

Desarrollo de un sistema web para el monitoreo en tiempo real de variables de generación fotovoltaica en la Facultad de Ingeniería UNaM

Daniel F. Bejarano ^{a*}, Ezequiel A. Burgos ^a, Fernando Botterón ^(a,b), Lucas Horianski ^a

^a Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.

^b GIDE, FI-UNaM, Oberá, Misiones, Argentina.

e-mails: danielbejar42@gmail.com, burgosezequieladonai@gmail.com, botteron@gmail.com,
lucas.horianski@fio.unam.edu.ar

Resumen

Este trabajo se realiza en el marco de la materia Proyecto y Diseño Electrónico, correspondiente al último año de la carrera Ingeniería Electrónica y trata del diseño y construcción de un sistema de monitoreo de un generador fotovoltaico conectado a la red eléctrica, instalado de la Facultad de Ingeniería UNaM de la ciudad de Oberá. La motivación del proyecto es proveer una herramienta que facilite la visualización y el relevamiento de las variables de generación y conversión entregadas por los paneles fotovoltaicos (PFV) y el convertidor electrónico de potencia conectado en la red de la facultad. Para ello, se diseñó un sistema de monitoreo por aplicación web para ver valores de tensiones y corrientes de continua y de alterna, potencia activa, frecuencia e irradiación solar en tiempo real de manera remota. El sistema posee una minicomputadora Raspberry Pi Zero 2W, que ofrece conexión WIFI y permite realizar la conexión web para enviar datos hacia un servidor que se encarga de realizar el procesamiento de la información recibida y así poder visualizarla. Además de ello se tiene un Arduino Mega (ATMEL 2560) con una interfaz de acondicionamiento de señales, reloj en tiempo real y registro de datos en tarjeta SD con comunicación Bluetooth que recopila información de los paneles solares (tensión, corriente e irradiancia) y son enviados a la minicomputadora. Toda la información recopilada y procesada del generador y el convertidor, son convenientemente presentados mediante una interfaz gráfica, la cual podrá ser consultada desde la Web de la facultad.

Palabras Clave –Monitorización, Paneles Solares, Arduino, Raspberry Pi, Tensión, Corriente, Potencia.

1. Introducción

En la actualidad la obtención de energía eléctrica a través de paneles solares se incrementa constantemente, fomentando así el uso de las energías renovables, disminuyendo los costos energéticos de los hogares, comercios, instituciones e industrias. Los paneles solares generan energía eléctrica a través de celdas fotovoltaicas que transforman la energía solar en una tensión continua, esta tensión debe ser convertida en una tensión alterna para poder ser utilizadas por electrodomésticos hogareños. Esta transformación requiere de un convertidor electrónico de potencia, comúnmente denominado inversor, que recibe una tensión continua de los paneles o de un conjunto de baterías (si el sistema de generación consta de almacenamiento de energía) y la transforman en tensión alterna como se mencionó anteriormente. En general, los inversores utilizados para la tarea descrita, poseen la característica de registro y visualización, haciendo posible que el usuario disponga de las variables de generación y conversión tales como tensiones (alternas y continuas), potencia, corrientes (alternas y continuas), energía generada, frecuencia entre

otras. Es importante señalar, que el inversor que se utiliza en este trabajo, es del tipo on-grid, el cual inyecta a la red eléctrica toda la potencia que le entregan los paneles fotovoltaicos. Con los datos que brinda el inversor, es posible realizar diferentes tipos de análisis, como en que horarios del día se genera mayor energía lo que puede servir como una medida de la producción eléctrica a partir de una fuente renovable que se tiene en un establecimiento.

Atendiendo a esto, se propone el diseño de un sistema de monitoreo de variables de generación y conversión de los paneles solares ubicados en la Facultad de Ingeniería de Oberá, perteneciente a la Universidad Nacional de Misiones. El sistema consta de una unidad central formada por una micro computadora Raspberry Pi Zero 2W, el cual se encargará de recibir información de un Arduino montado sobre una interfaz de acondicionamiento de señales y de un inversor AEG2800 on-grid que inyecta a la red toda la potencia entregada por los paneles FV. Tanto el generador fotovoltaico como el inversor, están ubicados en el laboratorio de electrónica en planta alta de la Facultad de Ingeniería. El Arduino toma los datos de tensión y corriente continua directamente de los paneles solares, calcula la potencia y además de ello obtiene los niveles de irradiancia del lugar donde se encuentra ubicados los paneles solares, estos datos se lo envía a la unidad central de manera inalámbrica para poder aislar galvánicamente una unidad de la otra. Por otro lado, el inversor AEG PV-2800, también envía información a la unidad central través de un cable RS232 a sus puertos Rx y Tx del mismo mediante un adaptador. La información recibida por ambos dispositivos en la unidad central es procesada y se eleva a una nube digital, en el cual se muestran los valores a través de gráficos analógicos, digitales e histogramas. De esta manera se obtiene una comparación en tiempo real de ambas mediciones realizadas.

2. Sistema de Monitoreo Propuesto

En la Figura 1 se presenta el diagrama de bloques del sistema de monitoreo, que ofrece una representación visual clara de su estructura y funcionalidad. En el centro del sistema se encuentra la minicomputadora Raspberry Pi Zero 2W este dispositivo recibe información del Arduino (tensión, corriente e irradiancia de los paneles solares) de manera inalámbrica mediante un transceptor bluetooth y del Inversor PV2800 mediante un cable RS232. Los datos recibidos son procesados y enviados a una nube digital, utilizando el entorno de programación Node-RED, permitiendo así poder acceder a ellos mediante cualquier dispositivo con acceso a internet.

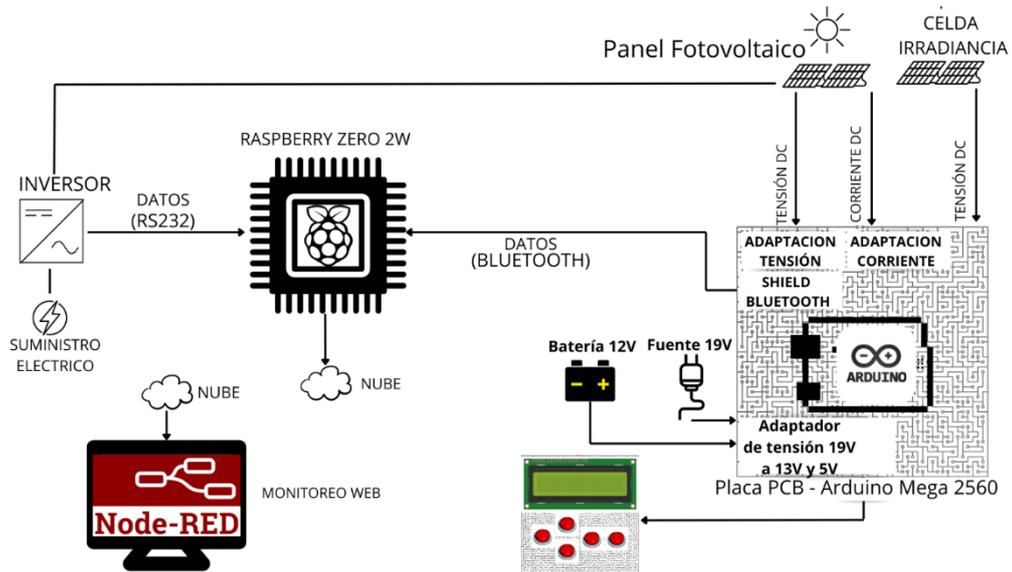


Fig. 1. Sistema de monitoreo: Esquema General.

También se observa en la Figura 1, que el sistema embebido formado por la interfaz y el Arduino consta de un sistema de energía de respaldo, que consiste en una batería de 12 V. Esta batería es alimentada por una fuente continua de 19V, que a través de un regulador disminuye la tensión de entrada a 13V para poder cargarse, seguido a esto se tiene otro regulador de 5V para alimentar el Arduino junto con los demás componentes de la interfaz que realiza el acondicionamiento de las señales del generador adquiridas por el Arduino.

2.1 Análisis de la placa PCB

En la Fig. 2 se muestran las partes que conforman el sistema embebido basado en Arduino, en donde además de las entradas para la toma de datos de los paneles, se observa una entrada de alimentación de batería para obtener un sistema ininterrumpido en caso de falta de energía en la red eléctrica.

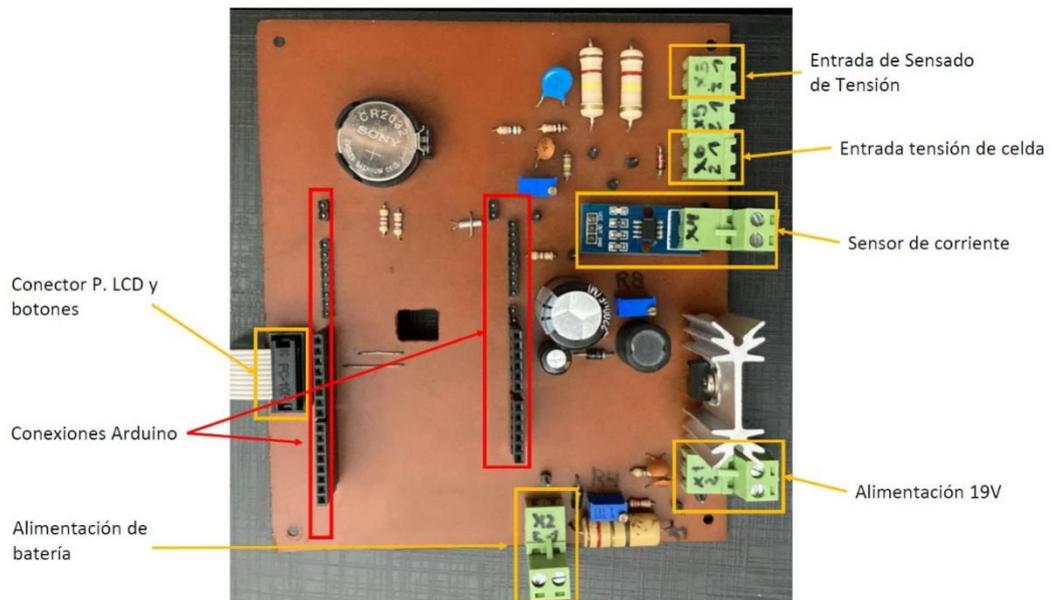


Fig. 2. Partes que conforman la placa PCB del Arduino Mega 2560.

En la Fig. 3 se muestra el esquema eléctrico del circuito de alimentación, el cual consta de 2 reguladores de tensión. El primero reduce la tensión proveniente de la fuente de 19V/2A a 13V mediante un regulador LM317 que baja la tensión de entrada a 13,8V para poder cargar la batería, luego se incorpora un regulador tipo Step Down basado en un LM2575 tipo switching, el cual disminuye el voltaje a 5V para alimentar el Arduino y demás componentes de la placa PCB.

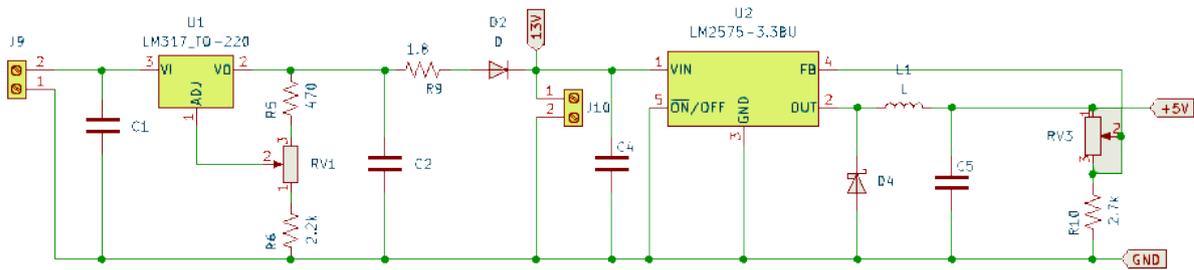


Fig. 3. Alimentación de la placa PCB.

El esquema eléctrico para el sensado de la tensión de los paneles FV, se muestra en la Fig. 4. La tensión CC que ingresa por los bornes 1 y 2, pasa por una red de atenuación y filtrado que acondicionan esta tensión a los niveles del convertor AD del Arduino Mega. El circuito posee un diodo limitador para evitar que la tensión supere el valor máximo de 5V soportado por la entrada del ADC y posee un varistor en la parte de alta tensión que limita las posibles sobretensiones que puedan presentarse.

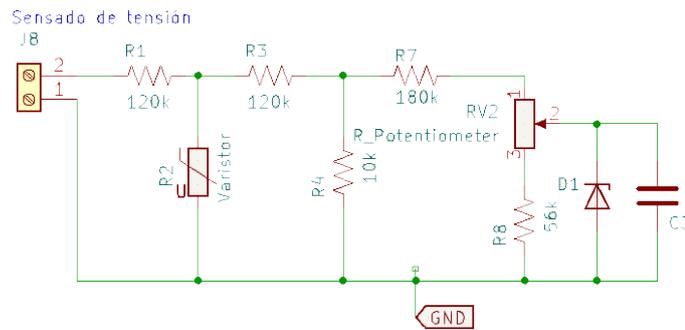


Fig. 4. Circuito adaptador para sensado de tensión.

Para medir la corriente que entregan los paneles FV se utiliza un sensor de corriente de efecto Hall, el ACS712ELCTR-30A-T de la compañía Allegro que se muestra en la Fig.5. Este dispositivo es capaz de medir en un rango de ± 30 A.



Fig. 5. Sensor de corriente ACS712.

Por último, en la Fig.6, se presenta el circuito de acondicionamiento para el sensado de la irradiancia, en donde se dispone de un diodo Zener de 5,1V de protección. El sensor de irradiancia entrega valores de tensión que van desde los 0V hasta los 0,6V.

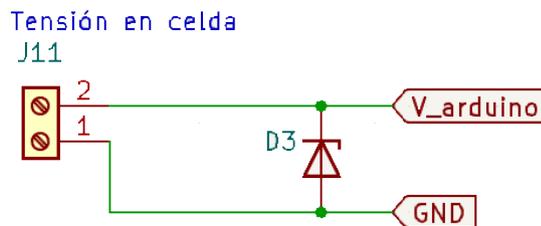


Fig. 6. Circuito de sensado de irradiancia.

2.2 Código Arduino:

El Arduino consta de un menú principal y un menú secundario. El diagrama de flujo del menú principal se muestra en la Fig.7, en el cual se muestra el nombre del proyecto y los integrantes por aproximadamente 1,5 segundos. Seguido de esto, se muestran todos los valores sensados por 4 segundos, además se leen los valores de las entradas analógicas convirtiéndolas en valores

digitales y se realiza el envío de datos hacia la unidad central. Para finalizar, se leen los estados de los pulsadores y si no se presiona ningún botón, la pantalla se apaga para ahorrar energía.

Al presionar el botón “U”, se enciende el display y se accede al menú secundario donde se encuentra la visualización individual de la corriente, al presionar el botón “L” o el botón “R” se pueden visualizar las demás variables como tensión, potencia o irradiancia. Además, en cada visualización individual se realiza la lectura de las entradas analógicas y el envío de datos a la unidad central. Para salir del modo de visualización individual se debe presionar el botón “D” y así se regresa al menú principal donde nuevamente se mostrarán todas las variables sensadas. Si se presiona nuevamente el botón “U” vuelve a la última variable individual que fue visualizada. El código que comprende ambos menús está realizado en lenguaje C y el diagrama de flujo de la unidad secundaria se muestra en la Fig.8.

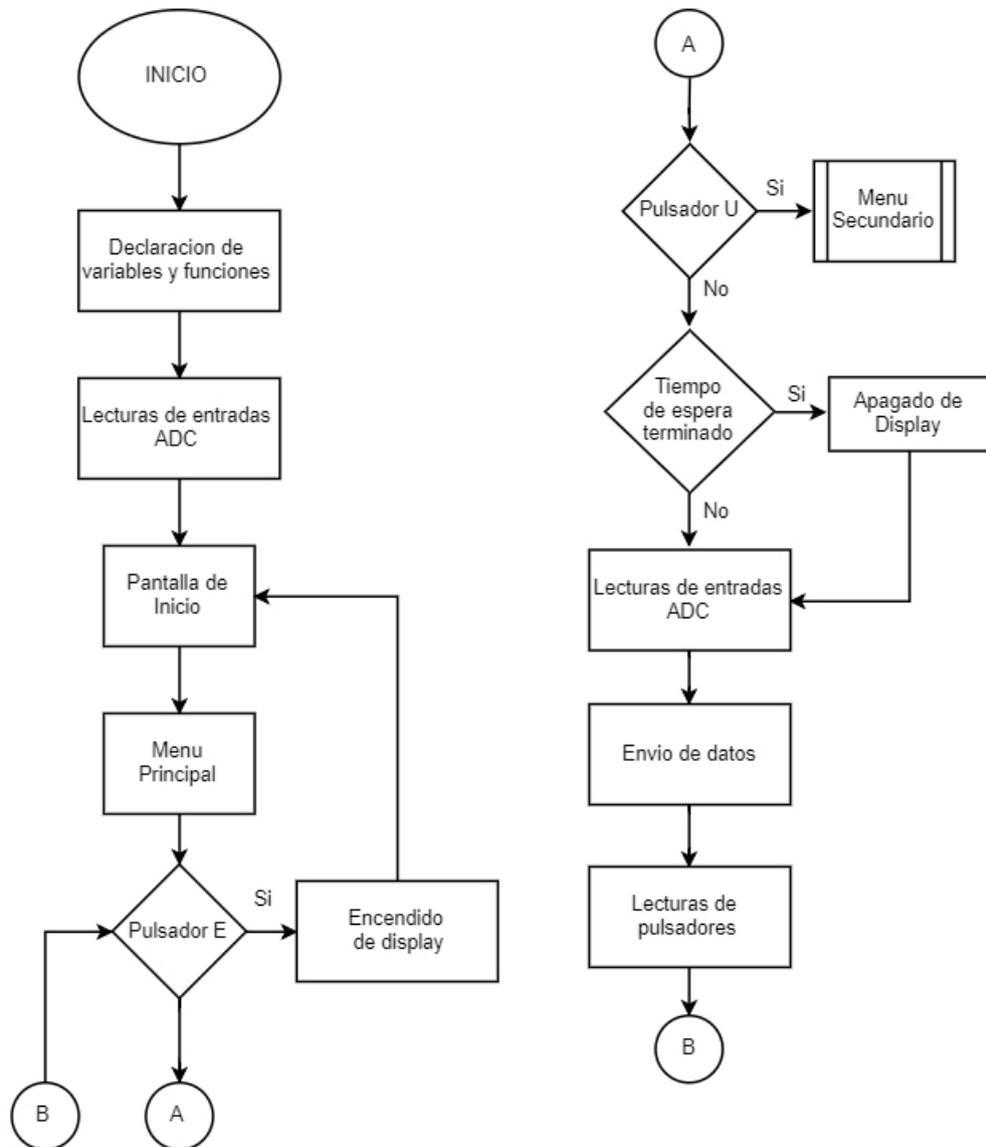


Fig. 7. Diagrama de flujo del menú principal.

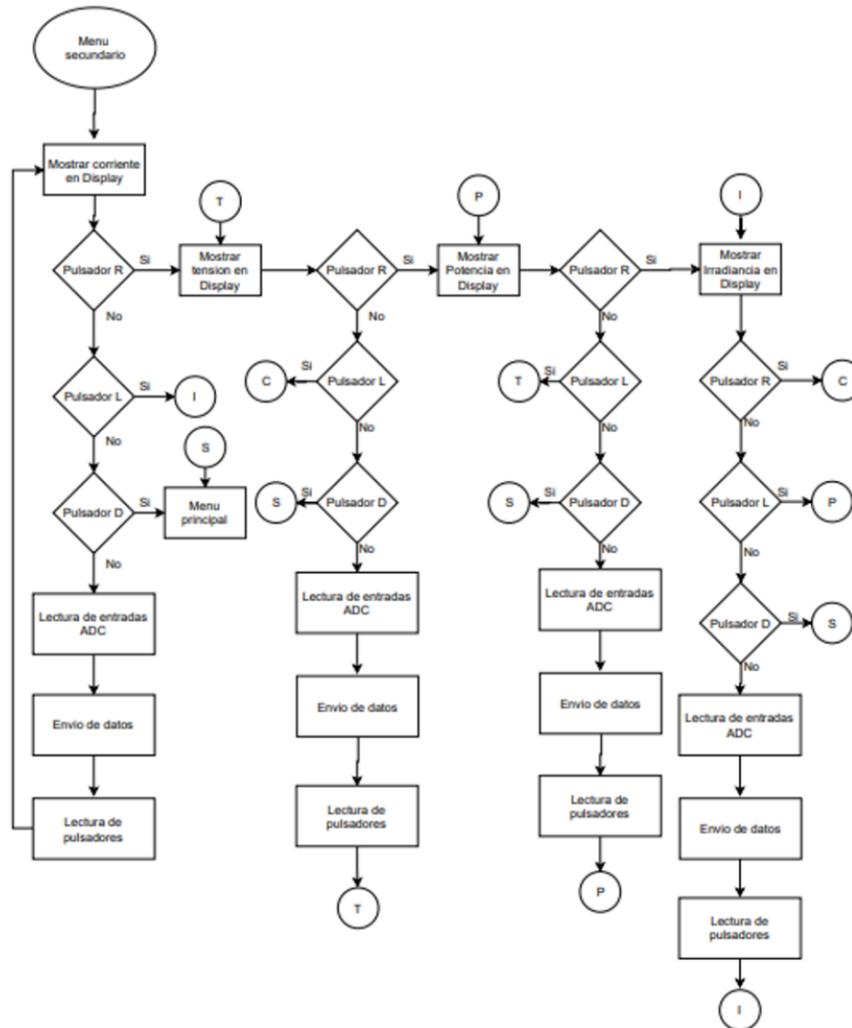


Fig. 8. Diagrama de flujo del menú secundario.

2.3 Decodificado de trama serial del inversor AEG PV-2800.

El inversor AEG PV-2800 posee un software propietario el cual permite instalarse en un sistema operativo Windows, este software permite visualizar datos de tensión, corriente, potencia, etc.

Con el objetivo de poder presentar los datos suministrados por el inversor y el sistema de adquisición basado en Arduino, de forma flexible y conveniente, se decidió no utilizar el software propietario de AEG. En este sentido, se hizo necesario analizar y decodificar los datos que envía el inversor a través del puerto serial mediante el protocolo RS232. Mediante el software “Serial Port Monitor”, se capturaron las tramas seriales leídas desde el puerto de comunicación serial de la PC, como se observa en la Fig. 9. Se analizaron los datos binarios convertidos a decimales y se los compararon con los datos obtenidos del software AEG.

```
[29/04/2024 14:43:44] Read data (COM1)
aa aa 00 01 01 00 01 82 28 01 ac 0c d8 00 0e 08
d3 13 8c 01 29 ff ff 00 01 f2 94 00 00 45 0b 00
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 09 1a
```

Fig. 9. Trama serial capturada con el software “Serial Port Monitor”.

Time	Temp(°C)	Vpv(V)	Iac(A)	Vac(V)	Fac(Hz)	Pac(W)	Zac(mOhm)	E-Total(kWh)	E-Today(kWh)	h-Total(h)	Vpv1(V)	Iac1(A)	Vac1(V)
2024/04/29 14:43:45	42.8	328.8	1.4	225.9	50.04	297	NA	12763.6	0.00	17675	328.8	1.4	225.9

Fig. 10. Datos obtenidos del software AEG.

A continuación, se ilustran las posiciones de los valores hexadecimales, junto con sus equivalencias en binario, decimal y correspondiente valor asociado de parámetro.

inicio pos	hex	binario	int 16 sin signo	valor asociado	Unidad
9 y 10	01 ac	0000 0001 1010 1100	428	42,8	Temp(°C)
11 y 12	0c d8	0000 1100 1101 1000	3288	328,8	Vpv(V)
13 y 14	00 0e	0000 0000 0000 1110	14	1,4	Iac(A)
15 y 16	08 d3	0000 1000 1101 0011	2259	225,9	Vac(V)
17 y 18	13 8c	0001 0011 1000 1100	5004	50,04	Fac(Hz)
19 y 20	01 29	0000 0001 0010 1001	297	297	Pac(W)
21 y 22	ff ff	11111111 11111111			
23, 24, 25 y 26	00 01 f2 94	0000 0000 0000 0001 1111 0010 1001 0100	127636	12.763,60	E-Total(kWh)
27 y 28	00 00	00000000 00000000			
29 y 30	45 0b	0100 0101 0000 1011	17675	17.675	h-Total(h)
31 y 32	00 01	00000000 00000001			
33 a 48	ceros				
49 y 50	09 1a	00001001 00011010	2330		

Fig. 11. Resultado de la decodificación.

2.4 Comunicación de la Raspberry Pi con Inversor PV2800.

Una vez decodificado el mensaje enviado por el AEG PV-2800, con el “Serial Port Monitor” se obtuvieron las tramas del inicio de comunicación entre el inversor y el software instalado en la PC. Esto se realizó para poder comunicar la Raspberry y el inversor sin necesidad de instalar el software propietario de la marca AEG. En las Fig.12, se observan enumeradas las diferentes tramas para iniciar el envío y recepción de información. Se identificó el nombre del dispositivo en la trama 2, el modelo y marca en la 3, la recepción del abecedario en la 4, entre otros.

Para poder replicar el mensaje se utilizó un IDE llamado Geany, junto con un script de Python que realiza el envío de los 7 mensajes principales y 2 mensajes secundarios. En la Fig.13 se puede observar la terminal del IDE Geany, en donde aparecen los mismos mensajes enviados y recibidos.

De esta manera, la Raspberry Pi puede recibir información del inversor a través de un adaptador de RS232 a micro USB, sin necesidad de instalar el software propietario del inversor y así obtener información de todas las variables que ofrece el inversor.


```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Enviado1: aaaa010000000004000159aaaa010000000004000159aaaa010000000004000159aaaa
010000000004000159aaaa01000000000000155
datos recibidos: aaaa0000010000800afffffffffffffffffffff0bd5
Enviado2: aaaa0100000000010bfffffffffffffffffff010b58
datos recibidos: aaaa000101000081010601de
Enviado7: aaaa010000000000000155
Enviado8: aaaa010000010102000159
Enviado7: aaaa010000000000000155
Enviado8: aaaa010000010102000159
datos recibidos: aaaa0001010001822801580b7300010918138c0006ffff000201e0000047830
0010000000000000000000000000000000000074b
Enviado7: aaaa010000000000000155
Enviado8: aaaa010000010102000159
datos recibidos: aaaa0001010001822801590b3f00020922138c0005ffff000201e0000047830
00100000000000000000000000000000000000722
Enviado7: aaaa010000000000000155
Enviado8: aaaa010000010102000159
datos recibidos: aaaa0001010001822801580b2e00010918138b0006ffff000201e0000047830
0010000000000000000000000000000000000705
Enviado7: aaaa010000000000000155
Enviado8: aaaa010000010102000159
datos recibidos: aaaa0001010001822801580b0f00020918138b0005ffff000201e0000047830
00100000000000000000000000000000000006e6
Enviado7: aaaa010000000000000155
```

Fig. 14. Tramas de sincronización Raspberry-Inversor.

2.5 Entorno Node-RED

Node-RED es una herramienta de desarrollo basada en flujo para programación visual desarrollada originalmente por IBM para conectar dispositivos de hardware, API y servicios en línea como parte de la Internet de las cosas. A su vez, proporciona un editor de flujo basado en navegador web que se puede utilizar para crear funciones de JavaScript. Los elementos de las aplicaciones se pueden guardar o compartir para su reutilización. Los flujos creados en Node-RED se almacenan mediante JSON y permite comunicación mediante MQTT, Modbus, entre otros. La programación o configuración del entorno Node-RED se realiza a través de flujos en formato JSON o a través de nodos creados por su interfaz visual. En la Fig. 15 se muestran algunos de los nodos utilizados para poder monitorizar el sistema y en la Fig. 16 la estructura de nodos interconectados resultante hasta el momento.

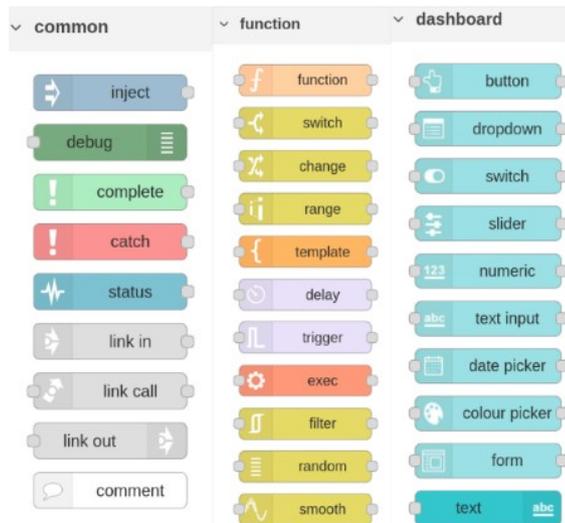


Fig. 15. Nodos utilizados en Node-RED.

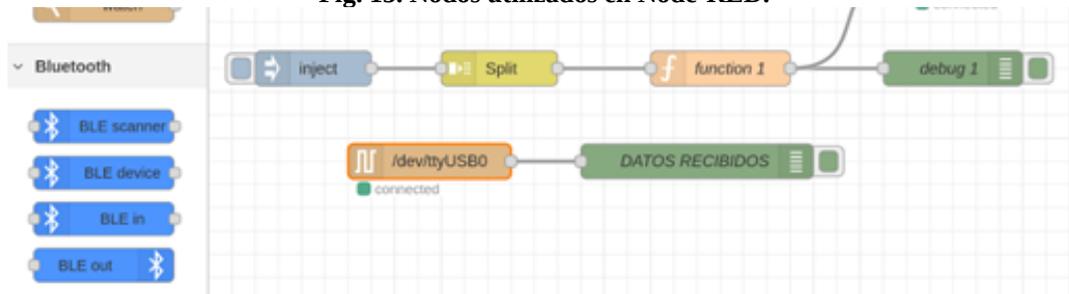


Fig. 16. Estructura de interconexión de nodos.

3. Resultados

3.1. Mediciones obtenidas con el Arduino Mega 2560.

Se comprobó el correcto funcionamiento del sistema embebido basado en el Arduino conectándolo a un valor conocido de corriente (1 A). Se presenta en la Fig. 17 el valor mostrado en el display LCD.



Fig. 17. Verificación del funcionamiento del sensado de corriente.

3.1. Interfaz visual en Node-RED

Mediante los nodos Dashboard mostrados en la Fig. 18 se realizó un boceto en el cual se muestran en la interfaz visual el orden de los valores sensados, en la cual se podrán comparar los datos recibidos del Arduino y del inversor.



Fig. 18. Nodos utilizados en Node-RED.

4. Conclusiones

Mediante este trabajo se logró con éxito la puesta en marcha del sistema embebido basado en Arduino existente en la instalación, tomar las medidas adecuadas y realizar la comunicación con la unidad central. Además de ello, se consiguió la correcta comunicación entre el inversor AEG PV-2800 con la unidad central. Se profundizó el aprendizaje de lenguajes de programación JavaScript, C y Python; así como entornos de comunicación destinados al Internet of Things (IoT) como Node-RED. Por otro lado, el desarrollo del conexionado de ambas unidades permitió poner en juego conocimientos relacionados con regulación de tensión, medición de corriente y tensión, así como el diseño de circuitos con microcontroladores y la programación asociada a estos.

Este proyecto también permitió la observación, análisis, funcionamiento y uso de dispositivos de transmisión mediante protocolos de comunicación como RS232. Su uso puede ser aplicado para transmisiones de datos, manejo de dispositivos, control a distancia, etc. Para finalizar, el trabajo requirió una investigación más profunda de los sistemas de comunicaciones basados en microcontroladores y dispositivos externos, así como una observación más minuciosa de la programación asociada a estos y el uso de librerías realizadas por terceros, pero adaptadas al controlador utilizado.

Referencias

- [1] Wyne Tomasi - Sistemas de Comunicaciones Electrónicas, 4 Edición. Editorial Pearson (2013)
- [2] Sistema de Comunicaciones - Jordi Mayné, Silica, 2009.

- [3] Robert L Boylestad-Louis Nashelsky. Teoría de los circuitos y dispositivos electrónicos. Editorial Pearson Educación, México, 2009.
- [4] Hoja de datos de LM2575.
Enlace:
<https://pdf1.alldatasheet.com/datasheet-pdf/view/3052/MOTOROLA/LM2575.html>
- [5] Hoja de datos de LM317:
Enlace:
<https://pdf1.alldatasheet.com/datasheet-pdf/view/1147586/ARTSCHIP/LM317.html>
- [6] Hoja de datos de ACS712:
Enlace:
https://drive.google.com/file/d/14Hm4Z91N9vIwvq8QjMAG3JmNM_XQWcM8/view
- [7] Node-RED Documentación:
Enlace: <https://nodered.org/docs/user-guide/>
- [8] Hoja de datos Arduino Mega 2560.
Enlace:
<https://pdf1.alldatasheet.com/datasheet-pdf/view/897466/ATMEL/MEGA2560.html>
- [9] Hoja de datos LCD: Enlace:
<https://datasheetspdf.com/pdf-file/519148/CA/LCD-1602A/1>.