

Control remoto para banco de carga en el laboratorio Unigrid

Mauricio Pacheco, Javier Rodriguez and John Solano

{apmauricio, ajavier, ortizmj}@uninorte.edu.co

Abstract— This paper presents the design, construction and evaluation of a control circuit to perform the control and change of resistance, inductance and capacitance values of modules belonging to the Unigrid microgrid, and its integration to a SCADA system for monitoring, visualization and control of the selected values in real time through the Modbus RS-485 communication protocol.

Index Terms- SCADA system, Potentiometer, Modbus, Arduino, RS-485 protocol, Resistance, Capacitance, Inductance, Remote.

Resumen— En este artículo se presenta el diseño, construcción y evaluación de un circuito de control realizar el control y cambio de valor de resistencia, inducción y capacitación de módulos pertenecientes a la microrred Unigrid, y su integración a un sistema SCADA para el monitoreo, visualización y control de los valores seleccionados en tiempo real por medio del protocolo de comunicación Modbus RS-485.

Index Terms— Sistema SCADA, Potenciómetro, Modbus, Arduino, protocolo RS-485, Resistencia, Capacitancia, Inductancia, Remoto.

I. Introducción

Con el avance de la tecnología y debido a la disminución de precios de materiales han surgido distintos elementos que permiten la simulación o emulación de ambientes o sistemas, permitiendo así hacer montajes de prototipos para tener versiones de menos consumo de recursos, ya sea para desarrollos preliminares o por cuestión de aprendizaje. Entre estos elementos podemos encontrar bancos de capacitores, bancos de resistencias, bancos de inductancias o controladores programables, entre otros. Siendo los módulos de resistencia elementos fundamentales a la hora de hacer diseño de circuitos eléctricos y también electrónicos.

Existen distintos modelos de bancos de resistencia con diferentes configuraciones que permiten crear un número específico de combinaciones o valores diferentes de resistencia, este puede tener por dentro 3 barras de resistencias las cuales se dividen en 3 secciones diferentes, que tienen valores diferentes que, al ser conmutadas por un selector, activan caminos que dan como resultado un valor predeterminado. Sin embargo, los módulos con los que se cuentan actualmente en el laboratorio Unigrid tienen aplicaciones limitadas ya que no permiten ser manipulados más allá de su configuración manual, lo que se ha vuelto un problema, ya que al no contar con la posibilidad de hacer manejo remoto obliga a los usuarios o diseñadores a estar presentes en los laboratorios o aulas para poder hacer uso de

estos elementos. Por tanto, surge la necesidad de encontrar soluciones o implementaciones que permitan suplir esta deficiencia con la que se cuenta actualmente.

Este proyecto se llevará a cabo en el departamento de Ingeniería Eléctrica y Electrónica de la Universidad del Norte en la microrred Unigrid, esta posibilita probar de forma física con sistemas GD (generación distribuida) y bancos de cargas. Este laboratorio se implementó con recursos de Colciencias y se usa con la finalidad de examinar la operación real del sistema GD con fuentes híbridas, que normalmente se hace con programa de simulación. En esta red se encuentran modelos de generación de energía por medio de fuentes convencionales y renovables [1].

II. Marco Teórico

A. Protocolo Modbus y RapidScada

Modbus es un protocolo de comunicación diseñado para permitir a equipos industriales tales como Controladores Lógicos Programables (PLCs), computadores, drivers para motores y otros tipos de dispositivos físicos de entrada/salida comunicarse sobre una red. Este protocolo es utilizado con el objetivo de supervisión y control de equipos de automatización, trabajando en conjunto con los protocolos TCP/IP y RS-485, y tiene como característica principal una comunicación tipo maestro-esclavo, en el que cada solicitud del maestro es tratada de forma independiente por el esclavo. Además, las operaciones de programación están orientadas a conexión, de tal manera que las máquinas de origen y destino deben establecer un canal de comunicaciones antes de transferir datos. Este tipo de operaciones son implementadas de diferentes maneras por las diversas variantes de Modbus, tales como Modbus RTU y Modbus ASCII. [2]

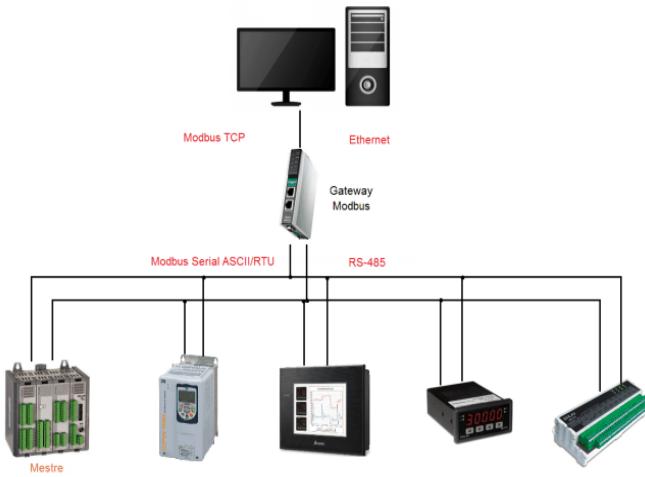


Figura 1: Protocolo Modbus RS-485 [3]

Por otra parte, el software RapidScada es una plataforma de automatización industrial de código abierto (open source), que proporciona herramientas para la creación rápida de sistemas de monitoreo y control, utilizando como núcleo para el desarrollo de soluciones el sistema SCADA (Supervisory Control and Data Acquisition). Este es un software que se ejecuta en servidores, computadoras integradas y en la nube. Las principales clases de sistemas desarrollados utilizando Rapid SCADA son las siguientes: Sistemas de automatización industrial y sistemas IoT, sistemas de control de procesos y sistemas de contabilidad de energía. [4]

B. Arduino y Módulo RS-485 a TTL

El Arduino Uno es un componente electrónico perteneciente a la familia de microcontroladores, de gran flexibilidad en su hardware y en su software puesto que son libres. Por lo anterior se puede decir aceptan cualquier tipo de periféricos a través de sus puertos y permite recibir información y procesarla de manera eficiente. El Arduino tiene 6 pines analógicos de entrada y 13 entradas y salidas digitales, además de eso usa una comunicación serial UART, para la cual sólo utiliza 3 puertos:

- Recepción de datos (RX)
- Transmisión de datos (TX)
- Punto común (GND)



Figura 2: Arduino UNO

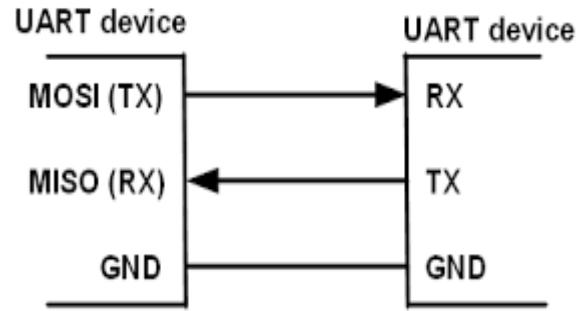


Figura 3: Formato de envío del protocolo UART[5]

Para conectar este dispositivo con la red Unigrid, se tiene que usar el conversor TTL a RS-485 el cual hace una conversión recíproca de la información.

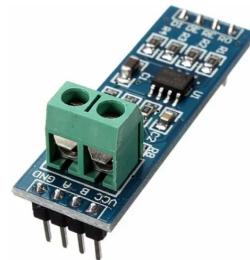


Figura 4: Módulo conversor de TTL a RS485.

C. Banco de resistencia: Resistencia modulable de 6 posiciones.

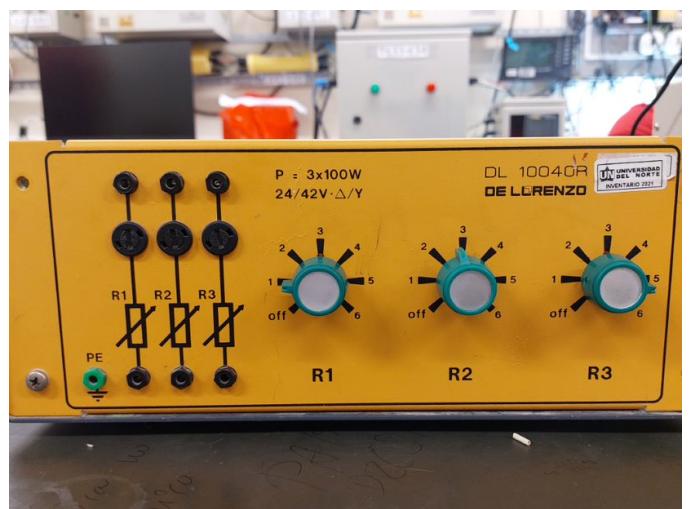


Figura 5: Módulo de resistencia

Es un banco de resistencia cuya función principal es proveer al usuario un número específico de valores diferentes de resistencia, este módulo cuenta con un total de 3 resistencias,

que pueden variar entre 6 valores distintos (Ver figura 5). Este módulo funciona a partir de una resistencia seccionada en tres partes, las cuales a partir de un selector permite conmutarlas. (Ver figura 6)

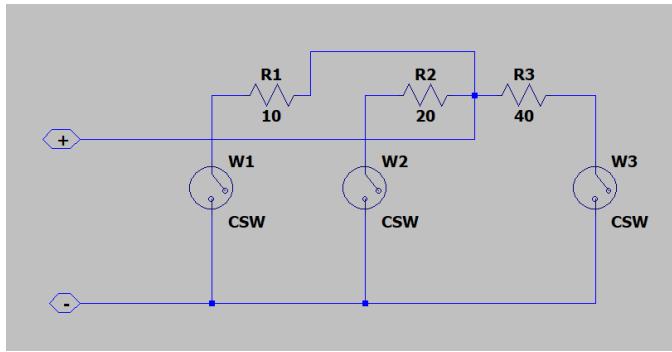


Figura 6: Modelo de Switch interno del módulo de resistencias.

Cada una de estas posiciones activa o desactiva los caminos de retorno de cada tramo de resistencia, generando 6 circuitos con un valor diferente a los anteriores, estos circuitos pueden ser visualizados en la figura 7.

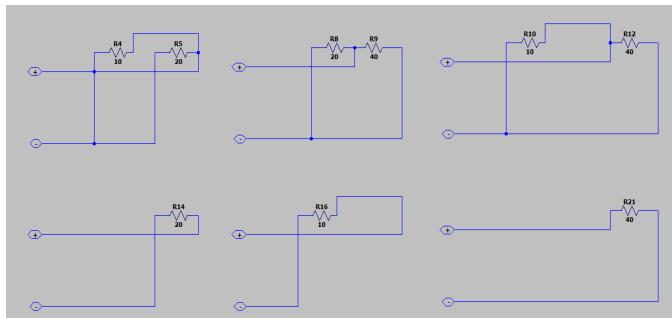


Figura 7: Estados comutados del módulo.

D. Potenciómetro 100k

Este componente electrónico es una resistencia variable de modo lineal que al rotar el eje en diferentes ángulos se obtiene un rango de 0-100 ohm y que se puede usar para crear un divisor de tensión ajustable.



Figura 8: Diagrama potenciómetro 100k

III. Requerimientos de Diseño

Los bancos de carga actualmente disponibles en la Unigrid se controlan manualmente por medio de selectores ubicados en la parte frontal de estos dispositivos. Lo anterior hace dispendioso la configuración de perfiles variables de carga para el desarrollo y prueba de diversos algoritmos de control de microrredes y sistemas de distribución en general. Por lo anterior, este proyecto busca automatizar la configuración de al menos un banco de carga disponible en el laboratorio Unigrid, de tal manera que el usuario de esta infraestructura pueda configurar valores específicos y perfiles de carga resistiva, inductiva y capacitiva desde el SCADA de la microrred.

Para el desarrollo del diseño se deben tener en cuenta los siguientes requerimientos:

- Este proyecto implica el diseño de un prototipo de selector automático y una interfaz de usuario para la operación remota del sistema sin eliminar la opción de manejo manual.
- Se requiere competencias básicas adquiridas en los cursos de Circuitos, Electrónica, Mediciones e Instrumentación, Control Automático, Microcontroladores, Diseño Electrónico.
- Debe implementarse un diseño modular que resista las temperaturas que se puedan generar al interior del módulo.
- Incorporar múltiples restricciones reales: El prototipo del selector y la interfaz gráfica deben adaptarse al banco existente y al software Rapid Scada.
- Normatividad de diseño de interfaz: Se seguirán las normas ISA para el diseño de interfaces gráficas.
- Fácil uso: Diseñar una interfaz sencilla de fácil interpretación para el usuario.
- No hay requerimiento de autoalimentación: El sistema puede contar con un sistema de alimentación independiente del banco.
- Mostrar localmente la selección de carga: Se debe implementar un mecanismo que permita conocer la selección del valor que se hizo.
- La solución permite agregar futuras ampliaciones: El diseño debe ser de configuración abierta tal que permita la ampliación de sus funciones o modificaciones de las funciones originales.

IV. Criterios de Diseño y Selección de Tecnologías

Tabla 1: Tecnologías como posibles soluciones.

función	tecnología
operación manual	<ul style="list-style-type: none"> • potenciómetro • botón • Perilla • Slider
Controlador	<ul style="list-style-type: none"> • Pic 16f877 • Arduino uno • Arduino mega • Raspberry
Sistema de conmutación	<ul style="list-style-type: none"> • relés mecánicos • Relés de estado sólido • Transistores • Multiplexor

Tabla 2: Selección en el controlador manual.

I= Regular 2- Bueno 3 - Óptimo	Potenciómetro	Botón	Perilla	Slider
Conexión (sin contar alimentación)	2	3	2	2
Tamaño y forma	3	3	2	1
Cantidad de estados	3	1	1	3
Facilidad de uso	2	1	3	2

En la Tabla 2 se pudo ver la escogencia del controlador manual y se nota con bastante ventaja, las características fuertes que deja al potenciómetro encima de los demás, por eso se escogió como el indicado para nuestro diseño del hardware.

Tabla 3: Selección conmutador

I= Regular 2- Bueno 3 - Óptimo	Relés Mecánicos	Relés de E. sólido	Transistores	Multiplexor
Facilidad de control	3	3	1	2
Configuración	3	3	1	2
Disponibilidad	3	1	3	3

Número de conexiones	2	2	2	3
----------------------	---	---	---	---

En la Tabla 3 se pudo ver la escogencia del conmutador y como resultado, las características que tienen los relés mecánicos pudieron imponerse en la decisión para la escogencia y diseño del hardware.

Tabla 4: Selección controlador lógico

I= Regular 2- Bueno 3 - Óptimo	Pic16F 877	Arduino 1	Arduino mega	Raspberry
Documentación existente	1	3	3	1
Adaptabilidad	2	2	2	3
Compatibilidad	2	3	3	2
Rendimiento	3	2	2	3

El Arduino 1 se lleva la elección del controlador lógico, debido a las características de facilidad, flexibilidad, que son adicionales a las ya nombradas en las tablas y eso lo hizo el candidato idóneo para integrar el diseño del proyecto.

a. Automatización del banco de carga (Driver):

Para lograr el objetivo de automatizar los bancos de carga, se usará un sistema de relés de referencia “JQC-3FF-S-Z”, los cuales serán controlados por medio del microcontrolador Arduino con el fin de manejar las conmutaciones de las secciones y de esta manera generar los distintos valores de resistencia. A su vez estos serán los encargados de generar los caminos de paso para el voltaje del módulo, lo cual evita que el controlador se lleve a quemar por exceso de corriente o voltaje lo que lo convierte en la mejor opción para el diseño propuesto. Así mismo, para este sistema se usará el microcontrolador Arduino debido a su alta disponibilidad en el mercado, precio accesible, la cantidad de librerías que posee para poder controlar los distintos elementos necesarios. Así como se decidió el uso de potenciómetros para el manejo manual. Ya que son más funcionales y no pierden la estética del diseño original.

b. Comunicación con Scada:

Como ya se mencionó previamente, uno de los requerimientos del diseño es tener conexión con el sistema de Scada del laboratorio el cual para su conexión hace uso del protocolo Modbus. Por esta razón fue necesario utilizar un adaptador a

RS-485 con el fin de realizar una conexión Modbus-Serial entre el controlador y la interfaz diseñada en Scada. En los requerimientos se pide que se realice conexión por medio de un cable RJ-45 sin embargo para las pruebas es necesario hacer uso de un conversor a usb, así que se decidió dejar en paralelo las salidas para las entradas A y B del conversor.

V. Solución Propuesta

Gracias a la información anteriormente expuesta en las secciones III y IV, se realizó el diseño del circuito de control con las tecnologías escogidas. Se ha seleccionado el microcontrolador *Arduino*, el cual controlará los valores de resistencia de salida del banco haciendo uso de un sistema de relés que crean combinaciones o circuitos que darán como resultado distintos valores de resistencia. Además, contará con potenciómetros los cuales van a estar conectados al circuito de control de los bancos de carga, de tal manera que, al variar la resistencia a través del potenciómetro, se produzca una variación en la resistencia del banco.

Cabe resaltar que el diseño del sistema consta de una serie de borneras que permiten hacer las conexiones de cada una de las líneas del circuito de resistencias (ver figura 7), este circuito consta de un total de 4 donde 3 de estas están conectadas a una resistencia, y la activación o desactivación de estas generará circuitos diferentes que traen como resultado un total de seis valores de resistencias. Adicionalmente unos cuatro cables que funciona como retorno. tal como se puede apreciar en la figura 6.

A. Programación del Arduino

Para hacer uso del sistema Modbus es necesario tal como se muestra en la figura 9 incluir las librerías “Modbus.h” y “ModbusSerial.h” los cuales permiten hacer uso de dicho protocolo, además creamos las direcciones donde se encontrarán los registros que se van a utilizar, así como las variables necesarias para el desarrollo de la lógica necesaria.

```
#include "Modbus.h"
#include "ModbusSerial.h"

// Modbus Registers Offsets (0-9999)
const int POT1_MB = 3000;
const int POT2_MB = 3001;
const int POT3_MB = 3002;
const int Estadol = 3010;
const int Estado2 = 3011;
const int Estado3 = 3012;
const int Manual_MB = 3003;
const int VManual_MB = 3013;
```

Figura 9: Parámetros necesarios para la comunicación

Adicionalmente se inicializan los pines donde se hará la conexión tanto de los relés, así como de los pines de control de lectura y escritura del Convertidor TTL, el led y la definición de ModbusSerial cómo mb. Además, se definen los pines

como salida para los relés y el led y como entrada los pines analógicos para las perillas. Por otro parte, se configuran los parámetros necesarios para el control por Modbus y se define su id de esclavo en este caso “202”.

```
// Config Modbus Serial (port, speed, byte format)
Serial.begin(9600);
mb.config(&Serial, 9600, SERIAL_8N1,En_WrRd_RS485);
// Set the Slave ID (1-247)
mb.setSlaveId(202);
```

Figura 10: Configuración Modbus

En el ciclo principal se estará leyendo el registro asociado a manual para activarlo o desactivarlo en caso de que el usuario lo requiera. Para la lógica de la interfaz, en el programa se lee el valor del registro asociado a cada una de las 3 perillas del módulo y a partir de una lógica se controlan los relés haciendo uso de la función select mb, además dicho condicional evita que se esté escribiendo esto cada que se lee el mismo valor, es decir que si no existe cambio de estado el sistema no opera tal como se aprecia en la figura 11. Una vez se active el modo manual, se sigue la misma lógica con las perillas con el fin de cambiar en la interfaz el valor actual, así como hacer la configuración correspondiente haciendo uso de la función de la figura 12. Es necesario aclarar que se desarrolló la lógica del estado anterior con el fin de que no hubiese conflicto al tener el estado manual activo ya que al ser un código lineal de ejecución continuo si no se tiene una condición que limite el cambio constante, tendríamos conflicto entre el estado manual y el estado remoto. Es por esto por lo que se ejecuta un cambio sólo cuando se detecta el envío de un estado diferente al actual.

```
mb.task();
//Obtenemos activacion de protocolo manual
Manual_ar=mb.Hreg(Manual_MB);
mb.Hreg(VManual_MB,Manual_ar);

// Logica de mb
estado=mb.Hreg(POT1_MB);
if (estado != estadolant) {
    estadolant=estado;
    mb.Hreg(Estadol,estado);
    selectmb(estado,13,12,11);
    delay(200);
}
```

Figura 11: Lectura y control por interfaz.

```

void selectmb(int valormb,int pinamb,int pinrmb,int pinvmb){
    if (valormb ==0) {
        //Apagado
        digitalWrite(pinamb, LOW);
        digitalWrite(pinrmb, LOW);
        digitalWrite(pinvmb, LOW);
    } else if (valormb ==1) {
        //Primer estado
        digitalWrite(pinamb, HIGH);
        digitalWrite(pinrmb, HIGH);
        digitalWrite(pinvmb, HIGH);
    } else if (valormb ==2) {
        //Segundo estado
        digitalWrite(pinamb, LOW);
        digitalWrite(pinrmb, HIGH);
        digitalWrite(pinvmb, HIGH);
    } else if (valormb ==3) {
        //Tercer estado
        digitalWrite(pinamb, LOW);
        digitalWrite(pinrmb, HIGH);
        digitalWrite(pinvmb, LOW);
    }
}

```

Figura 12: Función de conmutación.

B. Integración del circuito de control a la Red

En la microrred Unigrid se utiliza el software libre llamado *Rapid Scada*, en el cual se configura todo el sistema SCADA que se ejecuta en el laboratorio. Gracias al módulo RS-485 a UART que se encuentra en el circuito de control, el cual comunica al Arduino con la red, se realizó la integración del controlador diseñado a la red Modbus del laboratorio. Se le asignó la dirección 200 de la red y se realizaron las configuraciones en el administrador de Rapid Scada, como se muestra en la figura 11.

VI. Diseño de Experimentos

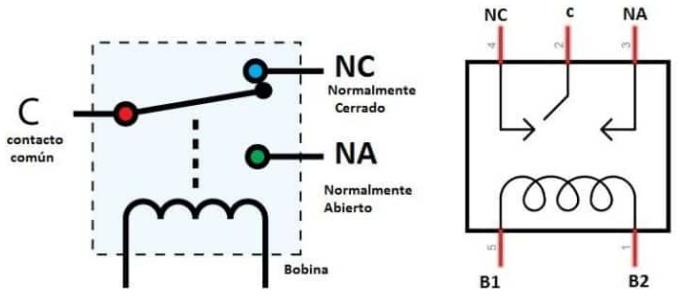
A. Evaluación del estado del módulo.

En primera instancia fue necesario hacer una evaluación completa del módulo, con el fin de conocer el estado actual del mismo, conocer si éste tiene algún daño y además conocer el sistema que permite la variación de resistencias con el fin de conocer cómo se puede reemplazar el mismo para hacer el sistema de manejo remoto. Una vez destapado y tras una evaluación exhaustiva del mismo, se pudo obtener una hipótesis del funcionamiento, la cual trajo como resultado los circuitos mostrados en las figuras 6 y 7.

B. Manejar banco haciendo uso de un controlador.

Para realizar el diseño del sistema automático es necesario poder controlar por medio de un controlador que permita eventualmente mandar comando por una interfaz y lograr una variación en el valor proporcionado por el banco. Por lo cual, con base en lo obtenido tras la revisión del módulo, se optó por un sistema de interruptores que permiten activar o desactivar estos caminos basado en código o instrucciones, siendo además que la opción escogida para cumplir con esta función fueron los Reales, ya que estos pueden ser puestos en

corto o circuito abierto a partir de una señal de control (ver figura 13).



Al meter corriente por la bobina los contactos abiertos se cierran y los cerrados se abren.

Figura 13: Funcionamiento de un relé [6].

Teniendo en cuenta que cada resistencia estaba dividida en tres secciones y que el módulo tiene un total de tres, es necesario utilizar nueve relés para conectar y habilitar cada una de estas secciones por medio de comandos dados por el microcontrolador.

C. Implementación de la interfaz.

La interfaz se realizó de tal manera que fuese fácil de entender y operar. Para esto se tomó como referencia el propio diseño del módulo y se realizó una réplica en el Scada para que fuese intuitivo. En dicha interfaz se programan los registros definidos en la figura 9, teniendo en cuenta la diferenciación entre comandos y grupos (ver figura 15). Se genera una tabla con las entradas y salidas y se dispone una lógica de control que permite enviar comandos y manipular la visualización del estado a través de esta, así como el indicador de activación de manejo manual (Ver figura 14).



Figura 14: Diseño de interfaz

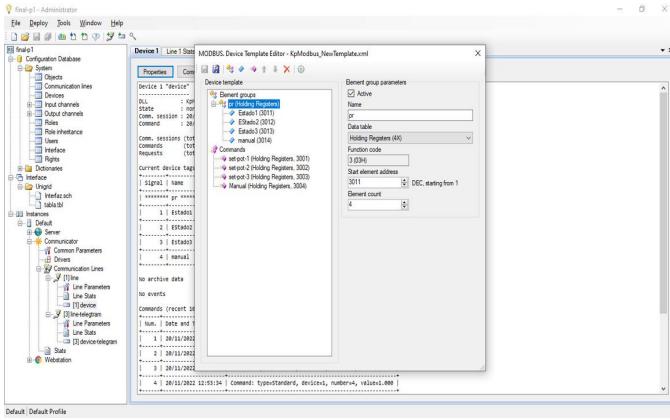


Figura 15: Activación de registros.

VII. Resultados

Se pudo obtener la digitalización del banco de carga a través del trabajo conjunto entre la codificación del Arduino, configurando en su lenguaje todos los aspectos a procesar con los demás componentes electrónicos, además de su alimentación y comunicación hacia los relés para commutar estados manual o automático haciendo variar las salidas y dar diferentes valores de resistencias de modo ascendente en los 6 estados disponibles. También el Arduino se comunica con el conversor TTL a RS-485 que permite tener comunicación Modbus el cual nos da la entrada hacia la interfaz del software Rapidscada. Ya con la comunicación establecida y dentro del Arduino codificado, se pasa al nivel de crear y establecer la interfaz la cual nos commuta los relés y obtener los diferentes estados valores de resistencia. Adicional a esto, un gran resultado se presentar al incluir los plugins que nos permiten tener acceso a tablas de históricos (Ver Figura 16)

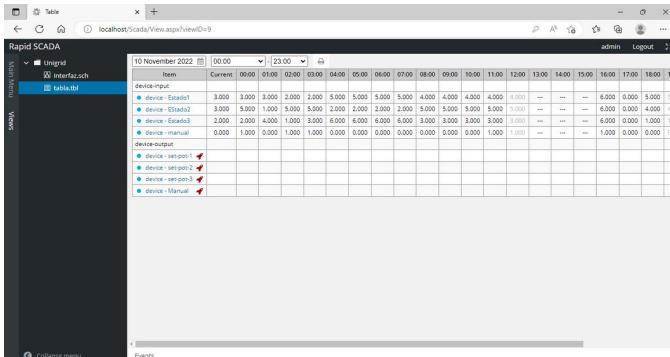


Figura 16: Tabla de históricos.

Además, se pueden generar tablas de los históricos o de los cambios en tiempo real, y exportar datos a tipo PDF, PNG y EXCEL. Cabe resaltar que este último se expresa en valores de tiempo y estado, lo cual permite tener valores y poder manipularlos de forma que se vea un comportamiento adecuado y tomar correcciones en el tiempo y tipo de estado. Por ultimo, se cumplió el deber de digitalización y control a nivel remoto del sistema dado que con las adecuaciones se puede trabajar el módulo desde cualquier parte de la red Unigrid siempre y cuando tengan la entrada de los medidores un puerto RJ-45 y un computador.

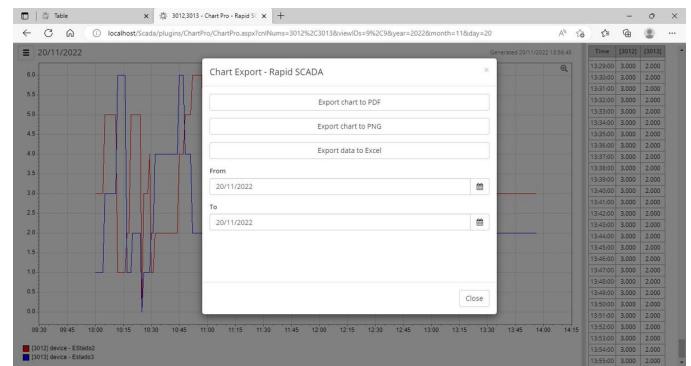


Figura 17: Gráfica de tabla de exportación

VIII. Conclusiones

Los bancos de carga físicos presentan riesgos de manipulación en algún instante, por los niveles de tensión y corriente manejados, además de presentar limitaciones para el cambio constante de valores. El paso a la digitalización de estos módulos o bancos de cargas puede llegar a ser una gran ventaja en la variación de datos, en la recreación de diferentes escenarios posibles a través del tiempo (circuitos eléctricos y electrónicos), en la optimización de proceso y sobre todo para un medio que permite obtener y manipular estados sin tener que estar presente en el lugar del módulo, a través de una interfaz sencilla y usable. Si bien hay muchas oportunidades de mejoras, el diseño, la realización y el montaje cumple con los objetivos planteados y demuestra cómo, por medio de un hardware, podemos entrar a la red y cambiar y mostrar datos, a distancia y sin contacto alguno con el objeto físico, que en este caso es el banco de carga de la Unigrid.

IX. Bibliografía

- [1]“NUESTRO CAMPUS LAB - INGENIERÍA ELÉCTRICA - UNINORTE,” INGENIERÍA ELÉCTRICA, 2020. [HTTPS://WWW.UNINORTE.EDU.CO/EN/WEB/INGENIERIA-ELECTRICA/NUESTRO-CAMPUS-LAB.](https://www.uninorte.edu.co/en/web/ingenieria-electrica/nuestro-campus-lab)
- [2]A. F. RUIZ OLAYA, A. BARANDICA LÓPEZ, Y F. G. GUERRERO MORENO, “IMPLEMENTACIÓN DE UNA RED MODBUS/TCP”, INYCOMP, VOL. 6, N.º 2, PP. 35–44, JUN. 2011.
- [3]A. SPERDUTY, “¿QUÉ ES MODBUS? ¿CÓMO FUNCIONA?,” ARS ELECTRONICA, MAR. 2019.

[HTTPS://BLOG.ARS-ELECTRONICA.COM.AR/QUE-ES-MODBUS-FUNCIONAMIENTO](https://blog.ars-electronica.com.ar/que-es-modbus-funciona-miento)

[4]”WHAT IS RAPIDSCADA” AVAILABLE ON: RAPID SCADA |
FREE, OPEN SOURCE, FULL FEATURED SCADA SOFTWARE

[5]M. PEREZ ESTESO, “PUERTOS Y BUSES 1: I2C Y UART,”
GEEKY THEORY.

[HTTPS://GEEKYTHEORY.COM/PUERTOS-Y-BUSES-1-I2C-Y-UART](https://geekytheory.com/puertos-y-buses-1-i2c-y-uart)

[6]“RELÉ: CÓMO FUNCIONAN, TIPOS, CONSTRUCCIÓN,
APLICACIONES, PRUEBA Y MÁS,” MUNDO MOTOR.
[HTTPS://WWW.MUNDODELMOTOR.NET/RELE/](https://www.mundodelmotor.net/rele/)

Anexos



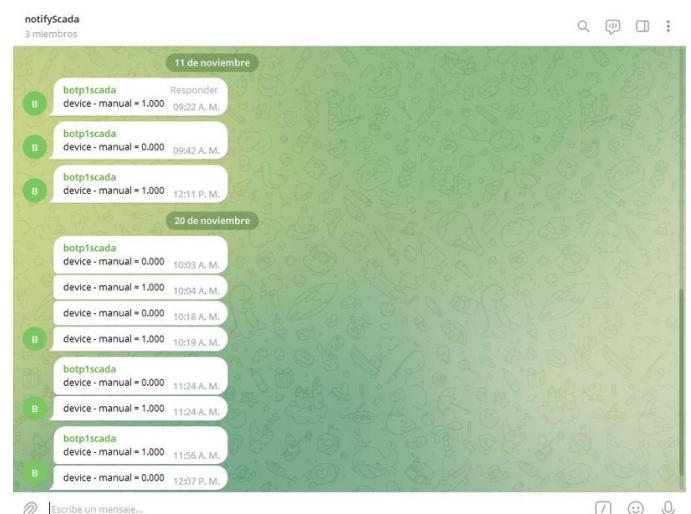
Anexo 1: Módulo de control por Modbus.



Anexo 3: Indicador de estado manual activo.



Anexo 2: Prueba de funcionamiento.



Anexo 4: Notificación de activación o desactivación de manejo manual por telegram.