



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

**“TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES”**

TRABAJO DE GRADUACIÓN

**“MODELAMIENTO DE UN SISTEMA FOTOVOLTAICO PARA MEJORAR LA
EFICIENCIA Y OPTIMIZACIÓN DE LA ENERGÍA EN EL BLOQUE B DE
INGENIERÍA”**

AUTOR(es):

NELLY JANNETH CARCHI CHACHA

VICTOR FABIAN CASTELO MARTINEZ

DIRECTOR:

ING. ALFONSO GUNSHA

RIOBAMBA - ECUADOR

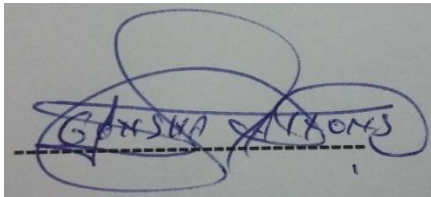
AÑO 2016

INFORME DEL TUTOR

Yo Ing. ALFONSO JAVIER GUNSHA MORALES, en mi calidad de Tutor del trabajo investigativo titulado “MODELAMIENTO DE UN SISTEMA FOTOVOLTAICO PARA MEJORAR LA EFICIENCIA DEL BLOQUE B DE INGENIERIA”, tengo a bien informar que el mencionado trabajo cumple con los requisitos exigidos para ser expuestos al público, luego se ser evaluado por el tribunal designado.

Riobamba, 11 de agosto del 2016.

Atentamente

A handwritten signature in blue ink on a grey background. The signature is stylized and appears to read 'GONSHA MORALES'. There are several loops and flourishes above the text. A horizontal dashed line is drawn across the signature.

Ing. Alfonso Gunsha

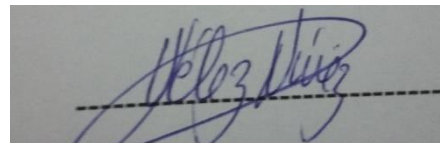
TUTOR

Los miembros del Tribunal de Graduación del proyecto de investigación “MODELAMIENTO DE UN SISTEMA FOTOVOLTAICO PARA MEJORAR LA EFICIENCIA DEL BLOQUE B DE INGENIERIA”, presentado por Nelly Janneth Carchi Chacha, Víctor Fabián Castelo Martínez y dirigida por Ing. Alfonso Javier Gunsha Morales.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente por uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

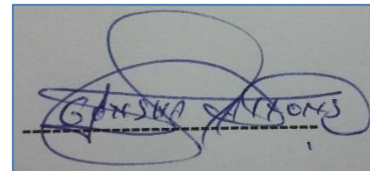
Para constancia de lo expuesto firman:

Ing. Paulina Velez
Presidente del Tribunal

A handwritten signature in blue ink, appearing to read 'Paulina Velez', written over a horizontal dashed line.

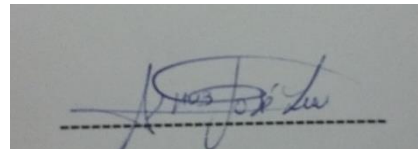
Firma

Ing. Alfonso Gunsha
Director del Proyecto

A handwritten signature in blue ink, appearing to read 'Alfonso Gunsha', written over a horizontal dashed line.

Firma

Ing. José Jínez
Miembro del Tribunal

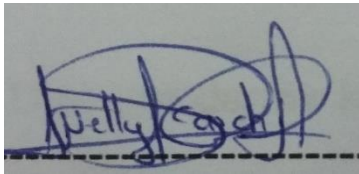
A handwritten signature in blue ink, appearing to read 'José Jínez', written over a horizontal dashed line.

Firma

AUTORÍA DE LA INVESTIGACIÓN

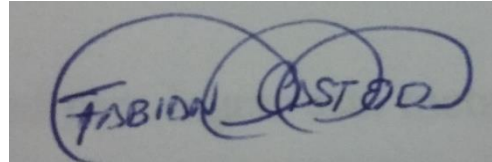
La responsabilidad del contenido de este Proyecto de Graduación, corresponde exclusivamente a Nelly Janneth Carchi Chacha, Víctor Fabián Castelo Martínez y el Director del Proyecto: Ing. Alfonso Javier Gunsha Morales y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

AUTORES



Nelly Janneth Carchi Chacha

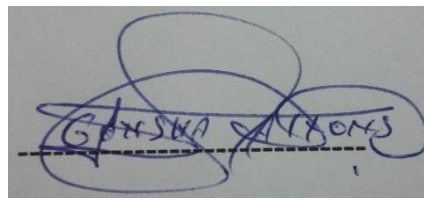
C.I. 0605747989



Víctor Fabián Castelo Martínez

C.I. 0603455489

Director del Proyecto



Ing. Alfonso Javier Gunsha Morales

C.I. 0603797853

AGRADECIMIENTO

A mis padres por sus enseñanzas, consejos y bendiciones que me han permitido seguir creciendo como persona y ser mejor cada día, a mis hermanos que han sido cómplices en toda mi vida, a mi familia y amigos por estar siempre pendientes de mí, brindándome su apoyo.

A mis profesores por compartir todos sus conocimientos y experiencias en especial a mi tutor Ing. Alfonso Gunsha quien me ha brindado toda la motivación para terminar de forma exitosa este proyecto, en especial a mi ángel de la guarda que desde el cielo me cuida y me protege y sus bendiciones siempre me acompañan en toda mi vida.

Nelly

A Dios por darme la fortaleza, el entendimiento, y la capacidad para lograr esta meta; a mis padres que con su cariño y su apoyo incondicional han contribuido con mis ideales, a mis hermanos que siempre me han brindado su apoyo en todo momento, a mis abuelitos, a mis tíos, primos, sobrinas, amigos y profesores por compartir todos sus conocimientos y experiencias. En especial al Ing. Alfonso Gunsha por la confianza brindada para desarrollar este proyecto de investigación.

Fabián

DEDICATORIA

Dedicado a Dios, porque me ha brindado su fortaleza y el valor para salir adelante y cumplir con mis objetivos planteados. A mis padres porque han sido mis guías y me han brindado todo su apoyo incondicional, en especial a mi madre que ha sido mi mejor amiga y confidente a mis hermanos que me han sabido brindar sus consejos y experiencias, a toda mi familia que me ha brindado todo su cariño y apoyo incondicional.

Nelly

Dedico este trabajo de Tesis a mis padres y a mi abuelito Víctor que con todo su amor y esfuerzo me han apoyado a cumplir mis metas, mis sueños y mis propósitos durante todos estos años.

Con todo cariño

Fabián

ÍNDICE

INFORME DEL TUTOR.....	i
AUTORÍA DE LA INVESTIGACIÓN.....	iii
AGRADECIMIENTO	iv
DEDICATORIA.....	v
ÍNDICE	vi
ÍNDICE DE GRÁFICOS	x
ÍNDICE DE TABLAS	xii
RESUMEN	xiii
SUMMARY	¡Error! Marcador no definido.
INTRODUCCIÓN	xv
CAPITULO I.....	1
1 FUNDAMENTACIÓN TEÓRICA	1
1.1 ENERGÍA RENOVABLE.....	1
1.1.1 ENERGÍA SOLAR FOTOVOLTAICA	1
1.1.2 VENTAJAS DEL ENERGÍA SOLAR FOTOVOLTAICA	2
1.1.3 DESVENTAJAS DE LA ENERGÍA FOTOVOLTAICA	2
1.2 CELDA Y PANELES SOLARES FOTOVOLTAICOS	2
1.2.1 PRINCIPIO FOTOVOLTAICO.....	2
1.2.2 CELDA FOTOVOLTAICA	4
1.2.3 ECUACIÓN CARACTERÍSTICA DE LA CELDA FOTOVOLTAICA	4
1.2.4 MODELO IDEAL DE LA CELDA FOTOVOLTAICA	6
1.2.5 CURVA CARACTERÍSTICA I-V DE LA CELDA FOTOVOLTAICA.....	7
1.2.6 FACTOR DE FORMA O FACTOR DE RELLENO (FI FACTOR FF) DE LA CELDA FOTOVOLTAICA.....	8
1.2.7 EFICIENCIA DE LA CELDA FOTOVOLTAICA.....	8
1.2.8 CIRCUITO EQUIVALENTE DE LA CELDA FOTOVOLTAICA (REAL).....	9
1.3 PANEL O MÓDULO SOLAR FOTOVOLTAICO.....	10
1.3.1 FUNCIONAMIENTO DEL PANEL SOLAR FOTOVOLTAICO	11
1.4 DISPOSITIVOS ELECTRÓNICOS	12
1.4.1 MICROCONTROLADORES.....	12

1.4.2	RELOJ EN TIEMPO REAL DS1307	14
1.4.3	PANTALLA DE CRISTAL LÍQUIDO (LCD)	14
1.4.4	CONTROLADOR DE POTENCIA DE UN MOTOR DC.....	15
1.4.5	DISPOSITIVOS DE MEDIDA (SENSORES)	16
1.4.6	ARDUINO	18
1.5	SOFTWARE UTILIZADOS.....	20
1.5.1	MIKROC	20
1.5.2	LABVIEW	21
1.5.3	MATLAB.....	23
CAPITULO II		25
2	METODOLOGÍA	25
2.1	TIPO DE ESTUDIO.....	25
2.2	POBLACIÓN Y MUESTRA.....	25
2.3	OPERACIONALIZACIÓN DE VARIABLES	26
2.4	PROCEDIMIENTOS.....	26
2.4.1	PROGRAMACIÓN DEL RELOJ EN TIEMPO REAL EN MIKRO C.....	26
2.4.2	DISEÑO DE LA ESTRUCTURA MECÁNICA EN SOLIDWORD	28
2.4.3	ETAPA DE CONTROL DEL MOTOR DC.....	30
2.4.4	SENSORES.....	33
2.4.5	ETAPA DE ADQUISICIÓN DE DATOS	35
2.5	PROCESAMIENTO Y ANÁLISIS	40
2.5.1	ANÁLISIS DEL PIC 16F877A.....	40
2.5.2	ANÁLISIS PRÁCTICO DE LA COMUNICAIÓN UART.....	40
2.5.3	TÉCNICAS DE PROCEDIMIENTO PARA EL ANÁLISIS	41
2.6	COMPROBACIÓN DE LA HIPÓTESIS	41
2.6.1	PLANTEAMIENTO DE LA HIPÓTESIS	41
2.6.2	ESTABLECIMIENTO DEL NIVEL DE SIGNIFICANCIA.....	42
2.6.3	ELECCIÓN DE LA PRUEBA DE HIPÓTESIS.....	42
2.6.4	MUESTRA.....	42
2.6.5	MEDIA.....	42
2.6.6	PRUEBA DE NORMALIDAD.....	43
2.6.7	PRUEBA T PARA MUESTRAS RELACIONADAS	43
CAPITULO III.....		46
3	RESULTADOS	46

3.1	PRUEBAS DEL RELOJ EN TIEMPO REAL.....	46
3.2	PRUEBAS DEL MOTOR DC EN LA ESTRUCTURA	46
3.2.1	PRUEBAS DE LA ADQUISICION DE DATOS	47
3.3	ANÁLISIS FINANCIERO	60
CAPITULO IV		61
4	CONCLUSIONES Y RECOMENDACIONES.....	61
4.1	CONCLUSIONES	61
4.2	RECOMENDACIONES	62
CAPITULO V		63
5	PROPUESTA	63
5.1	TÍTULO DELA PROPUESTA	63
5.2	INTRODUCCIÓN.....	63
5.3	DISCUSIÓN.....	63
CAPITULO VI.....		65
6	OBJETIVOS.....	65
6.1	OBJETIVO GENERAL	65
6.2	OBJETIVOS ESPECÍFICOS	65
6.3	FUNDAMENTACIÓN CIENTÍFICO – TÉCNICA	65
6.4	DESCRIPCIÓN DE LA PROPUESTA.....	65
6.5	DISEÑO ORGANIZACIONAL.....	66
CAPITULO VII		67
7	BIBLIOGRAFÍA	67
8	ANEXOS.....	69
ANEXO A		69
GLOSARIO DE TÉRMINOS.....		69
CÓDIGO EN MIKROC DE UN RELOJ EN TIEMPO REAL		69
ANEXO B		79
PROGRAMACIÓN DEL GIRO DEL MOTOR EN LENGUAJE MIKRO C.....		79
DATA SHEET DEL PIC 16F877A		81
CONFIGURACIÓN PARA LA LECTURA ADC		82
CONFIGURACIÓN UART PARA LA TX DE DATOS		84
ANEXO C		87
DATASHEETH PANEL SOLAR		87

DATASHEETH DEL SENSOR ACS712	88
DATASHHETH DEL SABERTOOTH 2X25 V2	90
DISEÑO DEL SEGUIDOR SOLAR.....	93
ESQUEMA DE CONEXIÓN DE LOS SENSORES	94
PROGRAMACION DEL MODELAMIENTO EN MATLAB	95
ACOPLAMIENTO SE LAS FASES DEL PROYECTO.....	102

ÍNDICE DE GRÁFICOS

<i>Figura 1: Energía solar fotovoltaica</i>	1
<i>Figura 2: Efecto fotovoltaico en una celda solar</i>	4
<i>Figura 3: Circulación de corriente en una celda fotovoltaica</i>	4
<i>Figura 4: Curva característica del diodo ideal</i>	5
<i>Figura 5: Curva característica de un fotodiodo</i>	6
<i>Figura 6: Celda fotovoltaica ideal</i>	6
<i>Figura 7: Curva característica I-V</i>	7
<i>Figura 8: Esquema eléctrico de una celda fotovoltaica real</i>	9
<i>Figura 9: Esquema ilustrativo de la composición de un panel</i>	11
<i>Figura 10: Funcionamiento de un panel solar</i>	12
<i>Figura 11: PIC 16F877A</i>	13
<i>Figura 12: Reloj en tiempo real DS1307</i>	14
<i>Figura 13: LCD</i>	15
<i>Figura 14: Sabertooth 2X25 V2.0</i>	16
<i>Figura 15: Sensor de corriente Acs712</i>	17
<i>Figura 16: Sensor de voltaje</i>	18
<i>Figura 17: Arduino mega 2560</i>	19
<i>Figura 18: Mikro C</i>	20
<i>Figura 19: Panel frontal Labview</i>	22
<i>Figura 20: Diagrama de bloques Labview</i>	23
<i>Figura 21: Software Matlab</i>	24
<i>Figura 22: Creación del Proyecto</i>	27
<i>Figura 23: Programación del reloj en tiempo real</i>	27
<i>Figura 24: Simulación reloj en tiempo real</i>	28
<i>Figura 25: Diseño de la estructura</i>	29
<i>Figura 26: Construcción de la estructura</i>	30
<i>Figura 27: Control del motor dc</i>	31
<i>Figura 28: Programación del giro del motor dc</i>	32
<i>Figura 29: Sensor de Voltaje</i>	33
<i>Figura 30: Conexión sensor Asc712</i>	35
<i>Figura 31: Sistema de adquisición de datos</i>	36
<i>Figura 32: Instalación firmware Arduino</i>	36
<i>Figura 33: Entradas y salidas SubVI Init.vi</i>	37
<i>Figura 34: Entradas y salidas SubVI close.vi</i>	38
<i>Figura 35: Panel frontal para la adquisición de datos en Labview</i>	38
<i>Figura 36: Diagrama de bloques adquisición de datos panel en movimiento</i>	39
<i>Figura 37: Diagrama de bloques adquisición de datos panel fijo</i>	39
<i>Figura 38: Comunicación UART</i>	41
<i>Figura 39: Aceptación de H1 y rechazo de H0</i>	45
<i>Figura 40: Visualización del reloj en tiempo real en el LCD</i>	46

<i>Figura 41: Pruebas del funcionamiento del motor dc.....</i>	<i>47</i>
<i>Figura 42: Sistema de adquisición de datos</i>	<i>47</i>
<i>Figura 43: Modelamiento</i>	<i>58</i>
<i>Figura 44: Modelamiento de la constante de Boltzman.....</i>	<i>58</i>
<i>Figura 45: Modelamiento de una celda fotovoltaica</i>	<i>59</i>
<i>Figura 46: Diseño organizacional del proyecto</i>	<i>66</i>

ÍNDICE DE TABLAS

<i>Tabla 1: Características Técnicas Arduino Mega 2560.....</i>	<i>19</i>
<i>Tabla 2: Características de las Variables.....</i>	<i>21</i>
<i>Tabla 3: Operacionalización de variables.....</i>	<i>26</i>
<i>Tabla 4: Media Calculada.....</i>	<i>42</i>
<i>Tabla 5: Pruebas de normalidad en SPSS.....</i>	<i>43</i>
<i>Tabla 6: Estadísticas de muestras relacionadas.....</i>	<i>43</i>
<i>Tabla 7: Correlación de muestras relacionadas.....</i>	<i>44</i>
<i>Tabla 8: Prueba de muestras relacionadas.....</i>	<i>44</i>
<i>Tabla 9: Datos adquiridos primera semana panel fijo.....</i>	<i>48</i>
<i>Tabla 10: Datos adquiridos primera semana seguidor solar.....</i>	<i>49</i>
<i>Tabla 11: Datos adquiridos segunda semana panel fijo.....</i>	<i>50</i>
<i>Tabla 12: Datos adquiridos segunda semana seguidor solar.....</i>	<i>51</i>
<i>Tabla 13: Datos adquiridos tercera semana panel fijo.....</i>	<i>52</i>
<i>Tabla 14: Datos adquiridos tercera semana seguidor solar.....</i>	<i>53</i>
<i>Tabla 15: Datos adquiridos cuarta semana panel fijo.....</i>	<i>54</i>
<i>Tabla 16: Datos adquiridos cuarta semana seguidor solar.....</i>	<i>55</i>
<i>Tabla 17: Comparación eficiencia segunda semana.....</i>	<i>56</i>
<i>Tabla 18: Comparación eficiencia tercera semana.....</i>	<i>57</i>
<i>Tabla 19: Comparación eficiencia cuarta semana.....</i>	<i>57</i>
<i>Tabla 20: Análisis financiero.....</i>	<i>60</i>

RESUMEN

El presente proyecto de investigación consiste en el modelamiento de un sistema fotovoltaico para el cual realizaremos el diseño y la construcción de un sistema de seguimiento solar utilizando el Pic. 16F877A, el mismo que controla el giro de un motor DC gradualmente mediante la programación de un reloj en tiempo real el mismo que visualizará la hora y fecha a través de un LC D, el motor DC es el dispositivo que va a permitir el movimiento del panel fotovoltaico en la estructura, el motor esta acoplado a un sistema de poleas el cual permitirá que el giro del panel sea preciso y que el viento no intervenga de manera negativa en su funcionalidad.

Los sensores están conectados al panel fijo y móvil, de tal manera que los datos recolectados durante un mes fueron utilizados para realizar la comparativa de la eficiencia entre los dos paneles, para la adquisición y el registro de datos utilizamos una interfaz en Labview – Arduino.

ABSTRACT



UNIVERSIDAD NACIONAL DE CHIMBORAZO CENTRO DE IDIOMAS INSTITUCIONAL

Ms. Lucia Larrea

10 de Agosto de 2016

ABSTRACT

The present research project consisting on the modeling of a photovoltaic system, in order to develop this we will make the design and construction of a solar tracking system using the PIC 16F877A, which controls the rotation of a DC motor gradually by programming a real time clock the same that will display the time and date through a LCD, DC motor is the device that will allow the movement of the photovoltaic panel in the structure, the engine is coupled to a pulley system which will allow the rotation of the panel is accurate and that the wind does not intervene in a negative way in its functionality.

The sensors are connected to the fixed panel and mobile, in such a way that the data collected during a month were used to carry out the comparison of efficiency between the two panels for the acquisition and recording of data we use an interface in Labview - Arduino.



INTRODUCCIÓN

En nuestro tiempo hemos podido observar que en todo el mundo ocurren varias situaciones, debido al calentamiento global que se origina por la contaminación ambiental. Todo esto ha originado la búsqueda de nuevas alternativas energéticas más limpias y eficientes.

Hoy en día las energías renovables están muy requeridas ya que su utilización no emite productos contaminantes para el medio ambiente y se obtienen a partir de fuentes naturales que son prácticamente inagotables.

La energía solar se presenta como una alternativa de mucha importancia ante el aumento del consumo energético del planeta, debido

a que la cantidad de energía del sol que llega a la superficie de la tierra en un día es diez veces más que la energía total consumida en nuestro planeta durante un año a través del efecto fotovoltaico la energía contenida en la luz del sol puede ser convertida en energía eléctrica. [1]

La energía solar ha tenido una gran importancia a nivel mundial, ya que es una energía limpia y renovable, de fácil instalación y mantenimiento.

La presente tesis está orientada a comparar la eficiencia de dos paneles solares:

El primer panel solar se coloca fijo en una determinada posición siendo esta la más eficiente para obtener la mayor radiación solar posible.

El segundo panel solar tiene la capacidad de moverse posicionándose frente a la incidencia de los rayos del sol de forma autónoma.

Además trata sobre modelar el funcionamiento de una celda fotovoltaica en Matlab, para el cual necesitaremos realizar una estructura mecánica para el panel en movimiento, para la recolección de datos que nos brindan los paneles optamos por adquirir cuatro sensores dos sensores de voltaje y dos sensores de corriente, estos sensores necesitan una interfaz para almacenar los datos en la pc la misma que la realizaremos en el módulo de Arduino.

CAPITULO I

1 FUNDAMENTACIÓN TEÓRICA

1.1 ENERGÍA RENOVABLE

Se denomina energía renovable a la energía que se adquiere de fuentes naturales supuestamente inagotables, capaces de regenerarse por medios naturales, conocidas también como energías alternativas ya que pueden reemplazar a las energías energéticas típicas, con un menor efecto contaminante, existen varias energías renovables tales como: energía solar fotovoltaica, eólica, geotérmica, mareomotriz, etc. [1]

1.1.1 ENERGÍA SOLAR FOTOVOLTAICA

La energía solar fotovoltaica se caracteriza por funcionar a bases de paneles solares que almacenan y aprovechan las radiaciones luminosas y la transforma en energía eléctrica. Los paneles solares están compuestos por celdas fotovoltaicas, que transforman la energía luminosa (fotones) en electricidad (electrones en movimiento).



Figura 1: Energía solar fotovoltaica

Fuente: <https://aprendemia.com/cursos/curso-energia-solar-fotovoltaica>

1.1.2 VENTAJAS DEL ENERGÍA SOLAR FOTOVOLTAICA

Los sistemas fotovoltaicos ofrecen varias ventajas:

- **Durabilidad:** los módulos fotovoltaicos tienen un periodo de vida de más de 25 años.
- **Bajos costos de mantenimiento:** los sistemas fotovoltaicos necesitan inspecciones periódicas y mantenimiento ocasional.
- **Reducción de contaminación:** es una fuente de energía amigable para el medio ambiente ya que no produce ninguna emisión.
- **Fiabilidad:** los arreglos fotovoltaicos impiden las fallas de energía costosa en situaciones en donde resulta difícil una operación continua.

1.1.3 DESVENTAJAS DE LA ENERGÍA FOTOVOLTAICA

Los sistemas fotovoltaicos ofrecen varias desventajas:

- **Costo inicial:** los costos de instalación son altos, requieren una gran inversión inicial.
- **Variabilidad de la radiación solar:** el estado del tiempo puede afectar considerablemente la energía de salida de cualquier sistema que está basado en el uso del sol.
- **Limitación de consumo:** no puede utilizarse más energía de la acumulada en periodos en donde no haya sol.

1.2 CELDAS Y PANELES SOLARES FOTOVOLTAICOS

1.2.1 PRINCIPIO FOTOVOLTAICO

El término fotovoltaico proviene del griego (luz) y del nombre físico italiano Volta, de donde proviene también voltio y voltaje, literalmente significa luz y electricidad, el efecto fotovoltaico es la base del proceso mediante el cual una celda fotovoltaica convierte la luz solar en electricidad.

Los Módulos solares se componen de celdas solares de silicio, estas son semiconductoras eléctricas debido a que el silicio es un material de características intermedias entre un conductor y un aislante.

Mediante el dopado de silicio con fósforo se obtiene un material con electrones libres o material con portadores de carga negativa (silicio tipo N). Realizando el mismo proceso pero agregando boro en lugar de fósforo, se obtiene un material de características inversas; esto es déficit de electrones o material con cargas positivas libres o huecos (silicio tipo P), Ambas capas separadas son eléctricamente neutras, pero al ser unidas, justamente en la unión (P-N) se genera un campo eléctrico debido a los electrones libres del silicio tipo N que ocupan los huecos de la estructura del silicio tipo P.

Al incidir la luz sobre la celda fotovoltaica, los fotones que la integran chocan con los electrones de la estructura del silicio dándoles energía y transformándolos en conductores. Debido al campo eléctrico generado en la unión (P-N), los electrones son orientados, fluyendo de la capa "P" a la capa "N". Mediante un conductor externo, se conecta la capa negativa a la positiva, generándose así un flujo de electrones que se denomina corriente eléctrica ver Figura 2, mientras la luz siga incidiendo en la celda, el flujo de electrones se mantendrá, Es por eso, que se considera inagotable a un dispositivo fotovoltaico, ya que produce energía eléctrica como respuesta a la energía lumínica que ingresa en el mismo. [2]

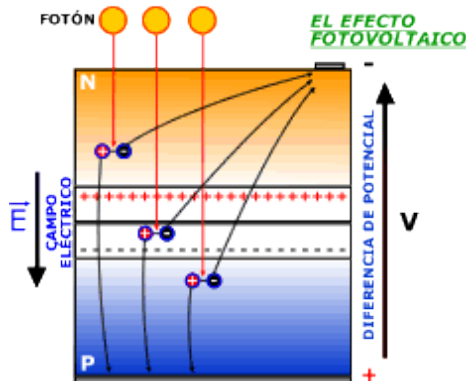


Figura 2: Efecto fotovoltaico en una celda solar

Fuente: <http://bibing.us.es/proyectos/abreproy/11375/fichero/MEMORIA%252FEstudio+filtro+iir.pdf>

1.2.2 CELDA FOTOVOLTAICA

Cuando se conecta una celda solar a una carga y se ilumina, circula una corriente I . En este caso se puede observar que la corriente circula de cátodo a ánodo, es decir internamente circula del semiconductor tipo N al tipo P (contrario al sentido de un diodo). Ver **Figura. 3**

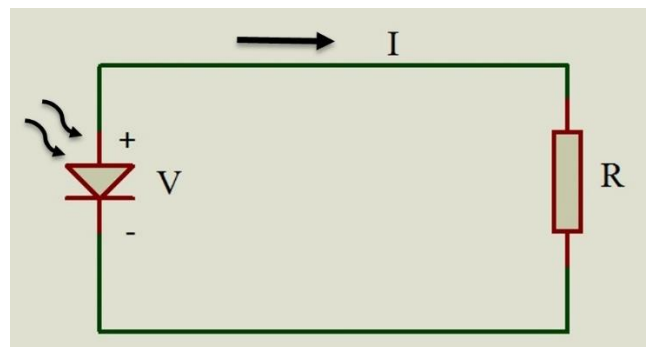


Figura 3: Circulación de corriente en una celda fotovoltaica

Fuente: Autor

1.2.3 ECUACIÓN CARACTERÍSTICA DE LA CELDA FOTOVOLTAICA

La celda solar se puede ver como un diodo, en general de silicio, diseñada para extender la absorción de fotones y disminuir la reflexión.

La ecuación de un diodo ideal, se puede expresar por:

$$I = I_s \left[e^{\frac{qV}{kT}} - 1 \right]$$

Ecuación 1

I_s : Corriente inversa de saturación

V : Tensión aplicada

q : Carga del electrón ($1,6 * 10e^{-19}C$)

k : Constante de boltzman ($1.38 * 10e^{-23} \frac{J}{K}$)

T : Temperatura absoluta (K) 300 C

kT/q : Potencial térmico VT (para el silicio 25°C es igual a 25,7mV) Tension de umbral del diodo

Al graficar de la expresión del diodo se obtiene la curva característica $I * V$.



Figura 4: Curva característica del diodo ideal

Fuente: <http://www.fceia.unr.edu.ar/eca1/files/teorias/Diodos%20-%202012.pdf>

$$VT = \frac{k * Tc}{q}$$

Ecuación 2

k : Constante de Boltzman

T_c : Temperatura de la celda en Kelvin (K)

q : Carga del electrón

La incidencia de la luz tiene efecto de mover la curva $I \cdot V$ hacia abajo en el cuarto cuadrante tal como se muestra en la **Figura.5**.

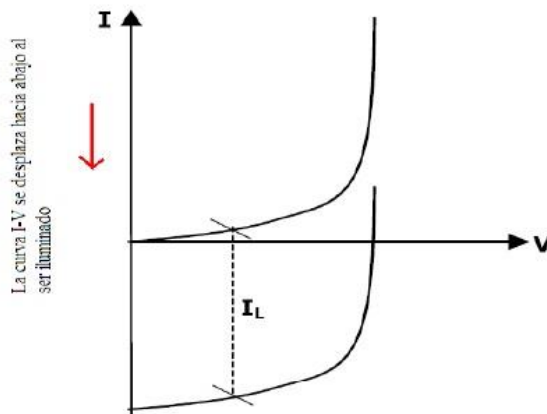


Figura 5: Curva característica de un fotodiodo

Fuente: <http://www.bolanosdj.com.ar/TEORIA/SENSORES OPTICOS.PDF>

1.2.4 MODELO IDEAL DE LA CELDA FOTOVOLTAICA

El modelo o circuito equivalente ideal de una celda fotovoltaica se simboliza por una fuente de corriente en paralelo con un diodo en la **Figura 6** se muestra el modelo ideal de una celda fotovoltaica cuando se conecta a una carga.

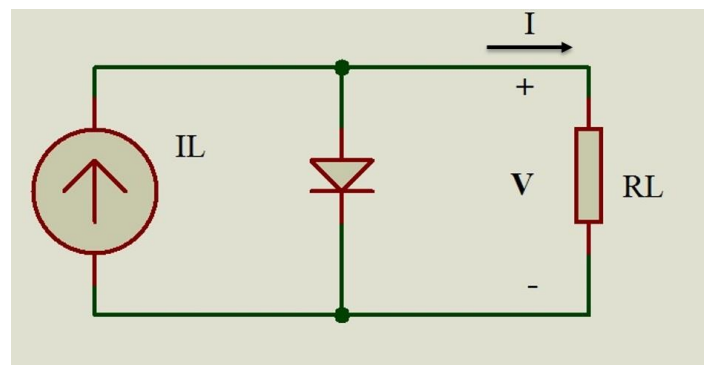


Figura 6: Celda fotovoltaica ideal

Fuente: Autor

1.2.5 CURVA CARACTERÍSTICA I-V DE LA CELDA FOTOVOLTAICA

La curva característica de un panel fotovoltaico, también llamada curva de intensidad-voltaje (curva I-V), representa los valores de tensión y corriente, medidos experimentalmente, de un típico panel fotovoltaico sometido a unas determinadas condiciones constantes de insolación y temperatura, y se puede representar por la siguiente ecuación:

$$I = IL - IS \left[e^{\frac{V}{m*VT}} - 1 \right]$$

Ecuación 3

Por tanto, la curva I-V característica resultaría como se muestra en la **Figura.7**

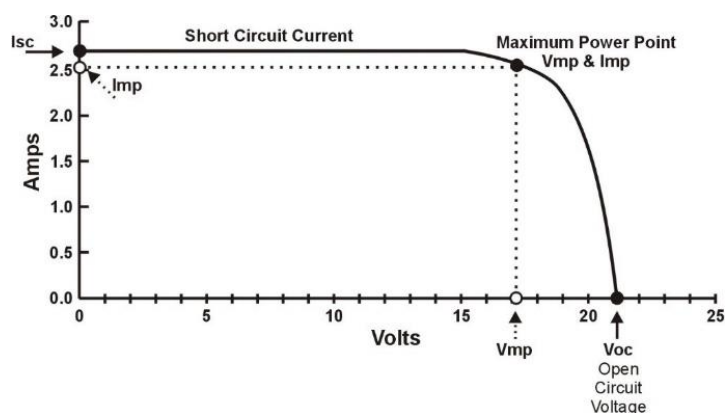


Figura 7: Curva característica I-V

Fuente: <http://globalem.co/wp-content/uploads/2015/06/Curva-I-V.jpg>

Si la celda está cortocircuitada ($R=0$), se genera una corriente de corto circuito cuyo sentido es el mismo que el de la corriente inversa de saturación de la unión P-N (diodo). Entonces $V=0$ y la corriente de cortocircuito de la celda es:

$$I_{sc} = I_L$$

Ecuación 4

Si la celda está en circuito abierto ($R=\infty$), la acumulación de cargas de signos diferentes en los 2 costados de la unión genera una tensión de circuito abierto V_{oc} .

Entonces $I=0$ y la tensión de la celda es:

$$V_{oc} = n \cdot VT * \ln\left(1 + \frac{IL}{I_S}\right)$$

Ecuación 5

Se puede observar que V_{oc} aumenta logarítmicamente con la irradiación incidente. Para tener una idea del orden de magnitud, se puede decir que una célula fotovoltaica de silicio monocristalino genera un voltaje de circuito abierto) entorno a los 0,7 V y una corriente de corto circuito (I_{sc}) que depende del área de la celda (aproximadamente 3 A para un área de 100 m^2). [3]

1.2.6 FACTOR DE FORMA O FACTOR DE RELLENO (FI FACTOR FF) DE LA CELDA FOTOVOLTAICA.

El factor de relleno es la relación de la potencia máxima de una celda solar para el producto de la tensión de circuito abierto y la corriente de cortocircuito. Es una medida de la cuadratura de la celda solar y también es el área del rectángulo más grande que se ajuste en la curva IV. [4]

$$FF = \frac{V_{oc} - \ln(V_{oc} + 0.72)}{V_{oc} + 1}$$

Ecuación 6

1.2.7 EFICIENCIA DE LA CELDA FOTOVOLTAICA.

El factor de eficiencia de conversión de energía solar de una celda (η), es el porcentaje de potencia convertida (de la luz solar absorbida por la celda) en energía eléctrica capaz de hacer operar un circuito. Es otro parámetro para comparar el rendimiento de una célula solar. La eficiencia se define como la relación de producción de energía de la célula solar a la energía de entrada del sol. [4]

$$P_{max} = V_{OC} I_{SC} FF$$

Ecuación 7

$$\eta = \frac{V_{OC} I_{SC} FF}{P_{in}}$$

Ecuación 8

Donde

V_{OC} : es la tensión en circuito abierto

I_{SC} : es la corriente de cortocircuito

FF : es el factor de relleno

η : es la eficiencia.

1.2.8 CIRCUITO EQUIVALENTE DE LA CELDA FOTOVOLTAICA (REAL)

La **Figura 8** se muestra el circuito equivalente real de una celda fotovoltaica.

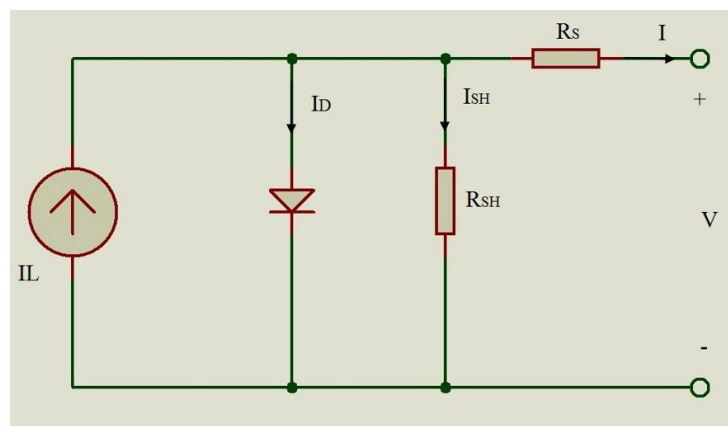


Figura 8: Esquema eléctrico de una celda fotovoltaica real
Fuente: Autor

El comportamiento de una celda cuando funciona como generador de corriente puede explicarse como la diferencia entre la corriente fotogenerada I_L debida a la generación de portadores en la iluminación (corriente fotovoltaica) y a la corriente del diodo que corresponde a la corriente de saturación, La ecuación describe adecuadamente la característica V-I de la mayoría de las celdas fotovoltaicas [5]

$$I = I_L - I_0 \left(\exp \left[\frac{q(V + IR_S)}{nkT} \right] - 1 \right) - \frac{V + IR_S}{R_{SH}}$$

Ecuación 9

Donde:

I_0 : Corriente de saturación inversa (Amper)

I_L : Corriente fotogenerada

n : Factor de idealidad del diodo (1 de diodo ideal)

q : Carga elementary (1.6×10^{-19})

k : Constante de Boltzmann

T : Temperatura absoluta

R_{SH} : Resistencia shunt

R_S : Resistencia parasita

1.3 PANEL O MÓDULO SOLAR FOTOVOLTAICO

Una celda solar suministra muy poca energía, y a muy baja tensión (superior a 0.5 V). Además, una sola celda es frágil y muy difícil de comercializar. El fabricante, agrupándolas para procurar que trabajen como una sola, busca suministrar niveles de tensión y potencia adecuados a cada aplicación, y las protege de los factores climatológicos adversos. Es lo que se llama panel o módulo solar fotovoltaico.

En el panel se asocian eléctricamente un determinado número de células solares y se protege todo el compacto sellándolo al vacío, en la **Figura. 9** se muestra un esquema ilustrativo de la composición de un panel.

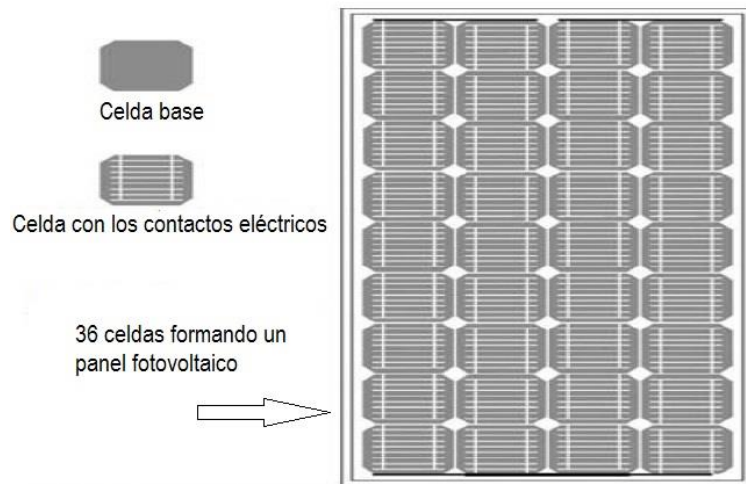


Figura 9: Esquema ilustrativo de la composición de un panel
 Fuente: https://www.ujaen.es/investiga/solar/07cursosolar/home_main_frame/04_componen/01_generator/01_basico/images/compo_panel.gif

1.3.1 FUNCIONAMIENTO DEL PANEL SOLAR FOTOVOLTAICO

Los paneles solares están formados por dos semiconductores ubicados uno al frente del otro. El panel, al ser traspasado por la luz solar, los rayos entran en contacto con el semiconductor negativo (Tipo-N) ubicado en la parte superior del panel solar, los electrones son sacados de los átomos del semiconductor, los cuales pasan por un circuito externo hacia el semiconductor positivo (Tipo-P) ubicado en la parte inferior del panel. Esto produce un ciclo donde los electrones al entrar en contacto con el semiconductor positivo son enviados de vuelta al semiconductor negativo formando un flujo que genera la electricidad. La energía adquirida por el panel es enviada al banco de batería, a través del regulador de carga, que regula la potencia traspasada a la batería de acuerdo con su carga.

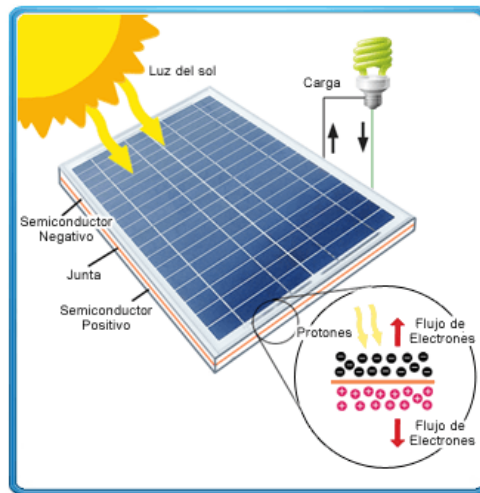


Figura 10: Funcionamiento de un panel solar

Fuente: http://www.damiasolar.com/productos/placas_solares/panel-solar-ecosolar-100w-monocristalino_da0093_15

1.4 DISPOSITIVOS ELECTRÓNICOS

1.4.1 MICROCONTROLADORES

Un microcontrolador es un circuito integrado de alta escala de integración VSLI (Very Long Scale Integration) que contiene internamente una unidad central de procesos CPU (Central Process Unit) memoria para datos, temporizadores, fuentes de interrupción y otros recursos necesarios para el desarrollo de aplicaciones, si bien un microcontrolador incluye los elementos necesarios para ser considerado como una computadora en un circuito integrado, no es tratado como tal, ya que es utilizado frecuentemente para desempeñar funciones de control interactuando con el mundo real. [6]

1.4.1.1 MICROCONTROLADOR 16F877A

Es un microcontrolador con memoria de programa tipo FLASH, lo que representa gran facilidad en el desarrollo de prototipos y en su aprendizaje ya que no se requiere borrarlo con luz ultravioleta como las versiones EPROM, sino que permite reprogramarlo nuevamente sin ser borrado con anterioridad.

El PIC16F877A es un microcontrolador de Microchip Technology fabricado en tecnología CMOS, su consumo de potencia es muy bajo y además es completamente estático, esto quiere decir que el reloj puede detenerse y los datos de la memoria no se pierden.

Todo microcontrolador requiere un circuito externo que le indique la velocidad a la que debe trabajar. Este circuito, que se conoce como oscilador o reloj, es muy simple pero de vital importancia para el buen funcionamiento del sistema. El PIC16F877 puede utilizar cuatro tipos de oscilador diferentes. Estos tipos son: [7]

- RC. Oscilador con resistencia y condensador.
- XT. Cristal (por ejemplo de 1 a 4 MHz).
- HS. Cristal de alta frecuencia (por ejemplo 10 a 20 MHz).
- LP. Cristal para baja frecuencia y bajo consumo de potencia.

En el momento de programar o “quemar” el microcontrolador se debe especificar qué tipo de oscilador se usa.

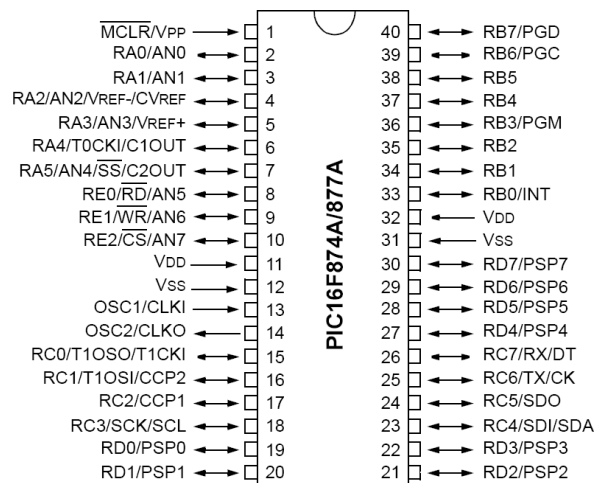


Figura 11: PIC 16F877A

Fuente: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>

1.4.2 RELOJ EN TIEMPO REAL DS1307

Este pequeño circuito integrado es uno de los más populares relojes RTC (Real Time Clock) del mercado por su sencillez de uso y por su confiabilidad a largo plazo, es un dispositivo de bajo consumo de energía, completo con código binario decimal (BCD), reloj/calendario más 56 bytes de NV SRAM. Dirección y datos son transferidos a través de 2 hilos serie, bus bi-direccional. El reloj/calendario provee información de, segundos, minutos, horas, día, fecha, mes y año. [8]

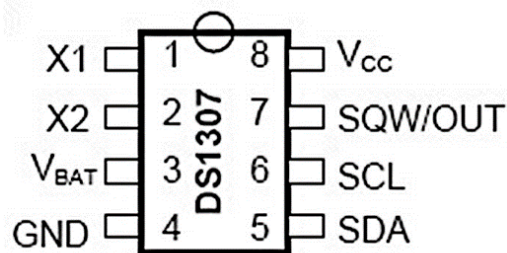


Figura 12: Reloj en tiempo real DS1307

Fuente: <http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>

1.4.3 PANTALLA DE CRISTAL LÍQUIDO (LCD)

Los display de caracteres LCD, son módulos prefabricados que contienen controladores incluidos. Estos displays cuentan con un bus de datos y un bus de control, para el manejo de estos dispositivos, los displays LCD, permiten graficar los caracteres contemplados en el código ASCII. Además del código ASCII, los displays LCD admiten graficar hasta 8 caracteres diseñados por el desarrollador, otra característica fundamental de los LCD, es la conexión del bus de datos, físicamente tienen 8 bits, pero es posible configurar las conexiones con solo 4 bits, a continuación observaremos la apariencia de un LCD. [9]

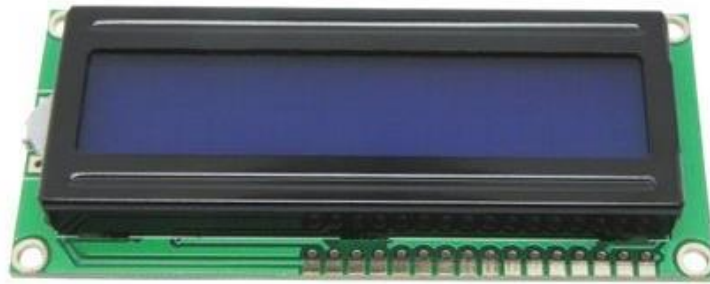


Figura 13: LCD

Fuente: Diseño y simulación de sistemas microcontrolados en lenguaje C

1.4.4 CONTROLADOR DE POTENCIA DE UN MOTOR DC

1.4.4.1 SABERTOOTH 2X25 V2.0

Sabertooth 2x25 V2.0 le permite controlar dos motores con: tensión analógica, control de radio, serie y paquetizado serial, Sabertooth tiene modos de funcionamiento independientes dirección y velocidad +, por lo que es ideal para el usuario el modo de funcionamiento se establece con los computadores DIP que tiene integrado. A continuación se detallan algunas características del dispositivo. [10]

- Corriente continua de 25 Amperios por canal.
- El valor de corriente por canal puede llegar a 50 Amperios por pocos segundos.
- Voltaje nominal de 6 a 30 V.
- Voltaje máximo absoluto de 33.6 V.
- Frecuencia de conmutación ultrasónica (32 KHz).
- Cuenta con protección térmica y sobre corrientes.
- Modos de entrada para control: Voltaje análogo, serial simplificado y paquetizado.
- Modos de operación: independiente y velocidad + dirección.
- Los modos de operación se seleccionan mediante DIP switches.

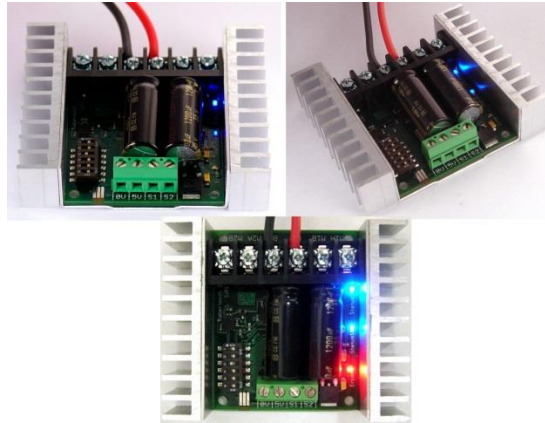


Figura 14: Sabertooth 2X25 V2.0

Fuente: <https://www.dimensionengineering.com/datasheets/Sabertooth2x25.pdf>

1.4.5 DISPOSITIVOS DE MEDIDA (SENSORES)

1.4.5.1 SENSOR DE CORRIENTE ACS712 30 A

El sensor Acs712 que se muestra en la **Figura. 15** es un sensor de corriente por efecto hall, que proporciona una solución económica y precisa para medir corriente en AC o DC, ya sea en ambientes industriales o comerciales. Este Sensor funciona transformando un campo magnético generado del paso de la corriente por un alambre de cobre interno en el sensor, y convirtiendo este campo en un voltaje variable, esto significa que a mayor cantidad de corriente que tengamos, mayor voltaje vamos a tener en un pin, El modelo que utilizamos es el ACS712ELCTR-30A-T que mide hasta 30 A. a continuación describiremos algunas características técnicas. [11]

- Bajo ruido en la ruta de la señal
- Ancho de banda del dispositivo se establece se establece a través del pin de filtro
- 5 μ s tiempo de subida en respuesta al paso de la corriente de entrada
- Ancho de banda de 80Khz
- Salida total de error de 1,5% a $T_a = 25^\circ \text{C}$
- 1.2 $\text{m}\Omega$ resistencia de interna del conductor

- Tensión de 2,1 kV RMS mínima de aislamiento en el pin 1-4 al pin 5-8
- Operación con una sola alimentación de 5V
- Sensibilidad de salida 66-185 mV /A
- Tensión de salida proporcional a las corrientes AC o DC
- Tensión extremadamente estable de la salida compensada
- Histéresis magnética casi 0
- Salida Proporcional de tensión de alimentación

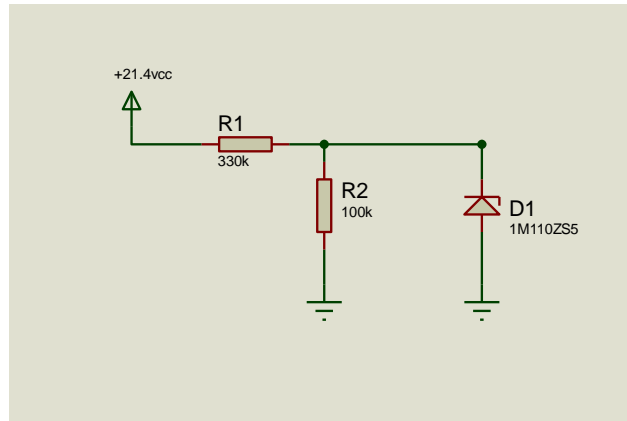


Figura 15: Sensor de corriente Acs712

Fuente: <http://pdf1.alldatasheet.com/datasheet-pdf/view/168326/ALLEGRO/ACS712.html>

1.4.5.2 SENSOR DE VOLTAJE

En nuestro caso el sensor de voltaje no es más que un divisor de voltaje que es un circuito simple que convierte un voltaje grande en uno más pequeño. Usando solo dos resistencias en serie, un voltaje de entrada y un diodo zener para asegurar la salida del voltaje, podemos crear un voltaje de salida que es una fracción del de entrada. En la siguiente figura mostramos el sensor de voltaje que utilizamos.



*Figura 16: Sensor de voltaje
Fuente. Autor*

1.4.6 ARDUINO

Es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware libre, flexible y fácil de usar, Arduino puede tomar información del entorno a través de sus pines de entrada, para esto toda una gama de sensores puede ser usada y puede afectar aquello que lo rodea controlando luces, motores y otros actuadores. El micro controlador de la placa de Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden efectuarse sin necesidad de conectarlo a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software.

El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields) que amplían las características de funcionamiento de la placa Arduino. Asimismo posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación serial con el computador. [12]

1.4.6.1 ARDUINO MEGA 2560

El Arduino Mega es probablemente el microcontrolador más capaz de la familia Arduino. Posee 54 pines digitales que funcionan como entrada/salida; 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un botón de reset y una entrada para la alimentación de la placa.

La comunicación entre la computadora y Arduino se produce a través del puerto serie, sin embargo posee un convertidor usb-serie, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB como el que utilizan las impresoras. [12]



Figura 17: Arduino mega 2560

Fuente: http://img.dxcn.com/productimages/sku_256335_4.jpg

ESPECIFICACIONES TÉCNICAS

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20mA
DC Current for 3.3V Pin	50mA
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Tabla 1: Características Técnicas Arduino Mega 2560

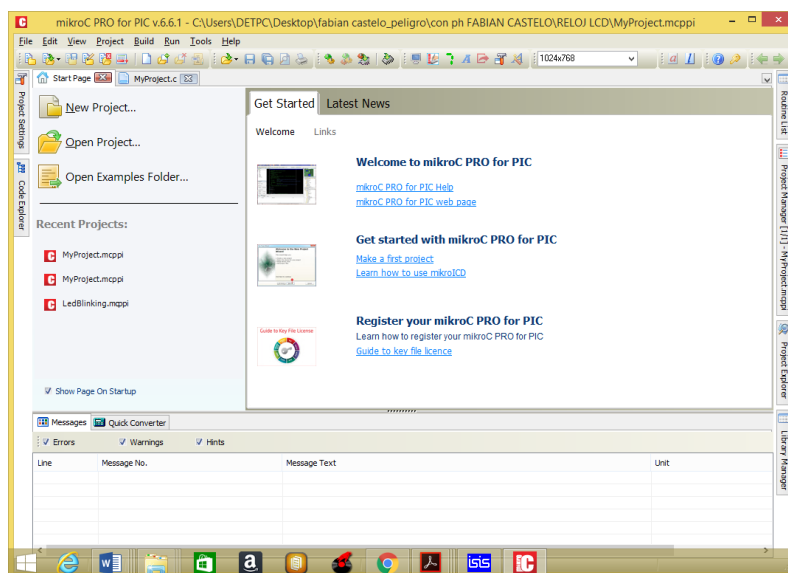
Fuente: Autor

1.5 SOFTWARE UTILIZADOS

1.5.1 MIKROC

“Es un paquete de software con una amplia variedad de ayudas y herramientas que facilita la creación de proyectos aplicativos para micro controladores de la familia 12F, 16F, 18F es un lenguaje compilador se basa en programar un código de máquina que contiene una a una las instrucciones del programa”. [13]

En la **Figura 18** podemos apreciar la apariencia visual de Mikro C.



*Figura 18: Mikro C
Fuente: Autor*

El compilador de alto nivel en lenguaje C utiliza estructuras que facilitan la programación, optimiza las operaciones matemáticas y los procesos por medio del uso de funciones definidas, así como también el uso de un conjunto de variables, de diferentes tipos como: carácter, entero, punto decimal.

“El compilador crea automáticamente el código ensamblador y a su vez un código similar consignado a un archivo con extensión *.hex, este archivo programa eléctricamente al micro controlador, también se puede realizar una simulación computacional”. [14]

En la **Tabla 2** se puede observar las características de las variables en Mikro C.

Tipo de variable	Tamaño en Bytes	Valores que soporta
Bit	1	0 ó 1
Char	1	-127 a 127
Short	1	-127 a 127
Int	2	-32767 a 32767
Long	4	-2147483647 a 2147483647
Float	4	-1.5x10 ⁴⁵ a 3.4x10 ³⁸
Double	4	-1.5x10 ⁴⁵ a 3.4x10 ³⁸
unsigned char	1	0 a 255
unsigned short	1	0 a 255
unsigned int	2	0 a 65535
unsigned long	4	0 a 4294967295

*Tabla 2: Características de las Variables
Fuente. Autor*

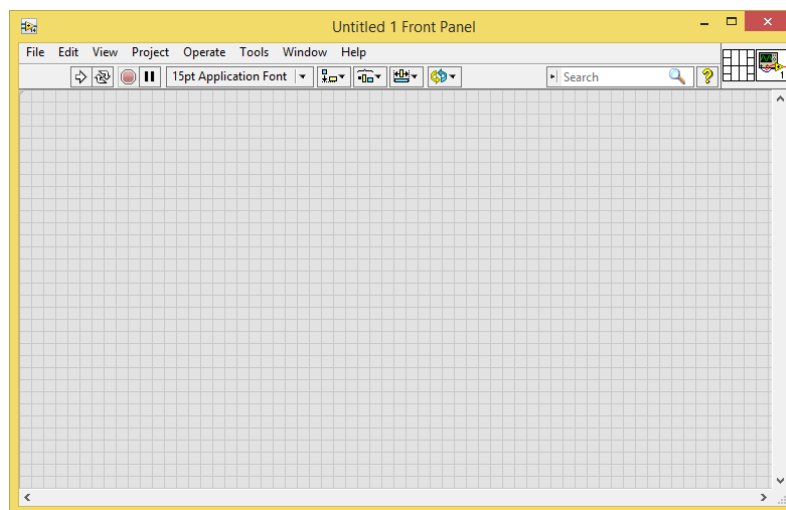
1.5.2 LABVIEW

Al lenguaje de programación que usa Labview también se le llama lenguaje G. La mayoría de los lenguajes se basan en una programación imperativa, que es simplemente una sucesión de operaciones. Sin embargo, el lenguaje G no usa una programación imperativa sino una ejecución basada en el flujo de datos.

Un programa en Labview consiste básicamente en una serie de funciones unidas mediante cables. Los datos circulan o fluyen por los cables. Una función solo podrá ejecutarse cuando tenga disponibles todos los datos que le sirven como entradas. [14]

1.5.2.1 PANEL FRONTAL

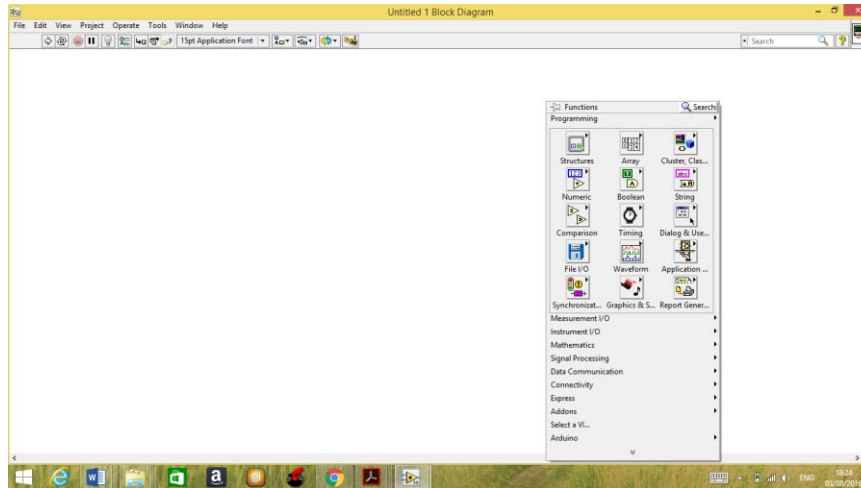
Se trata de la interfaz gráfica del VI con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc. Cada uno de ellos puede estar definido como un control o un indicador.



*Figura 19: Panel frontal Labview
Fuente: Autor*

1.5.2.2 DIAGRAMA DE BLOQUES

El diagrama de bloques constituye el código fuente del VI. En el diagrama de bloques es donde se realiza la implementación del programa del VI para controlar o realizar cualquier proceso de las entradas y salidas que se crearon en el panel frontal. El diagrama de bloques incluye funciones y estructuras integradas en las librerías que incorpora LabView.



*Figura 20: Diagrama de bloques Labview
Fuente: Autor*

1.5.3 MATLAB

Es un sistema es un sistema de trabajo interactivo y una herramienta importante para cualquier tarea que requiera cálculos matriciales, ya sea que involucren ecuaciones, sistemas característicos, mínimos cuadrados, etc, y la visualización grafica de los mismos. Se pueden resolver problemas numéricos relativamente complejos sin necesidad de escribir un programa para ello. Una de las capacidades más atractivos es la de realizar una amplia variedad de gráficos en dos y tres dimensiones, además tiene una gran capacidad de expansión ya que permite que el usuario defina sus propias funciones, es por ello que es considerado como un lenguaje de programación para cálculos técnicos y científicos donde se encuentran implicados elevados cálculos matemáticos y la visualización gráfica. [15]

En la **Figura 21** podemos apreciar la apariencia visual de Matlab.

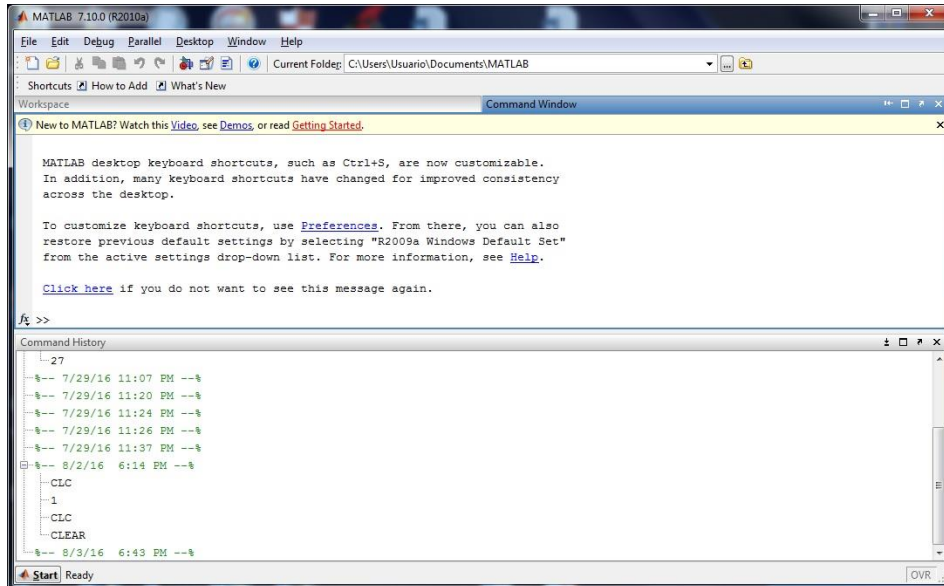


Figura 21: Software Matlab
Fuente: Autor

CAPITULO II

2 METODOLOGÍA

2.1 TIPO DE ESTUDIO

El presente trabajo está orientado a la línea de investigación de los sistemas fotovoltaicos.

Los tipos de estudio empleados en el presente proyecto de investigación son:

Método Analítico: el método de estudio se fundamenta en el análisis de los dispositivos que conforman el sistema fotovoltaico a modelar.

Investigativa: en este trabajo se necesitó recolectar la información para el diseño y la construcción del seguidor solar y del sistema de adquisición y registro de datos que se obtuvieron del panel fijo y del panel en movimiento.

De campo mediante esta investigación se presenta la obtención de los datos conseguidos del sistema de adquisición de datos posteriormente con ellos se realizara la comparación entre el seguidor solar y el panel solar fijo.

Observación: Aplicado a la práctica y utilización de los diferentes circuitos utilizados en el diseño e implementación, para el modelamiento del sistema fotovoltaico.

2.2 POBLACIÓN Y MUESTRA

El proyecto se implementará para los estudiantes, docentes y personal administrativo del bloque B de Ingeniería de la Universidad Nacional de Chimborazo, con una muestra de 124.

2.3 OPERACIONALIZACIÓN DE VARIABLES

En la **Tabla 3** se muestra la operacionalización de variables.

VARIABLES	DIMENSION	INDICADORES
Panel Solar.	Características de energía limpia y pura.	Estudio de las horas pico. Buenas condiciones.
Edificio del bloque B de ingeniería.	Características del edificio.	Número de espacios disponibles (oficinas, aulas) en el edificio.

*Tabla 3: Operacionalización de variables
Fuente: Autor*

2.4 PROCEDIMIENTOS

2.4.1 PROGRAMACIÓN DEL RELOJ EN TIEMPO REAL EN MIKRO C

El reloj en tiempo real está desarrollado el programa Mikro C, que facilita la creación de proyectos aplicativos para micro controladores de la familia 12F, 16F, 18F.

Procedemos a crear el programa en Mikro C, escogiendo la numeración del microcontrolador con el que vamos a trabajar y la frecuencia del reloj, como se observa en la **Figura 22**.

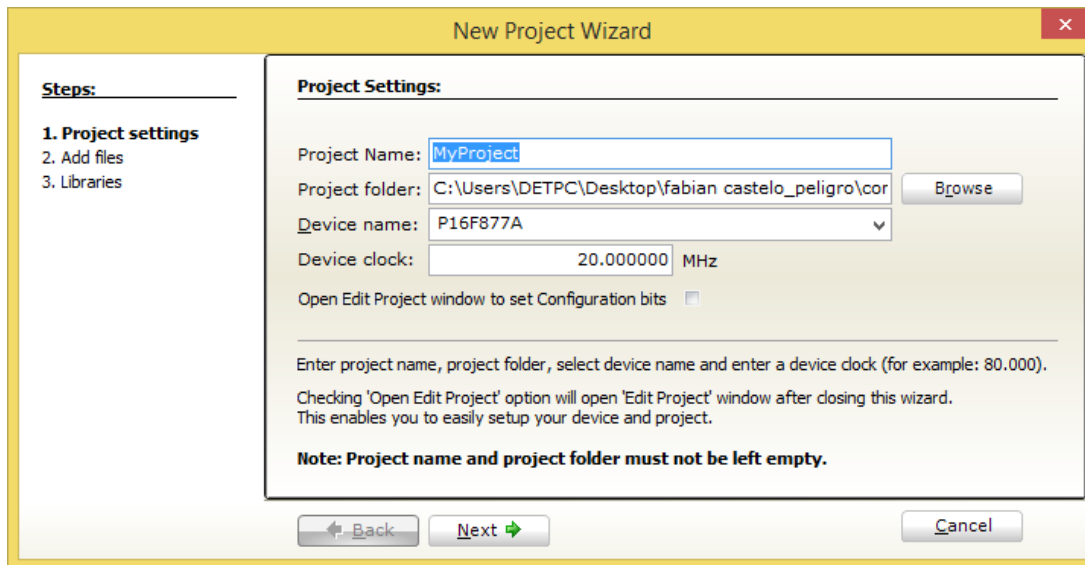


Figura 22: Creación del Proyecto
Fuente: Autor

A continuación procedemos con la programación para que en el LCD se muestre los datos del reloj en tiempo real, el código completo de la programación del reloj en tiempo real se puede observar en el **Anexo A**.

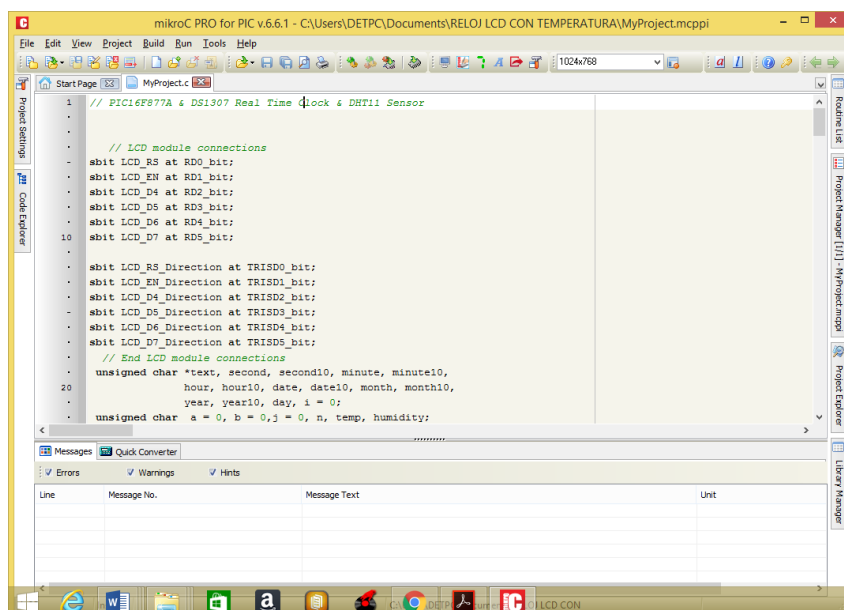
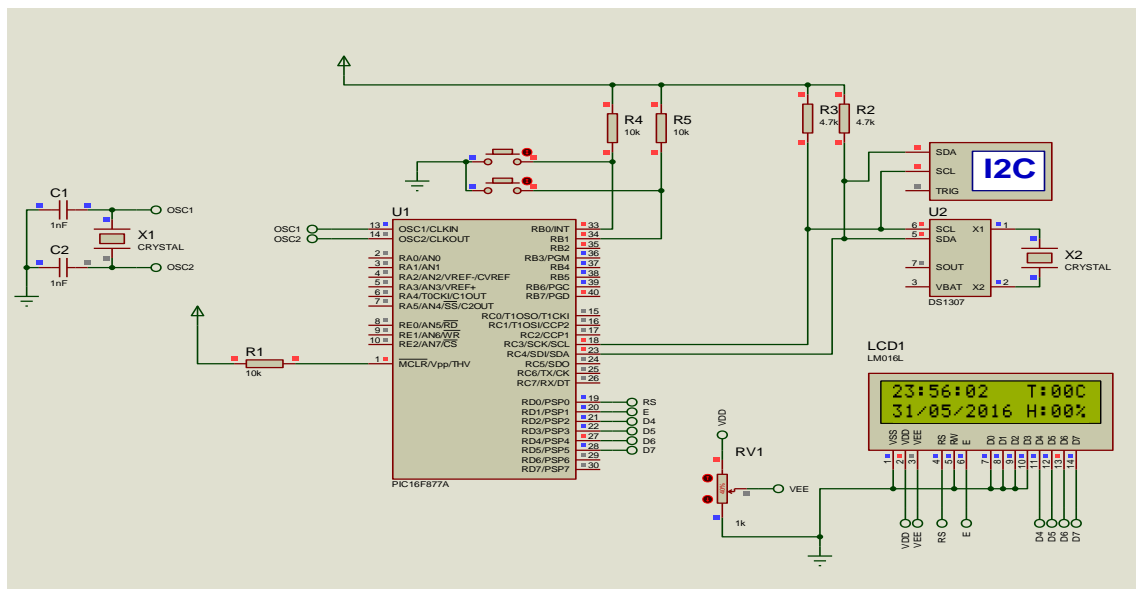


Figura 23: Programación del reloj en tiempo real
Fuente: Autor

2.4.1.1 SIMULACIÓN DEL RELOJ EN TIEMPO REAL EN PROTEUS

Realizamos la simulación del circuito en Proteus, en el cual conectamos los pines 18(RC3) y 23(RC4) del PIC 16F877A a los pines 6(SCL) y 5(SDA) del integrado DS1307 para lograr obtener los datos necesarios trabajamos con una comunicación serial I2C.

En la simulación se ubicara un LCD, el cual nos permitirá visualizar la fecha y la hora de nuestro reloj, como se muestra en la **Figura 24**.

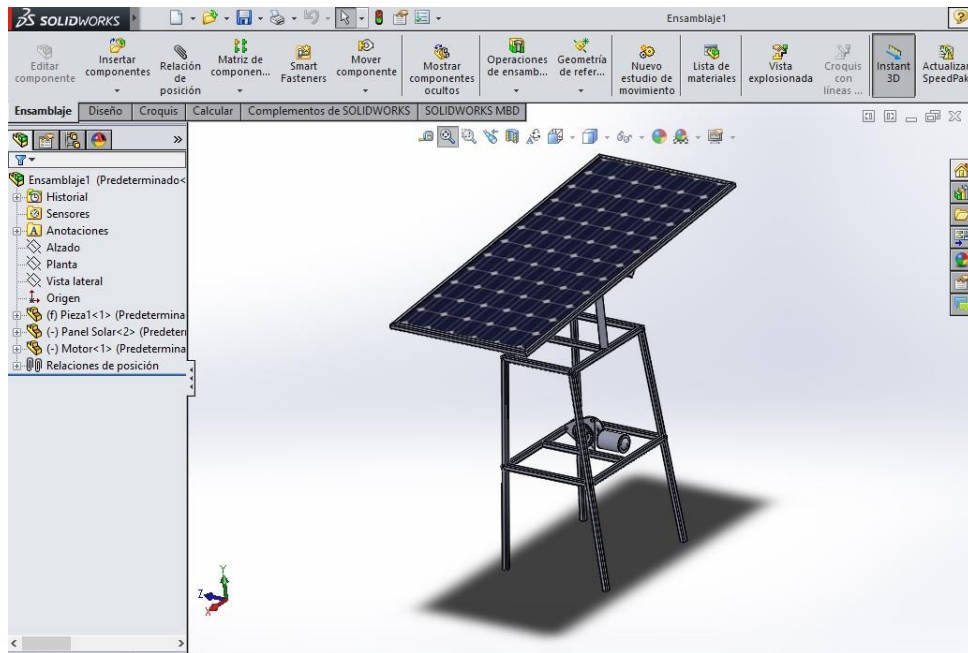


*Figura 24: Simulación reloj en tiempo real
Fuente: Autor*

2.4.2 DISEÑO DE LA ESTRUCTURA MECÁNICA EN SOLIDWORD

Para el diseño de nuestra estructura trabajamos con el software SOLIDWORD, el cual nos permite realizar un modelado mecánico en 3D, además de desarrollar productos de mejor calidad y más perfectos, dejándonos observar hasta el más mínimo detalle de nuestra estructura y de nuestro motor, de esta forma trabajamos de manera más rápida y productiva ahorrando

tiempo. El diseño de la estructura debe soportar el peso del panel solar y debe permitir el libre movimiento angular del motor.



*Figura 25: Diseño de la estructura
Fuente: Autor*

2.4.2.1 CONSTRUCCIÓN DE LA ESTRUCTURA

Después de realizar la simulación procedemos a construir la estructura para el seguidor solar, tomando en cuenta las medidas y características del simulador SolidWord.

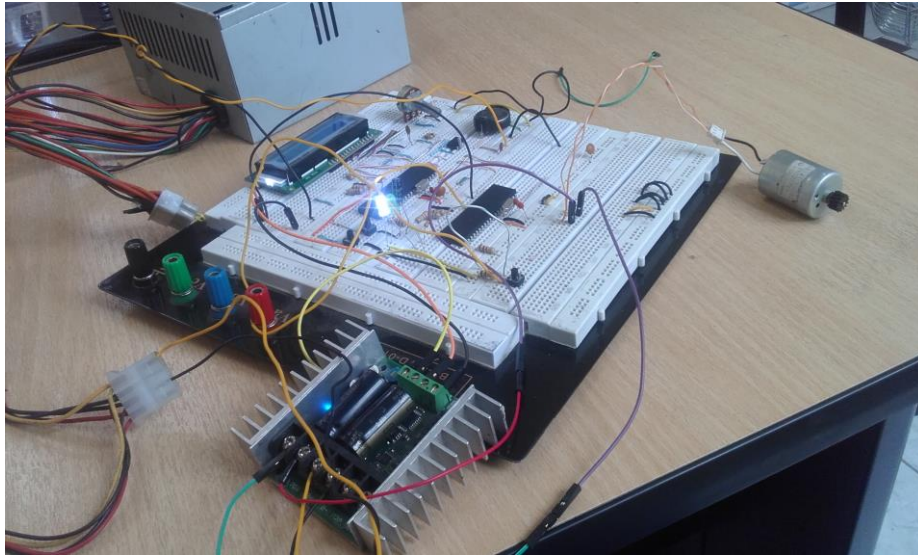
Nuestro seguidor solar está basado en un sistema de poleas, de esta manera obtendremos un mejor torque a la hora de girar el motor.



*Figura 26: Construcción de la estructura.
Fuente: Autor*

2.4.3 ETAPA DE CONTROL DEL MOTOR DC

En esta etapa utilizamos el controlador para motores Sabertooth 2x25 que permite controlar el giro del motor mediante la comunicación serial que recibe del microcontrolador, para la partida y el regreso del panel.



*Figura 27: Control del motor dc.
Fuente: Autor*

2.4.3.1 PROGRAMACIÓN DEL MOTOR DC

El programa está hecho en MikroC, para realizar la programación en el microcontrolador 16F877A es necesario tomar en cuenta, los grados y el tiempo en las que vamos a mover el panel solar, para esto debemos realizar los cálculos necesarios.

Nos hemos planteado trabajar desde las 8h00 am hasta las 18h00 pm que serían 10 horas y son 50 grados en los que vamos a mover el panel desde el inicio hasta el final.

DATOS:

Horas= 10

Grados= 50

CÁLCULOS:

$$Pos = \frac{50 \text{ grados}}{10 \text{ horas}}$$

$$Pos = 5$$

$$1 \text{ Hora} = 60 \text{ minutos}$$

$$1 \text{ minuto} = 60 \text{ segundos}$$

2.4.4 SENSORES

2.4.4.1 SENSOR DE VOLTAJE

Para recolectar los datos de voltaje proporcionados por el panel solar, simplemente realizamos un divisor de voltaje y necesitamos hacer un acoplamiento de una constante k en los cálculos para poder guardar los datos en la interface de Labview.

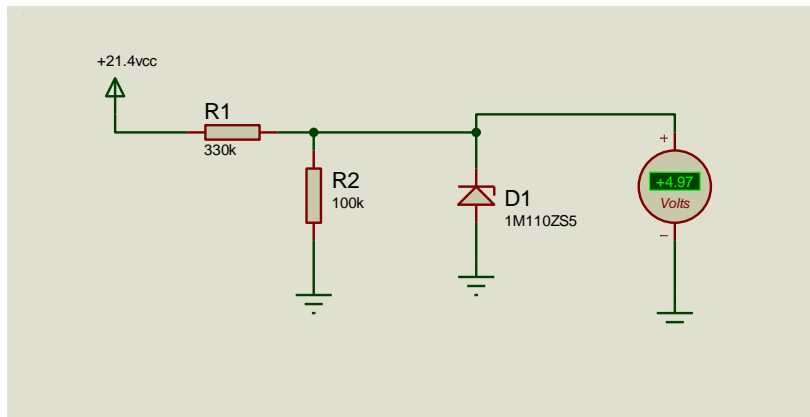


Figura 29: Sensor de Voltaje
Fuente: Autor

2.4.4.1.1 CÁLCULOS PARA EL SENSOR DE VOLTAJE

DATOS

$$V_o = 5V$$

$$V_i = 21.4V$$

$$R_2 = 100k$$

$$V_o = \frac{R_1}{R_1 + R_2} V_i$$

$$5V = \frac{100k}{R_1 + 100k} 21.4$$

$$5V(R_1 + 100k) = (100k) 21.4$$

$$5VR_1 + 500K = (100k) 21.4$$

$$5VR_1 = (100k)21.4 - 500k$$

$$5VR1 = 2100k - 500k$$

$$R1 = \frac{1.68k}{5}$$

$$R1 = 330k$$

$$Vi = \frac{R1 + R2}{R1} Vo$$

$$Vi = k Vo$$

$$k = \frac{R1 + R2}{R1}$$

$$k = \frac{330k + 100k}{100k}$$

$$k = \frac{430k}{100k}$$

$$k = 4.3$$

2.4.4.2 SENSOR DE CORRIENTE

Para recolectar los datos de corriente proporcionados por el panel solar, utilizamos el sensor Acs712 de 30 amperios el cual estará conectado al Arduino mega 2560 y dicho sensor necesita ser conectado a una carga para su funcionamiento como se muestra en la **Figura 30**, además realizamos un acoplamiento de una constante k en los cálculos para poder guardar los datos en la interface de Labview.

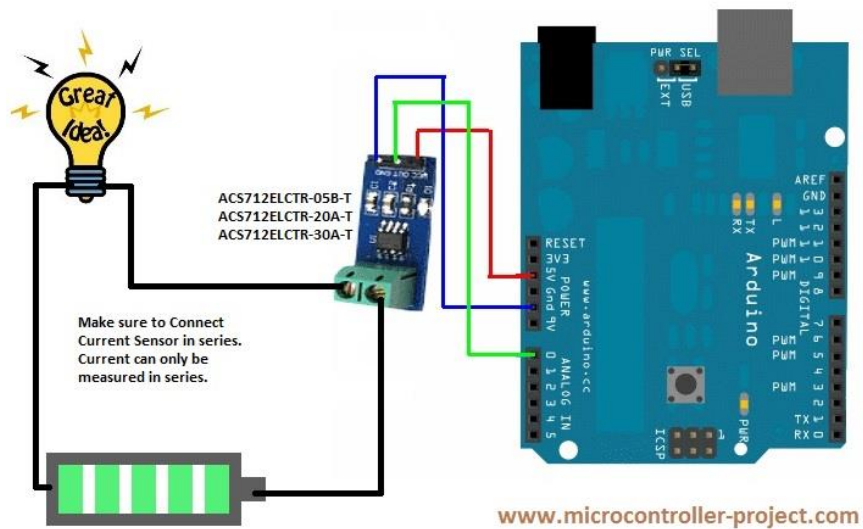


Figura 30: Conexión sensor Asc712
 Fuente: www.microcontroller-project.com

2.4.4.2.1 CÁLCULOS DEL SENSOR DE CORRIENTE

Regla de tres

$$1A = 0.185 \text{ (V/A)}$$



$$X = \text{DATO (V)}$$

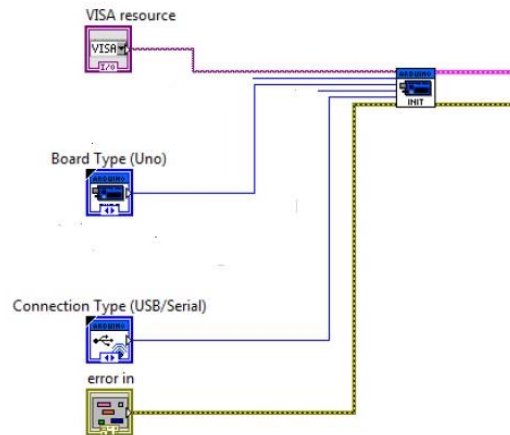
$$X = \frac{\text{DATOV} * 1}{0.185 \text{ V/A}}$$

$$X = \frac{\text{DATO}}{0.185} \text{ A}$$

2.4.5 ETAPA DE ADQUISICIÓN DE DATOS

En esta etapa logramos realizar la interfaz que nos permite visualizar los datos adquiridos de los sensores dicha adquisición se lo realiza con el Arduino Mega 2560 que realiza la función de una tarjeta de adquisición de datos, la información que proviene de los sensores son

Para iniciar la comunicación con la placa Arduino se coloca la subVI, esta subVI init.vi cuenta con 6 variables de entrada y 2 salidas ver **Figura 33** con las que definimos.



*Figura 33: Entradas y salidas SubVI Init.vi
Fuente: Autor*

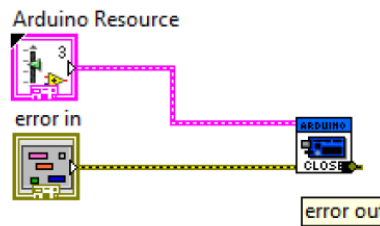
- **VISA resource:** Indicamos el puerto COM al cual tenemos conectado el Arduino en nuestro caso COM 5.
- **Boad type:** Elegimos que modelo de Arduino estamos utilizando en nuestro caso Arduino Mega.
- **Connection Type (USB serial):** Elegimos el tipo de conexión utilizado en la computadora para la conexión con Arduino.
- **Error in:** Es la entrada de errores.
- **Error out:** Salida de errores.

A su vez también es necesario terminar la comunicación con Arduino, y esto se realiza por medio del SubVI close.vi, que cuenta con dos estradas y una salida ver **Figura 34**.

Arduino resource: Salida de información obtenida de init.vi.

Error in: Error de entrada.

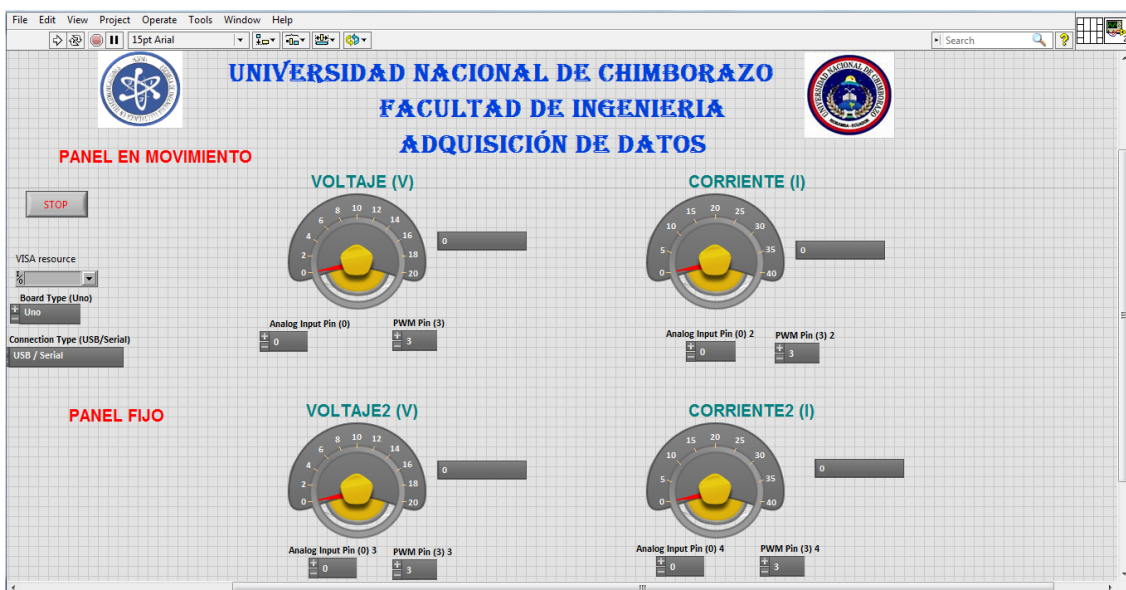
Error out: Error de salida.



*Figura 34: Entradas y salidas SubVI close.vi
Fuente: Autor*

2.4.5.2 DISEÑO DEL PROGRAMA PARA ADQUISICION DE DATOS

En la **Figura 35** podemos ver el panel frontal que se ha diseñado para la visualización y adquisición de los datos provenientes de los sensores que están conectados a los paneles solares, como podemos observar las primeras dos magnitudes corresponden al panel solar en movimiento y las siguientes dos corresponden al panel solar fijo.



*Figura 35: Panel frontal para la adquisición de datos en Labview
Fuente: Autor*

A continuación podemos observar el diagrama de bloques asociado a este vi debido a su tamaño vamos a dividir en dos grupos el primero que corresponde a la adquisición y el registro de los datos del panel solar en movimiento ver **Figura 36**, y la segunda parte corresponde al panel solar fijo ver **Figura 37**.

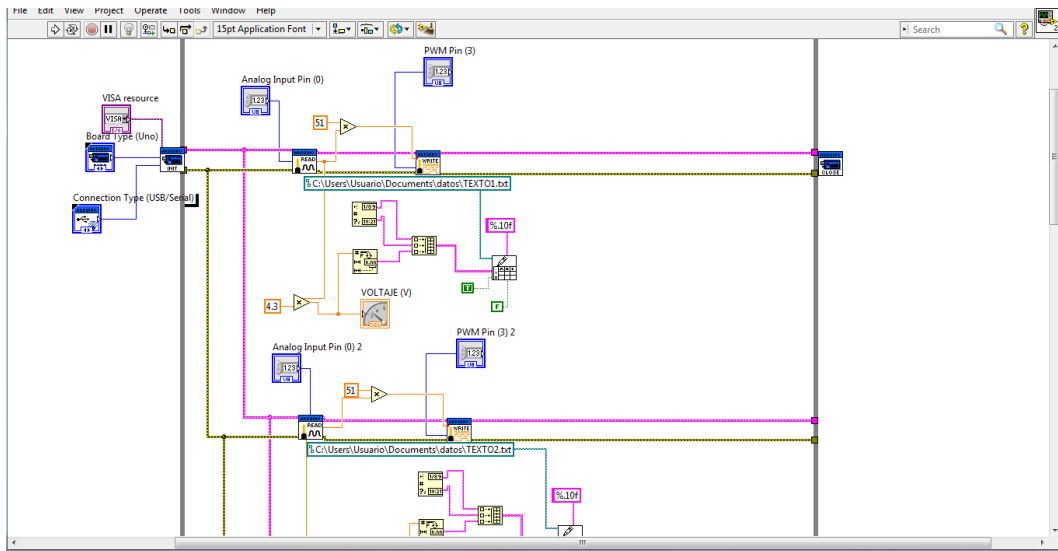


Figura 36: Diagrama de bloques adquisición de datos panel en movimiento
Fuente: Autor

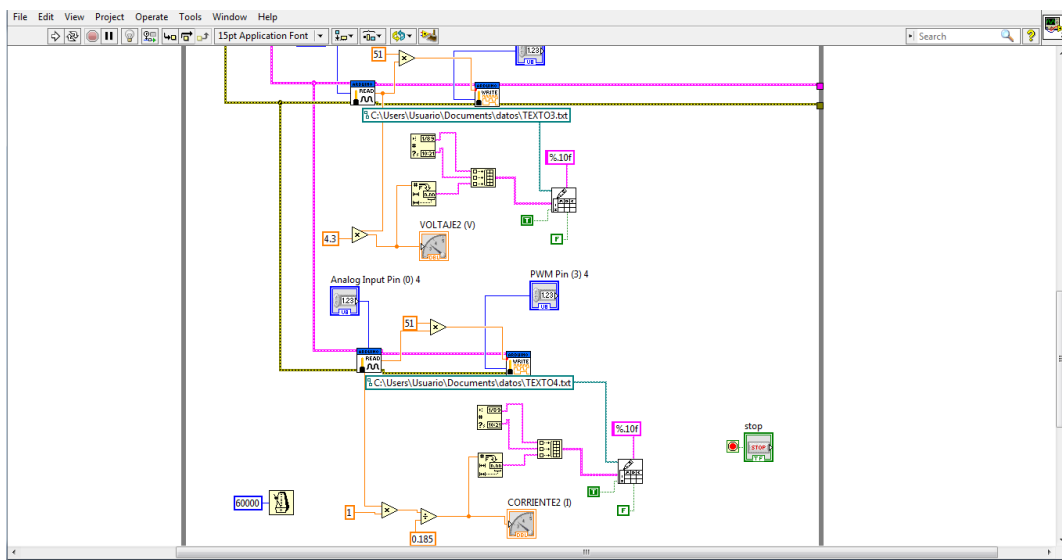


Figura 37: Diagrama de bloques adquisición de datos panel fijo
Fuente: Autor

2.5 PROCESAMIENTO Y ANÁLISIS

En el siguiente trabajo de investigación se obtiene la evaluación de la eficiencia del seguidor solar en comparación con el panel fotovoltaico fijo.

Con el protocolo de comunicación UART de MikroC podemos controlar los giros realizados por el motor.

Este estudio estuvo enfocado en diseñar y construir un sistema de seguimiento solar para obtener datos de voltaje y corriente y así poder realizar nuestra comparativa y nuestro modelamiento del sistema.

2.5.1 ANÁLISIS DEL PIC 16F877A

Podemos observar la etapa principal para el funcionamiento del seguidor solar. El pic es un microcontrolador 16F877A en cuyo interior se encuentra un programa almacenado para un fin determinado, poder controlar el tiempo y el giro de cada paso del motor. En el microcontrolador usamos la comunicación UART que permite transmitir los datos de giro del motor a una determinada velocidad.

2.5.2 ANÁLISIS PRÁCTICO DE LA COMUNICACIÓN UART

En la práctica observamos los datos transmitidos a través de la comunicación UART, una característica de la transmisión UART a tener en cuenta es la velocidad en la que se va a transmitir los datos en la se puede observar la Tx de datos que está determinada a una velocidad de 9600 baudios.

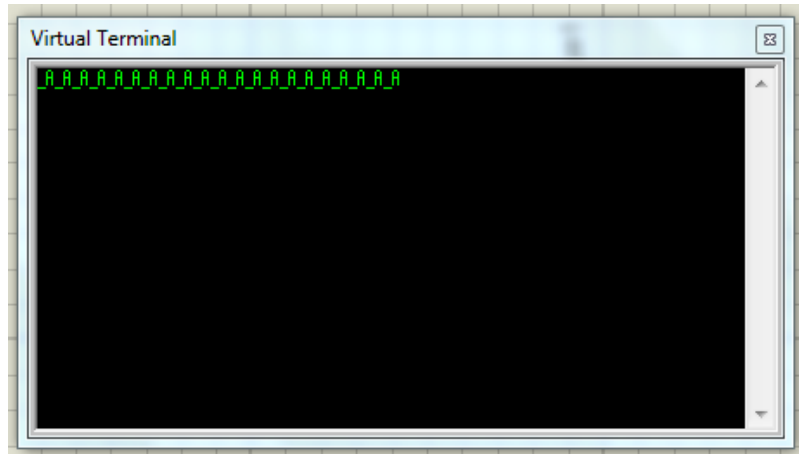


Figura 38: Comunicación UART
Fuente: Autor

2.5.3 TÉCNICAS DE PROCEDIMIENTO PARA EL ANÁLISIS

El análisis estadístico es cualitativo y cuantitativo. Para la parte cualitativa es comprobar la estabilidad y confiabilidad de la comunicación UART en la transmisión de datos para el giro del motor, para la parte cuantitativa se comprobará la eficiencia en el desarrollo de todas las prácticas de cada etapa de nuestro proyecto.

2.6 COMPROBACIÓN DE LA HIPÓTESIS

2.6.1 PLANTEAMIENTO DE LA HIPÓTESIS

Se plantea la hipótesis nula y la hipótesis afirmativa:

H₀ Con el modelamiento fotovoltaico y la comparativa de la eficiencia energética del panel fotovoltaico fijo con el panel fotovoltaico a controlar no generamos un ahorro económico y energético.

H₁ Con el modelamiento fotovoltaico y la comparativa de la eficiencia energética del panel fotovoltaico fijo con el panel fotovoltaico a controlar generamos un ahorro económico y energético.

2.6.2 ESTABLECIMIENTO DEL NIVEL DE SIGNIFICANCIA

El nivel de significancia es, $\alpha = 0,05$ % que representa el 5% y con un 95% de confianza.

2.6.3 ELECCIÓN DE LA PRUEBA DE HIPÓTESIS

Es una prueba de muestra relacionadas a un mismo grupo se aplica dos medidas en un tiempo distinto, es un estudio longitudinal, la variable fija que crea los grupos son dos medidas una antes y una media después, la variable aleatoria de comparación es la variable tiempo es una variable numérica, por lo tanto, se usa la prueba t student con muestras relacionadas para comprobar la hipótesis.

2.6.4 MUESTRA

La muestra realizada para la comprobación de la hipótesis es de 124.

2.6.5 MEDIA

Media

	N	Mínimo	Máximo	Media	Desv. típ.
Efic_Panel_Fijo	124	.11	67.31	23.8833	22.70796
Efic_Panel_Movil	124	1.17	98.23	37.0315	28.10057
N válido (según lista)	124				

*Tabla 4: Media Calculada
Elaborado: Autor*

Media calculada

$$M1 = (365.5) / 124 = 23.88 \quad \text{Media (Eficiencia panel fijo)}$$

$$M2 = (264) / 124 = 37.07 \quad \text{Media (Eficiencia panel móvil)}$$

2.6.6 PRUEBA DE NORMALIDAD

Para calcular la normalidad se usa el método de Shapiro-Wilk, se usa el software SPSS para encontrar el **p valor** (nivel de significación más pequeño posible que puede escogerse), para ver si la variable de tiempo se comporta con normalidad.

Pruebas de normalidad						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	Gl	Sig.	Estadístico	gl	Sig.
Efic_Panel_Fijo	.205	124	.000	.824	124	.000
Efic_Panel_Movil	.175	124	.000	.898	124	.000

*Tabla 5: Pruebas de normalidad en SPSS
Elaborado: Autor*

El valor de datos de p valor (nivel de significancia) son mayor que el nivel de α (error aceptado).

Los datos de tiempo provienen de una distribución normal, la variable tiempo se comporta normalmente.

2.6.7 PRUEBA T PARA MUESTRAS RELACIONADAS

Estadísticos de muestras relacionadas				
	Media	N	Desviación típ.	Error típ. de la media
Efic_Panel_Fijo	23.8833	124	22.70796	2.03924
Par 1 Efic_Panel_Movi 1	37.0315	124	28.10057	2.52351

*Tabla 6: Estadísticas de muestras relacionadas
Elaborado: Autor*

Correlaciones de muestras relacionadas

		N	Correlación	Sig.
Par 1	Efic_Panel_Fijo y Efic_Panel_Movil	124	.897	.000

*Tabla 7: Correlación de muestras relacionadas
Elaborado: Autor*

Prueba de muestras relacionadas

		Diferencias relacionadas				t	Gl	Sig. (bilateral)	
		Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza				
					para la diferencia				
Inferior	Superior								
Par 1	Efic_Panel_Fijo - Efic_Panel_Movil	- 13.14823	12.68099	1.13879	-15.40238	-10.89407	-11.546	123	.000

*Tabla 8: Prueba de muestras relacionadas
Elaborado: Autor*

El criterio a tomar en cuenta en la prueba de muestras relacionadas:

- Si la probabilidad obtenida de P-valor $\leq \alpha$ se rechaza H_0 y se acepta H_1
- Si la probabilidad obtenida de P-valor $\geq \alpha$ se rechaza H_1 y se acepta H_0 .

La variable tiempo antes y después de la prueba disminuye significativamente y tomando en cuenta el criterio de decisión de la prueba t, se llega a la conclusión que se rechaza H_0 y se acepta H_1 .

H1 Con el modelamiento fotovoltaico y la comparativa de la eficiencia energética del panel fotovoltaico fijo con el panel fotovoltaico a controlar generamos un ahorro económico y energético.

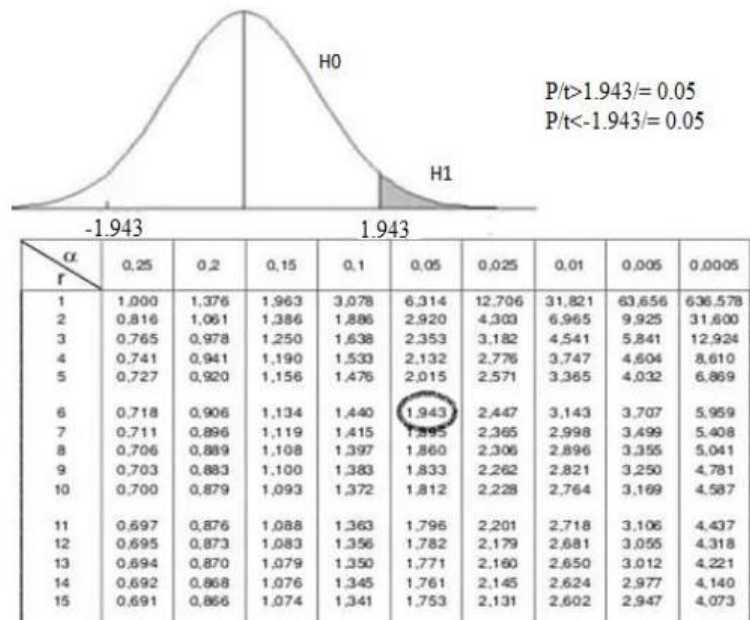


Figura 39: Aceptación de H1 y rechazo de H0
 Fuente: Autor

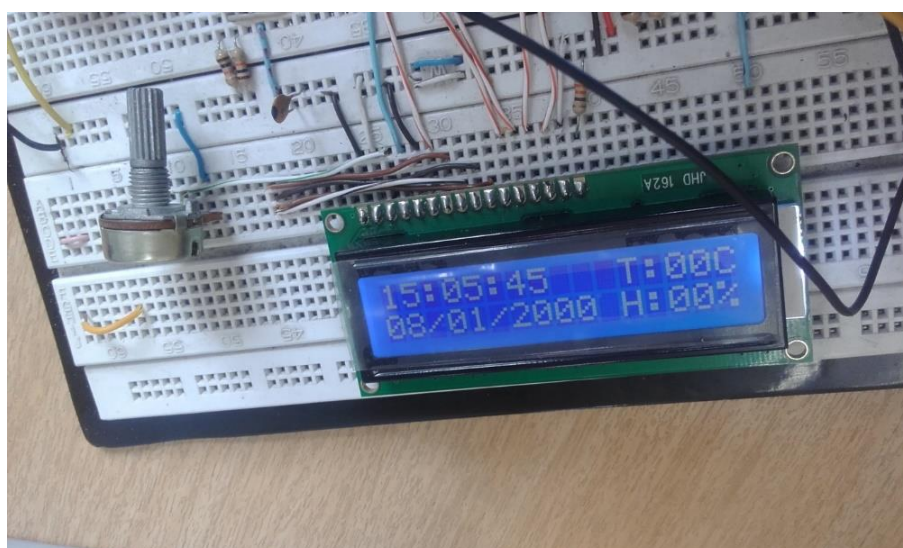
CAPITULO III

3 RESULTADOS

A continuación, se detalla las pruebas realizadas para la comunicación entre el PIC 16F877A y el motor.

3.1 PRUEBAS DEL RELOJ EN TIEMPO REAL

Se realiza el circuito del reloj en tiempo real y se procede a comprobar la funcionalidad del circuito, como se puede observar en la **Figura 40**.

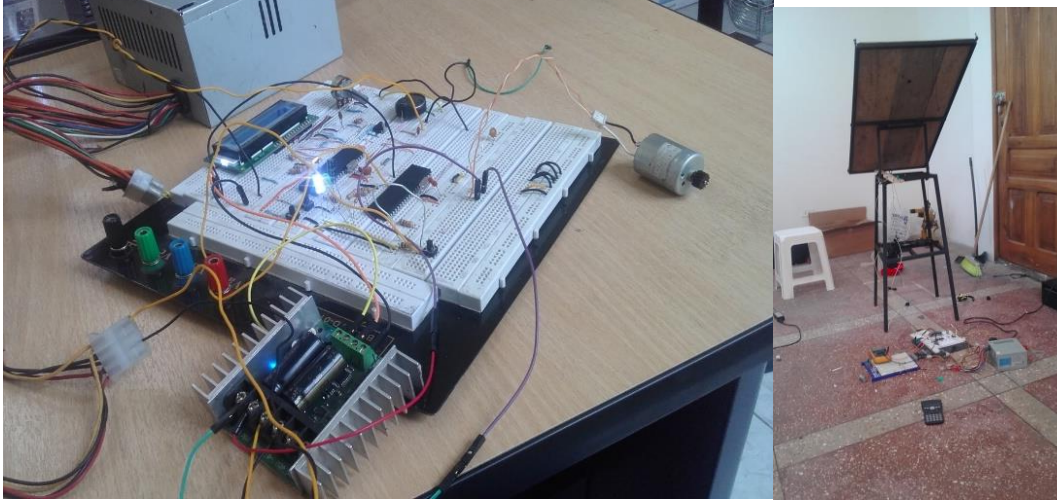


*Figura 40: Visualización del reloj en tiempo real en el LCD
Fuente: Autor*

El resultado de esta prueba fue la muestra de la fecha y la hora actual en el LCD.

3.2 PRUEBAS DEL MOTOR DC EN LA ESTRUCTURA

En esta parte logramos comprobar el funcionamiento del motor acoplado en la estructura mecánica como se muestra en la **Figura 41**.



*Figura 41: Pruebas del funcionamiento del motor dc
Fuente: Autor*

El resultado del giro del motor dc en la estructura fue exitoso.

3.2.1 PRUEBAS DE LA ADQUISICION DE DATOS

Realizamos y verificamos los datos adquiridos por la interfaz Labview-Arduino

En la **Figura 42** se puede observar los datos adquiridos por la interfaz Labview-Arduino.



*Figura 42: Sistema de adquisición de datos
Fuente: Autor*

Semana1

PANEL FOTOVOLTAICO FIJO						
SEMANA 1	HORA	MUESTRA	VOLTAJE (V)	CORRIENTE (A)	FF	EFICIENCIA
	8:02	1	6,1929	1,480	0,5922	6,03%
	8:14	2	6,6143	1,580	0,6070	7,05%
	8:26	3	4,6337	1,107	0,5247	2,99%
	9:00	4	4,8865	1,168	0,5373	3,41%
	9:15	5	2,5591	0,611	0,3854	0,67%
	9:40	6	4,4019	1,052	0,5125	2,64%
	10:15	7	4,6337	1,107	0,5247	2,99%
	10:30	8	6,1605	1,472	0,5910	5,95%
	10:50	9	8,7504	2,091	0,6669	13,56%
	11:15	10	8,6354	2,063	0,6642	13,15%
	11:30	11	6,9822	1,668	0,6190	8,01%
	11:45	12	4,6337	1,107	0,5247	2,99%
	12:15	13	4,5073	1,077	0,5181	2,79%
	12:30	14	4,8865	1,168	0,5373	3,41%
	12:50	15	6,4554	1,543	0,6015	6,66%
	13:10	16	8,6337	2,063	0,6641	13,14%
	13:30	17	8,1718	1,953	0,6527	11,57%
	13:45	18	10,2675	2,453	0,6985	19,55%
	14:00	19	10,3728	2,479	0,7005	20,01%
	14:20	20	7,8801	1,883	0,6451	10,63%
	14:45	21	10,1346	2,422	0,6960	18,98%
	15:00	22	7,0163	1,677	0,6200	8,10%
	15:15	23	6,9741	1,666	0,6187	7,99%
	15:20	24	8,6143	2,058	0,6637	13,08%
	16:10	25	8,6354	2,063	0,6642	13,15%
	16:30	26	7,5219	1,797	0,6351	9,54%
	16:50	27	8,0163	1,915	0,6487	11,07%
	17:00	28	5,9741	1,428	0,5840	5,53%
	17:30	29	4,2156	1,007	0,5022	2,37%
	17:45	30	3,5438	0,847	0,4608	1,54%
	18:00	31	1,5591	0,373	0,2873	0,19%

Tabla 9: Datos adquiridos primera semana panel fijo.
Fuente: Autor

PANEL FOTOVOLTAICO CON SEGUIDOR SOLAR						
SEMANA 1	HOR A	MUESTR A	VOLTAJE (V)	CORRIENTE (A)	FF	EFICIENCI A
	8:02	1	7,8056	1,865	0,6431	10,40%
	8:14	2	7,8898	1,885	0,6453	10,67%
	8:26	3	7,0373	1,682	0,6207	8,16%
	9:00	4	6,9952	1,671	0,6194	8,05%
	9:15	5	7,2481	1,732	0,6271	8,75%
	9:40	6	9,8607	2,356	0,6907	17,83%
	10:15	7	7,6694	1,833	0,6393	9,98%
	10:30	8	8,7441	2,089	0,6667	13,53%
	10:50	9	10,7457	2,568	0,7072	21,68%
	11:15	10	10,0391	2,399	0,6942	18,58%
	11:30	11	10,7554	2,570	0,7074	21,72%
	11:45	12	10,7116	2,560	0,7066	21,52%
	12:15	13	8,5122	2,034	0,6612	12,72%
	12:30	14	7,6062	1,817	0,6375	9,79%
	12:50	15	8,0276	1,918	0,6490	11,10%
	13:10	16	12,0811	2,887	0,7287	28,24%
	13:30	17	11,039	2,638	0,7122	23,04%
	13:45	18	11,5774	2,766	0,7210	25,66%
	14:00	19	15,7878	3,772	0,7734	51,18%
	14:20	20	10,0176	2,394	0,6938	18,48%
	14:45	21	13,7163	3,277	0,7506	37,49%
	15:00	22	8,0276	1,918	0,6490	11,10%
	15:15	23	9,0811	2,170	0,6744	14,77%
	15:20	24	13,3858	3,199	0,7465	35,51%
	16:10	25	13,8558	3,311	0,7523	38,35%
	16:30	26	9,4356	2,255	0,6820	16,12%
	16:50	27	12,7116	3,037	0,7376	31,64%
	17:00	28	7,7748	1,858	0,6422	10,31%
	17:30	29	8,0276	1,918	0,6490	11,10%
	17:45	30	6,71	1,603	0,6102	7,29%
	18:00	31	5,5414	1,324	0,5667	4,62%

*Tabla 10: Datos adquiridos primera semana seguidor solar
Fuente: Autor*

Semana 2

PANEL FOTOVOLTAICO FIJO						
SEMANA 2	HORA	MUESTRA	VOLTAJE (V)	CORRIENTE (A)	FF	EFICIENCIA
	8:02	1	5,7731	1,3795	0,5795	5,13%
	8:14	2	5,8574	1,39962	0,6167	5,62%
	8:26	3	6,9105	1,6513	0,6381	8,09%
	9:00	4	7,6273	1,82253	0,6271	9,69%
	9:15	5	7,2480	1,7319	0,6296	8,78%
	9:40	6	7,3323	1,75204	0,6393	9,13%
	10:15	7	7,6694	1,8326	0,6852	10,70%
	10:30	8	9,5868	2,29075	0,7605	18,56%
	10:50	9	14,5626	3,4797	0,7495	42,20%
	11:15	10	13,6254	3,25577	0,7814	38,51%
	11:30	11	16,6258	3,9727	0,7724	56,69%
	11:45	12	15,6874	3,74848	0,7888	51,54%
	12:15	13	17,4671	4,1737	0,7881	63,84%
	12:30	14	17,3827	4,15357	0,7874	63,17%
	12:50	15	17,3106	4,1363	0,7721	61,43%
	13:10	16	15,6604	3,74203	0,7842	51,06%
	13:30	17	16,9402	4,0478	0,7877	60,02%
	13:45	18	17,3448	4,14452	0,6835	54,59%
	14:00	19	9,5058	2,2714	0,7000	16,79%
	14:20	20	10,3453	2,47200	0,6327	17,98%
	14:45	21	7,4377	1,7772	0,6119	8,99%
	15:00	22	6,7634	1,61611	0,7342	8,92%
	15:15	23	12,4673	2,9790	0,7261	29,96%
	15:20	24	11,9076	2,84530	0,7446	28,03%
	16:10	25	13,2352	3,1625	0,6991	32,51%
	16:30	26	10,2987	2,46086	0,6823	19,21%
	16:50	27	9,4459	2,2571	0,6119	14,50%
	17:00	28	6,7634	1,61611	0,5761	7,00%
	17:30	29	5,7711	1,3790	0,4143	3,66%
	17:45	30	2,9076	0,69477	0,2425	0,54%
	18:00	31	1,1377	0,2719	0,3285	0,11%

*Tabla 11: Datos adquiridos segunda semana panel fijo
Fuente: Autor*

PANEL FOTOVOLTAICO CON SEGUIDOR SOLAR						
SEMANA 2	HORA	MUESTRA	VOLTAJE (V)	CORRIENTE (A)	FF	EFICIENCIA
	8:02	1	9,3972	2,2454	0,6812	15,97%
	8:14	2	9,0601	2,1649	0,6739	14,69%
	8:26	3	7,6694	1,8326	0,6393	9,98%
	9:00	4	8,5333	2,0390	0,6617	12,79%
	9:15	5	7,7343	1,8481	0,6411	10,18%
	9:40	6	11,48	2,7431	0,7194	25,17%
	10:15	7	14,4038	3,4418	0,7587	41,79%
	10:30	8	16,4248	3,9247	0,7795	55,83%
	10:50	9	17,467	4,1737	0,7888	63,89%
	11:15	10	18,4248	4,4026	0,7965	71,79%
	11:30	11	18,5612	4,4352	0,7976	72,96%
	11:45	12	16,4978	3,9421	0,7802	56,38%
	12:15	13	17,6566	4,2190	0,7904	65,42%
	12:30	14	19,8055	4,7325	0,8067	84,01%
	12:50	15	18,2887	4,3701	0,7955	70,64%
	13:10	16	17,1691	4,1025	0,7862	61,53%
	13:30	17	18,2875	4,3698	0,7955	70,63%
	13:45	18	19,9743	4,7728	0,8079	85,57%
	14:00	19	18,6907	4,4661	0,7986	74,07%
	14:20	20	18,9208	4,5211	0,8003	76,07%
	14:45	21	10,5546	2,5220	0,7038	20,82%
	15:00	22	9,99	2,3871	0,6933	18,37%
	15:15	23	15,5918	3,7256	0,7715	49,79%
	15:20	24	17,5546	4,1946	0,7895	64,60%
	16:10	25	18,8365	4,5010	0,7997	75,33%
	16:30	26	20,8837	4,9901	0,8139	94,24%
	16:50	27	15,7603	3,7659	0,7731	50,99%
	17:00	28	8,0698	1,9283	0,6501	11,24%
	17:30	29	6,7213	1,6060	0,6106	7,32%
	17:45	30	5,6046	1,3392	0,5693	4,75%
	18:00	31	4,3193	1,0321	0,5080	2,52%

Tabla 12: Datos adquiridos segunda semana seguidor solar
Fuente: Autor

Semana 3

PANEL FOTOVOLTAICO FIJO						
SEMANA 3	HORA	MUESTRA	VOLTAJE (V)	CORRIENTE (A)	FF	EFICIENCIA
	8:02	1	3,7099	0,8865	0,4717	1,72%
	8:14	2	6,8898	1,64631	0,6160	7,76%
	8:26	3	7,4377	1,7772	0,6327	9,29%
	9:00	4	8,1964	1,95852	0,6534	11,65%
	9:15	5	9,3875	2,2431	0,6810	15,93%
	9:40	6	11,1249	2,65828	0,7137	23,45%
	10:15	7	13,9904	3,3430	0,7539	39,18%
	10:30	8	15,2757	3,65011	0,7682	47,59%
	10:50	9	17,7600	4,2437	0,7912	66,26%
	11:15	10	16,9613	4,05288	0,7844	59,91%
	11:30	11	17,8884	4,2744	0,7923	67,31%
	11:45	12	14,6015	3,48901	0,7610	43,07%
	12:15	13	8,8072	2,1045	0,6682	13,76%
	12:30	14	16,1817	3,86659	0,7772	54,03%
	12:50	15	16,5934	3,9650	0,7811	57,10%
	13:10	16	17,6988	4,22910	0,7907	65,76%
	13:30	17	17,5934	4,2039	0,7898	64,91%
	13:45	18	9,7334	2,32578	0,6882	17,31%
	14:00	19	14,6647	3,5041	0,7617	43,49%
	14:20	20	16,9192	4,04282	0,7840	59,59%
	14:45	21	16,4767	3,9371	0,7800	56,22%
	15:00	22	17,2536	4,12272	0,7870	62,20%
	15:15	23	10,1768	2,4317	0,6968	19,16%
	15:20	24	6,2464	1,49257	0,5941	6,15%
	16:10	25	7,8363	1,8725	0,6439	10,50%
	16:30	26	8,2803	1,97857	0,6555	11,93%
	16:50	27	6,3842	1,5255	0,5991	6,48%
	17:00	28	5,3307	1,27376	0,5577	4,21%
	17:30	29	4,9093	1,1731	0,5384	3,44%
	17:45	30	4,4247	1,05728	0,5137	2,67%
	18:00	31	3,0341	0,7250	0,4242	1,04%

Tabla 13: Datos adquiridos tercera semana panel fijo

Fuente: Autor

PANEL FOTOVOLTAICO CON SEGUIDOR SOLAR						
SEMANA 3	HORA	MUESTRA	VOLTAJE (V)	CORRIENTE (A)	FF	EFICIENCIA
	8:02	1	4,551	1,087	0,5204	2,86%
	8:14	2	8,216	1,963	0,6538	11,72%
	8:26	3	8,501	2,031	0,6609	12,68%
	9:00	4	9,175	2,192	0,6765	15,12%
	9:15	5	11,217	2,680	0,7152	23,89%
	9:40	6	13,443	3,212	0,7472	35,85%
	10:15	7	16,751	4,003	0,7825	58,29%
	10:30	8	16,927	4,045	0,7841	59,65%
	10:50	9	18,789	4,490	0,7993	74,92%
	11:15	10	19,758	4,721	0,8064	83,57%
	11:30	11	19,371	4,629	0,8036	80,06%
	11:45	12	16,203	3,872	0,7774	54,19%
	12:15	13	13,295	3,177	0,7454	34,98%
	12:30	14	17,354	4,147	0,7878	62,99%
	12:50	15	17,414	4,161	0,7883	63,47%
	13:10	16	19,083	4,560	0,8015	77,49%
	13:30	17	18,757	4,482	0,7991	74,64%
	13:45	18	14,686	3,509	0,7619	43,63%
	14:00	19	16,414	3,922	0,7794	55,75%
	14:20	20	19,376	4,630	0,8037	80,11%
	14:45	21	20,519	4,903	0,8115	90,71%
	15:00	22	18,927	4,523	0,8004	76,13%
	15:15	23	14,870	3,553	0,7639	44,84%
	15:20	24	8,112	1,938	0,6512	11,38%
	16:10	25	9,376	2,240	0,6808	15,89%
	16:30	26	10,658	2,547	0,7056	21,28%
	16:50	27	7,459	1,782	0,6333	9,35%
	17:00	28	6,090	1,455	0,5884	5,79%
	17:30	29	5,671	1,355	0,5721	4,88%
	17:45	30	4,972	1,188	0,5414	3,55%
	18:00	31	4,235	1,012	0,5033	2,40%

Tabla 14: Datos adquiridos tercera semana seguidor solar

Fuente: Autor

Semana 4

PANEL FOTOVOLTAICO FIJO						
SEMANA 4	HORA	MUESTRA	VOLTAJE (V)	CORRIENTE (A)	FF	EFICIENCIA
	8:02	1	4,8882	1,1680	0,5373	3,41%
	8:14	2	4,9303	1,17809	0,5394	3,48%
	8:26	3	5,7731	1,3795	0,5762	5,10%
	9:00	4	6,2573	1,49517	0,5945	6,18%
	9:15	5	5,0371	1,2036	0,5444	3,67%
	9:40	6	6,6169	1,58110	0,6071	7,06%
	10:15	7	11,0199	2,6332	0,7119	22,95%
	10:30	8	13,3756	3,19608	0,7464	35,45%
	10:50	9	16,8981	4,0378	0,7838	59,42%
	11:15	10	16,6931	3,98879	0,7820	57,85%
	11:30	11	17,1299	4,0932	0,7859	61,22%
	11:45	12	17,7409	4,23916	0,7911	66,10%
	12:15	13	17,6145	4,2090	0,7900	65,08%
	12:30	14	16,8349	4,02268	0,7833	58,94%
	12:50	15	10,0082	2,3914	0,6936	18,45%
	13:10	16	13,7931	3,29584	0,7516	37,96%
	13:30	17	12,6039	3,0117	0,7361	31,05%
	13:45	18	13,0373	3,11524	0,7420	33,48%
	14:00	19	15,5544	3,7167	0,7711	49,53%
	14:20	20	16,0390	3,83250	0,7759	52,99%
	14:45	21	17,6355	4,2140	0,7902	65,25%
	15:00	22	16,9824	4,05792	0,7846	60,08%
	15:15	23	17,3406	4,1435	0,7877	62,89%
	15:20	24	15,5496	3,71556	0,7710	49,50%
	16:10	25	5,9628	1,4248	0,5836	5,51%
	16:30	26	9,3345	2,23047	0,6799	15,73%
	16:50	27	7,5009	1,7923	0,6345	9,48%
	17:00	28	4,6986	1,12272	0,5280	3,09%
	17:30	29	3,6872	0,8811	0,4702	1,70%
	17:45	30	2,8233	0,67462	0,4076	0,86%
	18:00	31	2,5916	0,6193	0,3882	0,69%

Tabla 15: Datos adquiridos cuarta semana panel fijo

Fuente: Autor

PANEL FOTOVOLTAICO CON SEGUIDOR SOLAR						
SEMANA 4	HORA	MUESTRA	VOLTAJE (V)	CORRIENTE (A)	FF	EFICIENCIA
	8:02	1	5,4992	1,3140	0,5649	4,54%
	8:14	2	6,5768	1,5715	0,6057	6,96%
	8:26	3	7,0584	1,6866	0,6213	8,22%
	9:00	4	9,6159	2,2977	0,6858	16,84%
	9:15	5	10,2627	2,4523	0,6984	19,53%
	9:40	6	13,1298	3,1373	0,7432	34,02%
	10:15	7	16,4556	3,9320	0,7798	56,06%
	10:30	8	15,9499	3,8112	0,7750	52,35%
	10:50	9	17,7085	4,2314	0,7908	65,84%
	11:15	10	17,6242	4,2113	0,7901	65,16%
	11:30	11	21,2877	5,0867	0,8164	98,23%
	11:45	12	19,3713	4,6287	0,8036	80,06%
	12:15	13	18,9531	4,5288	0,8006	76,35%
	12:30	14	18,4563	4,4101	0,7968	72,06%
	12:50	15	14,6436	3,4991	0,7614	43,35%
	13:10	16	15,3119	3,6588	0,7686	47,84%
	13:30	17	14,7949	3,5352	0,7631	44,35%
	13:45	18	15,6500	3,7396	0,7720	50,20%
	14:00	19	17,4256	4,1638	0,7884	63,56%
	14:20	20	17,4678	4,1739	0,7888	63,90%
	14:45	21	18,5188	4,4250	0,7973	72,59%
	15:00	22	17,7814	4,2488	0,7914	66,43%
	15:15	23	19,4036	4,6365	0,8039	80,35%
	15:20	24	17,6045	4,2066	0,7899	65,00%
	16:10	25	9,2354	2,2068	0,6778	15,35%
	16:30	26	10,4253	2,4911	0,7015	20,24%
	16:50	27	8,8128	2,1058	0,6683	13,78%
	17:00	28	6,7634	1,6161	0,6119	7,43%
	17:30	29	5,2675	1,2587	0,5549	4,09%
	17:45	30	4,0454	0,9666	0,4923	2,14%
	18:00	31	3,1815	0,7602	0,4353	1,17%

Tabla 16: Datos adquiridos cuarta semana seguidor solar

Fuente: Autor

EFICIENCIA PRIMERA SEMANA

PANEL FIJO		SEGUIDOR SOLAR	
HORA	EFICIENCIA	HORA	EFICIENCIA
9:15	0.67%	9:15	8.75%
18:00	0.19%	18:00	4.62%
15:00	8.10%	15:00	26.55%
15:15	7.99%	15:15	14.77%
13:45	19.55%	13:45	42.92%
14:00	20.01%	14:00	51.18%

*Tabla 17: Comparación eficiencia primera semana
Fuente: Autor*

EFICIENCIA SEGUNDA SEMANA

PANEL FIJO		SEGUIDOR SOLAR	
HORA	EFICIENCIA	HORA	EFICIENCIA
17:45	0.54%	17:45	4.75%
18:00	0.23%	18:00	2.52%
11:15	16.79%	11:15	50.49%
11:30	22.68%	11:30	59.60%
14:00	73.51%	14:00	89.81%
14:30	30.26%	14:30	95.98%

*Tabla 17: Comparación eficiencia segunda semana
Fuente: Autor*

EFICIENCIA TERCERA SEMANA

PANEL FIJO		SEGUIDOR SOLAR	
HORA	EFICIENCIA	HORA	EFICIENCIA
8:02	1.72%	8:02	2.86%
18:00	1.04%	18:00	2.40%
9:15	20.07%	9:15	11.72%
9:40	23.45%	9:40	16.15%
10:50	66.26%	10:50	74.92%
11:30	67.31%	11:30	80.06%

*Tabla 18: Comparación eficiencia tercera semana
Fuente: Autor*

EFICIENCIA CUARTA SEMANA

PANEL FIJO		SEGUIDOR SOLAR	
HORA	EFICIENCIA	HORA	EFICIENCIA
17:45	0.86%	17:45	2.14%
18:00	0.69%	18:00	1.17%
14:20	14.61%	14:20	36.00%
16:30	15.73%	16:30	20.24%
11:45	66.10%	11:45	80.60%
12:15	65.08%	12:15	76.35%

*Tabla 19: Comparación eficiencia cuarta semana
Fuente: Autor*

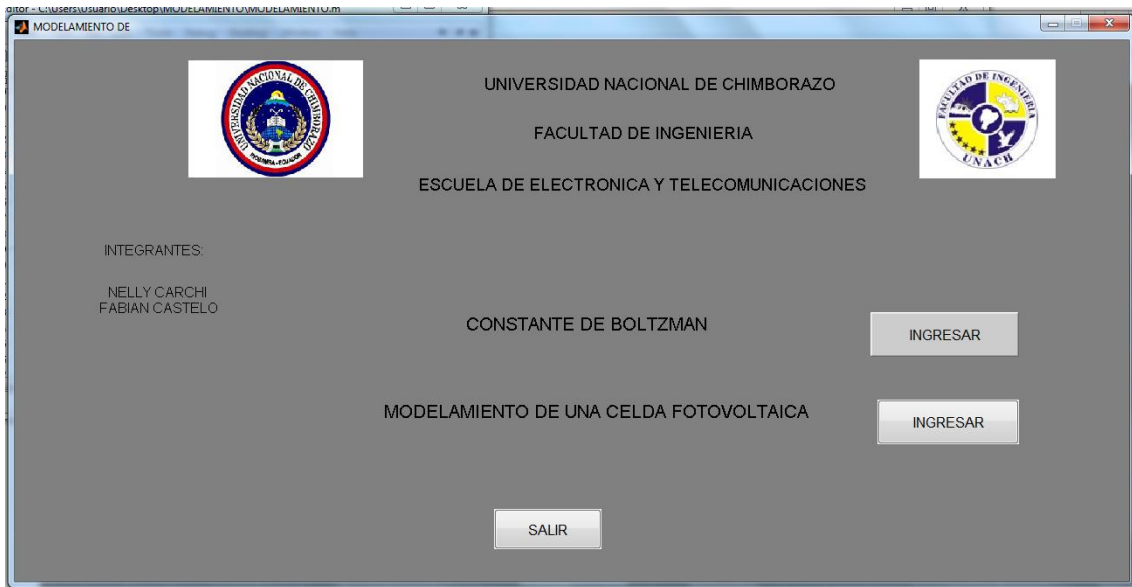


Figura 43: Modelamiento
Fuente: Autor

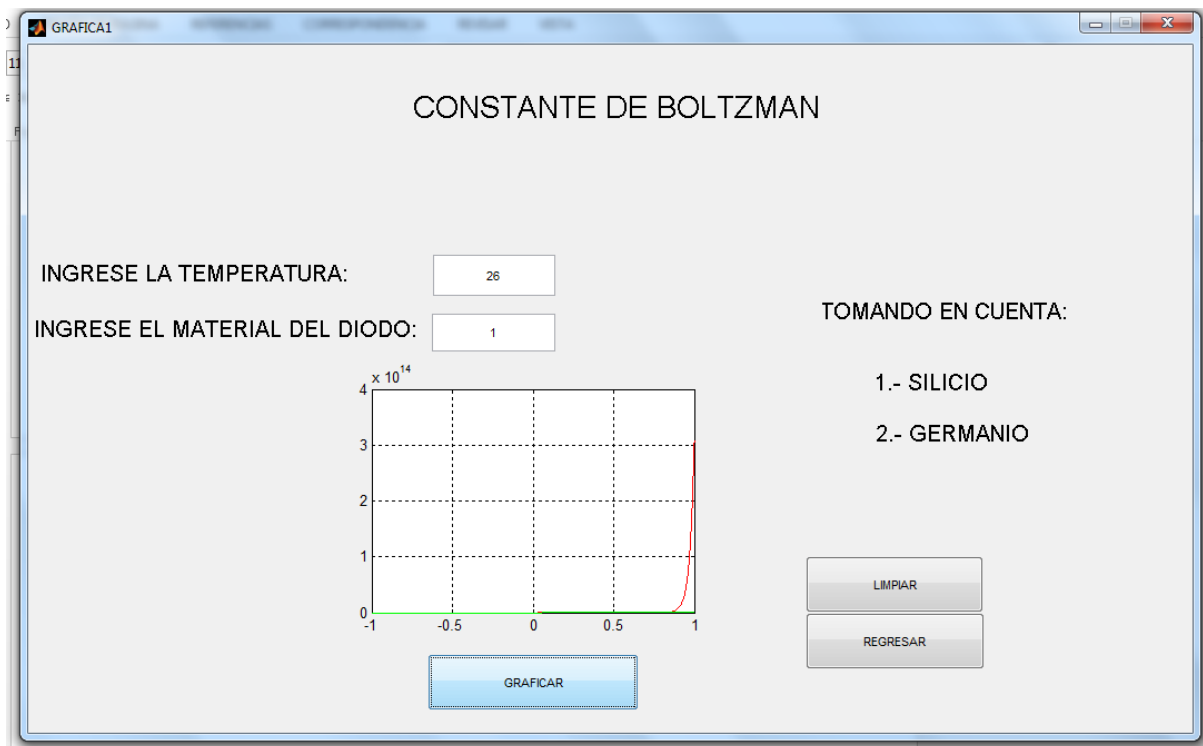
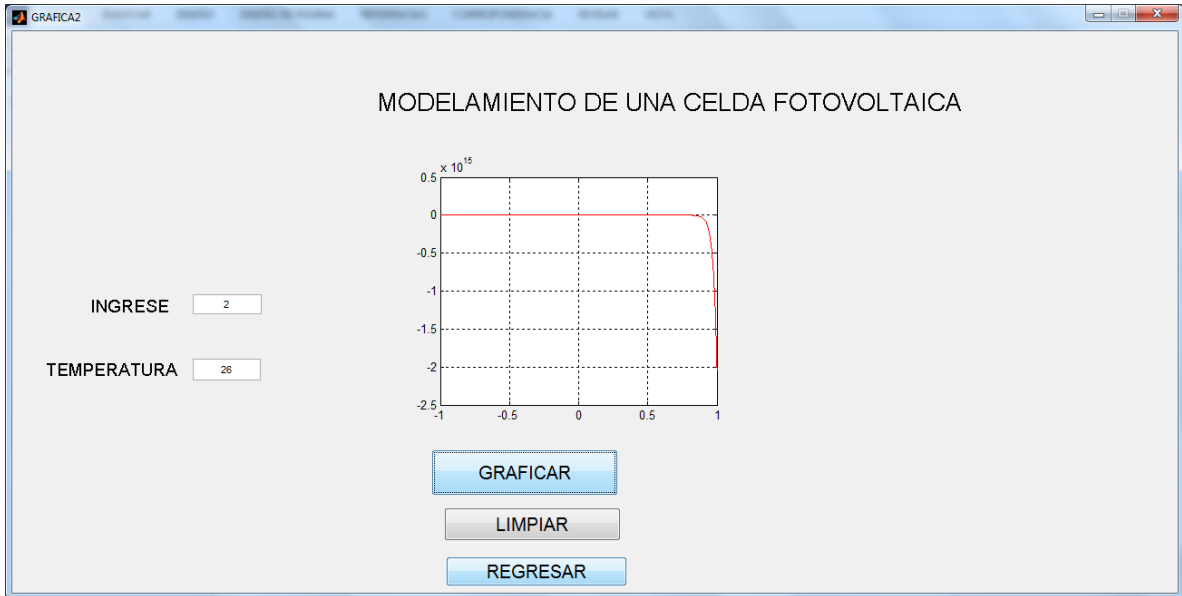


Figura 34: Modelamiento de una celda fotovoltaica

Figura 44: Modelamiento de la constante de Boltzman
Fuente: Autor



*Figura 45: Modelamiento de una celda fotovoltaica
Fuente: Autor*

Los resultados de estas pruebas fueron exitosos la programación completa del modelamiento en Matlab se puede observar en el **Anexo C**.

3.3 ANÁLISIS FINANCIERO

	Descripción	Cantidad	Valor Unitario Usd.	Valor Total Usd.
Hardware	Sensores	4	45,00	180,00
	Micro controlador	2	15,00	30,00
	Protoboard	1	35,00	35,00
	Motor DC	1	80,00	80,00
	Puente H	1	200,00	200,00
	Arduino	1	50,00	50,00
	Multímetro	1	25,00	25,00
	Estructura	1	300,00	300,00
	Integrado	1	10,00	10,00
Software	Mikro C	1	0,00	0,00
	Labview	1	0,00	0,00
	Matlab	1	0,00	0,00
Proyecto	Transporte	1	30,00	30,00
Varios	Cables, poleas	1	100,00	100,00
Subtotal				1030,00
TOTAL				1040,00

Tabla 20: Análisis financiero

Fuente: Autor

CAPITULO IV

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- La energía solar permite obtener un ahorro energético y financiero, a diferencia de otros sistemas de generación de energía además ayuda a la conservación del medio ambiente ya que es una energía pura e inagotable.
- Con la comparativa se demostró que en el seguidor solar se obtiene la máxima captación de energía durante todo el día y podemos asegurar que el sistema de seguidor solar es muy fiable ya que mejoran el rendimiento energético y económico ya que su mantenimiento es factible y económico.
- Se ha conseguido implementar un sistema mecánico fácil y sencillo que cumple con todos nuestros parámetros y requerimientos, para controlar el movimiento del panel se realizó la programación de un reloj y un contador el cual permite que el motor gire cada determinado tiempo.
- Se diseñó e implemento un sistema para la adquisición y el registro de datos, ya que no pudimos tener acceso a la tarjeta SWDM-US-10 con la que cuenta la institución es por ello que nos vimos en la necesidad de adquirir los datos por medio de sensores tanto de voltaje y de corriente, los sensores están conectados a una placa Arduino la misma que trabaja como una tarjeta de adquisición de datos mientras que para monitorear y visualizar las magnitudes de los sensores utilizamos una interfaz en Labview.

4.2 RECOMENDACIONES

- Es necesario realizar el cálculo matemático de la eficiencia con la Ecuación 7 descrita anteriormente ya que con esta ecuación los resultados obtenidos son más precisos.
- Las estructuras que se utilizan para colocar a los paneles solares deben estar correctamente ubicadas para que ningún objeto pueda obstaculizar los rayos del sol.
- Utilizar de mejor manera los equipos con los que cuenta la institución ya que son recursos importantes y necesarios que no se encuentran funcionando.
- Debemos asegurar bien el panel del seguidor solar a la estructura para que no exista ningún riesgo de que el panel se caiga mientras el motor gira gradualmente.

CAPITULO V

5 PROPUESTA

5.1 TÍTULO DE LA PROPUESTA

Modelamiento de un sistema fotovoltaico para mejorar la eficiencia y optimización de la energía en el bloque b de ingeniería.

5.2 INTRODUCCIÓN

Es necesario desarrollar un sistema de energía renovable para generar un ahorro económico y energético utilizando energía solar, obtenida a partir del aprovechamiento de la radiación procedente del Sol, en la actualidad, el calor y la luz del Sol puede aprovecharse por medio de diversos captadores como celdas fotovoltaicas.

Las celdas fotovoltaicas son arreglos de materiales semiconductores con un potencial de flujo de electrones controlado que al ser energizados por fotones de luz de energía solar generan una corriente eléctrica. Esta tecnología puede organizarse en módulos o sistemas para alimentar un circuito eléctrico.

5.3 DISCUSIÓN

La investigación se enfoca en el diseño e implementación de un seguidor solar usando nuestra propia tecnología. El costo aproximado asciende entre 1000 dólares.

Durante la evaluación de dicho seguidor solar se pudo constatar que el sistema de control ayudo mucho para evitar la inestabilidad de la estructura ya que la fuerza del viento pueden influir de manera negativa para nuestro sistema, con el sistema de poleas logramos un control de giro del motor.

Al ser un modelo experimental, el programa de usuario que corre dentro del microcontrolador puede ser modificado para mayor funcionalidad, como empezar el seguimiento de sol desde tempranas horas o de aumentar o disminuir el giro del motor.

Es de vital importancia seguir con el desarrollo de esta tecnología de captación, acumulación y distribución de la energía solar, para brindar a la Universidad un ahorro energético y económico. Es por ello que se inicia este tipo de trabajo de investigación como aporte a esta nueva alternativa de captación de energía, usando un seguidor solar diseñado y construido por nosotros mismos.

CAPITULO VI

6 OBJETIVOS

6.1 OBJETIVO GENERAL

- Modelar un sistema fotovoltaico para mejorar la eficiencia y optimización de la energía.

6.2 OBJETIVOS ESPECÍFICOS

- Desarrollar el modelamiento fotovoltaico en un software.
- Diseñar y construir el seguidor solar para para realizar la comparativa con el panel solar fijo.
- Implementar un sistema de adquisición y registro de datos en tiempo real con la ayuda del software Labview.
- Analizar los resultados adquiridos del panel fotovoltaico fijo con el panel fotovoltaico a controlar.

6.3 FUNDAMENTACIÓN CIENTÍFICO – TÉCNICA

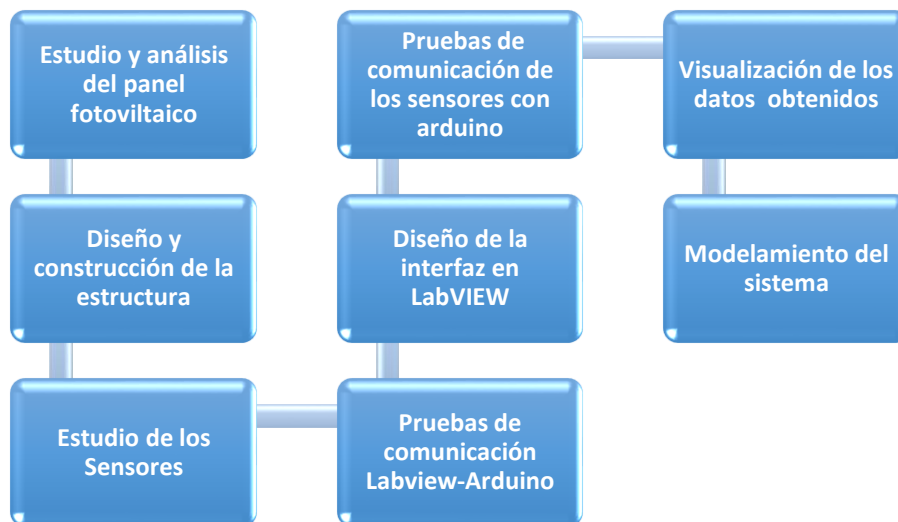
La energía solar fotovoltaica constituye una importancia alternativa en la producción de electricidad, no solo en zonas aisladas sino en centros poblados conectados a redes de distribución. El diseñador es el encargado de adaptar cada una de las etapas del sistema fotovoltaico para obtener un correcto funcionamiento a la hora de llevarlo a la práctica.

6.4 DESCRIPCIÓN DE LA PROPUESTA

El modelamiento de la celda fotovoltaica y la comparativa de los sistemas van a permitir la posibilidad de poder optar por mejores tecnologías, mejores software que ayudaran a obtener excelentes resultados para satisfacer todas las necesidades a la hora de producir electricidad no solo en zonas aisladas sino en centros poblados conectados a redes de distribución.

6.5 DISEÑO ORGANIZACIONAL

Se establecerá la estructura orgánica y funcional de la unidad administrativa que ejecutará la propuesta como se muestra en la **Figura 46**.



*Figura 46: Diseño organizacional del proyecto
Fuente: Autor*

CAPITULO VII

7 BIBLIOGRAFÍA

- [1] S. f. Manuales sobre energia renovable, «En línea,» Septiembre 2002. [En línea]. Available: <http://www.bunca.org/publicaciones/FOTOVOLT.pdf>.
- [2] Instituto de energia solar, «Efecto fotovoltaico,» en línea. [En línea]. Available: <http://bibing.us.es/proyectos/abreproy/11375/fichero/MEMORIA%252FEstudio+filtro+iir.pdf>.
- [3] M. T. Mohamed F, «Deprtament of Electrical and computer Engineering,» 2004. [En línea]. Available: <http://innovexpo.itee.uq.edu.au/2003/projects/s4002169/thesis.pdf>.
- [4] S. B. Christiana Honsberg, «PV EDUCATION.ORG,» en línea. [En línea]. Available: <http://pveducation.org/pvcdrom/solar-cell-operation/fill-factor>.
- [5] B. Hans y y. F. G. Gottschalg Ralf, «Modelling I-V Characteristics of Cste,,» 2013. [En línea]. Available: https://elektrochik.hs-magdeburg.de/Mitarbeite/bever/forsh/downloads/4AV_3_20.pdf.
- [6] C. Canto Q., «Microcontroladores,» 2013. [En línea]. Available: http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/2_INTROD.PDF.
- [7] Microchip, «Microchip,» 2001-2013. [En línea]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>.
- [8] Maxim Integrated Products, «datasheets real time clock ds1307,» 2015. [En línea]. Available: <http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>.
- [9] J. R. C. Mendoza, Diseño y simulación de sistemas microcontrolados en lenguaje C, Colombia : 1ra Edición, 2011.
- [10] D. Engineering, «Sabertooth 2x25 v2.0,» 2014. [En línea]. Available: <https://www.dimensionengineering.com/datasheets/Sabertooth2x25.pdf>.
- [11] A. MicroSystems, «datasheet acs712,» 2016. [En línea]. Available: <http://pdf1.alldatasheet.com/datasheet-pdf/view/168326/ALLEGRO/ACS712.html>.
- [12] T. d. r. y. e. e. d. c. d. mecatrónica, Libro básico de arduino, Bogotá: Primera Edición, 2012.
- [13] J. . R. Clavijo Mendosa, Diseño y Simulaciòn de Sistemas Microcontrolados en Lenguaje C, Colombia , 2011.

- [14] R. W. Larsen, Labview for Engineers, New Jersey: Pearson, 2015.
- [15] I. The Math Works, «Matlab, guia de usuario,» México, 1999.
- [16] S. B. Christiana Honsberg, «PV EDUCATION.ORG,» en línea. [En línea]. Available:
<http://pveducation.org/pvcdrom/solar-cell-operation/efficiency>.

8 ANEXOS

ANEXO A

GLOSARIO DE TÉRMINOS

SEGUIDOR SOLAR	Dispositivo mecánico
SENSOR	Dispositivo que capta magnitudes físicas
TX	Transmisión
ADC	Convertor analógico digital
UART	Comunicación de datos seriales
ADQUISICIÓN DE DATOS	Toma de muestras de mundo real

CÓDIGO EN MIKROC DE UN RELOJ EN TIEMPO REAL

//PIC16F877A & DS1307 Real Time Clock & DHT11 Sensor

```
unsigned int temp_res,temp_res2, dato,dato1;
```

```
    unsigned temp_res1;
```

```
    char uart_rd,DAT1=0;
```

```
    char txt[15];
```

```
    char error;
```

```
    unsigned short datos, datos1;
```

```
// LCD module connections
```

```
sbit LCD_RS at RD0_bit;
```

```
sbit LCD_EN at RD1_bit;
```

```

sbit LCD_D4 at RD2_bit;
sbit LCD_D5 at RD3_bit;
sbit LCD_D6 at RD4_bit;
sbit LCD_D7 at RD5_bit;

sbit LCD_RS_Direction at TRISD0_bit;
sbit LCD_EN_Direction at TRISD1_bit;
sbit LCD_D4_Direction at TRISD2_bit;
sbit LCD_D5_Direction at TRISD3_bit;
sbit LCD_D6_Direction at TRISD4_bit;
sbit LCD_D7_Direction at TRISD5_bit;

// End LCD module connections

unsigned char *text, second, second10, minute, minute10,
             hour, hour10, date, date10, month, month10,
             year, year10, day, i = 0, conta, conta10, y10, y, y20, contas, contas10;
unsigned char a = 0, b = 0, j = 0, n, temp, humidity;

void Interrupt(){ // FUNCION
if (TMR1IF_bit){ // si los timers se habilitan para interrupciones
    TMR1IF_bit = 0; //ponga un 0 tmr1 neutralice los timer
    TMR1H      = 0x6D; // 01101101 valor 109 en decimal registro de 8 bits
    TMR1L      = 0x84; // 10000100 132 en desimal registro de 8 bits
    n++;
    conta++; // n incrementa
}
}

```

```

void display(){
    //Split data into tow parts
    second10 = (second & 0x70) >> 4; //112 01110000
    second = second & 0x0F;
    minute10 = (minute & 0x70) >> 4;
    minute = minute & 0x0F;
    hour10 = (hour & 0x30) >> 4; //48 00110000
    hour = hour & 0x0F;
    date10 = (date & 0x30) >> 4;
    date = date & 0x0F;
    month10 = (month & 0x10) >> 4; //16 00010000
    month = month & 0x0F;
    year10 = (year & 0xF0) >> 4;
    year = year & 0x0F;
    //Display Time
    Lcd_Chr(1, 8, second + 48); // imprime n la fila 1 columna 8
    Lcd_Chr(1, 7, second10 + 48);
    Lcd_Chr(1, 5, minute + 48);
    Lcd_Chr(1, 4, minute10 + 48);
    Lcd_Chr(1, 2, hour + 48);
    Lcd_Chr(1, 1, hour10 + 48);
    //Display calendar
    Lcd_Chr(2, 2, date + 48);
    Lcd_Chr(2, 1, date10 + 48);
    Lcd_Chr(2, 5, month + 48);
    Lcd_Chr(2, 4, month10 + 48);

```

```

    Lcd_Chr(2, 10, year + 48);
    Lcd_Chr(2, 9, year10 + 48);
    //Display temperature and humidity
    // Lcd_Chr(1, 14, (temp/10) + 48);
    //Lcd_Chr(1, 15, (temp % 10) + 48);
    //Lcd_Chr(2, 14, (humidity/10) + 48);
    //Lcd_Chr(2, 15, (humidity % 10) + 48);
}

void write_value(char address, char data_){
    I2C1_Start();
    I2C1_Wr(0xD0); // 11010000
    I2C1_Wr(address);
    I2C1_Wr(data_);
    I2C1_Stop(); }

void main() {
    Soft_UART_Init(&PORTC,0,1,9600, 0);
    error = Soft_UART_Init(&PORTC, 0, 1, 9600, 0); // Initialize Soft UART at 14400 bps

    trisd.f7=1;
    portD.f7=0;
    trisd.f6=1;
    portD.f6=0;
    trisb.f4=1;
    portb.f4=0;
    trisb.f5=1;
    portb.f5=0;

```



```

TRISC.F1 = 1;

PORTC.F1=0;

UART1_Init(9600);          // Initialize UART module at 9600 bps

Delay_ms(100); //TIEMPO DE LA TRANSMISION A LA PC

//T2CON= 0b00000110;      //TIMR2 ACTIVADO CON UN PRESCALER DE 16

//CCPR2L= 0b00000000;    // R S GUARDAN LOS 8 BITS

// CCP2CON= 0b00000000;   // R S G LOS 2 BITS MAS SIGNIFICATIVOS

//PWM1_Start();

//UART1_Init(9600);      // Initialize UART module at 9600 bps

// Delay_ms(100); //TIEMPO DE LA TRANSMISION A LA PC

//Initiate Timer1 Interrupt

//T1CON    = 0x31;    // HABI TMR1 1:8 PRESCALER

//TMR1IF_bit  = 0;   // INTERRUPCION DEL TIMER1 DESA

//TMR1H    = 0x6D;   // 01101101

//TMR1L    = 0x84;   // 10000100

//TMR1IE_bit  = 1;   // BIT HABILITADI PARA ITER POR DESBORDE

//INTCON    = 0xC0;  //HABIL INTE GLOBA Y PERIFERICAS

//End Initiate Timer1 Interrupt

TRISB = 3;          // RB0-RB1

Lcd_Init();        // Initialize LCD

Lcd_Cmd(_LCD_CLEAR); // Clear LCD display

Lcd_Cmd(_LCD_CURSOR_OFF); // Turn cursor off

I2C1_Init(100000); // initialize I2C at 100KHz

return_;

text = ": : ";

```

```

Lcd_Out(1, 3, text);
text = "/ /20 ";
Lcd_Out(2, 3, text);
write_value(0, 0); //Reset seconds and start oscillator
//Lcd_Cmd(_LCD_CLEAR);
while(1){
if(i == 1){Lcd_Cmd(_LCD_CLEAR); // SI RB0 = 1
//Lcd_Cmd(_LCD_CLEAR);
text = "Adjust Minute:";
Lcd_Out(1, 2, text); // SE IMPRIME
minute = minute + minute10 * 10; // PARA AUMENTAR MINUTO DE 1 EN 1
//-----
while(1){
if (Button(&PORTB, 0, 100, 0)) i++; //INCREMENTA I
if(i!=1){
minute = ((minute/10) << 4) + (minute % 10);
write_value( 1 , minute);

goto return_;}
//-----
if (Button(&PORTB, 1, 100, 0)) minute++;
if (minute > 59) minute = 0;
Lcd_Chr(2, 8, (minute/10) + 48);
Lcd_Chr(2, 9, (minute % 10) + 48);}
}
//-----

```

```

if(i == 2){ Lcd_Cmd(_LCD_CLEAR);
text = "Adjust Hour:";
Lcd_Out(1, 2, text);
hour = hour + hour10 * 10;
while(1){
if (Button(&PORTB, 0, 100, 0)) i++;
if(i!=2){
hour = ((hour/10) << 4) + (hour % 10);
write_value( 2 , hour);
goto return_;}
if (Button(&PORTB, 1, 100, 0)) hour++;
if (hour > 23) hour = 0;
Lcd_Chr(2, 8, (hour/10) + 48);
Lcd_Chr(2, 9, (hour % 10) + 48);}
}
//-----
if(i == 3){ Lcd_Cmd(_LCD_CLEAR);
text = "Adjust Date:";
Lcd_Out(1, 2, text);
date = date + date10 * 10;
while(1){
if (Button(&PORTB, 0, 100, 0)) i++;
if(i!=3){
date = ((date/10) << 4) + (date % 10);
write_value( 3 , date);
goto return_;}
}

```

```

if (Button(&PORTB, 1, 100, 0)) date++;
if (date > 31) date = 1;
Lcd_Chr(2, 8, (date/10) + 48);
Lcd_Chr(2, 9, (date % 10) + 48);}
}
//-----

if(i == 4){ Lcd_Cmd(_LCD_CLEAR);
text = "Adjust Month:";
Lcd_Out(1, 2, text);
month = month + month10 * 10;
while(1){
if (Button(&PORTB, 0, 100, 0)) i++;
if(i!=4){
month = ((month/10) << 4) + (month % 10);
write_value( 4 , month);
goto return_;}
if (Button(&PORTB, 1, 100, 0)) month++;
if (month > 12) month = 1;
Lcd_Chr(2, 8, (month/10) + 48);
Lcd_Chr(2, 9, (month % 10) + 48);}
}
//-----

if(i == 5){ Lcd_Cmd(_LCD_CLEAR);
text = "Adjust Year:";
Lcd_Out(1, 2, text);
year = year + year10 * 10;

```

```

while(1){
if (Button(&PORTB, 0, 100, 0)) {i++;
if (i > 5) i = 0;}
if(i!=5){
year = ((year/10) << 4) + (year % 10);
write_value( 5 , year);
goto return_;}
if (Button(&PORTB, 1, 100, 0)) year++;
if (year > 99) year = 0;
Lcd_Chr(2, 7, 2 + 48);
Lcd_Chr(2, 8, 0 + 48);
Lcd_Chr(2, 9, (year/10) + 48);
Lcd_Chr(2, 10, (year % 10) + 48);}
}
//-----
if (n > 19){
n = 0;
//StartSignal();
//CheckResponse();
if (a == 1){
//ReadData();
//humidity = j;
//ReadData();
// ReadData();
temp = j;}
else {

```

```
humidity = 0;
temp = 0; }
}
if (Button(&PORTB, 0, 100, 0)) i++;
I2C1_Start();
I2C1_Wr(0xD0);
I2C1_Wr(0);
I2C1_Repeated_Start();
I2C1_Wr(0xD1);
second =I2C1_Rd(1);
minute =I2C1_Rd(1);
hour =I2C1_Rd(1);
day =I2C1_Rd(1);
date =I2C1_Rd(1);
month =I2C1_Rd(1);
year =I2C1_Rd(0);
I2C1_Stop();
display();
```

ANEXO B

PROGRAMACIÓN DEL GIRO DEL MOTOR EN LENGUAJE MIKRO C

```
conta=second;
conta10=second10;
contas=second;
contas10=second10;

if (portd.f6==0){
Delay_ms(200);
dato1=1;}
while(dato1==1){
if((conta10=0x02)){
y=y++;
Soft_UART_Write((y));
Delay_ms(990);
portb.f5=1;
portb.f4=0;
if((y==0xF0)){
y=0;
y10=y10++;
UART1_Write_Text("_"); //INICIO
Delay_ms(496);
UART1_Write_Text("A");
if(y10==0x32){
y10=0;
//y20=0;
dato1=2;
}
//goto return_;
}}}
while(dato1==2){
if((conta10=0x02)){ //
y=y++;
Soft_UART_Write((y));
```

```

Delay_ms(990);
portb.f5=0;
portb.f4=1;
if((y==0xF0)){
    y=0;
    y20=y20++;
    UART1_Write_Text("!"); //VUELTA
    Delay_ms(465); // AQUI PONER EL TIEMPO PARA QUE SE MUEVA CADA GRADO
    UART1_Write_Text("A");

if(y20==0x32){
    y20=0;
    //y10=0;
    dato1=1;
}
//goto return_;
} }
}
}
}

```


11.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the 40/44-pin devices.

The conversion of an analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low-voltage reference input that is software selectable to some combination of V_{DD}, V_{SS}, RA2 or RA3.

The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D clock must be derived from the A/D's internal RC oscillator.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be the voltage reference) or as digital I/O.

Additional information on using the A/D module can be found in the PIC® Mid-Range MCU Family Reference Manual (DS33023).

REGISTER 11-1: ADCON0 REGISTER (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7						bit 0	

bit 7-6 ADCS1:ADCS0: A/D Conversion Clock Select bits (ADCON0 bits in bold)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	Frc (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	Frc (clock derived from the internal A/D RC oscillator)

bit 5-3 CHS2:CHS0: Analog Channel Select bits

- 000 = Channel 0 (AN0)
- 001 = Channel 1 (AN1)
- 010 = Channel 2 (AN2)
- 011 = Channel 3 (AN3)
- 100 = Channel 4 (AN4)
- 101 = Channel 5 (AN5)
- 110 = Channel 6 (AN6)
- 111 = Channel 7 (AN7)

Note: The PIC16F873A/876A devices only implement A/D channels 0 through 4; the unimplemented selections are reserved. Do not select any unimplemented channels with these devices.

bit 2 GO/DONE: A/D Conversion Status bit

When ADON = 1:

- 1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)
- 0 = A/D conversion not in progress

bit 1 Unimplemented: Read as '0'

bit 0 ADON: A/D On bit

- 1 = A/D converter module is powered up
- 0 = A/D converter module is shut-off and consumes no operating current

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

PIC16F87XA

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADC02	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

bit 7 **ADFM**: A/D Result Format Select bit

1 = Right justified. Six (6) Most Significant bits of ADRESH are read as '0'.
0 = Left justified. Six (6) Least Significant bits of ADRESL are read as '0'.

bit 6 **ADC02**: A/D Conversion Clock Select bit (ADCON1 bits in shaded area and in bold)

ADCON1 <ADC02>	ADCON0 <ADC01:ADC00>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-4 **Unimplemented**: Read as '0'

bit 3-0 **PCFG3:PCFG0**: A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN8	AN6	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog Input D = Digital I/O

C/R = # of analog input channels/# of A/D voltage references

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

- n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

Note: On any device Reset, the port pins that are multiplexed with analog functions (ANx) are forced to be an analog input.

CONFIGURACIÓN UART PARA LA TX DE DATOS

PIC16F87XA

10.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI.) The USART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The USART can be configured in the following modes:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

Bit SPEN (RCSTA<7>) and bits TRISC<7:5> have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

The USART module also has a multi-processor communication capability using 9-bit address detection.

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
C8SRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7				bit 0			

bit 7	C8SRC: Clock Source Select bit <u>Asynchronous mode:</u> Don't care. <u>Synchronous mode:</u> 1 = Master mode (clock generated internally from BRG) 0 = Slave mode (clock from external source)
bit 6	TX9: 9-bit Transmit Enable bit 1 = Selects 9-bit transmission 0 = Selects 8-bit transmission
bit 5	TXEN: Transmit Enable bit 1 = Transmit enabled 0 = Transmit disabled Note: SPEN/CREN overrides TXEN in Sync mode.
bit 4	SYNC: USART Mode Select bit 1 = Synchronous mode 0 = Asynchronous mode
bit 3	Unimplemented: Read as '0'
bit 2	BRGH: High Baud Rate Select bit <u>Asynchronous mode:</u> 1 = High speed 0 = Low speed <u>Synchronous mode:</u> Unused in this mode.
bit 1	TRMT: Transmit Shift Register Status bit 1 = TSR empty 0 = TSR full
bit 0	TX9D: 9th bit of Transmit Data, can be Parity bit

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

PIC16F87XA

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
							bit 0
							bit 7

- bit 7 **SPEN**: Serial Port Enable bit
 1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
 0 = Serial port disabled
- bit 6 **RX9**: 9-bit Receive Enable bit
 1 = Selects 9-bit reception
 0 = Selects 8-bit reception
- bit 5 **SREN**: Single Receive Enable bit
Asynchronous mode:
 Don't care.
Synchronous mode – Master:
 1 = Enables single receive
 0 = Disables single receive
 This bit is cleared after reception is complete.
Synchronous mode – Slave:
 Don't care.
- bit 4 **CREN**: Continuous Receive Enable bit
Asynchronous mode:
 1 = Enables continuous receive
 0 = Disables continuous receive
Synchronous mode:
 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
 0 = Disables continuous receive
- bit 3 **ADDEN**: Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
 1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
- bit 2 **FERR**: Framing Error bit
 1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
 0 = No framing error
- bit 1 **OERR**: Overrun Error bit
 1 = Overrun error (can be cleared by clearing bit CREN)
 0 = No overrun error
- bit 0 **RX9D**: 9th bit of Received Data (can be parity bit but must be calculated by user firmware)

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

10.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 10-1 shows the formula for computation of the baud rate for different USART modes which only apply in Master mode (Internal clock).

Given the desired baud rate and F_{osc} , the nearest integer value for the SPBRG register can be calculated using the formula in Table 10-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the $F_{osc}/(16(X+1))$ equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

10.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

TABLE 10-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

Legend: X = value in SPBRG (0 to 255)

TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 - 010	0000 - 010
18h	RCSTA	SPEN	RX9	REN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

ANEXO C

DATASHEET PANEL SOLAR

TECHNICAL DATA SHEET

SM536-85, SM536-90, SM536-95



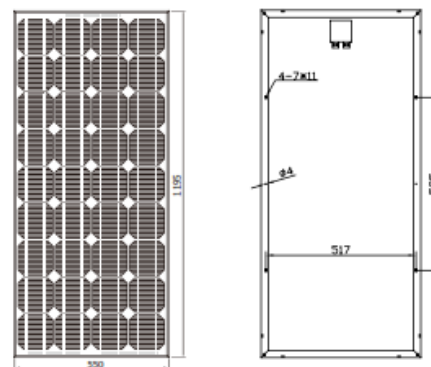
Technical Data

Type	SM536-85, SM536-90, SM536-95
Type of Solar Cell	Mono-crystalline, 125mm×125mm
Number of Cells	36 pcs
Size of module	1200×540×35mm (125×125mm cell)
Module, Weight	8 kg
Connector / Cross-Section	Cxl Renhe, TOP
Cables, Length (+/-)	Customizable
Front Cover Glass	Safety Glass
Front Cover Glass, Thickness	3.2 mm
Frame	Anodized Aluminium

Electrical Data

Maximum Power	$P_{MPP\ STC}$	85 W	90 W	95 W
Power Tolerance	$\Delta_{\ STC}$	± 2%	± 2%	± 2%
Maximum Power Voltage	$U_{MPP\ STC}$	17.4 V	17.6 V	17.8 V
Maximum Power Current	$I_{MPP\ STC}$	4.89 A	5.11 A	5.33 A
Open Circuit Voltage	$U_{OC\ STC}$	21.6 V	21.8 V	22.2 V
Short Circuit Current	$I_{SC\ STC}$	5.25 A	5.51 A	5.79 A
Cell Efficiency	$\eta_{\ STC}$	15.9%	16.8%	17.8%
Maximum System Voltage	U_{DC}	700 V	700 V	700 V

STC: Irradiance 1000 W/m²; Spectrum AM 1,5; Cell Temperature 25°C, Wind 0 m/s



Temperature Coefficients

Power Coefficient	$\alpha_p (P_{MPP})$	- 0.45 %/K
Voltage Coefficient	$\beta_v (U_{OC})$	- 0.35 %/K
Current Coefficient	$\phi_k (I_{SC})$	0.065 ± 0.015 %/K

Power Warranty

- 10 years performance warranty to 90 %
- 25 years performance warranty to 80 %
- 5 years warranty against production and material defects

Certifications

IEC 61215, IEC 61730, TUV, CE, ISO9001:2000

Simax(suzhou) Green New Energy Co.,Ltd Add:beibanjing riverside,suzhou road,taicang city,jiangsu province,China
 Tel: +86 – 512 – 53378555 Fax: +86-512-53378556
 Url: www.simaxsolar.com Email: info@simaxsolar.com jacky@simaxsolar.com



DATASHEET DEL SENSOR ACS712

ACS712

Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor

Description (continued)

the device at up to 5× overcurrent conditions. The terminals of the conductive path are electrically isolated from the signal leads (pins 5 through 8). This allows the ACS712 to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The ACS712 is provided in a small, surface mount SOIC8 package. The leadframe is plated with 100% matte tin, which is compatible with standard lead (Pb) free printed circuit board assembly processes. Internally, the device is Pb-free, except for flip-chip high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.

Selection Guide

Part Number	Packing*	T _A (°C)	Optimized Range, I _p (A)	Sensitivity, Sens (Typ) (mV/A)
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	-40 to 85	±5	185
ACS712ELCTR-20A-T	Tape and reel, 3000 pieces/reel	-40 to 85	±20	100
ACS712ELCTR-30A-T	Tape and reel, 3000 pieces/reel	-40 to 85	±30	66

*Contact Allegro for additional packing options.

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Supply Voltage	V _{CC}		5	V
Reverse Supply Voltage	V _{RCC}		-0.1	V
Output Voltage	V _{OUT}		5	V
Reverse Output Voltage	V _{ROUT}		-0.1	V
Output Current Source	I _{OUT(SOURCE)}		3	mA
Output Current Sink	I _{OUT(SINK)}		10	mA
Overcurrent Transient Tolerance	I _p	1 pulse, 100 ms	100	A
Nominal Operating Ambient Temperature	T _A	Range E	-40 to 85	°C
Maximum Junction Temperature	T _{J(MAX)}		165	°C
Storage Temperature	T _{stg}		-65 to 170	°C

Isolation Characteristics

Characteristic	Symbol	Notes	Rating	Unit
Dielectric Strength Test Voltage*	V _{ISO}	Agency type-tested for 60 seconds per UL standard 60950-1, 1st Edition	2100	VAC
Working Voltage for Basic Isolation	V _{WFSI}	For basic (single) isolation per UL standard 60950-1, 1st Edition	354	VDC or V _{pk}
Working Voltage for Reinforced Isolation	V _{WFR}	For reinforced (double) isolation per UL standard 60950-1, 1st Edition	184	VDC or V _{pk}

* Allegro does not conduct 60-second testing. It is done only during the UL certification process.

Parameter	Specification
Fire and Electric Shock	CAN/CSA-C22.2 No. 60950-1-03 UL 60950-1:2003 EN 60950-1:2001

ACS712

Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor

x05B PERFORMANCE CHARACTERISTICS¹ $T_A = -40^\circ\text{C}$ to 85°C , $C_F = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-5	-	5	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^\circ\text{C}$	180	185	190	mV/A
Noise	$V_{\text{Noise(PP)}}$	Peak-to-peak, $T_A = 25^\circ\text{C}$, 185 mV/A programmed Sensitivity, $C_F = 47\text{ nF}$, $C_{\text{OUT}} = \text{open}$, 2 kHz bandwidth	-	21	-	mV
Zero Current Output Slope	$\Delta V_{\text{OUT}(0)}$	$T_A = -40^\circ\text{C}$ to 25°C	-	-0.26	-	mV/°C
		$T_A = 25^\circ\text{C}$ to 150°C	-	-0.08	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^\circ\text{C}$ to 25°C	-	0.054	-	mV/A/°C
		$T_A = 25^\circ\text{C}$ to 150°C	-	-0.008	-	mV/A/°C
Total Output Error ²	E_{TOT}	$I_P = \pm 5\text{ A}$, $T_A = 25^\circ\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(\text{max})}$, is not exceeded.

²Percentage of I_P with $I_P = 5\text{ A}$. Output filtered.

x20A PERFORMANCE CHARACTERISTICS¹ $T_A = -40^\circ\text{C}$ to 85°C , $C_F = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-20	-	20	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^\circ\text{C}$	96	100	104	mV/A
Noise	$V_{\text{Noise(PP)}}$	Peak-to-peak, $T_A = 25^\circ\text{C}$, 100 mV/A programmed Sensitivity, $C_F = 47\text{ nF}$, $C_{\text{OUT}} = \text{open}$, 2 kHz bandwidth	-	11	-	mV
Zero Current Output Slope	$\Delta V_{\text{OUT}(0)}$	$T_A = -40^\circ\text{C}$ to 25°C	-	-0.34	-	mV/°C
		$T_A = 25^\circ\text{C}$ to 150°C	-	-0.07	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^\circ\text{C}$ to 25°C	-	0.017	-	mV/A/°C
		$T_A = 25^\circ\text{C}$ to 150°C	-	-0.004	-	mV/A/°C
Total Output Error ²	E_{TOT}	$I_P = \pm 20\text{ A}$, $T_A = 25^\circ\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(\text{max})}$, is not exceeded.

²Percentage of I_P with $I_P = 20\text{ A}$. Output filtered.

x30A PERFORMANCE CHARACTERISTICS¹ $T_A = -40^\circ\text{C}$ to 85°C , $C_F = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-30	-	30	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^\circ\text{C}$	63	66	69	mV/A
Noise	$V_{\text{Noise(PP)}}$	Peak-to-peak, $T_A = 25^\circ\text{C}$, 66 mV/A programmed Sensitivity, $C_F = 47\text{ nF}$, $C_{\text{OUT}} = \text{open}$, 2 kHz bandwidth	-	7	-	mV
Zero Current Output Slope	$\Delta V_{\text{OUT}(0)}$	$T_A = -40^\circ\text{C}$ to 25°C	-	-0.35	-	mV/°C
		$T_A = 25^\circ\text{C}$ to 150°C	-	-0.08	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^\circ\text{C}$ to 25°C	-	0.007	-	mV/A/°C
		$T_A = 25^\circ\text{C}$ to 150°C	-	-0.002	-	mV/A/°C
Total Output Error ²	E_{TOT}	$I_P = \pm 30\text{ A}$, $T_A = 25^\circ\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(\text{max})}$, is not exceeded.

²Percentage of I_P with $I_P = 30\text{ A}$. Output filtered.

DATASHHETH DEL SABERTOOTH 2X25 V2



Sabertooth 2x25 V2 User's Guide April 2012



Input voltage: 6-30V nominal, 33.6V absolute max.

Output current: Up to 25A continuous per channel. Peak loads may be up to 50A per channel for a few seconds.

5V switching BEC: Up to 1A continuous and 1.5A peaks across the entire range of input voltages.

Recommended power sources are:

- 5 to 20 cells high capacity NiMH or NiCd
- 2s to 8s lithium ion or lithium polymer. Sabertooth motor drivers have a lithium battery mode to prevent cell damage due to over-discharge of lithium battery packs.
- 6v to 30V high capacity lead acid
- 6v to 30V power supply (when in parallel with a suitable battery).

All batteries must be capable of maintaining a steady voltage when supplying 20+ amps (AA or 9V batteries aren't going to cut it! An 18Ah lead-acid battery is a good starting point)

Dimensions:

Size: 2.6" x 3.2" x .82" 65 x 80 x 21mm
Weight: 3.2oz / 90g

Hooking up the Sabertooth motor driver

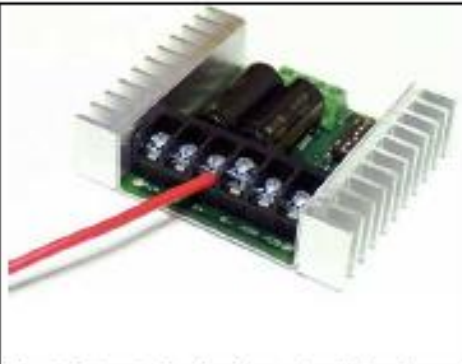
All connections to the Sabertooth are done with screw terminals. This makes it easy to set up and reconfigure your project. If you've never used screw terminal connections before, here is a quick overview.



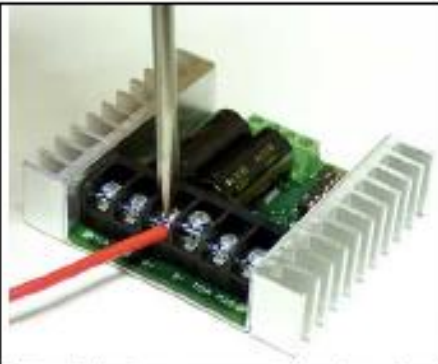
Step 1: Strip the wire which you are using approximately $\frac{1}{4}$ ". The wires may be 12 gauge to 30 gauge. Use thicker wire for high current applications.



Step 2: With a large screwdriver, turn the top screw counter-clockwise until it stops gently.



Step 3: Insert the stripped portion of the wire into the opening in the screw terminal.



Step 4: Turn the top screw clockwise until you encounter resistance, then tighten the screw firmly. Pull on the wire gently to ensure that it is secured.

Mode 3: Simplified Serial Mode

Simplified serial uses TTL level single-byte serial commands to set the motor speed and direction. This makes it easy to interface to microcontrollers and PCs, without having to implement a packet-based communications protocol. Simplified serial is a one-direction only interface. The transmit line from the host is connected to S1. The host's receive line is not connected to the Sabertooth. Because of this, multiple drivers can be connected to the same serial transmitter. If using a true RS-232 device like a PC's serial port, it is necessary to use a level converter to shift the -10V to 10V RS-232 levels to the 0V to 5V TTL levels the Sabertooth is expecting. This is usually done with a Max232 type chip. If using a TTL serial device like a microcontroller or a USB-to-TTL serial converter, the TX line may be connected directly to S1.

Because Sabertooth controls two motors with one 8 byte character, when operating in Simplified Serial mode, each motor has 7 bits of resolution. Sending a character between 1 and 127 will control motor 1. 1 is full reverse, 64 is stop and 127 is full forward. Sending a character between 128 and 255 will control motor 2. 128 is full reverse, 192 is stop and 255 is full forward. Character 0 (hex 0x00) is a special case. Sending this character will shut down both motors.

Baud Rate Selection

Simplified Serial operates with an 8N1 protocol – 8 data bytes, no parity bits and one stop bit. The baud rate is selected by switches 4 and 5 from the following 4 options:

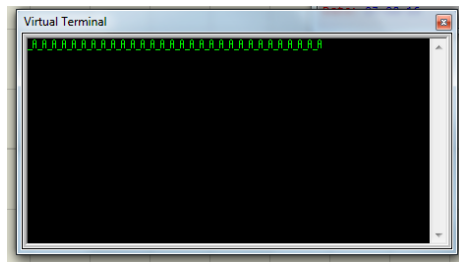
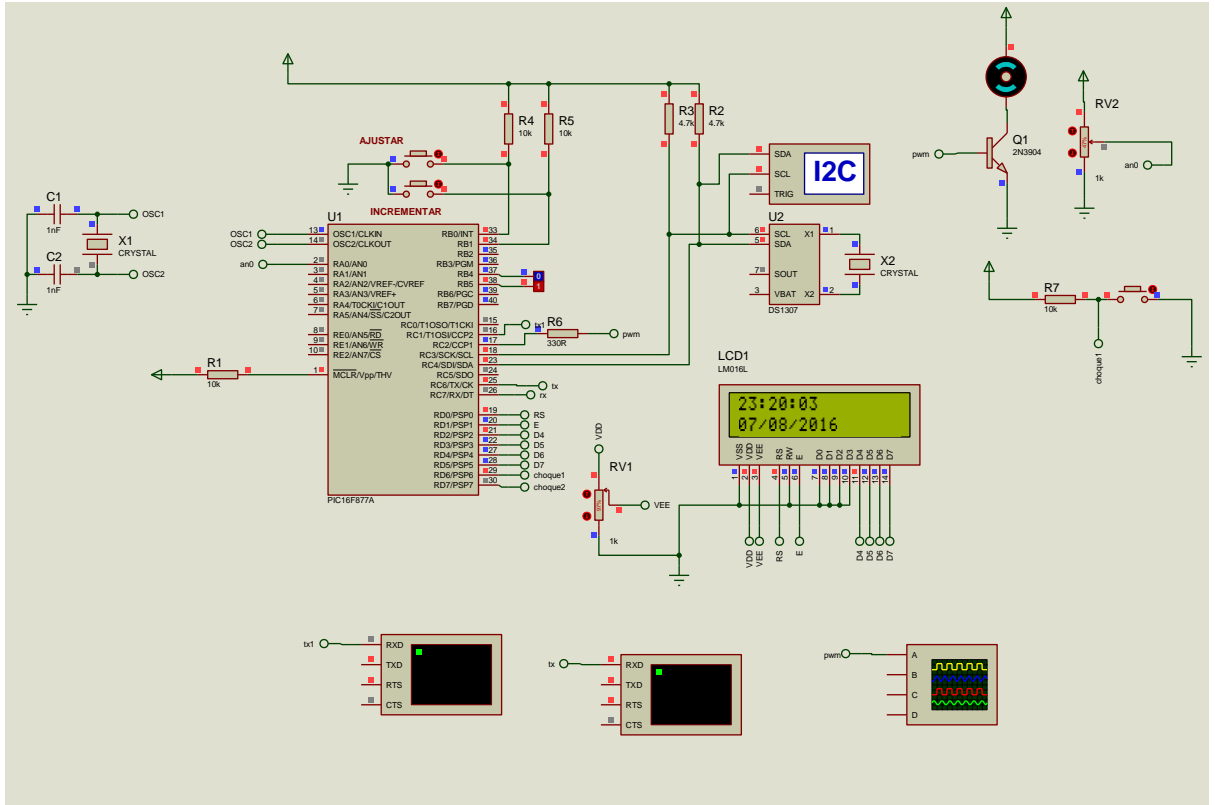
	
2400 Baud: 01x00x	9600 Baud: 01x10x
	
19200 Baud: 01x01x	38400 Baud: 01x11x

What baud rate to use is dependent on what your host can provide and the update speed necessary. 9600 baud or 19200 baud is recommended as the best starting points. If communication is unreliable, decrease the baud rate. If communications are reliable, you may increase the baud rate. The maximum update speed on the Sabertooth is approximately 2000 commands per second. Sending characters faster than this will not cause problems, but it will not increase the responsiveness of the controller either.

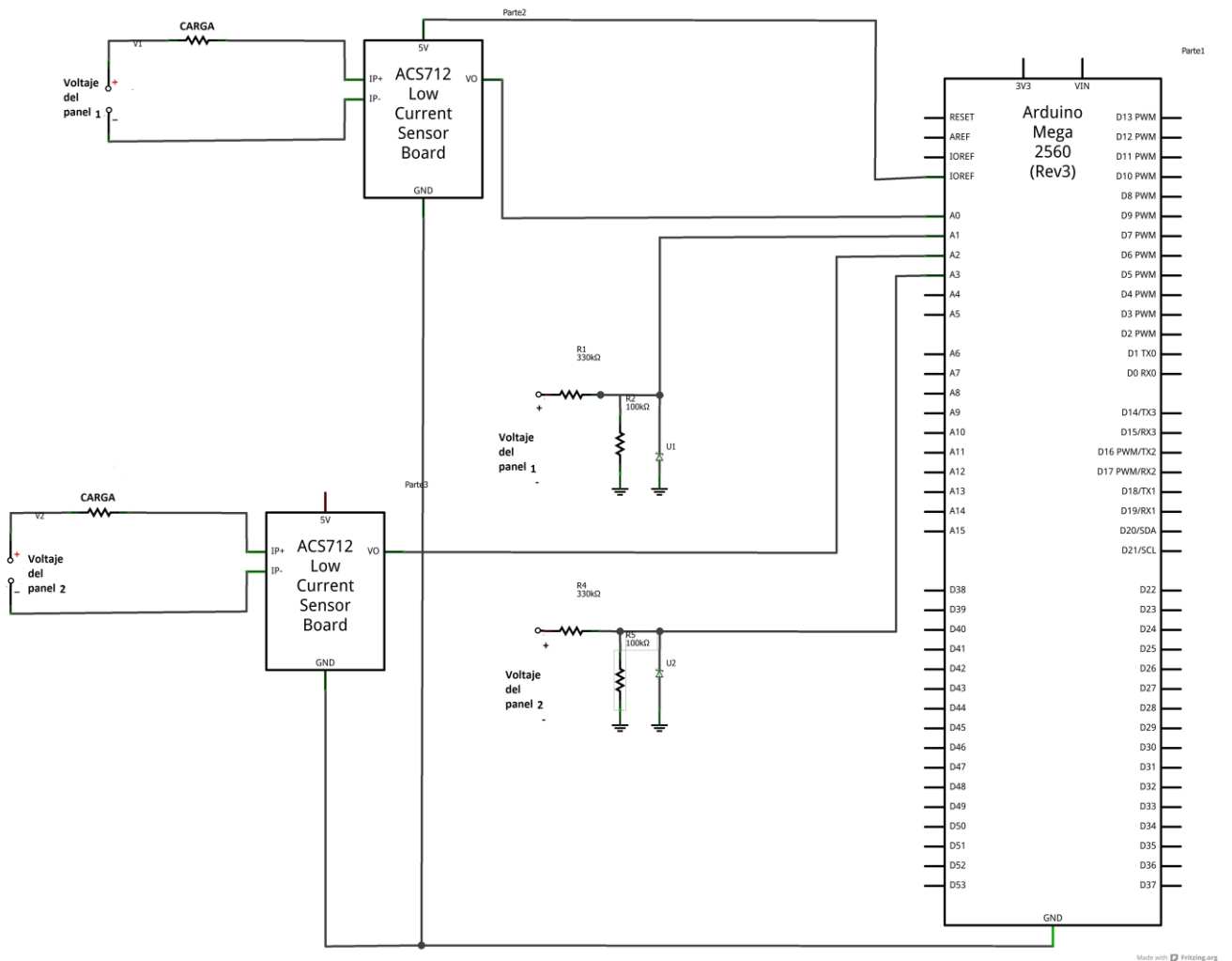
The baud rate may be changed with power on by changing the DIP switch settings. There is no need to reset or cycle power after a baud rate change.

There are 2 operating options for Simplified Serial, selected by the position of Switch 6.

DISEÑO DEL SEGUIDOR SOLAR



ESQUEMA DE CONEXIÓN DE LOS SENSORES



PROGRAMACION DEL MODELAMIENTO EN MATLAB

```
function varargout = GRAFICA1(varargin)

% GRAFICA1 M-file for GRAFICA1.fig

%   GRAFICA1, by itself, creates a new GRAFICA1 or raises the existing
%   singleton*.

%
%   H = GRAFICA1 returns the handle to a new GRAFICA1 or the handle to
%   the existing singleton*.

%
%   GRAFICA1('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GRAFICA1.M with the given input arguments.

%
%   GRAFICA1('Property','Value',...) creates a new GRAFICA1 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before GRAFICA1_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to GRAFICA1_OpeningFcn via varargin.

%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GRAFICA1
```

```

% Last Modified by GUIDE v2.5 29-Jul-2016 22:24:21

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;

gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @GRAFICA1_OpeningFcn, ...
                  'gui_OutputFcn', @GRAFICA1_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before GRAFICA1 is made visible.
function GRAFICA1_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject    handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

```



```

% handles  structure with handles and user data (see GUIDATA)
% varargin  command line arguments to GRAFICA1 (see VARARGIN)

% Choose default command line output for GRAFICA1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GRAFICA1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GRAFICA1_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject   handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function t1_Callback(hObject, eventdata, handles)
% hObject   handle to t1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of t1 as text
%     str2double(get(hObject,'String')) returns contents of t1 as a double
% --- Executes during object creation, after setting all properties.
function t1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to t1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function n_Callback(hObject, eventdata, handles)
% hObject    handle to n (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of n as text
%     str2double(get(hObject,'String')) returns contents of n as a double

% --- Executes during object creation, after setting all properties.
function n_CreateFcn(hObject, eventdata, handles)
% hObject    handle to n (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
% --- Executes on button press in pushbutton2.
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
% hObject handle to pushbutton2 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
n=1;
```

```
is=(20*10-3);
```

```
Tk=handles.t1+273;
```

```
k=(1.38*10(-23));
```

```
q=(1.6*10(-19));
```

```
Vt=(k*Tk)/(q);
```

```
disp(' LA CORRIENTE DEL DIODO DE SILICIO ES IGUAL A:');
```

```
Vd= -1:0.001:1;
```

```
id=is*(exp((Vd)/(n*Vt))-1)
```

```
plot(Vd,id)
```

```
grid on
```

```

hold on
plot(Vd,id,'r' )

%%%%%%%%

disp('      DIODO DE GERMANIO');
n=2;
is=(20*10^-3);

Tk=handles.t1+273;
k=(1.38*10^(-23));
q=(1.6*10^(-19));
Vt=(k*Tk)/(q);

Vd= -1:0.001:1;
id=is*(exp((Vd)/(n*Vt))-1)
plot(Vd,id)
grid on
hold on
plot(Vd,id,'g' )

% --- Executes during object creation, after setting all properties.
function pushbutton2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

%%%%%%%%%%

```

```

% --- Executes on button press in pushbutton3.

function pushbutton3_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.t1,'string',num2str("));
set(handles.n,'string',num2str("));

% --- Executes on button press in pushbutton4.

function pushbutton4_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

MODELAMIENTO;

close(gcf)

```

ACOPLAMIENTO DE LAS FASES DEL PROYECTO



