full / 100.0;
}

//-----//

void loop() {
/*
 * Recepcion de comandos desde el GUI
 */

if (Serial.available() > 0) {
    incoming = Serial.read();
    if (incoming == 'h') { // modo hibrido
        digitalWrite(CHARGER_SSR, LOW);
        auto_mode = true;
    }
    else if (incoming == 'r') { // modo conectado a la red
        auto_mode = false;
        use_grid = false;
    }
    else if (incoming == 'e') { // sistema encendido
        system_on = true;
    }
    else if (incoming == 'o') { // sistema apagado
        system_on = false;
        digitalWrite(SOLAR_SSR, LOW);
        digitalWrite(GRID_SSR, LOW);
        digitalWrite(CHARGER_SSR, LOW);
        digitalWrite(N_RELAY, HIGH);
    }
    else if (incoming == 'c') { // conectar cargador
        digitalWrite(CHARGER_SSR, HIGH);
    }
    else if (incoming == 'n') { // desconectar cargador
        digitalWrite(CHARGER_SSR, LOW);
    }
}

/*****/

/*
 * Obtiene Datos del sistema cada 1.5 segundos
 */
unsigned long current_millis = millis();
if (current_millis - previous_millis >= INTERVAL) {
    previous_millis = current_millis;
    Serial1.write(start, sizeof(start));
    Serial1.write(id);
    Serial1.write(cmd, sizeof(cmd));
}
}

```

```

int idx = 0;
for (int i = 0; i < 255; i++) {
    if (Serial1.available()) {
        buff1[idx] = Serial1.read();
        idx++;
    }
}
V_batt      = to_float(buff1, 9);
V_solar     = to_float(buff1, 11);
load_current = to_float(buff1, 15);
battery_max  = to_float(buff1, 19);
battery_full = buff1[27];
charge_current = to_float(buff1, 30);
I_rms       = ct.calcIrms(1480); // medicion de corriente rms

}
/*****

/*
 * Variables para el HMI
 */

grid_power = I_rms * V_AC; // Potencia aparente de la red
solar_power = V_solar * charge_current; // potencia de los paneles
bat_percent = map(V_batt, BAT_MIN, battery_max, 0, 100);
grid_status = digitalRead(GRID_PIN); // 1 on, 0 off
if (grid_flag == 0) load_power = load_current * V_batt; // potencia de la carga;
else load_power = grid_power;
/*****

/*
 * Bucle Principal
 */

/* Decide la seleccion del estado */
if (system_on == true) {
    if (auto_mode == true) { // modo hibrido
        if (load_power > solar_power) { // chequea el estado del sistema
            if (grid_status == 1) {
                if (bat_percent < 50 || use_grid == true) { // chequea la bateria
                    if (bat_percent > 90) state = SOLAR;
                    else state = GRID;
                }
            }
            else {
                state = SOLAR;
            }
        }
        else {
            state = SOLAR;
        }
    }
    else {
        state = SOLAR;
    }
}

```

```

    } // end bat_percent
    else { // there's more power than consumed
        state = SOLAR;
    } // end load > solar
} // end bat_percent
/*****
*****/
/*  rutinas para cada estado  */
switch (state) {
    case SOLAR:
        if (use_solar == false) { // chequea si el inversor esta acoplado
            digitalWrite(N_RELAY, HIGH); // NC
            digitalWrite(GRID_SSR, LOW);
            delay(5);
            digitalWrite(SOLAR_SSR, HIGH);
            use_solar = true; // indica que ya se acoplo al inversor
            use_grid = false; // la red esta desacoplada
            grid_flag = 0;

        } // end use_solar
        break;

    case GRID:
        if (use_grid == false) {
            digitalWrite(N_RELAY, LOW);
            digitalWrite(SOLAR_SSR, LOW);
            delay(5);
            digitalWrite(GRID_SSR, HIGH);
            use_grid = true; // indica que ya se acoplo al inversor
            use_solar = false; // la red esta desacoplada
            grid_flag = 1 ;
        }
        break;
    }
}
/*****
*****/

/* Rutina para modo Red */
else { // modo red
    if (grid_status == 0) {
        auto_mode = true;
    }
    else {
        if (use_grid == false) {
            digitalWrite(N_RELAY, LOW);
            digitalWrite(SOLAR_SSR, LOW);
            delay(5);
            digitalWrite(GRID_SSR, HIGH);
            use_grid = true; // indica que ya se acoplo al inversor
            use_solar = false; // la red esta desacoplada

```

```

        grid_flag = 1;
    }
}
}
}
}
/*****
****/
/* Envio de datos al HMI */
Serial.print(solar_power);    Serial.print(',');
Serial.print(grid_power);     Serial.print(',');
Serial.print(load_power);     Serial.print(',');
Serial.print(bat_percent);    Serial.print(',');
Serial.print(grid_flag);     Serial.print(',');
Serial.print(grid_status);    Serial.println();

} //Fin de programa

```

ANEXO B: Instrucciones de configuración del Raspberry Pi para el sistema de monitoreo

Initial Linux Setup

Update

```
sudo apt-get update -y
sudo apt-get upgrade -y
```

Change password, set international options & expand File-system

Using raspi-config utility:

```
sudo raspi-config
```

Raspbian minimal image is 2GB by default, expand to fill 4GB. Then reboot

Change hostname

From 'raspberry' to 'emonpi'

```
sudo nano /etc/hosts
sudo nano /etc/hostname
```

Change Password

From 'raspberrypi' to 'emonpi2016'

```
sudo passwd
```

Root filesystem read-only with RW data partition

Custom MOTD (message of the day)

Use custom motd to alert users they are logging into an emonPi with RW / RO toggle instructions:

```
sudo rm /etc/motd
sudo ln -s /home/pi/emonpi/motd /etc/motd
```

Memory Tweak

Append `gpu_mem=16` to `/boot/config.txt` this caps the RAM available to the GPU. Since we are running headless this will give us more RAM at the expense of the GPU

RasPi Serial port setup

To allow the emonPi to communicate with the RasPi via serial we need to disconnect the terminal console from `/tty/AMA0`.

Use custom `cmdline.txt`

```
sudo mv /boot/cmdline.txt /boot/original.cmdline.txt
sudo cp /home/pi/emonpi/cmdline.txt /boot/
```

This changes the a custom `cmdline.txt` file:

From:

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes
rootwait
```

To:

```
dwc_otg.lpm_enable=0 console=tty1 elevator=noop root=/dev/mmcblk0p2
rootfstype=ext4 fsck.repair=yes rootwait
```

Note changing `elevator=deadline` to `elevator=noop` disk scheduler. Noop that is best recommend for flash disks, this will result in a reduction in disk I/O performance

"Noop: Inserts all the incoming I/O requests to a First In First Out queue and implements request merging. Best used with storage devices that does not depend on mechanical movement to access data (yes, like our flash drives). Advantage here is that flash drives does not require reordering of multiple I/O requests unlike in normal hard drives" - [Full article](#). [Forum topic discussion](#).

Disable serial console boot

```
sudo systemctl stop serial-getty@ttyAMA0.service
sudo systemctl disable serial-getty@ttyAMA0.service
```

Enable serial uploads with avrdude and autoreset

```
git clone https://github.com/openenergymonitor/avrdude-rpi.git ~/avrdude-
rpi && ~/avrdude-rpi/install
```

Raspberry Pi 3 Compatibility

The emonPi communicates with the RasPi via GPIO 14/15 which on the Model B,B+ and Pi2 is mapped to UART0. However on the Pi3 these pins are mapped to UART1 since UART0 is now used for the bluetooth module. However UART1 is software UART and baud rate is dependant to clock speed which can change with the CPU load, undervoltage and temperature; therefore not stable enough. One hack is to force the CPU to a lower speed (add `core_freq=250` to `/boot/cmdline.txt`) which cripples the Pi3 performance. A better solution for the emonPi is to swap BT and map UART1 back to UART0 (ttyAMA0) so we can talk to the emomPi in the same way as before.

RasPi 3 by default has bluetooth (BT) mapped to `/dev/AMA0`. To allow use to use high performace hardware serial (`dev/ttyAMA0`) and move bluetooth to software serial (`/dev/ttyS0`) we need to add an overlay to `config.txt`.

See `/boot/overlays/README` for more details. See [this post for RasPi3 serial explained](#).

```
sudo nano /boot/config.txt
```

Add to the end of the file

```
dtoverlay=pi3-miniuart-bt
```

We also need to run to stop BT modem trying to use UART

```
sudo systemctl disable hciuart
```

See [RasPi device tree commit](#) for `pi3-disable-bt` and [forum thread discussion](#)

Reboot then test serial comms with:

```
sudo minicom -D /dev/ttyAMA0 -b38400
```

You should see data from emonPi ATmega328, sending serial `v` should result in emonPi returning it's firmware version and config settings.

To fix SSHD bug (when using the on board WiFi adapter and NO Ethernet). [Forum thread](#). Edit `/etc/ssh/sshd_config` and append:

```
IPQoS cs0 c
```

Install emonPi Services

```
sudo apt-get install git-core -y
cd /home/pi
git clone https://github.com/openenergymonitor/emonpi
git clone https://github.com/openenergymonitor/RFM2Pi
```

emonPi LCD service

Enable I2C module, install packages needed for emonPi LCD service `python-smbus` `i2c-tools` `python-rpi.gpio` `python-pip` `redis-server` and `pip install uptime redis paho-mqtt`, and install emonPi LCD python scrip service to run at boot

```
sudo emonpi/lcd/./install
```

Restart and test if I2C LCD is detected, it should be on address `0x27`:

```
sudo i2cdetect -y 1
```

Setup emonPi update

Add Pi user cron entry:

```
crontab -e
```

Add the cron entries to check if emonpi update or emonpi backup has been triggered once every 60s:

```
...
```

```
MAILTO=""
```

```
# # Run emonPi update script ever min, scrip exits unless update flag  
exists in /tmp  
* * * * * /home/pi/emonpi/service-runner >> /var/log/service-runner.log  
2>&1  
...
```

To enable triggering update on first factory boot (when emonpiupdate.log does not exist) add entry to ``rc.local``:

```
/home/pi/emonpi/./firstbootupdate
```

This line should be present already if the emonPi speicifc ``rc.local`` file has been symlinked into place, if not:

```
sudo rm /etc/rc.local  
sudo ln -s /home/pi/emonpi/emonpi/rc.local_jessieminimal  
/etc/rc.local
```

The ``update`` script looks for a flag in ``/tmp/emonpiupdate`` which is set when use clicks Update in Emoncms. If flag is present then the update script runs ``emonpiupdate``, ``emoncmsupdate`` and ``emonhubupdate`` and logs to ``~/data/emonpiupdate.log``. If log file is not present then update is ran on first (factory) boot.

Setup NTP Update

See [forum thread](#)

```
sudo crontab -e
```

Add

```
# Force NTP update every 1 hour
```

```
0 * * * * /home/pi/emonpi/ntp_update.sh >> /var/log/ntp_update.log 2>&1
```

Fix Random seed

Random seed is important for secure ssh and https connections. Using a RO root FS we need to symlink randomised to RW data partition to enable to service to run and be random. See [forum thread](#).

```
$ rpi-rw
$ cd /var/lib/systemd/
$ sudo mv random-seed ~/data/
$ sudo ln -s ~/data/random-seed
$ rpi-ro
$ sudo systemctl restart systemd-random-seed
$ sudo systemctl status systemd-random-seed
```

Random seed service should now be running.

Install mosquitto MQTT

Install & configure Mosquitto MQTT server, PHP MQTT

```
wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
sudo apt-key add mosquitto-repo.gpg.key
cd /etc/apt/sources.list.d/
sudo wget http://repo.mosquitto.org/debian/mosquitto-jessie.list
sudo apt-get update
sudo apt-get install mosquitto mosquitto-clients libmosquitto-dev -y
sudo pecl install Mosquitto-beta
(Hit enter to autodetect libmosquitto location)
```

To check which version of Mosquitto pecl has installed run `$ pecl list-upgrades`. Currently emonSD (Oct17) is running Mosquitto 0.3. See this [forum post](#) and [this one](#) discussing the choice of Mosquitto alpa.

```
pecl list-upgrades
Channel pear.php.net: No upgrades available
Channel pear.swiftmailer.org: No upgrades available
pecl.php.net Available Upgrades (stable):
=====
Channel      Package  Local          Remote          Size
pecl.php.net dio      0.0.6 (beta)   0.1.0 (beta)   37kB
pecl.php.net Mosquitto 0.3.0 (beta)   0.4.0 (beta)   24kB
pecl.php.net redis    2.2.5 (stable) 3.1.6 (stable) 196kB
```

To upgrade to Mosquitto 0.4 run (currently not fully tested as of Jan 18): *Update: seems to work fine*

```
$ sudo pecl install Mosquitto-0.4.0
```

Install PHP Mosquitto extension:

```
printf "extension=mosquitto.so" | sudo tee /etc/php5/mods-
available/mosquitto.ini 1>&2
sudo php5enmod mosquitto
```

Turn off Mosquitto persistence and enable authentication:

```
sudo nano /etc/mosquitto/mosquitto.conf
```

Set persistence false and add the lines:

```
allow_anonymous false
```

```
password_file /etc/mosquitto/passwd
```

Create a password file for MQTT user emonpi with:

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd emonpi
```

Enter password emonpimqtt2016 (default)

Test MQTT

Open *another shell window* to subscribe to a test topic:

```
mosquitto_sub -v -u 'emonpi' -P 'emonpimqtt2016' -t 'test/topic'
```

In the first shell publish to the test topic :

```
mosquitto_pub -u 'emonpi' -P 'emonpimqtt2016' -t 'test/topic' -m  
'helloWorld'
```

If all is working we should see helloWorld in the second shell

emonHub

Install other EmonHub dependencies if they have not been installed already by emonPi LCD service

```
sudo apt-get install -y python-pip python-serial python-configobj  
sudo pip install paho-mqtt pydispatcher
```

Install emonHub (emon-pi) variant:

```
cd ~/  
git clone https://github.com/openenergymonitor/emonhub.git &&  
emonhub/install
```

Check log file:

```
tail /var/log/emonhub/emonhub.log
```

We can check data is being posted to MQTT by subscribing to the base topic emon/# if new node variable MQTT topic structure is used or emonhub/# if old MQTT CSV topic structure is used

```
mosquitto_sub -v -u 'emonpi' -P 'emonpimqtt2016' -t 'emon/#'
```

Install Emoncms Core

V9 Core (stable branch)

Follow [Emoncms Raspberry Pi Raspbian Jessie install guide](#)

emonPi specific Emoncms settings

- MYSQL root password: emonpimysql2016
- MYSQL emoncms user password: emonpiemoncmsmysql2016
- Use emonPi default settings:
 - `cd /var/www/emoncms && cp default.emonpi.settings.php settings.php`
- Create feed directories in RW partition /home/pi/data instead of /var/lib:
 - `sudo mkdir /home/pi/data/{phpfina,phptimeseries}`
 - `sudo chown www-data:root /home/pi/data/{phpfina,phptimeseries}`

Note: at time of writing the version of php5-redis included in the Raspbian Jessie sources (2.2.5-1) caused Apache to crash (segmentation errors in Apache error log). Installing the latest stable version (2.2.7) of php5-redis from github fixed the issue. This step probably won't be required in the future when the updated version of php5-redis makes it's way into the sources. The check the version in the sources: `sudo apt-cache show php5-redis`

```
git clone --branch 2.2.7 https://github.com/phpredis/phpredis
cd phpredis
(check the version we are about to install:)
cat php_redis.h | grep VERSION
phpize
./configure
sudo make
sudo make install
```

Move MYSQL database location

The MYSQL database for Emoncms is usually in /var/lib, since the emonPi runs the root FS as read-only we need to move the MYSQL database to the RW partition home/pi/data/mysql. Follow stepsL [Moving MYSQL database in Emoncms Read-only documentation](#)

Low write mode Emoncms optimisations

[Related forum discussion thread](#)

Follow [Raspberry Pi Emoncms Low-Write guide](#) from [Setting up logging on read-only FS](#) onwards (we have earlier setup read-only mode):

- [Setting up logging on read-only FS](#)
 - After running install we want to use emonPi specific rc.local instead:
 - `sudo rm /etc/rc.local`
 - `sudo ln -s /home/pi/emonpi/rc.local_jessieminimal /etc/rc.local`
- [Move PHP sessions to tmpfs \(RAM\)](#)

- Configure Redis
 - After redis config **Ensure all redis databases have been removed from /var/lib/redis**
- Disable apache access log:
 - `sudo nano /etc/apache2/conf-enabled/other-vhosts-access-log.conf`
 - comment out the access log
- Ensure apache error log goes to /var/log/apache2/error.log
 - `sudo nano /etc/apache2/envvars`
 - Ensure `export APACHE_LOG_DIR=/var/log/apache2/$SUFFIX`
- Reduce garbage in syslog due to Raspbian bug:
 - Edit the /etc/rsyslog.conf and comment out the last 4 lines that use xconsole:

```
#daemon.*;mail.*;\n\n# news.err;\n\n# *.=debug;*.=info;\n\n# *.=notice;*.=warn | /dev/xconsole
```

No need to Enable Low-write mode in emoncms since these changes to settings.php are already set in default.emonpi.settings.php that we installed earlier.

Install Emoncms Modules

```
cd /var/www/emoncms/Modules\n git clone https://github.com/emoncms/dashboard.git\n git clone https://github.com/emoncms/app.git\n git clone https://github.com/emoncms/wifi.git\n git clone https://github.com/emoncms/config\n git clone https://github.com/emoncms/graph\n\ncd /home/pi/\n git clone https://github.com/emoncms/backup\n git clone https://github.com/emoncms/postprocess\n git clone https://github.com/emoncms/usefulscripts
```

- After installing modules check and apply database updates in Emoncms Admin.

Configure Emoncms WiFi Module

Follow install instructions in [WiFi module Readme](#) to give web user permission to execute system WLAN commands.

Move `wpa_supplicant.conf` (file where WiFi network authentication details are stored) to RW partition with symlink back to `/etc`:

```
sudo mv /etc/wpa_supplicant/wpa_supplicant.conf
/home/pi/data/wpa_supplicant.conf
sudo ln -s /home/pi/data/wpa_supplicant.conf
/etc/wpa_supplicant/wpa_supplicant.conf
```

Install wifi-check script

To check if wifi is connected every 5min and re-initiate connection if not.

Add wifi-check script to `/usr/local/bin`:

```
sudo ln -s /home/pi/emonpi/wifi-check /usr/local/bin/wifi-check
```

Make wifi check script run as root crontab every 5min:

```
sudo crontab -e
```

Add the line:

```
*/5 * * * * /usr/local/bin/wifi-check > /var/log/wifichk.log 2>&1"
mycron ;
```

Setup Emoncms Backup & import module

Follow [instructions on emonPi backup module page](#) to symlink the Emoncms module from the folder in `/home/pi/backup` that also contains the backup shell scripts

Emoncms Language Support

```
sudo apt-get install gettext
sudo dpkg-reconfigure locales
```

Select required languages from the list by pressing [Space], hit [Enter to install], see [language translations supported by Emoncms](#)

```
sudo apache2 restart
```

[more info on gettext](#)

Install Emoncms MQTT input service

To enable data posted to base topic `emon/` topic to appear in Emoncms inputs e.g `emon/emontx/power1 10` creates an input from `emonTx` node with `power1` with value 10.

[Follow install guide](#)

```
sudo service mqtt_input start
```

[Forum Thread discussing emonhub MQTT support](#)

Previously EmonHub posted decoded data to MQTT topic with CSV format to emonhub/rx/5/values 10,11,12 which the Nodes module subscribed to and decoded. Using the MQTT input service in conjunction with the new EmonHub MQTT node topic structure depreciates the Nodes module

Lightwave RF MQTT service

[Follow LightWave RF MQTT GitHub Repo Install Guide](#)

Configure Firewall

Apache web server `sudo ufw allow 80/tcp` and `sudo ufw allow 443/tcp`

SSH server: `sudo ufw allow 22/tcp`

Mosquitto MQTT: `sudo ufw allow 1883/tcp`

Redis: `sudo ufw allow 6379/tcp`

OpenHAB: `sudo ufw allow 8080/tcp`

NodeRed: `sudo ufw allow 1880/tcp`

```
sudo ufw enable
```

ANEXO C: Instrucciones de Instalación de NodeRED en LINUX

Install

<http://nodered.org/docs/hardware/raspberrypi.html>

At time of writing (Assuming running Raspbian Jessie):

```
sudo apt-get install nodered
```

Start node Red at boot:

```
sudo systemctl enable nodered.service
```

View NodeRED log

In var/syslog filter with

```
$ sudo journalctl -f -u nodered -o cat
```

Move user data partition to RW ~/data partition on emonPi

```
mv /home/pi/.node-red ~/data/node-red
```

Test with:

```
node-red-pi --max-old-space-size=128 --userDir /home/pi/data/node-red
```

Create symlink to /home/pi/data/node-red from standard location /home/pi/node-red

```
sudo ln -s /home/pi/data/node-red /home/pi/.node-red
```

Note: creating symlink to ~/data from default data folder location is better then setting custom data folder location in nodered service since symlink method will survive a node-RED update cycle.

Save an exit nano then run:

```
sudo systemctl daemon-reload
sudo service nodered restart
```

Install emoncms flow

```
$ sudo apt-get install npm
$ cd ~/data/node-red
$ sudo npm install node-red-node-emoncms
$ sudo npm install node-red-node-weather-undergroun
```

ANEXO D: Hoja de datos del sensor de corriente



Product Specification

Date:2015-8-7

Product Name	Current transformer	Model	SCT013-030
--------------	---------------------	-------	------------

Characteristics: Opening size 13mm*13mm, 1m leading wire, standard $\Phi 3.5$ three-core plug output.
voltage output type.

Purpose: Used for current measurement, monitor and protection for AC motor, lighting equipment, air compressor etc.
Patent No.: ZL 2015 3 0060067. X

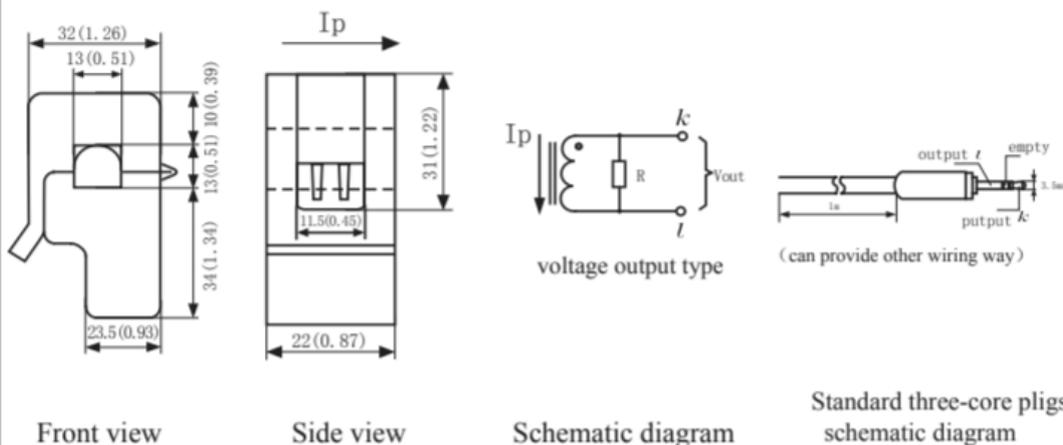
Technical Data

I_{PN}	Rated input	0-30A
I_{PM}	Max. detection input	
I_{OUT}	Rated output	0-1V
X	Accuracy	$\pm 1\%$
ϵ_L	Linearity	$\leq 0.2\%$
N	Turns ratio	1:1800
Φ	Phase shift	
R_L	Max. Sampling resistance	
V_{PN}	Work voltage	660V
f	Work frequency	50-1KHz
T_A	Operating temperature	-25..+70 $^{\circ}$ C
T_S	Storage temperature	-40..+85 $^{\circ}$ C
Vd	Dielectric strength, 50 Hz, 1 min	3KV



Fire resistance	UL94-V0
Material of core	Ferrite
Mounting type	Suspension
Weight	55g

Dimension (mm(in). 1 mm= 0.0394 inch)



<http://www.yhdc.com>

ANEXO E: Protocolo de Comunicación del controlador de carga EPSolar Tracer 1210rn

alexruhmenn@body-soft.de

PROTOCOL OF TRACER SEREIS AND MT-5 Ver 3
(REVISED ON 13, Dec.2011.)

1, instruction on communication date format.

PWL start up 6 byte, (AAH, 55H, AAH, 55H, AAH, 55H)

+ communication synchronization, the first 6 byte (EBH, 90H, EBH, 90H, EBH, 90H)

+ controller ID(1 byte)

+ command (1 byte) (see details in part3: instructions on command)

+ data packet length (1 byte, bytes of data content , maximumly don't exceeds 200 bytes)

+ data content (low byte in front)

+ CRC Cyclic redundancy check code (2 byte, high byte in front)

+7FH

e.g.: computer send a real time monitoring command string to E01 series controller(ID=16H) :

AA 55 AA 55 AA 55 EB 90 EB 90 EB 90 16 A0 00 B1 A7 7F

For example , responds is as following:

AA 55 AA 55 AA 55 EB 90 EB 90 EB 90 _____ 12 bytes to start the synchronization head

16 A0 18		ID, command, data
length		
CE 04 A4 06 00 00 E8 03 56 04 A0 05 01 00 00 50 00 00 00 00 37 E8 03 00		
17 datas		
9A 38 7F		2 byte check code, 1 byte exit
code		

二、 Pre description as following:

1、 Controller cannot identify PWL start up 6 byte AA 55 AA 55 AA 55, in order to save , some device won't send , so the host computer is not sensible to such 6 byte AA 55

AA 55 AA 55.

- 2、 6 byte PWL start up + 6 byte communication synchronization is 12 byte, abbreviated as "synchronous head"
- 3、 Floating point data such as voltage, current , etc, are indicates as 100 times integer of the actual value, e.g., 12.5V, showed as 1250 (0x04E2 (hexadecimal 2-byte integer). Please switch by yourself.
- 4、 Order of integers of more than 2 byte are transferred in sequence : Low byte in front and high byte in behind.
- 5、 ID is 1 byte data, the present version won't deal with it now.
- 6、 The following commands added shall follow the above principals.

三、 Command byte definitions are as following:

1、 Command A0H(read the real time collected data) : used for real time monitoring

Special delivery format and respond format:

Host send: synchronous head + ID + A0H + 00H + CRCH + CRCL + 7FH

Sub device send back: synchronous head + ID + A0H + 0DH + collected data strings + CRCH + CRCL + 7FH

The data strings content and sequence to be collected:

No.	items	bytes	instructions
1	Battery voltage	2 byte	e.g., 12.5V(1250 showed as <u>0x04E2</u> , low byte is in front when sending.
2	Pv voltage	2 byte	same as above
3	Reserved	2 byte	"0" all time.
4	Load current	2 byte	e.g. 10.0A(1000 showed as <u>0x03E8</u> , low byte is in front when sending.
5	over discharge voltage	2 byte	same as above
6	Battery full voltage	2 byte	same as above
7	Load on/off	1 byte	=0, load off; =1, load on
8	Over load	1	=0, normal ; =1, load current over load, over load

		byte	protection would happen.
9	Load short circuit	1 byte	=0, normal; =1, load short circuit, output would be stopped.
10	reserved	1 byte	Not used
11	Battery overload	1 byte	=0, normal; =1, battery voltage over voltage, controller would stop charging and discharging.
12	Over discharge	1 byte	=0, normal, =1, battery voltage too low and output would be stopped.
13	Full indicator	1 byte	=0, not fully charged; =1, battery voltage is full
14	Charging indicator	1 byte	=0, not charging; =1, being charged.
15	Battery temp.	1 byte	E.g, 25°C(55 showed as 0x37, there is 30 difference in value with the real value.
16	Charging current	2 byte	e.g. 10.0A(1000 showed as <u>0x03E8</u> 。 Low byte in front when sending.
17	reserved	1 byte	0

2. Command 0xAA: (manual control command):

used for controlling of load switch through remote meter or computer

Host send: synchronous head + ID + AAH + 01H + control switch command + CRCH + CRCL + 7FH

Sub device send back: synchronous head + ID + AAH + 01H + load switch state + CRCH + CRCL + 7FH

Control switch command:

=1, Load ON

=0, Load OFF

Load switch state:

=1, Load ON

=0, Load OFF

3. Command 0xAD: remote meter or computer send control data:

Format:

Host send: synchronous head + ID + ADH + 00H + control parameter data strings + CRCH + CRCL + 7FH

Sub device send back: synchronous head + ID + ADH + 00H + CRCH + CRCL + 7FH

Control parameter data strings:

No.	Items	bytes	instructions
1	Load type	1byte	unuse

2	Charging mode	1byte	unuse
3	Temperature compensation coefficient	1byte	0~10
4	Battery type	1byte	0: gel, 1: sealed, 2: flooded
5	Battery capacity	1byte	Show 1/10 of normal battery capacity (AH), need recover process
6	Load control mode 1	1byte	As below chart
7	Load control mode 2	1byte	As below chart

Load control mode 1 code

code	instruction
0	Dusk to Dawn
1~15	1~15 hours timer
16	Test
17	Manual
18	unable

Load mode 2 code only include 1~15、 18、

CRC process example:

CRC_Buff –

The first address of data strings to be proceed

crc_len -- Byte numbers to be processed, from ID byte to final byte of CRC check

result.

```
u16 Crc(u8 *CRC_Buff,u8 crc_len)
```

```
{
```

```
    u8 crc_i,crc_j,r1,r2,r3,r4;
```

```
    u16 crc_result;
```

```
    r1=*CRC_Buff;
```

```
    CRC_Buff++;
```

```
    r2=*CRC_Buff;
```

```
    CRC_Buff++;
```

```
    for (crc_i=0;crc_i<crc_len-2;crc_i++)
```

```
    {
```

```
        r3=*CRC_Buff;
```

```
        CRC_Buff++;
```

```

    for (crc_j=0;crc_j<8;crc_j++)
    {
        r4=r1;
        r1=(r1<<1);
        if ((r2&0x80)!=0)r1++;
        r2=r2<<1;
        if((r3&0x80)!=0)r2++;
        r3=(r3<<1);
        if ((r4&0x80)!=0)
        {
            r1=r1^0x10;
            r2=r2^0x41;
        }
    }
}
crc_result=r1;
crc_result=crc_result<<8 | r2;
return(crc_result);
}

```

example.:

```

Main()
{
    ...
    //
    calculate the CRC checkword before sending
    t_buf[t_buf[2]+3] = 0;    // remove all command.
    t_buf[t_buf[2]+4] = 0;
    CRC_Result=Crc(t_buf,t_buf[2]+5); // calculate
    t_buf[t_buf[2]+3] = CRC_Result>>8;
    t_buf[t_buf[2]+4] = CRC_Result & 0xFF;
    t_buf[t_buf[2]+5] = 0x7F;    // exit code
    ...
    ...
}

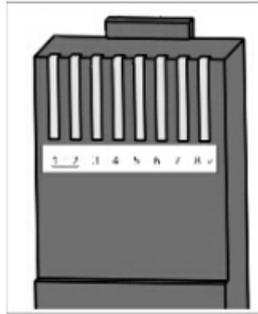
```

Connector Pin definitions:

Pin number	definition
1	Power supply + 12V output
2	Power ground/signal ground
3	Power +12V output

4	Power ground/signal ground
5	TXD, the sender
6	RXD, the receiving end
7	Power ground/signal output
8	Power ground/signal output

RJ45crystal plus pin number as shown





**Presidencia
de la República
del Ecuador**



**Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes**



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Peñafiel Caicedo, Joseph Adrián**, con C.C: # **092550983-8** autor/a del trabajo de titulación: **Estudio de un sistema de acoplamiento híbrido inteligente para energía solar fotovoltaica y la red eléctrica de una vivienda**, previo a la obtención del título de **Ingeniero Electrónico en Control y Automatismo** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **12 de marzo de 2018**

f. _____

Nombre: **Peñafiel Caicedo, Joseph Adrián**

C.C: **0925509838**



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Estudio de un sistema de acoplamiento híbrido inteligente para energía solar fotovoltaica y la red eléctrica de una vivienda		
AUTOR	Peñañiel Caicedo, Joseph Adrián		
REVISOR(ES)/TUTOR(ES)	Ing. Mendoza Merchán, Eduardo Vicente, Mgs.		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Carrera de Ingeniería Electrónica en Control y Automatismo		
TÍTULO OBTENIDO:	Ingeniero Electrónico en Control y Automatismo		
FECHA DE PUBLICACIÓN:	12 de marzo de 2018	No. DE PÁGINAS:	108
ÁREAS TEMÁTICAS:	Energía renovable, domótica, electrónica de potencia		
PALABRAS CLAVES/ KEYWORDS:	Sistemas fotovoltaicos, microcontroladores, red eléctrica, acoplamiento, controlador de carga, inversor de potencia, acumuladores		
RESUMEN/ABSTRACT (150-250 palabras):	<p>El trabajo realizado en esta investigación se orientó al diseño de un sistema de acoplamiento automatizado de cargas eléctricas a un sistema fotovoltaico y la red eléctrica. Se realizó un prototipo que demuestre el funcionamiento del sistema utilizando microcontroladores para la lectura de los sensores de las variables del sistema y el control del proceso que conlleva a la selección de la fuente a la que las cargas se deben acoplar. Junto a esto se desarrolló un sistema de visualización remota utilizando un ordenador de placa reducida con sistema operativo Linux. Utilizando metodologías de investigación de tipo experimentales y bibliográficas se pudo desarrollar dicho sistema de acoplamiento automatizado. El resultado de la investigación es un sistema que monitorea la potencia de producción solar y la potencia de consumo para poder decidir el acoplamiento de las cargas al sistema fotovoltaico o la red eléctrica, además de una interfaz remota de control y visualización del sistema, mediante una conexión de red LAN. Al finalizar la lectura de esta investigación, se puede llegar a la conclusión que el uso de microcontroladores es de gran ayuda para el desarrollo de cualquier tipo de sistemas que requieran control automatizado.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593996433367	E-mail: josephpenafiel@outlook.com	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::	Nombre: Ing. Mendoza Merchán, Eduardo Vicente, Mgs.		
	Teléfono: +593985086815		
	E-mail: eduardo.mendoza01@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			