



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y  
COMPUTACIÓN**

**“DISEÑO E IMPLEMENTACIÓN DE FUENTES DE ENERGÍA SOLAR PARA  
RASPBERRY PI PROGRAMADO PARA LA DETECCIÓN DE EVENTOS EN  
MODO DE BAJO CONSUMO DE ENERGÍA”**

**TESINA DE SEMINARIO**

**Previa la obtención del Título de:**

**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**Presentado por:**

**Jonathan Alejandro Cisneros Romo**

**Jorge Mario Peralta Verdezoto**

**GUAYAQUIL – ECUADOR**

**AÑO 2015**

## AGRADECIMIENTO

A Dios por guiarme y acompañarme a lo largo de mi carrera por ser mi fortaleza en los momentos de debilidad porque siempre está en mi camino y me muestra esa luz de esperanza y me dota siempre de inteligencia.

*Jonathan Alejandro Cisneros Romo*

A Dios por darme las fuerzas necesarias para culminar con éxito esta etapa de mi vida.

*Jorge Mario Peralta Verdezoto*

## DEDICATORIA

Le dedico este trabajo a mi hermana Estefanía una mujer luchadora, que me ha enseñado a tener Fe en Dios en todo momento.

A mi hermano Jonathan al cual admiro mucho y extraño, que este pequeño logro sea una inspiración en su vida y así nunca deje de luchar por sus sueños.

A mi sobrino Sebastián que este trabajo sea un incentivo para que coseche inmejorables éxitos a lo largo de su vida.

*Jorge Mario Peralta Verdezoto*

## DEDICATORIA

A Dios por guiarme por el buen camino y por darme la fuerza para poder seguir adelante sin perder y permitirme llegar a este momento tan grande de mi vida.

A mis padres por ser el pilar de mi vida y formarme con los mejores valores, sentimientos y hábitos, lo cual me ha ayudado a salir adelante en los momentos difíciles, mis hermanos por estar en cada paso que doy y a una persona que a la distancia me ha dado ánimo y ayuda en esta meta.

*Jonathan Alejandro Cisneros Romo*

## **TRIBUNAL DE SUSTENTACIÓN**

---

Ing. Carlos Valdivieso A.

**PROFESOR DEL SEMINARIO DE GRADUACIÓN**

---

Ing. Hugo Villavicencio

**PROFESOR DELEGADO DE LA UNIDAD ACADÉMICA**

## **DECLARACIÓN EXPRESA**

“La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”.

(Reglamento de Graduación de la ESPOL)

---

**Jonathan Alejandro Cisneros Romo**

---

**Jorge Mario Peralta Verdezoto**

## RESUMEN

El presente trabajo abarca la investigación del uso de la minicomputadora Raspberry Pi tanto de su hardware como su software y del Sistema Fotovoltaico SFV como fuente de energía, la interconexión que se realizara entre la minicomputadora Raspberry Pi y el microcontrolador Arduino UNO para que así se pueda consumir la menor cantidad de energía almacenada en el bloque de batería.

Las generalidades de la parte principal de este proyecto, es decir el conocimiento del uso de la Raspberry Pi, del Arduino UNO y los Sistemas Fotovoltaicos SFV se ha recopilado en los capítulos 1 y 2. Donde en el capítulo 1 se presentan los antecedentes, objetivos y las limitaciones que se tienen al realizar el proyecto, capítulo 2 se presentan el fundamento teórico de las partes principales que caracterizan a la Raspberry Pi, Arduino UNO y a los Sistemas Fotovoltaicos.

La parte final la conforman los códigos de la programación del proyecto que se encuentran en el capítulo 3 y las pruebas y simulaciones de los mismos en el

capítulo 4. Por último se dan a conocer las conclusiones y recomendaciones del proyecto en general.



## ÍNDICE GENERAL

RESUMEN.....	VII
ÍNDICE GENERAL .....	IX
ÍNDICE DE TABLAS.....	XIV
ÍNDICE DE FIGURAS.....	XV
INTRODUCCIÓN.....	XVIII
CAPÍTULO 1.....	1
DESCRIPCIÓN GENERAL DEL PROYECTO .....	1
1.1 ANTECEDENTES .....	1
1.2 OBJETIVOS .....	7
1.2.1 OBJETIVOS GENERALES.....	7
1.2.2 OBJETIVOS ESPECÍFICOS .....	8
1.3 IDENTIFICACIÓN DEL PROBLEMA .....	8
1.4 LIMITACIONES.....	10
CAPÍTULO 2.....	12
FUNDAMENTO TEÓRICO .....	12
2.1 RESUMEN .....	12
2.2 MINICOMPUTADORA RASPBERRY PI MODELO B.....	13

2.2.1 CARACTERÍSTICAS.....	15
2.2.2 PROCESADOR.....	17
2.2.3 DISPOSITIVO DE ARRANQUE .....	20
2.2.4 SISTEMAS OPERATIVOS SOPORTADOS.....	22
2.2.5 PUERTO GPIO .....	25
2.2.5.1 CARACTERÍSTICAS GENERALES DE LOS GPIO.....	26
2.2.5.2 GPIO DE LA RPI .....	27
2.2.5.3 DIFERENCIAS OUTS DEL GPIO (BCM Y BOARD).....	31
2.3 ARDUINO UNO.....	32
2.3.1 DIFERENTES PRODUCTOS DE ARDUINO.....	34
2.3.2 ENTRADAS Y SALIDAS .....	36
2.4 INTRODUCCION A LAS FUENTES DE ENERGÍA RENOVABLES .....	39
2.5 ENERGÍA SOLAR .....	44
2.6 DISTRIBUCION ESPECTRAL DE LA RADIACIÓN SOLAR.....	46
2.7 ENERGIA SOLAR FOTOVOLTÁICA.....	49
2.8 EFECTO FOTOELÉCTRICO.....	50
2.9 CELDAS FOTOVOLTÁICAS .....	52
2.10 TIPOS DE CELDAS FOTOVOLTÁICAS.....	53
2.10.1 MATERIALES DENSAMENTE CRISTALINOS .....	53
2.10.2 MATERIALES DE PELICULA DELGADA .....	55

2.11 EFICIENCIA DE CONVERSIÓN .....	56
2.12 CURVAS CARACTERÍSTICAS DE CELDAS FOTOVOLTÁICAS .....	57
2.12.1 CURVA DE CORRIENTE VS. TENSIÓN.....	57
2.13 FACTORES AMBIENTALES SOBRE LAS CARACTERÍSTICAS DE SALIDA DEL DISPOSITIVO.....	61
2.13.1 EFECTO DE LA INTENSIDAD DE LA RADIACION SOLAR...	61
2.13.2 EFECTO DE LA TEMPERATURA.....	61
2.14 INTERACCIÓN DEL DISPOSITIVO FOTOVOLTÁICO CON LA CARGA .....	63
2.15 CONSTITUCIÓN DE UN MÓDULO FOTOVOLTÁICO .....	65
2.16 SISTEMA FOTOVOLTÁICO.....	66
2.16.1 PRINCIPALES VENTAJAS.....	68
2.16.2 PRINCIPALES APLICACIONES.....	68
2.16.3 REGULADOR DE CARGA .....	69
2.16.4 ACUMULADORES O BATERÍAS .....	70
2.16.4.1 BATERÍAS PLOMO-ÁCIDO .....	72
2.16.4.2 NÍQUEL-CADMIO .....	73
2.16.4.3 SELECCIÓN DE TIPO DE BATERÍA.....	74
2.16.5 INVERSORES Y CONVERTIDORES.....	74
2.17 UNIDADES UTILIZADAS EN ENERGIA SOLAR FOTOVOLTÁICA ...	75

CAPÍTULO 3.....	77
DESARROLLO DEL PROYECTO .....	77
3.1 RESUMEN .....	77
3.2 EJERCICIO 1: CÓMO TRABAJAR CON LOS PUERTOS DEL ARDUINO .....	78
DIAGRAMA DE FLUJO .....	79
ALGORITMO.....	80
CÓDIGO FUENTE .....	80
CONCLUSIÓN .....	81
3.3 EJERCICIO 2: CÓMO IMPRIMIR MENSAJES EN UNA LCD.....	81
DIAGRAMA DE FLUJO .....	82
ALGORITMO.....	83
CÓDIGO FUENTE .....	83
CONCLUSIÓN .....	84
3.4 PROYECTO TERMINADO .....	84
DIAGRAMA DE BLOQUE .....	85
DIAGRAMA DE FLUJO .....	86
CÓDIGO FUENTE .....	87
ALGORITMO.....	110

CAPÍTULO 4.....	112
SIMULACIÓN, PRUEBAS Y EVALUACIÓN .....	112
4.1 RESUMEN .....	112
4.2 SIMULACIÓN DEL CIRCUITO CONMUTADOR EN PROTEUS .....	113
4.2.1 IMPLEMENTACIÓN DEL CIRCUITO CONMUTADOR EN PROTOBOARD.....	115
4.3 SIMULACIÓN DE RUTINA EN RASPBERRY PI.....	116
4.4 DISEÑO DEL SISTEMA FOTOVOLTÁICO .....	119
4.5 SIMULACIÓN DEL PROYECTO .....	129
CONCLUSIONES.....	132
RECOMENDACIONES.....	135
BIBLIOGRAFÍA.....	137
ANEXOS .....	143

## ÍNDICE DE TABLAS

Tabla 2.1 Unidades utilizadas en la energía solar fotovoltaica .....	76
Tabla 4.1 Valores de irradiación para la ciudad de Guayaquil .....	120
Tabla 4.2 Consumo de potencia .....	121
Tabla 4.3 Consumo de energía Wh/día.....	122

## ÍNDICE DE FIGURAS

Figura 1.1 Raspberry Pi modelo B .....	4
Figura 1.2 Partes del Raspberry Pi modelo B .....	5
Figura 1.3 Arduino UNO .....	7
Figura 2.1 Minicomputadora Raspberry Pi modelo A y B .....	13
Figura 2.2 Partes del Raspberry Pi modelo B .....	17
Figura 2.3 Procesador ARM en una impresora HP .....	19
Figura 2.4 Puerto GPIO del Raspberry Pi .....	25
Figura 2.5 Cabecera P1 conector GPIO .....	28
Figura 2.6 Tabla con los pines de la cabecera P1 de la RPi .....	30
Figura 2.7 Numeración de los pines para los dos modos .....	31
Figura 2.8 Arduino UNO .....	32
Figura 2.9 Pines del Arduino UNO con Atmega 168/328 .....	39
Figura 2.10 Flujo de consumo mundial anual de energía primaria .....	41
Figura 2.11 Espectro Electromagnético .....	47
Figura 2.12 Composición espectral de la radiación solar .....	48
Figura 2.13 Efecto fotoeléctrico .....	53
Figura 2.14 Material cristalino .....	55
Figura 2.15 Material policristalino .....	55
Figura 2.16 Material amorfo .....	56

Figura 2.17 Curva característica I-V de una celda FV .....	59
Figura 2.18 Efecto de la intensidad de radiación .....	62
Figura 2.19 Potencia máxima de salida durante el día.....	63
Figura 2.20 Potencia máxima de un módulo en función de la hora del día .....	64
Figura 2.21 Constitución de un módulo fotovoltaico.....	66
Figura 2.22 Esquema de un SFV que entrega tensión continua y alterna.....	67
Figura 3.1 Diagrama de flujo ejercicio #1 .....	79
Figura 3.2 Diagrama de flujo ejercicio #2.....	82
Figura 3.3 Diagrama de bloque del proyecto .....	85
Figura 3.4 Diagrama de flujo del proyecto .....	86
Figura 4.1 Simulación del circuito conmutador.....	113
Figura 4.2 Ingreso de horas y minutos.....	115
Figura 4.3 Implementación del circuito conmutador en el protoboard .....	116
Figura 4.4 Sensor DHT11 .....	117
Figura 4.5 Detector de humedad relativa y temperatura del ambiente .....	117
Figura 4.6 Ejecución de rutina en Raspberry Pi.....	118
Figura 4.7 Diagrama de bloques del sistema fotovoltaico .....	119
Figura 4.8 Controlador de carga .....	125
Figura 4.9 Diagrama de bloques de la placa reguladora de carga .....	126
Figura 4.10 Máxima corriente de salida del convertidor DC/DC.....	128



Figura 4.11 Voltaje a la salida del panel solar ..... 129

Figura 4.12 Voltaje a la salida del controlador de carga..... 130

## INTRODUCCIÓN

Este proyecto abarca el uso de la Energía Solar y aprovecharla por medio del uso de un Sistema Fotovoltaico para la carga respectiva de una batería de Ion-Litio que servirá para la alimentación del Raspberry Pi y del Arduino UNO, siendo un sistema autónomo tanto en el día como en la noche.

Se ha dividido en dos partes este proyecto los cuales son hardware y software. El hardware está constituido por el Raspberry Pi, el Arduino y el Sistema Fotovoltaico SFV, tarjetas electrónicas para la regulación y carga de la batería, y el uso del RTC (Real Time Clock) circuito electrónico. El Raspberry Pi posee un puerto de comunicación llamado GPIO que por medio de este se realiza la comunicación con el Arduino UNO, para indicar que se debe apagar al Raspberry Pi y el tiempo que permanecerá apagado hasta volver a encenderlo.

En lo que consiste al software, el Sistema Operativo de la Raspberry Pi se ha desarrollado en Linux, la distribución de Linux que se utilizará como Sistema Operativo para la minicomputadora será la denominada Raspbian.

El funcionamiento del sistema consiste en captar la energía del sol por medio del panel solar, transformándola en energía eléctrica, siendo regulado el voltaje necesario para cargar la batería de Ion-Litio, el voltaje de la batería es elevado por medio de una placa electrónica que es un regulador DC a DC, para alimentar al Raspberry PI y al Arduino. El Raspberry PI realiza un evento, como por ejemplo, adquisición de datos ambientales (temperatura, humedad relativa, etc.) al culminar este evento envía una señal por medio del puerto GPIO al Arduino UNO indicándole que debe apagarlo un cierto tiempo ya ajustado por el usuario, siendo el Arduino UNO un reloj en tiempo real usando el RTC (Real Time Clock) para conmutar la alimentación dada de +5 voltios al Raspberry PI, siendo este proceso continuo logrando así ahorrar la energía almacenada en la batería y así conseguir un bajo consumo de energía para el Raspberry PI.

## **CAPITULO 1**

### **DESCRIPCIÓN GENERAL DEL PROYECTO**

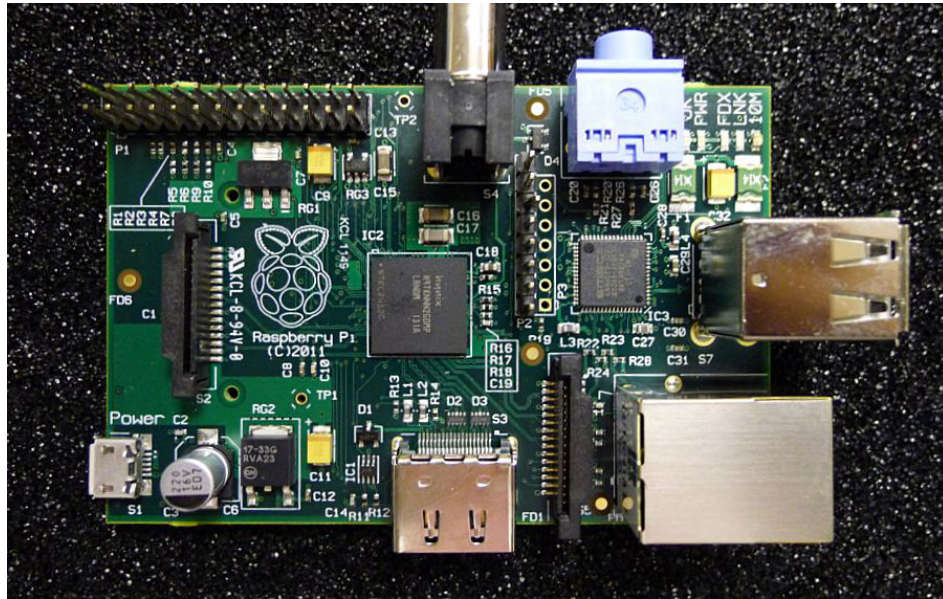
#### **1.1 ANTECEDENTES**

Las computadoras personales como dispositivos electrónicos de consumo empezaron aproximadamente en 1977 con la introducción de las microcomputadoras, aunque ya se habían implementado antes computadoras como centrales, sistemas monousuario. Después del

desarrollo del microprocesador, las computadoras personales llegaron a ser más económicas y se popularizaron.

Antes de la introducción del microprocesador a principios de los años 1970, las computadoras eran sistemas grandes y costosos cuyos diseños eran para grandes corporaciones, universidades, agencias o instituciones de tamaño similar. La interacción usuario-maquina no era directamente sino que se preparaba tareas para el computador en unas tarjetas, estos procesos se lo hacía por lotes y después se recogía los resultados. El trabajo asignado al computador podía tardar horas o días hasta obtener un resultado. Poco a poco se fue reduciendo el tamaño del computador y el precio, cuando se fabrica el microchip, este no contenía al microprocesador en dicho chip, esto significaba que todavía eran grandes y difíciles de fabricar, aumento la fabricación porque esto hizo que se redujeran placas inmensas electrónicas por un chip que hacia el mismo trabajo. En 1972 se introdujo al mercado el primer microprocesador INTEL 4004 y así fueron evolucionando los microprocesadores, diferentes empresas competían en ese tiempo y hoy en día siguen compitiéndose en crear un procesador más rápido, más reducido, más compacto y que no consuma tanta energía.[1]

En estos últimos años se ha incentivado en crear placas electrónicas de uso didáctico para universidades, colegios y sea accesible en precio. En 2006 se inicia los primeros diseños de Raspberry PI se basaban en el microcontrolador Atmel ATmega644. En mayo del 2009 la fundación Raspberry PI fue fundada en Caldecote, South de Cambridgeshire, Reino Unido como una asociación caritativa que es regulada por la Comisión de Caridad de Inglaterra y Gales. Luego el administrador de la fundación, Eben Upton contactó a un grupo de profesores, académicos y entusiastas de la informática para crear un ordenador que impulsen a los niños en aprender informática, el primer prototipo basado en ARM se lo creo del mismo tamaño que una memoria USB, este contaba con un puerto USB de un extremo y puerto HDMI del otro. **[2]**

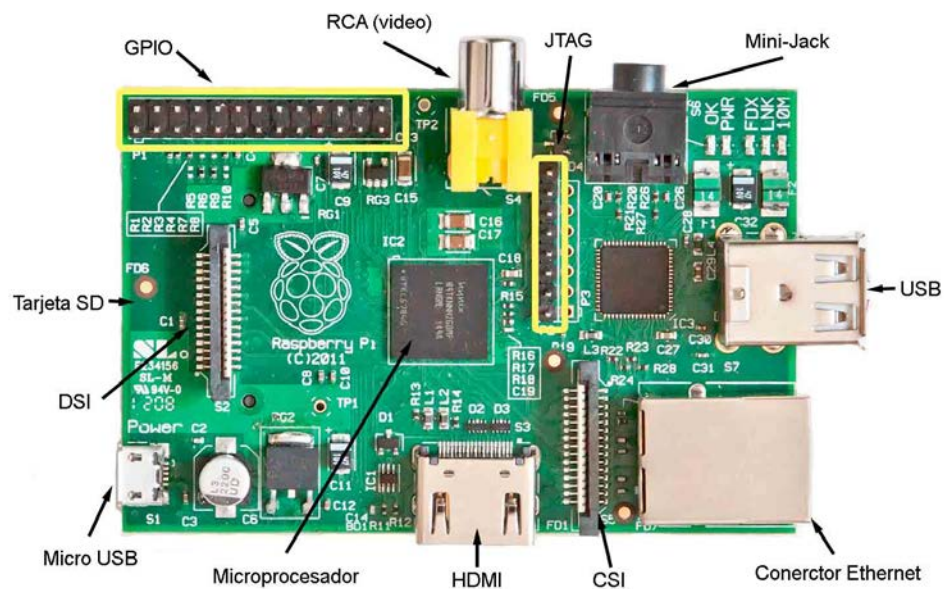


**Fig. 1.1 Raspberry Pi Modelo B [2]**

El pre-lanzamiento se dio en el 2011 y el lanzamiento al mercado en el año 2012 [2], en sí, es una placa computadora (SBC) de bajo costo desarrollada en Reino Unido por la fundación Raspberry Pi, con dimensiones de placa base de 85x54 milímetros en el que se aloja un chip Broadcom BCM2835 con procesador ARM hasta 1Ghz de velocidad (modo turbo haciendo overclock), GPU VideoCore IV y 512 Mbytes de memoria RAM (Las primeras placas contaban con solo 256 de RAM). Usa como medio de almacenamiento tarjetas de memorias SD o micro SD, cargador micro USB de al menos 1000mah, hay varios modelos actualmente pero siempre en todos se dispone de un puerto de salida HDMI, otro de tipo RCA, mini-jack

de audio, puerto USB 2.0 (Modelos A y A+, B dispone de dos USB y B+ y Pi 2 disponen de 4 USB) en el cual se puede conectar un teclado y ratón.

En cuanto a la conexión a la red, se dispone de un puerto Ethernet (los modelos A y A+ no disponen de puerto Ethernet) para enchufar un cable RJ-45 directamente al router o adaptador inalámbrico Wifi compatible. [3]



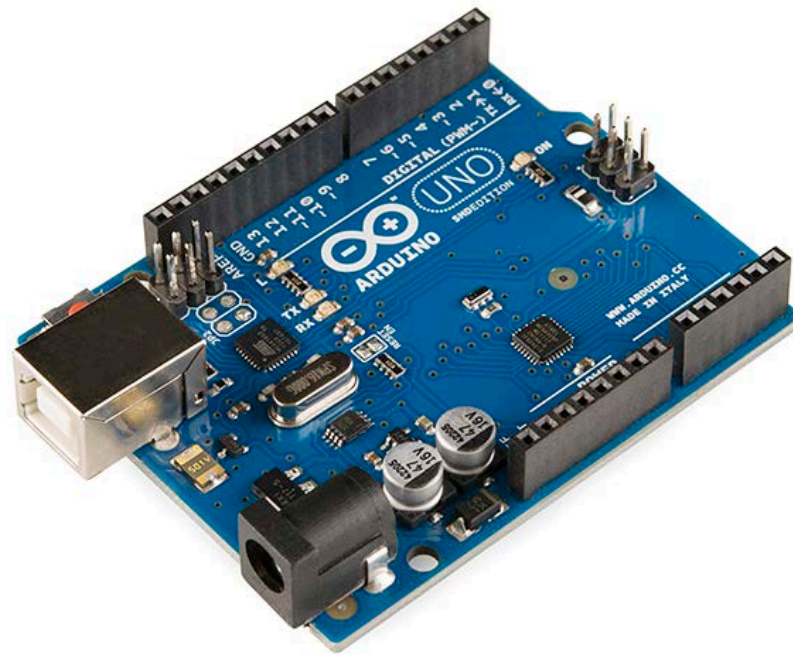
**Fig. 1.2 Partes del Raspberry PI Modelo B [4]**

Arduino se inició en el año 2005 como un proyecto para estudiantes en el Instituto IVREA, en Ivrea (Italia). Los estudiantes en ese tiempo usaban un



microcontrolador BASIC Stamp, cuyo costo era 100 dolares americanos, por ende era muy costoso. Uno de los fundadores de Arduino, Massimo Banzi, daba clases en Ivrea. Se unieron varios estudiantes para su creación, cuando ya se tenía hecha la plataforma, los investigadores trabajaron para hacerlo más ligero, más económico y disponible para la comunidad de código abierto (Hardware y código abierto). Más tarde se cerró el instituto y por necesidad de subsistir ante el cierre del instituto de diseño interactivo IVREA y no por hacerlo comercial se creó la placa Arduino. Para la primera producción se tomó en cuenta que no sobrepase el valor de venta de 30 euros, que fuera ensamblado en color azul, debía ser Plug and Play y que trabajara en diferentes plataformas como MacOSX, Windows, y GNU/Linux.

Entonces Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo que facilita el uso de la electrónica en proyectos multidisciplinarios. El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, Atmega8 por su sencillez y bajo costo. En software consiste en un entorno de desarrollo que implementa el lenguaje Processing/Wiring y el cargador de arranque que es ejecutado en la placa. **[5]**



**Fig. 1.3 Arduino UNO [5]**

## **1.2 OBJETIVOS**

### **1.2.1 OBJETIVO GENERAL**

- El objetivo principal para la implementación de nuestro proyecto de tesina es el usar la minicomputadora Raspberry Pi conectada al Arduino que es un reloj en tiempo real que apagará y encenderá al Raspberry Pi cuando haya culminado el evento en ejecución, todo esto alimentado por medio de un Sistema

Fotovoltaico, teniendo como resultando un bajo consumo de energía del sistema.

### **1.2.2 OBJETIVOS ESPECÍFICOS**

- Cargar la batería por medio del panel solar.
- Alimentar al sistema en el día por medio del panel solar y por la noche con la batería de plomo - ácido.
- Implementar un reloj en tiempo real usando el Arduino UNO y el RTC.
- Implementar un código en lenguaje C para el Arduino UNO.
- Implementar un Código en lenguaje Python para realizar el evento que ejecutara el Raspberry Pi y para el envío de las señales al Arduino UNO por el GPIO.
- Implementar la conmutación de la alimentación del Raspberry Pi al apagarlo por medio del Arduino UNO.

### **1.3 IDENTIFICACIÓN DEL PROBLEMA**

En la actualidad se está incursionando en el uso del Raspberry Pi y del Arduino, empezándose con proyectos en las universidades y últimamente con adolescentes con cursos accesibles, pero se necesita más personas

con los conocimientos necesarios para estos dispositivos, también que sea más común la venta del Raspberry Pi y Arduino para que sea más fácil el uso y el aprendizaje. Estos dispositivos fueron creados para un aprendizaje más sencillo ya sea para informática usando el Raspberry Pi o el Arduino para los microcontroladores, es ahí donde se enfoca esta tesina, en usar nuevos dispositivos que hoy en día existen y son evoluciones de partes electrónicas que ya se han manejado anteriormente pero ahora con visión hacia la enseñanza y conocimiento temprano ya sean en colegios, universidades, etc.

Un problema a solucionar y que es impedimento para acceder a este conocimiento es la escases de estos dispositivos en nuestro país, que sean más personas que estén trabajando y creando nuevos proyectos, sabiendo de antemano que estos dispositivos son de tamaño portable, bajo costo y que permiten crear infinidad de usos desde algo sencillo hasta algo complejo, la demanda de este conocimiento haría que en nuestro país el uso de estos dispositivos sea más común y comercial.

La implementación de un proyecto de tesina mediante el uso de panel solar como alimentación para el Raspberry Pi y el Arduino demanda

conocimientos en la parte electrónica para la regulación del voltaje y carga de la batería, también usar lo aprendido en programación como es el lenguaje C y otro lenguaje que se debió como autoaprendizaje, leyendo libros o el uso del internet para lograr los objetivos propuestos y así poder competir y ser creador de cosas nuevas como en otros países que a partir de conocimientos adquiridos de algo ya creado se pueda innovar y ayudar a que sea accesible a todos y principalmente a la juventud.

#### **1.4 LIMITACIONES**

Unas de las limitaciones en esta tesina es en conocer los lenguajes de programación que se pueden usar en la ejecución del proyecto, conceptos básicos, configuraciones y demás que están relacionados con los dos dispositivos como son el Raspberry Pi y el Arduino. **[6][7]**

Es importante tener todas las herramientas necesarias como un monitor para que muestre los avances en el código del Raspberry Pi, los programas previamente seleccionados a usar, instalados en la PC y drivers que permitan la comunicación de estos dispositivos con la PC u otros periféricos.

Las limitaciones debido a la alimentación que se le da al Raspberry Pi y al Arduino UNO, ya que dependerá de la capacidad de almacenamiento de la batería de plomo - ácido que se use y de la radiación solar la cual estará sometido el Sistema Fotovoltaico SFV.

En cuanto a las limitaciones de la adquisición del Raspberry Pi y del Arduino UNO se hace un poco complicado, porque no es tan común la venta de estos dispositivos y si se lo hace importando el tiempo de culminar los objetivos propuestos en esta tesina se extendería y por ende dificultaría obtener rápidos los resultados esperados.

La minicomputadora Raspberry Pi y el Arduino UNO está ingresándose poco a poco en el mercado del País y esto hace que su estudio sea más consultado, ya sea vía internet o libros que sean un estudio autodidacta para nuevos proyectos, ya sean por investigación, universitarios o en colegios técnicos.

La limitación también es que el reloj RTC que se ejecuta en el Arduino UNO solo puede conmutar, en el tiempo asignado por el usuario previamente, un solo dispositivo, es decir en este caso del Raspberry Pi.

## **CAPITULO 2**

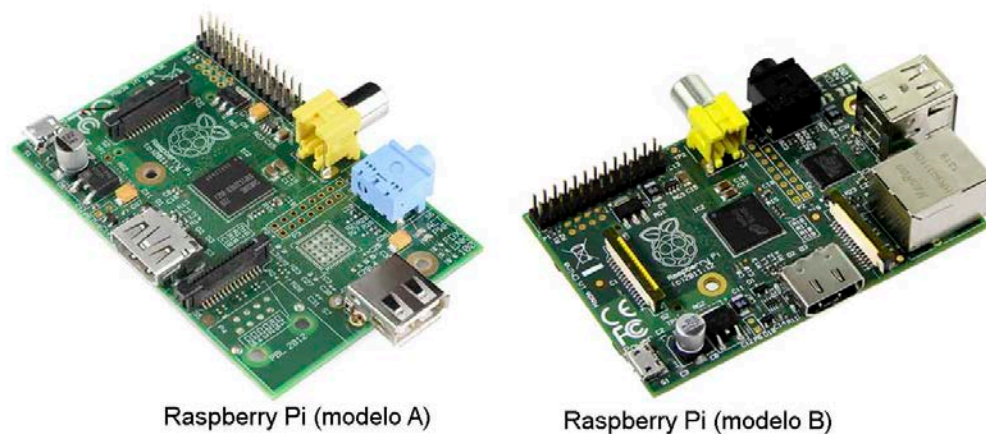
### **FUNDAMENTO TEÓRICO**

#### **2.1 RESUMEN**

Para comprender con más claridad la estructura de este proyecto en este capítulo se describirá con detalle todas las partes del proyecto su funcionamiento, como se realizará las conexiones, parte de hardware y software que se usará, como también los periféricos, descripción de un sistema fotovoltaico, actuadores (pulsadores) y el RTC (Real Time Clock),

como también las especificaciones técnicas del Raspberry Pi y del Arduino UNO.

## 2.2 MINICOMPUTADORA RASPBERRY PI MODELO B



**Fig. 2.1 Minicomputadora Raspberry Pi modelo A y B [8]**

El Raspberry Pi se puede definir como un ordenador de bajo costo integrado en una placa de dimensiones pequeñas de 85 x 53 mm, pero Raspberry Pi va mucho más allá de todo esto, ya que es una plataforma abierta (Open Hardware) que permite a los usuarios cargar varios sistemas operativos y distribuciones ligeras donde pueden desarrollar sus propias aplicaciones a un costo reducido y con una gran portabilidad ya que comparado con un



ordenador convencional en nuestros hogares, los periféricos y almacenamientos están por separados.

La Raspberry Pi tiene un procesador central ARM1176JZF-S el cual trabaja una frecuencia de 700MHz (aunque es posible “overclockearlo” a 1GHz), como procesador gráfico usa un Broadcom VideoCore IV y 256Mb de memoria RAM. Como particularidad, el Raspberry Pi no usa un disco duro como almacenamiento del sistema operativo, sino que almacena en una tarjeta SD, de la cual extrae los datos para cargar el programa.

Existen 2 modelos de Raspberry Pi básicos, el modelo A y el modelo B, actualmente el modelo A no se comercializa, ya que ha ido el modelo B desplazándolo. La diferencias fundamentales entre el modelo A y el modelo B, es que el modelo A tan solo tiene un puerto USB, carece de controladora Ethernet, por supuesto este modelo era más barato que el modelo B, pero carecer de estos elementos hace menos adquisitivo al consumidor sabiendo que la diferencia entre el precio de compra no está tanto, figura 2.1. **[8]**

### 2.2.1 CARACTERÍSTICAS

A continuación describimos las especificaciones técnicas más relevantes del fabricante:

- 1 puerto Ethernet 10/100.
- 1 conector mini-jack para la salida de audio.
- 1 conector micro-USB que se usa para la alimentación de la placa, podemos usar un cargador de móvil que nos dé 5V y al menos 700mA.
- 1 lector para la tarjeta SD en la cual cargaremos el sistema operativo o la distribución ligera que queramos ejecutar.
- 1 conector DSI "*Display Serial Interface*", que es un puerto serie de alta velocidad al que se le conecta un display LCD.
- 1 conector CSI "*Camera Serial Interface*" de similares características que el conector DSI, pero específico para conectar un módulo de cámara para la Raspberry.

- 1 un conector RCA el cual nos dará una señal de video analógico por si el dispositivo al cual fuéramos a conectar la Raspberry no tuviese entrada HDMI.
- 1 conector HDMI del cual podemos sacar.
- 2 puertos USB 2.0 a los cuales podemos conectar cualquier tipo de periférico.
- Una señal de video con una resolución de 1080p a 30Hz.
- 1 Puerto GPIO "*General Port Input Output*" este es un puerto que usa nuestra Raspberry para interactuar con el exterior, podrá sacar señales que activen cosas o leer información de sensores.
- 1 Un conector JTAG "*Joint Test Action Group*" el cual es un conector que se usa para pruebas y depurado de aplicaciones.

### **[8]**

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora); y promueve principalmente el aprendizaje del lenguaje de

programación Python. Otros lenguajes también soportados son Tiny BASIC, C, Perl y Ruby. [9]

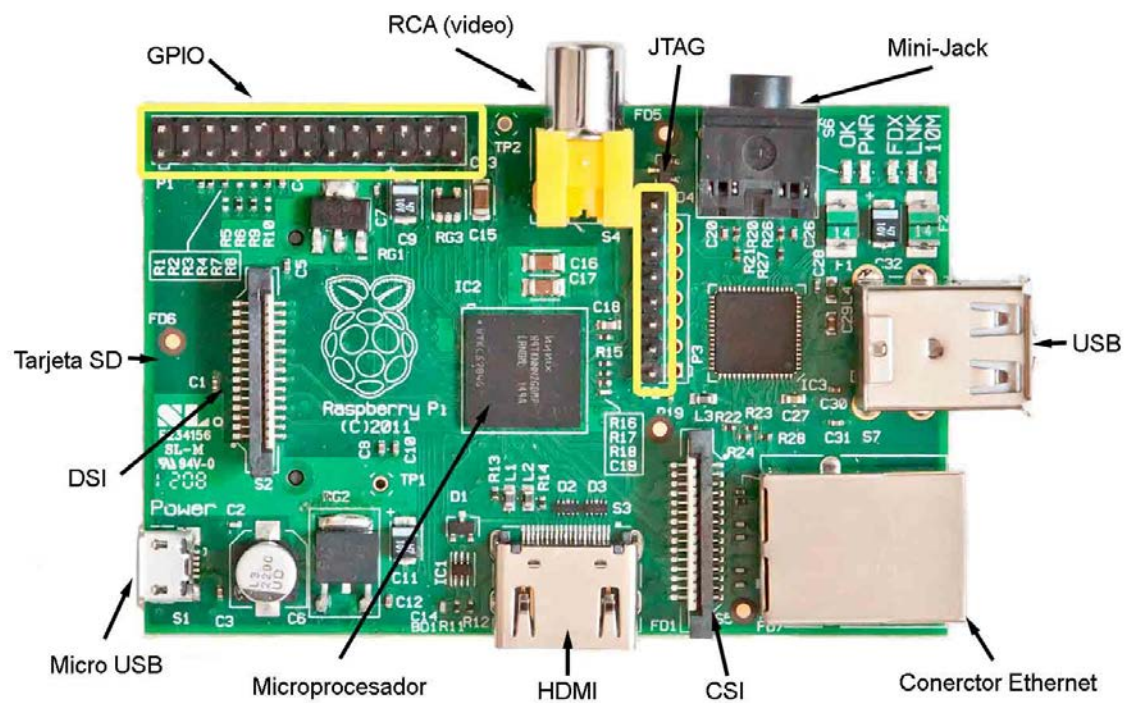


Fig. 2.2 Partes del Raspberry Pi modelo B [8]

## 2.2.2 PROCESADOR

ARM es una arquitectura RISC (Reduced Instruction Set Computer ó Ordenador con conjunto Reducido de Instrucciones) de 32 bits desarrollada por ARM Holdings. La arquitectura ARM es el conjunto de instrucciones de 32 bits más ampliamente utilizado en unidades

producidas. Concebida originalmente por Acorn Computers para uso en ordenadores personales.

La simplicidad de los procesadores ARM los hace ideales para aplicaciones de baja potencia. Como resultado se han convertido en dominantes en el mercado de la electrónica móvil e integrada, encarnados en microprocesadores y microcontroladores pequeños, de bajo consumo y bajo costo. Se utilizan ampliamente en la electrónica de consumo, incluyendo PDA, tabletas, teléfono inteligente, teléfonos móviles, videoconsolas portátiles, calculadoras, reproductores digitales de música y medios (fotos, videos, etc.) y periféricos de ordenador como discos duros y routers.

La arquitectura ARM es licenciable. Las empresas que son titulares de licencias ARM actuales o anteriores incluyen a Alcatel-Lucent, Apple Inc., AppliedMicro, Atmel, Broadcom, Cirrus Logical, Digital Equipment Corporation, Ember, Energy Micro, Freescale, Intel (a través de DEC), LG, Marvell Technology Group, Microsemi, Microsoft, NEC, Nintendo, Nokia, Nuvoton, Nvidia, Sony, MediaTek, NXP (antes Philips Semiconductors), Oki, ON Semiconductor, Psion,

Qualcomm, Samsung, Sharp, STMicroelectronics, Symbios Logic, Texas Instruments, VLSI Technology, Yamaha, y ZiiLABS.

Los procesadores ARM son desarrollados por ARM y los titulares de licencias de ARM. Prominentes familias de procesadores ARM desarrollados por ARM Holdings incluyen el ARM7, ARM9, ARM11 y Cortex. Los procesadores ARM notables desarrollados por los licenciatarios incluyen Applied Micro Circuits Corporation X-Gene, DEC StrongARM, Freescale i.MX, Marvell Technology Group XScale, NVIDIA Tegra, Qualcomm Snapdragon, Texas Instruments OMAP, Samsung Exynos, Apple Ax, St-Ericsson NovaThor, Huawei K3V2 e Intel Medfield. [10]



**Fig. 2.3 Procesador ARM en una impresora HP [10]**

### 2.2.3 DISPOSITIVO DE ARRANQUE

El dispositivo de arranque para que la Raspberry Pi pueda funcionar es una tarjeta de memoria SD y su capacidad es recomendable que sea por lo menos de 4Gb porque el sistema operativo de Linux a usar el Raspbian Debian Wheezy que aproximadamente pesa 974mgb y más los programas que se necesitaran descargar como complementos para el proyecto, se puede adquirir memorias SD de más capacidad pero su valor de compra es proporcional, es decir será mayor.

Antes de encender el Raspberry Pi y ya adquirido la memoria SD, ahora se debe decidir cuál de varios sistemas operativos que soporta el procesador del Raspberry Pi se usará para la ejecución del proyecto, y por ende descargarlo. **[11]**

Como primer paso para empezar a “grabar” el sistema operativo en la memoria SD, previamente seleccionado cual se desea usar, en este caso “Raspbian Debian Wheezy”, se puede descargar de la propia página de dicho nombre de la minicomputadora, la versión

más reciente y de qué forma desea descargarlo en su ordenador.

**[12]**

Además la página de la fundación Raspberry Pi nos da dos opciones de descarga, puede ser por ZIP o por torrent, en este caso si se desea no perder el progreso de la descarga al apagar el ordenador se recomienda usar la opción torrent.

Para obtener el inicio de la descarga torrent, es necesario tener el software compatible con el servidor, en este caso se puede descargar e instalarlo el programa uTorrent que es muy sencillo y practico en su uso. **[13]**

La distribución de Linux Raspbian Debian Wheezy es la más común que se usa para iniciar el aprendizaje en el Raspberry Pi, porque es similar a los comandos en Linux propiamente dicho, y se hace más sencillo que el estudiante de informática u otras carreras que tengan un conocimiento básico, ayudaría a reforzar sus conocimientos o para aprender desde cero, obviamente queda a disposición escoger el sistema operativo que se adapte a sus necesidades.



Para reducir el tiempo de descarga en la página de la fundación Raspberry Pi al descargar el sistema operativo, viene comprimido, tener en cuenta cuando se haya culminado la descarga, ir a donde se descargó el archivo y descomprimirlo en una carpeta o donde sea más conveniente.

Una vez que se ha descomprimido el archivo el cual será una imagen, el cual contiene el sistema operativo seleccionado previamente, este está hecho para que se ajuste y sea ejecutable desde la memoria SD como si fuera el disco duro para la minicomputadora, y arranque desde allí el encender el Raspberry Pi.

Los pasos a seguir para descargar el sistema operativo y guardarlo en la memoria SD, se encuentra más detallado en la parte de ANEXO.

#### **2.2.4 SISTEMAS OPERATIVOS SOPORTADOS**

La Raspberry Pi soporta una gran cantidad de sistemas operativos y distribuciones ligeras, aunque normalmente son sistemas Linux que

la versión más usada y fue creada propiamente para esta placa es Raspbian versión basada en Debian. **[14]**

### **Sistemas operativos completos:**

- Windows 10
  
- AROS
  
- Linux
  - Android
  - Arch Linux ARM
  - Debian Whezzy Soft-Float, versión de Debian sin soporte para coma flotante por hardware
  - Firefox OS
  - Gentoo Linux
  - Google Chromium OS
  - Kali Linux
  - Open webOS
  - PiBang Linux, distribución Linux derivada de Raspbian con diferente escritorio y aplicaciones
  - Pidora, versión Fedora Remix optimizada

- QtonPi, distribución linux con un framework de aplicaciones multiplataforma basado en Qt framework
  - Raspbian, versión de Debian Wheezy para ARMv6 con soporte para coma flotante por hardware
  - Slackware ARM, también conocida como ARMedslack
- 
- Plan 9 from Bell Labs
  
  - RISC OS 5
  
  - Unix
    - FreeBSD
  
    - NetBSD

### **Distribuciones ligeras multipropósito:**

- Moebius, distribución ligera ARM HF basada en Debian que usa el repositorio de Raspbian y que cabe en una tarjeta SD de 1GB, usa pocos servicios y está optimizada para usar poca memoria
  
- Squeezed Arm Puppy, una versión de Puppy Linux (Puppi) para ARMv6 (sap6) específicamente para Raspberry Pi

- Minibian, distribución ligera basada en Raspbian.

### Distribuciones ligeras de único propósito:

- Instant WebKiosk, sistema operativo con solo un navegador
- IPFire
- Micro Elastix, solución de código abierto para comunicaciones unificadas
- Raspbmc
- Xbian [15]

## 2.2.5 PUERTO GPIO

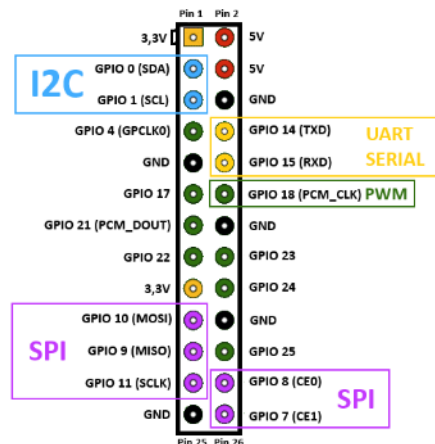
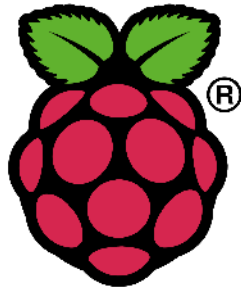


Fig. 2.4 Puerto GPIO del Raspberry Pi [16]

GPIO (General Purpose Input Output) entrada y salida de propósitos generales, son pines que se encuentran en un chip, cuyo control es a través de software en tiempo de ejecución por parte del usuario. Estos pines GPIO no tienen definido un propósito específico ni tampoco predeterminado en ningún dispositivo que los lleve incorporados. Estos pines ofrecen controles externos digitales permitiendo una funcionalidad agregada al sistema. Las GPIO se usan en varios dispositivos programables como por ejemplo en Arduino, para controlar motores, usando PWM, mostrar datos en pantallas LCD, etc.

#### **2.2.5.1 CARACTERÍSTICAS GENERALES DE LOS GPIO**

Se pueden configurar como entrada o como salida. Se tiene un registro por cada pin del puerto GPIO llamado DDR (Data Direction Register), si en este se escribe un "1", se establece como output y si se escribe un "0" como input. Estos también puede ser activados y desactivados cuando se lo desee. Los valores de entrada se pueden leer como un alto "1" y nivel bajo "0", los valores de salida puede ser de lectura o de escritura. Los valores de entrada a menudo

se pueden utilizar como IRQs (interrupt requests), es decir interrupciones de Hardware.

### **2.2.5.2 GPIO DE LA RPI**

Por medio de la GPIO se puede encender o apagar elementos electrónicos, responder a eventos o tomar mediciones de algún sensor.

La ventaja de la RPi es que es un ordenador barato que cualquier persona que no tiene mucho conocimiento en la electrónica puede usarlo sin dañar un ordenador de mayor costo. Las GPIO pueden ser entradas y salidas, en este caso como ordenador pueden ser un teclado como entrada y como salida los altavoces o video. Pero además se puede conectar otras cosas al ordenador.

#### **Tipos de señales en componentes electrónicos:**

- Señal digital: Un ordenador al verificar si se ha presionado una tecla, él lo reconoce como un alto

“1” o un bajo “0”, solo esos dos estados, por ende usa el ordenador señales digitales “0” o “1”.

- Señal analógica: Es una señal continua que varía a lo largo del tiempo, la mayoría de estas señales pueden representar: temperatura, luminosidad, humedad, etc., son señales analógicas. En la RPi si desea introducir señales analógicas deben ser convertidas a digitales para su procesamiento.

La RPi considera a 3.3 voltios que es un estado HIGH (1) y a 0 voltios como LOW (0). También ofrece interfaces de bajo nivel destinados a conectarse directamente con los chips y módulos de los subsistemas que contiene la placa. Como se muestra en la figura 2.5 el puerto GPIO se encuentra a la altura de los puertos USB.



**Fig. 2.5** Cabecera P1 conector GPIO [17]

El número de pines de la cabecera P1 cuenta con 26 pines (2 x 13) y pueden ser usados para SPI, I2C, transmisión serial UART, alimentación 3.3 voltios, 5 voltios, GND, PWM, y demás E/S. Estos pines no son “plug and play” por eso hay que tener cuidado en las conexiones que se realiza. Los pines usan 3.3 voltios y no soportan 5 voltios, no hay protección de sobre-voltaje.

Si no se usan las funciones como SPI, I2C, y UART se podrían deshabilitarlos para obtener 17 pines GPIO disponibles (8 pines GPIO + 5 pines SPI + 2 pines I2C + 2 pines UART). Cada pin GPIO puede entregar hasta 16mA, pero en total todos los pines no pueden sobrepasar los 50mA. Por lo cual si se desea usar los 17 pines a la vez, ninguno tendría que consumir más de 3mA.



BCM2835	Conector GPIO P1		BCM2835
3.3v	1	2	5v
I2C SDA GPIO 2	3	4	5v
I2C SCL GPIO 3	5	6	GND
GPIO 4 GPCLK	7	8	UART TXD GPIO 14
GND	9	10	UART RXD GPIO 15
GPIO 17	11	12	PWM GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
3.3v	17	18	GPIO 24
SPI MOSI GPIO 10	19	20	GND
SPI MISO GPIO 9	21	22	GPIO 25
SPI SCLK GPIO 11	23	24	SPI CEO GPIO 8
GND	26	26	SPI CE1 GPIO 7

*Revisión 2*

**Fig. 2.6** Tabla con los pines de la cabecera P1 de la RPi [17]

Para saber la revisión que se tiene en la placa que se está usando para el proyecto, se pueden buscar en la web el modelo de la Raspberry Pi que se tiene, porque cada revisión cambia el orden de los pines. [17]

### 2.2.5.3 DIFERENCIAS PINOUTS DEL GPIO (BCM Y BORD)

Hay dos tipos de numeración de los pines, la física o BOARD y la numeración del chip SoC Broadcom que los controla. [18]

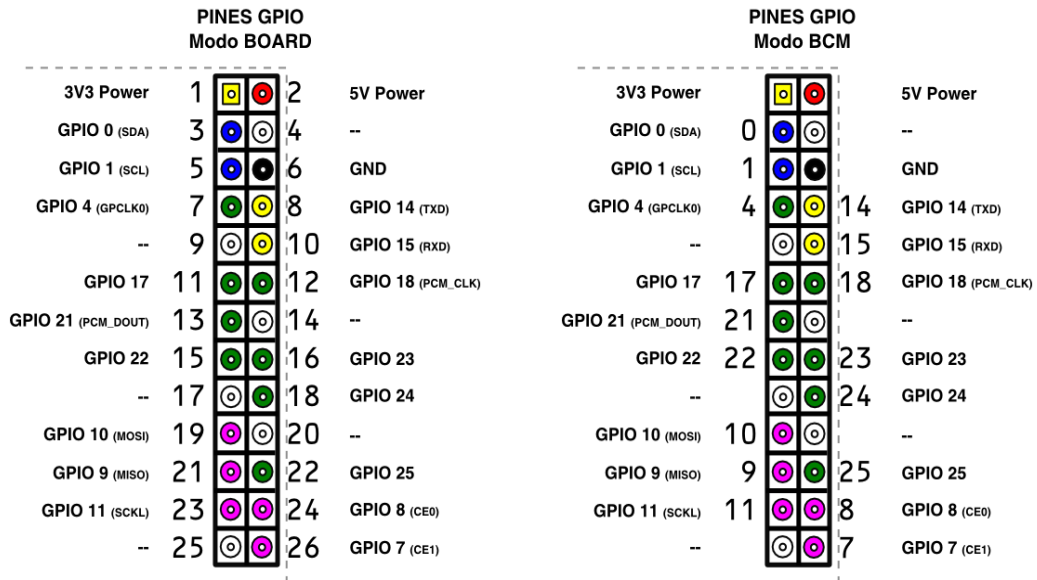
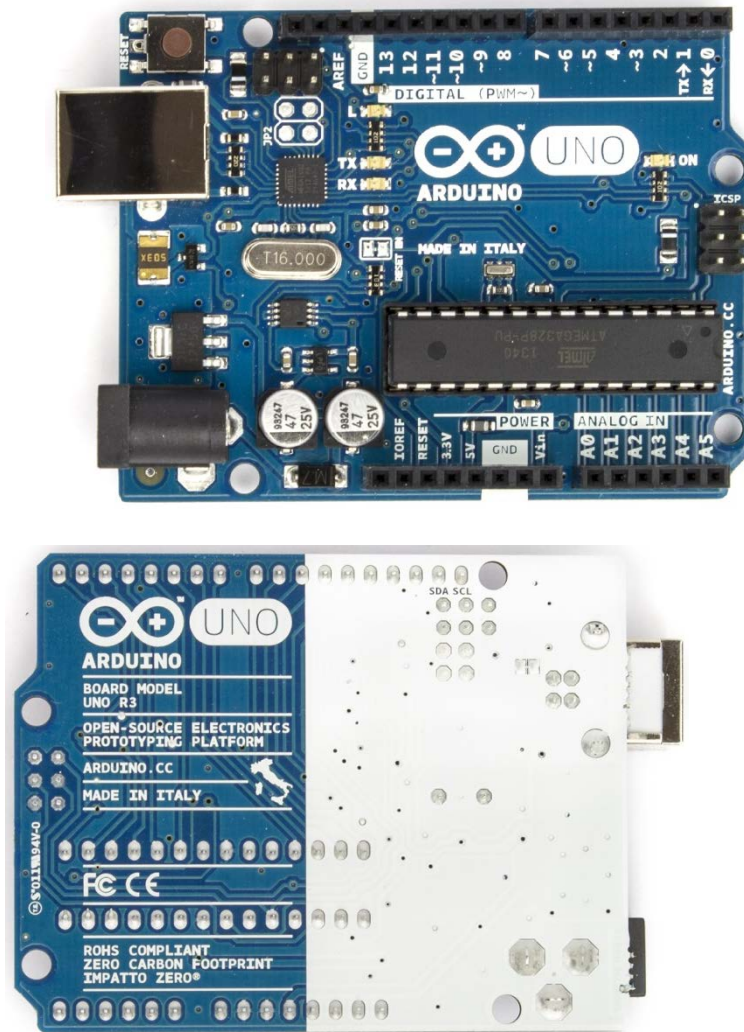


Fig. 2.7 Numeración de los pines para los dos modos [18]

## 2.3 ARDUINO UNO



**Fig. 2.8 Arduino Uno. [21]**

Arduino es una plataforma electrónica de código abierto basado hardware y software de fácil uso. Está diseñado para aficionados en la electrónica, microcontroladores, y cualquier persona que desee crear objetos con una

placa que interactúe con el usuario por medio de dispositivos conectados a esta placa que es de bajo costo y alta portabilidad. **[19]**

Hardware consiste en un microcontrolador de bajo costo y sencillez, como son el Atmega 168, 328, 1280,8 que permiten desarrollar múltiples diseños. Software consiste en un entorno de desarrollo que usa el lenguaje de programación Processing/Wiring y el arranque que se ejecuta en la misma placa.

El Arduino también se puede conectar a software como Adobe Flash, Processing, Max/MSP, Pure Data. El diseño de la placa se puede crear por nuestra cuenta, porque el diseño e implementación se encuentra en descarga libre por los creadores.

El microcontrolador del Arduino se programa con el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing), el proyecto creado no depende que esté conectada la PC para su funcionamiento.

### 2.3.1 DIFERENTES PRODUCTOS DE ARDUINO.

Se dividen en 4 diferentes productos, tales como: tablas, escudos, kits, y accesorios.

- Arduino Uno
- Arduino Leonardo
- Arduino Due
- Arduino Yún
- Arduino Tre (En Desarrollo)
- Arduino Zero (En Desarrollo)
- Arduino Micro
- Arduino Esplora
- Arduino Mega ADK
- Arduino Ethernet
- Arduino Mega 2560
- Arduino Robot

- Arduino Mini
- Arduino Nano
- LilyPad Arduino Simple
- LilyPad Arduino SimpleSnap
- LilyPad Arduino
- LilyPad Arduino USB
- Arduino Pro Mini
- Arduino Fio
- Arduino Pro

**Escudos:**

- Arduino GSM Shield
- Arduino Ethernet Shield
- Arduino WiFi Shield
- Arduino Wireless SD Shield

- Arduino USB Host Shield
- Arduino Motor Shield
- Arduino Wireless Proto Shield
- Arduino Proto Shield

**Kits:**

- The Arduino Starter Kit
- Arduino Materia 101

**Accesorios:**

- TFT LCD Screen
- USB/Serial Light Adapter
- Arduino ISP
- Mini USB/Serial Adapter [20]

**2.3.2 ENTRADAS Y SALIDAS**

Consta de 14 entradas digitales que son configurables como entrada y como salida que operan a 5 voltios usando “pinMode()”,

“digitalWrite()”, y funciones de “digitalRead()”. Cada pin puede proporcionar y recibir como máximo 40 mA, y contiene una resistencia Pull-Up interna (desconectado por defecto) de 20-50 Kohms.

- **Serial: 0 (RX) y 1 (TX).** Receptor (RX) y transmisor (TX) datos TTL serial. Estos pines son conectados a los correspondientes del Atmega 328.
- **Interrupciones externas: 2 y 3.** Estos pines pueden configurarse para provocar una interrupción en un valor de nivel bajo o un cambio en el valor. Esta función de “attachInterrupt()” se usa para este propósito.
- **PWM: 3, 5, 6, 9, 10, y 11.** Provee salidas PWM de 8-bit con la función “analogWrite()”.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** Estos pines soportan comunicación SPI usando “SPI library”.
- **LED: 13.** Se incorpora un LED conectado a pin digital 13. Cuando el pin es de valor alto, el LED está encendido, nivel bajo esta apagado.

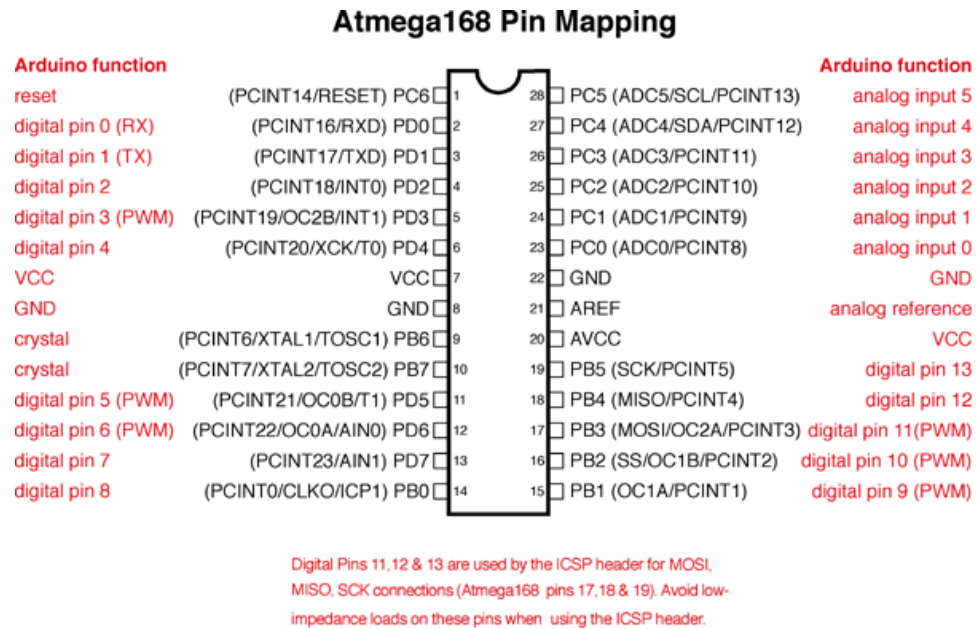


El Arduino Uno dispone de 6 entradas analógicas, denominado A0 a A5, cada uno de los cuales 10 bits de resolución (es decir 1024 Valores diferentes). Medidas de 0 a 5 voltios, aunque es posible cambiar el extremo superior del rango mediante el pin AREF y el “analogReference()”. Además, algunas patillas tienen funciones especializadas:

- TWI: A4 or SDA pin and A5 or SCL pin. **Soporta comunicacion TWI usando “Wire library”.**

Par de pines de la placa:

- **AREF.** Voltaje de referencia para entradas analogicas. Usando “analogReference()”.
- **Reset.** Nivel bajo para este pin, resetea al microcontrolador.



**Fig. 2.9 Pines del Arduino Uno con Atmega 168/328 [21]**

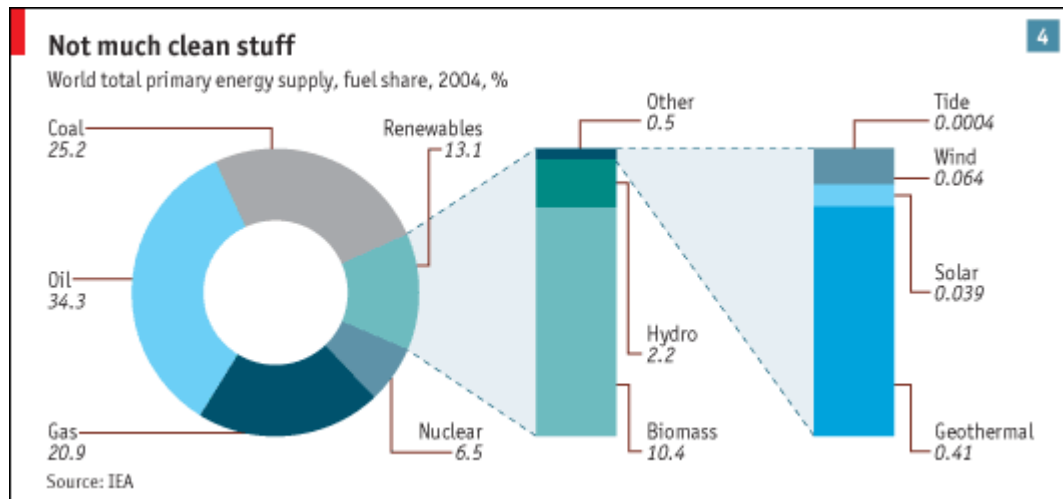
## 2.4 INTRODUCCIÓN A LAS FUENTES DE ENERGÍA RENOVABLE

Las fuentes de energías renovables han sido aprovechadas por el hombre desde hace mucho tiempo, acompañadas de la energía animal y su empleo continuo durante toda la historia hasta la llegada de la “Revolución Industrial”, en la que la aparición del carbón, con su densidad energética muy superior a la de la biomasa y su menor precio, desplazó a esta. Posteriormente, el petróleo fue desplazando en muchas aplicaciones al carbón debido a su mayor limpieza, mayor poder calorífico y su carácter fluido. Durante los últimos años, pensando en el futuro agotamiento de las

fuentes de energías fósiles, la gran dependencia de muchos países de estas, el incremento progresivo de su coste y en los problemas medioambientales derivados de su explotación, se está produciendo un renacer de las energías renovables.

Las energías renovables son aquellas que se producen de manera continua y son inagotables en escala humana. Además tiene la ventaja adicional de poder complementarse entre sí, favoreciendo la integración entre ellas. Estas se obtienen a partir de fuentes o recursos energéticos como pueden ser el sol, el viento o el agua. Son respetuosas con el medio ambiente y su impacto es mucho menor que el de las energías convencionales como combustibles fósiles (petróleo, gas y carbón), energía nuclear, etc.

Las fuentes de energías como el sol, el viento, el agua, el carbón, el petróleo o el gas se denominan energías primarias, ya que son recursos naturales que no se han sometido a ningún proceso de transformación. En la Figura 2.10 podemos observar un diagrama de la revista *The Economist* sobre el consumo mundial de energía fósil en el año 2004, en términos de energía primaria.



**Fig. 2.10 Flujo de consumo mundial anual de energía primaria. [22]**

Del total de la producción mundial de energía en un año, el petróleo tiene el: 34,3%; carbón: 25,2%; gas: 20,9%; renovables: 13,1% y nuclear: 6,5%. Este 13.1% representado por las energías renovables en el consumo mundial de energía, puede ser desglosado: biomasa: 10,4%; hidroeléctrica: 2,2% y otras energías: 0,5%. Finalmente, centrémonos en este 0,5% de la producción energética mundial representada por “otras energías”: geotérmica: 0,41%; eólica: 0,064%; solar: 0,039% y mareas: 0,0004%. Podemos observar en el diagrama el minúsculo peso relativo que constituye la tecnología solar en la producción energética mundial. No obstante, las energías solar y eólica ya juegan un papel muy importante en la generación eléctrica de varios países industrializados, sobre todo en Europa. Alrededor

del 20% de la electricidad consumida por Dinamarca procede de sus instalaciones eólicas, campo en el que España también ocupa un lugar predominante. Mientras tanto, en China, alrededor del 80% del agua sanitaria es calentada con energía solar. [22]

La demanda de consumo de energía a nivel mundial es cada vez mayor, sobre todo desde los últimos siglos y se espera que crezca aún más en el futuro. Hay dos fuentes principales de aumento de la demanda de energía a nivel mundial: el crecimiento de la población y el crecimiento tecno-económico de los países, particularmente los países en desarrollo. Como ambos crecen la demanda de energía crece proporcionalmente. Es aquí donde se hace fundamental buscar fuentes de energía alternativas para cubrir la demanda de consumo a nivel mundial, siendo las energías renovables una de las mejores opciones para cubrir dicha demanda.

Las energías renovables son parte de la energía que el sol aporta a la Tierra en cada momento. El Sol es la fuente de energía de la Tierra. Se recibe en forma de radiación que retiene la atmosfera y permite que la tierra mantenga una temperatura más o menos constante posibilitando que haya vida. La radiación solar además de proporcionar luz, también se transforma en

biomasa por medio del efecto de la fotosíntesis, en viento por los gradientes térmicos que se producen en la atmosfera o en energía hidráulica por la evaporación de los mares.

Dentro del marco de las energías renovables se pueden destacar las que tienen mayor desarrollo tecnológico y por lo tanto mayores posibilidades de competir dentro del mercado. El Sol se presenta en todas ellas. Estas energías renovables tiene la siguiente clasificación: Energía Eólica, Energía Hidráulica, Energía de Biomasa, Energía Geotérmica y Energía Solar. Siendo esta última, la Energía Solar, la utilizada como fuente de energía para el desarrollo de este proyecto. **[23]**

En el Ecuador se ha tomado la iniciativa de invertir en fuentes de energía renovable con el objetivo de impulsar las capacidades socio-económicas del sector rural y urbanos-marginal. Constituyéndose entonces la energía eléctrica, como una de las herramientas que requiere la población para el desarrollo de sus actividades productivas y mejoramiento de su calidad de vida, atendiendo sus necesidades de comunicación, alumbrado y principalmente para el desarrollo de sus actividades agropecuarias, artesanales, comerciales e industriales. Las soluciones basadas en

extensión de red eléctrica, se han ido agotando al tornarse cada vez más costosas y presentan graves efectos en el deterioro de la calidad del servicio. Es indudable por otro lado, que existen zonas aún no servidas, ubicadas especialmente en la Amazonía e incluso en la Provincia de Pichincha, en las que no resulta pertinente llegar con extensión de red atendiendo a razones de carácter económico, técnico, de impacto ambiental y sobre todo de respeto a la cultura de las etnias locales. [24]

## **2.5 ENERGÍA SOLAR**

En el interior del Sol, como en todas las estrellas, se llevan a cabo reacciones de fusión nuclear. En este tipo de reacciones se unen los núcleos de átomos ligeros, como el hidrogeno y el helio, para formar átomos más pesados y en el proceso se liberan grandes cantidades de energía; la energía que nos envía el Sol es, por lo tanto, de origen nuclear. Durante las reacciones nucleares, parte de la masa de las partículas que intervienen se convierte en energía, la cual se puede calcular empleando la fórmula de Einstein  $E=mc^2$  (donde E equivale a la energía, m a la masa y c a la velocidad de la luz, que es igual a  $3 \times 10^8$  m/s).

De esta forma, el Sol irradia la energía proveniente de la fusión de los núcleos atómicos que lo componen y como lo hace en todas direcciones, una parte nos llega a la Tierra. El Sol existe desde hace 4600 millones de

años y se cree que seguirá viviendo durante un periodo similar; por lo tanto, para cualquier fin práctico, el Sol es una fuente inagotable de energía. [25]

El aprovechamiento de la Energía Solar se la puede hacer de diferentes formas: Directa, Térmica y Fotovoltaica.

- Directa: Una de las aplicaciones de la energía solar es directamente como luz solar, por ejemplo, para la iluminación de recintos. En este sentido, cualquier ventana es un colector solar. Otra aplicación directa, muy común, es el sacado de algunos productos en procesos de producción con tecnología simple.
- Térmica: Se denomina “térmica” la energía solar cuyo aprovechamiento se logra por medio del calentamiento de algún medio. La climatización de viviendas, calefacción, refrigeración, secado, etc., son aplicaciones térmicas. La mayoría de este curso se centra en este tipo de aprovechamiento de la energía solar.
- Fotovoltaica: Se llama “fotovoltaica” la energía solar aprovechada por medio de celdas fotoeléctricas, capaces de convertir la luz en un potencial eléctrico, sin pasar por un efecto térmico.



## 2.6 DISTRIBUCION ESPECTRAL DE LA RADIACIÓN SOLAR

El Sol emite radiación en toda la gama del espectro electromagnético, desde los rayos gamma, hasta las ondas de radio. Sin embargo, para los fines de aprovechamiento de su energía, solo es importante la “llamada radiación térmica” que incluye solo el ultravioleta (UV), la radiación visible (VIS) y la infrarroja (IR). Todos los cuerpos emiten cierta cantidad de radiación en virtud de su temperatura. A mayor temperatura ocurren dos cambios en la radiación emitida:

- La intensidad de la emisión es mayor, refiriéndose a, un mayor número de watts por metro cuadrado abandonan el cuerpo.
- El color o tipo de radiación hacia una menor longitud de onda, esto es, del IR al VIS y al UV, a medida que aumenta la temperatura.

En la Figura 2.11 pueden apreciarse las diferentes formas de radiación electromagnética, que dependen de la cantidad de energía que ésta posea. Para nosotros la más común es la luz visible, pero también los rayos X o los rayos infrarrojos constituyen otras formas de radiación electromagnética. [24]

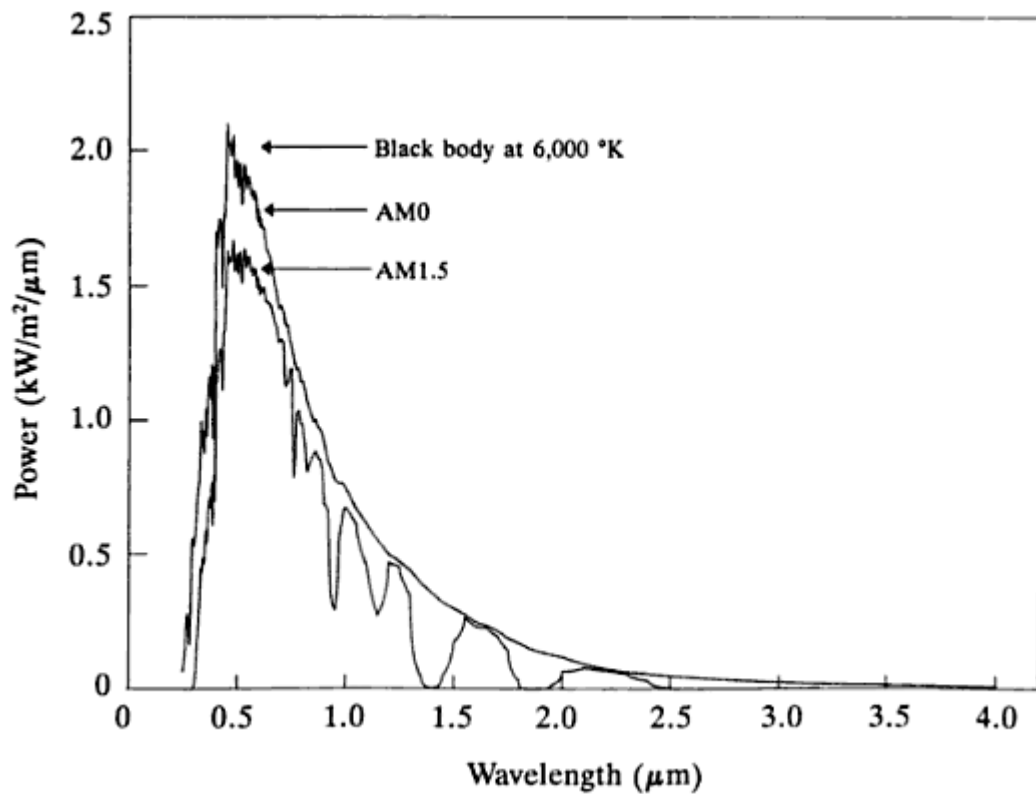
	$\nu$	$\lambda$	E	
	$10^{22}$	$10^{-13}$	$10^8$	RAYOS GAMMA
	$10^{21}$		$10^7$	
			$10^6 \approx 1 \text{ MeV}$	RAYO X
	$10^{18}$	$10^{-10} \text{ 1A}^{\circ}$	$10^5$	
		$10^{-9} \text{ 1nm}$	$10^4$	
			$10^3 \approx 1 \text{ kev}$	ULTRAVIOLETA
	$10^{15}$		$10^2$	
			1 ev	VISIBLE
			$10^{-1}$	INFRARROJO
	$10^{12}$		$10^{-2}$	
1 THZ (terahertz)		$10^{-3} \approx 1 \text{ mm}$	$10^{-3}$	
		$10^{-2} \approx 1 \text{ cm}$	$10^{-4}$	MICROONDAS
	$10^9$		$10^{-5}$	
		$10^0 \approx 1 \text{ m}$	$10^{-6}$	Radar
			$10^{-7}$	
			$10^{-8}$	VHF (frecuencia ultra alta) VHF(Tv) FM (frecuencia muy alta)
1 MHZ (megahertz)	$10^6$	$10^{-3} \approx 1 \text{ km}$	$10^{-9}$	
			$10^{-10}$	Transmisión de radio
			$10^{-11}$	RADIOFRECUENCIA
1 KHZ (kilohertz)	$10^3$	$10^0 \approx 1 \text{ mm}$	$10^{-12}$	
			$10^{-13}$	
			$10^{-14}$	
1 HZ	$10^0$		$10^{-15}$	

Frecuencia (Hertz)      Longitud de onda (metros)      Energía (electrón-volts)      Radiación

**Fig. 2.11 Espectro Electromagnético. [25]**

La fotosfera es la zona visible donde se emite luz visible del Sol, la cual se encuentra a unos  $6000^{\circ}$  K aproximadamente y, por tanto, emite un cierto

flujo de energía correspondiente a esta temperatura. Su distribución espectral es como se indica en la Figura 2.12. Esta figura muestra la irradiación espectral (energía por unidad de tiempo, en la unidad de área, por unidad de longitud de onda) en función de la longitud de onda.



**Fig. 2.12 Composición Espectral de la Radiación Solar. [26]**

La línea continua pero irregular corresponde a la irradiancia observada, medida desde la Tierra. La curva punteada representa la irradiación que tiene un cuerpo negro, que se encontraba a la temperatura de 5800 K. Esta

temperatura corresponde a la que debería tener dicho cuerpo, para tener la misma emisión de energía que el Sol. [26]

## **2.7 ENERGÍA SOLAR FOTOVOLTAICA**

La Energía Solar Fotovoltaica permite transformar en electricidad la radiación solar por medio de celdas FV, que en conjunto conforman los paneles FV. Estas celdas FV están compuestas de un material que presenta efecto fotoeléctrico: se excitan al recibir la radiación solar en forma de fotones y generan una diferencia de potencial entre sus terminales. Cada célula FV genera una diferencia de potencial tan pequeña que para conseguir tensiones de varios voltios hay que conectar en serie muchos de estos celdas.

La electricidad que genera un conjunto de celdas FV (paneles solares) puede ser utilizada de manera directa, almacenarlas en acumuladores para su uso posterior o incluso se puede introducir en la red de distribución eléctrica por medio de inversores.

La Energía Solar Fotovoltaica, era al menos 50 veces más cara que cualquier otra forma convencional de energía lo que, naturalmente,

mantenía a las células fotovoltaicas totalmente alejadas del mercado terrestre. En esos años, la Energía Solar Fotovoltaica estaba fuertemente ligada a la carrera espacial, siendo el Vanguard I, lanzado al espacio el 17 de marzo de 1958, el primer satélite que utilizaba baterías de mercurio como fuente de alimentación y 6 celdas solares desarrolladas por los Laboratorios Bell. [27]

## **2.8 EFECTO FOTOELÉCTRICO**

Fotovoltaica es la conversión directa de la luz en la electricidad a nivel atómico. Algunos materiales presentan una propiedad conocida como efecto fotoeléctrico descrito anteriormente. Cuando estos electrones libres son capturados, el resultado es una corriente eléctrica que puede ser utilizada como electricidad.

Los fotones pueden tener diferentes longitudes de onda, y por tanto, diferente energía. Solamente los fotones que son absorbidos por la célula fotovoltaica generan electricidad al transferir su energía a un electrón de átomo de la célula. Con la energía aportada, el electrón es capaz de abandonar su posición en el átomo e iniciar un movimiento formando parte de una corriente eléctrica.

El primero en notar el efecto fotoeléctrico, mientras efectuaba experimentos con una pila electrolítica de dos electrodos sumergidos en una sustancia electrolítica, fue el físico francés Edmond Becquerel, en 1839. Él encontró que ciertos materiales producían pequeñas cantidades de corriente eléctrica cuando eran expuestos a la luz. En 1905, Albert Einstein describió la naturaleza de la luz y el efecto fotoeléctrico, en el cual está basada la tecnología fotovoltaica. Por este trabajo, se le otorgó más tarde el premio Nobel de física. El primer módulo fotovoltaico fue construido en los Laboratorios Telefónicos Bell en 1954 por los investigadores G. Pearson, D. Chapin y C. Fuller, los cuales utilizaron silicio con impurezas en lugar de selenio para fabricar las celdas solares como una solución para tener una fuente de energía en las instalaciones telefónicas de las áreas rurales. Así surgieron las primeras celdas solares que tenían una eficiencia del 6% y resultaban demasiado costosas como para justificar su utilización a gran escala. Por esto, cuando se empezaban a olvidar las celdas solares, la NASA en la década de los 60s, comenzó por primera vez hacer uso de esta tecnología para proveer la energía eléctrica a bordo de las naves espaciales y dedicó mucho tiempo y dinero para perfeccionarlas y producirlas. **[28]**

## 2.9 CELDAS FOTOVOLTAICAS

La célula fotovoltaica es un semiconductor formado por una fina capa de un material semiconductor, generalmente silicio, con un tratamiento especial.

La célula fotoeléctrica es un verdadero generador eléctrico, muy pequeño, que proporciona una tensión en torno a 0,5 V y una pequeña corriente eléctrica.

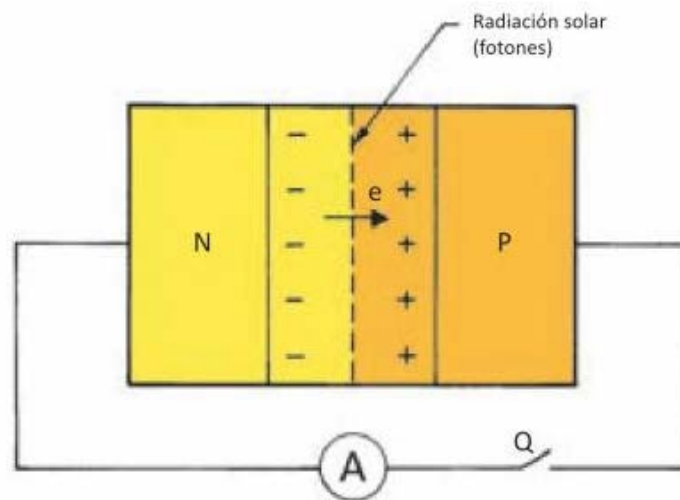
El efecto fotovoltaico es una propiedad que tienen ciertos semiconductores contruidos a base de silicio, para transformar la luz solar (fotones), en electricidad.

El elemento fotovoltaico (célula solar fotovoltaica) está constituido por dos materiales (P y N) de diferente conductividad, que son: Zona P y Zona N.

Zona P: Las impurezas de boro en el silicio generan la zona P, que es positiva por tener un electrón menos que el silicio.

Zona N: Las impurezas de fósforo en el silicio generan la zona N, que es negativa por tener un electrón más que el silicio.

Al cerrar el circuito estando sometido a la radiación solar, se crea una corriente desde N hacia P, tal como se indica en la Figura 2.13.



**Fig. 2.13 Efecto Fotoeléctrico. [28]**

## 2.10 TIPOS DE CELDAS FOTOVOLTAICAS

Existen tres tipos de celdas; se las diferencia por medio del método de fabricación y su coloración.

### 2.10.1 MATERIALES DENSAMENTE CRISTALINOS

Las celdas de silicio monocristalinas se obtienen a partir de barras cilíndricas de silicio monocristalino producidas en hornos especiales. Las celdas se obtienen por corte de las barras en forma de obleas cuadradas



delgadas (0,4 – 0,5 mm de espesor). Este elaborado proceso de manufactura, que insume enormes cantidades de energía eléctrica, incrementando substancialmente el costo del material semiconductor.

La versión policristalina se obtiene fundiendo el material semiconductor, el que es vertido en moldes rectangulares. En los moldes, el silicio se enfría lentamente, solidificándose. Su estructura cristalina no es uniforme, de ahí el nombre de poli (muchos) y cristalino (cristales). Estas celdas son menos caras de fabricar y su eficiencia en conversión de luz solar en electricidad es algo menor a las de silicio monocristalino.

Los dos tipos pueden ser identificados a simple vista, ya que la estructura monocristalina provee un color azul oscuro uniforme, mientras que la policristalina muestra una coloración azul más intensa. La Figura 2.14 y 2.15 muestran esta diferencia. **[29]**



**Fig. 2.14 Material Cristalino [29]**



**Fig. 2.15 Material Policristalino [29]**

### 2.10.2 MATERIALES DE PELÍCULA DELGADA

Algunos paneles celdas FV, como los que se muestran en la Figura 2.16, se obtienen mediante la deposición de capas muy delgadas de silicio sobre superficies de vidrio o metal. La superficie activa de estas celdas no tiene una estructura cristalina, y por ello se la denomina amorfa (a = sin; morfos = forma). La ausencia de una estructura cristalina aumenta la posibilidad de que una carga libre sea atrapada, lo que se traduce en una menor eficiencia de conversión. Este proceso permite la fabricación de un panel FV flexible, donde puede ser utilizada para aplicaciones donde la flexibilidad es más importante que el rendimiento. Este tipo celdas amorfas es, por ejemplo, el

que se emplea en aplicaciones solares como relojes o calculadoras. La celda amorfa tiene un color homogéneo marrón, y la constitución de un panel FV con celdas amorfas no presenta conexión visible entre sí. [30]



**Fig. 2.16 Material Amorfo. [30]**

## 2.11 EFICIENCIA DE CONVERSIÓN

La eficiencia de conversión, en un panel FV, es la relación entre la energía eléctrica generada y la energía luminosa utilizada para obtenerla. Esta relación es dada en forma porcentual, como se muestra a continuación:

$$\eta\% = \frac{\text{Energía Generada}}{\text{Energía Incidente}} \times 100 \quad (2.1)$$

El símbolo  $\eta$  es la letra griega eta, que es usualmente utilizada para expresar eficiencias. Los porcentajes de eficiencia o rendimiento de los diferentes tipos de paneles FV se describen a continuación:

Panel de silicio monocristalino: Eficiencia de un 12% - 15%.

Panel de silicio policristalino: Eficiencia de un 11% - 13% .

Panel de silicio amorfo: Eficiencia de un 5% - 6%.

Como se puede observar los paneles FV que utilizan semiconductores monocristalinos tienen una eficiencia mayor a las que utilizan el semiconductor policristalino, debido a que las imperfecciones en la estructura de este último disminuyen el número de pares de carga que quedan libres para conducir la corriente. Las celdas FV usadas en los satélites espaciales utilizan, exclusivamente, semiconductores monocristalinos, ya que el costo no es un factor en consideración. [29]

## **2.12 CURVAS CARACTERISTICAS DE CELDAS FOTOVOLTAICAS**

### **2.12.1 CURVA DE CORRIENTE VS TENSION**

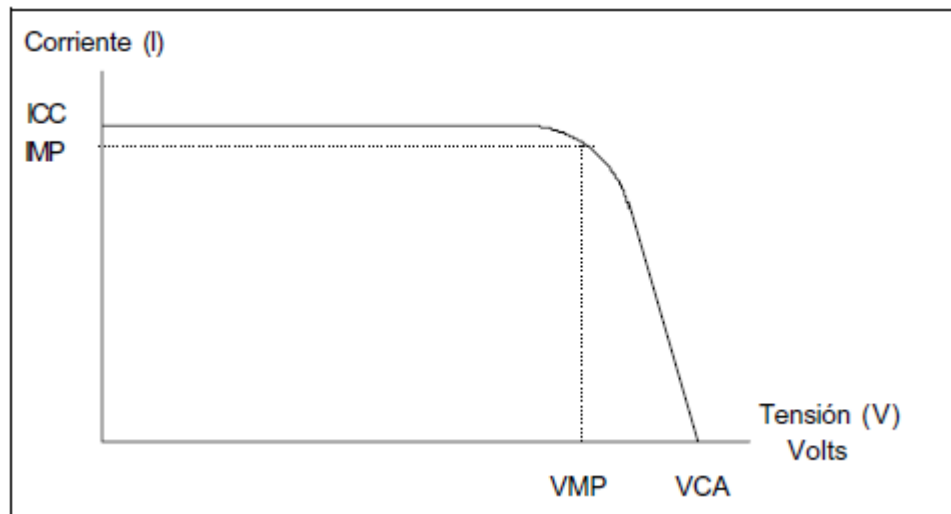
En esta se podrá observar el comportamiento, a la salida del panel FV, tanto del voltaje como la corriente.

La representación típica de la característica de salida de un dispositivo FV (celda, modulo, sistema) se denomina curva de corriente vs tensión (curva I-V). La corriente de salida se mantiene prácticamente constante dentro del rango de tensión de operación y, por lo tanto, el dispositivo se puede considerar como una fuente de corriente constante en este rango como se muestra en la Figura 2.17.

Ambos parámetros de operación (I,V), están determinados por la radiación solar incidente, por la temperatura ambiente, y por las características de la carga conectada al mismo. La siguiente ecuación representa todos los pares de valores (I-V: corriente – voltaje) en que puede trabajar una celda fotovoltaica.

$$I = I_{CC} \left( 1 - e^{\frac{-e(V_{CA} - V)}{mKT}} \right) \quad (2.2)$$

La curva característica I-V de una celda FV tendrá la siguiente forma como lo muestra la Figura 2.17.



**Fig. 2.17 Curva característica I-V de una celda FV. [31]**

Donde, los valores trascendentales de esta curva son:

- La corriente de cortocircuito (ICC): el valor de este parámetro se obtiene cuando el voltaje de salida es nulo, lo que indica que los dos terminales del panel están cortocircuitados. En esta situación, la potencia de salida del panel es nula, lo que indica que un cortocircuito no dañara la estructura interna del panel FV.
- Tensión de circuito abierto (VCA): es el máximo voltaje que se puede obtener en el panel. En este caso, la corriente es nula, puesto que no hay cargas conectadas. Al igual que en el caso anterior, la potencia de salida es nula.

- Potencia pico (PMP): es el máximo valor de potencia que puede entregar el dispositivo. Corresponde al punto de la curva en el cual el producto (V.I) es máximo.
- Corriente a máxima potencia (IMP): es la corriente que entrega el dispositivo a potencia máxima bajo condiciones determinadas de radiación y temperatura. Se utiliza como corriente nominal del mismo.
- Tensión a máxima potencia (VMP): es la tensión que entrega el dispositivo a potencia máxima bajo condiciones determinadas de radiación y temperatura. Se utiliza como tensión nominal del mismo.

Dónde:

$e$  : es la carga del electrón.

$m$  : es un parámetro constructivo de la celda, normalmente=1.

$K$  : es la constante de Boltzman.

$T$  : temperatura en °K. **[31]**

## **2.13 FACTORES AMBIENTALES SOBRE LA CARACTERISTICA DE SALIDA DEL DISPOSITIVO**

El desempeño de los paneles fotovoltaicos depende, en gran medida, de los factores ambientales. A continuación se analizará el comportamiento del panel solar ante estos factores.

### **2.13.1 EFECTO DE LA INTENSIDAD DE LA RADIACIÓN SOLAR**

La variación en la corriente de salida para cualquier valor de tensión, se produce debido al cambio en la intensidad de radiación.

La corriente de salida varía con la radiación en forma directamente proporcional. La tensión se mantiene prácticamente constante como lo muestra la Figura 2.18.

### **2.13.2 EFECTO DE LA TEMPERATURA**

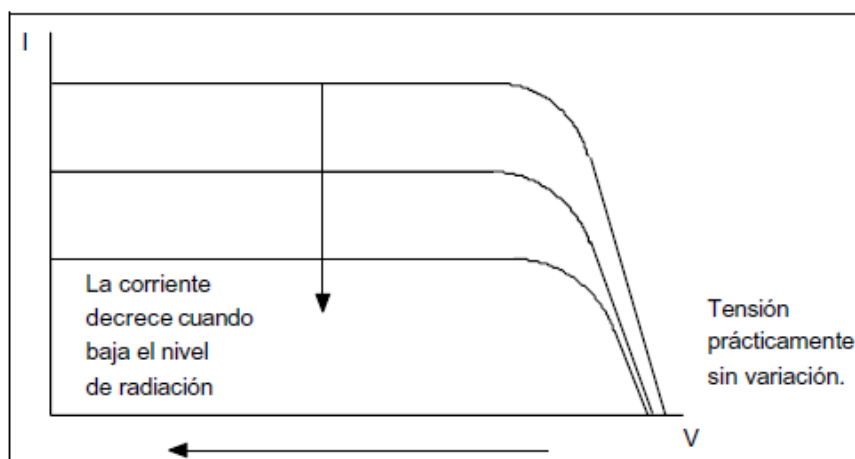
El principal efecto provocado por el aumento de la temperatura del módulo, es una reducción de la tensión en forma directamente proporcional.

Es por ello que para lugares donde las temperaturas ambientales son muy altas son aptos módulos que poseen mayor cantidad de celdas en serie para



que los mismos tengan la suficiente tensión de salida para cargar las baterías.

La tensión en el punto máximo de potencia de salida para una celda es de aproximadamente 0,5 voltios a pleno sol. La corriente que entrega una celda es proporcional a la superficie de la misma y a la intensidad de la luz. Es por ello que para lograr módulos con corriente de salida menores se utilizan en su fabricación tercios, cuartos, medios, etc de celdas solares.

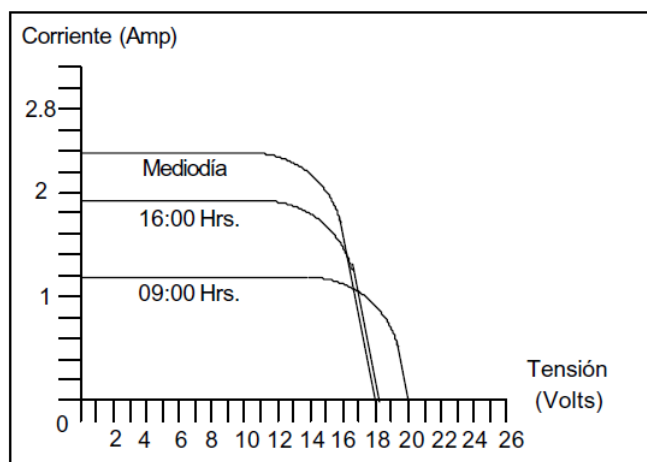


**Fig. 2.18 Efecto de la intensidad de radiación. [31]**

## 2.14 INTERACCIÓN DEL DISPOSITIVO FOTOVOLTAICO CON LA CARGA

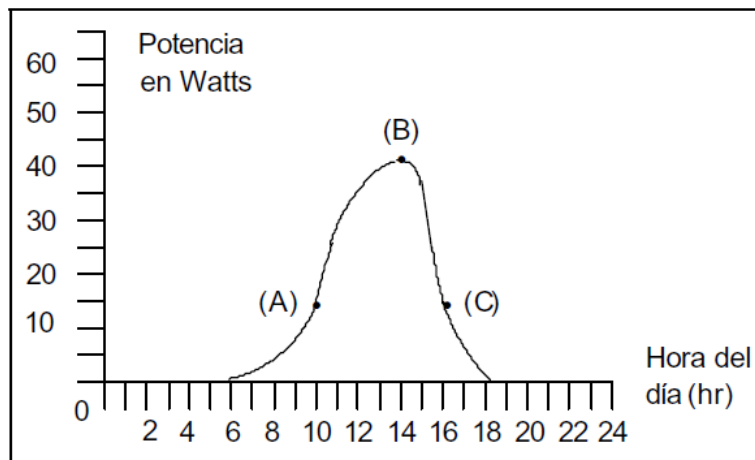
La curva I-V para las condiciones ambientales reinantes, es sólo parte de la información necesaria para saber cuál será la característica de salida de un módulo.

La otra información imprescindible es la característica operativa de la carga a conectar. Es la carga la que determina el punto de trabajo en la curva I-V. La curva I-V del módulo varía con las condiciones ambientales (radiación, temperatura), por lo tanto habrá una familia de curvas I-V que nos mostrarán las características de salida del módulo durante el día y una época del año, como se muestra en la Figura 2.19.



**Fig. 2.19 Potencia máxima de salida durante el día. [31]**

La curva de potencia máxima de un módulo en función de la hora del día tiene la forma indicada en la Figura 2.20.



**Fig. 2.20 Potencia máxima de un módulo en función de la hora del día. [31]**

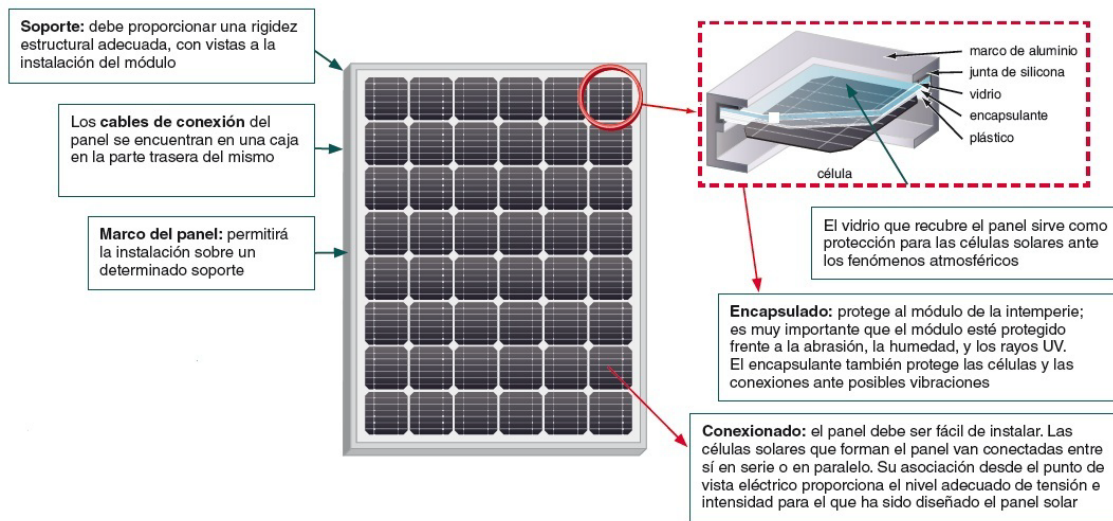
La cantidad de energía que el módulo es capaz de entregar durante el día está representada por el área comprendida bajo la curva de la figura 2.12 y se mide en Watts hora /día. Se observa que no es posible hablar de un valor constante de energía entregada por el módulo en Watts hora ya que varía dependiendo de la hora del día.

Será necesario entonces trabajar con valores de cantidad de energía diarios entregados (Watts hora/día). [31]

## 2.15 CONSTITUCIÓN DE UN MÓDULO FOTOVOLTAICO

Una célula solar aislada proporciona una potencia reducida; con el fin de conseguir potenciales útiles para aparatos de media potencia, es necesario unir cierto número de células en serie. Dichas uniones de células fotovoltaicas constituyen un bloque llamado panel o módulo fotovoltaico. Estos módulos suelen estar conectados eléctricamente, encapsulados, y montados sobre una estructura de soporte o marco. Posteriormente estos módulos se conectan entre sí en serie y/o en paralelo con el fin de obtener la tensión deseada (6 V, 12 V, 24 V).

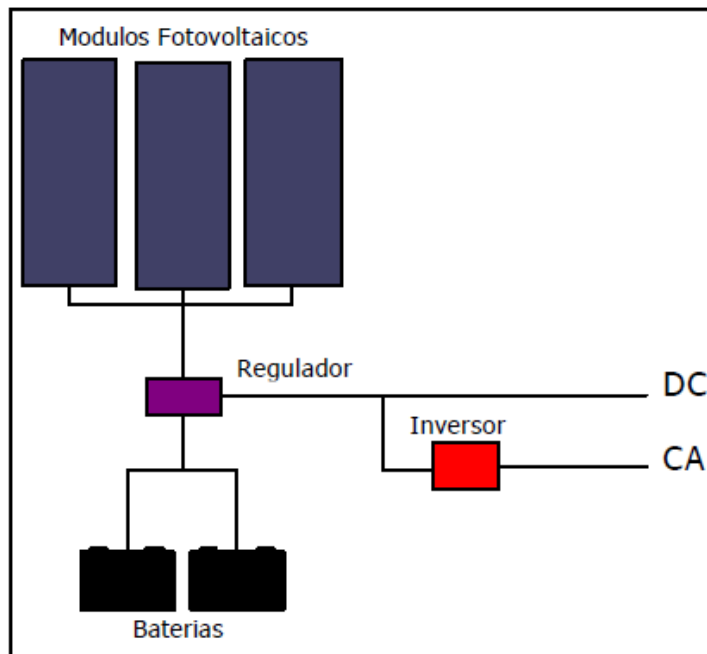
En la Figura 2.21 se muestra las principales características de un panel FV y un esquema típico de su construcción. **[32]**



**Fig. 2.21 Constitución de un módulo fotovoltaico. [32]**

## 2.16 SISTEMA FOTOVOLTAICO

Los sistemas fotovoltaicos SFV transforman la radiación solar en energía eléctrica, a través de unos conjuntos de celdas solares, de esta forma se logra abastecer una amplia variedad de consumos. La energía excedente producida en las horas y días de mayor radiación, es almacenada en bancos de baterías, de esta forma se logra abastecer el consumo en los días nublados y durante la noche. La Figura 2.22 muestra un esquema de un SFV que entrega tensión continua y alterna.



**Fig. 2.22 Esquema de un SFV que entrega tensión continua y alterna. [31]**

Los SFV generan energía eléctrica en corriente continua, sin embargo si el consumo es en corriente alterna será necesario intercalar un inversor DC/AC entre las baterías y dichos consumos, como se observa en la Figura 2.22.

La capacidad de un SFV dependerá del recurso solar disponible en el lugar de instalación y del tamaño de los paneles. Los SFV se diseñan de tal manera que la energía que entregue sea equivalente al requerido por los

consumos conectados, en la mayoría de los casos el cálculo se lo realiza bajo las peores condiciones (invierno).

### **2.16.1 PRINCIPALES VENTAJAS**

Las principales ventajas que presenta un SFV son las siguientes:

- No consumen combustible
- No contaminan el medio ambiente
- Son totalmente silenciosos
- Requieren un mínimo de mantenimiento
- Sistemas de bajo costo

### **2.16.2 PRINCIPALES APLICACIONES**

- En las zonas rurales y viviendas aisladas en las cuales tener una red eléctrica sería sumamente costoso, los sistemas fotovoltaicos son una alternativa rentable.
- En las comunicaciones son muy comunes la utilización de los sistemas fotovoltaico debido a su capacidad de abastecer a equipos de

comunicación que se encuentran aislados, tales como: repetidoras de señales de TV, radioenlaces, estaciones meteorológicas, etc.

- En la navegación los SFV son usados como fuente de energía para alimentar faros, boyas balizas, plataformas y embarcaciones.
- En el transporte terrestre usados para la alimentación de radioteléfonos o puestos de socorro alejados de la red eléctrica convencional. Señalización de pasos a desnivel o cambios de via en los ferrocarriles.
- En la agricultura utilizada en su mayoría en los sistemas de bombeo de agua para riego, ya que este sector se encuentra alejado de la red eléctrica.

### **2.16.3 REGULADOR DE CARGA**

Para un correcto funcionamiento del sistema fotovoltaico, hay que instalar un sistema de regulación de carga en la unión entre los paneles solares y las baterías. Este elemento tiene como misión evitar situaciones de sobredescarga de la batería, dado que no se puede almacenar toda la energía eléctrica producida por los módulos fotovoltaicos, y así alargar su vida útil.



Las principales características de los reguladores de carga son mencionadas a continuación:

- Mantiene cargada la batería del sistema.
- Posee un controlador del sistema el cual, de manera automática, mantiene cargada la batería y cuando esta se encuentra cargada, la desconecta de los paneles para evitar una sobrecarga.
- Este controlador nos asegura que la utilización de la batería sea eficiente, logrando así alargar su vida útil.
- Ajuste de voltaje de salida
- Evita que exista posibles descargas por la noche a través de los paneles solares.

#### **2.16.4 ACUMULADORES O BATERIAS**

La llegada de la energía solar a los módulos fotovoltaicos no se produce de manera uniforme, sino que presenta variaciones por diferentes motivos. Algunas de estas variaciones son predecibles, como la duración de la noche o las estaciones del año, pero existen otras muchas causas que pueden

producir alteraciones de manera aleatoria en la energía recibida, como puede ocurrir con un aumento de la nubosidad en un determinado instante.

Este hecho hace necesario utilizar algún sistema de almacenamiento de energía para aquellos momentos en que la radiación recibida sobre el generador fotovoltaico no sea capaz de hacer que la instalación funcione en los valores diseñados. Para ello se utilizara las baterías o acumuladores.

Uno de los parámetros más importantes que debemos de tener en cuenta a la hora de elegir un acumulador es la capacidad. La cual se define como la cantidad de electricidad que puede lograrse en una descarga completa del acumulador partiendo de un estado de carga total del mismo. Se mide en amperios hora (Ah), y se calcula como el producto de la intensidad de descarga del acumulador durante el tiempo en el que está actuando.

La batería cumple con dos misiones importantes. La primera misión es suministrar niveles de potencia, durante breves momentos, mayores a los entregados por los paneles en las condiciones más desfavorables. La segunda misión es mantener un nivel de tensión estable independiente del nivel de incidencia que tengas los paneles solares.

Las baterías se pueden conectar en serie si lo que quiere es tener más voltaje o en paralelo si lo que se requiere es más corriente. Las baterías más usadas son las de plomo-ácido y las de níquel-cadmio. [33]

#### **2.16.4.1 BATERIAS PLOMO - ÁCIDO**

Las baterías más comunes en las aplicaciones fotovoltaicas son las denominadas baterías de plomo ácido, están compuestas de placas de dióxido de plomo, placas negativas de plomo y como electrolito ácido sulfúrico diluido.

Las baterías de plomo ácido están clasificadas de acuerdo a su uso. Las principales baterías de plomo ácido a ser consideradas en sistemas fotovoltaicos son las siguientes:

- Baterías de arranque de motores.
- Baterías de tracción.
- Baterías estacionarias o de reserva.
- Baterías selladas o regulada por válvula.
- Baterías abiertas o ventiladas.

Siendo la batería estacionaria o de reserva la usada en este proyecto. Este tipo de batería está diseñada para sistemas de alarma de incendios. Alumbrado de emergencia, sistemas de alimentación ininterrumpida (o UPS) y telecomunicaciones, entre otros. Las baterías estacionarias están constantemente siendo cargadas para compensar la pérdida de capacidad debido a la autodescarga, y están construidas para resistir descargas profundas esporádicas. [33]

#### **2.16.4.2 NIQUEL - CADMIO**

Las principales características de estas baterías se detallan a continuación:

- El electrolito es alcalino.
- Admiten descargas profundas de hasta el 90% de su capacidad nominal.
- Bajo coeficiente de autodescarga.
- Alto rendimiento ante variaciones extremas de temperatura.
- La tensión nominal por elemento es de 1,2 voltios.
- Alto rendimiento de absorción de carga.

- Muy alto costo comparado con las baterías ácidas.
- Elevado rendimiento en amperios – hora.

#### **2.16.4.3 SELECCIÓN DE TIPO DE BATERÍA**

La selección del tipo de batería dependerá netamente del consumo a abastecer, para sistemas fotovoltaicos pequeños se puede hacer uso de baterías ácidas, donde no se requiere que la batería esté sometida a ciclos constantes de carga o descarga profunda. Desde el punto de vista económico es más barato usar una batería de plomo – ácido que una batería de níquel – cadmio. Es por eso, que proyectos orientados a la alimentación de sistemas pequeños como por ejemplo estaciones meteorológicas, sistema de radiofrecuencia en embarcaciones entre otros, las baterías de plomo – ácido son comúnmente usadas por su bajo costo y bajo nivel de mantenimiento. [33]

#### **2.16.5 INVERSORES Y CONVERTIDORES**

El inversor se encarga de convertir la corriente continua de la instalación en corriente alterna, igual a la utilizada en la red eléctrica: 220V de valor eficaz y una frecuencia de 60 Hz.

Es un elemento imprescindible en las instalaciones conectadas a red, y estará presente en la mayoría de instalaciones autónomas, sobre todo en aquellas destinadas a la electrificación de viviendas. **[31]**

El convertidor tiene como función cambiar el nivel de voltaje entregado por la batería o panel solar, manteniendo dicho voltaje en corriente continua. Este voltaje convertido tendrá que ser el requerido por la carga.

## **2.17 UNIDADES UTILIZADAS EN ENERGÍA SOLAR FOTOVOLTAICA**

La insolación, la energía solar, así como muchas otras variables pueden medirse en diversos tipos de unidades. La siguiente Tabla 2.1 muestra una visión generalizada de las unidades más utilizadas en energía solar fotovoltaica y sus conversiones.

UNIDAD	EXPLICACIÓN	CONVERSIÓN
<b>Potencia Solar</b>		
Wp	Vatio pico	-
W	Vatio	-
KW	Kilovatio (1000 w)	-
W/m <sup>2</sup>	Vatio por metro cuadrado	
<b>Energía Solar</b>		<b>A kWh/m<sup>2</sup></b>
kWh/m <sup>2</sup>	kWh por metro cuadrado	1
kJ/cm <sup>2</sup>	kJ por centímetro cuadrado	2,778
MJ/m <sup>2</sup>	MJ por metro cuadrado	0,2778
kcal/cm <sup>2</sup>	1000 calorías por centímetro cuadrado	11,67
BTU/pie <sup>2</sup>	Unidades Térmicas británicas por pie cuadrado	0,0428
Langley	Caloría por centímetro cuadrado	0,0116

**Tabla 2.1 Unidades utilizadas en la energía solar fotovoltaica. [34]**

## **CAPÍTULO 3**

### **DESARROLLO DEL PROYECTO**

#### **3.1 RESUMEN**

En el presente capítulo se detalla las diferentes etapas que tiene el proyecto final, como el desarrollo y funcionamiento del código, los pasos que se siguió para el código implementado, los algoritmos basados en lenguaje de programación Arduino (basado en Wiring), y el uso correcto de los pines del



puerto del Arduino cumpliendo así finalmente los objetivos propuestos inicialmente en el desarrollo de este proyecto.

Previo a la ejecución de los ejercicios de ejemplo que ayudan a comprender al código final del proyecto, se deben conocer los programas necesarios para la compilación de los códigos de desarrollo, en este caso el programa desarrollado por Arduino de fácil entorno y carga del mismo en el Atmega 328p.

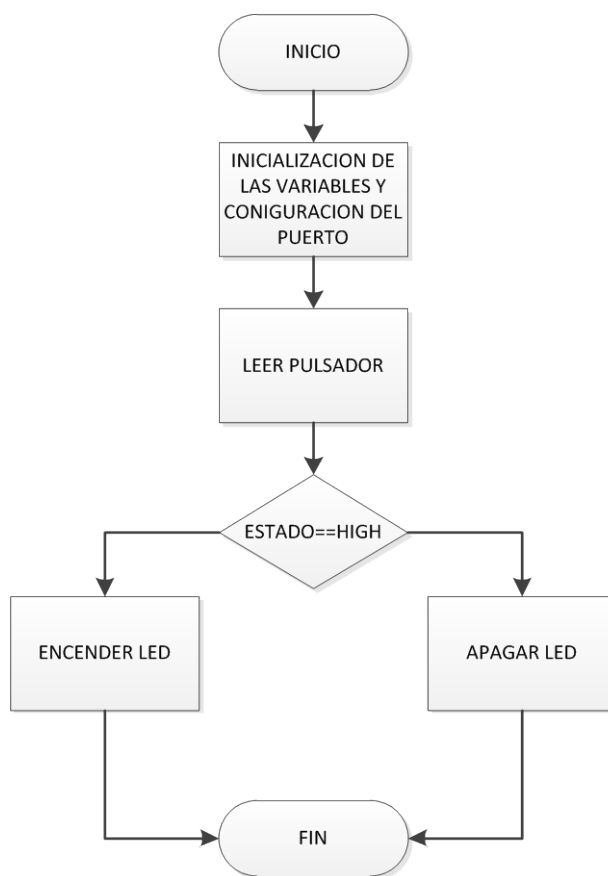
Para ejecutar el programa se lo arranca desde diferentes sistemas operativos, en este caso se lo hizo en Windows 7 con el entorno de desarrollo (Processing) de Arduino.

### **3.2 EJERCICIO 1: CÓMO TRABAJAR CON LOS PUERTOS DEL ARDUINO.**

En este primer ejercicio se va a leer un pulsador conectado al puerto del Arduino que al accionar el pulsador encenderá un led y al soltarlo se apagará, en este caso se usará las funciones básicas que tiene el lenguaje de programación Arduino que se asemeja al lenguaje C, por eso que este entorno de programación es más amigable y sencillo, estas primeras pruebas que se harán servirán para asemejarse a cómo trabaja el Arduino y posterior

a esto usar estos conceptos básicos para el desarrollo del código final. En este caso el pulsador usa lógica positiva, es decir que al presionar el pulsador dará un nivel alto (+5V) y al soltar un nivel bajo (0V).

### DIAGRAMA DE FLUJO



**Fig. 3.1 Diagrama de Flujo Ejercicio 1. [35]**

## ALGORITMO

1. Inicialmente se inicializa las variables con el tipo de dato correcto.
2. Se configura los pines a usar en este caso son dos, uno de salida para el Led y otro de entrada para leer el pulsador, todo son digitales.
3. Después se entra a un lazo en la cual se lee el pulsador, si este ha sido presionado, el Led se enciende caso contrario se mantiene apagado el Led.
4. Fin de programa.

## CÓDIGO FUENTE

```
int buttonPin = 2; //definición de variables
int ledPin = 13;
int estado = 0;

void setup() {
  pinMode(ledPin, OUTPUT); //Configuracion de los pines
  pinMode(buttonPin, INPUT);
}

void loop(){ //Programa principal
  estado = digitalRead(buttonPin);
  if (estado == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}
```

```
}  
}
```

## **CONCLUSIÓN**

Para este primer ejercicio concluimos que el entorno es similar al lenguaje C usado en microcontroladores y demás plataformas, pero en la parte de condicionar alguna variable o comparaciones como el “==” es similar al lenguaje C, este entorno de desarrollo es muy sencillo de comprenderlo y cargarlo directamente en el Atmega 328p.

### **3.3 EJERCICIO 2: CÓMO IMPRIMIR MENSAJES EN UNA LCD**

En el presente ejercicio se usara un LCD 16x2 en la cual se imprimirá un mensaje, usando la librería que viene predefinida en el programa Arduino, “Liquid Crystal” que reduce la líneas de código y el consumo de memoria del Arduino, también el uso de una función muy sencilla pero interesante como “millis()” que nos da los milisegundos que ha transcurrido desde que empezó a ejecutar el Arduino las líneas de código.

**DIAGRAMA DE FLUJO****Fig. 3.2 Diagrama de Flujo Ejercicio 2. [36]**

## ALGORITMO

- Se incluye primeramente la librería del LCD.
- Se define los pines que se conectarán en el Arduino con la LCD.
- Se define qué tipo de LCD se trabajara para asignar los valores en las funciones de configuración del LCD.
- Se imprime el valor en la LCD.
- Y por último los segundos que transcurrieron en cada leída de las líneas de código por el Arduino se imprime también, previamente fijado en que fila y columna saldrán los mensajes.

## CÓDIGO FUENTE

```
//Control de pantalla LCD

#include <LiquidCrystal.h> //Biblioteca necesaria para LCDs

//Iniciamos los pines a utilizar
LiquidCrystal lcd(12,11,5,4,3,2);

void setup() {
  //Expresamos el número de columnas y filas de nuestro LCD
  lcd.begin(16, 2);
  //Imprimimos el mensaje deseado en el LCD
  lcd.print("Comohacer.eu");
}
```

```
void loop() {  
  //Preparamos el cursor en la columna 0 de la linea 1  
  //Esto indica donde comenzara a imprimirse el texto  
  lcd.setCursor(0,1);  
  lcd print(millis()/1000); //imprime los milisegundos convertido a entero  
}
```

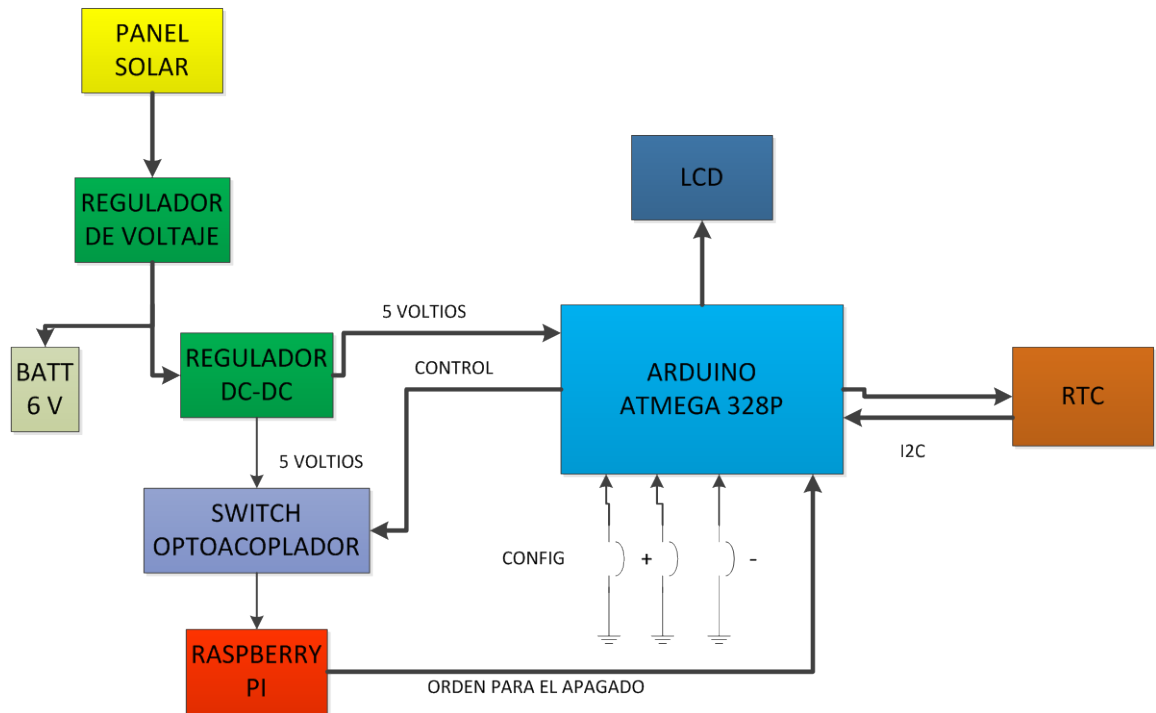
## **CONCLUSIÓN**

Este ejercicio es muy didáctico pero a la vez muy importante ya que nos muestra cómo usar las funciones predefinidas por el compilador del Arduino, la facilidad de imprimir datos al LCD con funciones sencillas y prácticas, este código nos servirán para varios proyectos como en este caso para el proyecto final como objetivo.

### **3.4 PROYECTO TERMINADO**

Teniendo en cuenta los ejercicios anteriores realizados, estos serán etapas del proyecto final, en nuestro caso es crear un control para el Raspberry Pi haciendo este que consuma menos energía a la batería que es cargada por el panel solar, porque el sistema debe ser capaz de mantenerse en funcionamiento todo el día y sea al mismo tiempo bajo consumo para el sistema, entonces con el Arduino será la parte de control, y en si será un reloj temporizado fijado previamente por el usuario dependiendo el fin que se le dará, y controlar el encendido y a apagado del Raspberry Pi.

## DIAGRAMA DE BLOQUES



**Fig. 3.3 Diagrama de Bloques del Proyecto.**



DIAGRAMA DE FLUJO

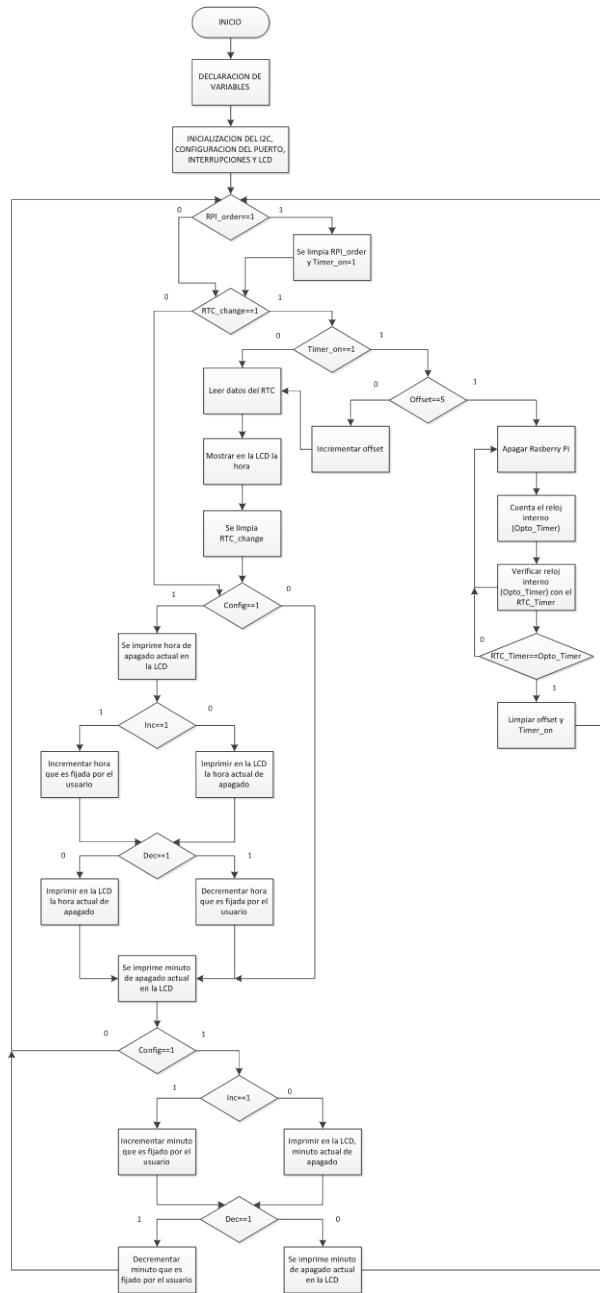


Fig. 3.4 Diagrama de Flujo del proyecto

## CÓDIGO FUENTE

```
#Include <wire.h>
```

```
Volatile int rtc_change=0;
```

```
Volatile int rpi_order=0;
```

```
Int en=13, rs=12, d7=11, d6=10, d5=9, d4=8;
```

```
Int opto=7, config=6, inc=5, dec=4;
```

```
Int timer_on=0, offset=0;
```

```
Static byte rtc_data[6]={0x23,0x15,0x00,0x07,0x10,0x14};
```

```
Static char rtc_txt[20]="00:00:00 01/01/2000";
```

```
Static char hora[17]="    hora    ";
```

```
Static char minuto[17]="    minuto  ";
```

```
Static int rtc_timer[2]={0,0};
```

```
Static int opto_timer[3]={0,0,0};
```

```
Void setup (){
```

```
Wire.begin();    // Inicia el puerto I2C

//RTC_Set2();

//RTC_Set(RTC_Data);

//Serial.begin(9600); // Inicia puerto SERIE a 9600bps

LCD_ConfigPin();

pinMode(Opto,OUTPUT);

pinMode(Config,INPUT);

pinMode(Inc,INPUT);

pinMode(Dec,INPUT);

digitalWrite(Opto,HIGH);

LCD_Ini();

LCD_Clr();

attachInterrupt(0, RTC_ISR, FALLING);

attachInterrupt(1, RPI_ISR, RISING);
```

```
}
```

```
void loop (){
```

```
  if(RPI_Order){
```

```
    RPI_Order=0;
```

```
    Timer_On=1;
```

```
  }
```

```
  if(RTC_Change){
```

```
    if(Timer_On==1){
```

```
      if(Offset==5){
```

```
        digitalWrite(Opto,LOW);
```

```
        Contar();
```

```
        Verificar();
```

```
      }
```

```
else

    Offset++;

}

RTC_Get(RTC_Data);

RTC_Text(RTC_Data,RTC_Txt);

//Serial.println(RTC_Txt);

LCD_PrintRTC(RTC_Txt);

RTC_Change=0;

}

if(digitalRead(Config)){

    detachInterrupt(0);

    LCD_PrintHora();

    Soltar(Config);

    while(!digitalRead(Config)){
```

```
if(digitalRead(Inc)){  
  
    if(RTC_Timer[0]==23)  
  
        RTC_Timer[0]=0;  
  
    else  
  
        RTC_Timer[0]++;  
  
    LCD_PrintHora();  
  
    Soltar(Inc);  
  
}  
  
if(digitalRead(Dec)){  
  
    if(RTC_Timer[0]==0)  
  
        RTC_Timer[0]=23;  
  
    else  
  
        RTC_Timer[0]--;  
  
    LCD_PrintHora();
```

```
Soltar(Dec);

}

}

LCD_PrintMinuto();

Soltar(Config);

while(!digitalRead(Config)){

    if(digitalRead(Inc)){

        if(RTC_Timer[1]==59)

            RTC_Timer[1]=0;

        else

            RTC_Timer[1]++;

        LCD_PrintMinuto();

        Soltar(Inc);

    }

}
```

```
if(digitalRead(Dec)){  
  
    if(RTC_Timer[1]==0)  
  
        RTC_Timer[1]=59;  
  
    else  
  
        RTC_Timer[1]--;  
  
    LCD_PrintMinuto();  
  
    Soltar(Dec);  
  
    }  
  
}  
  
delay(20);  
  
while(digitalRead(Config));  
  
delay(20);  
  
attachInterrupt(0, RTC_ISR, FALLING);  
  
}
```



```
}
```

```
void Soltar(int num){
```

```
    delay(20);
```

```
    while(digitalRead(num));
```

```
    delay(20);
```

```
}
```

```
void Contar(){
```

```
    Opto_Timer[2]++;
```

```
    if(Opto_Timer[2]==60){
```

```
        Opto_Timer[2]=0;
```

```
        Opto_Timer[1]++;
```

```
        if(Opto_Timer[1]==60){
```

```
    Opto_Timer[1]=0;

    Opto_Timer[0]++;

}

}

}

void Verificar(){

    if(Opto_Timer[0]==RTC_Timer[0] && Opto_Timer[1]==RTC_Timer[1]){

        digitalWrite(Opto,HIGH);

        Opto_Timer[0]=0;

        Opto_Timer[1]=0;

        Timer_On=0;

        Offset=0;

    }
```

```
}
```

```
void RTC_ISR(){
```

```
    RTC_Change=1;
```

```
}
```

```
void RPI_ISR(){
```

```
    RPI_Order=1;
```

```
}
```

```
void RTC_Set(byte Data[6]){
```

```
    Wire.beginTransmission(0x68); // Inicio de Transmision
```

```
    Wire.write(0x00);           // Direccion Inicial 00H
```

```
    Wire.write(Data[2]);       // #Segundo
```

```
Wire.write(Data[1]);    // #Minuto

Wire.write(Data[0]);    // #Hora

Wire.write(0x00);      // Dia de la Semana

Wire.write(Data[3]);    // #Dia

Wire.write(Data[4]);    // #Mes

Wire.write(Data[5]);    // 20-#Año

Wire.write(0x90);      // Oscilador Encendido 1Hz

Wire.endTransmission(); // Fin de Transmision

}

void RTC_Set2(){

Wire.beginTransmission(0x68); // Inicio de Transmision

Wire.write(0x07);        // Direccion Inicial 00H

Wire.write(0x90);        // Oscilador Encendido 1Hz
```

```
Wire.endTransmission(); // Fin de Transmision

}

void RTC_Get(byte Data[6]){

Wire.beginTransmission(0x68); // Inicio de Transmision

Wire.write(0x00); // Direccion Inicial 00H

Wire.endTransmission(); // Inicio de Transmision

Wire.requestFrom(0x68, 7); // Solicito 7 bytes

Data[2]=Wire.read()&0x7F; // Captura de byte 1 + Filtro

Data[1]=Wire.read(); // Captura de byte 2

Data[0]=Wire.read(); // Captura de byte 3

Wire.read(); // Desecho de byte 4

Data[3]=Wire.read(); // Captura de byte 5

Data[4]=Wire.read(); // Captura de byte 6
```

```
Data[5]=Wire.read();    // Captura de byte 7  
  
}
```

```
void RTC_Text(byte Data[6], char Txt[20]){
```

```
    // Transformando Horas
```

```
    Txt[0]= ((Data[0]&0xF0)>>4)+'0';
```

```
    Txt[1]= (Data[0]&0x0F)+'0';
```

```
    // Transformando Minutos
```

```
    Txt[3]= ((Data[1]&0xF0)>>4)+'0';
```

```
    Txt[4]= (Data[1]&0x0F)+'0';
```

```
    // Transformando Segundos
```

```
    Txt[6]= ((Data[2]&0x70)>>4)+'0';
```

```
    Txt[7]= (Data[2]&0x0F)+'0';
```

```
    // Transformando Dia
```

```
Txt[9]= ((Data[3]&0xF0)>>4)+'0';

Txt[10]= (Data[3]&0x0F)+'0';

// Transformando Mes

Txt[12]= ((Data[4]&0xF0)>>4)+'0';

Txt[13]= (Data[4]&0x0F)+'0';

// Transformando Año

Txt[17]= ((Data[5]&0xF0)>>4)+'0';

Txt[18]= (Data[5]&0x0F)+'0';

}
```

```
void LCD_ConfigPin(){

    pinMode(D7,OUTPUT);

    pinMode(D6,OUTPUT);

    pinMode(D5,OUTPUT);
```

```
pinMode(D4,OUTPUT);

pinMode(EN,OUTPUT);

pinMode(RS,OUTPUT);

digitalWrite(D7,LOW);

digitalWrite(D6,LOW);

digitalWrite(D5,LOW);

digitalWrite(D4,LOW);

digitalWrite(EN,LOW);

digitalWrite(RS,LOW);

}
```

```
void LCD_Ini(){

    delay(40);

    LCD_High(0x30);
```



```
delay(5);

LCD_High(0x30);

delayMicroseconds(200);

LCD_High(0x30);

delayMicroseconds(200);

LCD_High(0x20);

LCD_Cmd(0x2C);

LCD_Cmd(0x0C);

}

void LCD_Clr(){

    LCD_Cmd(0x01);

}
```

```
void LCD_Fila1(){  
  
    LCD_Cmd(0x80);  
  
}
```

```
void LCD_F1_C5(){  
  
    LCD_Cmd(0x84);  
  
}
```

```
void LCD_F2_C4(){  
  
    LCD_Cmd(0xC3);  
  
}
```

```
void LCD_F2_C8(){  
  
    LCD_Cmd(0xC7);  
  
}
```

```
}
```

```
void LCD_Chr(char Data){
```

```
    digitalWrite(RS, HIGH);
```

```
    LCD_High(Data);
```

```
    LCD_Low(Data);
```

```
    delayMicroseconds(50);
```

```
}
```

```
void LCD_Cmd(char Data){
```

```
    digitalWrite(RS, LOW);
```

```
    LCD_High(Data);
```

```
    LCD_Low(Data);
```

```
    delay(2);
```

```
}
```

```
void LCD_High(char Data){
```

```
    if(bitRead(Data,7))
```

```
        digitalWrite(D7, HIGH);
```

```
    else
```

```
        digitalWrite(D7, LOW);
```

```
    if(bitRead(Data,6))
```

```
        digitalWrite(D6, HIGH);
```

```
    else
```

```
        digitalWrite(D6, LOW);
```

```
    if(bitRead(Data,5))
```

```
        digitalWrite(D5, HIGH);
```

```
    else
```

```
    digitalWrite(D5, LOW);

    if(bitRead(Data,4)

        digitalWrite(D4, HIGH);

    else

        digitalWrite(D4, LOW);

    digitalWrite(EN, HIGH);

    delayMicroseconds(1);

    digitalWrite(EN, LOW);

}

void LCD_Low(char Data){

    if(bitRead(Data,3)

        digitalWrite(D7, HIGH);

    else
```

```
digitalWrite(D7, LOW);  
  
if(bitRead(Data,2))  
  
    digitalWrite(D6, HIGH);  
  
else  
  
    digitalWrite(D6, LOW);  
  
if(bitRead(Data,1))  
  
    digitalWrite(D5, HIGH);  
  
else  
  
    digitalWrite(D5, LOW);  
  
if(bitRead(Data,0))  
  
    digitalWrite(D4, HIGH);  
  
else  
  
    digitalWrite(D4, LOW);  
  
digitalWrite(EN, HIGH);
```

```
    delayMicroseconds(1);

    digitalWrite(EN, LOW);

}

void LCD_PrintNum(int num){

    LCD_Chr((num/10)+48);

    LCD_Chr((num%10)+48);

}

void LCD_PrintHora(){

    int i;

    LCD_Clr();

    LCD_Fila1();

    for(i=0;i<16;i++)
```

```
LCD_Chr(Hora[i]);  
  
LCD_F2_C8();  
  
LCD_PrintNum(RTC_Timer[0]);  
  
}
```

```
void LCD_PrintMinuto(){  
  
int i;  
  
LCD_Clr();  
  
LCD_Fila1();  
  
for(i=0;i<16;i++)  
  
LCD_Chr(Minuto[i]);  
  
LCD_F2_C8();  
  
LCD_PrintNum(RTC_Timer[1]);  
  
}
```



```
void LCD_PrintRTC(char Txt[20]){  
  
    int i;  
  
    LCD_F1_C5();  
  
    for(i=0;i<8;i++)  
  
        LCD_Chr(Txt[i]);  
  
    LCD_F2_C4();  
  
    for(i=9;i<19;i++)  
  
        LCD_Chr(Txt[i]);  
  
}
```

### **ALGORITMO**

1. Al encender el sistema mostrara el Arduino la hora obtenida por el RTC y el Raspberry Pi también estará ejecutándose.

2. El usuario fija la hora o minuto con los pulsadores, que mantendrá apagado al Raspberry Pi por medio del Arduino y relé de 5 voltios que sería la parte de control de bajo consumo de energía.
3. El sistema del panel solar y el regulador, estarán cargando continuamente la batería de ion-litio en el día.
4. El Raspberry Pi al terminar su rutina, envía un pulso de nivel alto al Arduino que a su vez activara el apagado fijado por el usuario después de completarse 5 segundos aproximadamente, necesarios para que el Raspberry Pi se coloque en modo de desconexión.
5. Después de transcurrir el tiempo de apagado fijado por el usuario el Arduino cambia de estado al relé de 5 voltios que conectara al Raspberry Pi con el voltaje proporcionado por la batería que serían de 5 voltios, y se repite el proceso inicial.

## **CAPÍTULO 4**

### **SIMULACIÓN, PRUEBAS Y EVALUACIÓN**

#### **4.1 RESUMEN**

En este capítulo detallaremos los diferentes ejercicios e implementaciones para poder lograr el ensamblaje de nuestro prototipo, dando una descripción de cada una de las etapas, especificando el uso de cada uno de los materiales y dispositivos electrónicos que hemos utilizados, así también como los resultados obtenidos en cada una de estas etapas.

A continuación mencionamos paso a paso las prácticas elaboradas para alcanzar el objetivo y la evaluación del proyecto.

## 4.2 SIMULACIÓN DEL CIRCUITO CONMUTADOR EN PROTEUS

Mediante del simulador de circuitos electrónicos PROTEUS se simulo el circuito conmutador compuesto por el Arduino UNO, un LCD Display de 16x2 y un RTC el cual esta conformado principalmente por el integrado DS1307. La Figura 4.1 nos muestra la simulación en PROTEUS del proyecto.

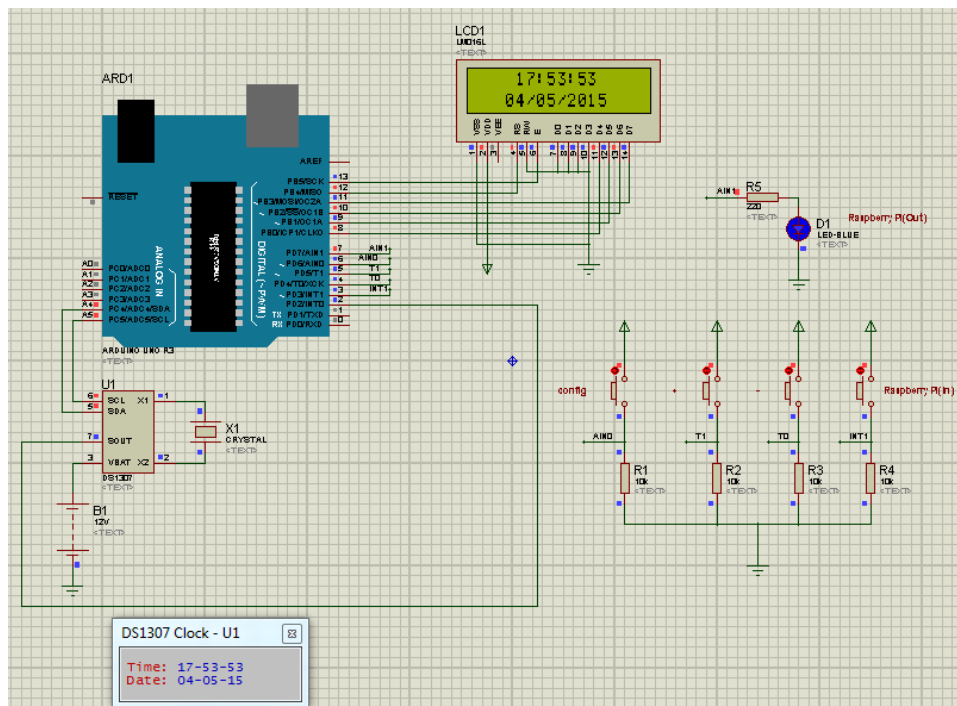
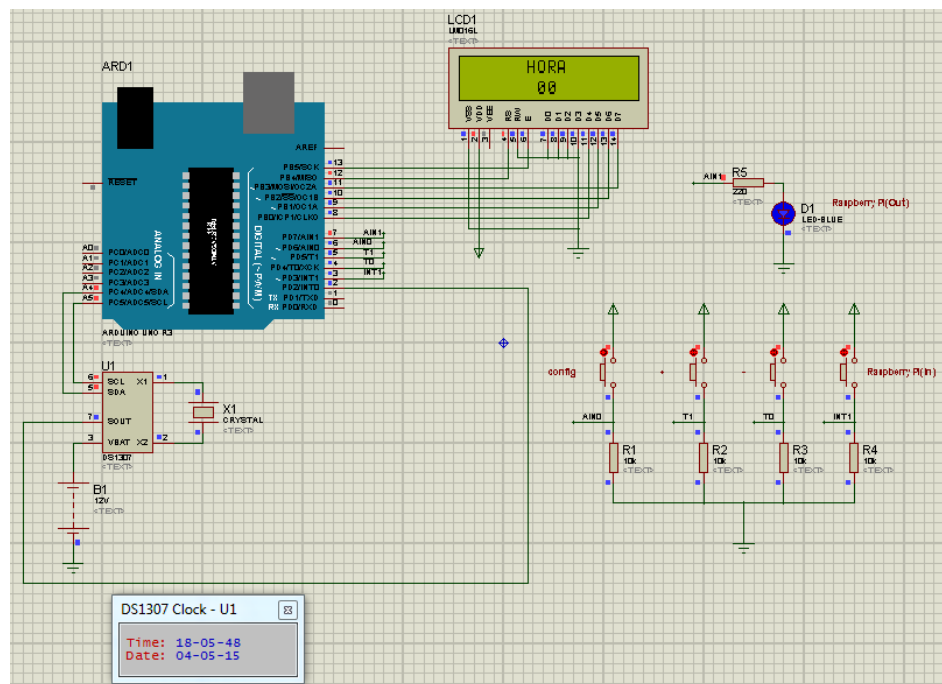


Fig. 4.1 Simulación del circuito conmutador.

El usuario puede programar el tiempo de desconexión de la fuente de alimentación del Raspberry, mediante el ingreso de hora y minuto. De esta forma se puede tener control del tiempo que el Raspberry Pi permanecerá apagado, logrando así la optimización de la batería. La Figura 4.2 para el ingreso de hora y minuto respectivamente.



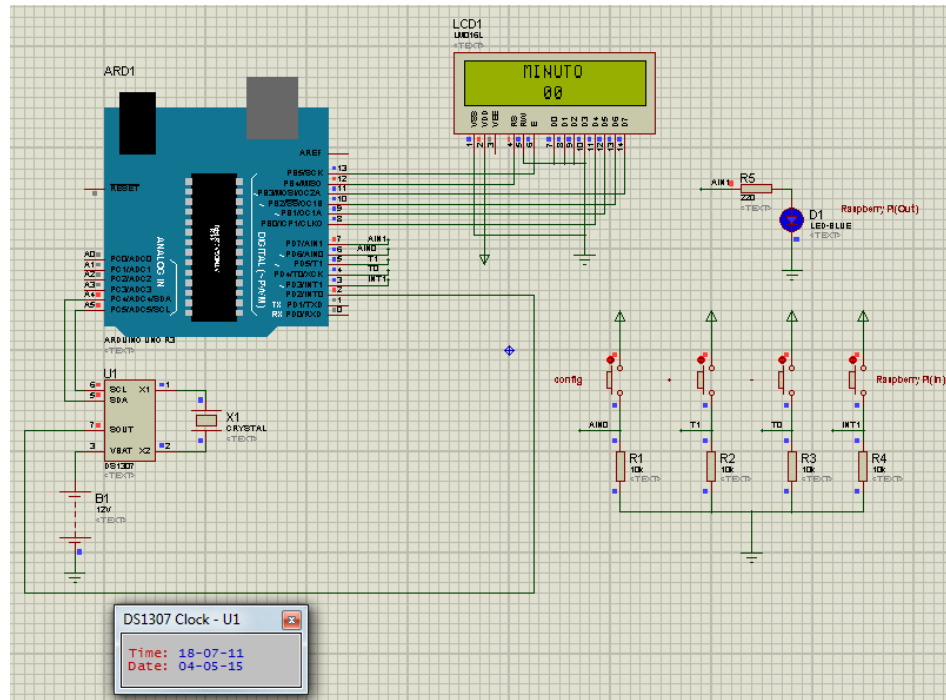


Fig. 4.2 Ingreso de hora y minuto.

#### 4.2.1 IMPLEMENTACIÓN DEL CIRCUITO CONMUTADOR EN PROTOBOARD

En este ensayo armamos en el protoboard el circuito conmutador compuesto por el Arduino UNO y el Real Time Clock, mediante un display se puede observar la fecha y la hora entregada por el Real Time Clock.

Además para que el usuario pueda programar la conmutación se hace uso de pulsadores, los cual permite al usuario establecer horas o minutos en los cuales el Arduino UNO procedería hacer la conmutación.

La Figura 4.3 muestra la implementación del circuito conmutador en el protoboard, así como la interface para ingreso de las horas o minutos que durara la conmutación.



**Fig. 4.3 Implementación de circuito conmutador en el protoboard.**

#### **4.3 SIMULACION DE RUTINA EN RASPBERRY PI**

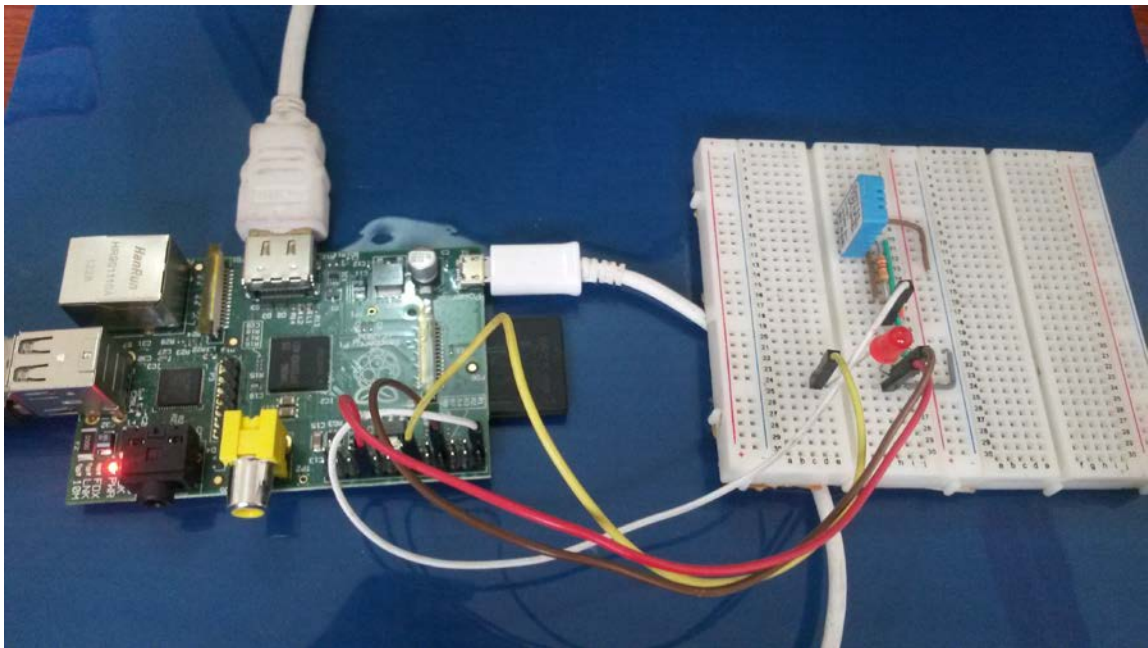
Para esta simulación se tomara una rutina cualquiera la cual será ejecutada por medio del Raspberry Pi. La rutina que hemos utilizado es la detección de la Humedad Relativa y Temperatura del ambiente.

Para esta rutina se utiliza el sensor DHT11 el cual es capaz de detectar la humedad relativa y la temperatura del ambiente. La Figura 4.4 muestra el sensor DHT11 con su distribución de pines.



**Fig. 4.4 Sensor DHT11.**

La Figura 4.5 muestra la implementación del circuito detector de la humedad relativa y temperatura del ambiente conectado con el Raspberry Pi mediante los puertos GPIO.

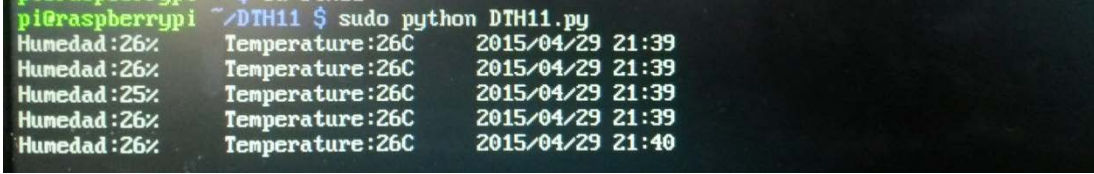


**Fig. 4.5 Detector de humedad relativa y temperatura del ambiente.**



El led indicador rojo que se observa en la Figura 4.5 se encenderá cuando el Raspberry Pi haya terminado de tomar las muestras de la humedad relativa y temperatura del ambiente. Esta será la señal que el Arduino UNO recibirá a través de sus puertos de entrada, la misma será el aviso para que el Arduino UNO desconecte al Raspberry Pi de su fuente de alimentación.

La Figura 4.6 muestra ejecución de la rutina realizada por el Raspberry Pi, para este caso se ha programado tomar 5 muestras de humedad relativa y temperatura del ambiente.



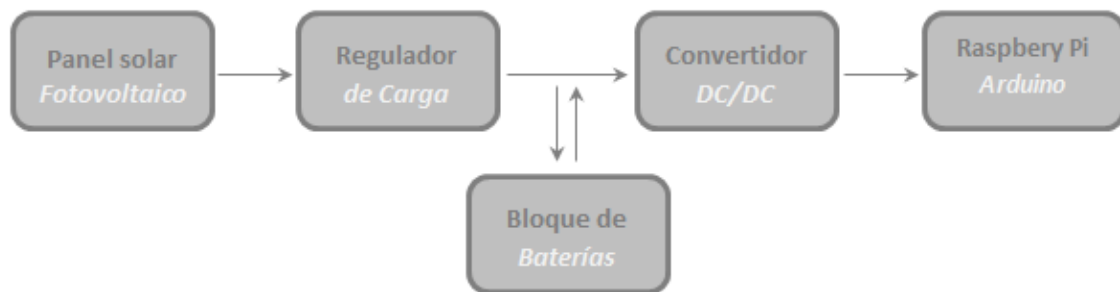
```
pi@raspberrypi ~/DTH11 $ sudo python DTH11.py
Humedad:26% Temperature:26C 2015/04/29 21:39
Humedad:26% Temperature:26C 2015/04/29 21:39
Humedad:25% Temperature:26C 2015/04/29 21:39
Humedad:26% Temperature:26C 2015/04/29 21:39
Humedad:26% Temperature:26C 2015/04/29 21:40
```

**Fig. 4.6 Ejecución de rutina en Raspberry Pi.**

Una vez terminada la rutina por parte del Raspberry Pi este estará listo para ser desconectado de su fuente de alimentación, así se lograra optimizar el consumo de potencia. Cuando el Raspberry Pi es conectado nuevamente a su fuente de alimentación, se ejecutara la rutina de detección de humedad relativa y temperatura del ambiente automáticamente.

#### 4.4 DISEÑO SISTEMA FOTOVOLTAICO

Como se analizó en el Capítulo 2 un sistema fotovoltaico es un dispositivo que, a partir de la insolación, produce energía eléctrica en condiciones de ser aprovechada por el hombre. La Figura 4.7 se muestra el diagrama de bloques del sistema fotovoltaico utilizado en este proyecto.



**Fig. 4.7 Diagrama de bloques de sistema fotovoltaico.**

Para realizar el cálculo fotovoltaico es necesario conocer la ubicación geográfica en el cual se instalara el sistema fotovoltaico. El lugar donde se pretende instalar el SFV es en la ciudad de Guayaquil. La siguiente tabla muestra los niveles de irradiación difusa correspondientes a la ciudad de

**Guayaquil. [33]**

IRRADIACIÓN TOTAL DIFUSA PARA LA CIUDAD DE GUAYAQUIL (KWh/m <sup>2</sup> día)												
	<i>Ene</i>	<i>Feb</i>	<i>Mar</i>	<i>Abr</i>	<i>May</i>	<i>Jun</i>	<i>Jul</i>	<i>Ago</i>	<i>Sep</i>	<i>Oct</i>	<i>Nov</i>	<i>Dic</i>
<i>Irradiación Total</i>	4,41	4,48	4,65	4,36	4,27	4,14	4,3	4,62	4,97	4,88	4,94	4,93

**Tabla 4.1 Valores de irradiación para la ciudad de Guayaquil. [33]**

De la Tabla 4.1 tomamos el nivel de irradiación más bajo, asumiendo que nuestro sistema trabajará en las peores condiciones de irradiación. El panel utilizado en este proyecto consta con las siguientes especificaciones técnicas:

- Potencia máxima: 5W.
- Voltaje máximo: 10V.
- Corriente máxima: 500mA.

La potencia máxima entrega por el panel solar nos indica que, en un día despejado y soleado, a las 12 del mediodía, el panel estará entregando una potencia de 5W a luz solar plena, indiferente de donde sea instalado.

Multiplicando el valor de potencia máxima del panel y el nivel de irradiación más bajo tomado de la tabla 4.1, obtenemos la potencia que genera el panel en KW/día. Por lo tanto nuestro panel genera una potencia de 20,7 Wh/día.

El consumo racional de la energía es uno de los factores más importantes al momento de dimensionar un sistema fotovoltaico, por esta razón es necesario conocer el objetivo a electrificar. Se debe conocer las características eléctricas de los equipos a alimentar y el tiempo de empleo por parte del usuario del sistema. Es decir, se hace necesario conocer o estimar la corriente y la tensión o voltaje de trabajo de los equipos instalados y el número de horas diarias de trabajo, teniendo en cuenta las futuras ampliaciones que se hagan a futuro en el sistema.

Para objeto de este proyecto los equipos a electrificar son: Raspberry Pi y Arduino UNO, junto con un lcd display 16x2. La Tabla 4.2 muestra el consumo de potencia de los equipos utilizados en el prototipo de este proyecto.

<i>EQUIPO</i>	<i>REQUERIMIENTO DE POTENCIA</i>
<i>Raspberry Pi</i>	3,5
<i>Arduino UNO, Lcd display 16x2</i>	1

**Tabla 4.2 Consumo de potencia.**

Estimando que nuestro sistema a alimentar tiene como tiempo de operación 5 horas/día, la Tabla 4.3 muestra el consumo de potencia por día.

<i>EQUIPO</i>	<i>POTENCIA W</i>	<i>USO horas/días</i>	<i>ENERGÍA Wh/día</i>
<i>Raspberry Pi</i>	3,5	4	14
<i>Arduino UNO, Lcd display 16x2</i>	1	4	4

**Tabla 4.3 Consumo de energía Wh/día.**

Sumando los valores de energía tomados de la Tabla 4.3 obtenemos el total de requerimiento de energía por día del sistema. El valor de energía total es igual a 18 Wh/día. Por lo tanto el panel que, entrega una energía aproximada de 20,7 Wh/día en el peor caso de insolación, será capaz de mantener operativo al sistema a alimentar.

Basado en el consumo propuesto por el sistema a alimentar, se requiere mantener el sistema operativo cuando los paneles solares no están entregando ningún nivel de voltaje y corriente (noche). Para ello se hace uso de baterías.

Por lo tanto tomando los valores potencia diaria del sistema y la tensión de corriente continua del sistema obtenemos la carga diaria de corriente.

$$\frac{18 \text{ Wh/día}}{5 \text{ V}} = 3,6 \text{ Ah} \quad (4.1)$$

Al valor obtenido en la ecuación 4.1 le multiplicación el factor de seguridad que es igual a 1,2.

$$3,6 \text{ Ah} \times 1,2 = 4,32 \text{ Ah} \quad (4.2)$$

El valor obtenido en la ecuación 4.2 es la carga de corriente diaria más el 20% como factor de seguridad. Para poder dimensionar la capacidad nominal de la batería a utilizar es importante saber los días de reserva de la batería, este valor se lo toma entre un rango de 1 a 5 días. El valor elegido en este proyecto será de 1 día. Por lo tanto la capacidad nominal de la batería se obtiene de multiplicar la el resultado obtenido de la ecuación 4.2 y los días de reserva.

$$4,32 \text{ Ah} \times 1 \text{ día} = 4,32 \text{ Ah} \quad (4.3)$$

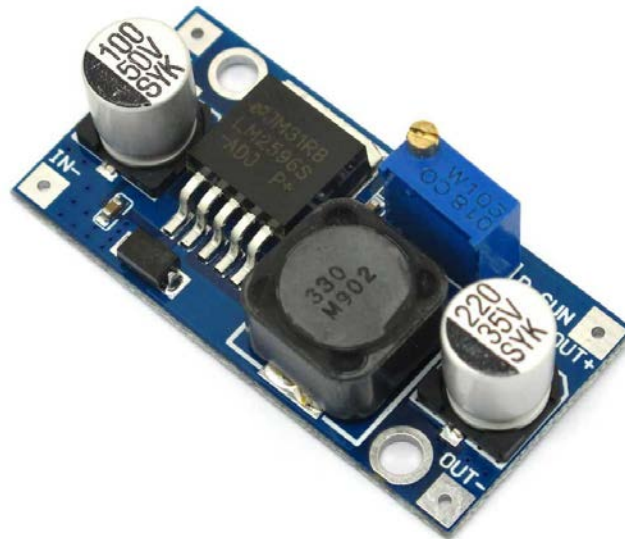
El resultado obtenido en la ecuación 4.3 nos indica que necesitaríamos una batería con capacidad nominal de 4,32 Ah para poder abastecer a nuestro sistema durante 1 día.

La batería utilizada en este proyecto es una batería de plomo – ácido, las especificaciones técnicas se detalla a continuación:

- Tensión nominal: 6V.
- Capacidad nominal: 4Ah.
- Intensidad nominal: 0,2A.
- Intensidad máxima de carga: 1,35A.
- Intensidad máxima de descarga: 45 A(5 sec).

Para este caso no sería necesario más que una sola batería para poder abastecer el sistema en situaciones de nubosidad o de niveles de irradiación cero (noche).

Como controlador de carga se utiliza una placa controladora de carga compuesta principalmente por el integrado LM2596S–ADJ, el cual será el encargado de acondicionar el voltaje que entrega el panel solar para la correcta carga de la batería y al mismo tiempo desconectarla cuando esta ya se encuentra cargada. La Figura 4.8 muestra la tarjeta controlado de carga.



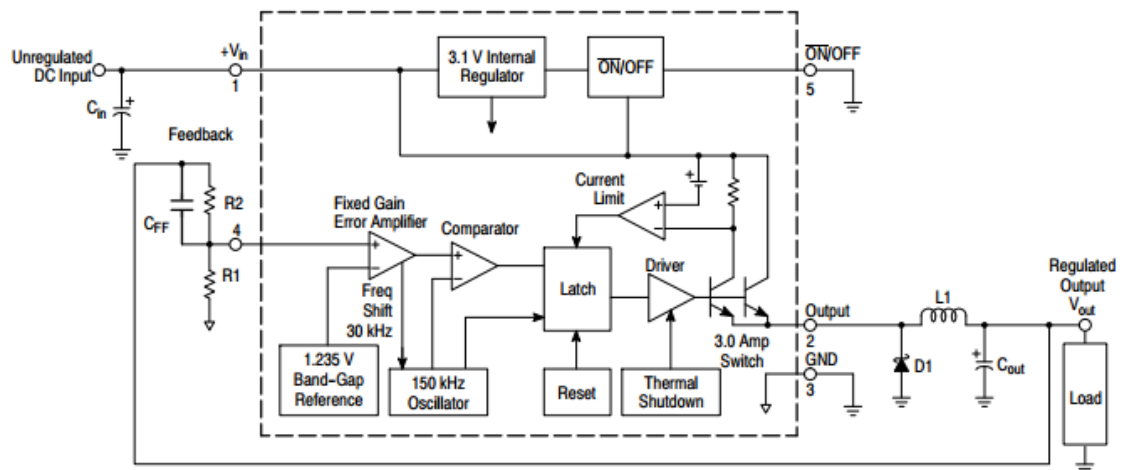
**Fig. 4.8 Controlador de carga.**

El integrado LM2596s-ADJ es un Switcher Step-Down con voltaje de salida regulado, véase las especificaciones en ANEXO. Las principales características de este integrado se detallan a continuación:

- Voltaje de salida ajustable: 1.2 V a 37 V.
- Voltaje de entrada: hasta 40 V.
- Corriente de salida máxima: 3A.
- Frecuencia de Switching: 150 KHz.



La Figura 4.9 muestra el diagrama de bloques de la placa reguladora de carga, así como la distribución de pines del integrado LM2596s-ADJ.



**Fig. 4.9 Diagrama de bloques de la placa reguladora de carga.**

Centraremos nuestro análisis en las resistencias R1 y R2 que se observa en la Figura 4.9. Estas resistencias permitirán ajustar el voltaje de salida para la carga. EL datasheet del integrado nos da como dato que el voltaje que aparece en el pin 4 es igual al voltaje de referencia 1,235 V, ver ANEXO. Las ecuaciones para el cálculo de las resistencias R1 y R2 se muestran a continuación.

$$V_{out} = 1,235V \left(1 + \frac{R2}{R1}\right) \quad (4.4)$$

$$R2 = R1 \left( \frac{V_{out}}{1,235V} 1 \right) \quad (4.5)$$

Se conoce que la placa tiene como 330  $\Omega$  como resistencia R1 y el voltaje de salida  $V_{out}$  que se necesita para poder cargar la batería es de 6V, con estos valores se puede hallar la resistencia R2 como un voltaje de referencia de 1,235V. Reemplazando estos valores en la ecuación 4.5 obtenemos el valor de R2.

$$R2 = 330\Omega \left( \frac{6V}{1,235V} 1 \right) = 1,6 K\Omega \quad (4.5)$$

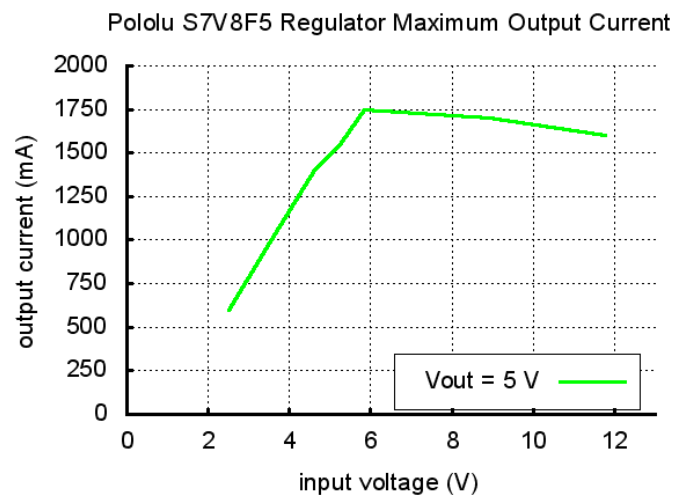
El valor obtenido en la ecuación 4.5 es el necesario para obtener un voltaje de salida de 6V. Cuando la batería esta carga completamente presentará en sus terminales un voltaje igual a 6V, este voltaje con las resistencias R1 y R2 ajustadas en 330  $\Omega$  y 1,6 K $\Omega$  respectivamente, reflejará en el pin 4 del LM2596s-ADJ un voltaje aproximado al voltaje de referencia  $V_{ref} = 1,235V$ , en este momento se desconecta la batería de su carga para evitar una sobre carga de la misma.

Por ultimo analizaremos el convertidor DC/DC colocado entre la batería y la carga a alimentar. Este convertidor será el encargado de establecer el voltaje ideal para la carga a alimentar, para nuestro prototipo necesitamos

un voltaje constante de 5V y una corriente de 700mA. Las principales características técnicas del convertidor step up/step down fabricado por POLOLU son las siguientes:

- Voltaje de entrada: 2,7V - 11,8V.
- Voltaje de salida: 5V.
- Corriente de salida: 500mA – 1A para ciertos niveles de voltajes.

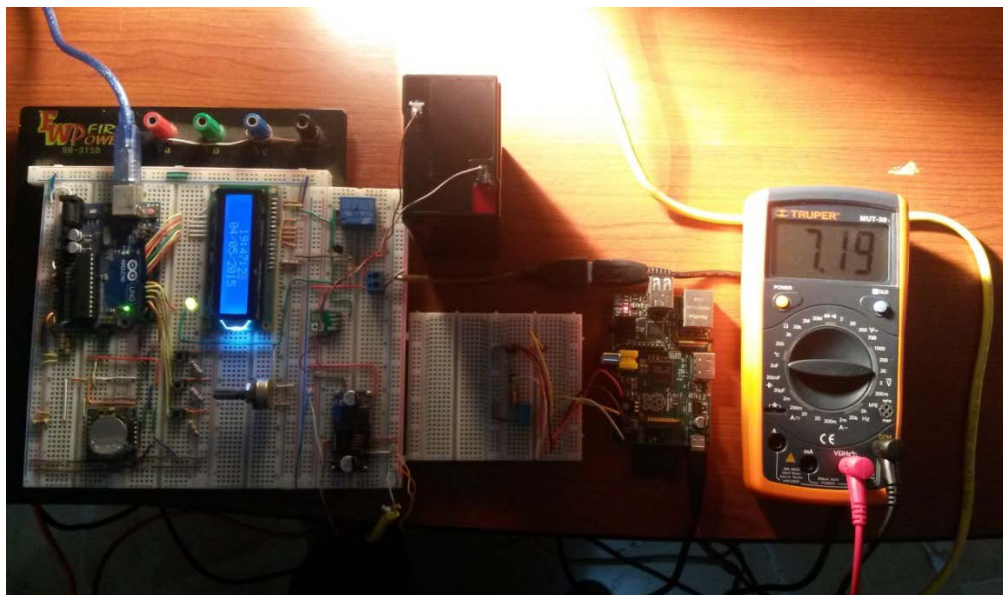
La corriente de salida del convertido DC/DC estará definida por los niveles de voltaje de entrada tal como se muestra en la Figura 4.10.



**Fig. 4.10** Máxima corriente de salida del convertidor DC/DC.

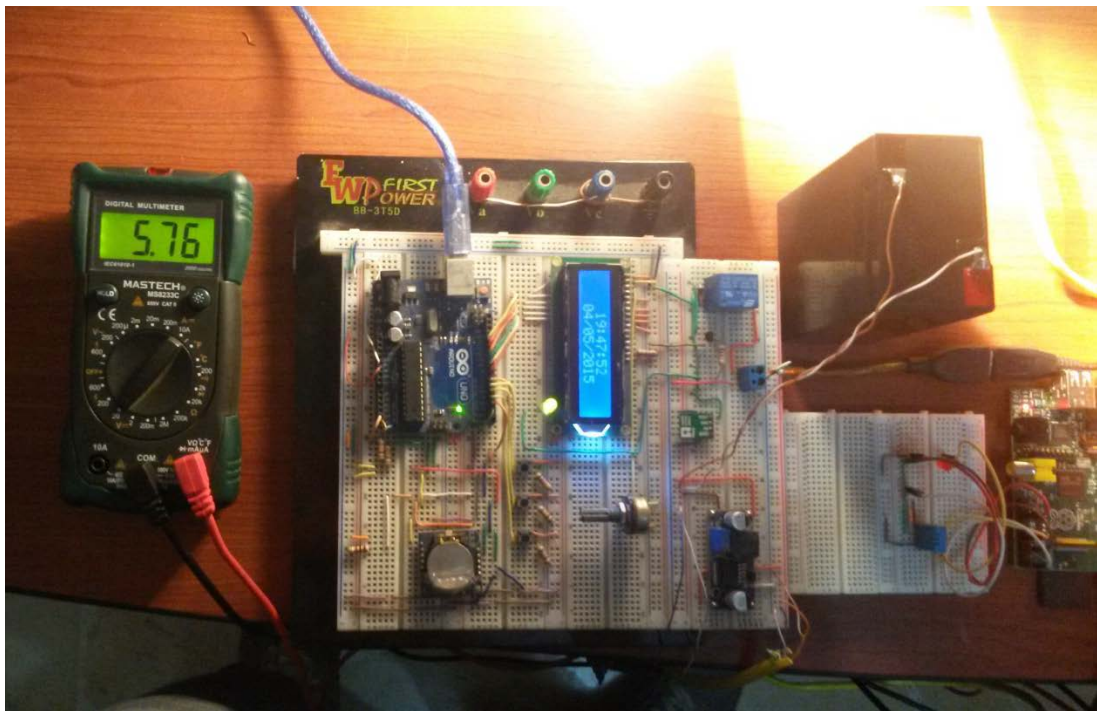
## 4.5 SIMULACIÓN PROYECTO

La Figura 4.11 contiene la implementación del proyecto final asignado, cuyo objetivo principal es el diseño de un sistema fotovoltaico como fuente de alimentación para una microcomputadora Raspberry Pi programada en bajo consumo de potencia. A diferencia de los microcontroladores, la Raspberry Pi no puede ser puesta en bajo consumo de potencia ya que está diseñada para consumir la menor potencia posible, es por eso que, como parte de la implementación de este proyecto, se utiliza un Arduino UNO como switch desconectado al Raspberry Pi de su fuente de alimentación cuando este está sin realizar ningún proceso.



**Fig. 4.11 Voltaje a la salida del panel solar.**

Se puede observar de la Figura 4.11 que el voltaje a la salida de panel es de aproximadamente 7,19 voltios, se usa como fuente lumínica un reflector para poder generar energía en el panel solar. Este voltaje es el mismo que ingresa al controlador de carga en la Figura 4.12 podemos observar que el voltaje a la salida del controlador de carga es aproximadamente de 5,76 voltios.



**Fig. 4.12 Voltaje a la salida del controlador de carga.**

El voltaje a la salida del controlador, es con el cual se cargara la batería, y a su vez dicho voltaje pasará por un convertidor DC/DC step up/step down para que

así acondicionar la tensión entregada por la batería o el panel solar a niveles de tensión óptimos para la alimentación de la carga.

## **CONCLUSIONES**

1. De manera concluyente y en relación con nuestros objetivos, la fuente de energía solar para nuestro sistema ha sido eficiente, es un sistema de bajo costo y cumpliendo con las especificaciones y demanda del sistema pudo obtenerse el objetivo trazado en este proyecto, la alimentación del Raspberry Pi utilizando un microcontrolador como un sistema para el control de la alimentación del mismo.
2. En nuestro proyecto de tesis el Raspberry Pi debe estar en bajo consumo de energía, pero como se trata de una minicomputadora no es posible colocarlo en ese estado ya que el procesador y demás bloques lógicos internos que conforma al Raspberry Pi deben necesariamente estar operativos, por este

motivo se usó un Arduino UNO como parte de control para el bajo consumo de la batería que proporciona la energía necesaria para el sistema, se concluye que es una manera efectiva y acoplable con los puertos GPIO del Raspberry Pi ya que tienen la misma función de proporcionar pines de conexión I/O para infinidad de aplicaciones, en este caso, para la comunicación entre el Raspberry Pi y el Arduino UNO, obteniendo el objetivo propuesto que el sistema sea de bajo consumo de energía.

3. La energía proveniente del panel solar se debió almacenarlo en una batería, en nuestro caso de Plomo-Ácido, por su rendimiento y amperios-horas de capacidad, ya que en sistemas que demandan monitoreo y un rendimiento en situaciones que no haya suficiente luz solar o días nublados pueda esta batería mantener en funcionamiento el sistema y no provocar pérdidas de mediciones u otros problemas que originen resultados finales erróneos.
4. La implementación de la parte de control que incluye un Arduino UNO y RTC (Real Time Clock) el cual es un chip que contiene días, semanas y meses, como también horas, minutos y segundos hasta el 2100, nos brinda una precisión de apagado y encendido aproximadamente similar a la obtenida en la simulación, comunicación sencilla I2C y autonomía de energía adecuada



que nos da como conclusión un sistema autónomo y de compresión de elementos electrónicos en la implementación.

## RECOMENDACIONES

1. Se debe considerar, al momento de dimensionar el sistema fotovoltaico, cual es la demanda de potencia diaria de la carga alimentar. Ya que sin este valor de consumo de potencia diaria no se podría saber qué capacidad de panel solar y de batería que se podría utilizar.
2. Es importante al momento de adquirir un panel solar este cumpla con las especificaciones técnicas, ya que un panel solar que no cumpla con las especificaciones técnicas del fabricante podría producir que nuestro sistema fotovoltaico sea ineficiente.

3. Documentar el código detalladamente para que un posterior ajuste o mejora sea sencillo entender y comprender en que línea de código se deba realizar el cambio y así obtener el resultado deseado.
  
4. Se recomienda al momento de instalar el sistema fotovoltaico protegerlo de fenómenos climáticos como la lluvia y humedad, ya que se está utilizando dispositivos electrónicos que son susceptible a la corrosión.

## BIBLIOGRAFÍA

- [1] Fundación Wikimedia.Inc, Historia de las computadoras personales, [http://es.wikipedia.org/wiki/Historia\\_de\\_las\\_computadoras\\_personales](http://es.wikipedia.org/wiki/Historia_de_las_computadoras_personales), fecha de consulta Marzo 2014.
  
- [2] Fundación Wikimedia.Inc, Descripción del Raspberry Pi, [http://es.wikipedia.org/wiki/Raspberry\\_Pi](http://es.wikipedia.org/wiki/Raspberry_Pi), fecha de consulta Marzo 2014.
  
- [3] Raspberry Shop, Descripción Raspberry Pi, <http://www.raspberrystore.com/que-es-raspberry-pi.php>, fecha de consulta Marzo 2014.
  
- [4] Diosdado, R., Placa Raspberry Pi, <http://www.zonamaker.com/raspberry/intro-raspberry>, fecha de consulta Marzo 2014.
  
- [5] Fundación Wikimedia.Inc, Descripción Arduino, <http://es.wikipedia.org/wiki/Arduino>, fecha de consulta Marzo 2014.
  
- [6] Raspberry Pi Foundation, Descripción del software para Raspberry Pi, <http://www.raspberrypi.org/help/faqs/#softwareOS>, fecha de consulta Marzo 2014.

- [7] Fundación Wikimedia.Inc, Descripción del software para Arduino, [http://es.wikipedia.org/wiki/Arduino#Lenguaje\\_de\\_programaci.C3.B3n\\_Arduino](http://es.wikipedia.org/wiki/Arduino#Lenguaje_de_programaci.C3.B3n_Arduino), fecha de consulta Marzo 2014.
- [8] Diosdado, R., Modelos del Raspberry Pi y descripción, <http://www.zonamaker.com/raspberry/intro-raspberry>, fecha de consulta Marzo 2014.
- [9] Fundación Wikimedia.Inc, RaspberryPi, [http://es.wikipedia.org/wiki/Raspberry\\_Pi](http://es.wikipedia.org/wiki/Raspberry_Pi), fecha de consulta Marzo 2014.
- [10] Fundación Wikimedia.Inc, Arquitectura ARM, [http://es.wikipedia.org/wiki/Arquitectura\\_ARM](http://es.wikipedia.org/wiki/Arquitectura_ARM), fecha de consulta Marzo 2014.
- [11] Fundación Wikimedia.Inc, Sistemas operativos soportados por el Raspberry Pi, [http://es.wikipedia.org/wiki/Raspberry\\_Pi#Sistemas\\_operativos](http://es.wikipedia.org/wiki/Raspberry_Pi#Sistemas_operativos), fecha de consulta Marzo 2014.

- [12] Raspberry Pi Foundation, Descarga del Raspbian Debian Wheezy, Raspberry Pi, <http://www.raspberrypi.org/downloads/>, fecha de consulta Marzo 2015.
- [13] BitTorrent, Inc, Descarga del uTorrent, <http://www.utorrent.com/intl/es/>, fecha de consulta Marzo 2014.
- [14] Diosdado, R., Descripción del Software del Raspberry Pi, <http://www.zonamaker.com/raspberry/intro-raspberry>, fecha de consulta Marzo 2014.
- [15] Fundación Wikimedia, Inc, Tipos de Software del Raspberry Pi, [http://es.wikipedia.org/wiki/Raspberry\\_Pi#Sistemas\\_operativos](http://es.wikipedia.org/wiki/Raspberry_Pi#Sistemas_operativos), fecha de consulta Marzo 2014.
- [16] DIY Makers, Puerto GPIO del Raspberry Pi, <http://diymakers.es/usando-el-puerto-gpio/>, fecha de consulta Marzo 2014.
- [17] Victor, Descripción del puerto GPIO del Raspberry Pi, <http://muyraspi.blogspot.com/2013/02/gpio.html>, fecha de consulta Marzo 2014.
- [18] Peatonet, Numeración de los pines para los dos modos del puerto GPIO del Raspberry Pi, <http://www.peatonet.com/raspberry-pi-y-los-pines-gpio->

controlando-un-led-con-bash-y-con-python/, fecha de consulta Marzo del 2014.

[19] Adafruit, Arduino Uno rv3, <http://www.adafruit.com/category/17>, fecha de consulta, fecha de consulta Marzo 2014.

[20] Fundación Wikimedia.Inc, Descripción de Arduino, <http://es.wikipedia.org/wiki/Arduino> , fecha de consulta Marzo 2014.

[21] Arduino, Arduino Uno, <http://arduino.cc/en/Main/ArduinoBoardUno>, fecha consulta Marzo 2014.

[22] The Economist, Sunlit Uplands, <http://www.economist.com/node/9217928>, fecha de consulta Mayo 2014.

[23] Méndez, J., Cuerva, R., Energía Solar Fotovoltaica. FC Editorial, Madrid, p 16 y 17, 2007.

[24] Min. Electricidad y Energía Renovables, Electrificación Rural con Energías Renovables, <http://www.energia.gob.ec/electrificacion-rural-con-energias-renovables/>, fecha de consulta Mayo 2014.

- [25] Saecsa, La Energía Solar: Datos Técnicos, [http://saecsaenergiasolar.com/renovables/energia\\_solar/](http://saecsaenergiasolar.com/renovables/energia_solar/), fecha de consulta Mayo 2014.
- [26] Lorenzo, E., Solar Electricity Engineering of Photovoltaic Systems, Institute of Solar Energy, Polytechnic University of Madrid, p 63, 1994.
- [27] Lorenzo, E., Solar Electricity Engineering of Photovoltaic Systems, Institute of Solar Energy, Polytechnic University of Madrid, p 46, 1994.
- [28] Roldán, J., Energías Renovables. Ediciones Paraninfo S.A., p 94-96, 2012.
- [29] Enalmex, La Celda Fotovoltaica, <http://www.enalmex.com/docpdf/libro/ch03.pdf.pdf>, fecha de consulta Junio 2014.
- [30] Velleman, Placas Solares Guía, [http://www.velleman.eu/downloads/6/placas\\_solares\\_guia.pdf](http://www.velleman.eu/downloads/6/placas_solares_guia.pdf), fecha de consulta Junio 2014.
- [31] Feunzalida, J., Estudio y Simulación de un Inversor por Paneles Solares Para Ser Aplicado a una Motobomba en el Uso de Regadío, <http://lep.eie.ucv.cl/jaimefuenzalida.pdf>, fecha de consulta Junio 2014.



- [32] McGraw-Hill, Componentes de una Instalación Solar Fotovoltaica, <http://www.mcgraw-hill.es/bcv/guide/capitulo/8448171691.pdf>, fecha de consulta Junio 2014.
- [33] Proyecto Conama, Guía Técnica sobre el Manejo de Baterías de Plomo Ácido Usadas, [www.sinia.cl/1292/articles-47018\\_recurso\\_1.pdf](http://www.sinia.cl/1292/articles-47018_recurso_1.pdf), fecha de consulta Junio 2014.
- [34] Conelec, Atlas Solar del Ecuador con Fines de Generación Eléctrica, [http://www.conelec.gob.ec/archivos\\_articulo/Atlas.pdf](http://www.conelec.gob.ec/archivos_articulo/Atlas.pdf), fecha de consulta Junio 2014.
- [35] Ejercicio #1 Manejar pulsadores con Arduino, <http://www.tecnosalva.com/arduino-pr%C3%A1ctica-n%C2%BA-2-control-mediante-pulsadores>, fecha de consulta Mayo 2014.
- [36] Ejercicio #2 Manejar LCD con Arduino, <http://comohacer.eu/como-controlar-una-pantalla-lcd-con-arduino/>, fecha de consulta Mayo 2014.

# ANEXOS

## Descargando el sistema operativo para la Raspberry Pi

En la configuración del Raspberry Pi se utilizara el sistema operativo Raspbian, el mismo que puede ser descargado desde la web, existen más sistemas operativos compatibles para Raspberry Pi, en este caso instalaremos la distribución RaspbianWheezy.

✚ Ingresar a la página [www.raspberrypi.org](http://www.raspberrypi.org)

✚ Ir al menú descargas <http://www.raspberrypi.org/downloads>

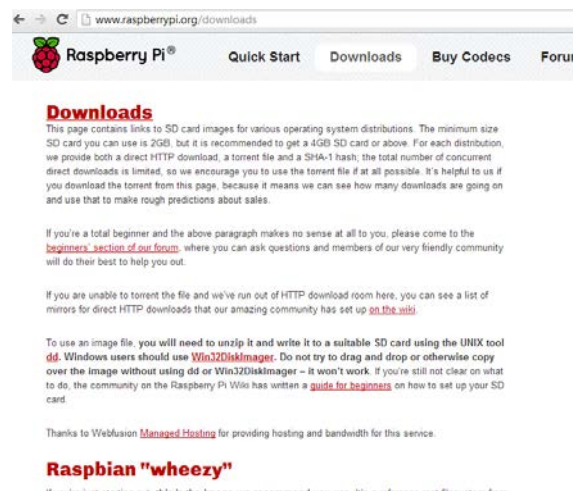
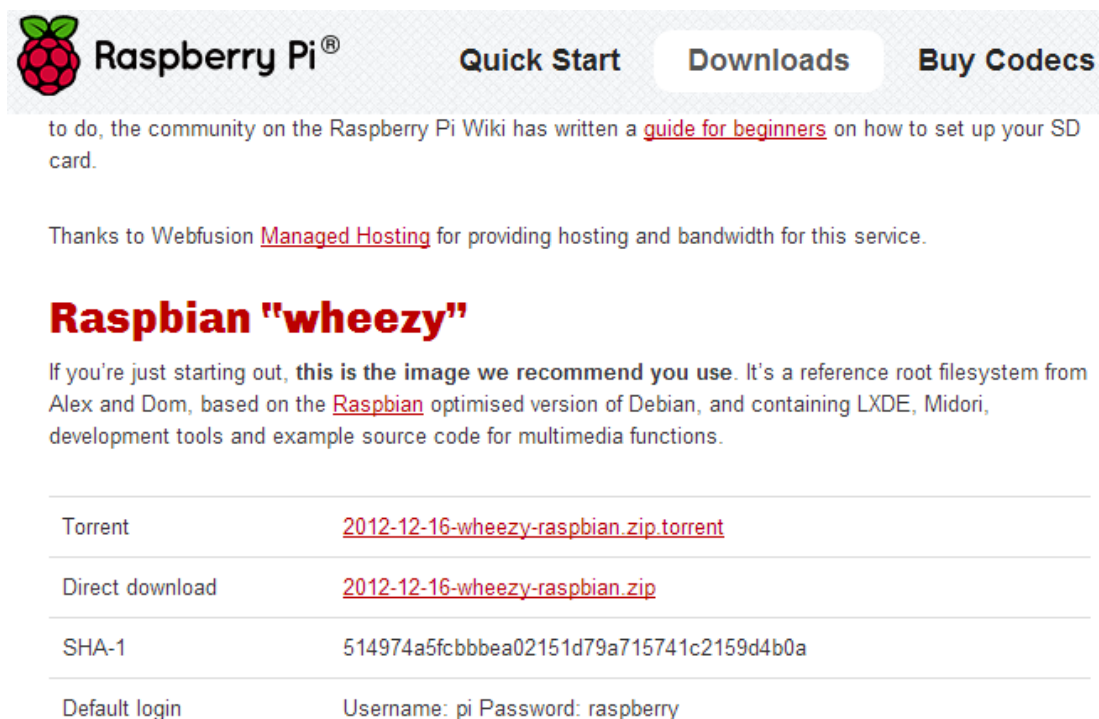


Fig. A1 Página de Raspberry Pi org opción descargas

En el menú descargas se procederá a descargar el programa Win32DiskImager que nos servirá para grabar el sistema operativo pre instalado en la memoria SD.

- ✓ En descargas se descargara la versión más actualizada de Raspbian.



The screenshot shows the Raspberry Pi website's 'Downloads' section. At the top, there is a navigation bar with the Raspberry Pi logo, 'Raspberry Pi®', and links for 'Quick Start', 'Downloads', and 'Buy Codecs'. Below the navigation bar, there is a paragraph of text: 'to do, the community on the Raspberry Pi Wiki has written a [guide for beginners](#) on how to set up your SD card.' This is followed by another paragraph: 'Thanks to Webfusion [Managed Hosting](#) for providing hosting and bandwidth for this service.' The main heading is 'Raspbian "wheezy"' in a large, bold, red font. Below this heading is a paragraph: 'If you're just starting out, **this is the image we recommend you use**. It's a reference root filesystem from Alex and Dom, based on the [Raspbian](#) optimised version of Debian, and containing LXDE, Midori, development tools and example source code for multimedia functions.' Below the text is a table with four rows: 'Torrent' with the link '2012-12-16-wheezy-raspbian.zip.torrent', 'Direct download' with the link '2012-12-16-wheezy-raspbian.zip', 'SHA-1' with the hash '514974a5fcbbbea02151d79a715741c2159d4b0a', and 'Default login' with 'Username: pi Password: raspberry'.

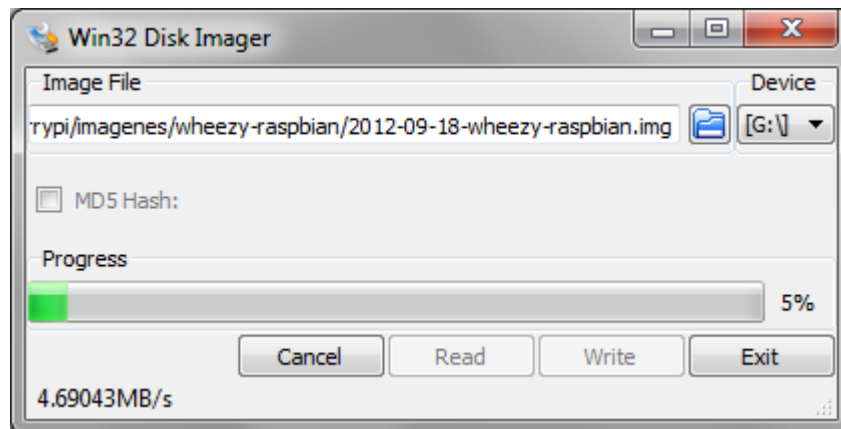
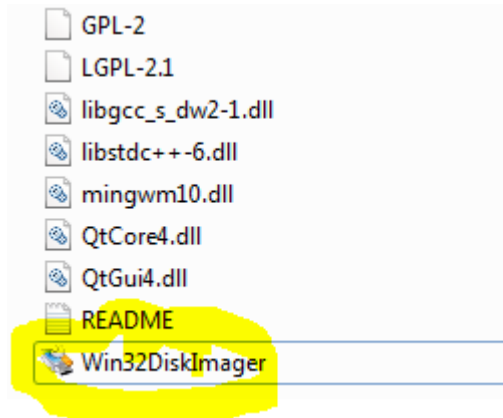
Torrent	<a href="#">2012-12-16-wheezy-raspbian.zip.torrent</a>
Direct download	<a href="#">2012-12-16-wheezy-raspbian.zip</a>
SHA-1	514974a5fcbbbea02151d79a715741c2159d4b0a
Default login	Username: pi Password: raspberry

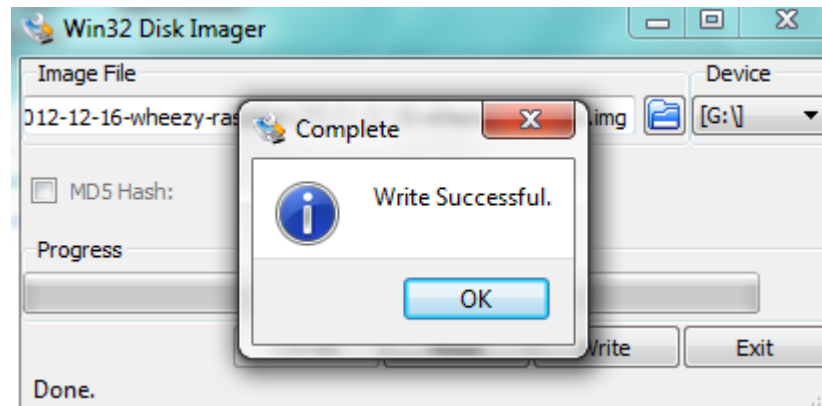
## Soft-float Debian "wheezy"

This image is identical to the Raspbian "wheezy" image, but uses the slower soft-float ABI. It is only intended for use with software such as the [Oracle JVM](#) which does not yet support the hard-float ABI used by Raspbian.

**Fig. A2 Opción descarga de sistema operativo**

- ✓ Una vez descargado, descomprimir la imagen y “quemar” en la memoria SD la versión de raspbian usando el programa win32DiskImage.





**Fig. A3 Descargando el sistema operativo en la memoria usando  
DiskImager**

Una vez realizado este proceso ya se puede utilizar la memoria en el dispositivo Raspberry pi.

- ✓ Una vez colocada la imagen en la tarjeta usando Win32DiskImager, podemos insertar la tarjeta en el Raspberry.
- ✓ Se conecta el teclado, mouse, el HDMI y el cable de alimentación a sus respectivos puertos.



Entre las opciones tenemos:

**info** - Información sobre qué es esta herramienta.

**expand\_rootfs** - Para que el sistema de ficheros de nuestra distribución de Linux utilice todo el espacio que permita nuestra tarjeta SD.

**configure\_keyboard** - Para elegir la configuración de teclado (en este caso, uno "Generico de 105 teclas", que aparece como "Generic 105 key") y su distribución (se propone "English (UK)", pero deberíamos escoger "Other" y luego "Spanish"; se nos preguntará ciertas variantes, como el teclado de Apple Macintosh, o el catalán, pero generalmente bastará con dejar simplemente "Spanish").

Finalmente se nos preguntará también cómo queremos que se comporte la tecla AltGr (se puede dejar con su valor por defecto) y si queremos que la combinación Ctl+Alt+Borrar pueda cerrar el entorno gráfico, algo que a nosotros por lo general nos gusta conservar, por si el entorno gráfico llegase a bloquearse, algo que es muy poco frecuente.

**change\_locale** -Para indicar la configuración regional. Lo razonable en España es escoger "es\_ES.UTF-8". Aun así, tenemos que asumir que lo habitual es que los mensajes de error de consola, e incluso los menús en modo gráfico, no estén traducidos al castellano y nos aparezcan en inglés.

**change\_timezone** - Para indicar la zona horaria, que en nuestro caso podría ser Europe / Madrid, aunque no es algo crítico, porque el Pi no tiene reloj interno, y la fecha y hora se perderá de una sesión de trabajo a la siguiente.

**memory\_split** - Para elegir el reparto entre memoria principal y memoria de video. Como mi uso no va a ser intensivo en gráficos, prefiero dejar más memoria principal (224+32).

**ssh** - Para habilitar o bloquear el servidor de ssh, por si quisiéramos conectar al Pi desde la red.

**boot\_behaviour** - Para indicar el comportamiento de arranque: si queremos que comience con entorno gráfico o con una consola en modo texto. A mí me gusta que arranque en consola, y, si quiero usar el entorno gráfico, sólo tengo que teclear "startx".

**update** - Para intentar actualizar la utilidad raspi-config. En mi caso, no tengo el Pi conectado a Internet, así que no me he molestado en probarlo.



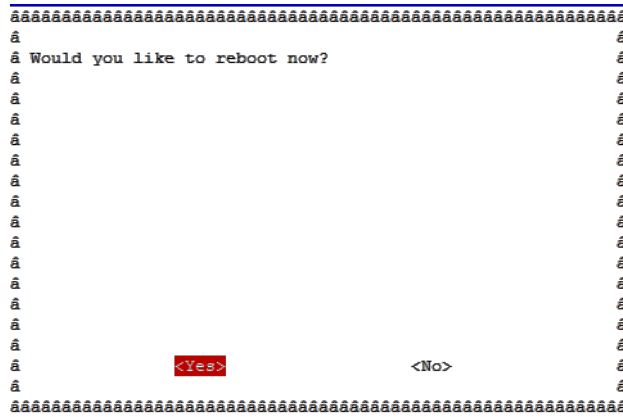






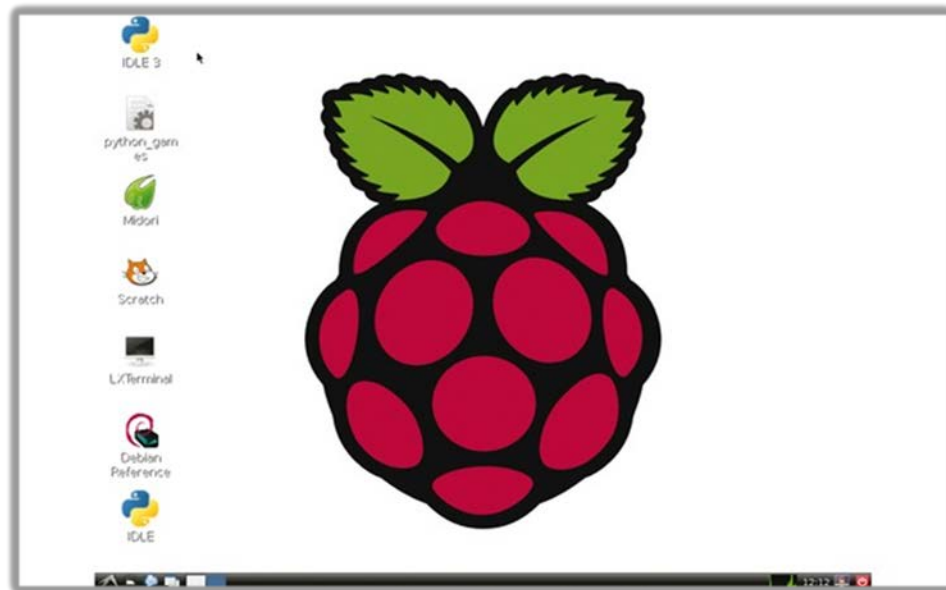
**Fig. A10 Confirmación para arrancar en modo Gráfico**

- ✓ Luego finalizamos para iniciar el sistema operativo, se requiere reiniciar.



**Fig. A11 Reiniciar S.O luego de configurar.**

**Pantalla inicial de instalación del sistema operativo.**



**Fig. A12 Pantalla principal de S.O en Raspberry pi**