

SISTEMA DE SEGUIMIENTO SOLAR PARA PANELES FOTOVOLTAICOS

Ramiro A. Echenique¹, Ernesto J. Perino², Luis A. Odicino³, Daniel A. Perelló⁴, Roberto M. Murdocca⁵.

Laboratorio de Electrónica, Investigación y Servicios (L.E.I.S.) - Laboratorio de Energía Solar y Medio Ambiente - Facultad de Ciencias Físico Matemáticas y Naturales -U.N.S.L.
Tel. 2664530000-int. 6119, e-mail: Ra.echenique@gmail.com, ejperino@unsl.edu.ar, laodicino@hotmail.com, adp2006@gmail.com, mmurdocca@unsl.edu.ar.

Recibido 13/08/18, aceptado 27/09/18

RESUMEN: El objetivo del presente trabajo es desarrollar un seguidor solar de dos ejes para ser aplicado a paneles fotovoltaicos. Esta aplicación permite que el panel incremente su producción en un 20 a 30%, dependiendo del mes y la hora del día. El sistema desarrollado contempla 5 modos de funcionamiento. Se utilizó una placa Arduino Mega ya que esta satisface los requerimientos de memoria y potencia de cálculo requeridos. Se usaron sensores LDR para el posicionamiento; además, el sistema contempla la posibilidad de conectarse mediante telefonía celular. Para el modelo a escala se usaron servomotores marca Tower Pro modelo SG90 por sus bajos requerimientos energéticos que permiten modularizar el sistema. Se realizaron simulaciones donde el error resultante es inferior al 1% para el posicionamiento y el aumento de energía producida está dentro de los valores esperados. Se proponen mejoras en el software de manera de hacerlo más accesible.

Palabras clave: seguidor solar, paneles fotovoltaicos, energía solar, Arduino, tecnología.

INTRODUCCION

Un seguidor solar, es un dispositivo capaz de orientar sus paneles de forma que estos permanezcan aproximadamente perpendiculares a los rayos solares a lo largo de la jornada.

Existen básicamente dos tipos de seguidores solares según su rango de movimientos: de un eje con un único grado de libertad (horizontal, vertical u oblicua) y, de dos ejes con dos grados de libertad, posibilitando un seguimiento solar más preciso. En la Figura 1 se muestran ambas posibilidades. En el presente trabajo se optó por el de dos grados de libertad.

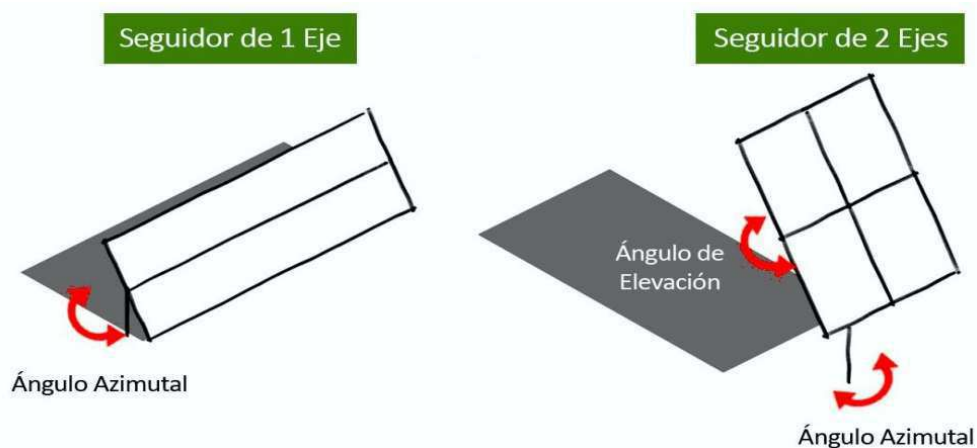


Figura 1: Esquema de seguimiento solar.

¹Profesional Independiente

²Investigador U.N.S.L.

³Investigador U.N.S.L.

⁴Investigador U.N.S.L.

⁵Investigador U.N.S.L.

Se desarrolló un prototipo económico con componentes de bajo costo y con disponibilidad en el mercado local. Dicho prototipo cuenta con un registro de datos (Datalogging) que le permite almacenar la información recolectada. Una vez configurado, puede hacer un seguimiento solar de manera desatendida y automática. Además, cuenta con la posibilidad de actualizar el firmware, lo que le proporciona entre otras cosas la adaptación mecánica y el tipo de motores que se pueden emplear. Por último y, no menos importante, se puede apreciar que el prototipo cuenta con un programa que, mediante una aplicación, puede ser manipulado desde un Smartphone, brindando practicidad y versatilidad.

Entre las metas se destacan:

El sistema deberá ser totalmente autónomo.

Deberá poseer independencia de la ubicación geográfica: se ingresa latitud y longitud del lugar de operación.

El enfoque del dispositivo presentará un error máximo de $\pm 3^\circ$.

El sistema deberá estar optimizado en cuanto al consumo energético.

MODOS DE OPERACIÓN

El sistema permitirá 5 modos de operación, denominados A, S, M, E y O, se detallan a continuación:

Modo A: cálculo automático de la trayectoria; modo S: utilización de sensores; modos M: combinación de modo A y S; modo E: emulación solar; y modo O: apagado del seguidor solar.

Para poder efectuar los cálculos se emplea el modo A; en él, el prototipo requiere de las coordenadas geográficas del lugar donde se encontrará (latitud y longitud), la fecha del año y hora del día. A partir de dichos datos de entrada, el sistema calcula las coordenadas del sol y acciona los servomotores, orientando el panel perpendicularmente a la radiación solar. Adicionalmente el procesador gestiona las lecturas de los paneles fotovoltaicos y las guarda en un archivo, como se esquematiza en la Figura 2.



Figura 2: Diagrama del sistema modo A

Para el cálculo de la trayectoria se implementaron las ecuaciones propuestas por Muhammad Iqbal.

Cálculo de la declinación:

$$\delta = (0.006918 - 0.399912 \cos \Gamma + 0.070257 \sin \Gamma - 0.006758 \cos 2\Gamma + 0.000907 \sin 2\Gamma - 0.002697 \cos 3\Gamma + 0.00148 \sin 3\Gamma) (180/\pi) \quad (1)$$

Donde Γ , en radianes, se denomina ángulo diario y está expresado por:

$$\Gamma = \frac{2\pi(d_n - 1)}{365} \quad (2)$$

Cálculo para la corrección del tiempo

$$E_t = (0.000075 + 0.001868 \cos \Gamma - 0.032077 \sin \Gamma - 0.014615 \cos 2\Gamma - 0.04089 \sin 2\Gamma) (229.18) \quad (3)$$

El modo S consiste en dirigir el soporte mecánico del seguidor solar a través de la lectura de sensores. El procesador interpreta esas lecturas y acciona los motores para que el dispositivo quede orientado hacia el sol. El mismo, también toma la lectura de los paneles y las almacena en un archivo. En la Figura 3 se observa el diagrama en bloques de este modo.

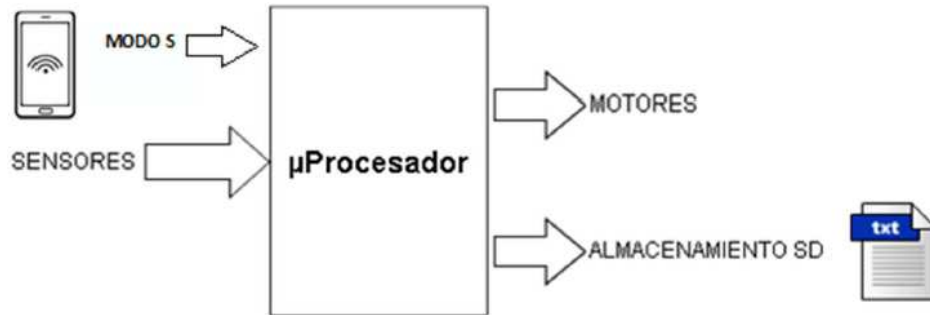


Figura 3: Diagrama del sistema modo S

El modo M o mixto es una combinación de los dos modos anteriores (A y S). En cada intervalo de tiempo (10 minutos) el procesador calcula la ubicación del sol. Luego, en ese instante, mediante la lectura de los sensores realiza un ajuste, repitiendo el ciclo mientras permanezca en dicho modo. Esto se observa en la Figura 4 a) y b).

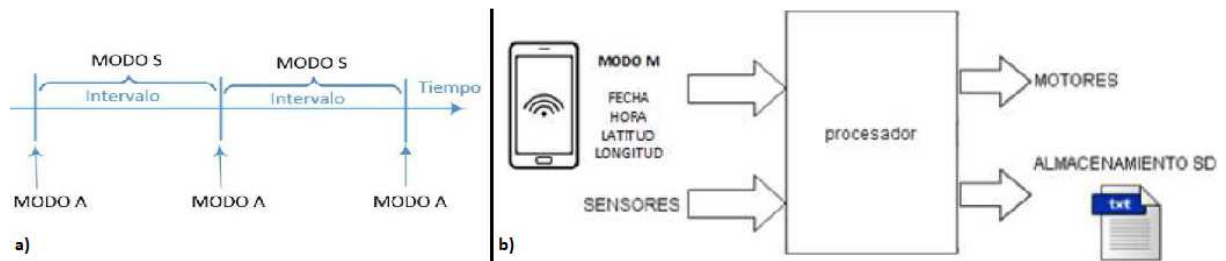


Figura 4: Diagrama del modo M

El modo E emula el movimiento solar en un corto lapso de tiempo, en el cual se modifica la hora más rápido que en su funcionamiento normal. Se hace uso del modo A. Emula un recorrido diario en 2 minutos aproximadamente (depende de la duración del día actual).

El modo O u off, consiste en llevar los motores de cada eje a una posición inicial pre-establecida.

Desarrollo

Diagrama en bloques general del sistema

El sistema consta de 3 partes bien diferenciadas, como se ilustra en la Figura 5:

Plataforma móvil.

Placa electrónica.

Celular o Tablet con la aplicación.

La primera consta básicamente de un soporte mecánico donde se monta el panel solar, los motores y los sensores; esta parte es la adaptable a la instalación que se desea realizar.

La segunda es donde se centra el proyecto y es la encargada de controlar los movimientos de la plataforma móvil para lo cual usa sensores y ecuaciones astronómicas, según sea el modo de operación. Utiliza un reloj en tiempo real para obtener precisión en los cálculos y una memoria externa para almacenar datos referidos a los paneles solares.

La tercera parte es el dispositivo móvil con la aplicación del seguidor solar, este tiene la función de configuración y monitorización del sistema. El sistema tiene la comodidad de realizar las configuraciones de manera inalámbrica y la posibilidad de utilizar cualquier dispositivo Android para esto.

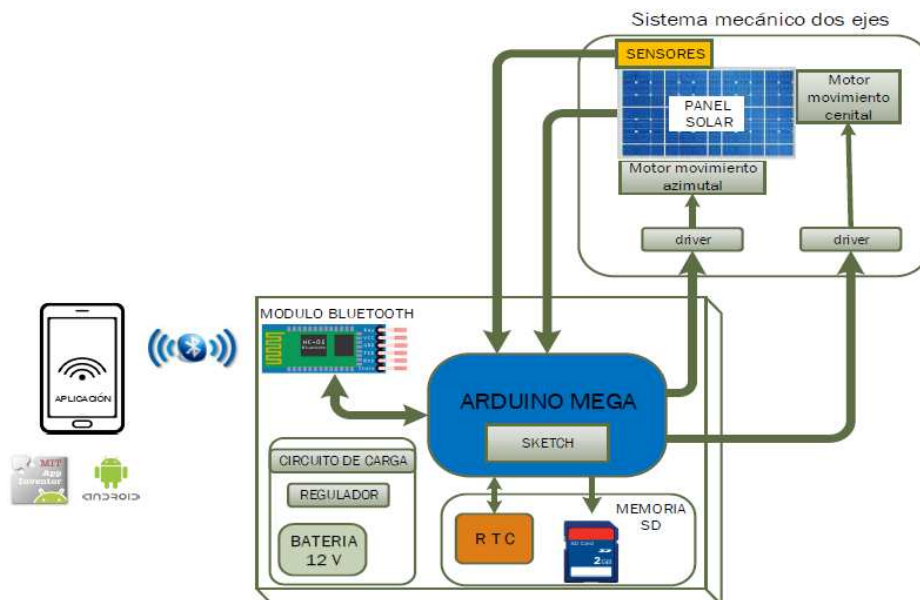


Figura 5: Bloques y dispositivos principales del seguidor solar

Hardware

Se describe a continuación el hardware utilizado en el presente trabajo.

Arduino

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa con entradas y salidas analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing. Pueden conectarse placas de expansión (shields), que brindan versatilidad de funcionamiento de la placa. Se han desarrollado varias versiones de Arduino debido a que es una plataforma de código abierto.

Más precisamente, Arduino Mega está basado en el micro controlador ATmega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16 MHz, conexión USB, jack de alimentación, conector ICSP y botón de reinicio. El Arduino Mega es compatible con la mayoría de las placas de expansión (shields) diseñados para Arduino Duemilanove, Diecimila o UNO.

Posee las siguientes especificaciones técnicas:

Procesador: ATmega2560

Voltaje Operativo: 5V

Pines digitales de entrada/salidas: 54 (de los cuales 15 proveen salida PWM)

Entradas analógicas: 16

Capacidad de corriente por pin: 40 mA

Memoria Flash: 256 KB (8KB son empleado por el bootloader arduino)

Memoria SRAM: 8 KB

Memoria EEPROM: 4 KB

Velocidad del reloj: 16 MHz

Se utilizó la placa Arduino Mega, principalmente, por poseer un potente procesador para realizar cálculos complejos en punto flotantes, suficiente memoria para poder llevar a cabo dichos cálculos y demás procesos, la versatilidad de tener varias UART, la cantidad necesaria de pines de entrada/salida para abastecer el presente proyecto y su amplia disponibilidad en el mercado local.

Reloj de Tiempo Real y Memoria de almacenamiento SD.

Para este proyecto se utilizó un shield del tipo 2 en 1 para simplificar la circuitería, el cual incluye un Reloj de Tiempo Real (RTC) y un lector de memorias SD.

La placa elegida es del fabricante Deek Robot en su modelo Data LoggingShield V1.0.

Esta placa tiene las siguientes características: Interface para tarjetas SD: en los sistemas de archivos FAT 16 y FAT 32. Reloj en Tiempo Real basado en el integrado DS 1307 de Maxim (reloj más fiable

que el interno de Arduino). Portabatería incluido para mantener fecha y hora actualizada (hasta dos años), incluso con Arduino desconectado. Regulador de 3.3V para alimentar la tarjeta SD. En el seguidor solar un reloj en tiempo real seguro (DS 1307) es muy importante, ya que las ecuaciones de trayectoria basan su cálculo en la fecha y hora de cada día para ir calculando la posición del Sol. Por otro lado, al tener integrado un lector de memorias SD se pueden ir registrando cada cierto intervalo el nivel de voltaje de los paneles solares entre otras cosas.

Sensores de posición solar.

Para el posicionamiento en el modo S se emplearon sensores LDR (Light Dependent Resistor), que consiste en una resistencia variable en función de la luz incidente sobre su superficie. Cuanto mayor sea la intensidad de la luz que incide en la superficie del LDR menor será su resistencia y cuanto menos luz incida mayor será su resistencia. Para el presente proyecto se dispusieron 4 sensores LDR, uno para cada cuadrante (superior derecho e izquierdo, inferior derecho e izquierdo), como se ilustra en la Figura 6.

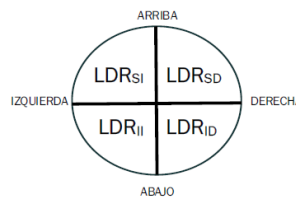


Figura 6: Distribución de los LDR.

El sistema siempre lleva a que los 4 sensores tengan la misma intensidad de luz, lo que indica que el panel está orientado hacia el sol. Las señales acondicionadas provenientes de cada LDR son leídas en entradas independientes del Arduino Mega, el cual realiza cálculos y ejecuta los movimientos de los servomotores.

Comunicación mediante módulo bluetooth HC-05.

Este módulo realiza la comunicación entre el Smartphone y el Arduino del seguidor solar. Los parámetros enviados desde el celular hacia Arduino Mega, a través del módulo bluetooth HC-05, son: fecha, hora, latitud, longitud, modos de funcionamiento. Los enviados desde Arduino son: los estados y los cálculos realizados.

Movimiento mediante servomotores.

Para el desplazamiento del sistema se emplearon dos servomotores, uno en cada eje. El servomotor consiste en un motor eléctrico, un juego de engranes, un potenciómetro y una tarjeta de control, todo dentro de una carcasa de plástico, la cual recibe una señal PWM (Pulse Width Modulation), la que controla la dirección y posición del servo entre 0° y 180°. En el proyecto solo se utilizan 90° del movimiento cenital y 180° del azimutal. Los servomotores utilizados en el modelo a escala son los SG90 de Tower Pro. Son micro servos de 9 gramos y sus requerimientos energéticos son bajos.

En la Figura 7 se muestra el circuito de conexión de los servomotores SG90. Se utilizaron los pines 44 y 45 (ambos PWM) como señales de control para el movimiento vertical y horizontal respectivamente. Se usó un regulador de voltaje 7805 para mantener una tensión de alimentación en los servos estable, la tensión V_{in} es la alimentación de todo el sistema es de 12V.

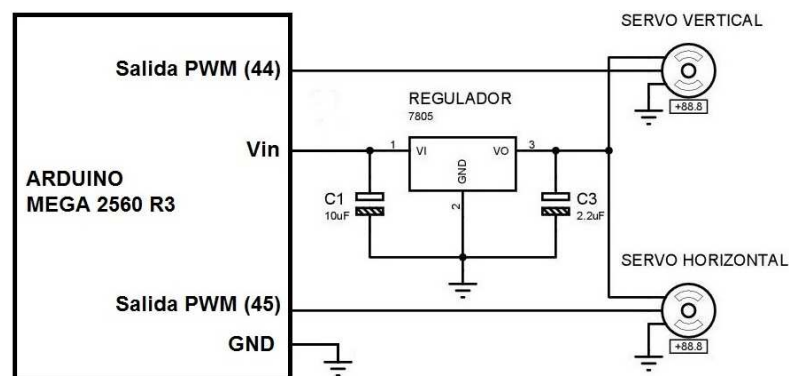


Figura 7: Diagrama de conexión de servomotores.

Cargador solar.

Su función es controlar constantemente la tensión de la batería, cuando está por debajo de cierto nivel de tensión (11,5V), el control de Arduino activa el circuito de carga que conecta el panel FV de 18V.

Software

IDE Arduino

Dado que el Arduino es una placa que ejecuta una serie de códigos previamente introducidos, se necesita un programa denominado IDE (Integrated Development Environment o Entorno de Desarrollo Integrado) para cargar dichos códigos. Este IDE, instalado en una PC, es un entorno muy sencillo de usar y en él se escribe el programa a ejecutar por Arduino. Una vez escrito, se carga a través del puerto USB y Arduino comenzará a trabajar de forma autónoma.

Estructura de programación

La estructura básica de un proyecto en Arduino es bastante simple y se compone de al menos dos partes: `setup()` y `loop()`. Ambas partes son obligatorias y encierran los bloques de instrucciones. El `setup()` es la parte encargada de la configuración y el `loop()` es la que contiene el programa que se ejecutará cíclicamente. Ambas funciones son indispensables para la correcta ejecución del programa.

La función de configuración (*setup*) realiza la configuración e inicialización de los elementos, y sólo se ejecuta una vez, justo después de reiniciar (o cuando se energiza), y no vuelve a ejecutarse hasta un nuevo reinicio. La función de lazo (*loop*), que se ejecuta inmediatamente después de `setup`, contiene el código que se ejecutará continuamente (lectura de entradas, activación de salidas, etc). La función `loop()` es el núcleo de los programas y se usa para el control activo de la placa Arduino.

Los elementos principales de un programa o sketch de Arduino son: Variables (espacio de memoria donde se almacenan datos variables); Funciones (trozos de código que pueden ser llamadas desde cualquier parte del sketch); `setup()` y `loop()` (funciones especiales y obligatorias de todo sketch); Librerías (conjunto de funciones pre creadas por terceros que pueden importarse en el código); Comentarios (para documentar el proyecto). La Figura 8 exhibe el sketch básico de Arduino.

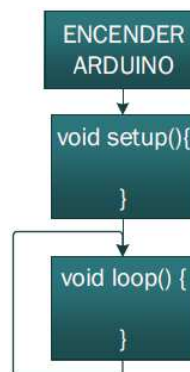


Figura 8: Diagrama de flujo básico de un sketch en Arduino.

Programación del seguidor solar.

Debido al hardware utilizado en el proyecto se requieren de librerías especiales para poder acceder a cada una de sus funciones. Las librerías utilizadas son: `Wire.h` maneja la comunicación del bus I²C entre el DS1307 y el Arduino. Viene por defecto añadida a la IDE; `RTClib.h` se apoya en la librería `Wire.h` y permite obtener y establecer el tiempo del RTC del DS1307; `SPI.h` para el manejo y funcionamiento del bus SPI (Serial Peripheral Interface), permitiendo comunicar los dispositivos SPI, con el Arduino como maestro; `SD.h` permite leer y escribir en tarjetas SD; `Servo.h` es una librería estándar en Arduino y sirve para el control de servomotores; `Math.h` para cálculos matemáticos, incluye un gran número de funciones útiles para manipular números en punto flotante.

Además, el presente proyecto se encuentra organizado y estructurado mediante el uso de funciones. Esto permite realizar tareas repetitivas y generar una programación más flexible, ordenada e intuitiva.

Para el programa de control del seguidor solar se desarrollaron las siguientes funciones: Parámetros, Temporizadores, Día del año, Tarjeta SD, TemporizadorPI (temporizador de posición inicial) y Modos (Modo Automático, Modo Sensor, Modo Emulador y Ubicación inicial).

En la Figura 9 se muestra un diagrama de flujo de `setup()`. Éste sólo se ejecuta una vez al energizar o resetear el Arduino. Al comienzo se declaran todas las variables globales que se van a utilizar tanto en el `setup()`, `loop()` o en cualquier de las funciones anteriormente mencionadas. En la función `setup()` se configuran e inicializan los módulos SD y RTC.



Figura 9: Diagrama de flujo de `setup()`.

En la Figura 10 se exhibe el diagrama de flujo de `loop()`, el cual se detalla a continuación.

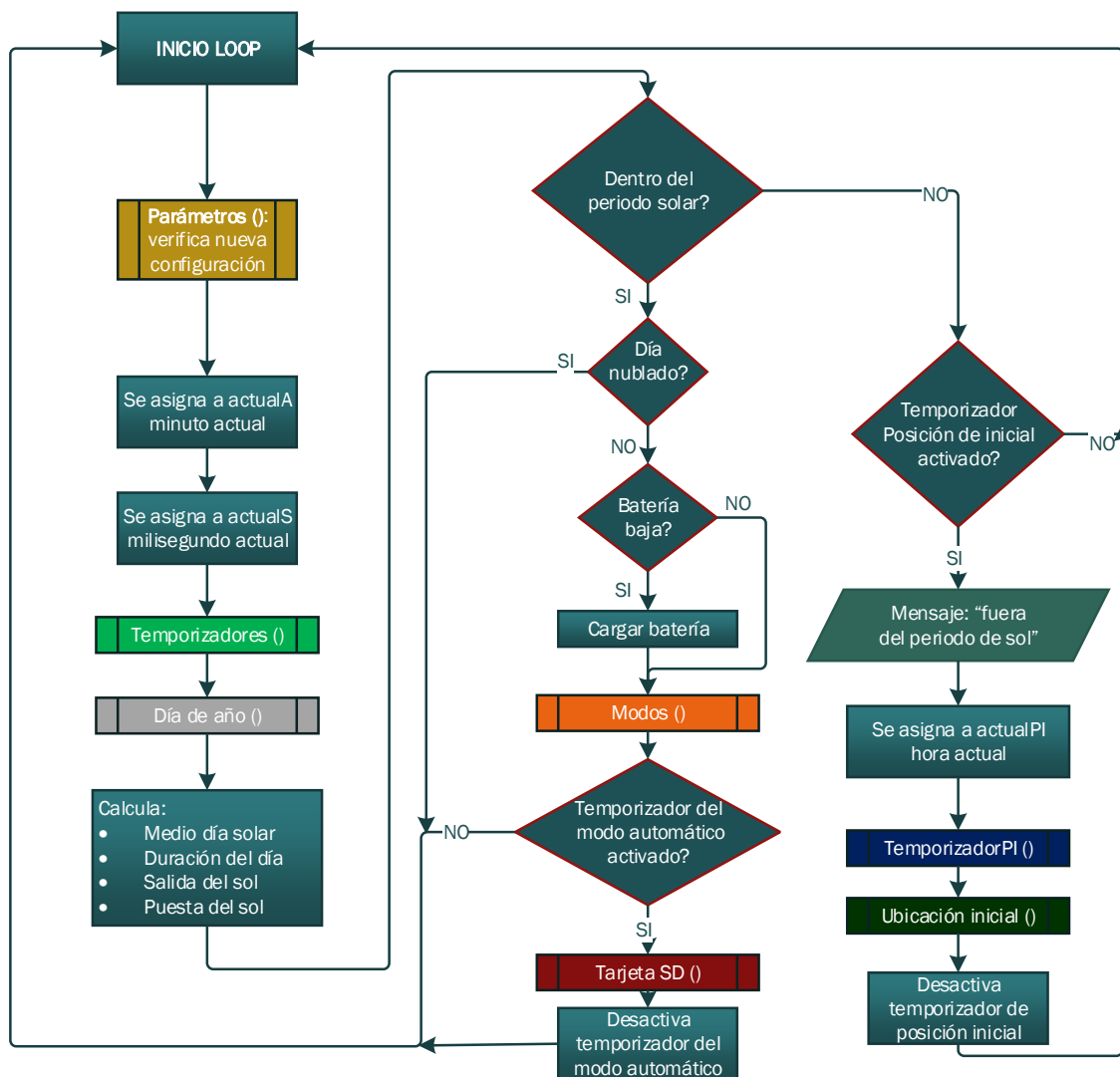


Figura 10: Diagrama de flujo del `loop()`.

Dentro de la función `loop()`, la primera tarea es verificar si el usuario ha enviado alguna nueva configuración, mediante el llamado a la función `parámetros`. También actualiza las variables (*actualA* y *actualS*) de los temporizadores principales y los inicializa, tanto para el modo automático como el modo sensor. A continuación, calcula el día del año mediante la función `Día del año ()`, esto es necesario para el cálculo de la trayectoria. Realiza los cálculos matemáticos de la puesta y salida del Sol de cada día, estos datos son necesarios para obtener el lapso del sol en el horizonte, que en definitiva determina el periodo de trabajo del seguidor solar. Si la hora actual se encuentra en este rango, primero verifica si el día está muy nublado, si es así no realiza ninguna otra acción hasta que se despeje. Si el día está despejado, el siguiente paso es verificar el estado de la batería y si su nivel es bajo comienza la carga de la misma. Una vez comprobado estos dos estados anteriores llama a la función `Modos ()`, para realizar las distintas tareas según el modo configurado (A, S, M, E y O). Si el temporizador del modo automático se encuentra activado realiza la grabación en la tarjeta SD de las lecturas de los paneles solares.

Fuera de este intervalo de tiempo, lleva el seguidor solar a una posición inicial por única vez y se queda a la espera de que comience un nuevo periodo solar (día siguiente).

Interfaz de usuario.

En este caso se decidió realizar como interfaz una aplicación para Smartphone o Tablet que funciona bajo el sistema operativo Android. Para el desarrollo de la aplicación se utilizó App Inventor 2 que es un entorno de desarrollo de aplicaciones para dispositivos Android del Instituto Tecnológico de Massachusetts y de un equipo de Google Education. App Inventor 2 es un lenguaje de programación basado en bloques, y orientado a eventos.

Aprovechando las características actuales de los teléfonos o tablets, se desarrolló una aplicación que crea un menú interactivo para el usuario del seguidor solar.

Dicha aplicación emplea 5 recursos: Sensor de ubicación para obtener la latitud y longitud; Cliente bluetooth para realizar la comunicación entre ambos dispositivos (smartphone y seguidor solar); 2 relojes temporizadores, uno para refrescar continuamente los datos del sensor de ubicación y el otro para borrar cada 10 segundos el monitor; Notificador para informar cuando se ha conectado y desconectado correctamente.

La aplicación toma automáticamente del dispositivo (celular o tablet) los datos necesarios para los cálculos de la trayectoria solar como son sus coordenadas geográficas (latitud y longitud), la fecha y la hora actual.

Como se muestra en la Figura 11a, el usuario puede elegir entre 5 modos de funcionamiento: Automático, realiza cálculos de la trayectoria; Sensor, utiliza sensores LDR para su posicionamiento; Mixto, combina los modos Automático y Sensor; Emulación Solar, simula el movimiento diurno de sol desde la puesta hasta el ocaso; Apagar Seguidor Solar, lo lleva a su posición inicial. Luego el usuario debe presionar “*Enviar a Seguidor Solar*” para que la aplicación envíe las órdenes y parámetros al prototipo y este pueda realizar la tarea solicitada.

Para corroborar que el prototipo esté operando correctamente, la aplicación posee un monitor serie, como se exhibe en la Figura 11 b. El prototipo informa los cálculos y modos de funcionamiento a través de este monitor. El mismo borra sus datos automáticamente cada 10 segundos, pero además posee un botón para borrar manualmente los datos que se reciban.



Figura 11: Modos y Monitor en la aplicación.

Resultados

Diferencia de cálculo entre Planilla de Cálculo y Arduino Mega.

Se realiza una comparación entre los cálculos hechos mediante una Planilla de Cálculo (Excel, Origin, etc.) y los obtenidos con el Arduino Mega 2560, para el ángulo de inclinación o altura solar y para el ángulo azimut. Se tomó como muestra un día aleatorio durante el periodo de Sol, desde el alba hasta el ocaso. En la figura 12 se representa el porcentaje de error de la altura solar calculada a partir de una Planilla de Cálculo y de Arduino Mega, los cuales no superan el 0.5 por ciento.

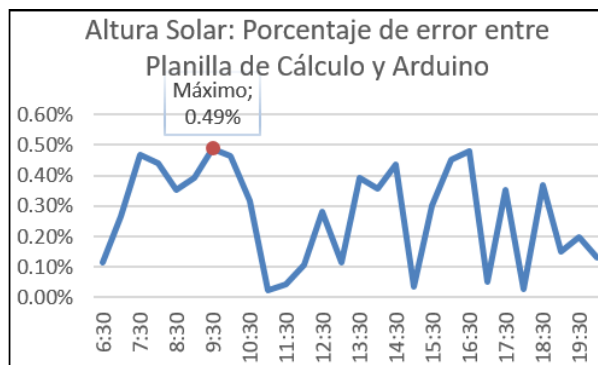


Figura 12: Variación del porcentaje de error de la altura solar entre Planilla de cálculo y Arduino.

En la figura 13 se representa el porcentaje de error del ángulo azimut calculado a partir de una Planilla de Cálculo y de Arduino Mega, los cuales no superan el 0.5 por ciento.

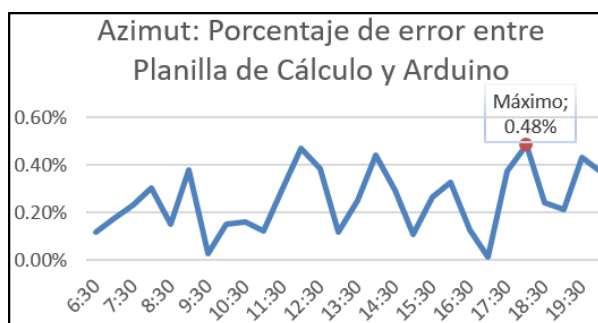


Figura 13: Variación del porcentaje de error azimutal entre Planilla de cálculo y Arduino.

Comparación entre un sistema fijo y este seguidor solar de dos ejes.

Se realizó una comparación entre un sistema fijo con respecto al seguidor solar de este proyecto. Para estas pruebas se montó adicionalmente un panel solar fijo orientado al norte y con una inclinación óptima para San Luis en el período estival. Además, para realizar una comparación correcta, ambos paneles empleados poseen las mismas características.

En la Figura 14 muestra una medición en el modo cálculo Automático de la trayectoria. Se observó mayor diferencia (24% y 30% en el voltaje) al comienzo y al final del día respectivamente, entre el panel con seguidor solar y el fijo. Cerca del mediodía solar, las diferencias entre las lecturas del panel fijo y el móvil con seguidor solar, disminuyen al 8% aproximadamente.

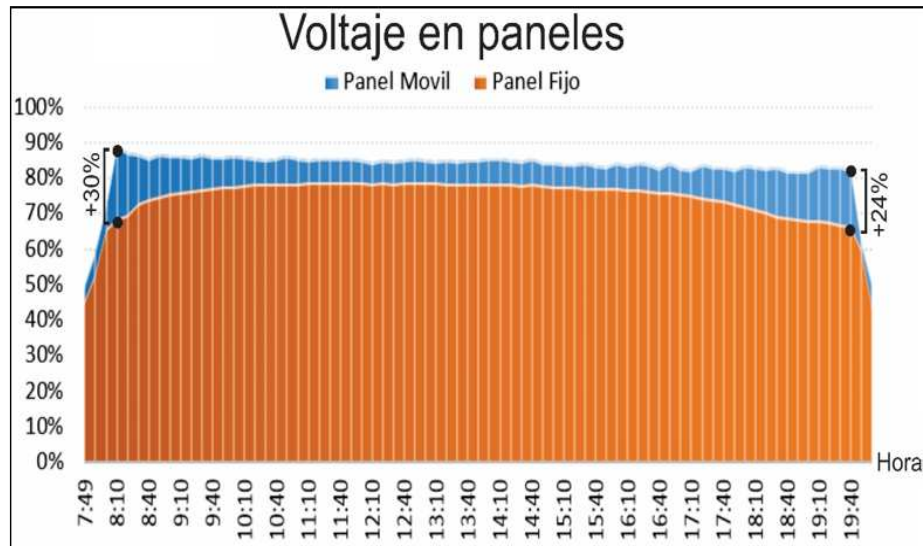


Figura 14: Comparación del voltaje medido en un panel fijo respecto de uno con el seguidor solar en modo A.

CONCLUSIONES

El objetivo del proyecto fue realizar un prototipo que pueda seguir el Sol en forma desatendida. Por medio de una placa Arduino Mega y el software IDE de Arduino, se desarrolló un sistema de seguimiento (hardware + software) que satisface las expectativas, ya que el dispositivo funciona de forma autónoma e independientemente de la ubicación geográfica.

Se puede concluir que el micro controlador de Arduino Mega es capaz de realizar los cálculos de punto flotante con gran precisión y sin inconvenientes.

El programa implementado posee la flexibilidad de adaptarse al sistema mecánico, ya que posibilita cambiar tanto el tipo de servomotor como el soporte mecánico.

Se logró una interfaz de usuario amigable y flexible bajo el sistema operativo Android. De esta manera se realizó un proyecto completo, que abarca desde el control del seguidor solar hasta el almacenamiento de datos. Además, suministra la versatilidad y comodidad de comunicarse con un dispositivo Android de forma inalámbrica.

Los resultados obtenidos abalan que el seguidor solar de dos ejes desarrollado obtiene mejores prestaciones energéticas respecto de un sistema fotovoltaico con paneles fijos. Se prevé instalar cuatro paneles solares idénticos en las instalaciones del Laboratorio de Energía Solar, dos con el seguidor propuesto y dos fijos.

REFERENCIAS

Muhammad Iqbal, "An Introduction to Solar Radiation" - Academic Press, 1983.

F. Grassi, G.M Salum y J.C. Ansaldi, "Diseño de un Seguidor Solar de dos ejes parametrizado por PC", ASADES 2012.

Andrés Escobar Mejía, Holguín Londono y Osorio, “Diseño e implementación de un seguidor solar para la optimización de un sistema fotovoltaico”, Scientia et Technica Año XVI, No 44, abril de 2010. Universidad Tecnológica de Pereira. ISSN 0122-1701.

Perez García M. A., Anton Álvarez J. C., Campo Rodríguez J. C., Grillo Ortega G. J., “Instrumentación Electrónica”, Ed. THOMSON, 2004.

SanDisk Corporation, “SanDisk Secure Digital Card” - Versión 1.9. Nota de Aplicación.

Nathan Waddington y Russell Taylor, “Arduino & Open Source Design”.

Alasdair Allan, “Make: Bluetooth. Bluetooth LE Projects with Arduino, Raspberry Pi, and Smartphones”, maker media, 2015

Sitio oficial de la plataforma Arduino. <https://www.arduino.cc/>

Ing. Noel M. Toranzo, Dra. Ania L. Cervantes, Ing. Leandro L. Oro Carralero, Lic. Jorge Bonzon H., Ing. Orlando E. Costa, “Seguidor Solar, optimizando el aprovechamiento de la energía solar”, Universidad de Oriente, Santiago de Cuba, Cuba, Energética vol.36 no.2 LaHabana Mayo. Ago. 2015.

SUMMARY: The objective of the present work is to develop a two axis solar tracker to be applied to photovoltaic panels. This application allows the panel to increase its production by 20 to 30%, depending on the month and the time of day. The developed system contemplates 5 modes of operation. An Arduino Mega board was used as it satisfies the required memory and computing power requirements. LDR sensors were used for positioning; In addition, the system contemplates the possibility of connecting by cellular telephony. For the scale model, a Tower Pro model SG90 servomotors are used for their low energy requirements that allow modularizing the system. Simulations were carried out where the error was less than 1% for the positioning and the increase of the energy produced within the expected values. Improvements are proposed in the software in order to make it more accessible.

Keywords: solar tracker, photovoltaic panels, solar energy, Arduino, technology.