

EQUIPO DE INTEROPERABILIDAD PARA INVERSORES FOTOVOLTAICOS

Mg. Ing. Marcos Politi ⁽¹⁾, Jorge Andrés Niño ⁽¹⁾, Maximiliano Gulfo ⁽¹⁾, Dr. Héctor Laiz ⁽²⁾

mpoliti@inti.gob.ar

(1) Departamento de Gestión de Sistemas Integrados, Centro de Energías Renovables - INTI

(2) Gerencia de Metrología - INTI

Palabras Clave: LoRa, MODBUS, ATmega328, Ethernet, Interoperabilidad, IoT/IoE.

INTRODUCCIÓN

Los inversores fotovoltaicos tienen abiertos sus puertos Modbus [1], para poder consultar posiciones de memoria, en las cuales se puede acceder a distintos parámetros de generación fotovoltaica. Se confeccionó un equipo intérprete de códigos *MODBUS Ethernet*, constituido por un sistema electrónico, cuyos componentes más importantes son: un microcontrolador de 8 bits y una etapa con conectividad ethernet, que le permite comunicarse con los inversores fotovoltaicos. Por otro lado, es necesario concatenar toda la información de estos inversores y administrarla dentro de una misma plataforma, para tal fin, es necesario incorporar una nueva etapa. Al ser equipos para instalaciones de generación distribuida, estas podrían estar separadas kilómetros entre sí. Debido a esto, se optó por la incorporación de una etapa de comunicación de radiofrecuencia de largo alcance, en este caso, implementada a través de LoRa [2], con lo cual al hardware se le agrega una etapa de conectividad de esta naturaleza.

ESTRUCTURA GENERAL

a. Estructura de Gateway

La estructura del sistema consiste en un *Gateway* (GTW) y una serie de nodos para cada inversor fotovoltaico. Se desarrolló de manera tal que el GTW y el Nodo sean reversibles (excepto en el código), es decir que puedan utilizarse tanto conectado a los inversores fotovoltaicos como a un router.

b. Nodo Lora

La implementación del nodo LoRa es idéntica en hardware y, utilizando el código adecuado, nos permite comunicarnos con el inversor.

HARDWARE DEL EQUIPO

El *hardware* del dispositivo fue diseñado basado en un ATmega328 como procesador central, un adaptador SPI-ETH W5100 y un módulo LoRa

RFM95. A continuación, se detallan los componentes de cada parte del circuito [4].

a. Fuente de alimentación

Se empleó para la fuente de alimentación un regulador *step-down* LM2576 [3], configurado para entregar 5V alimentándolo con una fuente externa de 9-12V. Además, se agregaron dos reguladores de 3.3V para poder suministrar la corriente necesaria al circuito de control y para las comunicaciones.

b. Microcontrolador y Chip LoRa

Se conectó al ATmega328 un buzzer de alarma, una serie de conectores exteriores para manipular circuitos externos de ser necesario, un indicador LED de funcionamiento del circuito de control y un botón de reset. A su vez se conectó el LoRa al puerto SPI (el mismo que ETH) y se administra mediante un CS diferente.

c. Circuito de Ethernet

El circuito de Ethernet consiste en el adaptador SPI-ETH W5100, el cual recibe la información de configuración de la Red desde el ATmega328 a través del puerto SPI y se encarga de llevar a cabo la máquina de estados de una conexión Ethernet. Esta sección consta del conector RJ45 con su correspondiente circuito de adaptación de señal. En el controlador se sumaron varios diodos LED que permiten ver el estado de este chip e identificar el estado de la conexión. El chip se conecta al puerto SPI que posee el procesador con lo cual comparte líneas de comunicación con el módulo LoRa, lo que obliga a implementar un programa que administre las operaciones una por una y no en paralelo



Figura 1: PCB terminado, montado en el inversor.

IMPLEMENTACIÓN DEL CÓDIGO

Para la implementación del código se emplearon las bibliotecas “*MsgModbus*” [5], públicas en internet. Se incluyó una biblioteca propia desarrollada con los registros de un inversor SMA, la cual se usa para acceder a los parámetros y la configuración del dispositivo. A su vez se emplearon las bibliotecas para Ethernet y LoRa [6] de Arduino, con ciertas modificaciones, las cuales se detallarán a continuación, así como el algoritmo general:

a. Modificación de la biblioteca LoRa

Se modificó la biblioteca LoRa para incluir un mecanismo que dejara un *flag* indicador de que se han recibido datos y que se han guardado en el buffer, de tal manera que la interrupción del RFM95 no altere el flujo del algoritmo si este se encuentra operando el SPI con el W5100.

Primero, se chequea que no hayan entrado datos mediante la verificación de la interrupción *IRQ_in*. De ser así, se cargan los mismos desde el buffer del RFM95 al buffer propio del microcontrolador. Luego, mientras tenemos el chip seleccionado, enviamos los últimos datos cargados en la lectura del inversor mediante LoRa, o enviamos la petición de dato al Nodo mediante LoRa. Finalmente, deshabilitamos el LoRa y habilitamos el W5100, operando según corresponda, subiendo datos (*Gateway*) o pidiendo datos al inversor (Nodo)

Este mecanismo se implementó modificando la rutina de interrupción de tal manera que generara un *flag* en un objeto que puede ser modificado por el programa principal, funcionando como una cola de interrupciones.

b. Implementación de modo de operación en Nodo y Gateway

Se implementó una función mediante el Timer de ATmega328 para poder direccionar cada 10 segundos al *Modbus Ethernet* del inversor. Durante los intervalos, se apaga el CS del W5100 y se comunica con LoRa para enviar datos. Esta operación corresponde al Nodo. En el mismo se implementa también el algoritmo de lectura de LoRa con el objetivo de poder recibir peticiones del GTW en futuros desarrollos. En el caso del GTW se implementó el método para leer el estado de la interrupción de LoRa. Luego cada 3 segundos se envían los datos mediante protocolo MQTT a través de Ethernet.

Finalmente, se llevó a cabo el Sistema Nodo-Gateway sobre un inversor SMA. El GTW fue ubicado a 100 metros del nodo, en un entorno protegido, con acceso a internet. Se implementó el servicio sobre EC2 de Amazon Web Service, la cual fue delegada por DNS a smartgrid.ar, donde se instaló un paquete de MQTT para la administración de los datos, NodeRed para la visualización, y MySQL para el guardado de los

mismos. En la plataforma fue posible observar la evolución diaria de los sistemas solares.



Figura 2: Dashboard mostrando los datos de tres nodos.

RESULTADOS Y CONCLUSIONES

La implementación de Modbus sobre LoRa permite que la información de los equipos eléctricos pueda llegar a destino a pesar de eventuales daños en la línea eléctrica. La solución implementada es de bajo costo y permite grandes alcances en instalaciones fotovoltaicas con una tecnología relativamente simple. Por otro lado, su implementación a mediana escala permitiría la administración sencilla y eficaz de redes inteligentes.

Este sistema ejecutado sobre servicios *Cloud*, permite a cooperativas y distribuidoras despreocuparse por el hardware en sus instalaciones, a su vez permitiendo mejorar el desempeño del mismo instalando a futuro Python y ejecutando rutinas de inteligencia artificial. De esta manera, se podrá incorporar un cierto grado de inteligencia al dispositivo, con el objetivo de que sea capaz de autogestionarse, maximizando la eficiencia con que se gestiona la energía eléctrica generada.

AGRADECIMIENTOS

Mención especial a Marianela Bornancín, Patricia A. Leonardi, Valeria A. Saladino, por su gestión administrativa y acompañamiento durante el proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- [1] “*Modbus Messaging On TCP/IP Implementation Guide V1.0b*” <http://www.Modbus-IDA.org>. Oct 2006
- [2] LoRa Alliance (2020, Junio), [Online]. Disponible: <https://lora-alliance.org/about-lorawan>
- [3] Switcher, Simple. “LM2576/LM2576HV Series SIMPLE SWITCHER ® 3A Step-Down Voltage Regulator.” (1999).
- [4] “*Chip W5100*” <https://www.wiznet.io/product-item/w5100/>. ©2011 WIZnet Co., All Rights Reserved
- [5] “*Modbus Library*” <https://myarduinooprojects.com/modbus> ©2011, v0.1.1

[6] "T SHIELD EMGING LORA 915MHZ V1.1 – TEST"
<http://emging.com.ar/tutorial/shield-emging>