

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

**“UTILIZACIÓN DE LA MINICOMPUTADORA RASPBERRY PI PARA LA
ADQUISICIÓN Y EVALUACIÓN DE DATOS DE NIVELES DE RADIACIÓN
SOLAR.”**

TESINA DE SEMINARIO

Previa la Obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Ever Jair Carpio Inga

Boris Wladimir Puerto Candela

GUAYAQUIL – ECUADOR

AÑO 2014

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

“UTILIZACIÓN DE LA MINICOMPUTADORA RASPBERRY PI PARA LA ADQUISICIÓN Y EVALUACIÓN DE DATOS DE NIVELES DE RADIACIÓN SOLAR.”

TESINA DE SEMINARIO

Previa la Obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Ever Jair Carpio Inga

Boris Wladimir Puerto Candela



CIB - ESPOL

GUAYAQUIL – ECUADOR

AÑO 2014

espol

Biblioteca



D-99483

AGRADECIMIENTO

A Dios,

A mis Padres, apoyo incondicional,

Al Ing. Carlos Valdivieso, por su desinteresada ayuda y consejos para terminar con éxito nuestro proyecto.

Boris Wladimir Puero Candela

A Dios porque Él me ha ayudado en este largo caminar y sigue manteniéndome en fe.

A mis padres porque ellos han sido un soporte muy fuerte en mi vida y nunca me dejaron solo.

Ever Jair Carpio Inga



CIB - ESPOL

DEDICATORIA

Le dedico este trabajo a mi madre, en especial a mis hijos Diana, Boris, Bryan, Irene, Ericka, Milena, Bruno y Susan, ya que con sus experiencias me he retroalimentado en el propósito de culminar mis estudios, y espero que este trabajo sea su incentivo para que continúen y culminen con éxito inmejorable sus carreras.

A todos ellos porque son quienes estarán orgullosos y felices por la finalización de este proyecto en mi carrera.

Boris Wladimir Puero Candela



CIB - ESPOL

DEDICATORIA

Les dedico este trabajo a mi padre Jair Carpio ya que él me inculco la virtud de la constancia, a mi madre Carmita Inga ya que este logro es por y para ella, que me demostró que para emprender algo hay que hacerlo con vigor y optimismo, a mi hermano para que le sirva de ejemplo que con motivación y dedicación se pueden lograr las cosas y a mi esposa Guisella que siempre me dio palabras de aliento para poder seguir y no desmayar cuando creía que todo lo había perdido.

Ever Jair Carpio Inga



TRIBUNAL DE SUSTENTACIÓN

A handwritten signature in black ink, appearing to read 'Carlos Valdivieso A.', written over a horizontal line.

Ing. Carlos Valdivieso A.

PROFESOR DEL SEMINARIO DE GRADUACIÓN

A handwritten signature in black ink, appearing to read 'Hugo Villavicencio V.', written over a horizontal line.

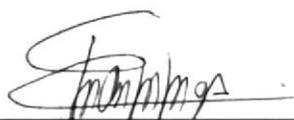
Ing. Hugo Villavicencio V.

PROFESOR DELEGADO POR LA UNIDAD ACADEMICA

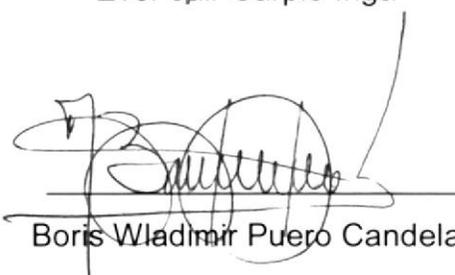
DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de Graduación de la ESPOL)



Ever Jair Carpio Inga



Boris Wladimir Puerto Candela



CIB - ESPOL

RESUMEN

Este trabajo abarca la investigación del uso de la minicomputadora Raspberry Pi tanto de su hardware como su software, la interconexión que se realizara con una celda solar y la realización de los objetivos planteados el cual es adquirir y evaluar los niveles de radiación solar con la utilización de la minicomputadora.

En el Capítulo I detallamos una descripción general del proyecto, elementos que se usarán y además aplicaciones prácticas para su posterior uso y en el Capítulo II, se encuentra el fundamento teórico de herramientas de hardware y software que fueron de utilidad para la realización del proyecto.

En el Capítulo III explica el diseño e implementación del prototipo, programación de la base de datos, la representación gráfica de la data y adquirirla por medio de un browser HTML.

En el cuarto capítulo se expone las pruebas y evaluación realizadas con el prototipo, simulación de la interfaz que interactúa con la Base de Datos, las pruebas y los resultados obtenidos y por último se dan a conocer las conclusiones y recomendaciones del proyecto en general.



ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
DEDICATORIA	IV
RESUMEN	VII
ÍNDICE GENERAL	VIII
ÍNDICE DE FIGURAS	XIII
INTRODUCCIÓN.....	XVI
CAPÍTULO 1	1
DESCRIPCIÓN GENERAL DEL PROYECTO	1
1.1 ANTECEDENTES	1
1.2 OBJETIVOS	2
1.3 RADIACION SOLAR	4
1.3.1 RADIACIÓN SOLAR FUERA DE LA ATMOSFERA	7
1.3.2 VARIACIÓN DE LA RADIACIÓN EN LA ATMOSFERA.....	8
1.3.3 RADIACIÓN SOLAR VS. MOVIMIENTOS TERRESTRES	11

1.3.4	RADIACIÓN SOLAR Y SUS CONSECUENCIAS	13
1.4	MUESTREO	14
1.5	CAPTURA	14
1.5.1	CONVERSIÓN ADC – TRANSMISIÓN Y RECEPCIÓN	15
1.5.2	ARCHIVADO EN EL RASPBERRY PI	15
1.6	PRESENTACIÓN WEB DE LA INFORMACIÓN	16
1.7	LIMITACIONES	16
CAPÍTULO 2		18
FUNDAMENTO TEÓRICO		18
2.1	REQUERIMIENTOS DEL PROYECTO	18
2.2	HERRAMIENTAS DE SOFTWARE	20
2.2.1	CCS COMPILADOR DE PIC LENGUAJE C	20
2.2.2	LINUX	21
2.2.3	DEBIAN	22
2.2.4	RASPBIAN	23
2.2.5	PROTEUS	24



2.2.6	APACHE	25
2.2.7	MySQL	25
2.2.8	PHP	26
2.3	HERRAMIENTAS DE HARDWARE	27
2.3.1	PIC 16F88	27
2.3.2	SENSOR SOLAR BPW34	29
2.3.3	RASPBERRY PI	31
2.3.4	PICKIT 3	32
CAPITULO 3		34
DESARROLLO DEL PROYECTO		34
3.1	RESUMEN	34
3.1.1	DIAGRAMA DE MODULOS GENERAL	35
3.1.2	DIAGRAMA DE COMUNICACIÓN MODULOS 1,2 Y 3	36
3.2	MODULO 1: PROGRAMACION PIC 16F88	37
3.2.1	ALGORITMO PROGRAMACION PIC 16F88	37
3.2.2	DIAGRAMA DE FLUJO PROGRAMACION PIC16F88	38

3.2.3 CODIGO FUENTE DEL PIC16F8839

3.3 MODULO 2: PROGRAMACION MAIN.C 40

3.3.1 DIAGRAMA DE FLUJO: MODULO GENERAL41

3.3.2 DIAGRAMA DE FLUJO: COMUNICACIÓN SERIAL.....42

3.3.3 ALGORITMO COMUNICACIÓN SERIAL42

3.3.4 DIAGRAMA DE FLUJO: CONEXIÓN BASE DE DATOS43

3.3.5 ALGORITMO COMUNICACIÓN SERIAL44

3.3.6 DIAGRAMA DE FLUJO: ENABLE ADC45

3.3.7 ALGORITMO COMUNICACIÓN SERIAL46

3.3.8 DIAGRAMA DE FLUJO: EJECUTAR QUERY47

3.3.9 DIAGRAMA DE FLUJO: GETSERIAL DATA48

3.3.10 ALGORITMO COMUNICACIÓN SERIAL49

3.3.11 DIAGRAMA DE FLUJO CREAR IMAGENES49

3.3.12 DIAGRAMA DE FLUJO VISUALIZAR IMAGENES50

3.4 CODIGO FUENTE MAIN.C51

CAPITULO 457



SIMULACION, PRUEBAS Y EVALUACION.....	57
4.1 RESUMEN	57
4.2 PRÁCTICA 1: USO SIMULADOR PROTEUS PARA CAPTURAR DATOS EN EL PROTOTIPO.	58
4.2.1 IMPLEMENTACION EN PROTOBOARD	59
4.3 PRÁCTICA 2: CREACION BASE DE DATOS.....	61
4.4 PRÁCTICA 3: CREACION DE UNA GRAFICA	68
4.5 SOLUCION DEL PROYECTO.....	71
4.6 GRAFICAS HTML DEL MINIMO RADIACION SOLAR	83
4.7 GRAFICAS HTML DEL MAXIMO RADIACION SOLAR.....	84
4.8 EVALUACION DE DATOS DE RADIACION SOLAR	85
BIBLIOGRAFIA.....	91

ÍNDICE DE FIGURAS

Figura 1.1 Índice Radiación Solar	7
Figura 2.1 Configuración de pines PIC 16F88	30
Figura 2.2 Sensor BPW34	32
Figura 2.3 RASPBERRY PI Modelo B	33
Figura 2.4 Hardware y Software Pickit 3	34
Figura 3.1 Diagrama de modulo general	37
Figura 3.2 Diagrama comunicación en los módulos 1,2 y 3	38
Figura 3.3 Secuencia conversión adc	40
Figura 3.4 Diagrama de flujo modulo general	44
Figura 3.5 Diagrama de flujo comunicación serial	45
Figura 3.6 Diagrama de flujo conexión base de datos.....	47
Figura 3.7 Diagrama de flujo enable adc	49
Figura 3.8 Diagrama de flujo ejecutar Query	50
Figura 3.9 Diagrama de flujo get serial data	51
Figura 3.10 Diagrama de flujo crear imágenes.....	52
Figura 3.11 Diagrama de flujo visualizar imágenes.....	53



Figura 4.1 Obtención de datos capturados simulación PROTEUS.....	65
Figura 4.2 Implementación del circuito captura en protoboard.....	66
Figura 4.3 Inicio de sesión phpMyAdmin.....	68
Figura 4.4 Inicio de sesión en phpMyAdmin, usuario y contraseña.....	68
Figura 4.5 Creación base de datos en phpMyAdmin.....	69
Figura 4.6 Asignación nombre base de datos phpMyAdmin	70
Figura 4.7 Inicialización de creación de tablas en phpMyAdmin	70
Figura 4.8 Definición de 3 columnas en phpMyAdmin	71
Figura 4.9 Definición tipo de variable de las columnas phpMyAdmin.....	71
Figura 4.10 Finalización de creación de Tablas phpMyAdmin	72
Figura 4.11 Inserción de datos en las tablas phpMyAdmin	73
Figura 4.12 Ingreso del primer dato en la base de datos	73
Figura 4.13 Ingreso de datos 19 – 20 en la base de datos	74
Figura 4.14 Ingreso del último dato en la base de datos	74
Figura 4.15 Comando RRDTOOL	75
Figura 4.16 Gráfica del ejercicio 3.....	76



Figura 4.17 Ensamblado general del prototipo	78
Figura 4.18 Gráfica de medición mínima (lectura nocturna).....	79
Figura 4.19 Gráfica de medición máximo (lectura día)	80
Figura 4.20 Datos de niveles de radiación	82

INTRODUCCIÓN

Hoy en día podemos hacer uso de ordenadores o dispositivos electrónicos con mucha destreza y sin la necesidad de consultar manuales de operación, adaptarse a las nuevas tendencias tecnológicas es algo innato, por esto se pretende con este prototipo demostrar que con el estudio de diferentes elementos tecnológicos y las pruebas necesarias se pueden construir proyectos económicos y fácilmente utilizables para el beneficio de la comunidad y con esto dar a conocer la minicomputador "Raspberry PI", que inicialmente fue creada para impulsar la enseñanza de las ciencias de la computación a niveles académicos pero la realidad ha sido distinta a pesar de cumplir con los propósitos establecidos, su reducido tamaño lo han convertido en una herramienta ideal para el desarrollo de diferentes proyectos, tanto en el ámbito empresarial, personal y de investigación.

Este proyecto describe la arquitectura de una interface construida con la celda solar BPW34 y el microcontrolador PIC 16F88, en combinación del minicomputador Raspberry PI y software "Open Source", se diseñó un dispositivo que sirve para capturar, almacenar, procesar y graficar datos.

El procesamiento de la data capturada es importante para poder mostrar datos actuales e históricos, encasillándose en la tecnología semántica que es



aquella que trata de manejar los datos y proporcionar un criterio inteligente que sirva para mediar entre las intenciones de los usuarios y la información disponible ya que actualmente es de interés científico poder estudiar los niveles de radiación solar para así analizar las variaciones del calentamiento global y el desarrollo de proyectos que necesiten de adquirir datos ya sea de: calor, temperatura, humedad, ruido, etc.



CAPÍTULO 1

DESCRIPCIÓN GENERAL DEL PROYECTO

1.1 ANTECEDENTES

La imperiosa necesidad de buscar nuevos recursos de energía renovable deberíamos en primer lugar dejar de crear ciertas formas de contaminación, dejar la contaminación eléctrica, nuclear y química. El uso excesivo de la electricidad en todo el mundo está afinando y destruyendo la ionosfera que dependemos para nuestra supervivencia. La contaminación radiactiva que hemos creado poco a poco está dañando nuestra estructura celular. Los productos químicos que liberan en nuestro entorno evitará la regeneración de vida, si continuamos estaremos envenenándonos a nosotros mismos. En segundo lugar, en un futuro cercano se desplazará el eje magnético de la



tierra. El detalle de mayor importancia es que tras este desplazamiento, la electricidad no trabajara, la nueva polaridad de la tierra no permitirá generar estructuras moleculares magnéticas en la ionosfera y formar electrones. [1]

Esto nos lleva a la importancia de realizar estudios de cómo adquirir energía renovable, en este caso la energía solar. El análisis a la radiación solar nos ha revelado lo beneficioso y perjudicial que este ha sido en el transcurso del tiempo para nuestras vidas.

Grandes instituciones mundiales para la salud nos informan el alto índice de riesgo en la piel por la inadecuada exposición del ser humano a los rayos solares. En otros casos, grandes científicos desarrollan estudios para almacenar y utilizar la energía inagotable del sol.

1.2 OBJETIVOS

OBJETIVO GENERAL

- Diseño y construcción de un dispositivo prototipo que adquiera y almacene datos de la radiación solar y con el uso de base de datos representar gráficamente los históricos de Radiación Solar vs. Tiempo.

OBJETIVOS ESPECIFICOS

- Implementar sensor de radiación solar en una interface al Raspberry PI.
- Medición y captura de datos de la radiación para su almacenamiento y procesamiento de dichos datos.
- Uso, configuración y aplicación del microcontrolador PIC 16F88 para la conversión análogo a digital.
- Manejo del microcomputador Raspberry PI en la inicialización del puerto GPIO, habilitando los puertos de transmisión Tx y Rx y configurando la velocidad de recepción de los datos.
- Determinar con los datos los índices de radiación solar en una localización específica.
- Capturar, muestrear, archivar y procesar los datos adquiridos con un sensor solar BPW34, para lo cual vamos a usar el PIC16F88, programado con MikroC PRO que se encargará en convertir el dato analógico en digital, recibir la información periódicamente en el RASPBERRY PI y almacenarlo en una Base de Datos MYSQL, procesando esta tabla estructurada de datos y con el uso de un servidor WEB RPI/APACHE pode realizar la gráfica en internet

1.3 RADIACION SOLAR

Se conoce por radiación solar al conjunto de radiaciones electromagnéticas y de partículas (electrones, protones y núcleos atómicos pesados raros) emitida por el sol; se ha observado que el espectro de la energía electromagnética tiene longitudes de ondas que varían entre 0.1 nm (un nanómetro es la millonésima parte de un metro) a 30 kilómetro la mayor parte de la energía se observa en el espectro de la luz visible (400-800nm). El espectro solar se extiende mucho más lejos de los límites que se observan en ambas direcciones. El sol emite también un flujo continuo de electrones con explosiones más cortas de lluvias de electrones y protones lo bastante intensas para afectar la ionización de la atmosfera terrestre exterior por ende la radiación solar se distribuye desde el infrarrojo hasta el ultra violeta. [2]

Toda la energía que utiliza, independientemente de la fisiología que actúa directamente sobre los seres animados provienen del sol.

El sol es una esfera de 109 veces el diámetro de la tierra, que irradia una cantidad enorme de energía originado por las reacciones nucleares en cadena.



Gran parte de la energía que produce sirve para mantener su propia temperatura y el resto la envía hacia el espacio.

La energía luminosa emitida por el sol, cuya emisión tiene lugar a través del fenómeno físico denominado radiación, se propaga en todas las direcciones del espacio sin soporte material.

La radiación cumple con la función de transportar energía de una forma parecida a como lo hacen las ondas de radio, televisión o la propia luz. Todos ellos son ondas electromagnéticas, aunque las del sol son de longitudes de onda más cortas.

Las ondas electromagnéticas emitidas por los rayos solares, se clasifican en tres grupos principales: rayos infrarrojos, luz visible, rayos ultravioletas.

En un haz de rayos solares, los que integran la luz blanca son perceptibles por el ojo humano (luz visible) y van del rojo al violeta pasando por el anaranjado, amarillo, verde, azul y violeta; pero por debajo del rojo existen otros denominados infrarrojos, y por encima del violeta los llamados ultravioletas, ambos invisibles para el ojo humano. Estos últimos son actínicos, es decir que no producen calor sino cambios químicos. **[3]**

La luz que llega a la tierra es energía radiante que va de la ultravioleta que es bloqueada en gran parte por la capa de ozono a la visible y la infrarroja. La mayor parte de la radiación solar se encuentra en la región de la luz visible del espectro. Del rango completo de la radiación solar que llega a la atmosfera de la tierra las longitudes de onda son de entre 400 y 700 nm. comparado con la luz visible. Las longitudes de ondas más cortas que las del espectro visible constituyen la radiación ultravioleta. Las radiaciones con longitudes de ondas más largas que las del espectro visible se denominan infrarrojas las cuales tienen una longitud de onda de 740 a 4000 nm.

La luz que llega a la superficie terrestre no es la misma que la que llega a la atmosfera. **[4]**

Por ello la intensidad de la radiación solar antes de entrar en la atmosfera llega a los límites más altos, pero el valor medio es de 1353 w/m^2 . En realidad no se trata de un valor rigurosamente constante, pues existe una variación de aproximadamente el 5,3% debido a la excentricidad de la órbita terrestre respecto al sol (-3.27% en el afelio, $\pm 3.42\%$ en el perihelio).

En el siguiente grafico se esquematiza los niveles de incidencia que tiene los rayos solares en todo el planeta.



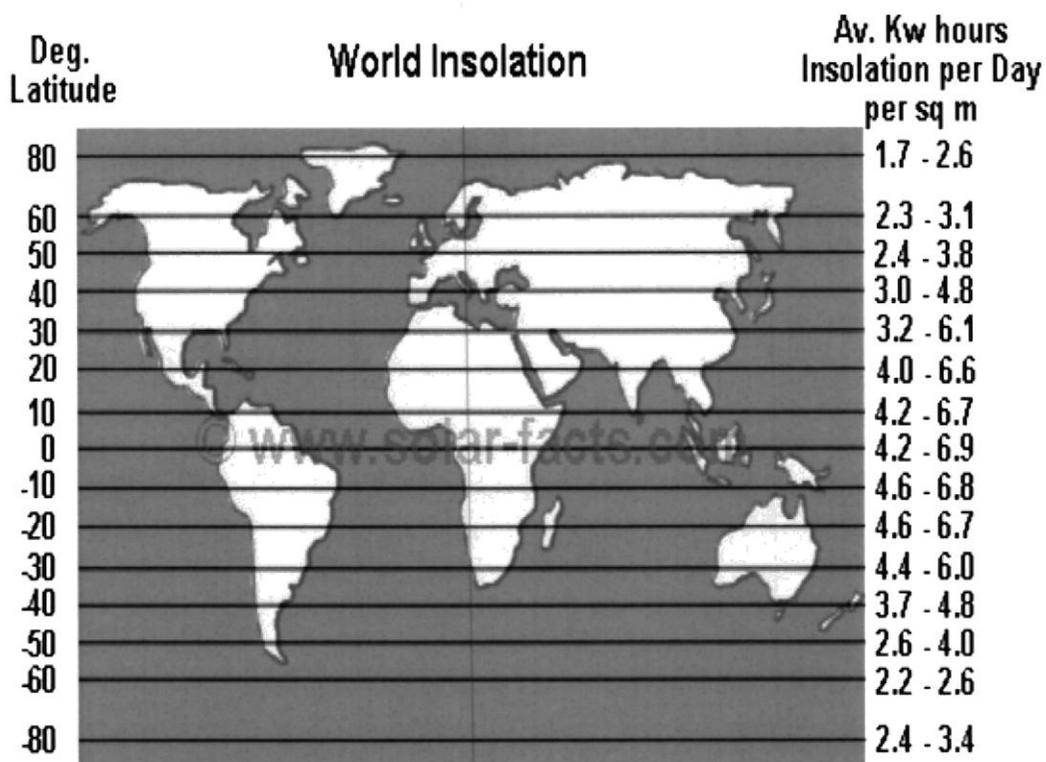


Figura 1.1 Indicé Radiación Solar [6]

1.3.1 RADIACIÓN SOLAR FUERA DE LA ATMOSFERA

El espectro de la radiación electromagnética, emitida por la superficie solar se extiende desde 0,2 μm . hasta 0,3 μm . De la energía total transmitida para los rayos solares, resulta que el 7% de la energía emitida cae en la zona ultravioleta del espectro, es decir desde 0,2 μm hasta 0,4 μm .; el 42% está comprendido en la zona "visible", es decir desde 0,2 μm (violeta y 0,8 μm . rojo). Mientras que el 51% restante se halla en la zona infrarroja, entré 0,8 μm y 3 μm .



1.3.2 VARIACIÓN DE LA RADIACIÓN EN LA ATMOSFERA

La atmosfera es bastante menos transparente de lo que pudiera parecer a simple vista esta modifica la calidad y la cantidad de la radiación solar que llega a la superficie terrestre, como consecuencia de la suma de varios factores [3],

La capa de ozono de la atmosfera exterior absorbe casi todas las longitudes de ondas, pero más intensamente las violetas y las azules de la luz, debido a la dispersión de la luz, gran parte de la radiación solar que llega a la tierra lo hace como luz difusa del cielo o también llamada como radiación difusa.

La contribución relativa de la luz directa y de la radiación difusa varía a lo largo del día y globalmente en función de la latitud. La luz indirecta representa una mayor proporción de la luz total interceptada en latitudes altas como resultado del bajo ángulo de la luz del sol. [4]

Es importante hacer las siguientes consideraciones en cuanto a la medida de las variables de la radiación solar en la superficie terrestre:

- **Radiación solar directa:** Es la radiación que proviene directamente del disco solar, y se la mide utilizando sistemas de seguimiento del movimiento del sol en su trayectoria.
- **Radiación solar difusa:** Es la radiación que proviene de las reflexiones en la atmósfera, y ha de medirse utilizando sistemas de sombreamiento del disco solar.
- **Radiación solar reflejada:** Es la radiación que, tras ser reflejada por el entorno de la superficie receptora, incide en una superficie inclinada. No suele medirse y su valor medio es el 20% de la radiación global incidente sobre superficie horizontal. [5]

La radiación solar sufre distintos fenómenos de reflexión y absorción acompañado de re-irradiación y difusión; que podemos esquematizar del siguiente modo:

- a) Una parte de la radiación es absorbida en modo selectivo y re-irradiada en todas las direcciones y por consiguiente con una pérdida efectiva en intensidad de la radiación incidente, responsables de este fenómeno son las moléculas del ozono O_3 a las que se debe la fuerte absorción provocada en el campo ultravioleta; y las moléculas de O_2 , H_2O , CO_2 ,

CH presente en la estratosfera, que causan fuerte absorción en el campo infrarrojo.

- b) Una parte de la radiación se propaga en todas las direcciones a causa del choque con las moléculas de gas y polvillo presentes en la atmosfera.
- c) Una parte de la radiación resulta reflejada por las nubes hacia el espacio.
- d) La parte restante llega directamente a la superficie terrestre y constituye la radiación directa al cruzar la atmosfera.

Es de importancia distinguir que además de la radiación directa, procedente del sol sin haber sufrido ningún cambio de dirección llega también al suelo parte de la radiación re-irradiada por las moléculas y de la difusa por los gases y los polvillo que en conjuntó constituyen la radiación difusa.

Esta última se encuentra siempre presente en un porcentaje del 20% cuando los días son completamente despejados, mientras que en los días de cielo cubierto es la única presente.

La cantidad de la radiación directa que llega al suelo depende de distintos factores los cuales determinan una diferente atenuación:



- a) La masa de aire circulando atenúa la radiación en la atmosfera.
- b) Las condiciones atmosféricas; estos son determinantes dado que la radiación en el suelo se reduce sensiblemente a fuertes concentraciones de humedad.
- c) El ángulo de latitud (θ) del lugar de observación, desplazándose desde el Ecuador al Polo Norte, el sol resulta cada vez más bajo sobre el horizonte por consiguiente aumenta la masa aire.

1.3.3 RADIACIÓN SOLAR VS. MOVIMIENTOS TERRESTRES

Las mayores variaciones de la radiación solar en el suelo de la tierra se deben a los movimientos de la tierra, es decir al movimiento anual de la revolución alrededor del sol y al movimiento diario de rotación. Esto se debe a las siguientes causas:

- a) Variación del ángulo de inclinación entre $+23^{\circ},44$ min y $-23^{\circ},44$ min de inclinación del eje terrestre respecto al plano de las orbitas, producido por la órbita plano-elíptico que describe la tierra alrededor del sol.
- b) La latitud del lugar a observar la variación de flujo de la energía solar produce fundamentalmente la alternancia de las estaciones y en segundo orden condiciona la evolución climática de las distintas regiones terrestres. [3],

Esto explica el porqué de la necesidad de desarrollar un prototipo que mida los niveles de radiación solar de una forma más precisa y refinada posible.

Su unidad es el W/m^2 (watio por metro cuadrado), el Sol tiene regiones totalmente inaccesibles a la observación directa y es allí donde ocurren temperaturas de unos 20 millones de grados, necesarios para producir reacciones nucleares que crean su energía.

La superficie de la fotosfera aparece conformada de un gran número de gránulos brillantes producidos por las células de convección. También aparecen fenómenos cíclicos que conforman la actividad solar como manchas solares, fáculas, protuberancias solares, etc.

Estos procesos tienen lugar en diferentes profundidades, van acompañados siempre de emisión de energía que se superpone a la principal emisión de la Fotosfera, haciendo que el Sol se aleje ligeramente en su emisión de energía del cuerpo negro a cortas longitudes de onda por la emisión de rayos X y; a largas longitudes por los fenómenos nombrados, destacando que no es la emisión igual cuando el Sol está en calma que activo. Además la Cromosfera y corona absorben y emiten radiación que se superpone a la principal fuente que es la Fotosfera.

1.3.4 RADIACIÓN SOLAR Y SUS CONSECUENCIAS

El verano y la radiación solar nos traen energía, ayudan a sintetizar vitaminas, etc., sin embargo nos pueden ocasionar quemaduras solares, daños en la dermis, foto-envejecimiento y cánceres cutáneos.

Dentro de esta radiación existen varios tipos de radiación ultravioleta, siendo la de tipo A, presente durante todo el día, la que nos produce bronceamiento y estímulo de vitamina D (importante en la prevención de osteoporosis), el abuso de esta radiación sin protección nos predispone a foto-envejecimiento y manchas.

La radiación ultravioleta B presente a partir de las 11, tiene su pico máximo entre 12:00 y 13:00, para luego disminuir a partir de las 15:00. Esta radiación puede ser carcinogénica o sea predispone a la aparición de cánceres y quemaduras solares, dependiendo la predisposición de otros factores.

Actualmente se ha visto incrementado el número de pacientes que acuden por problemas cutáneos desencadenados por el sol, como lo son las quemaduras, cáncer en la piel, foto-envejecimiento, el conocido defecto en la capa de ozono; ante estos agravantes, la prevención cobra una relevante importancia.

Los protectores solares preservan la dermis frente a la radiación solar; y su eficiencia se mide a través del valor del factor de protección solar (FPS) y de la correcta aplicación del mismo. [7]

1.4 MUESTREO

Los sensores electrónicos son empleados para la detección de valores de voltajes que están proporcionalmente relacionados a la función de repuesta, en este caso el nivel de radiación solar que este expuesto al sensor dará como salida una señal de voltaje. El fabricante nos provee en un datasheet la curva de relación Radiación Solar vs. Voltaje.

Los diseños digitales, se simulan para encontrar fallos antes de ser construidos. Aunque tales simulaciones no reproducen exactamente las señales como se las puede obtener de un prototipo real, pero cubren la mayoría de las necesidades reales en cuanto a la depuración de un programa.

1.5 CAPTURA

La interface dará una señal lógica de inicio para la adquisición de datos este guardará los datos en tablas estructuradas.

En el programa de inicio se correrá un proceso (bucle) con periodicidad de $\frac{1}{2}$ segundo.

Como esta señal es analógica, una de las limitantes del RASPBERRY PI es imprescindible convertirla en señal digital.

1.5.1 CONVERSIÓN ADC – TRANSMISIÓN Y RECEPCIÓN

El dispositivo electrónico PIC 16F88 bajo programación MikroC hará la conversión analógica a digital.

Este dispositivo tendrá la sincronización en la transmisión de la data hacia los pines GND (6), TX (8), RX (10) del puerto GPIO en el RASPBERRY PI, dicho proceso es denominado como una transmisión serial.

1.5.2 ARCHIVADO EN EL RASPBERRY PI

La mini computadora RASPBERRY PI tiene un sistema operativo derivado del LINUX que para propósitos de este dispositivo creó un sistema operativo RASPBIAN ARM 32 bits.

El equipo está programado para capturar la data, escribirla en modo de guardar en una base de datos MYSQL que es un programa Open Source.

1.6 PRESENTACIÓN WEB DE LA INFORMACIÓN

Los grandes avances tecnológicos nos ponen a la disposición los medios necesarios para la transmisión y recepción remota de datos por medio del internet.

El RASPBERRY PI operará los datos y tendrá la configuración de un servidor de web el cual permitirá el acceso desde cualquier lugar del mundo a los reportes del sistema en captura de datos.

1.7 LIMITACIONES

Una de las limitantes en este trabajo es obtener la calibración del equipo, ya que se debe determinar un máximo y un mínimo en la lectura de la celda solar haciendo mediciones con otro equipo que midan niveles de radiación solar el cual no se posee por ser muy costosos además de depender de las condiciones climatológicas idóneas para adquirir estos datos.

El hecho de que nuestra celda solar permanezca siempre fija en posición horizontal hace que nuestra medición pierda precisión ya que solo captamos los niveles de radiación solar directo y no los difusos que son los que se esparcen debido a las partículas de la atmosfera.



Es importante tener en cuenta que el fotodiodo posee un área de captura alrededor de los mm^2 , esto implica en realizar la conversión por aproximación a m^2 con el mínimo de errores por estimación.

El microcomputador RASPBERRY PI es relativamente un nuevo dispositivo en el mercado, esto hace que la investigación del mismo sea muy exhaustivo debido a que no hay mucha información específicas disponible en internet solo contamos con un gran número de investigadores que crean foros de discusión sobre diferente proyectos con el uso del RASPBERRY PI.

Todos los componentes utilizados son de adquisición por importación, este proceso incurre en un tiempo de espera que afecta en la realización oportuna del proyecto.



CAPÍTULO 2

FUNDAMENTO TEÓRICO

2.1 REQUERIMIENTOS DEL PROYECTO

En este proyecto se involucran dos ramas tecnológicas de primer nivel, como lo es Software y Hardware.

SOFTWARE:

- **PIC C (CCS).**- para programar y compilar el PIC 16F88 es de MIKRO
- **RASPBIAN.**- para el RASPBERRY PI procede del (Debian » Linux).
- **PHP- MYSQL.**- manejador de base de datos, crea, modifica, elimina tablas de datos.
- **PHP5-MYSQL.**- Este paquete proporciona módulos para las conexiones de base de datos MySQL directamente desde scripts PHP.

- **PHPMYADMIN.-** Herramienta escrita en PHP el cual maneja la administración de MySQL en el entorno HTLM. Crea, elimina y edita Tablas y Bases de Datos.
- **RRD TOOL.-** Es el acrónimo de *Round Robín Data base Tool*. Esta técnica trabaja con una cantidad de datos fija, definida en el momento de crear la base de datos, y un puntero al elemento actual.
- **RRDTOOL-MYSQL.-** Es una librería que se requiere para interactuar con los datos de un motor como el MySQL y generar gráficos con respecto al tiempo (monitoreo).
- **SERVIDOR-WEB APACHE.-** Servicio permite trabajar en http.
- **WIRING PI.-** Es la librería necesaria en Linux para manejar los puertos GPIO del RASPBERRY PI.

HARDWARE:

- **PIC 16F88.-** microcontrolador convertidor ADC.
- **BPW34.-** fotodiodo electrónico.
- **RASPBERRY PI.-** minicomputador.

2.2 HERRAMIENTAS DE SOFTWARE

2.2.1 CCS COMPILADOR DE PIC LENGUAJE C

Este es un programa diseñado para el desarrollo práctico de aplicaciones para microcontroladores en Lenguaje C el ambiente de desarrollo integrado de PCWHD le da a los programadores la capacidad de producir rápidamente códigos muy eficientes usando un lenguaje fácil y manejable, es un software libre de licencia pero de alto nivel.

EL CCS IDE incluye un poderoso editor de C, debugger, líneas de comando PCB, PCM, PCH y PCD, un terminal de comunicación serial perfecto, generación de estadísticas y más. Usted será capaz de compilar códigos para las familias:

- 12-bit PIC®MCU (PIC10F200, PIC16C55, etc.)
- 14-bit PIC®MCU (PIC16F877A, PIC16F88, etc.)
- 16-bit PIC18 MCU (PIC18F452, PIC18F8722, etc.)
- 24-bit PIC24/ds PIC®MCU (dsPIC30F2010, dsPIC30F5011, etc.).

Usted podrá integrar PCWHD al MPLAB*IDE u otros ambientes.

Sistema operativo: Windows (XP, Vista, Windows 7) x86 & x64

Este entorno de desarrollo cuenta con una amplia variedad de características tales como IDE fácil de usar, muy compacta y de eficiente código, hardware y software de bibliotecas, documentación completa, simulador de software, soporte de hardware depurador. [8]

2.2.2 LINUX

Es uno de los tantos flavors de Unix. Se trata de un sistema operativo de 32 bits de libre distribución, desarrollado originalmente por Linus Torvalds, un estudiante de la universidad finlandesa de Helsinki, quien, en 1991, se abocó a la tarea de reemplazar a Minix, un clon de Unix de pequeñas proporciones.

A medida que avanzaba en su desarrollo, Linux fue dejando el código fuente de las sucesivas versiones del kernel y utilidades de Linux a disponibilidad de los usuarios de Internet. Este fue sin duda un gran acierto, ya que hizo posible que una multitud de desarrolladores de todo el mundo se familiarizaran con el código, lo cual en primera instancia significó un gran aporte de sugerencias, evolucionando luego hacia un espectacular ejemplo de desarrollo distribuido de software: centenares de desarrolladores independientes tomaron a su cargo la producción de software para Linux, ya sea escribiéndolo desde cero o portándolo desde otras plataformas Unix.



Esta modalidad de desarrollo continúa aún y ha permitido a Linux alcanzar un alto nivel y madurez, así como también un amplio grado de aceptación.

Actualmente, Linux posee todas las características que pueden encontrarse en cualquier sistema Unix moderno, incluyendo direccionamiento lineal de 32 bits, memoria virtual, multitarea real, shared libraries, módulos de kernel cargables on-demand, soporte TCP/IP (incluyendo SLIP, PPP, NFS, etc.), y sistema X-Windows (incluyendo servidores X para los adaptadores de video más populares, y clones de Motif, OpenLook, NextStep y Windows95 como Windows managers). [9]

2.2.3 DEBIAN

Es un sistema operativo (S.O.) libre para su computadora. El sistema operativo es el conjunto de programas básicos y utilidades que hacen que funcione su computadora.

Debian ofrece más que un S.O. puro; viene con 29000 paquetes, programas pre-compilados distribuidos en un formato que hace más fácil la instalación en su computadora. [10]

2.2.4 RASPBIAN

Raspbian es un sistema operativo libre basado en Debían, optimizado para el hardware Raspberry Pi. Sin embargo, Raspbian ofrece más que un sistema operativo puro; viene con 35000 paquetes, programas pre-compilados desarrollados en un formato amigable para una instalación sencilla en su Raspberry Pi.

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

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

Destaca también el menú "raspi-config" que permite configurar el sistema operativo sin tener que modificar archivos de configuración manualmente.

Entre sus funciones, permite expandir la partición root para que ocupe toda la tarjeta de memoria, configurar el teclado, aplicar overclock, etc. **[10]**

2.2.5 PROTEUS

Este software ofrece una amplia gama de prestaciones al realizar diferentes tipos de proyectos electrónicos, en todas sus etapas: diseño, simulación, depuración y construcción; ya sea analógica, digital o combinada, usando diferentes componentes electrónicos, como también puertos existentes en el mercado, herramientas de medición como multímetros, osciloscopios, analizadores lógicos y demás.

Esta herramienta nos permite simular y depurar el funcionamiento de todo el sistema paso a paso, de tal manera que podremos observar el contenido de registros y diferentes posiciones de memoria, que nos facilitará determinar si la respuesta del hardware es la correcta o existen errores.

Proteus posee una herramienta llamada ARES que la utilizamos para el diseño de circuitos impresos (PCB) que se puede generar automáticamente a partir de nuestro diseño. **[11]**

2.2.6 APACHE

Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Apache es una tecnología gratuita de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podremos saber, sin ningún secreto. Apache es un servidor altamente configurable de diseño modular. Lo importante es que cualquiera que posea una experiencia programación de C o Perl puede escribir un módulo para realizar una función determinada.

Apache admite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. [12]

2.2.7 MySQL

Es un sistema de base de datos operacional, una de las características más interesantes de MySQL es que podemos recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos. Por otro lado,

MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores. Las plataformas que utiliza son de variado tipo y entre ellas podemos mencionar LAMP, MAMP, SAMP, BAMP y WAMP (aplicables a Mac, Windows, Linux, BSD, Open Solaris, Perl y Python entre otras).

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM. Sea cual sea el entorno en el que se va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación. **[13]**

2.2.8 PHP

Es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación, que se incrustan dentro del código HTML.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante

de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. [13]

2.3 HERRAMIENTAS DE HARDWARE

2.3.1 PIC 16F88

Trata de uno de los microcontroladores más populares del mercado actual, ideal para principiantes, debido a su arquitectura de 8 Bits, 18 pines, y un set de instrucciones RISC muy amigable para memorizar y fácil de entender, internamente consta de:

- Memoria Flash de programa (1K x 14).
- Memoria EEPROM de datos (64 x 8).
- Memoria RAM (68 registros x 8).
- Un temporizador/contador (timer de 8 bits).
- Un divisor de frecuencia.
- Varios puertos de entrada-salida
- 13 pines en dos puertos, 5 pines el puerto A y 8 pines el puerto B.
- Manejo de interrupciones (de 4 fuentes).
- Perro Guardián (WATCHDOG TIMER).
- Modo Sleep.
- Frecuencia de reloj externa máxima 10MHz. La frecuencia de reloj interna es un cuarto de la externa. así pues se ejecutan 5 Millones de Instrucciones por Segundo (5 MIPS)



- Pipe-line de 2 etapas, 1 para búsqueda de instrucción y otra para la ejecución de la instrucción (los saltos ocupan un ciclo más).
- Repertorio de instrucciones reducido (RISC), 30 instrucciones distintas.

En los últimos años se ha popularizado el uso de este microcontrolador debido a su bajo costo y tamaño. Se ha usado en numerosas aplicaciones, que van desde los automóviles a decodificadores de televisión. Es muy popular su uso por los aficionados a la robótica y electrónica.

Puede ser programado tanto en lenguaje ensamblador como en Basic y principalmente en C, donde existen numerosos compiladores. Cuando se utilizan los compiladores BASIC, es posible desarrollar útiles aplicaciones en tiempo récord, especialmente dirigida al campo educacional. **[14]**

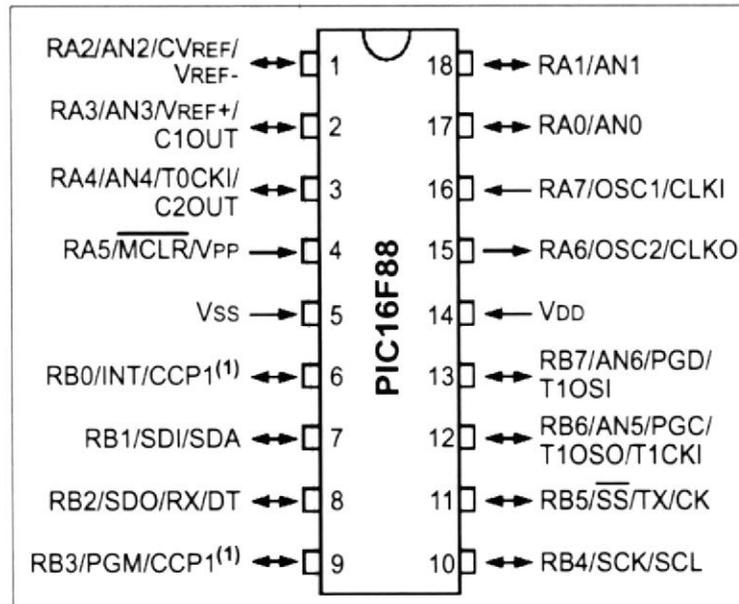


Figura 2.1 Configuración de pines PIC 16F88 [14]

2.3.2 SENSOR SOLAR BPW34

El BPW34 es un diminuto fotodiodo Pin de propósito general. Este fotodiodo tiene una gran cantidad de usos, uno de ellos es como una celda solar para brindar alimentación a un proyecto. La celda es sensible a un gran rango de longitudes de onda de luz (430-1100nm). El voltaje de circuito abierto es de 350mV (900nm, con una fuente de luz de 1mW/cm²), y la corriente de corto circuito es de 47μA.

Al iluminar a un BPW34 con un LED infrarrojo de 940nm se puede producir 0.5VDC en circuito abierto. Colocando cuatro de ellas en serie es posible

encender un LED. Esta mini-celda también puede generar un pequeño voltaje ~250mV al ser colocada en un cuarto iluminado con luz fluorescente.

Características:

- Superficie sensible a la luz (en mm²): 7.5.
- Alta foto-sensibilidad.
- Sensible a luz visible e infrarroja.
- Rápidos tiempos de respuesta.
- Ángulo de media sensibilidad: $\phi = \pm 65^\circ$.
- RoHS. [15]



Figura 2.2 Sensor BPW34 [15]

2.3.3 RASPBERRY PI

La Raspberry es una computadora de reducidas dimensiones capaz de realizar muchas tareas que tu PC puede lograr como hojas de cálculo, procesamiento de palabras y juegos. También posee salida de video HD; además de correr diferentes versiones de Linux.

Dimensiones: 85.60mm x 56mm x 21mm.

Características:

- Broadcom BCM2835 SoC.
- 700 MHz ARM1176JZF-S core CPU.
- Broadcom Video Core IV GPU.
- 512 MB RAM.
- 2 x USB2.0 Puertos.
- Video: Compuesto (PAL y NTSC), HDMI o Raw LCD (DSI).
- Audio: 3.5mm Jack o Audio sobre HDMI.
- Almacenamiento: SD/MMC/SDIO.
- 10/100 Ethernet (RJ45).
- 8 x GPIO - UART.
- 12C BUS
- SPI BUS con dos Chips selects.
- +3.3 VDC / + 5 VDC / GROUND.

- Requerimientos: 5V @ 700 mA via Micro USB o Header GPIO.
- Soporta Debian GNU/Linux, Fedora, Arch Linux, RISC OS [16]

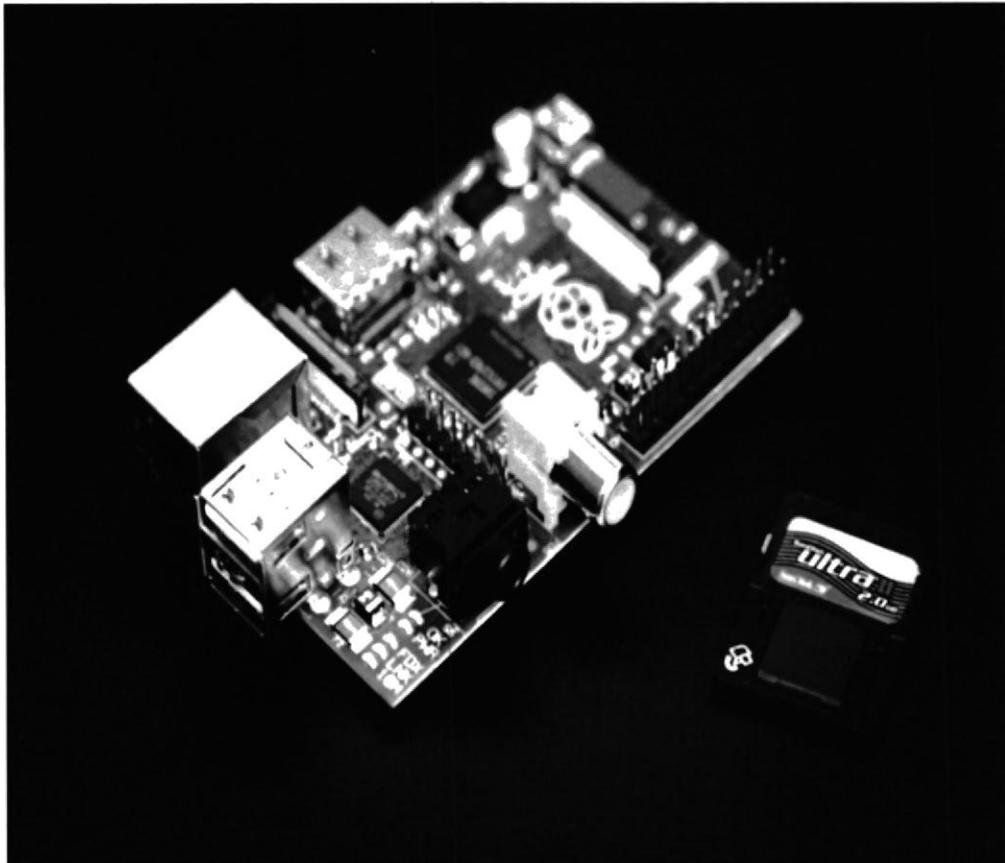


Figura 2.3 RASPBERRY PI Modelo B [16]

2.3.4 PICKIT 3

El PICKIT 3 es un programador fabricado por Microchip© para programar toda su línea de micro-controladores PIC's® desde los PIC10, PIC12, PIC14, PIC16, PIC18, PIC24, dsPIC30 y dsPIC33 (todos los micro-controladores con memoria Flash sin excepción). El programador fue diseñado para programar

los microcontroladores en circuito (ICSP); lo que significa que puede programar los microcontroladores montados directamente en tu aplicación y/o protoboard sin necesidad de tener que sacarlo y meterlo cada vez que se modifica el programa. [17]



Figura 2. 4 Hardware y Software Pickit 3 [17]

CAPITULO 3

DESARROLLO DEL PROYECTO

3.1 RESUMEN

En este capítulo mostraremos el desarrollo de nuestro proyecto, los pasos generales para llevar a cabo el ensamblaje del prototipo con el fin de entender el funcionamiento de cada uno de los componentes utilizados y mostrar cómo han sido combinados para alcanzar el objetivo propuesto.

Para esto empezaremos con la programación del PIC 16F88 con el uso del software PIC C (CCS) tomaremos los datos analógicos del sensor solar para convertirlos en digital con una periodicidad de 10 segundos, recibiendo la data en el RASPBERRY PI por medio del puerto GPIO con la programación,

configuración y utilizando los comando de consola del RASPBIAN almacenaremos estos datos en su memoria.

En un browser de navegación presentar el dato actual adquirido y/o la gráfica de los datos por periodos de tiempo, para mejor comprensión dividiremos en varias partes el proyecto mostrados como ejercicios individuales.

3.1.1 DIAGRAMA DE MODULOS GENERAL

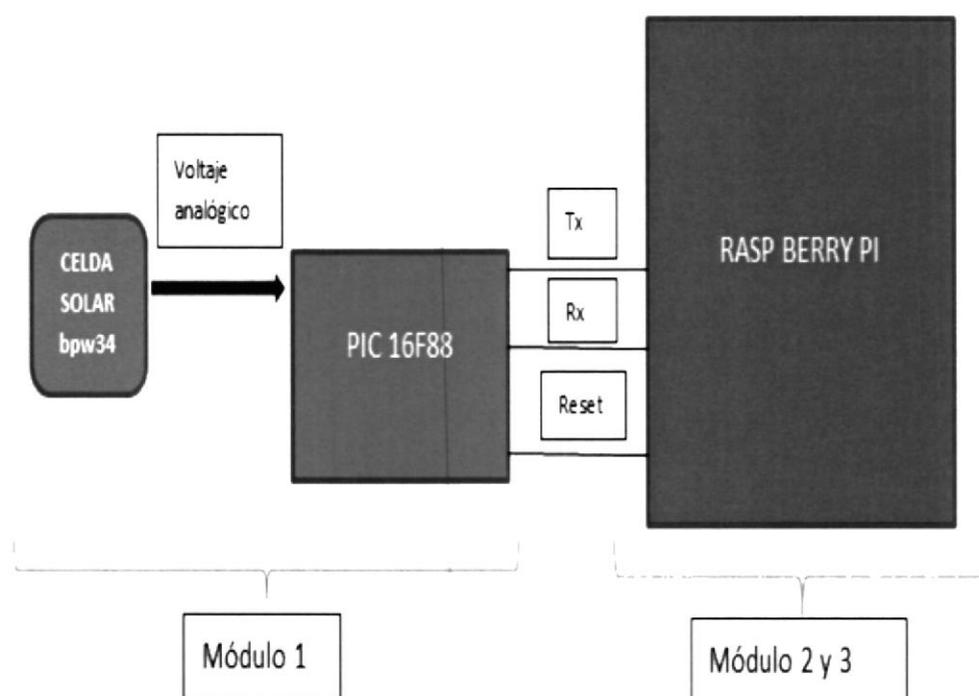


Figura 3.1 Diagrama de módulos general

3.1.2 DIAGRAMA DE COMUNICACIÓN MODULOS 1,2 Y 3

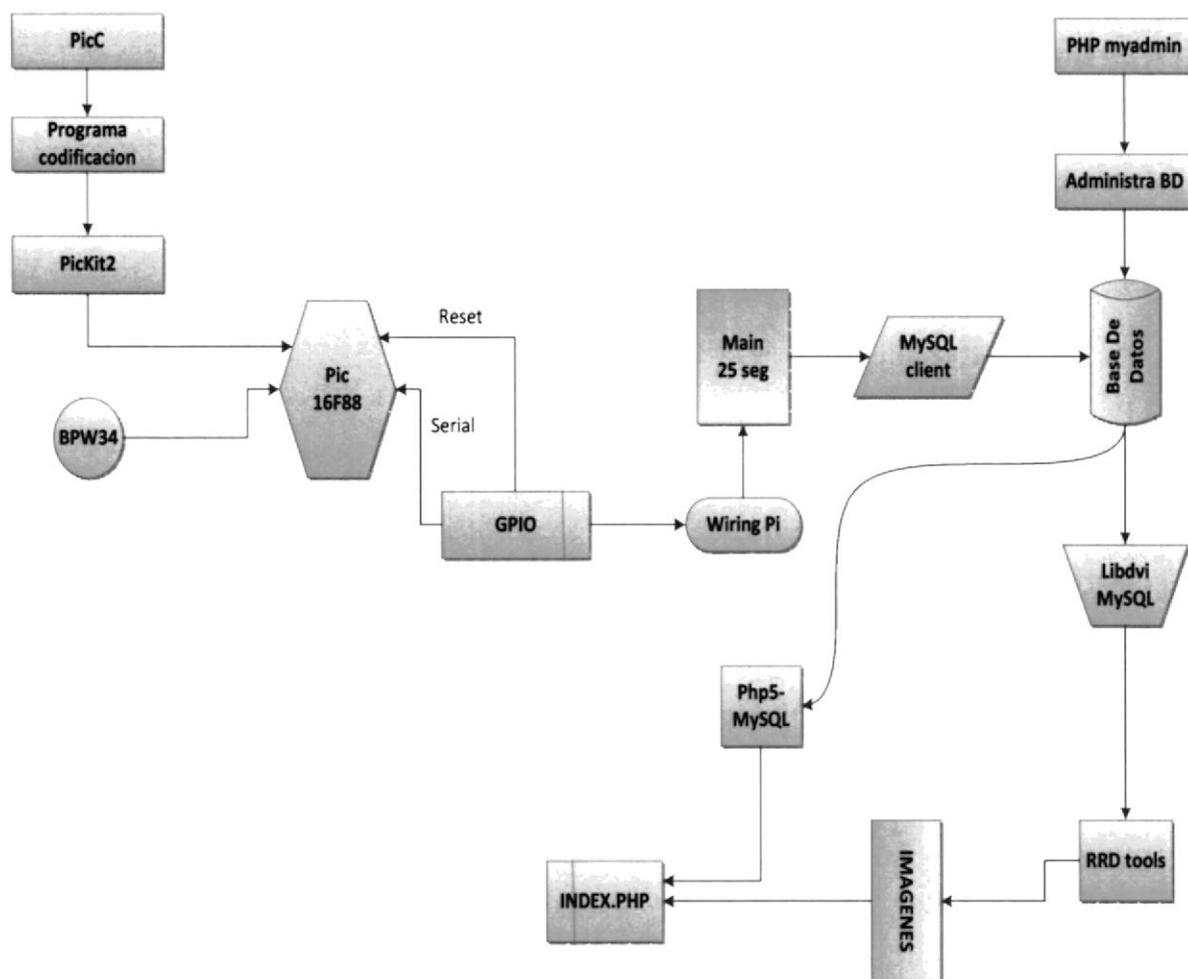


Figura 3.2 Diagrama comunicación en los módulos 1, 2 y 3

3.2 MODULO 1: PROGRAMACION PIC 16F88

Para la medición de la radiación solar usamos el fotodiodo BPW34 este sensor nos proporciona como dato análogo 1uA por cada 0.01 mW/cm² empleando el convertidor ADC tendremos su equivalente en digital.

3.2.1 ALGORITMO PROGRAMACION PIC 16F88

1. Iniciamos ADC con ayuda del RASPBERRY PI.
2. Configuramos los puertos de salida PIC 16F88.
3. Configuramos Led indicador de muestreo.
4. Habilitamos puertos ADC.
5. Leemos el ADC y esperamos la conversión 10 bits.
6. Deshabilitamos la lectura ADC.
7. Transmitimos los datos por el puerto serial Tx.
8. Prendo Led Indicador anunciando salida al RASPBERRY PI.

3.2.2 DIAGRAMA DE FLUJO PROGRAMACION PIC16F88

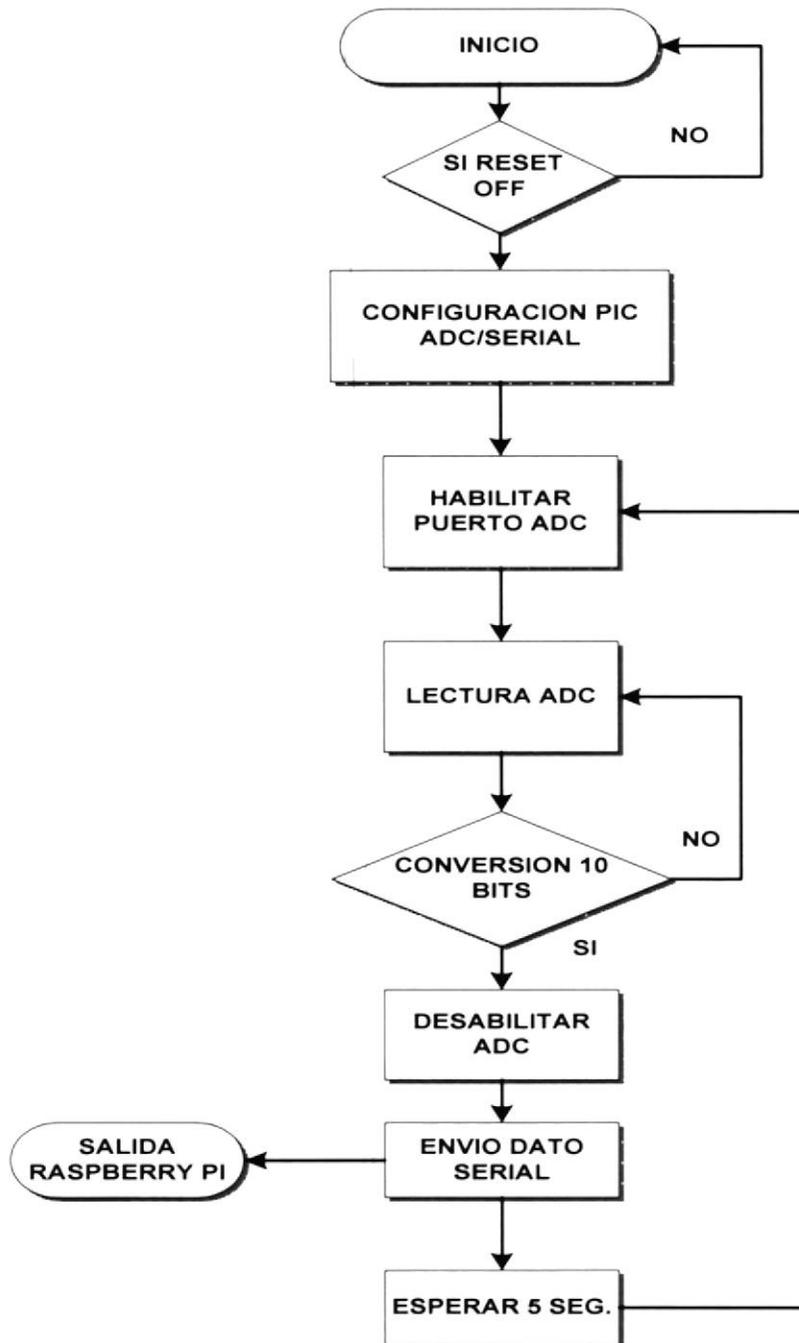


Figura 3.3 Secuencia conversi3n adc

3.2.3 CODIGO FUENTE DEL PIC16F88

```
//***** LIBRERIAS A USAR *****  
  
#include "test-adc.h"  
  
#include <16F88.h>  
  
//***** ENCABEZADO ESTANDAR PARA PIC16F88 *****  
  
#device PIC16F88  
  
#list  
  
#device adc=10  
  
// ***** INICIALIZACION ADC *****  
  
void main()  
{  
  
    int16 adc_value;  
  
    float volts;  
  
    setup_adc_ports(sAN0);  
  
    setup_adc(ADC_CLOCK_DIV_8);  
  
    set_adc_channel(0);  
  
    delay_us(20);  
  
// ***** LECTURA ADC *****  
  
    while(1)  
    {  
  
        read_adc(ADC_START_ONLY);  
  
        adc_value = read_adc(ADC_READ_ONLY);  
  
    }  
  
}
```

```
    volts = (float)(adc_value * 5)/1023.0;

    volts = (volts*1000);

    printf( "%4.2f mV \n\r", volts);

    output_low(PIN_B1);

    delay_ms(500);

// ***** ENVIO PUERTO TRANSMISION TX *****

    output_high(PIN_B1);

    delay_ms(500);

}

}
```

3.3 MODULO 2: PROGRAMACION MAIN.C

Para la adquisición de la data desde el PIC 16F88 detallamos:

- COMUNICACIÓN SERIAL
- CONEXIÓN A LA BASE DE DATOS
- CONFIGURACION AL PUERTO GPIO
- TIEMPO PERMANENCIA HISTORIAL



3.3.1 DIAGRAMA DE FLUJO: MODULO GENERAL

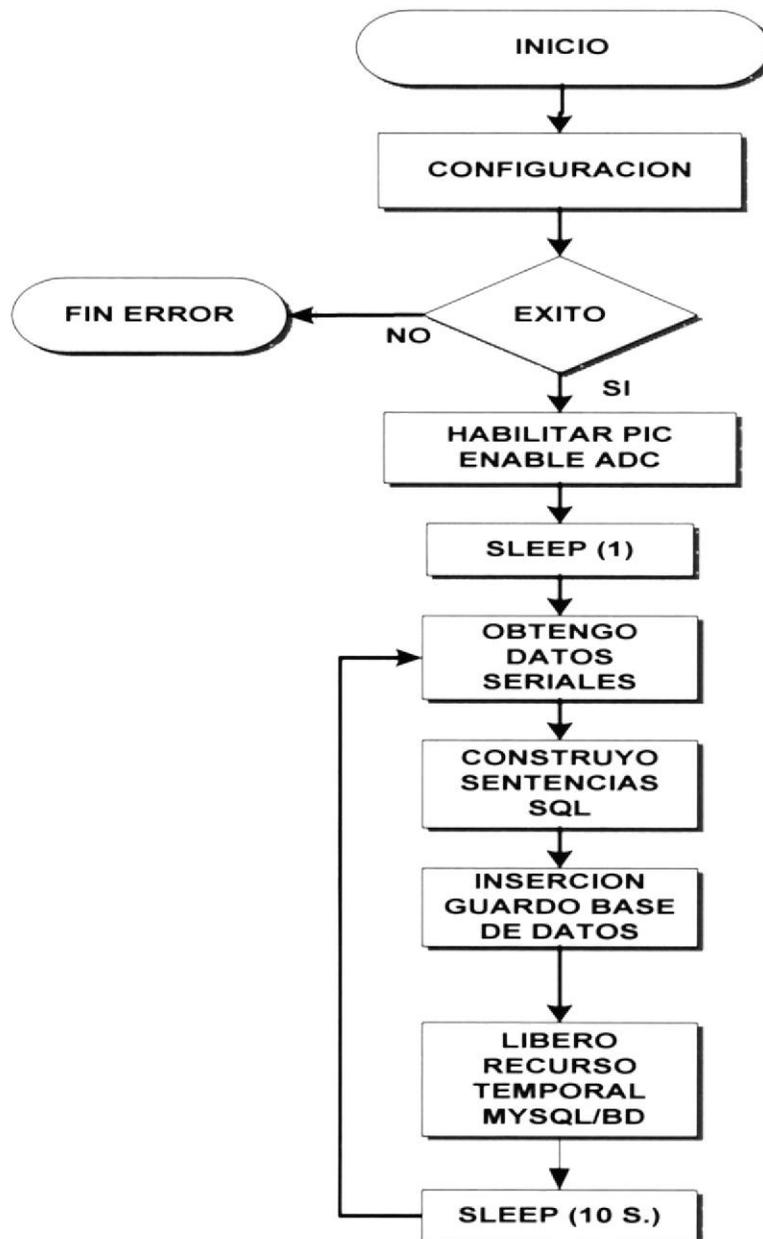


Figura 3.4 Diagrama de flujo modulo general



3.3.2 DIAGRAMA DE FLUJO: COMUNICACIÓN SERIAL

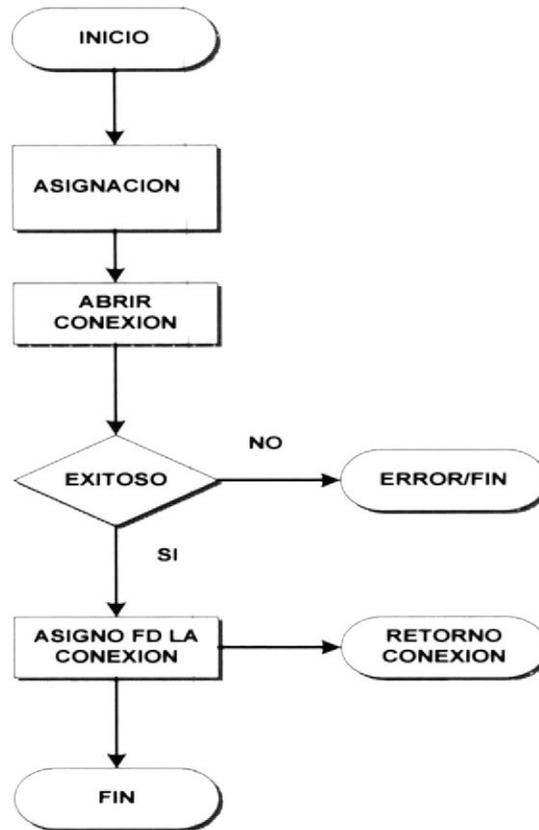


Figura 3.5 Diagrama de flujo comunicación serial

3.3.3 ALGORITMO COMUNICACIÓN SERIAL

- INICIALIZACIÓN DE LOS PUERTOS DE COMUNICACIÓN
- ASIGNACIÓN DE IDENTIFICACION PUERTO TTY
- ASIGNAR VELOCIDAD DE TRANSMISION BAUD=9600
- TRANSMITIR PAQUETE DE DATOS
- VERIFICACION DE ÉXITO EN TRANSMISION
- RETORNO A CONEXION

3.3.4 DIAGRAMA DE FLUJO: CONEXIÓN BASE DE DATOS

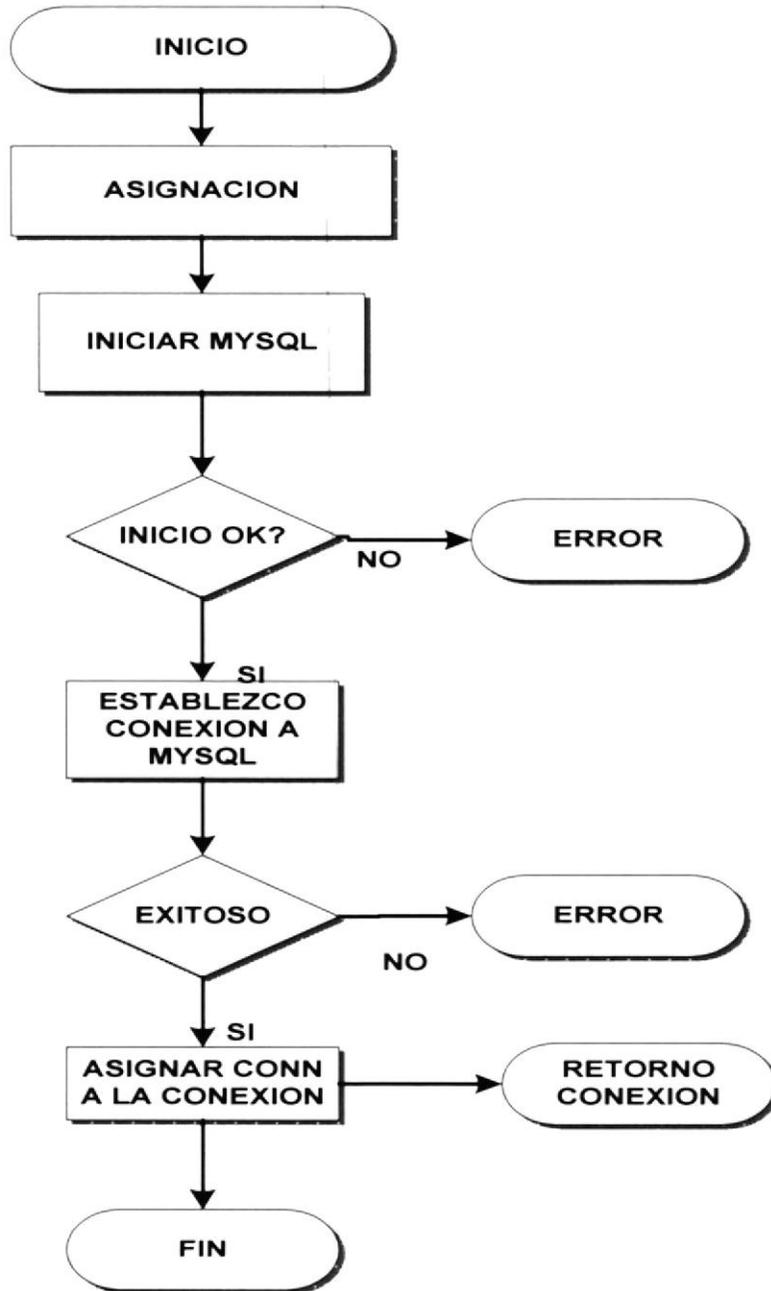


Figura 3.6 Diagrama de flujo conexión base de datos

3.3.5 ALGORITMO COMUNICACIÓN SERIAL

- ASIGNACION CONTROL DE ACCESO A BASE DE DATOS.
- SERVER = LOCALHOST 127.0.0.1 - USER = ROOT.
- PASSWORD = SOLAR123 - DATABASE = ADCDB.
- INICIALIZACION MYSQL.
- VALIDACION DE LAS SEGURIDADES PARA LA CONEXIÓN.
- RETORNO CONEXIÓN.

3.3.6 DIAGRAMA DE FLUJO: ENABLE ADC

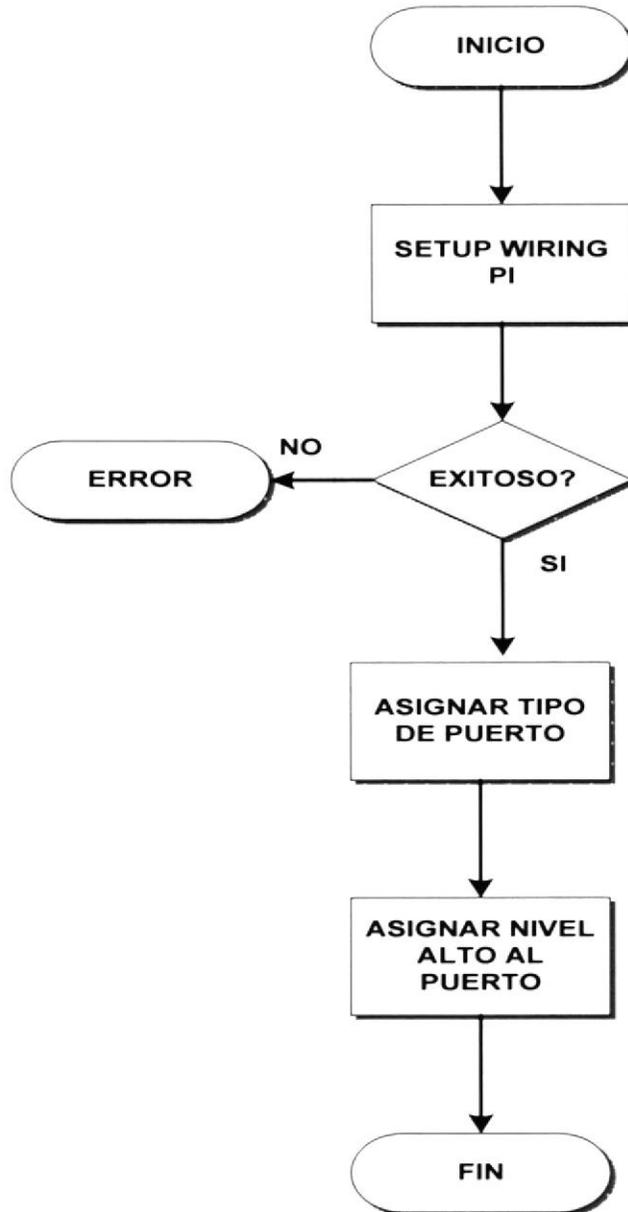


Figura 3.7 Diagrama de flujo enable adc

3.3.7 ALGORITMO COMUNICACIÓN SERIAL

- INICIALIZACION DEL ADC CON EL RASPBERRY PI.
- ASIGNACION DE MEMORIA RAM/ROM RASPBERRRY PI.
- ACCESO HARDWARE GPIO
- CONFIGURACION PUERTO I/O PINES TX – RX.
- ASIGNACION NIVEL ALTO PUERTO DE SALIDA.



3.3.8 DIAGRAMA DE FLUJO: EJECUTAR QUERY

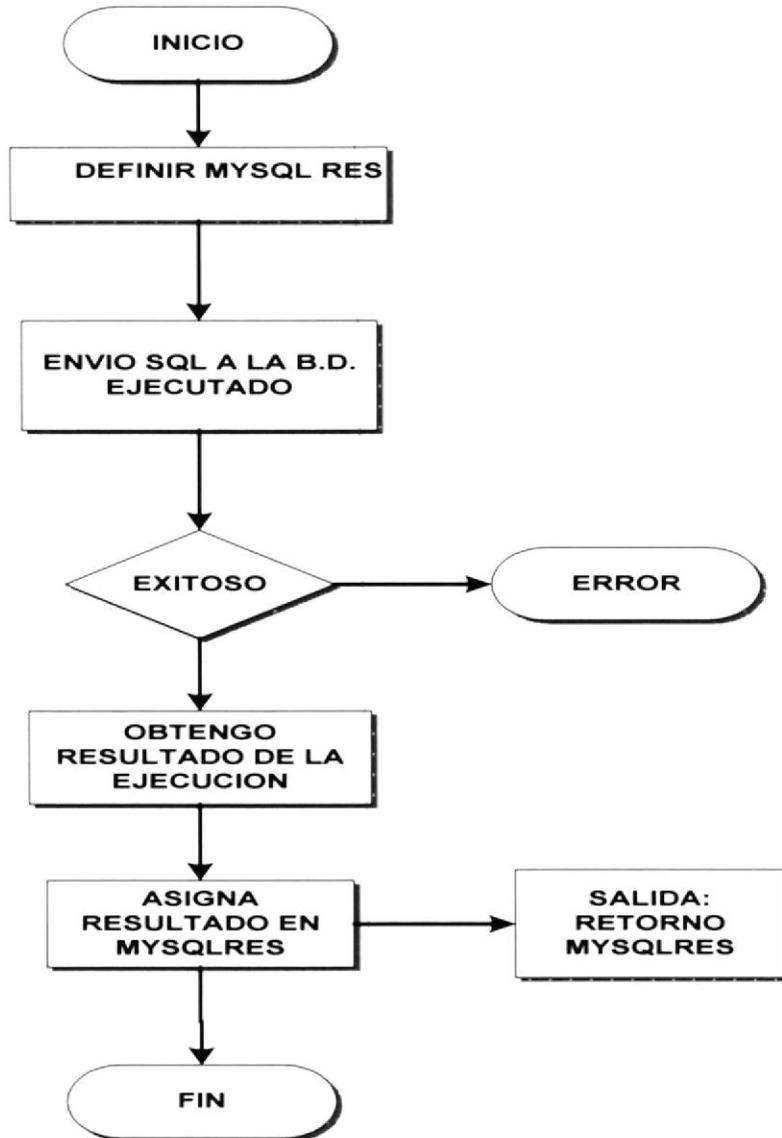


Figura 3.8 Diagrama de flujo ejecutar query

3.3.9 DIAGRAMA DE FLUJO: GETSERIAL DATA

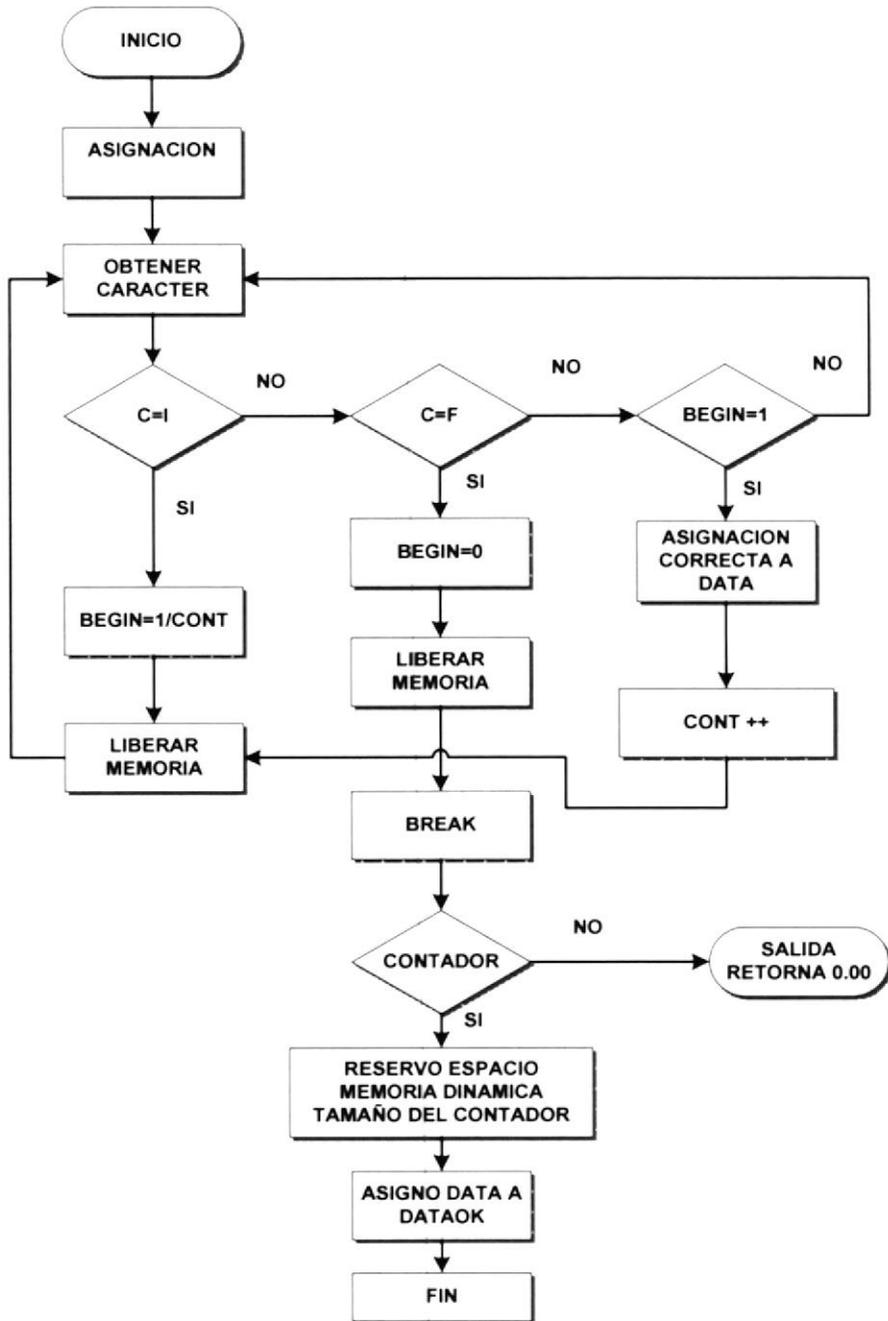


Figura 3.9 Diagrama de flujo get serial data



3.3.10 ALGORITMO COMUNICACIÓN SERIAL

- ASIGNAR VECTOR DATA [6] BEGIN=0, CONTA=0, DATAOK="0.00".
- RECIBO CARÁCTER QUE INDICA PRINCIPIO Y FINAL DEL DATO.
- MANEJO DE MEMORIA DINAMICA.
- CAPTURA / ALMACENAMIENTO DE LA DATA.

3.3.11 DIAGRAMA DE FLUJO CREAR IMAGENES

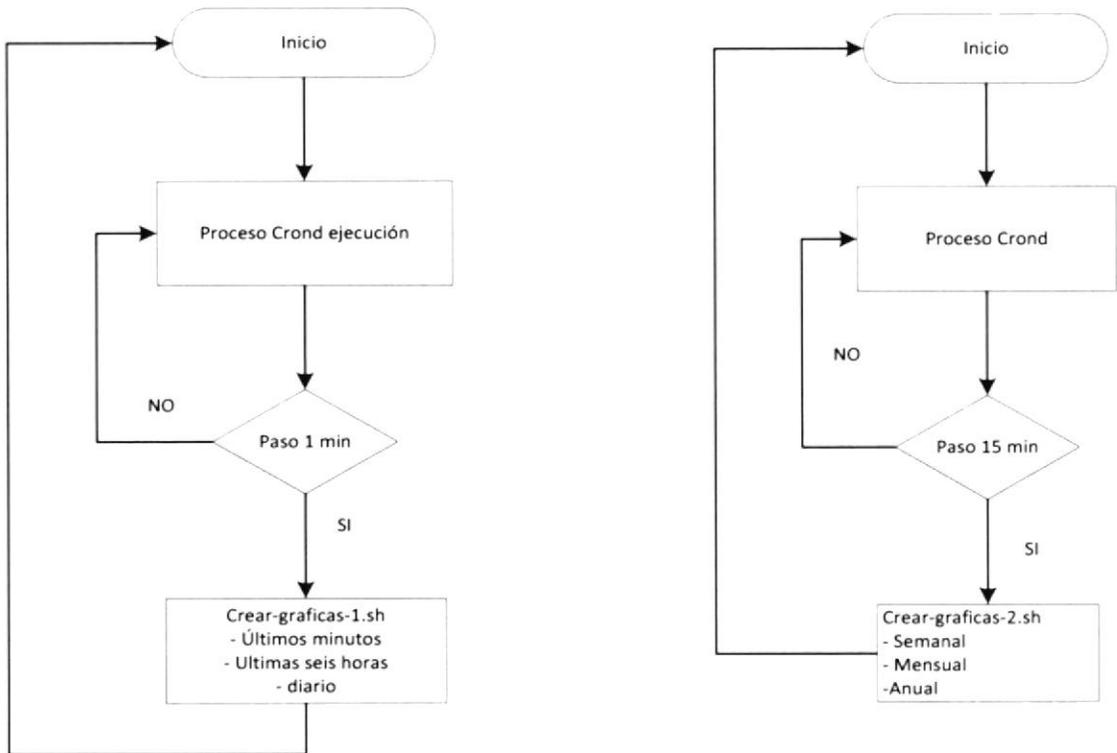


Figura 3.10 Diagrama de flujo crear imágenes

3.3.12 DIAGRAMA DE FLUJO VISUALIZAR IMAGENES

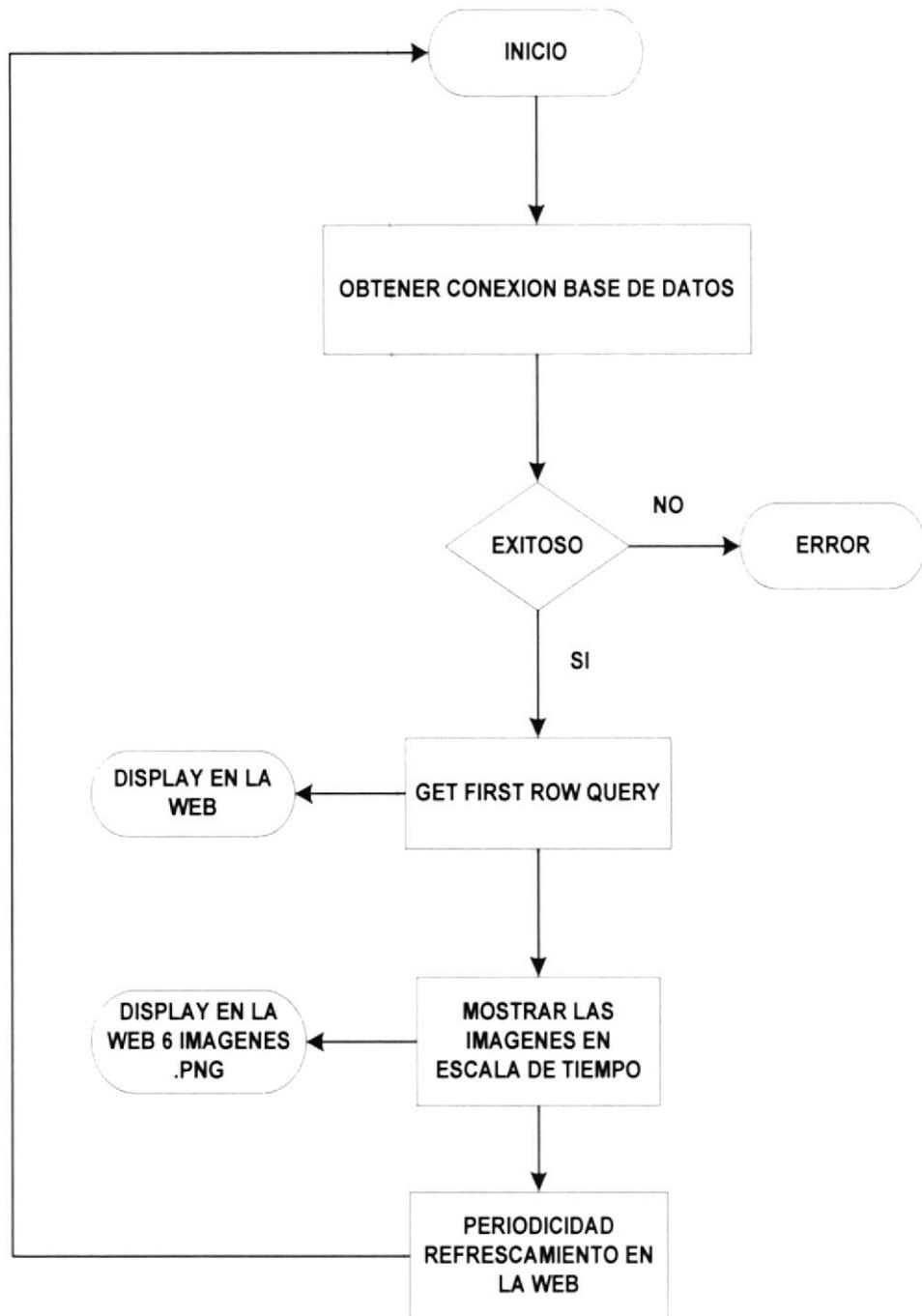


Figura 3.11 Diagrama de flujo visualizar imágenes

3.4 CODIGO FUENTE MAIN.C

```
/** Simple C program that connects to MySQL Database server */  
  
#include <mysql.h>  
  
#include <stdio.h>  
  
#include <stdlib.h>  
  
#include <wiringPi.h>  
  
#include <wiringSerial.h>  
  
#include <string.h>  
  
#include <errno.h>  
  
MYSQL * getMySQLConnect();  
  
MYSQL_RES *execQuery(MYSQL *conn, char *sql, int len);  
  
int getSerialConnect();  
  
char *getSerialData(int fd);  
  
void enableADC();  
  
void disableADC();  
  
int main()  
{  
  
    int    fd;  
  
    MYSQL  *conn;  
  
    MYSQL_RES *res;  
  
    char   *data;  
  
    char   *stat;
```



```
int    len;

char   query[100];

fd = getSerialConnect();

if(fd < 0) exit(1);

conn = getMySQLConnect();

if(!conn) exit(1);

enableADC();

sleep(1);

while(1){

    data = getSerialData(fd);

    disableADC();

    stat = "INSERT INTO muestra(valor) VALUES(%s)";

    len = snprintf(query,strlen(stat)+strlen(data)-1,stat,data);

    printf("Muestra: %s m\n",data);

    res = execQuery(conn,query,len);

    stat = "UPDATE muestra_actual set fecha=CURRENT_TIMESTAMP,
valor=%s";

    len = snprintf(query,strlen(stat)+strlen(data)-1,stat,data);

    res = execQuery(conn,query,len);

    mysql_free_result(res);

    sleep(25);

    enableADC();
```

```
}  
  
mysql_close(conn);  
  
}  
  
/***** FUNCIONES PARA MYSQL *****/  
  
MYSQL_RES *execQuery(MYSQL *conn, char *sql, int len)  
{  
    MYSQL_RES *res; /* send SQL query */  
    if (mysql_real_query(conn,sql,len)) {  
        fprintf(stderr, "%s\n", mysql_error(conn));  
        return NULL;  
    }  
  
    res = mysql_use_result(conn);  
  
    return res;  
}  
  
MYSQL *getMySQLConnect()  
{  
    MYSQL *conn = NULL;  
  
    char *server = "localhost";  
  
    char *user = "root";  
  
    char *password = "solar123";  
  
    char *database = "adcdb";  
  
    /* Connect to database */
```

```

conn = mysql_init(NULL);

if (!mysql_real_connect(conn,server,user,password,database,0,NULL,0)){

    fprintf(stderr, "%s\n", mysql_error(conn));

    return NULL; }

return conn;}

/***** FUNCIONES PARA RS232/SERIAL *****/

int getSerialConnect()

{

    int fd    = -1;

    char *device = "/dev/ttyAMA0";

    int baud   = 9600;

    if((fd = serialOpen (device,baud)) < 0){

        fprintf(stderr, "Unable to open serial device: %s\n", strerror (errno));

        return -1;

    }

    return fd;

}

char *getSerialData(int fd)

{

    int c;

    char data[6];

    char *dataok = "0.00";

    int begin = 0;

```

```
int conta = 0;

int i = 0;

while(1){

    c = serialGetchar(fd);

    if(c == 'I'){

        begin = 1;

        conta = 0;

    }

    else if(c == 'F'){

        begin = 0;

        fflush(stdout);

        break;

    }

    else if(begin == 1){

        data[conta] = c;

        conta++;

    }

    fflush(stdout);

}

if(conta > 0){

    dataok = NULL;

    dataok = (char *)calloc(conta,sizeof(char));
```



```
for(i=0; i<conta; i++)
    dataok[i]=data[i];
}
return dataok;
}
/***** FUNCION PARA EL RESET *****/
void enableADC()
{
    if(wiringPiSetup() == -1){
        fprintf(stderr, "%s\n", "wiringPiSetup error");
        exit(1);
    }
    pinMode(4,OUTPUT) ;
    digitalWrite(4,HIGH) ;
}
void disableADC()
{
    if(wiringPiSetup() == -1){
        fprintf(stderr, "%s\n", "wiringPiSetup error");
        exit(1);
    }
    pinMode(4,OUTPUT) ;
    digitalWrite(4,LOW) ;
}
```

CAPITULO 4

SIMULACION, PRUEBAS Y EVALUACION

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. Detallaremos los procesos con el uso de simuladores, creación de base de datos, sincronización flujo de datos y creación de imágenes con el uso de tablas estructuras.

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

4.2 PRÁCTICA 1: USO SIMULADOR PROTEUS PARA CAPTURAR DATOS EN EL PROTOTIPO.

El prototipo fue implementado en un simulador PROTEUS para probar con eficiencia la adquisición de la data y observar la consistencia en la operación de los códigos de programación.

Notamos que en el PORT A tenemos una entrada de voltaje regulado por un potenciómetro para simular a la fotocelda BPW34. Todo el sistema esta sincronizado para que transmita la data cada 25 segundos donde el diodo led será el piloto de la transmisión.

Visualizamos la salida de la data en el Virtual Terminal el cual representa una cadena de datos.

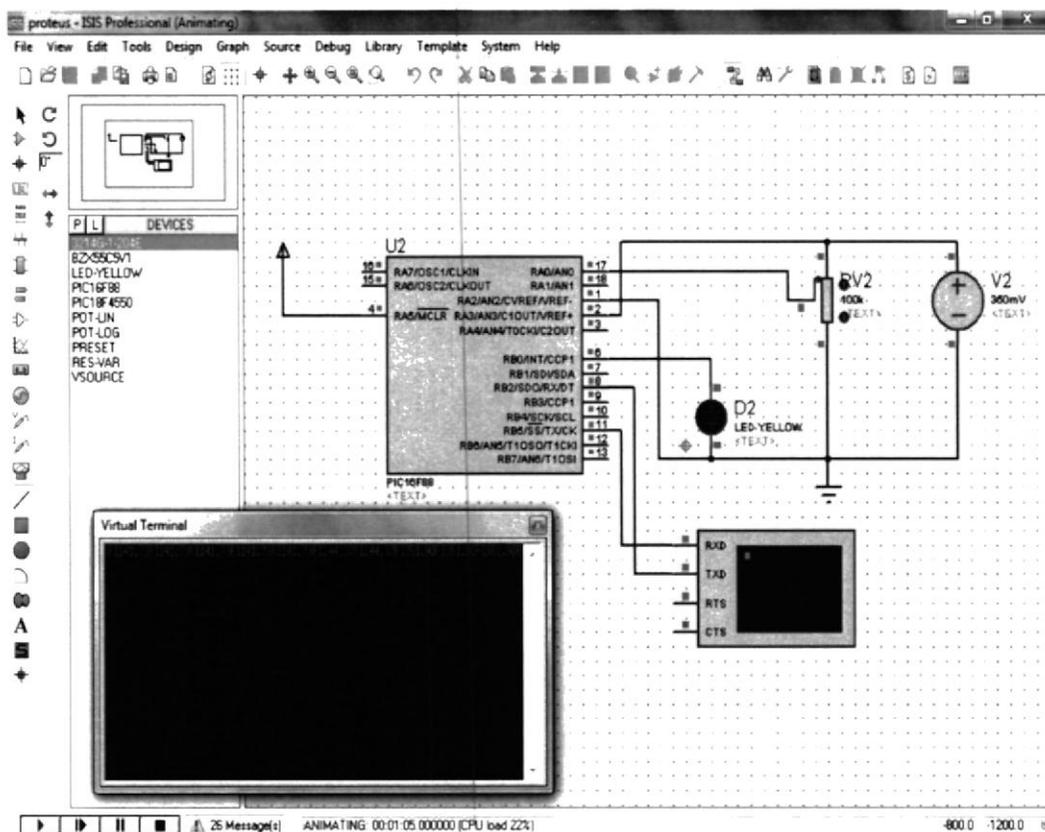


Figura 4.1 Obtención de datos capturados simulación PROTEUS

4.2.1 IMPLEMENTACION EN PROTOBOARD

En este ensayo armamos el circuito en el protoboard el cual nos permitió aprender con precisión la conversión analógica a digital, la transmisión de la información capturada por la celda solar hacia la Raspberry Pi este lo receipta de manera digital por medio del puerto GPIO.

Además de poder interactuar con el PIC 16F88 microcontrolador electrónico que lo utilizamos para la conversión y transmisión de datos, la figura muestra

el funcionamiento de la captación de datos el cual se podrá observar por el encendido del led cada vez que se transmitan datos a la comunicación serial cada 5 segundos.

La fuente de alimentación es suministrada desde el RASPBERRY PI con una señal de control RESET que determina el instante de operación del circuito.

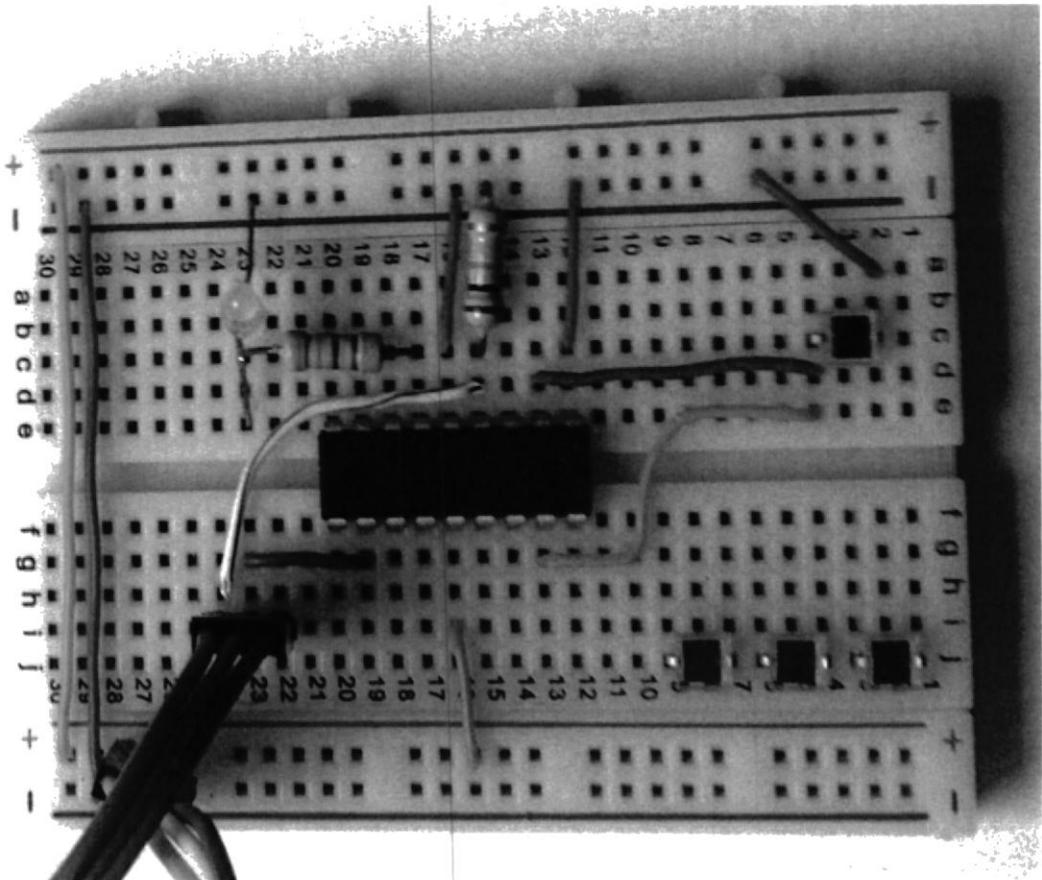


Figura 4.2 Implementación del circuito captura en protoboard



4.3 PRÁCTICA 2: CREACION BASE DE DATOS

Fue necesario realizar esta práctica para poder aprender y dinamizar habilidades en crear, editar, modificar, eliminar y ejecutar sentencias de SQL en tablas de bases de datos que el prototipo requiere para el almacenamiento de la información receptada desde la celda solar, para lo cual se utilizó el software PHPMYADMIN esta herramienta facilita la administración de MySQL, a través de un browser se logra operar con los códigos HTML en una página web utilizando el Internet.

Procedimos a instalar los servicios del phpmyadmin en la plataforma Raspbian para establecer en entorno necesario:

```
apt-get install php5-mysql; apt-get install phpmyadmin
```

Debemos de abrir un navegador y poner la IP de la RASPBERRY PI para de esta forma ingresar a la plataforma administrativa.

1.- Ingresamos la siguiente dirección: //192.168.0.60/phpmyadmin

phpMyAdmin
Bienvenido a phpMyAdmin

Idioma - Language
Español - Spanish

Iniciar sesión

Usuario:
Contraseña:

Continuar

Las cookies deben estar activadas

Figura 4.3 Inicio de sesión phpMyAdmin

2.- Ingresamos el usuario: root y la contraseña: solar123 luego se puede seleccionar el idioma con el cual se desea trabajar.

Iniciar sesión

Usuario: root

Contraseña:

Continuar

Figura 4.4 Inicio de sesión en phpMyAdmin, usuario y contraseña

3.- Con este asistente crearemos una base de datos, dándole click en **Base de datos**.

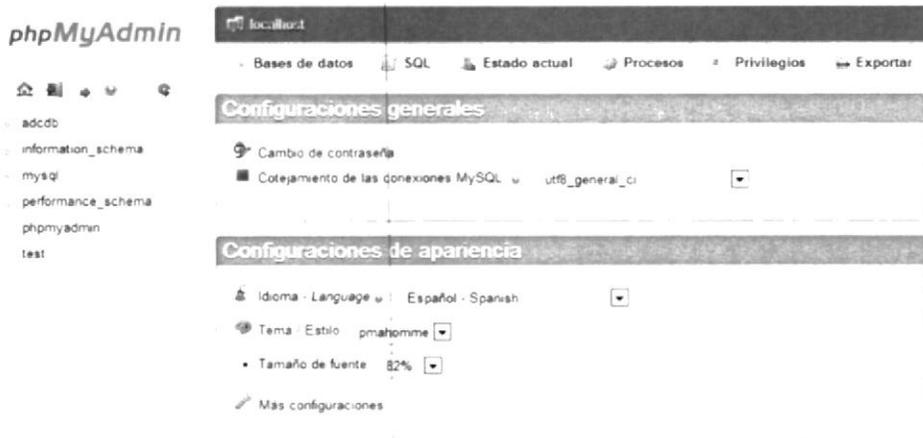


Figura 4.5. Creación base de datos en phpMyAdmin

4.- Se coloca el nombre, en este caso **práctica 2** luego pulsar crear.

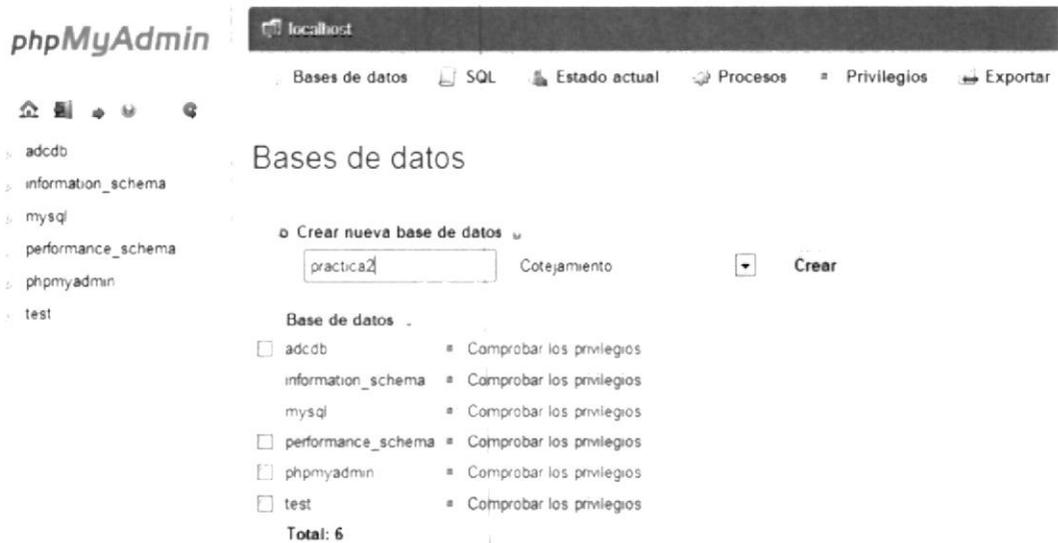


Figura 4.6 Asignación nombre base de datos phpMyAdmin

5.- Creamos la tabla **muestras** que la definiremos con **3 columnas**

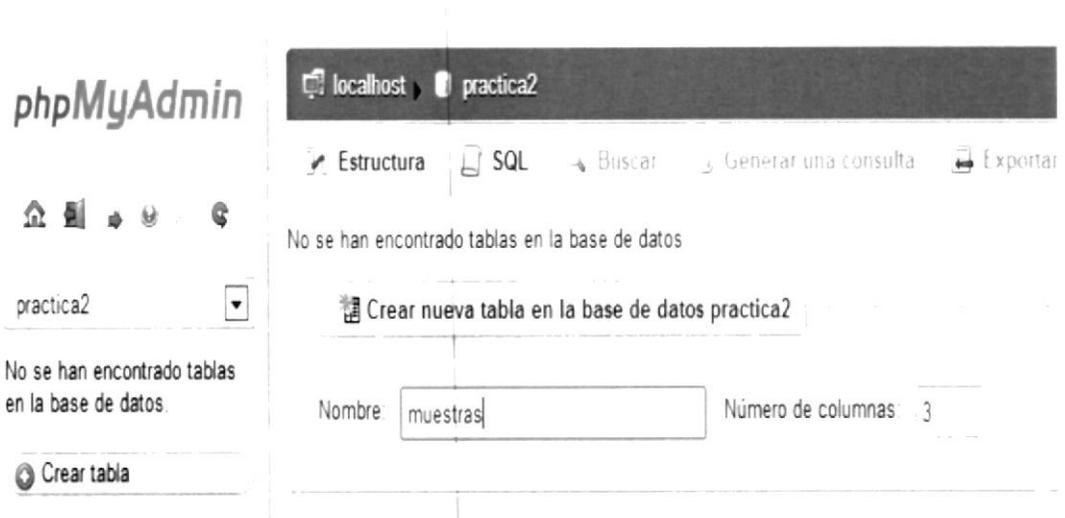


Figura 4.7 Inicialización de creación de tablas en phpMyAdmin

Como vemos tenemos la estructura de las 3 columnas creadas.

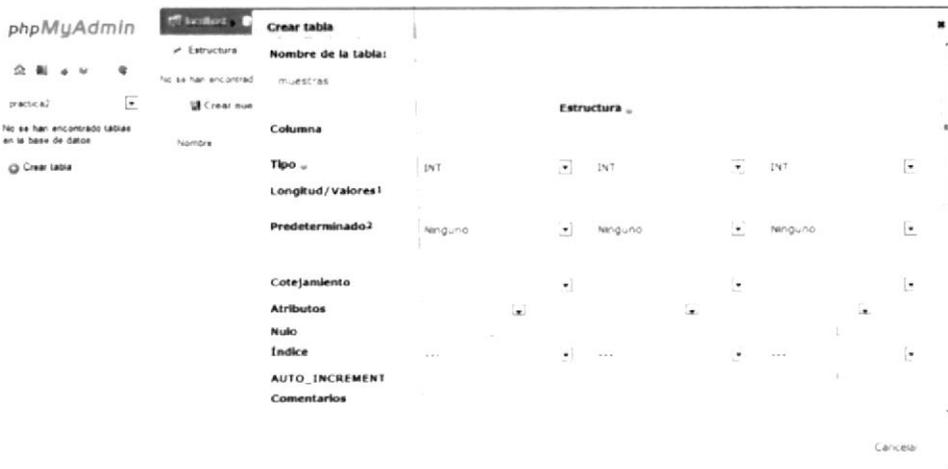


Figura 4.8 Definición de 3 columnas en phpMyAdmin

El primer campo es un **ID tipo entero**, el segundo campo es **FECHA tipo timestamp**, el tercer campo es **VALOR tipo float**.

Figura 4.9 Definición tipo de variable de las columnas phpMyAdmin

Damos click en **Guardar** y ya está creada la Tabla.

Figura 4.10 Finalización de creación de Tablas phpMyAdmin



Ahora vamos a insertar datos a la tabla en la base de datos practica2.

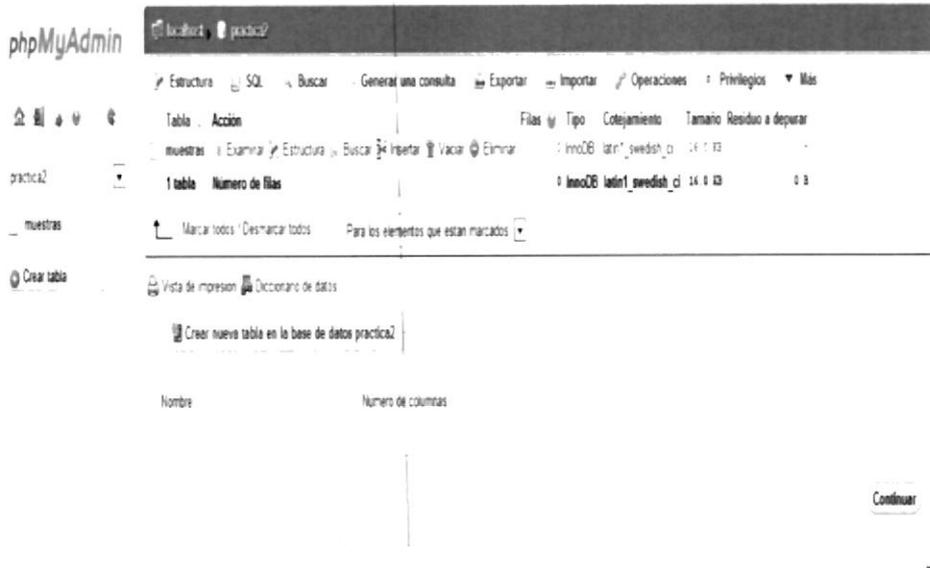


Figura 4.11 Inserción de datos en las tablas phpMyAdmin

Detallamos como ingresamos los valores desde el número 1 al 30.

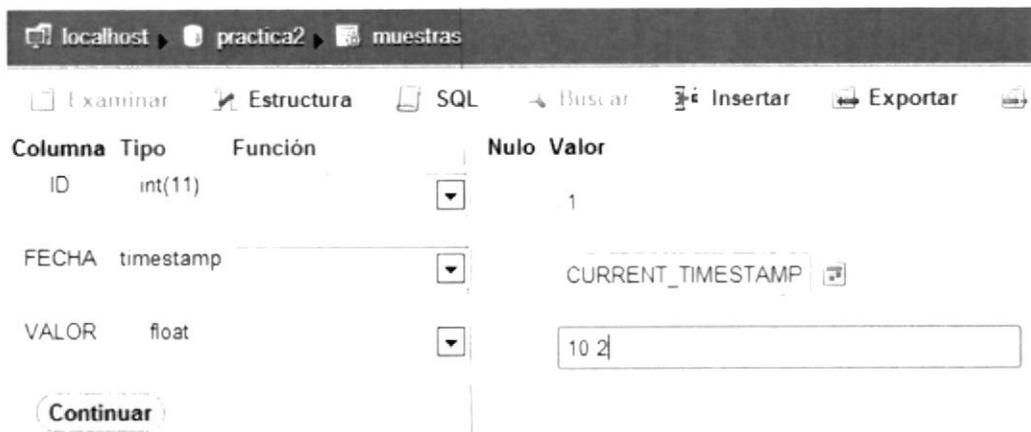


Figura 4.12 Ingreso del primer dato en la base de datos

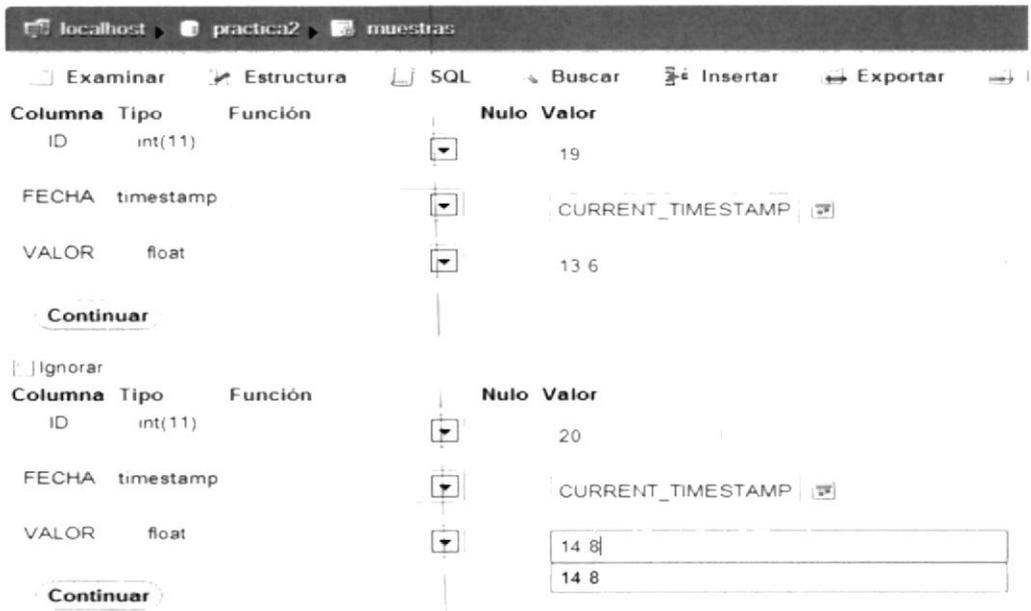


Figura 4.13 Ingreso de datos 19 – 20 en la base de datos

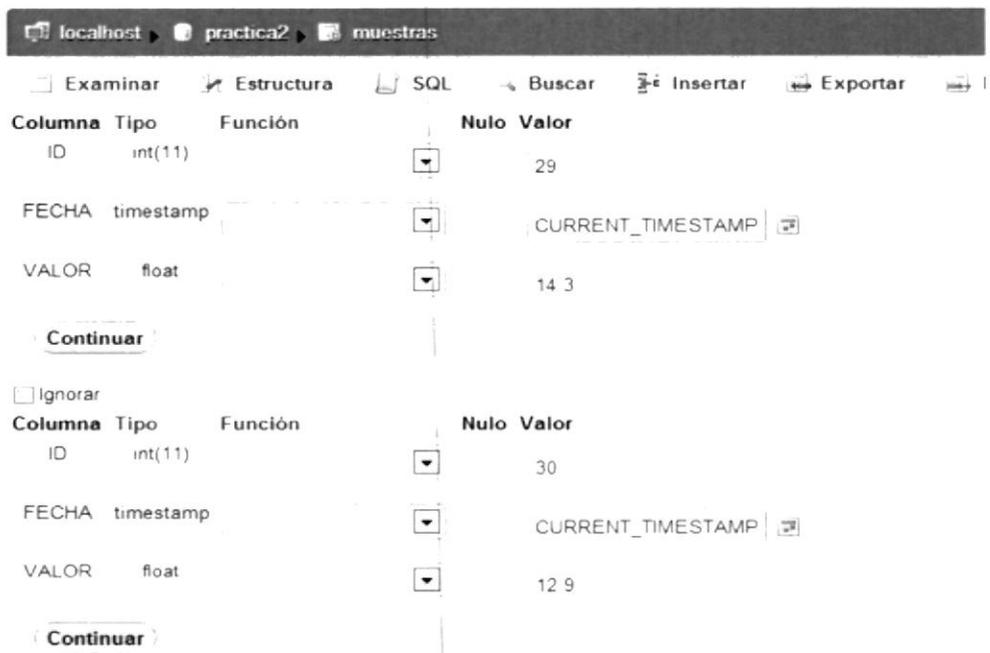


Figura 4.14 Ingreso del último dato en la base de datos

4.4 PRÁCTICA 3: CREACION DE UNA GRAFICA

Instalamos la librería **rrdtool** en el Raspbian que nos permite el manejo y la creación de gráficas, **rrdtool** necesita los datos que se halla en la base de datos en este caso utilizaremos los datos guardados.

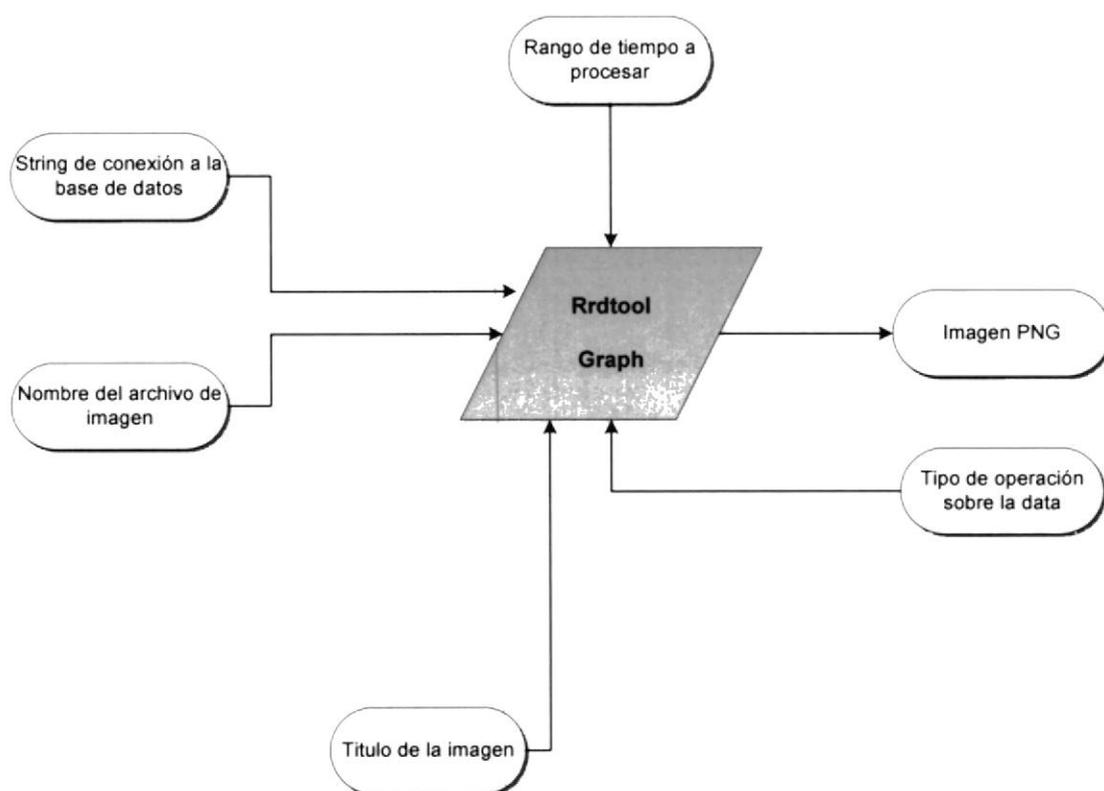


Figura 4.15 Comando RRDTOOL

El String de conexión de base de datos tiene un formato especial que se construye de la siguiente forma:



```
"sql//mysql/host-127.0.0.1/dbname-  
practica2/username=root/password=solar123/muestras/*fecha/valor".
```

Como se puede observar se usara el mismo usuario y contraseña para autenticarnos a la base de datos, luego hay que ejecutar el comando:

```
DS_BASE="sql//mysql/host=127.0.0.1/dbname=practica2/username=root/pas  
sword=solar123//muestras/*fecha/valor/"
```

Esto me guarda el String de conexión de la base de datos.

```
rrdtool graph /var/www/adc-solar/web/p3.png --start -2d --end -1d DEF:  
g1=$DS_BASE:avg:AVERAGE: step=100 AREA: g1#009900:"Practica3"
```

Esta sentencia indicara desde donde inicia hasta donde culmina la fecha de los datos que tomaremos, además del título de la imagen que en este caso es "Practica3".

Finalmente para poder observar la imagen creada abrimos un navegador se ingresa la IP de la RASPBERRY PI en este caso:

http://ip_raspberry/p3.png damos enter para ingresar a la página y se mostrara la gráfica.

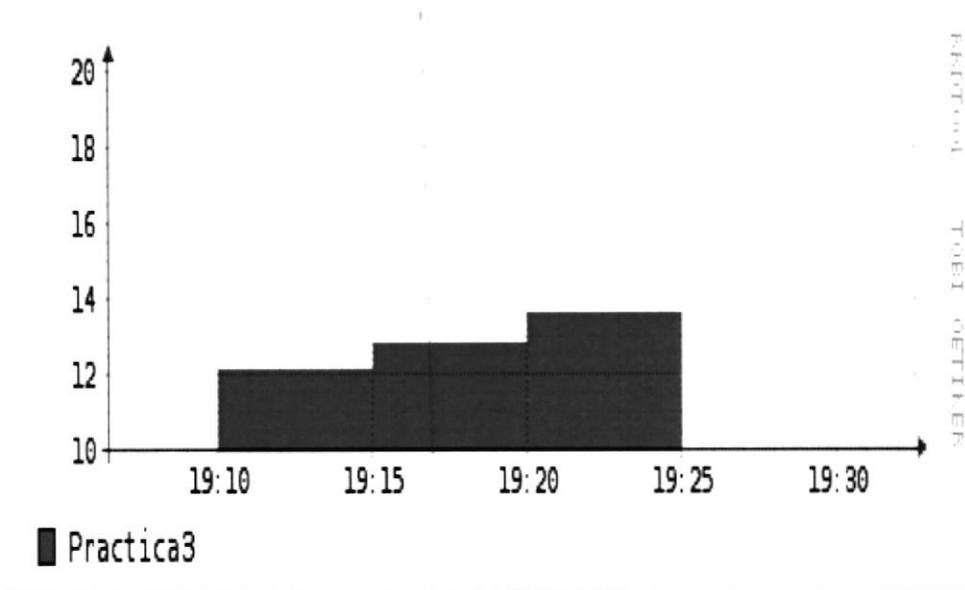


Figura 4.16 Grafica del ejercicio 3

Como se menciona esta imagen se la obtuvo con el comando: `rrdtool graph /var/www/adc-solar/web/p3.png --start -3166min --end -3140minDEF:g1=$DS_BASE:avg:AVERAGE:step=100AREA:g1#009900:"Practica3"`.

4.5 SOLUCION DEL PROYECTO

CONVERSION ANALOGICA A DIGITAL

Se necesitó utilizar un Pic 16F88 para realizar una conversión analógica a digital debido a que los puertos seriales del Raspberry Pi no reciben señales analógicas las cuales son transmitidas por la celda solar BPW34, a continuación se mostrará el código utilizado para este propósito.

```
#include "adc.h"

void main()
{
    int16 adc_value;

    float volts, milivoltios;

    setup_adc_ports(AN0);

    setup_adc(ADC_CLOCK_DIV_8);

    set_adc_channel(0);

    output_low(PIN_B1);

    delay_us(20);

    while(1)
    {
        setup_adc(ADC_CLOCK_DIV_8);

        read_adc(ADC_START_ONLY);
```

```
adc_value = read_adc(ADC_READ_ONLY);  
volts = (float)(adc_value * 5)/1023.0;  
milivoltios = (volts*1000); // lo llevo a milivoltios  
  
//Formato de envio del mensaje  
  
//I: indica inicio de mensaje  
  
//F: indica fin de mensaje  
  
//ex: I255.34F o I0.00F  
  
//El mensaje tiene entre 4 a 6 caracteres  
  
printf("I%4.2fF", milivoltios);  
  
setup_adc(ADC_OFF);  
  
delay_ms(4000);  
  
output_high(PIN_B1);  
  
delay_ms(500);  
  
output_low(PIN_B1);  
  
delay_ms(500);  
  
}  
  
}
```

PROGRAMA PRINCIPAL

Luego de que los datos estén alojados en los puertos seriales se necesita que esta información sea verificada y si es correcta deberá guardarse en la base de datos que se creara.

/* Simple C program that connects to MySQL Database server*/

```
#include <mysql.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <wiringPi.h>
```

```
#include <wiringSerial.h>
```

```
#include <string.h>
```

```
#include <errno.h>
```

```
MYSQL * getMySQLConnect();
```

```
MYSQL_RES *execQuery(MYSQL *conn, char *sql, int len);
```

```
int getSerialConnect();
```

```
char *getSerialData(int fd);
```

```
void enableADC();
```

```
void disableADC();
```

```
int main()
```

```
{
```

```
    int    fd;
```

```
    MYSQL  *conn;
```

```
    MYSQL_RES *res;
```

```
    char   *data;
```

```
    char   *stat;
```

```
    int    len;
```

```
char    query[100];

fd = getSerialConnect();

if(fd < 0) exit(1);

conn = getMySQLConnect();

if(!conn) exit(1);

enableADC();

sleep(1);

while(1){

    data = getSerialData(fd);

    disableADC();

    stat = "INSERT INTO muestra(valor) VALUES(%s);";

    len = snprintf(query,strlen(stat)+strlen(data)-1,stat,data);

    printf("Muestra: %s mV\n",data);

    res = execQuery(conn,query,len);

    stat = "UPDATE muestra_actual set fecha=CURRENT_TIMESTAMP,
valor=%s;";

    len = snprintf(query,strlen(stat)+strlen(data)-1,stat,data);

    //printf("Muestra: %s mV - %s",data,query);

    res = execQuery(conn,query,len);

    mysql_free_result(res);

    sleep(25);
```

```

    enableADC();
}

mysql_close(conn);
}

/** FUNCIONES PARA MYSQL **/

MYSQL_RES *execQuery(MYSQL *conn, char *sql, int len)
{
    MYSQL_RES *res;

    /* send SQL query */
    if (mysql_real_query(conn,sql,len)) {
        fprintf(stderr, "%s\n", mysql_error(conn));
        return NULL;
    }

    res = mysql_use_result(conn);

    return res;
}

MYSQL *getMySQLConnect()
{
    MYSQL *conn = NULL;

    char *server = "localhost";

    char *user = "root";

```



```

char *password = "solar123";

char *database = "adcdb";

/* Connect to database */

conn = mysql_init(NULL);

if (!mysql_real_connect(conn,server,user,password,database,0,NULL,0)){

    fprintf(stderr, "%s\n", mysql_error(conn));

    return NULL;

}

return conn;

}

/** FUNCIONES PARA RS232/SERIAL **/

int getSerialConnect()

{

    int fd    = -1;

    char *device = "/dev/ttyAMA0";

    int baud   = 9600;

    if((fd = serialOpen (device,baud)) < 0){

        fprintf(stderr, "Unable to open serial device: %s\n", strerror (errno));

        return -1;

    }

    return fd;

}

```

```
char *getSerialData(int fd)
{
    int c;
    char data[6];
    char *dataok = "0.00";
    int begin = 0;
    int conta = 0;
    int i = 0;
    while(1){
        c = serialGetchar(fd);
        if(c == 'I'){
            begin = 1;
            conta = 0;
        }
        else if(c == 'F'){
            begin = 0;
            fflush(stdout);
            break;
        }
        else if(begin == 1){
            data[conta] = c;
            conta++;
        }
    }
}
```

```
    }  
  
    fflush(stdout);  
  
}  
  
if(conta > 0){  
  
    dataok = NULL;  
  
    dataok = (char *)calloc(conta,sizeof(char));  
  
    for(i=0; i<conta; i++)  
  
        dataok[i]=data[i];  
  
}  
  
return dataok;  
  
}  
  
/** FUNCION PARA EL RESET **/  
  
void enableADC()  
  
{  
  
    if(wiringPiSetup() == -1){  
  
        fprintf(stderr, "%s\n", "wiringPiSetup error");  
  
        exit(1);  
  
    }  
  
    pinMode(4,OUTPUT) ;  
  
    digitalWrite(4,HIGH) ;  
  
}  
  
void disableADC()
```



```
{  
  if(wiringPiSetup() == -1){  
    fprintf(stderr, "%s\n", "wiringPiSetup error");  
    exit(1);  
  }  
  pinMode(4,OUTPUT);  
  digitalWrite(4,LOW);  
}
```

CREACION BASE DE DATOS DEL PROYECTO

El almacenamiento en la base de datos nos ayudara a que la información receptada por los puertos seriales de la RaspBerry Pi puedan ser organizados, administrados y visualizados.

1. Instalación MySQL y PHPmyAdmin

```
Sudo apt-get install mysql_server
```

```
Sudo apt_PHPmyadmind
```

2. Visualización Base de datos del proyecto

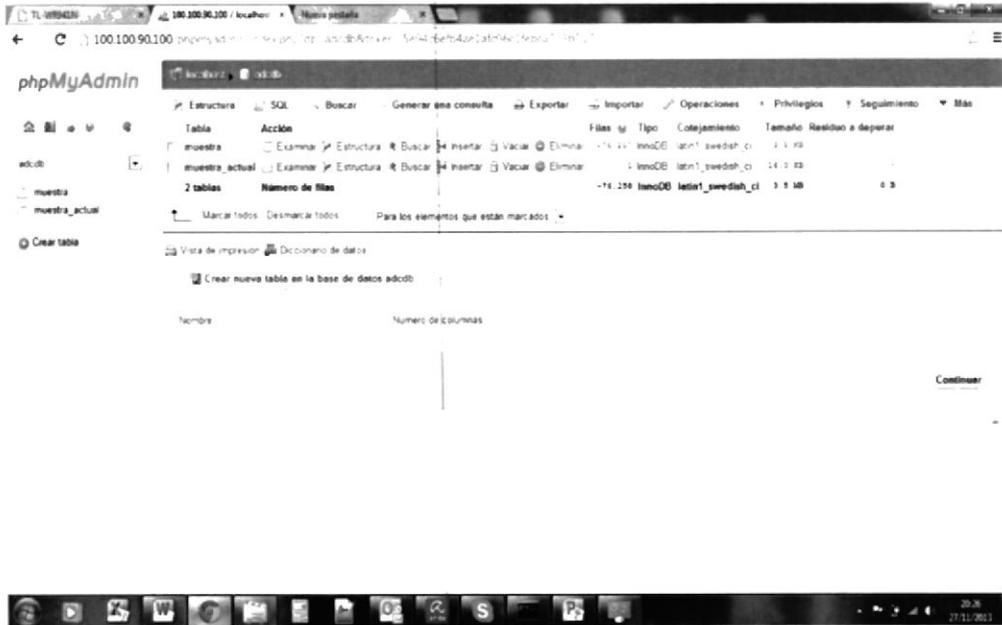


Figura 4.17 Estructura de la base de datos adodb

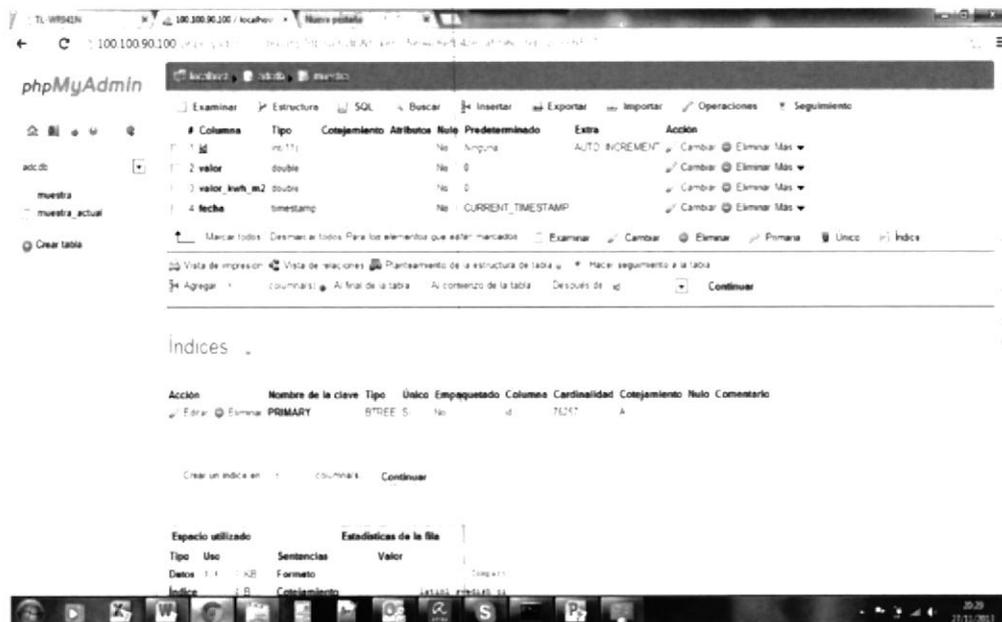


Figura 4.18 Estructura de la tabla muestra

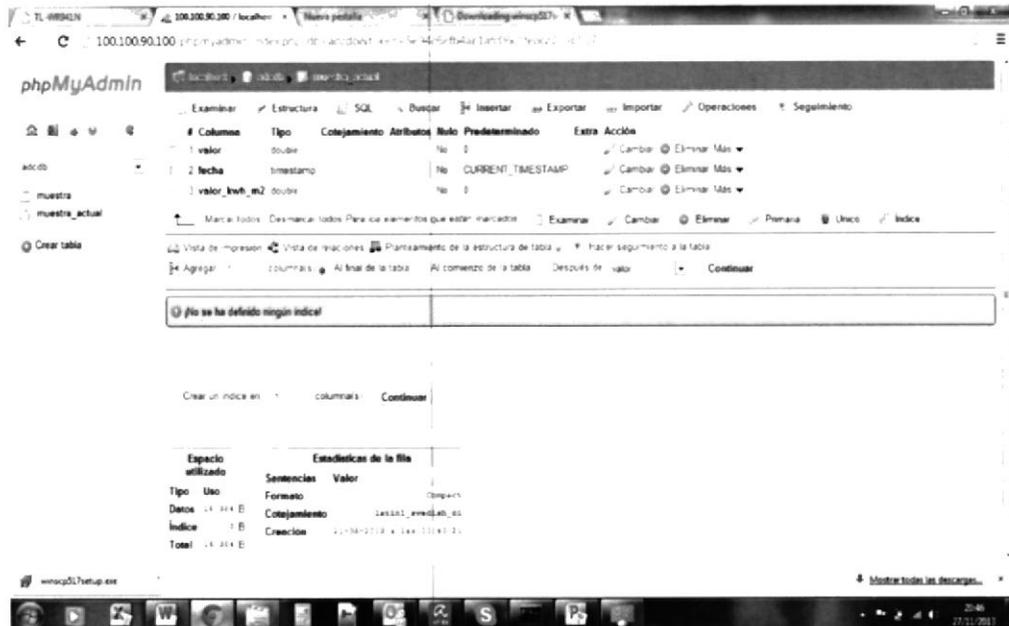


Figura 4.19 Estructura de la tabla muestra actual

SCRIPTS DE CREACION DE LAS IMÁGENES

Son de importancia su elaboración para poder observar las imágenes que nos proporciona rrdtool conforme avanza el tiempo, ya que estos son programas incrustados en el crond, procesos ocultos que siempre se ejecutan cada cierto tiempo en este caso se realizó dos scripts el primero de 1 min y el segundo de 9min.

SCRIPTS 1:

```
#!/bin/bash
```

```
DS_BASE="sql//mysql/host=127.0.0.1/dbname=adcdb/username=root/pa  
ssword=solar123//muestra/*fecha/valor/"  
DIR="/var/www/adc-solar/web/"  
TITLE="Milivoltios"  
COLOR="#009900"
```

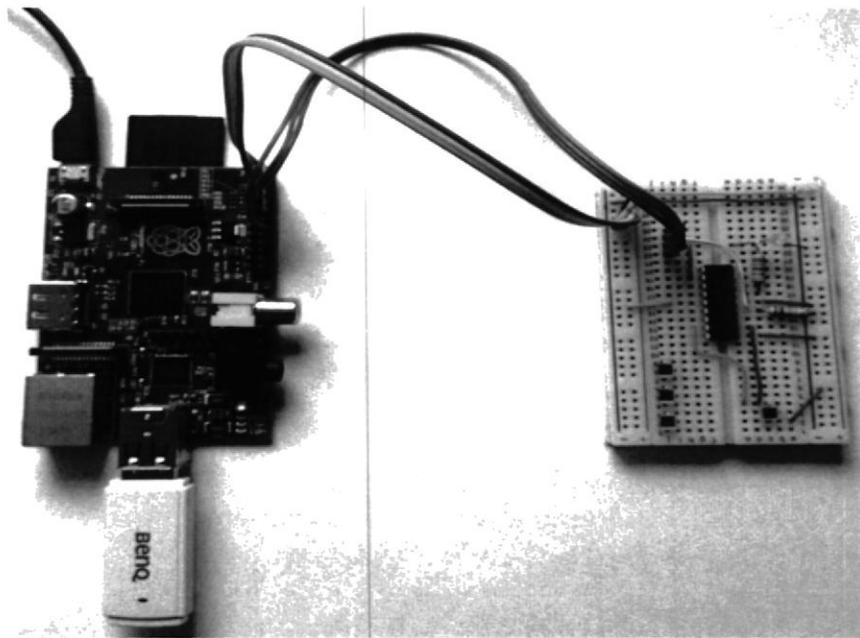


Figura 4.20 *Ensamblado general del prototipo*

4.6 GRAFICAS HTML DEL MINIMO RADIACION SOLAR

Fecha de muestra
2013-05-17 20:29:33

Muestra actual
0 Watt/m²

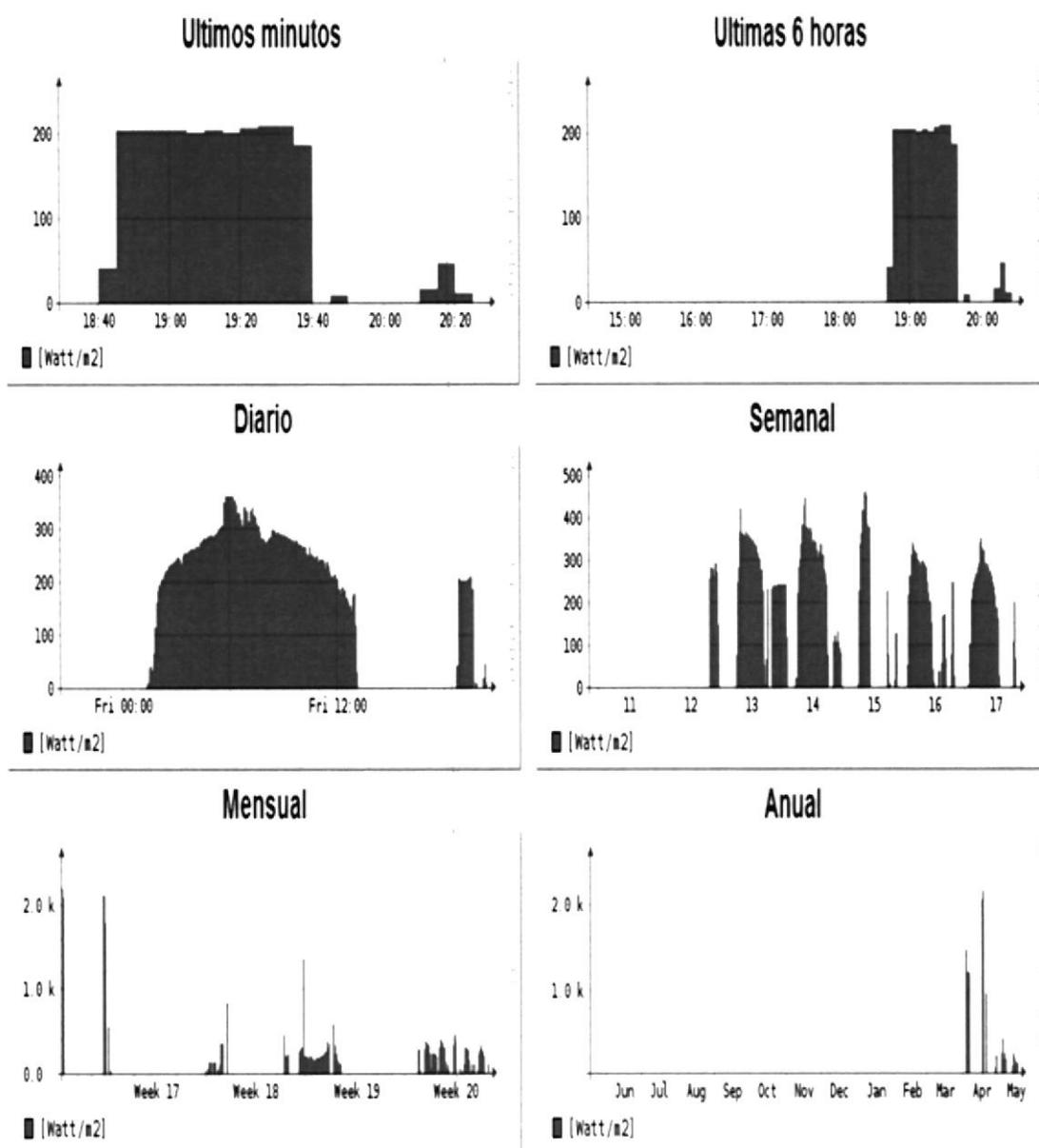


Figura 4.21 Grafica de medición mínima (lectura nocturna)

4.7 GRAFICAS HTML DEL MAXIMO RADIACION SOLAR

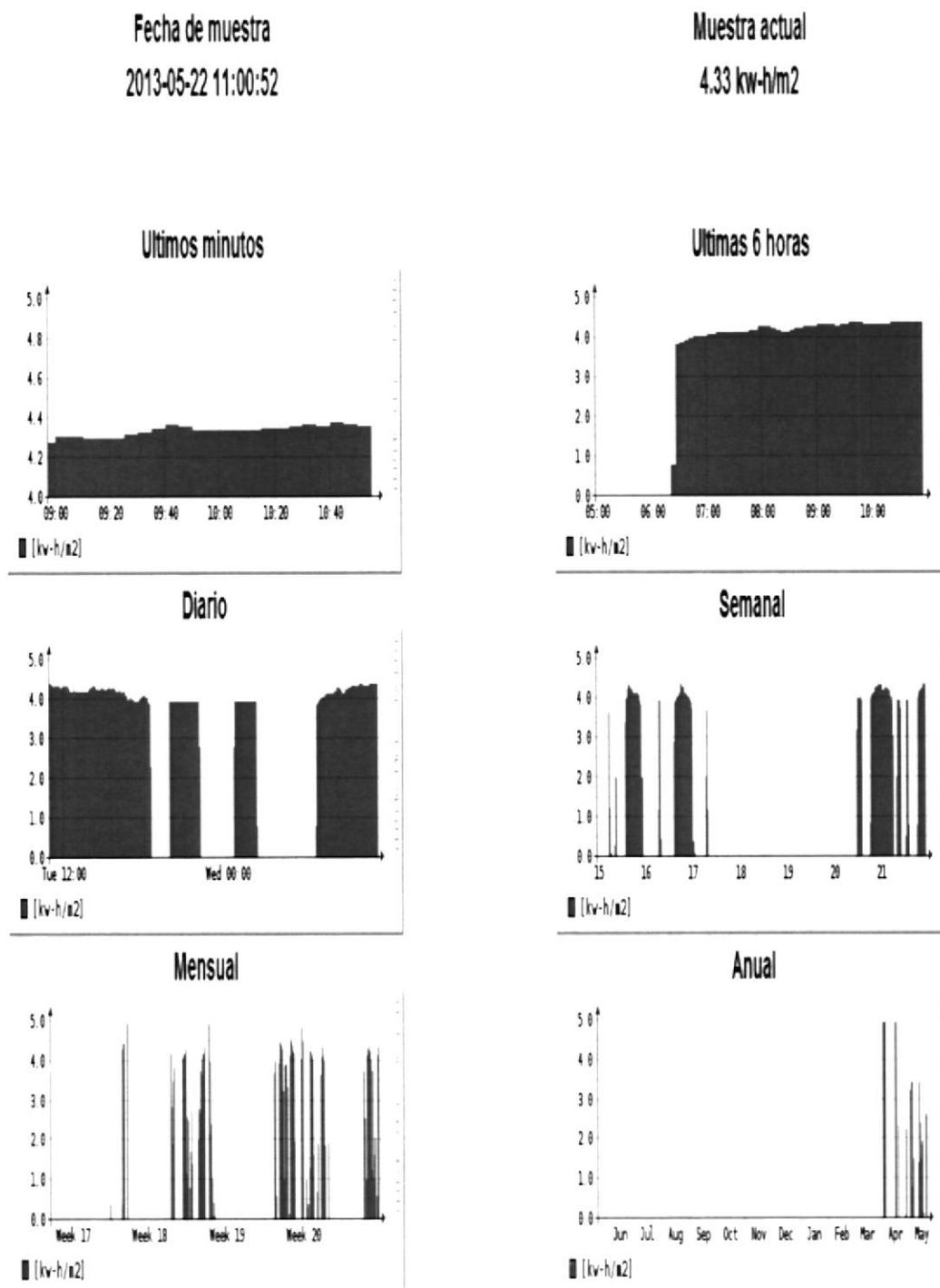


Figura 4.22 Grafica de medición máxima (lectura día)

4.8 EVALUACION DE DATOS DE RADIACION SOLAR

Hemos estimado los valores de radiación solar adquiridos en la ciudad de Guayaquil-Ecuador por el prototipo realizado para medir los niveles de radiación solar utilizando los datos recogidos durante un período de 22 años de los niveles de radiación solar en Guayaquil obtenidos de solar Electricity Hand Book el cual proporciona datos de insolación promedio mensual el cual genera información diaria promedio de cuanto un sistema fotovoltaico producirá en cualquier mes dado, de esta forma se aproximó el valor de los niveles de radiación solar dado que al tener 4,9 kW-h/m² que es el valor máximo en Guayaquil de forma empírica vamos a asociarlo con 470mv que fue el valor máximo obtenido de la celda solar en un día soleado y tener 3.7 kW-h/m² que es el valor mínimo en Guayaquil de forma empírica vamos a asociarlo con 200mv que fue el valor obtenido de la celda solar apenas hubo luz de sol, esto fue como de 8 a 9 am. Podemos decir que existe 1.2 kW-h/m² de rango de 3.7 a 4.9 kW-h/m². Además de que existe 270mv de rango desde 200mv hasta 470mv. Para obtener una relación en estas dos escalas se ha realizado una división de $1.2/270$ lo que nos da un valor de 0.00444 esto indica que por cada 1mv tenemos una variación de 0.00444 kW-h/m² se debe tener en cuenta que esto es válido a partir de los 200mv valores menores a 200mv con lo cual se interpreta que no hay luz solar y se lo representa con 0 kW-h/m². También cualquier valor mayor a 470mv será interpretado como 4.9 kW-h/m². Luego de probar los valores arriba descritos

se ha llegado a manejar los siguientes rangos de valores para tener una mejor interpretación de las gráficas, por lo cual los valores son:

5.5 kW-h/m² relacionado con 500mv, 3.0 kW-h/m² relacionado con 180mv esto nos da la relación entre 320mv a 2.5 KW-h/m², con esto podemos relacionar 2.5/320 dando un valor de 0.00781 esto indica que por cada 1mv tenemos 0.00781 kW-h/m². A consecuencia los valores menores a 180mv se interpreta que no hay luz solar y se lo representa con 0 kW-h/m². También cualquier valor mayor a 500mv será interpretado como 5.5 kW-h/m². [18]

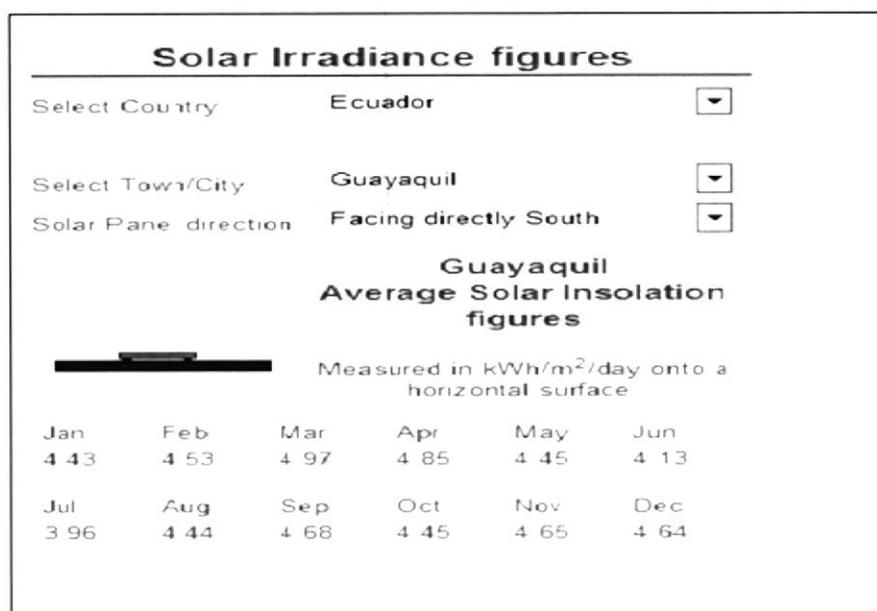


Figura 4.23 Datos de niveles de radiación [18]

CONCLUSIONES

1. En el desarrollo de este proyecto podemos concluir que uso de las diversas tecnologías de punta en Hardware y Software se pudo lograr la realización del prototipo, se usó una celda solar, un pic microcontrolador y el minicomputador Raspberry Pi componentes creados para diversas aplicaciones, en nuestro caso nos proporcionó la capacidad de crear interfaces para trabajar juntos en este proyecto.
2. Se comprobó que la radiación solar en Ecuador Guayaquil Latitud ≈ 0 está en un mínimo 3.7 KW-h/m^2 y un máximo de 4.9 KW-h/m^2 comparando con las tablas de radiación solar ya establecidas en solarelectricityhandbook es evidente que nuestra medición tendría un margen error debido a que el tiempo de medición en nuestro caso ha sido corto y en la celda BPW34 su superficie adquisición es de 3 mm^2 .



3. Se concluyó que el uso del minicomputador RASPBERRY PI ha reducido el uso de componentes electrónicos para la implementación del prototipo, el mismo que permite almacenar, administrar y graficar los datos adquiridos por la celda solar, la representación gráfica KW/m² vs. hora nos puede demostrar los instantes beneficiosos y/o perjudiciales del uso de los rayos solares.

RECOMENDACIONES

1. Se debe considerar en aprender e investigar para tener el dominio del funcionamiento de cada uno de los componentes que serán utilizados en este proyecto antes de combinar sus funciones ya que son independientes y cada cual posee su propia lectura y lenguaje.
2. Es importante documentar las codificaciones y los procesos de implementación, ya que no dejaran de existir las “pruebas y errores” que son necesarios para el perfeccionamiento del proyecto que es el objetivo esperado permitiéndonos seguir las secuencias de los procesos obteniendo una mejor comprensión del mismo.
3. Para utilizar el puerto serial e interactuar con otros dispositivos (ejemplo: Arduino), la consola serial sesión puerto debe de estar

deshabilitado, además se debe observar las sintaxis e instrucciones instalando las librerías necesarias para el uso de los servicios y plataforma del software deberán estar incluidas antes de compilar el código. Precisar los parámetros de sincronización en la transmisión y recepción de los datos para la captura y almacenamiento de la data. Sugerimos realizar la conexión SSH para acceder a la RASPBERRY PI.

BIBLIOGRAFIA

- [1] Clerk, James, Magnetic Energy, Courier daver publications, 1891.
- [2] Parker, Sybil, Ciencia y Tecnología, McGraw-Hill ,1998.
- [3] ALBUJA, W., “Diseño y construcción de un destilador solar en cascada”, tesis, facultad de ingeniería mecánica, escuela superior politécnica del litoral, 1989.
- [4] Bernard, J., Ciencias ambientales, Prentice-Hall 6ta edición, Consultado el 12 de Marzo del 2013.
- [5] Nishati Foundation, Radiación solar directa, difusa y reflejada, <http://nishati.org/hardware>, Consultado el 26 de Marzo del 2013.
- [6] Lindsell, D., Índice Radiación Solar, www.solar-facts.com/world-solar/world-insolation.php, Consultado el 25 de marzo del 2013.
- [7] Antonio CH, Consecuencias De La Radiación Solar. <http://www.inppares.org/revistasss/Revista%20II%202009/14%20-%20Radiacion.pdf> , Consultado el 14 de marzo del 2013.

- [8] Garcia,E.,Copilador c-ccs para microcontroladores pic,Alfaomega ,2008

- [9] Pons,N., Linux Basico,Eni,2010

- [10] Hertzang, R., the Debian Administrator Handbook, Freexian, 2003.

- [11] Tojeiro, German, Proteus, MARCOMBO, 2008.

- [12] Bowen, Rich, Apache, AmayaMultimedia, 2008.
- [13] Cobo, Angel, php y MySQL, Díaz De Santos, 2005.

- [14] Moliner, M., PIC 16F88, <http://www.ele-mariamoliner.dyndns.org/~fperal/lodi/9B-pic16f88.pdf>, Consultado el 19 de marzo del 2013.

- [15] Sparkfun, Celda Solar Miniatura BPW34, <https://www.sparkfun.com/products/9541>, Consultado el 16 de marzo del 2013.

- [16] Raspberry Pi Foundation, Placa Raspberry Pi, <http://www.raspberrypi.org/faqs>, Consultado el 19 de Marzo del 2013.

- [17] Microchip Technology Inc., PICKIT 3, <http://www.electronicaestudio.com/microchip.htm>, Consultado el 17 de marzo del 2013.

- [18] Geenstream Publishing, Datos radiación solar, <http://solarelectricityhandbook.com/solar-irradiance.html>, Consultado el 20 de mayo del 2013.



CIB - ESPOL

ANEXOS

ANEXO A

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



Figura 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.



Raspberry Pi®

[Quick Start](#)

[Downloads](#)

[Buy Codecs](#)

to do, the community on the Raspberry Pi Wiki has written a [guide for beginners](#) on how to set up your SD card

Thanks to Webfusion [Managed Hosting](#) for providing hosting and bandwidth for this service

Raspbian "wheezy"

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.

Torrent	2012-12-16-wheezy-raspbian.zip.torrent
Direct download	2012-12-16-wheezy-raspbian.zip
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.

Figura 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.

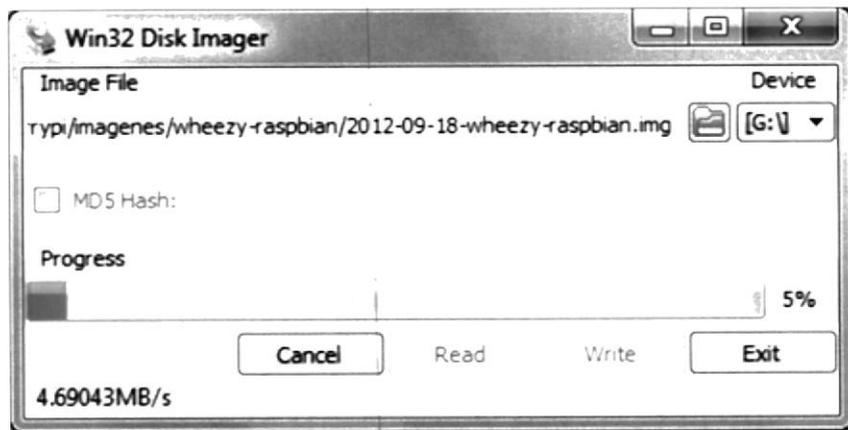
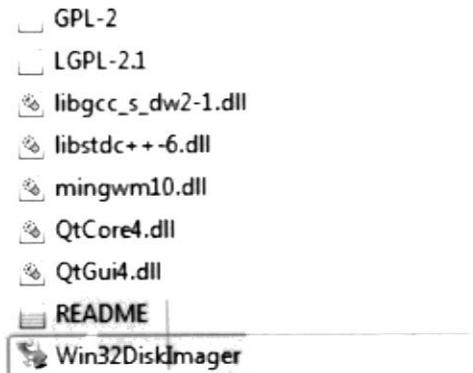


Figura A3 Descargando el sistema operativo en la memoria usando DiskImager



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 - El sistema en la plataforma Linux utilice todo el espacio que permita la memoria SD.

configure_keyboard - Para configurar el teclado (en este caso, uno "Generico de 105 teclas", aparece como "Generic 105 key") y su distribución (se propone "English (UK)", pero deberíamos escoger "Other" y luego "Spanish").

Finalmente nos preguntará 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 estamos acostumbrados, por si el entorno gráfico llegase a bloquearse.

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.

- Se pueden configurar varias opciones, en este caso vamos a expandir la memoria, el fin es expandir la partición root a la capacidad de la memoria.

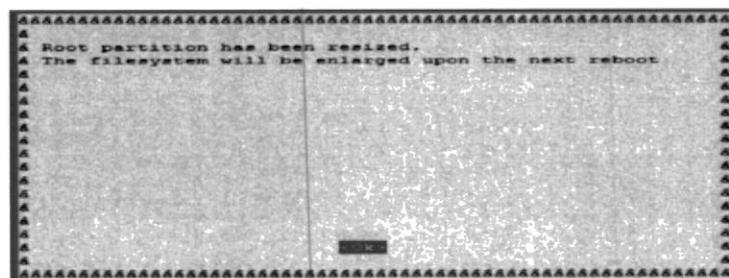


Figura B3 Expandir memoria para mayor capacidad en root

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

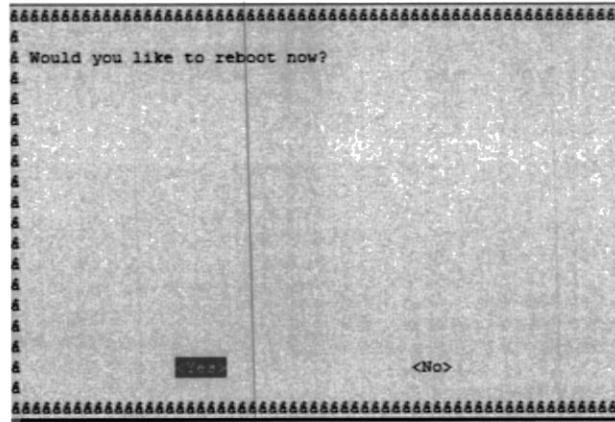


Figura B8 Reiniciar S.O luego de configurar.

Pantalla inicial de instalación del sistema operativo.

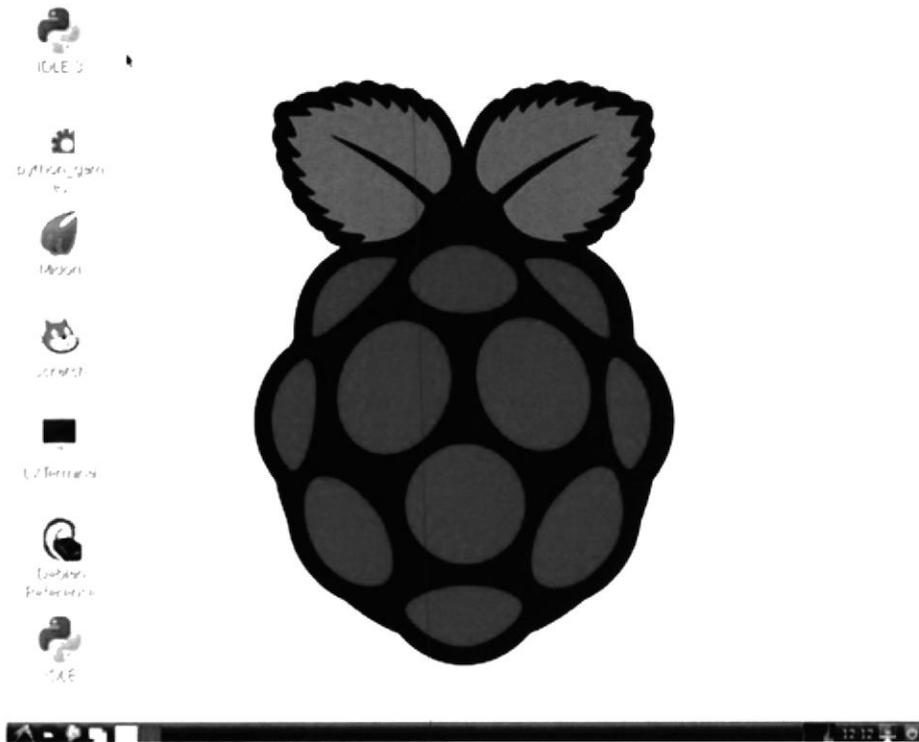


Figura B9 Pantalla principal de S.O en Raspberry pi



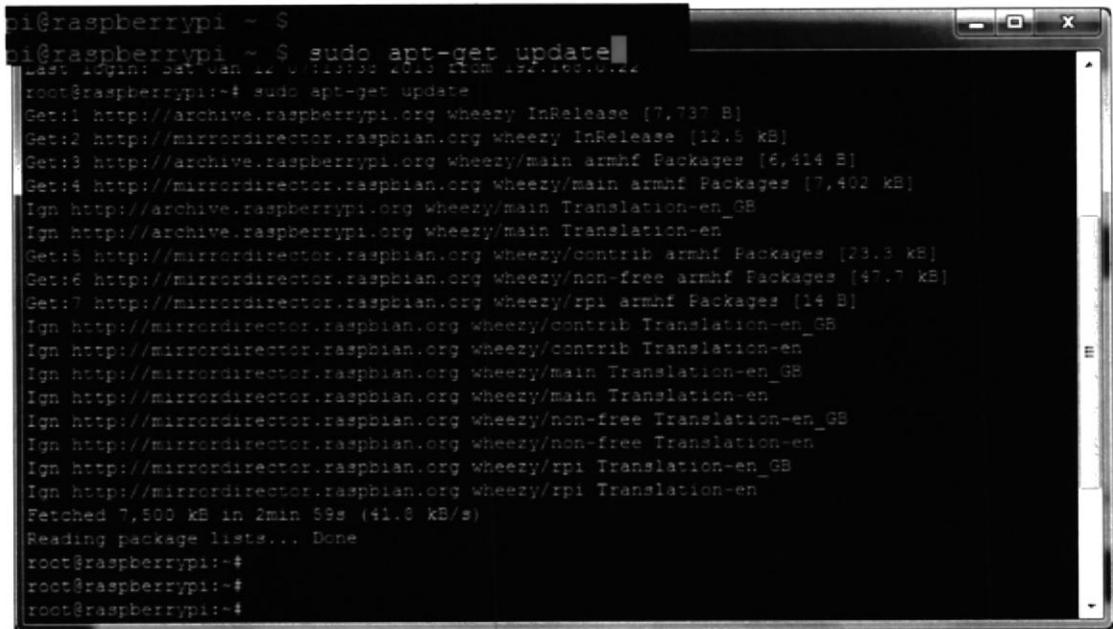
ANEXO C

Instalando un programa

Se recomienda actualizar La base local de repositorio de Raspberry PI antes de instalar cualquier programa, en este caso instalaremos un programa que nos ayudara a tener acceso remoto al dispositivo.

Actualizar la base local de repos en el raspberry pi, esto lo hacemos con el comando: pi@raspberrypi ~ \$ sudo apt-get update

Nota: Se recomienda realizar esto para actualizar y evitar errores



```
pi@raspberrypi ~ $
pi@raspberrypi ~ $ sudo apt-get update
Last login: Sat Jan 12 0 19:00 2013 from 192.168.0.22
root@raspberrypi:~# sudo apt-get update
Get:1 http://archive.raspberrypi.org wheezy InRelease [7,737 B]
Get:2 http://mirrordirector.raspbian.org wheezy InRelease [12.6 kB]
Get:3 http://archive.raspberrypi.org wheezy/main armhf Packages [6,414 B]
Get:4 http://mirrordirector.raspbian.org wheezy/main armhf Packages [7,402 kB]
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
Get:5 http://mirrordirector.raspbian.org wheezy/contrib armhf Packages [23.3 kB]
Get:6 http://mirrordirector.raspbian.org wheezy/non-free armhf Packages [47.7 kB]
Get:7 http://mirrordirector.raspbian.org wheezy/rpi armhf Packages [14 B]
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Fetched 7,500 kB in 2min 59s (41.8 kB/s)
Reading package lists... Done
root@raspberrypi:~#
root@raspberrypi:~#
root@raspberrypi:~#
```

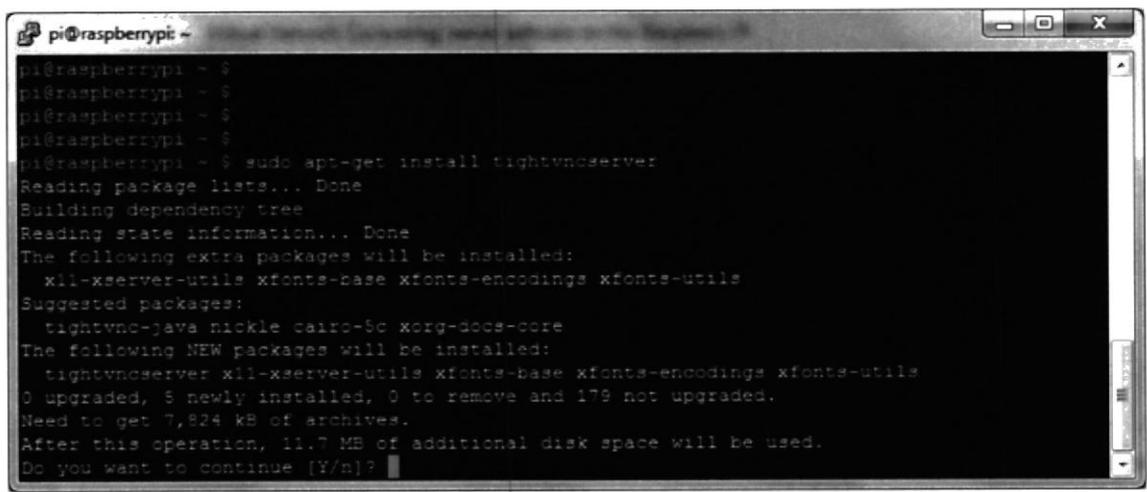
Figura C1 Comando de Actualización

Instalando VNC

Ante todo, VNC *Virtual Network Computing* (Computación Virtual en Red).

Es un programa de software libre basado en una estructura cliente-servidor el cual permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente.

Ejecutamos el comando, `sudo apt-get install tightvncserver` , con esto instalamos el paquete de servidor VNC



```
pi@raspberrypi ~ $  
pi@raspberrypi ~ $  
pi@raspberrypi ~ $  
pi@raspberrypi ~ $  
pi@raspberrypi ~ $ sudo apt-get install tightvncserver  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  x11-xserver-utils xfonts-base xfonts-encodings xfonts-utils  
Suggested packages:  
  tightvnc-java nickle cairo-1.0 xorg-docs-core  
The following NEW packages will be installed:  
  tightvncserver x11-xserver-utils xfonts-base xfonts-encodings xfonts-utils  
0 upgraded, 5 newly installed, 0 to remove and 179 not upgraded.  
Need to get 7,824 kB of archives.  
After this operation, 11.7 MB of additional disk space will be used.  
Do you want to continue [Y/n]?
```

instead.
pi@raspberrypi ~ \$ sudo apt-get install tightvncserver

Figura C2 Comando instalación de VNC

Ejecutamos el servidor VNC el cual nos pedirá la contraseña para acceder al usuario pi desde otra pc de la red. Opcionalmente pide la contraseña para solo abrir una conexión de vista.



```
pi@raspberrypi ~ $ tightvncserver
You will require a password to access your desktops.
Password:
Verify:
Would you like to enter a view-only password (y/n)? n
New 'X' desktop is raspberrypi:1
Creating default startup script /home/pi/.vnc/xstartup
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log
pi@raspberrypi ~ $
```

Figura C3 *Requerimientos para realizar conexión vnc*

Una vez realizado esto, usted puede empezar a establecer conexión con el servidor VNC. Como podrá ver en la imagen de arriba le indica que usted puede abrir una sesión VNC en el display raspberrypi:1, lo que también puede ser ip_raspberrypi:1. Ejemplo, 192.168.0.25:1

Nota: Usted debe siempre ejecutar el comando tightvncserver cada vez que reinicia el raspberrypi. Opcionalmente usted puede usar el equivalente de comando tightvncserver por vncserver.

Conectarse remotamente, como cliente desde una PC

Para lograr acceder remotamente al dispositivo se debe instalar el "Cliente" de VNC para lograr visualizar, descargar el cliente de VNC.

<http://www.realvnc.com/download/viewer/>

Conectarnos remotamente nos ayuda para visualizar la interfaz del sistema operativo Raspbian, antes de realizar esto se debe conocer la dirección IP del Raspberry PI.

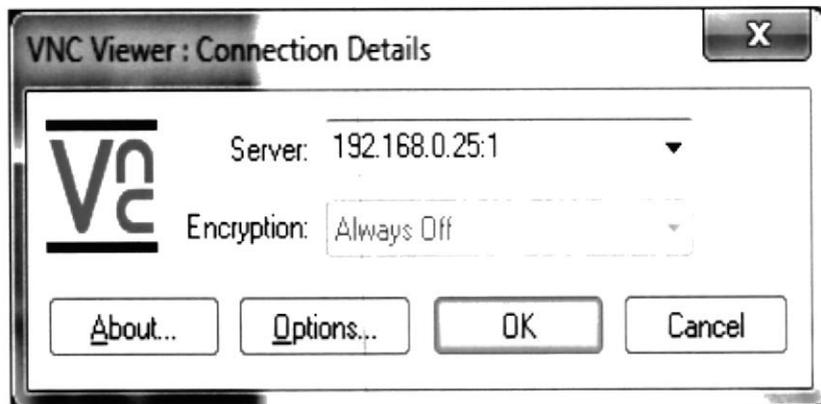


Figura C4 Interfaz VNC

De esta forma ya tenemos acceso remoto a nuestro Raspberry desde el programa VNC

ANEXO D

Instalación librería para trabajar en lenguaje C

Para la instalación de WiringPi, la cual podremos usar para programar los pines GPIO, seguiremos los siguientes pasos.

Por línea de comandos, instalamos git-core si no lo tenemos instalado

```
sudo apt-get install git-core
```

Ahora obtenemos wiringPi mediante git

```
git clone git://git.drogon.net/wiringPi
```

Ahora nos movemos a la carpeta donde se descomprimió que es wiringPi e instalamos.

```
cd wiringPi
```

```
./build
```



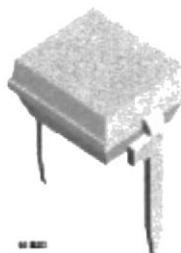
CIB - ESPOL

BPW34, BPW34S

Vishay Semiconductors



Silicon PIN Photodiode, RoHS Compliant



FEATURES

- Package type: loaded
- Package form: top view
- Dimensions (L x W x H in mm): 5.4 x 4.3 x 3.2
- Radiant sensitive area (in mm²): 7.5
- High photo sensitivity
- High radiant sensitivity
- Suitable for visible and near infrared radiation
- Fast response times
- Angle of half sensitivity: $\theta = \pm 60^\circ$
- Lead (Pb)-free component in accordance with RoHS 2002/95/EC and WEEE 2002/96/EC



RoHS
CONF. PART

DESCRIPTION

BPW34 is a PIN photodiode with high speed and high radiant sensitivity in miniature, flat, top view, clear plastic package. It is sensitive to visible and near infrared radiation.

BPW34S is packed in tubes, specifications like BPW34.

APPLICATIONS

- High speed photo detector

PRODUCT SUMMARY

COMPONENT	I_{ph} (μA)	θ (deg)	$A_{s.d.}$ (mm)
BPW34	50	± 65	430 to 1100
BPW34S	50	± 65	430 to 1100

Note

Test condition see table "Basic Characteristics"

ORDERING INFORMATION

ORDERING CODE	PACKAGING	REMARKS	PACKAGE FORM
BPW34	Bulk	MOQ: 3000 pcs, 3000 pcs/bulk	Top view
BPW34S	Tube	MOQ: 1800 pcs, 45 pcs/tube	Top view

Note

MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Reverse voltage		V_{R}	60	V
Power dissipation	$T_{amb} \leq 25^\circ\text{C}$	P_V	215	mW
Junction temperature		T_j	100	$^\circ\text{C}$
Operating temperature range		T_{amb}	-40 to +100	$^\circ\text{C}$
Storage temperature range		T_{stg}	-40 to +100	$^\circ\text{C}$
Soldering temperature	$t < 3$ s	T_{stg}	260	$^\circ\text{C}$
Thermal resistance junction/ambient	Connected with Cu wire 0.14 mm ²	R_{thJA}	350	K/W

Note

T_{amb} : 25 $^\circ\text{C}$, unless otherwise specified



BASIC CHARACTERISTICS						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Breakdown voltage	$I_0 = 100 \mu\text{A}, E = 0$	$V_{(BR)}$	60			V
Reverse dark current	$V_D = 10 \text{ V}, E = 0$	I_0		2	30	nA
Diode capacitance	$V_D = 0 \text{ V}, f = 1 \text{ MHz}, E = 0$	C_D		70		pF
	$V_D = 3 \text{ V}, f = 1 \text{ MHz}, E = 0$	C_D		25	40	pF
Open circuit voltage	$E_0 = 1 \text{ mW/cm}^2, \lambda = 950 \text{ nm}$	V_o		350		mV
Temperature coefficient of V_o	$E_0 = 1 \text{ mW/cm}^2, \lambda = 950 \text{ nm}$	TK_{V_o}		-2.6		mV/K
Short circuit current	$E_A = 1 \text{ klx}$	I_k		70		μA
	$E_0 = 1 \text{ mW/cm}^2, \lambda = 950 \text{ nm}$	I_k		47		μA
Temperature coefficient of I_k	$E_0 = 1 \text{ mW/cm}^2, \lambda = 950 \text{ nm}$	TK_{I_k}		0.1		%/K
Reverse light current	$E_A = 1 \text{ klx}, V_D = 5 \text{ V}$	I_{0r}		75		μA
	$E_0 = 1 \text{ mW/cm}^2, \lambda = 950 \text{ nm}, V_D = 5 \text{ V}$	I_{0r}	40	50		μA
Angle of half sensitivity		θ		± 65		deg
Wavelength of peak sensitivity		λ_p		900		nm
Range of spectral bandwidth		$\lambda_{0.1}$		430 to 1100		nm
Noise equivalent power	$V_D = 10 \text{ V}, \lambda = 950 \text{ nm}$	NEP		4×10^{-14}		W/Hz
Rise time	$V_D = 10 \text{ V}, R_L = 1 \text{ k}\Omega, \lambda = 820 \text{ nm}$	t_r		100		ns
Fall time	$V_D = 10 \text{ V}, R_L = 1 \text{ k}\Omega, \lambda = 820 \text{ nm}$	t_f		100		ns

Note

$T_{amb} = 25 \text{ }^\circ\text{C}$, unless otherwise specified

BASIC CHARACTERISTICS

$T_{amb} = 25 \text{ }^\circ\text{C}$, unless otherwise specified

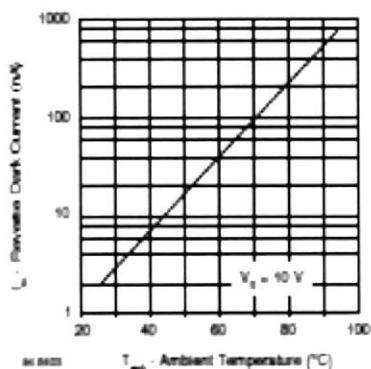


Fig. 1 - Reverse Dark Current vs. Ambient Temperature

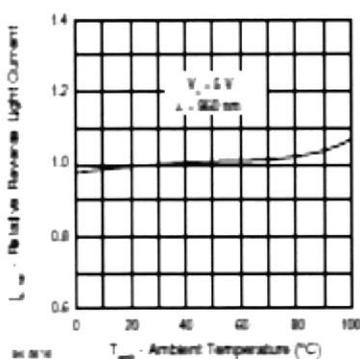


Fig. 2 - Relative Reverse Light Current vs. Ambient Temperature

BPW34, BPW34S

Vishay Semiconductors

Silicon PIN Photodiode, RoHS Compliant

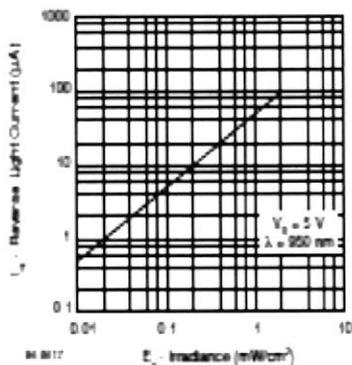


Fig. 3 - Reverse Light Current vs. Irradiance

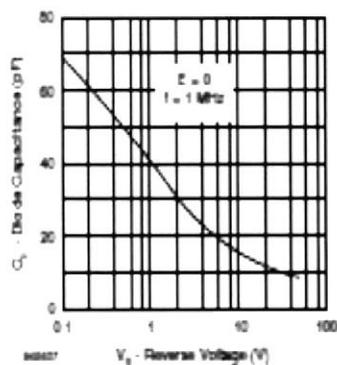


Fig. 6 - Diode Capacitance vs. Reverse Voltage

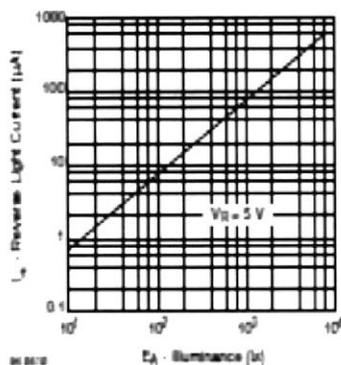


Fig. 4 - Reverse Light Current vs. Illuminance

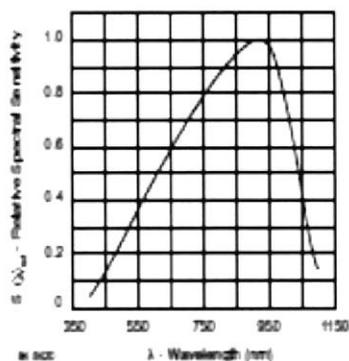


Fig. 7 - Relative Spectral Sensitivity vs. Wavelength

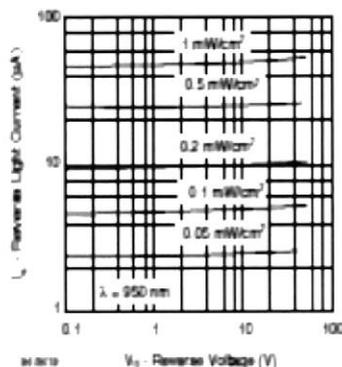


Fig. 5 - Reverse Light Current vs. Reverse Voltage

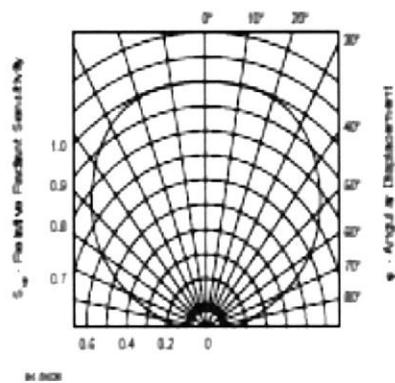


Fig. 8 - Relative Radiant Sensitivity vs. Angular Displacement



PIC16F87/88

18/20/28-Pin Enhanced Flash MCUs with nanoWatt Technology

Low-Power Features:

- Power-Managed modes:
 - Primary Run: RC oscillator, 76 μ A, 1 MHz, 2V
 - RC_RUN: 7 μ A, 31.25 kHz, 2V
 - SEC_RUN: 9 μ A, 32 kHz, 2V
 - Sleep: 0.1 μ A, 2V
- Timer1 Oscillator: 1.8 μ A, 32 kHz, 2V
- Watchdog Timer: 2.2 μ A, 2V
- Two-Speed Oscillator Start-up

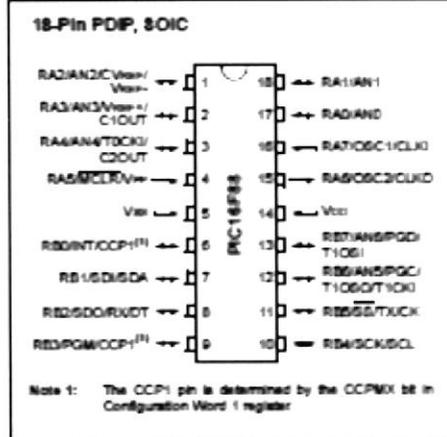
Oscillators:

- Three Crystal modes:
 - LP, XT, HS: up to 20 MHz
- Two External RC modes
- One External Clock mode:
 - ECIO: up to 20 MHz
- Internal oscillator block:
 - 8 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz

Peripheral Features:

- Capture, Compare, PWM (CCP) module:
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit, 7-channel Analog-to-Digital Converter
- Synchronous Serial Port (SSP) with SPI (Master/Slave) and I²C™ (Slave)
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART/SCI) with 9-bit address detection:
 - RS-232 operation using internal oscillator (no external crystal required)
- Dual Analog Comparator module:
 - Programmable on-chip voltage reference
 - Programmable input multiplexing from device inputs and internal voltage reference
 - Comparator outputs are externally accessible

Pin Diagram



Special Microcontroller Features:

- 100,000 erase/write cycles Enhanced Flash program memory typical
- 1,000,000 typical erase/write cycles EEPROM data memory typical
- EEPROM Data Retention: > 40 years
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Processor read/write access to program memory
- Low-Voltage Programming
- In-Circuit Debugging via two pins
- Extended Watchdog Timer (WDT):
 - Programmable period from 1 ms to 256s
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O Pins	10-bit A/D (ch)	CCP (PWM)	AUSART	Comparators	SSP	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)							
PIC16F87	7168	4096	368	256	18	N/A	1	Y	2	Y	2/1
PIC16F88	7168	4096	368	256	18	1	1	Y	2	Y	2/1