



TECNOLOGICO NACIONAL CAMPUS TUXTLA GUTIERREZ

CFE Transmisión

Gerencia Regional de Transmisión Sureste

Zona de Transmisión Tuxtla

REPORTE DE RESIDENCIA PROFESIONAL PROYECTO

**Medición y Análisis en tiempo real de variables para protección a
Bancos de Baterías e Integración al sistema ZABBIX**

PRESENTA

Alex Beltrán Solís Martínez

ASESOR INTERNO

Ing. Odilio Orozco Magdalena

CICLO ESCOLAR, ENERO 2019

INDICE

| | |
|---|-----------|
| CAPÍTULO I GENERALIDADES | 6 |
| 1.1 Introducción | 6 |
| 1.2 EPS Transmisión (CFE). | 7 |
| Objetivo EPS Transmisión | 7 |
| 1.3 Política de Calidad de la Empresa | 7 |
| 1.4 Información General de la CFE | 8 |
| Misión | 8 |
| Visión | 8 |
| Ambición | 8 |
| Valores institucionales | 8 |
| Retos | 9 |
| Organigrama de la empresa | 9 |
| 1.5 Área Específica Relacionada Directamente con el Proyecto | 10 |
| 1.6 Antecedentes | 12 |
| 1.7 Planteamiento del Problema | 13 |
| 1.8 Nombre del Proyecto | 13 |
| 1.9 objetivo general y específico | 13 |
| Objetivo General | 13 |
| Objetivo Específico | 14 |
| 1.10 Alcances y Limitaciones del Proyecto | 14 |
| Alcances | 14 |
| Limitaciones | 14 |
| 1.11 Metodología del Proyecto | 14 |
| CAPITULO II FUNDAMENTO TEÓRICO | 15 |
| 2.1 Selección de Componentes a Utilizar | 15 |
| 2.2 Arduino Mega2560 | 15 |
| 2.3 Arduino ethernet shield | 16 |
| Descripción del módulo Arduino Ethernet Shield | 17 |

| | |
|--|-----------|
| Comunicación del módulo Arduino Ethernet Shield | 17 |
| 2.4 Max6675 Módulo Interfaz para Termopar tipo K | 18 |
| Max6675 | 18 |
| Termopar tipo K | 19 |
| Rangos de Temperatura | 20 |
| 2.5 Sensor de Humedad Relativa DHT11 | 20 |
| Características Técnicas | 21 |
| 2.6 Divisor de Tensión con Amplificador Buffer Opamp Im741 (Seguidor de Voltaje) | 21 |
| 2.7 sensor de efecto hall wcs1800-35 ampers | 24 |
| Características del Diagrama | 26 |
| Factor de Corrección | 26 |
| Conexión con Arduino | 27 |
| 2.8 Módulo de Reloj | 28 |
| Características DS3231 | 28 |
| 2.9 Memoria Micro SD | 29 |
| 2.10 Pantalla LCD Display Blue 20x4 | 29 |
| 2.11 Regulador de Voltaje de 48-54 Volts a 12 Volts | 30 |
| 2.12 Regulador de Voltaje de 12 Volts a 5 Volts | 32 |
| Especificaciones del regulador LM7805 | 32 |
| CAPITULO III DESARROLLO E IMPLEMENTACIÓN DEL PROYECTO | 33 |
| 3.1 Descripción del Sistema de Fuerza del Área de Transmisión, zona Tuxtla; incluyendo los diagramas de todas las partes que lo conforman | 33 |
| Sistema modular de fuerza | 33 |
| Módulo de entrada, salida y alarmas | 33 |
| Módulo de distribución | 34 |
| Banco de baterías | 34 |
| Diagrama de Bloques de todo el Sistema de Fuerza | 36 |
| 3.2 Selección de Materiales a Utilizar y los Costos de cada Componente; incluyendo un diagrama de bloques del nuevo sistema de fuerza | 36 |
| Diagrama de Bloques del nuevo Sistema de Fuerza | 38 |

| | |
|--|-----------|
| 3.3 Diseño de Circuitos en PCB WIZARD | 41 |
| 3.4 Simulaciones en Proteus | 41 |
| <i>Simulación del divisor de tensión con alta impedancia (sensor de voltaje)</i> | 41 |
| <i>Simulación del sensor de temperatura, termopar modulo max6675 tipo k</i> | 44 |
| <i>Simulación del sensor dht11 humedad-temperatura ambiente</i> | 44 |
| <i>Simulación de la fuente regulable de 48-54 a 12 Volt y 12 a 5 volt</i> | 46 |
| 3.5 Diseño de la Placa Etapa de Divisor de Tensión (Diferencia de Potencial) en PCB y vista 3D | 47 |
| 3.6 Placa de Adquisición de Datos | 48 |
| 3.7 Diseño de la Placa de etapa Adquisición de Datos en PCB y en 3D | 49 |
| 3.8 Materiales para Construcción de las Placas Fenólicas | 50 |
| 3.9 Construcción de la Placa de Adquisición de Datos Divisor de Tensión (Diferencia de Potencial) y Fuente Regulable de 48-54 Volts | 51 |
| <i>Construcción de la placa adquisición de datos</i> | 51 |
| <i>Construcción del Proceso de adaptación a los pines sobrantes del Arduino ATmega2560 y Ethernet Shield</i> | 53 |
| <i>Construcción de la placa Divisor de Tensión (Sensor de Voltaje)</i> | 53 |
| <i>Construcción de la placa fuente regulable 48-54 Volt</i> | 54 |
| 3.10 Vistas de las Placas de los Diseños | 56 |
| 3.11 Prueba de Funcionamiento del Proyecto | 57 |
| 3.12 Implementación del Proyecto en Sistema de Fuerza (Banco de Baterías) | 59 |
| 3.13 Diagramas de las Conexiones de los Circuitos de la Placa de Adquisición de Datos y Diferencia de Potencial | 60 |
| <i>Adquisición de datos</i> | 60 |
| <i>Sensor de voltaje</i> | 61 |
| <i>Programación para el Funcionamiento del nuevo Sistema de Fuerza</i> | 62 |

| | |
|---|-----------|
| 3.14 Comunicación a la Plataforma ZABBIX | 83 |
| 3.15 Datos Guardados en la Micro SD del Arduino Ethernet Shield y enviados a Plataforma ZABBIX | 84 |
| 3.16 Diagrama de los Datos Obtenidos | 85 |
| Observaciones y Sugerencias | 87 |
| Conclusión | 88 |
| Bibliografía | 89 |

CAPÍTULO I GENERALIDADES

1.1. Introducción

El Departamento de Comunicaciones de la Empresa Productiva Subsidiaria (EPS) Transmisión de la Comisión Federal de Electricidad, Zona de Transmisión Tuxtla, tiene instalados varios Sistemas de Fuerza de 48VCD, los cuales proporcionan respaldo de alimentación a equipos de Fibra óptica (SDH/MPLS), Teleprotección, Datos (Router/Switch), Conmutadores telefónicos PBX e IP's, equipos de VHF y OPLAT, en condiciones de falla de la alimentación principal o afectación de servicios propios de corriente alterna (VCA), a los equipos de comunicaciones antes mencionados.

Para el respaldo de alimentación, estos sistemas de fuerza utilizan bancos de baterías de 48VCD, los cuales se encuentran distribuidos por dos camas de baterías de 12 volts cada celda, dependiendo de la capacidad del banco.

En estos sistemas se puede tener el monitoreo de alarmas de alto y bajo voltaje, falla de alimentación VCA, así como otras alarmas propias del cargador, los cuales son monitoreadas por protocolo SNMP, a través del servidor ZABBIX.

Cada Sistema de Fuerza cuenta con un módulo de control, en el cual es posible observar su comportamiento, sin embargo, no proporciona los datos específicos del o mediciones del Banco de Baterías.

El presente proyecto consiste en la adquisición, medición, almacenamiento y análisis en tiempo real, de diferentes variables, tales como: voltaje, corriente, temperatura y humedad del Banco de Baterías, para poder prevenir anomalías o condiciones críticas del mismo.

Estas mediciones y análisis consisten en obtener la temperatura, humedad, el voltaje y la corriente en cada celda que conforma el banco, con la finalidad de determinar condiciones anormales que pueden ocasionar daños en los bancos de baterías. Por lo consiguiente afectación de alimentación a los equipos conectados al sistema de fuerza.

Para ello, los datos adquiridos para el análisis del tiempo real, utilizando nuevas tecnologías hoy en día de bajo costo, serán integradas al servidor ZABBIX bajo la responsabilidad de la Zona de Transmisión Tuxtla.

1.2. EPS Transmisión (CFE)

La Comisión Federal de Electricidad, es una empresa productiva del Estado de propiedad exclusiva del Gobierno Federal, con personalidad jurídica y patrimonio propios, gozara de autonomía técnica, operativa y de gestión, conforme a lo dispuesto de la nueva ley de la Comisión Federal de Electricidad, y que tiene por objeto prestar, en términos de la legislación aplicable, el servicio público de transmisión y distribución de energía eléctrica, por cuenta y orden del Estado Mexicano para más de 39.6 millones de clientes, lo que representa más de 122 millones de habitantes e incorpora anualmente más de un millón de clientes nuevos.

El compromiso de la empresa es ofrecer servicios de excelencia, garantizando altos índices de calidad en todos sus procesos, al nivel de las mejores empresas eléctricas en el mundo.

Objetivo EPS Transmisión

Asegurar el acceso a la Red Nacional de Transmisión de la CFE, mediante su operación, mantenimiento, expansión y modernización, además de suministrar otros productos y servicios asociados para crear valor económico al Estado.

1.3. Política de Calidad de la Empresa

Prestar el servicio público de transmisión de energía eléctrica y otros servicios relacionados en el área de las Telecomunicaciones, que generen valor económico y rentabilidad, procurando el mejoramiento de la productividad con sustentabilidad para el desarrollo nacional. Considerando los aspectos ambientales y de seguridad, y a una mejora continua de la eficacia del sistema integral de Gestión, con el compromiso de:

- Formar y desarrollar el capital humano
- Gestión eficiente de los riesgos
- Prevenir la contaminación y aprovechar de manera responsable los recursos naturales
- Cumplir con la legislación, reglamentación y otros requisitos

1.4. Información General de la CFE

Misión

“Prestar el Servicio de Transmisión de Energía Eléctrica, mediante la operación, mantenimiento, expansión y modernización de la Red Nacional de Transmisión, garantizando un acceso abierto y no indebidamente discriminatorio y cumpliendo con las condiciones reguladas de disponibilidad, continuidad y eficiencia, para crear valor económico y rentabilidad para el estado mexicano”.

Visión

“Ser una empresa de Servicio Político de Transmisión con un desempeño equiparable a las mejores empresas del mundo, con presencia internacional y fortaleza financiera, mediante el máximo aprovechamiento de su infraestructura y contribución de su capital humano”.

Ambición

“Transformarnos en una empresa transportista de energía que cumple su mandato de ley como Empresa productiva del Estado, operando, manteniendo, modernizando y expandiendo la RNT con costos eficientes y altos niveles de servicio, que cuenta con una estructura organizacional esbelta, está actualizada tecnológicamente e incursiona en nuevos negocios mediante asociaciones y alianzas que le brindan ventajas competitivas”.

Valores Institucionales

- Integridad
- Productividad
- Responsabilidad

Retos

- Implementar la Gestión de Activos mediante el uso de la Red Eléctrica Inteligente, con el propósito de optimizar la vida útil de los activos de la Red Nacional de Transmisión.
- Cero fallas en Red Nacional de Transmisión
- Desarrollo del talento de los trabajadores de Transmisión, para el cumplimiento de las nuevas funciones.

Organigrama de la empresa de la CFE

La empresa se encuentra estructuradamente jerárquicamente en la zona de transmisión Tuxtla poniendo su organigrama se muestra en la Figura 1.1, que se ve a continuación.

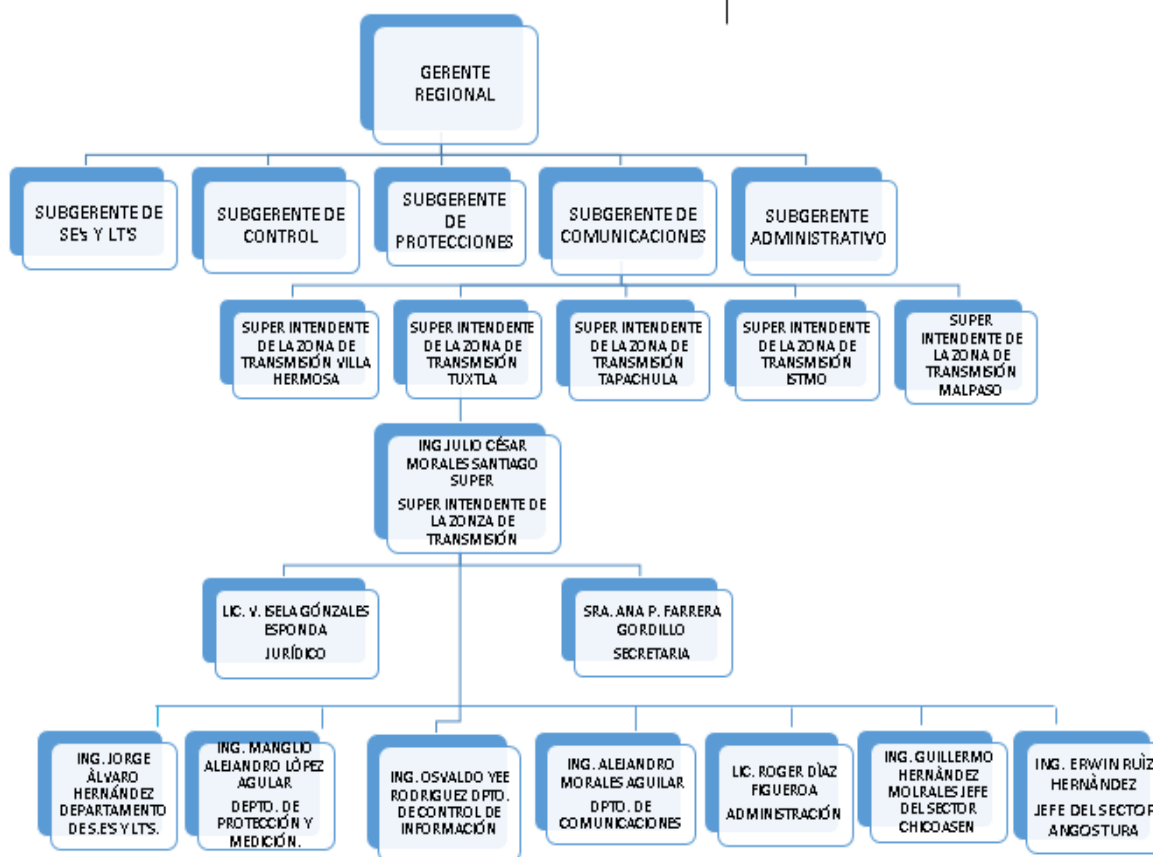


Figura 1.1 Organigrama de la empresa

1.5. Área Específica Relacionada Directamente con el Proyecto

La Gerencia Regional de Transmisión Sureste (GRTSE), forma parte de la EPS Transmisión y tiene injerencia en los estados de Chiapas, Tabasco, Oaxaca y parte de Veracruz a través de sus cinco Zonas de Transmisión y una Zona de Operación que la conforman; asegurando en todo momento el servicio a sus clientes regionales y nacionales alrededor de 20 subestaciones eléctricas de potencia en voltajes de 400, 230 y 115 kV, mismas que están interconectadas y enlazadas al Sistema Eléctrico Nacional; y de acuerdo a la Ley de la Industria Eléctrica, publicado en el Diario Oficial de la Federación, tomará el control de la sección de las Subestaciones con tensión mayor o igual a 69 kV.

La Gerencia Regional de Transmisión Sureste (GRTSE), tiene el control de Subestaciones de 400 kV, 230 kV y 115 kV, que forma parte del Sistema Eléctrico Nacional (SEN), en donde las subestaciones deben estar supervisadas y tele controladas desde los centros de control del Centro Nacional de Control de Energía (CENACE), con el fin de coordinar la operación del SEN. El proceso de transmisión requiere asegurar la confiabilidad física y operativa de sus instalaciones, orientada a mantener una alta seguridad de la red eléctrica.

La Zona de Transmisión Tuxtla, perteneciente a la Gerencia Regional de Transmisión Sureste, se encuentra en la ciudad de Tuxtla Gutiérrez, Chiapas. Atiende actualmente las Subestaciones Eléctricas Manuel Moreno Torres “Chicoasén”, Angostura y El Sabino, las cuales se encuentran conectadas al Sistema Interconectado Nacional mediante enlaces de 400 kV y a la red de 115 kV, con la finalidad de suministrar la demanda de energía de las principales ciudades del Estado de Chiapas. Cuenta con 3 Secciones Sindicales del SUTERM: Sección 155 Chicoasén, Sección 130 Angostura y Sección 47 Tuxtla. Las oficinas Sedes están ubicadas en Carretera Panamericana km. 1077 No. 5675 interior 300 mts. Col. Plan de Ayala de la Ciudad de Tuxtla Gutiérrez, como se muestra gráficamente en el mapa de la Figura 1.2.

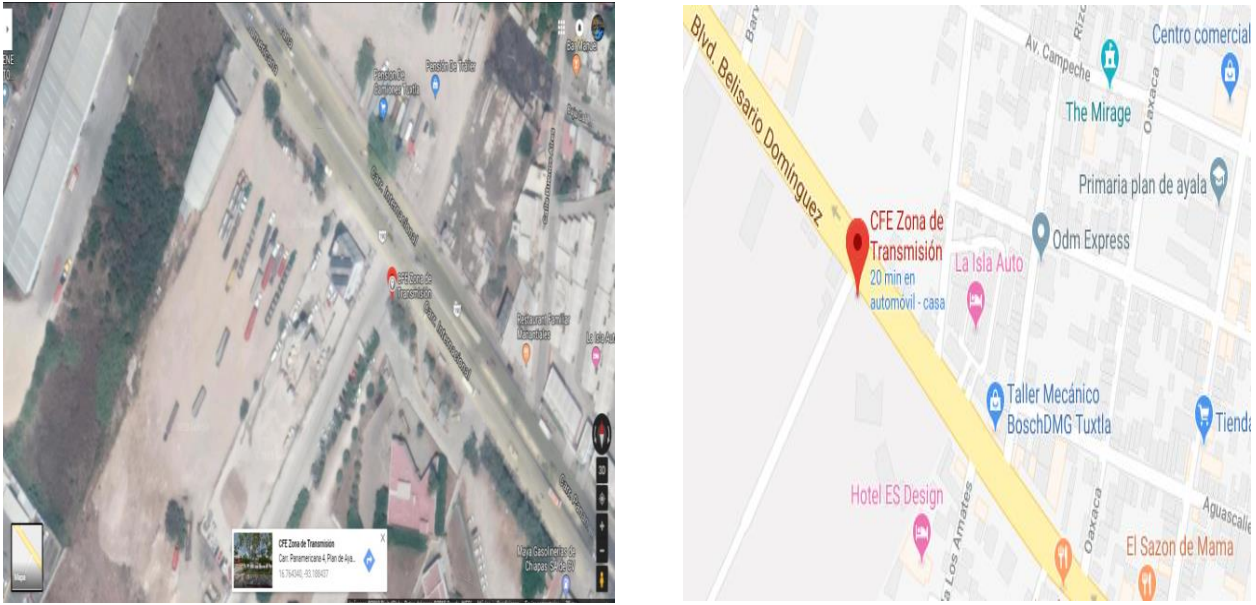


Figura 1.2 Ubicación de la empresa

Está conformada por las especialidades de Comunicaciones, Protección y Medición, Control e Informática, Administración, así como de Subestaciones y Líneas. Siendo cada una de estas importantes para el logro de los objetivos y metas programadas por parte de la Zona Tuxtla.

La Zona de Transmisión Tuxtla, también tuvo la necesidad y el compromiso de proporcionar servicios de Telecomunicaciones a clientes internos y externos, dentro de una Red de Fibra Óptica troncal y de última milla como parte del back toné en la ciudad de Tuxtla Gutiérrez, Cintalapa, Copainalá y Chiapa de Corzo; misma que se encuentra instalada en la Red Eléctrica de Distribución en postaría de 13.8 kV; con la finalidad de proporcionar servicios de datos o Telecomunicaciones a clientes internos y externos como son: SCT, CJF, BANOBRAS, SAGARPA, SCJN, PEMEX, EPS Distribución y EPS Generación. El objetivo principal es mantener en operación los servicios que se proporcionan mediante los enlaces de fibra óptica que se brindan a nuestros clientes internos y externos para cumplir con los requisitos de los contratos, satisfacción del cliente, disponibilidad y la confiabilidad de los servicios.

1.6. Antecedentes

A medida que pasaban los años y la tecnología evolucionaba, se evidenció la necesidad de construir equipos electrónicos portátiles, es por esta razón que las baterías tomaron gran importancia como sistemas indispensables y comenzaron su evolución. De esta forma las baterías de ion-litio se hicieron un espacio importante en el mercado, útiles en distintas aplicaciones. Hoy en día se utilizan estas baterías en dispositivos portátiles, ya que son una alternativa costo eficiente y de pequeño tamaño. Estas razones hacen que el mercado de baterías de litio presente un crecimiento a lo largo de los años. Sin embargo, en muchas de las aplicaciones se requieren voltajes más altos a los entregados por una sola celda de ion-litio, por esta razón surge la necesidad de realizar arreglos en serie.

Como resultado del proceso de fabricación de las baterías, no existen dos celdas exactamente iguales, por esto cada una de las celdas se carga y se descarga de forma diferente, por esta razón al conectarlas en serie se corre el riesgo de sobrecargar o sobre descargar una de las celdas, generando daños directos, afectando su desempeño disminuyendo sus ciclos de carga.

Existen daños causados por sobrecarga, como aumento en la temperatura y generación de partículas de litio, que pueden ocasionar cortos circuitos dentro de la celda, de igual forma, si aumenta la temperatura de la celda por encima del rango permitido, puede acabar con la explosión de la misma.

A pesar de las cualidades de las baterías de litio existen inconvenientes ya que hay que mantenerlas dentro de su rango de operación seguro para evitar daños, disminución en su vida útil o prevenir accidentes. Debido a estos inconvenientes es necesario adicionar circuitos de protección para controlar su carga y su descarga como por ejemplo los BMS (battery management system).

El BMS se encarga de monitorear variables de interés como voltaje y temperatura de cada una de las celdas y a partir de esta toma de datos procurar que las baterías permanezcan dentro de la zona segura de operación, permitiendo aprovechar la batería en mayor proporción, ya que evita que su vida útil disminuya y previniendo efectos de sobrecarga y sobre descarga [3].

El proyecto Sistema de Gestión de Carga para Baterías de Ion-Litio, está centrado en el diseño e implementación de un gestor de carga para 4 baterías de ion litio, con un voltaje nominal de 3,7 V y una capacidad de 2000 mAh conectadas en serie. A partir de las muestras de voltaje, temperatura y corriente, se buscará mantener las baterías dentro de su rango de operación mediante la implementación de circuitos de igualación y de control.

1.7. Planteamiento del Problema

Actualmente en la EPS Zona de Transmisión Tuxtla, los Sistemas de Fuerza, no cuenta con un sistema de monitoreo en los Bancos de Baterías, por lo que es necesario implementar un sistema que tenga la capacidad de realizar mediciones, registrar y proporcionar datos del comportamiento del Banco de Baterías en los Sistemas de Fuerza de 48VCD y para su análisis.

1.8. Nombre del Proyecto

Medición y análisis en tiempo real, de variables para protección a Bancos de Baterías e Integración al Sistema ZABBIX.

1.9. Objetivo General y Especifico

Objetivo General

Diseñar e implementar un sistema de medición y análisis de variables en tiempo real, de forma remota, detectando cualquier anomalía que se pueda presentar en el Banco de Baterías con la finalidad de prevenir cualquier afectación a los equipos de comunicaciones que se encuentran energizados a través del Sistema de Fuerza.

Objetivo Específico

1. Diseñar un dispositivo de monitoreo en tiempo real con ayuda de la tarjeta de adquisición de datos Arduino para el Banco de Baterías.
2. Enviar las variables a medir (temperatura, humedad, voltaje y corriente) mediante Ethernet al Servidor de monitoreo ZABBIX.
3. Mantener actualizado los valores instantáneos de voltaje, corriente, temperatura y humedad en el sistema a implementar.

1.10. Alcances y Limitaciones del Proyecto

Alcances

Realizar la medición y análisis en tiempo real al Sistema de Fuerza 48VCD instalado en el laboratorio de comunicaciones y enviar las variables a medir mediante Ethernet al Sistema de Monitoreo ZABBIX.

Limitaciones

Este Sistema solo es posible hacer el monitoreo de las variables a medir, donde se tenga red de datos TCP/IP.

1.11. Metodología del Proyecto

Realizar una investigación teórica durante tres semanas el cual proporcione información necesaria para realizar el proyecto.

Verificar los materiales que se necesitarán para realizar el sistema de monitoreo de las variables a medir (temperatura, humedad, voltaje y corriente) en tiempo real a través del Arduino Mega2560 para enviarlas al Sistema de Monitoreo ZABBIX mediante Ethernet.

Realizar diseños y pruebas para el correcto funcionamiento para la adquisición de las variables a medir.

Desarrollo del programa Software con un Microcontrolador Arduino para las pruebas y ajustes de los resultados obtenidos.

CAPITULO II FUNDAMENTO TEÓRICO

2.1. Selección de Componentes a Utilizar.

Se determinó de acuerdo a las especificaciones del proyecto sobre el monitoreo al Sistema de Fuerza de 48 VCD, que tipo de tarjeta de adquisición de datos, sensores y de más componentes a utilizar para así poder empezar el desarrollo del programa Software y empezar las pruebas hacia el Banco de Baterías.

2.2. Arduino Mega2560

Dispone de 54 entradas/salidas digitales, 14 de las cuales se pueden utilizar como salidas PWM (modulación de anchura de pulso). Además, dispone de 16 entradas analógicas, 4 UARTs (puertas series), un oscilador de 16MHz, una conexión USB, un conector de alimentación, un conector ICSP y un pulsador para el reset. Para empezar a utilizar la placa sólo es necesario conectarla al ordenador a través de un cable USB, o alimentarla con un adaptador de corriente AC/ DC. También, para empezar, puede alimentarse mediante una batería. (Figura 1.3 Arduino ATmega2560)

Arduino Mega posee las siguientes especificaciones:

- **Microcontrolador:** ATmega2560
- **Voltaje Operativo:** 5V
- **Voltaje de Entrada:** 7-12V
- **Voltaje de Entrada(límites):** 6-20V
- **Pines digitales de Entrada/Salida:** 54 (de los cuales 15 proveen salida PWM)
- **Pines análogos de entrada:** 16
- **Corriente DC por cada Pin Entrada/Salida:** 40 mA
- **Corriente DC entregada en el Pin 3.3V:** 50 mA
- **Memoria Flash:** 256 KB (8KB usados por el bootloader)
- **SRAM:** 8KB
- **EEPROM:** 4KB
- **Clock Speed:** 16 MHz

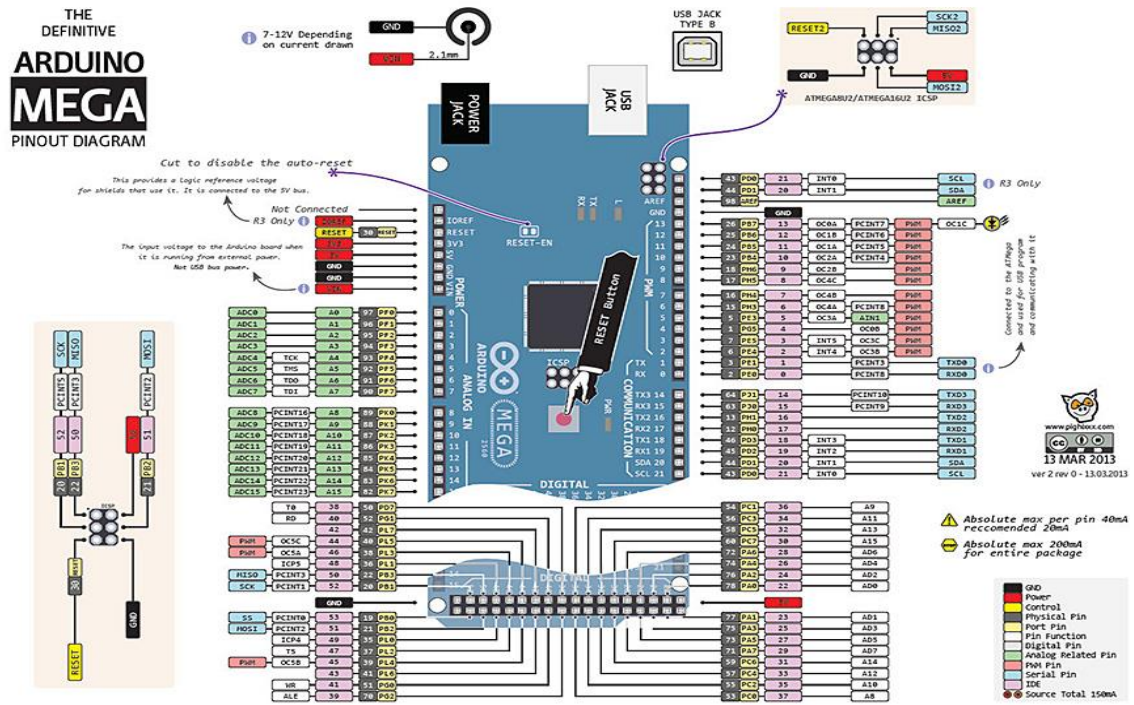


Figura 1.3 Arduino Mega

2.3. Arduino Ethernet Shield

El módulo Arduino Shield Ethernet se conecta con la placa Arduino a Internet sin ningún problema, ya que ambos dispositivos son compatibles. Sólo hay que conectar este módulo en la placa Arduino, conectarlo a su red con un cable RJ45 y seguir algunas instrucciones para controlar el mundo Arduino a través de internet. Todos los elementos de la plataforma Arduino, hardware, software y documentación, es de libre acceso y código abierto. Esto significa que puede aprender exactamente cómo se hace y utilizar su diseño como punto de partida para sus propios circuitos sin ningún costo adicional. (Figura 1.4 Módulo Arduino Shield).



Figura 1.4 Módulo Arduino Shield

Descripción del módulo Arduino Ethernet Shield

El Shield Ethernet permite a una placa Arduino conectarse a internet. Está basada en el chip ethernet Wiznet W5100. Este provee de una pila de red IP capaz de conectarse a través de TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para escribir programas que se conecten a internet. La ethernet shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino.

Comunicación del módulo Arduino Ethernet Shield

Arduino se comunica tanto con el chip W5100 como con la tarjeta SD usando el bus SPI (a través de la cabecera ICSP). Para ello utiliza los pines digitales 10, 11, 12, y 13 en las placas Arduino/Genuino Uno y los pines 50, 51 y 52 en el Arduino ATmega2560. En ambas placas, el pin 10 se utiliza para seleccionar el chip W5100 y el pin 4 para la tarjeta SD. Estos pines no se pueden utilizar para otra entradas o salidas. (Figura 1.5 Módulo Arduino Ethernet Shield, Arduino/Genuino Uno y ATmega2560)

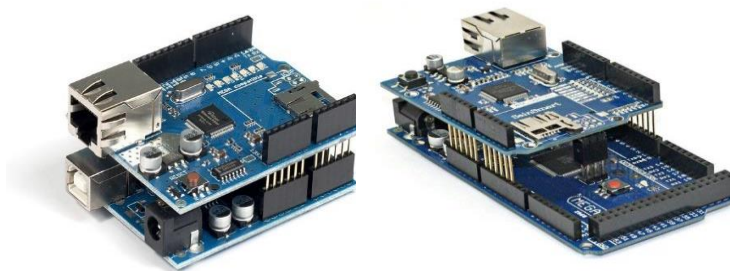


Figura 1.5 Módulo Arduino Ethernet Shield
Arduino/Genuino Uno y ATmega2560

Hay que tener en cuenta que debido a que la cuota del bus SPI del chip W5100 y la tarjeta SD, sólo uno puede estar activo en cada momento. Si está utilizando ambos periféricos en su programa, esto debe ser atendido por las bibliotecas correspondientes.

Si no se está utilizando uno de los periféricos en su programa, se tendrá que anular la selección explícitamente. Para hacer esto con la tarjeta SD, ajuste el pin 4 como salida y de un valor alto a la misma. Para el chip W5100, establezca pin digital 10 como una salida alta.

El botón de reinicio en el escudo restablece tanto el chip W5100 como la placa Arduino/Genuino.

El módulo Arduino Shield Ethernet contiene una serie de LEDs informativos:

- **PWR:** indica que la placa y el módulo están encendidos.
- **LINK:** indica la presencia de un enlace de red y parpadea cuando el módulo transmite o recibe datos.
- **FULLD:** indica que la conexión de red es full dúplex
- **100M:** indica la presencia de una conexión de red a Mb/s 100 (en contraposición a 10 Mb/s)
- **RX:** Parpadea cuando el escudo recibe datos
- **TX:** parpadea cuando el escudo envía datos
- **COLL:** parpadea cuando se detectan colisiones de red

2.4. Max6675 Modulo Interfaz para Termopar tipo K

Max6675

El MAX6675 es un convertidor Analógico a digital especializado para termopares tipo K. Con este módulo es posible conectar fácilmente un termopar a cualquier microcontrolador a través de una interfaz SPI unidireccional. Dentro de este pequeño circuito se encuentra la electrónica necesaria para amplificar, compensar y convertir a digital el voltaje generado por el termopar, lo que hace muy sencilla la tarea de conectar un termopar a un microcontrolador. (Figura 1.6 Módulo Max6675)



Figura 1.6 Módulo Max6675

El MAX6675 se conecta con un microcontrolador mediante una interfaz de 3 líneas compatible con el estándar SPI. El formato en el que el MAX6675 envía datos al microcontrolador es el siguiente. (Tabla 1.1 Módulo Max6675)

| BIT | DUMMY SIGN BIT | 12-BIT TEMPERATURE READING | | | | | | | | | | | THERMOCOUPLE INPUT | DEVICE ID | STATE | |
|-----|-------------------|-------------------------------|----|----|----|----|---|---|---|---|---|---|-----------------------|--------------|-------|-----------------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | 0 | MSB | | | | | | | | | | | LSB | | 0 | Three- state |

Tabla 1.1 Módulo Max6675

Termopar tipo K

El termopar tipo k es un sensor de temperatura analógico que no necesita de ser alimentado por un voltaje ya que es un sensor autogenerado. En general se define un termopar como un sensor de temperatura que suministra una señal de tensión eléctrica en función de la temperatura que registra. Un termopar no mide directamente las temperaturas, sino la diferencia de temperatura entre el extremo caliente y el extremo frío. La combinación de diferentes metales induce determinadas señales (Efecto Seebeck) que permite una medición eficaz de la temperatura. (Figura 1.7 Termopar tipo K).



Figura 1.7 Módulo Max6675

Entre todas las versiones el termopar tipo K es la versión más habitual de los termopares y consiste en las combinaciones de materiales NiCr-Ni o NiCr-NiAl. Son ideales para la utilización en atmósferas oxidantes o de gas inerte hasta 1200 °C (ASTM E230: 1260 °C). Debido a su elevada susceptibilidad a la oxidación es menor que la de otras versiones, se aplican sobre todo para temperaturas superiores a 550°C como se muestra en las siguientes tablas. (Tabla 1.2 Rangos de temperatura) y (Tabla 1.3 Limite de Error).

Rangos de Temperatura

| | | | | | |
|------------------------|-----------------|------------------|------------------|------------------|------------------|
| TERMOPAR TIPO K | 8 = 3.25 | 14 = 1.63 | 20 = 0.81 | 24 = 0.51 | 28 = 0.33 |
| K | ***** | 370 °C | 260 °C | 200 °C | 150 °C |

Tabla 1.2 Rangos de Temperatura

Límite de Error

| TERMOPAR TIPO K | TEMPERATURA °C | GRADO ESTANDAR | GRADO ESPECIAL |
|------------------------|-----------------------|-----------------------|-----------------------|
| K | 0 a 1250 | ±2.2 °C o ±0.75 °C | ±1.1 °C o ±0.4 °C |

Tabla 1.3 Limite de Error

2.5. Sensor de Humedad Relativa DHT11

El sensor DHT11, mide Humedad Relativa (HR). Los rangos de medida que maneja el sensor son; de 20-90% para el HR, con una precisión de $\pm 5\%$. El tiempo de respuesta puede variar en relación con la humedad ambiental, sin embargo, se puede establecer en un rango de 6 a 30 segundos. En relación con la calidad del producto destaca que cada uno de los sensores está calibrado, así también, la hoja de datos técnicos, lo refiere como estable, preciso y con una salida de señal digital. (Figura 1.8 Sensor DHT11).



Figura 1.8 Sensor DHT11

Características Técnicas

- Tamaño: 30 * 21mm .
- Peso: 4g.
- Voltaje de entrada: 3.3V, 5V.
- Puerto: digitales de dos vías solo bus.
- Humedad: 20-90%.
- HR: $\pm 5\%$ de humedad relativa.
- Plataforma: Arduino, MCU, ARM, Raspberry Pi.

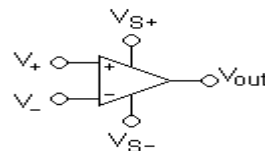
(Tabla 1.4 Características Técnicas).

| Pin | Nombre | Descripción |
|-----|--------|-------------------------------------|
| 1 | VDD | Alimentación de energía 3 – 5.5V DC |
| 2 | DATA | Datos de salida |
| 3 | NC | Sin conexión |
| 4 | GND | Conexión de tierra |

Tabla 1.4 Características Técnicas

2.6. Divisor de Tensión con Amplificador Buffer Opamp Im741 (Diferencia de Potencial)

Un amplificador operacional es un circuito electrónico (normalmente se presenta como circuito integrado) que tiene dos entradas y una salida. La salida es la diferencia de las dos entradas multiplicada por un factor (G) (ganancia): $V_{out} = G \cdot (V_+ - V_-)$.



Un seguidor de voltaje (también llamado amplificador buffer, amplificador de ganancia unitaria o amplificador de aislamiento) es un circuito amplificador operacional que tiene una ganancia de voltaje de 1. (Figura 1.10 Seguidor de Voltaje).

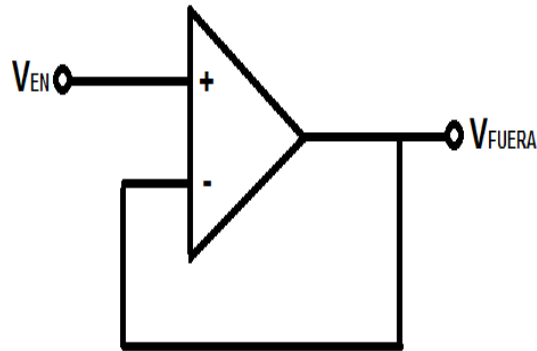


Figura 1.9 Alta impedancia

Esto significa que el amplificador operacional no proporciona ninguna amplificación a la señal. La razón por la que se llama un seguidor de tensión es porque el voltaje de salida sigue directamente el voltaje de entrada, significando que el voltaje de salida es igual que el voltaje de entrada. Así, por ejemplo, si 10V entra en el amplificador operacional como entrada, 10V sale como salida. Un seguidor de voltaje actúa como un buffer, no proporcionando ninguna amplificación o atenuación a la señal.

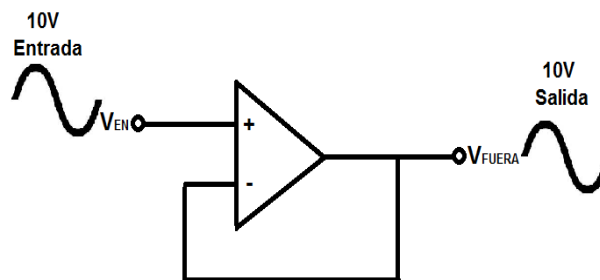


Figura 1.10 Seguidor de Voltaje

El Seguidor de Voltaje es muy importante en el Divisor de Tensión

Divisor de tensión o divisor resistivo:

Un divisor de tensión es una configuración de circuito eléctrico que reparte la tensión eléctrica de una fuente entre una o más impedancias conectadas en serie.

Un divisor de tensión es una configuración de circuito eléctrico que reparte la tensión eléctrica de una fuente entre una o más impedancias conectadas en serie, los dos resistores forman, un circuito serie cuya resistencia equivalente es.

$$R_T = R_1 + R_2 \quad \text{Ec. 1.1 Resistencia Total}$$

Dada esta configuración, la intensidad de corriente que circulará por este circuito será.

$$I = \frac{V_E}{R_T} \quad I = \frac{V_1}{(R_1 + R_2)} \quad \text{Ec. 1.2 Corriente}$$

Observa que la tensión en la salida (V_s), es la tensión que cae en R_2 . Por lo que V_s será igual a.

$$V_s = I * R_2 \quad V_s = \frac{V_E \cdot R_2}{(R_1 + R_2)} \quad \text{Ec. 1.3 Voltaje de Salida}$$

El circuito se usa como "buffer" para aislar una sección de un circuito de otra. Sirve por tanto para minimizar efectos del circuito sobre la fuente (alterar muy poco una medición y para acoplar impedancias). Considere por ejemplo el circuito de la. (Figura 2.1 Divisor de Tensión con Divisor de Voltaje).

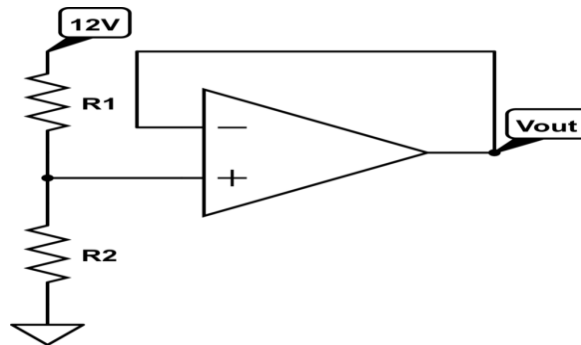


Figura 2.1 Divisor de Tensión con Seguidor de Voltaje

2.7. Sensor de Efecto Hall WCS1800-35 Ampers.

El sensor de corriente Winston WCS1800 proporciona precisión y solución para detección de corriente AC Y DC en aplicaciones industriales, comerciales y sistemas de comunicaciones. El paquete único proporciona una implementación fácil sin romper el sistema original y hace posible la detección de corriente. las aplicaciones incluyen control de motor, detección y gestión de carga, detección de fallas por sobre corriente y cualquier sistema inteligente de administración de energía.

El WCS1800 consiste en una sala lineal derivada de baja temperatura con compensación de la misma y un diámetro de 9.0 mm a través del orificio. Los usuarios pueden usar el propio cable eléctrico del sistema pasándolo a través de este agujero para medir la corriente de paso. (Figura 2.2 Sensor de Corriente WCS-1800)

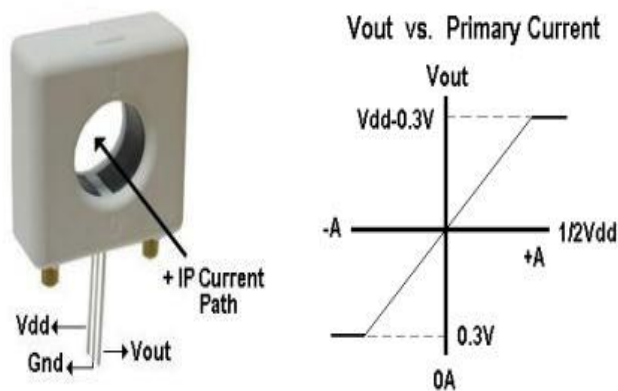


Figura 2.2 Sensor de Corriente WCS-1800

Este diseño permite a los diseñadores de sistemas monitorear cualquier ruta actual sin romper o cambiar el diseño original del sistema en absoluto. Cualquier corriente que fluya a través de este agujero generará un campo magnético que es detectado por el efecto Hall del IC. (Figura 2.3 Conexión del Circuito)

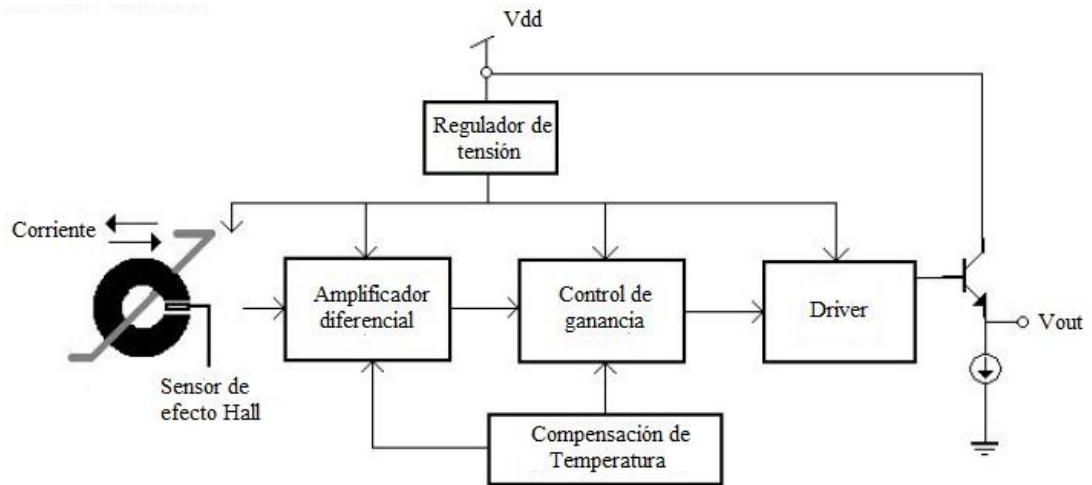
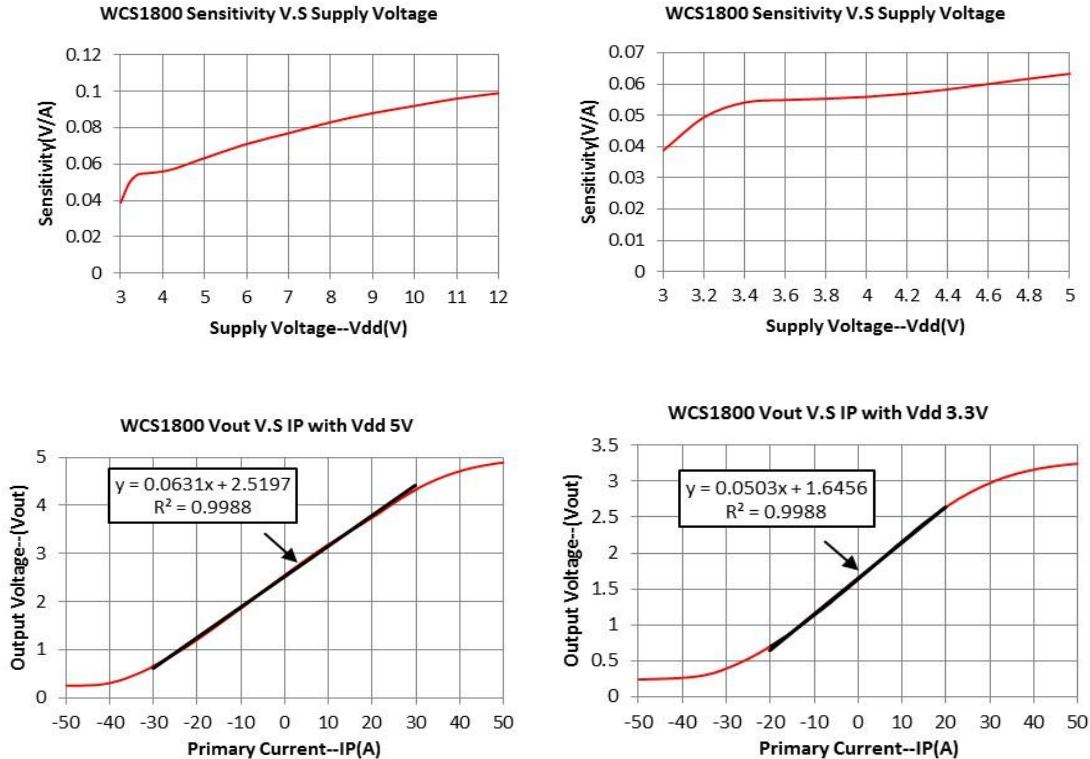


Figura 2.3 Conexión del circuito

Características:

- Diámetro 9.0mm conductor a través del orificio
- Voltaje de salida proporcional a la corriente CA y DC
- Rango de corriente de detección amplia 0 ~ 35 A a 5V voltios.
- Alta sensibilidad 66mV / A
- Amplio rango de tensión de funcionamiento de 3.0 ~ 5VDC.
- Baja corriente de funcionamiento 3mA
- Tensión de aislamiento 4000V
- Salida radiométrica a partir de la tensión de alimentación
- Ancho de banda de 23 KHz
- Salida Digital y Analógica.

Características del Diagrama.



Grafica 1.1 Características del Diagrama

Factor de Corrección.

Se hizo un factor de corrección al sensor de corriente WCS-1800 HW-671 para que se pudiera establecer en las mediciones, como se muestra en la gráfica de arriba (Grafica 1. Características del diagrama), que al tener 0, las cuentas de resolución del Arduino estarían en 512 y al llegar a los 35 A, que es su máxima corriente de medición del sensor, la resolución estaría en 1024, pero en este caso variaba mucho las mediciones, para poder establecer correctamente la corriente a medir, se hicieron las siguientes ecuaciones.

$$P = VI$$

$$P = V * V/R \quad \text{Ec. 1.4 Factor de Corrección}$$

$$P = V^2/R$$

$$R = V^2/P$$

En estas fórmulas se calcularon el voltaje que suministra cada batería como también la corriente para tener una potencia adecuada y después así sacar su valor de la resistencia para poder tener un mejor resultado.

En esta ecuación, la (Y) significa el valor que está midiendo el sensor en mili-voltios, la (X) es la pendiente de la recta que tiene un rango el sensor de corriente, y la (C) es la constante del valor que medirá, para poder sacar los valores dependiendo el número de cuentas en el Arduino es donde ira muestreando los valores medidos.

$$Y = CX + E \quad \text{Ec. 1.5 Constante de la pendiente}$$

Teniendo en cuenta, los valores con esta ecuación podemos calcular la recta pendiente que va de -35 a 35 amperes.

$$C = \frac{X2 - X1}{Y2 - Y1} \quad \text{Ec. 1.6 Recta de la pendiente}$$

Conexión con Arduino.

La conexión resulta sencilla gracias al módulo, Arduino tiene salida de 5 V y de tierra que será VDD y GND correspondiente al sensor. Por último, conectaremos el pin de salida de señal del sensor (0V a 5V DC) al pin de lectura analógica de Arduino (pin ADC – A0). (Figura 2.4 Conexión con Arduino).

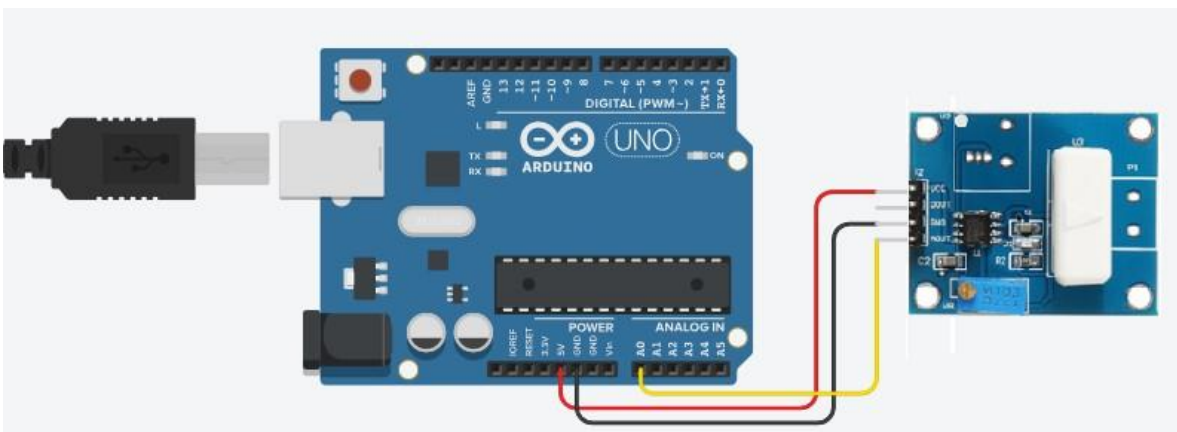


Figura 2.4 Conexión con Arduino

2.8. Módulo de Reloj

El DS3231 es un reloj en tiempo real de alta exactitud que cuenta con un oscilador a cristal con compensación de temperatura (TCXO). El dispositivo incorpora una entrada para batería auxiliar y mantiene la cuenta precisa del tiempo incluso cuando la energía es interrumpida. La integración del cristal en el propio integrado asegura la exactitud a largo plazo del reloj. (Figura 2.5 Módulo de Reloj).

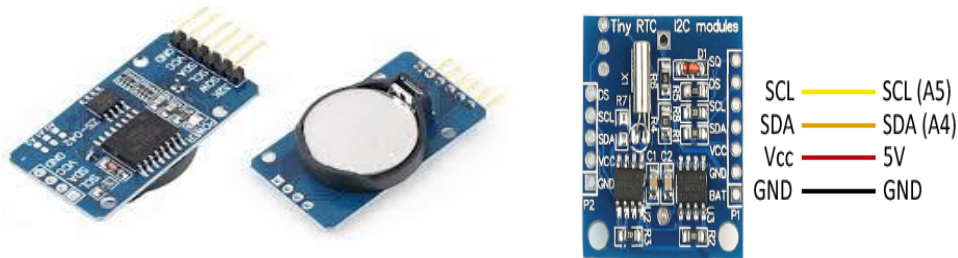


Figura 2.5 Módulo de Reloj

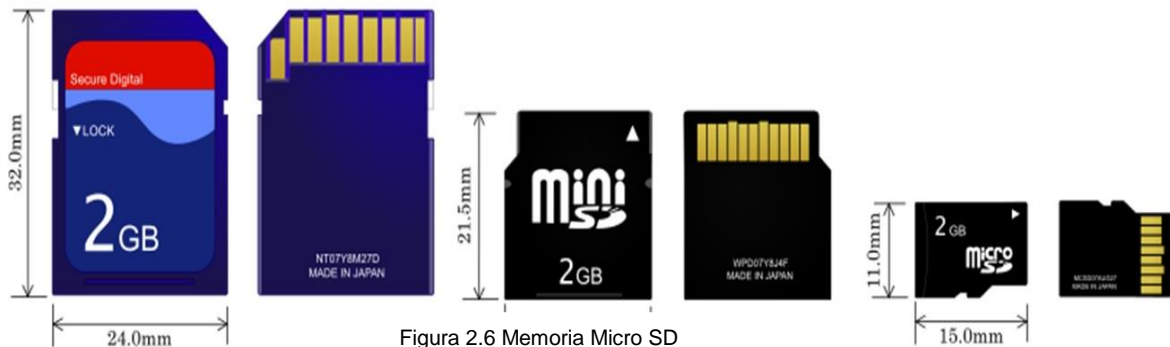
Características DS3231

- RTC Alta exactitud, maneja todas las funciones para el mantenimiento de fecha/hora
- Exactitud de ± 2 ppm operando a una temperatura de 0°C a $+40^{\circ}\text{C}$
- Módulo cuenta con reloj DS3231 y memoria EEPROM I2C
- Cuenta con batería de respaldo
- Registro de segundos, minutos, horas, día de la semana, fecha, mes y año con compensación de años bisiestos hasta 2100
- El DS3231 Incluye sensor de temperatura con exactitud de ± 3 grados centígrados
- alarmas programables por hora/fecha
- Salida de señal cuadrada programable

2.9. Memoria Micro SD

Las memorias SD son las más usadas por dispositivos portátiles, por su gran capacidad y su reducido tamaño, debido a su gran demanda son fáciles de conseguir en diferentes capacidades y precios. Estas características nos dan una buena alternativa de almacenamiento para usarlo en Arduino, sobre todo cuando necesitamos guardar gran cantidad de información.

Estas memorias vienen en tres tamaños, SD estándar, Mini SD y Micro SD, siendo este último el tamaño más común, funcionalmente son iguales, pudiéndose usar adaptadores para utilizarlos en sockets de diferente tamaño. (Figura 2.6 Memoria Micro SD).



2.10 Pantalla LCD Display Blue 20x4

Display LCD de 4 líneas de 20 caracteres con comunicaciones I2C, serie y retroiluminación controlable por software. Gracias a su iluminación posterior, los mensajes se ven claramente incluso en total oscuridad. (Figura 2.7 Pantalla LCD Display Blue 20x4).

Principales características:

- Pantalla – 20 caracteres por línea, 4 líneas.
- Alimentación: 5 V.
- Consumo: 125mA, con retroiluminación de pantalla encendida.
- Comunicaciones: I2C o serial a 9600 bps.
- Teclado: Escaneo automático de teclado matricial 3x4.

- Caracteres personalizados: hasta 8 caracteres personalizados fácilmente definidos.

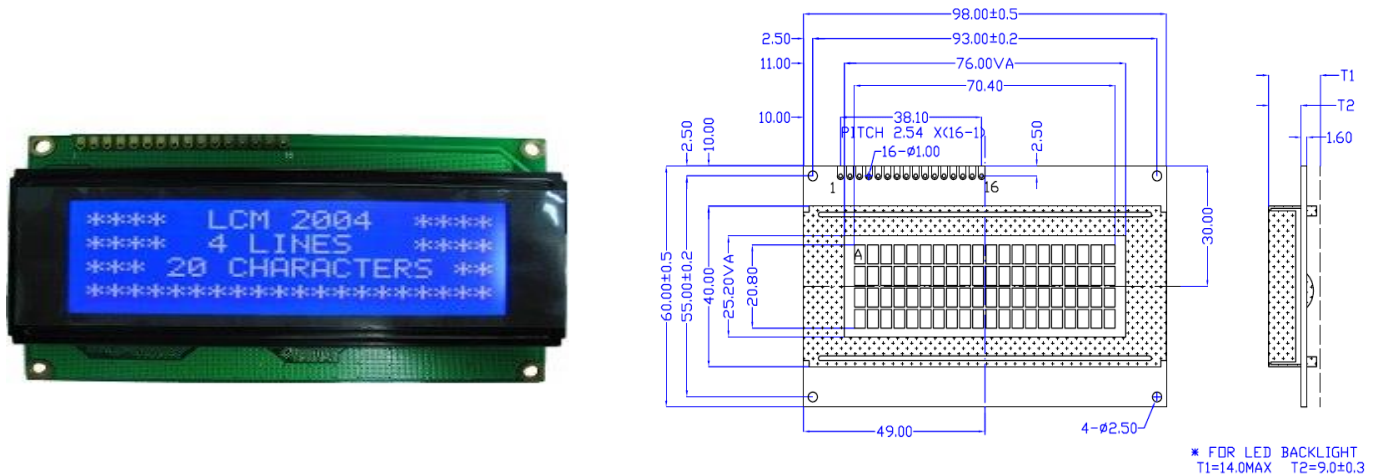


Figura 2.7 Pantalla LCD Display Blue 20x4

2.11 Regulador de Voltaje de 48-54 Volts a 12 Volts

Se trata de una potencia elevada, en el cual se debe tener en cuenta, aquellos elementos que, puedan manejar potencias arriba de los 60 V, cableados de sección adecuada, semiconductores capaces de soportar tensiones y corrientes elevadas superiores a las que tengan que entregar normalmente. Por ese motivo he decidido tratar en parte este aspecto en esta fuente de alimentación.

Después de revisar varios artículos sobre fuentes de alimentación, algunos con resultados satisfactorios por su calidad/precio. Se requiere para este proyecto (Banco de Baterías) una fuente de tensión, cuyas características son de 48-54 volts a 12 volts de salida y una corriente máxima de 1 A.

Naturalmente los 48-54V de entrada requieren de unos cálculos concretos, esto pasa por obtener tensión de 60V y con una carga de 1 A, la potencia mínima de 140W y por seguridad se utilizó una potencia de 200W.

El circuito electrónico está basado en el circuito integrado LM723, un regulador muy completo del que tantos proyectos se han desarrollado. No obstante, en esta ocasión vamos a basarnos en una versión capaz de soportar cerca de los 100Volts, como se puede apreciar en la figura siguiente: (Figura 2.8 Esquema de partida).

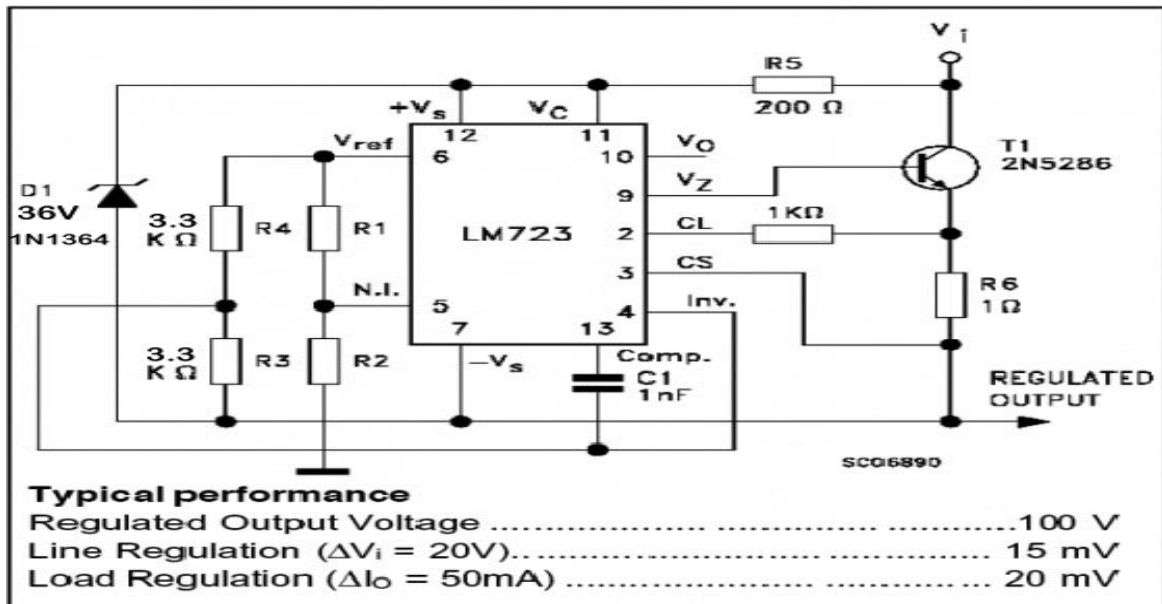


Figura 2.8 Esquema de partida

Hay que tener en cuenta que el fabricante del circuito integrado LM723, recomienda los siguientes parámetros: (Tabla 1.5 parámetros del LM723).

recommended operating conditions

| | MIN | MAX | UNIT |
|---|-----|-----|-------|
| Input voltage, V_i | 9.5 | 40 | V |
| Output voltage, V_o | 2 | 37 | V |
| Input-to-output voltage differential, $V_C - V_o$ | 3 | 38 | V |
| Output current, I_o | | 150 | mA |
| Operating free-air temperature range, T_A | | 0 | 70 °C |

Tabla 1.5 Parámetros del LM723

2.12 Regulador de Voltaje de 12 Volts a 5 Volts

Este es un regulador de tensión de entrada son 12 voltios y la tensión de salida es de 5 voltios que me servirá para poder alimentar mis sensores y demás componentes como también el Arduino, es muy fácil de a ser, pero eso si se necesita calcular bien los valores de los capacitores para que te de la tensión de salida correcta. (Figura 2.9 Regulador de Voltaje 7805).

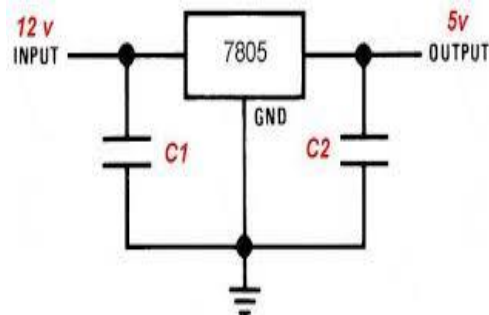


Figura 2.9 Regulador de Voltaje de 12 Volts a 5 Volts

Especificaciones del regulador LM7805.

(Tabla 1.6 Parámetros del LM7805)

| Parameter | Symbol | Conditions | MC7805/LM7805 | | | Unit | |
|--------------------------|---------------------|---|---------------------------------|------|------|-------------------|----|
| | | | Min. | Typ. | Max. | | |
| Output Voltage | V _O | T _J = +25 °C | 4.8 | 5.0 | 5.2 | V | |
| | | 5.0mA ≤ I _O ≤ 1.0A, P _O ≤ 15W V _I = 7V to 20V | 4.75 | 5.0 | 5.25 | | |
| Line Regulation (Note1) | Regline | T _J = +25 °C | V _O = 7V to 25V | - | 4.0 | 100 | mV |
| | | | V _I = 8V to 12V | - | 1.6 | 50 | |
| Load Regulation (Note1) | Regload | T _J = +25 °C | I _O = 5.0mA to 1.5A | - | 9 | 100 | mV |
| | | | I _O = 250mA to 750mA | - | 4 | 50 | |
| Quiescent Current | I _Q | T _J = +25 °C | - | 5.0 | 8.0 | mA | |
| Quiescent Current Change | ΔI _Q | I _O = 5mA to 1.0A | - | 0.03 | 0.5 | mA | |
| | | V _I = 7V to 25V | - | 0.3 | 1.3 | | |
| Output Voltage Drift | ΔV _O /ΔT | I _O = 5mA | - | -0.8 | - | mV/°C | |
| Output Noise Voltage | V _N | f = 10Hz to 100KHz, T _A = +25 °C | - | 42 | - | μV/V _O | |
| Ripple Rejection | RR | f = 120Hz V _O = 8V to 18V | 62 | 73 | - | dB | |
| Dropout Voltage | V _{Drop} | I _O = 1A, T _J = +25 °C | - | 2 | - | V | |
| Output Resistance | r _O | f = 1KHz | - | 15 | - | mΩ | |
| Short Circuit Current | I _{SC} | V _I = 35V, T _A = +25 °C | - | 230 | - | mA | |
| Peak Current | I _{PK} | T _J = +25 °C | - | 2.2 | - | A | |

Tabla 1.6 Parámetros del LM7805

CAPITULO III DESARROLLO E IMPLEMENTACIÓN DEL PROYECTO

3.1 Descripción del Sistema de Fuerza del Área de Transmisión, Zona Tuxtla; incluyendo un diagrama de bloques de todas las partes que lo conforman el sistema

Sistema modular de fuerza

El Sistema de Fuerza de 48 VCE tiene integrado un Banco de Baterías selladas libres de mantenimiento (FRONT TERMINAL) de 48-54VCD y capacidad total de 360AH., capaz de soportar una carga de 73.0 A durante un lapso de 5.00 horas aproximadamente. El sistema cuenta con una etapa de corte por bajo voltaje para proteger a las baterías de una descarga profunda. Está conformado por un módulo de control, panel de distribución de carga (módulo de distribución) y módulo de entrada y salida para alarmas.

Módulo de entrada, salida y alarmas

Este módulo contiene el interruptor general de entrada de C.A. y también está localizado el interruptor de entrada de C.A. (Auxiliar). Cuenta con las terminales de salida de C.A. del sistema y terminales de entrada y salida de alarmas, también se encuentra alojada la barra de tierra y el supresor de transitorios. (Figura 3.1 Modulo de Entrada, Salida y Alarmas).

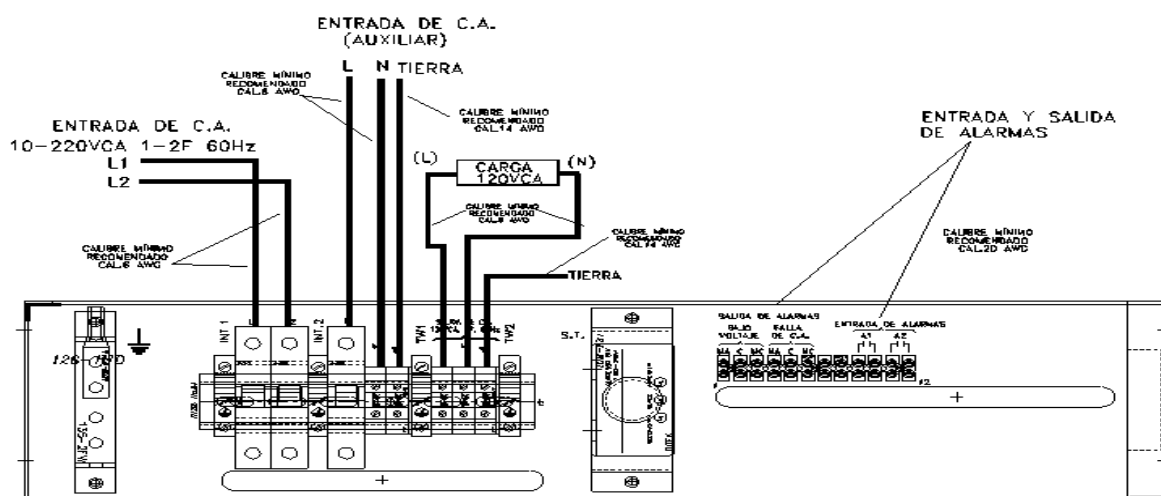


Figura 3.1 Modulo de Entrada, Salida y Alarmas

Módulo de distribución

En este módulo se encuentran alojados los interruptores termomagnéticos de Distribución de C.D., así como el interruptor del inversor y el del Banco de Baterías, así como las terminales de conexión. (Figura 3.2 Modulo de Distribución).

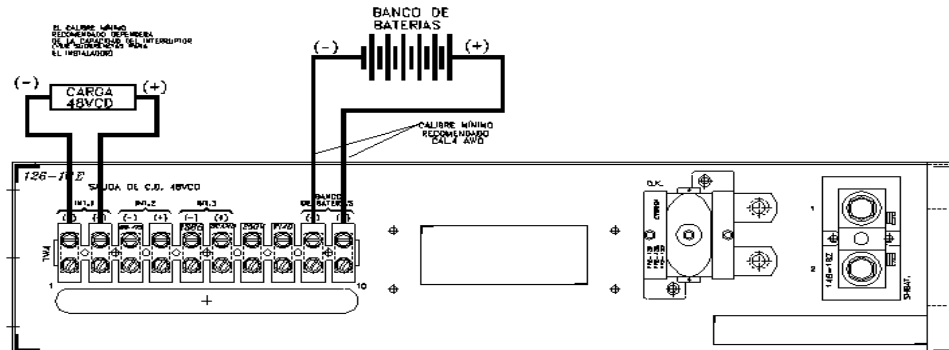


Figura 3.2 Modulo de Distribución

Banco de baterías

El sistema de Fuerza cuenta con un banco de baterías selladas libres de mantenimiento (48-54VDC, 360AH) capaz de respaldar una carga de 73.0Amps durante 5.00 horas.

Todos los módulos forman parte del Sistema de Fuerza de 48 VCD, está en un gabinete metálico de Acero al Carbón para montaje y aseguramiento de los equipos.

El sistema está instalado en la parte inferior y ocupa una altura máxima de 23 pulgadas.

El acceso y conexiones del Sistema de Fuerza será por la parte posterior del gabinete; para cambiar o reemplazar alguna batería se deberá quitar en la parte frontal. (Figura 3.3 Banco de Baterías).

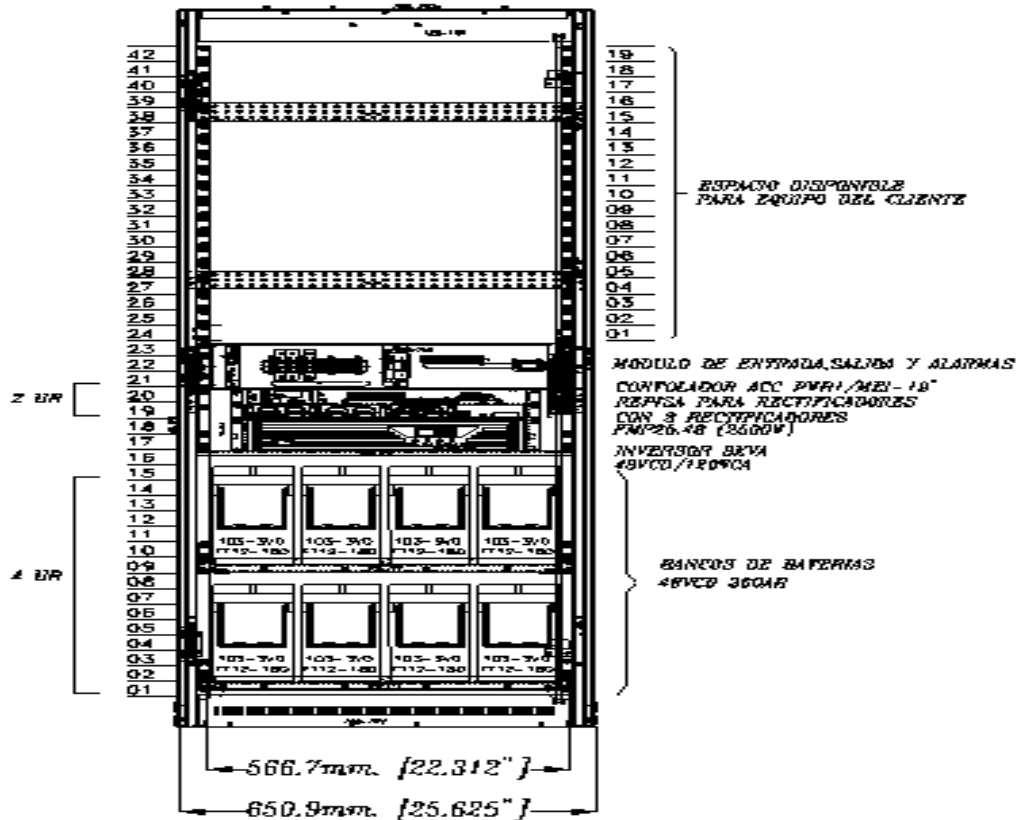


Figura 3.3 Banco de Baterías

Diagrama de Bloques de todo el Sistema de Fuerza



Figura 3.4 Diagrama de bloques de todo el Sistema de Fuerza

En este diagrama a continuación se muestra como va clasificado todas las etapas que conforman el Sistema de Fuerza como se ve en la siguiente figura 3.4

3.2 Selección de Materiales a Utilizar y los Costos de cada Componente; incluyendo un diagrama de bloques del nuevo Sistema de Fuerza

Para la selección de materiales era necesario saber cuáles eran los puntos claves para la verificación de las variables en el Banco de Baterías y posteriormente definir qué variables son las más importantes, variables que trabajen en perfecto estado para evitar daño en los equipos y con esto disminuir los gastos excesivos por mantenimiento o por pérdida completa de la batería, debido a que los equipos que forman parte de este tipo de sistema de fuerza son bastantes costosos. Por ello, surgió la idea de diseñar un sistema de monitoreo en las dos celdas que conforman el banco de baterías.

Se determinó que las variables más importantes de monitorear para dicho sistema, fueran las siguientes:

- Temperatura
- Voltaje
- Corriente
- Temperatura y Humedad en la Habitación

Además de monitorear las variables especificadas, el sistema debe operar como real unidad de control para ello se consideró que la información obtenida de las variables monitoreadas se pudiera guardar en una memoria micro SD y después enviarlas a la plataforma ZABBIX que tienen a su disposición por medio de la red Wifi.

Para que el sistema funcionara de acuerdo a las condiciones que se deseaba se eligió utilizar dos tarjetas de Arduino, la tarjeta Arduino ATmega2560 y Ethernet Shield, las cuales proporcionarían para el proyecto suficientes entradas analógicas y digitales, además que maneja protocolos I2C, comunicación bus tipo SPI la comunicación entre las dos tarjetas Arduino, Micro SD para guardar las mediciones de las variables obtenidas.

El protocolo I2C es un bus de comunicaciones en serie que utiliza dos líneas para transmitir la información; una para los datos y otra para señal del reloj. Para el caso del Arduino mega los pines SDA Y SCL se localizan en las terminales 20, y 21 y los otros dos se encuentran a continuación en la terminal AREF, y en el Arduino Ethernet Shield solo se encuentran en AREF. Y la comunicación SPI permite alcanzar velocidades muy altas para enviar y recibir datos al mismo tiempo ya que trabaja de manera síncrona con líneas de datos en tiempo real, como también nos sirve para la comunicación entre las dos tarjetas Arduino mega y la Ethernet Shield como se muestra en la siguiente figura 3.5.

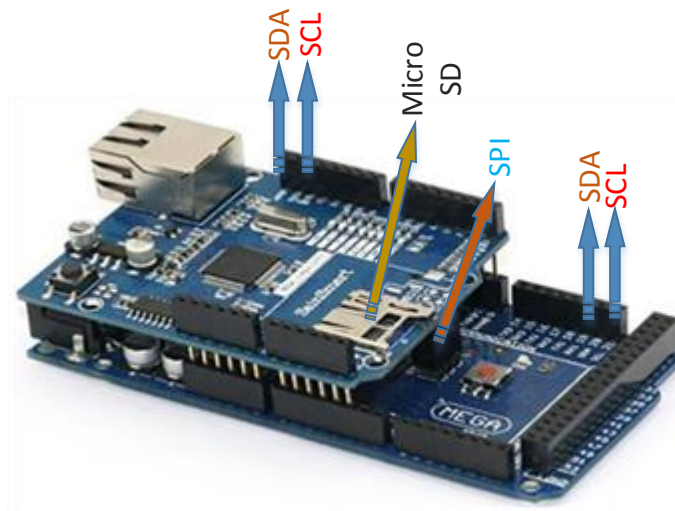


Figura 3.5 Comunicación entre las dos tarjetas Arduino

Se estableció las dos tarjetas Arduino, ATmega2560 y Ethernet Shield como placa de CPU para tener una comunicación entre las dos tarjetas y así guardar, enviar los datos que se obtuvieran de la lectura de los sensores a la plataforma ZABBIX mediante la red de Wifi.

Servidor

El servidor realiza el sondeo y captura de datos, calcula disparadores, envía notificaciones a los usuarios. Es el componente central al que los agentes y proxies de ZABBIX informan datos sobre disponibilidad e integridad de sistemas.

El servidor puede revisar remotamente los servicios en red (como servidores web y servidores de correo) mediante comprobaciones de servicio simples.

El servidor es el repositorio central en el que se almacenan todos los datos de configuración, estadísticos y operativos, además funge como entidad que alertará activamente a los administradores cuando surjan problemas en cualquiera de los sistemas supervisados.

Diagrama de Bloques del nuevo Sistema de Fuerza

El presente proyecto, tiene como objetivo diseñar e implementar un sistema que permita medir las variables de funcionamiento del banco de baterías, tales como: Corriente eléctrica, diferencia de potencial y temperatura. Para esto, se procedió a investigar la estructura y funcionamiento del banco de batería; con base a los resultados de esta investigación, se propuso que el sistema de medición podía ser implementado como se muestra en el diagrama de bloques de la Fig.3.6.

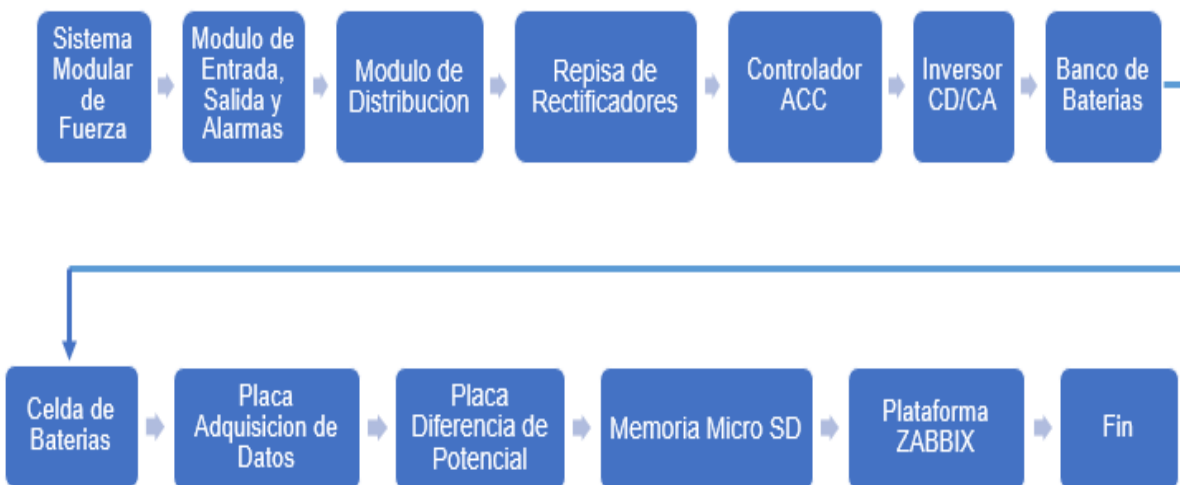


Figura 3.6 Diagrama de Bloques del nuevo Sistema de Fuerza

Después de hacer un estudio de todos los materiales a utilizar, se agregaron en una tabla, con su nombre, modelo y cantidad de los cuales se indican a continuación Tabla 1.7 Materiales a usar para el Banco de Baterías.

Materiales a usar para el Banco de Baterías

| NOMBRE | MODELO | CANTIDAD | COSTO DEL COMPONENTE | COSTO TOTAL DE LOS COMPONENTES |
|--------------------------------------|-------------------------------|----------|----------------------|--------------------------------|
| ARDUINO ETHERNET | SHIELD W510 | 1 | \$199.00 | \$199.00 |
| ARDUINO MEGA | ATmega2560 | 1 | \$320.00 | \$320.00 |
| SENSOR DE EFECTO HALL | WCS-1800 35 A | 2 | \$450.00 | \$900.00 |
| PANTALLA LCD DISPLAY BLUE LIGHT 20X4 | HD44780 | 1 | \$135.00 | \$135.00 |
| SENSOR DE TEMPERATURA | TERMOPAR K CON MODULO MAX6675 | 5 | \$130.00 | \$650.00 |
| MODULO DE RELOJ | DS3231 | 1 | \$50.00 | \$50.00 |
| MODULO DE ADAPTADOR LCD | I2C | 1 | \$35.00 | \$35.00 |
| MICRO SD 2GB | SANDISK | 1 | \$40.00 | \$40.00 |
| SENSOR DE HUMEDAD RESISTIVO | DHT11 | 1 | \$45.00 | \$45.00 |
| AMPLIFICADOR OPERACIONAL | LM741 | 10 | \$7.00 | \$70.00 |
| TIRA DE 40 PINES | HEADER MACHO 2.54MM | 2 | \$5.00 | \$10.00 |
| RESISTENCIAS | 10 MEGA OHMS Y 1 MEGA OHMS | 50 | \$2.00 | \$100.00 |

| | | | | |
|-----------------|--|----|---------|-------------------|
| RESISTENCIAS | 220-K-OHMS, 100K-OHMS, 10K-OHMS Y 4.7 K-OHMS | 10 | \$1.00 | \$10.00 |
| TERMINAL BLOCK | 2 TORNILLOS | 18 | \$3.00 | \$54.00 |
| TERMINAL BLOCK | 3 TORNILLOS | 12 | \$4.00 | \$48.00 |
| CLORURO FERRICO | 250 ML CIRCUITOS IMPRESOS | 2 | \$35.00 | \$70.00 |
| PLACA FENOLICA | 20X15 | 2 | \$50.00 | \$100.00 |
| TOTAL | | | | \$2,836.00 |

Tabla 1.7 Materiales a usar para el Banco de Baterías.

3.3 Diseño de Circuitos en PCB WIZARD

Con la ayuda del Software PCB WIZARD, se crearon tres diseños de placas, la primera etapa es la del Divisor de Tensión con alta impedancia (sensor de voltaje), la segunda etapa es la de adquisición de datos y la tercera es la fuente regulable de 48-54VCD a 12VOLTS.

3.4 Simulaciones en Proteus

Con la ayuda del Software Proteus versión 8.5, se hicieron todas las simulaciones de los sensores mencionados junto con el Arduino, en la simulación de cada etapa mencionada se fue introduciendo el programa para ver la reacción de cada sensor, antes de llevar acabo la realización de los circuitos físicamente.

Simulación del divisor de tensión con alta impedancia (Diferencia de Potencial)

La etapa de Divisor de Tensión es de suma importancia para la medición de voltaje de cada Batería, por que obtenemos mejores resultados con mayor precisión sin que la Batería pueda obtener algún daño. (Figura 3.7 Placa de Divisor de Tensión con alta Impedancia).

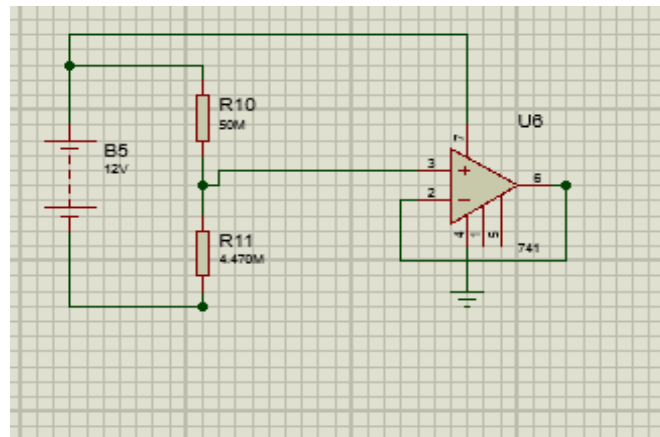


Figura 3.7 Placa de Divisor de Tensión con alta Impedancia

En la figura 3.8 Se muestra la etapa de simulación del Divisor de Tensión con Alta Impedancia, se utilizó el circuito integrado Amplificador Operacional UA741, cinco resistencias de 10MOHMS, cuatro resistencias de 1MOHMS y una resistencia de 470K.

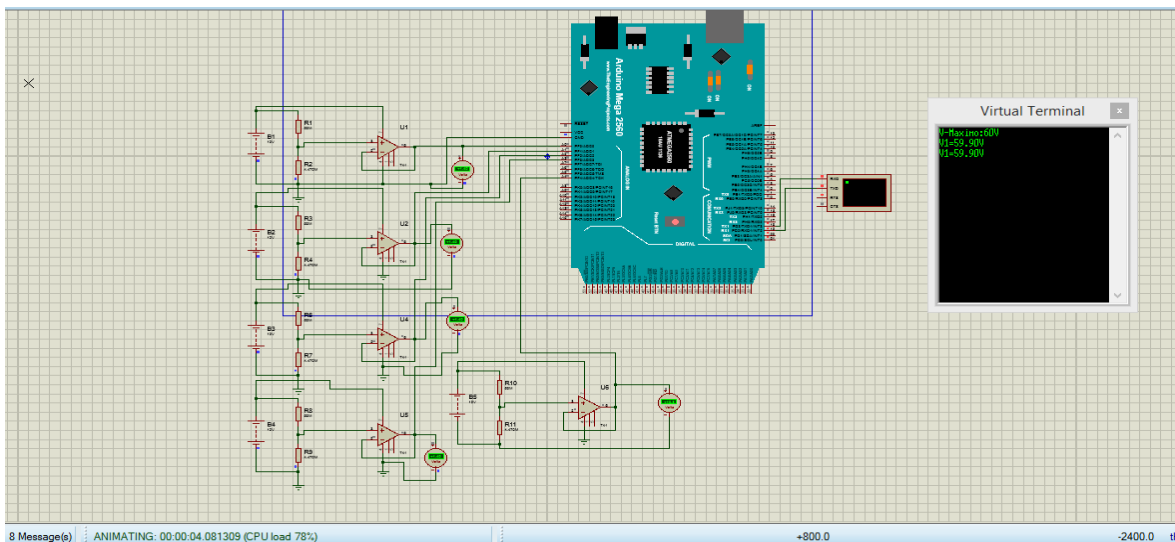


Figura 3.8 Simulación de las 4 Entradas Analógicas

En la simulación se puede apreciar los voltajes leídos en las entradas Analógicas por parte del Divisor de Tensión en A0, A1, A2, A3, A4 donde están midiendo el voltaje de cada Batería que hay en el Sistema de Fuerza. Debido al tipo de sensor que se manejó para la detección de voltaje, se utilizó los pines mencionados anteriormente en las entradas analógicas del Arduino ya que se representan a través de una conversión en mili-voltios, se hizo la siguiente programación.

```

int analogInput1 = A0;

float Vinput = 4.924;
float vout1 = 0.0;
float vin1 = 0.0;
float R1 = 49940000;
float R2 = 4400000;
int value1 = 0.0;

void setup(){
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.init(); // initialize the lcd
  lcd.init();
  lcd.backlight();
  lcd.clear();
  pinMode(analogInput1, INPUT);
  lcd.print("V-Maximo:");
  lcd.print((int)(Vinput / (R2 / (R1 + R2))));
  lcd.println("V");
}

void loop(){
  value1 = analogRead(analogInput1);
  vout1 = (value1 * (4.86 / 1024.0));
  vin1 = vout1* (R1 + R2) / R2 ;

  if (vin1<0.0043){
    vin1 = 0.0;
  }
}

```

Figura 3.9 Código del Divisor de Tensión (Sensor de Voltaje)

Para la prueba del Divisor de Tensión diseñado con los circuitos integrados OPAM UA741, se realizó el siguiente circuito físicamente en un protoboard para comprobar si este circuito funcionaba de manera adecuada como se deseaba, con la ayuda de una fuente de alimentación de 12 V externa y una pantalla LCD 16x2 conectada al Arduino como se observa en la Figura 3.10 Prueba del Divisor de Tensión.

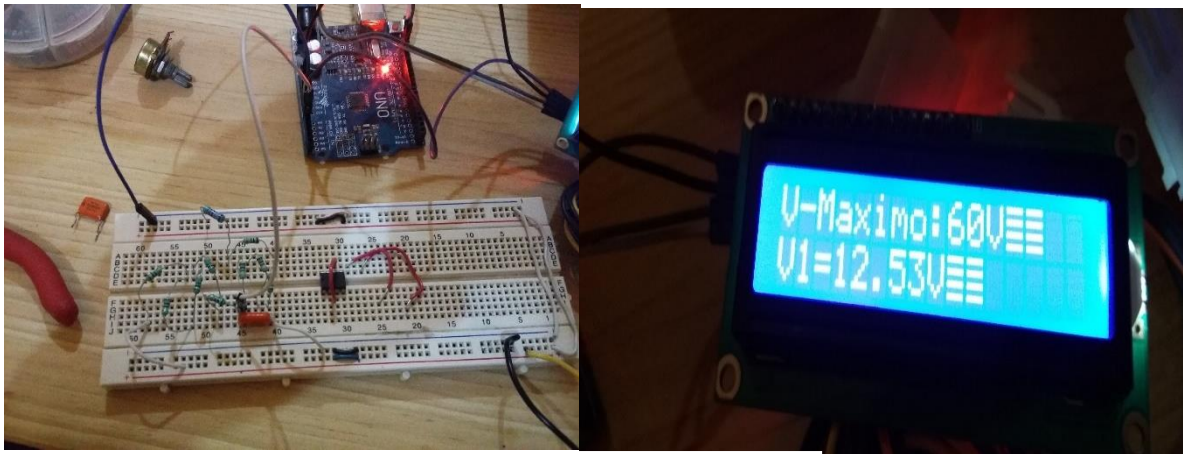


Figura 3.10 Prueba del Divisor de

Simulación del sensor de temperatura, termopar modulo max6675 tipo k

El termopar es de suma importancia en las mediciones de temperatura, porque es de mejor precisión y puede medir altas temperaturas, además de eso, es un sensor tipo industrial. Donde optamos en utilizarlo al Sistema de Fuerza, para medir la temperatura a cada batería.

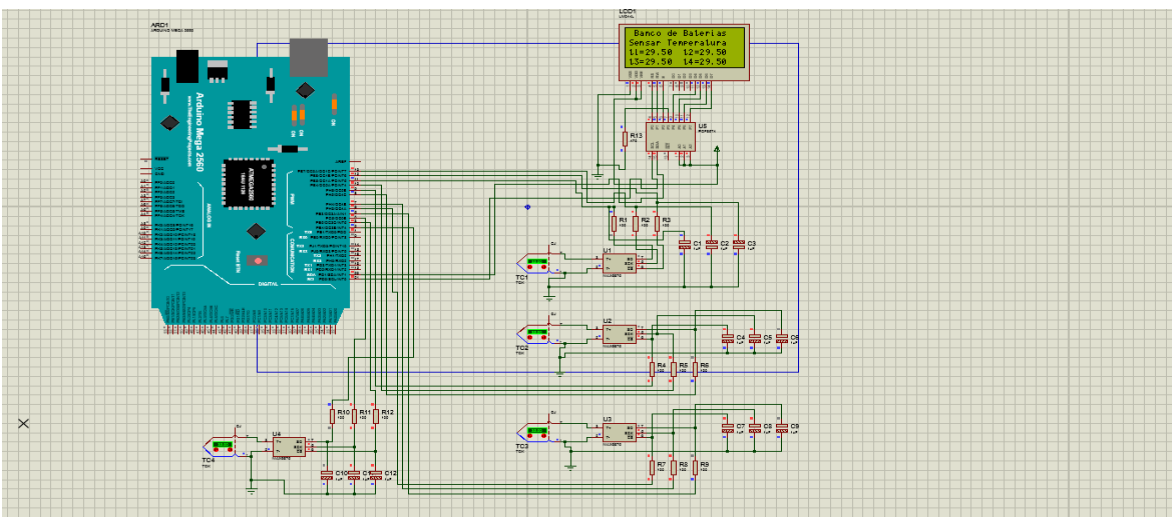


Figura 3.11 Simulación del Termopar Con Modulo Max6675 Tipo k

El diagrama de las conexiones del Sensor de Temperatura (Termopar Modulo Max6675 Tipo K) con Arduino Mega2560 como se muestra en la Figura 3.11 con sus respectivas conexiones.

Simulación del sensor dht11 humedad-temperatura ambiente

El sensor DHT11 nos sirve para saber el porcentaje de humedad y los grados de temperatura que hay en el ambiente, por eso es de gran importancia, para el Banco de Baterías para saber la humedad y temperatura que tienen externamente. En la siguiente Figura 3.12 y 3.13 nos muestra las respectivas conexiones y el código que se utilizó.

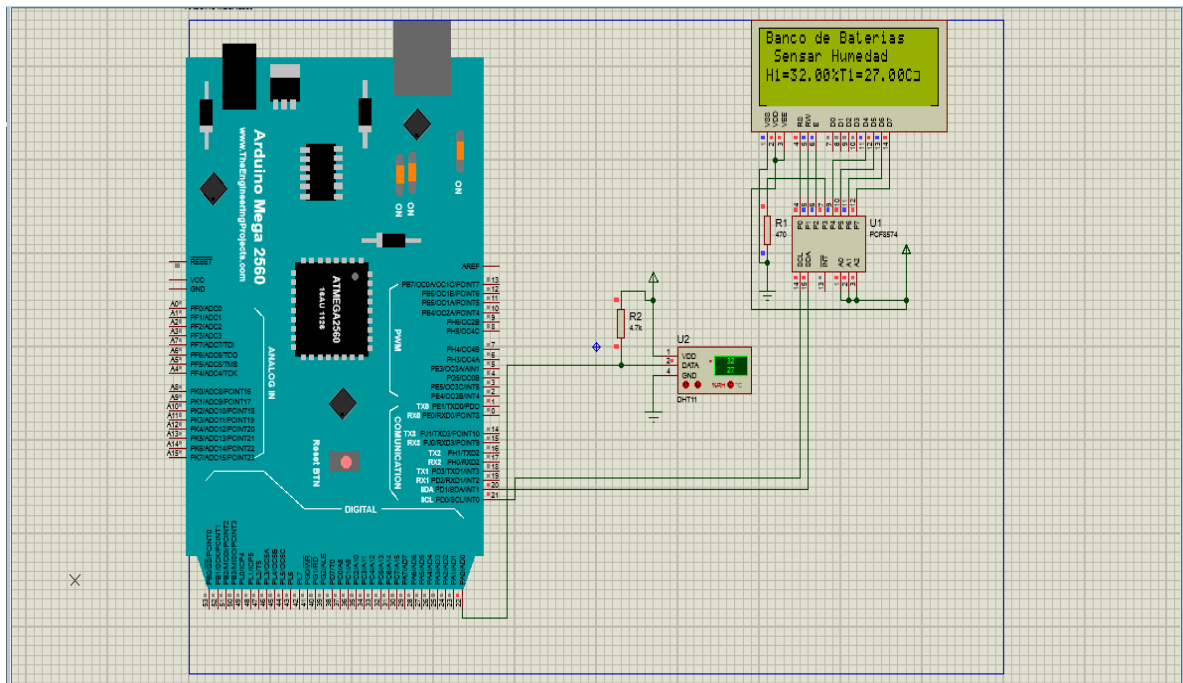


Figura 3.12 Simulación del sensor DHT11 con sus respectivas conexiones

```

void setup(){
  Serial.begin(9600);
  dht1.begin();
  lcd.begin(20, 4);
  lcd.clear();
  lcd.init();          // initialize the lcd
  lcd.init();
  lcd.backlight();
}
void loop(){

  float h1 = dht1.readHumidity();
  float t1 = dht1.readTemperature();

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Banco de Baterias");
  lcd.setCursor(0, 1);
  lcd.print(" Sensor Humedad");
  lcd.setCursor(0, 2);
  lcd.print("H1=");
  lcd.print(h1);
  lcd.print("%");
  lcd.setCursor(9, 2);
  lcd.print("T1=");
  lcd.print(t1);
}

```

Figura 3.13 Código del Sensor DHT11

Simulación de la Fuente Regulable de 48-54 a 12 Volt y 12 a 5 Volts

Esta etapa proporcionará la alimentación de los circuitos integrados OPAM UA741 y los componentes de la etapa de Adquisición de Datos.

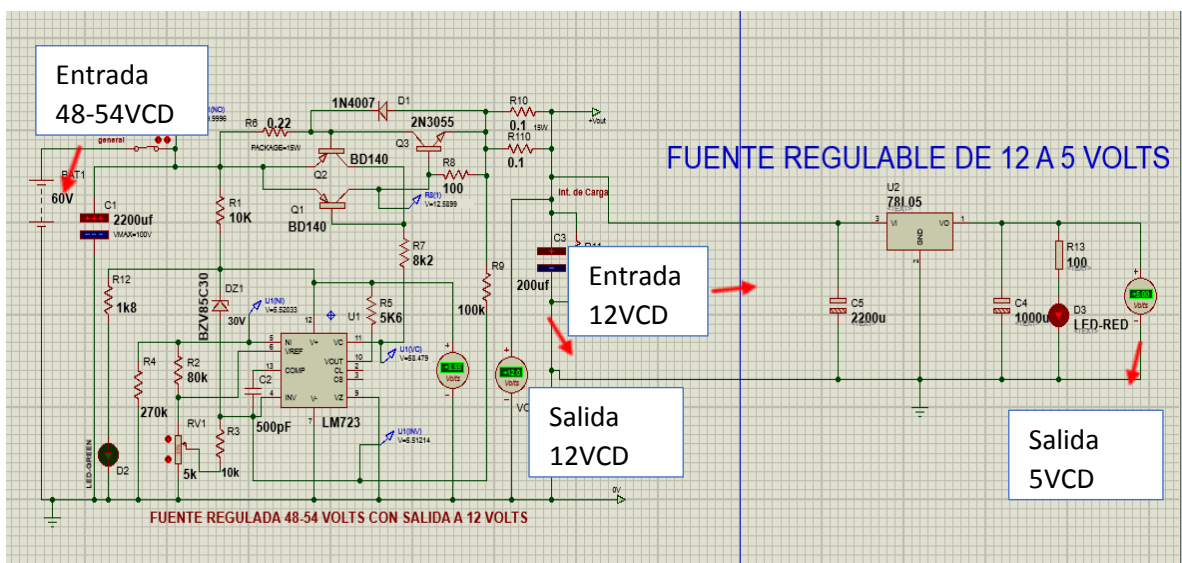


Figura 3.14 Fuente Regulable de 48-54 a 12

Para la etapa de la Fuente de Regulación de Voltaje se utilizaron los siguientes componentes.

- 2 capacitores de 2200uF a 63 Volts
- 1 capacitor de 470pF
- 1 capacitor de 22pF
- 1 circuito integrado LM723
- 2 resistencias de 10K
- 1 resistencia de 8.2K
- 1 resistencia de 5.6K
- 1 resistencia de 33K
- 1 resistencia de 47K
- 2 resistencias de 1.5K
- 1 resistencia de 150 Ohms
- 1 resistencias de 100K
- 1 resistencia de 270K
- 3 resistencias de 1.5 Ohms a 5W
- 1 transistor 2N3055
- 2 transistores BD140

3.5 Diseño de la Placa etapa de Divisor de Tensión (Diferencial de Potencial) en PCB y Vista 3D

En esta etapa podemos ver cómo fue diseñada la placa del Divisor de Tensión en el PCB Wizard y como se ve en 3D.

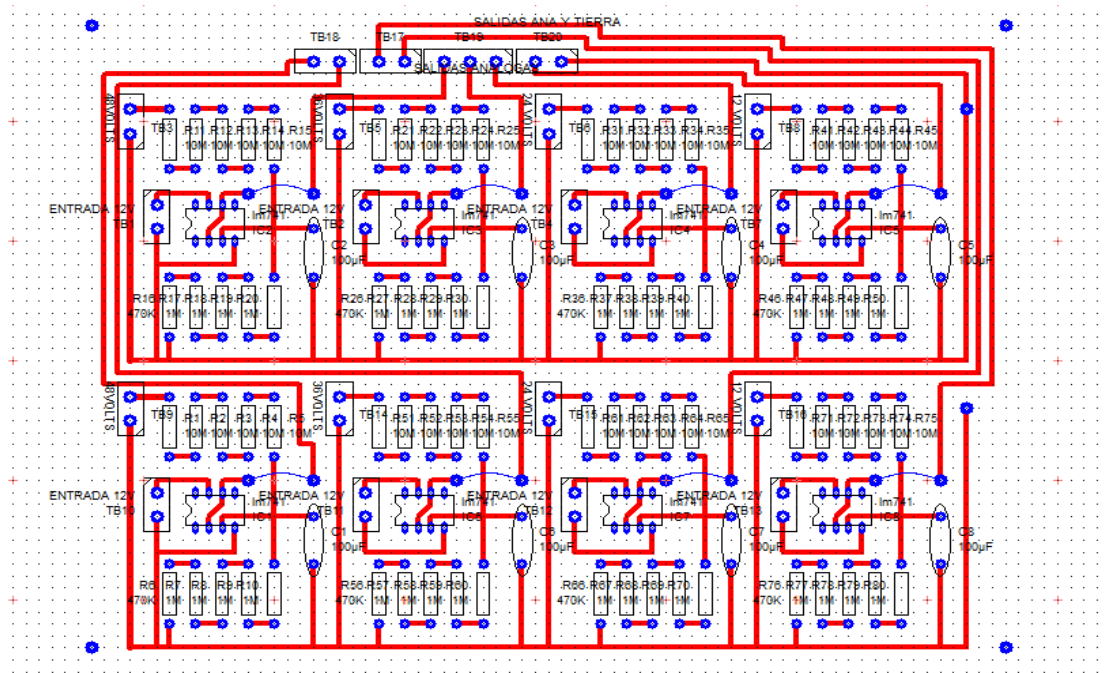


Figura 3.15 Diseño en PCB WIZARD Divisor de Tensión

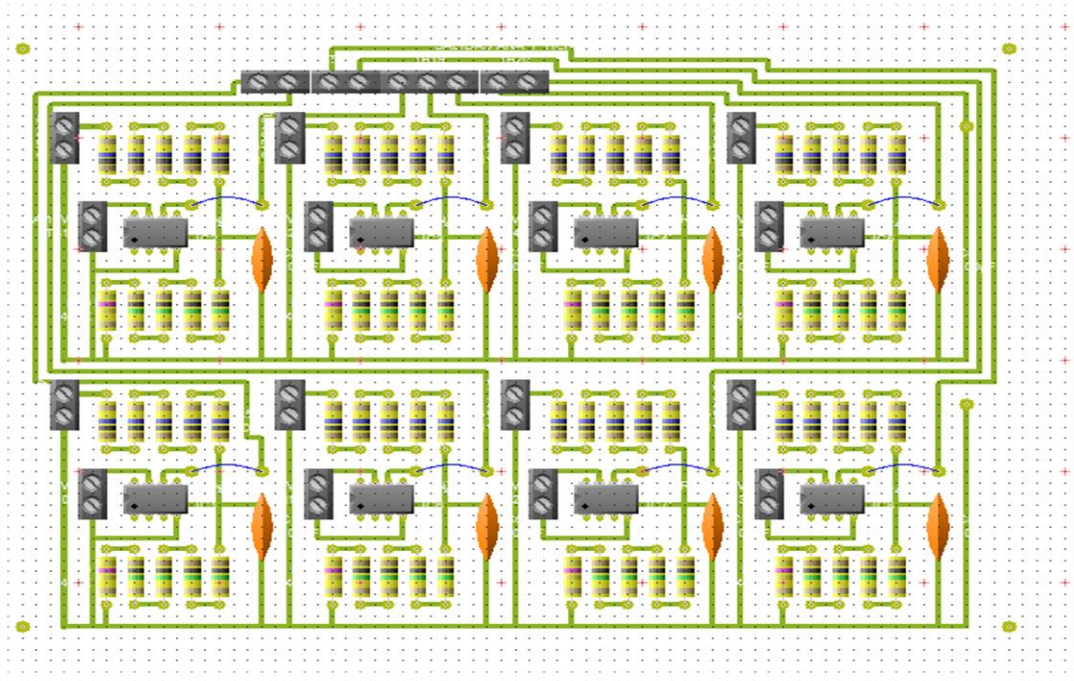


Figura 3.16 vista en 3D Divisor de

3.6 Placa de Adquisición de Datos

La placa de adquisición de datos es la encargada de procesar todos los valores medidos y obtenidos de las variables que se están monitoreando en el Sistema de Fuerza.

Está conformada por los siguientes componentes:

- Arduino ATmega2560
- Arduino Shield Ethernet
- Módulo de Reloj
- Sensor de Humedad y Temperatura DHT11
- Sensor de Temperatura, Termopar Tipo K
- Sensor de Corriente WCS1800-35 Ampers
- Pantalla LCD 20X4 Blue con I2C
- Botón Reset

En la figura 3.17 se puede apreciar el diseño de la etapa de adquisición de datos desarrollado en software Proteus versión 8.5

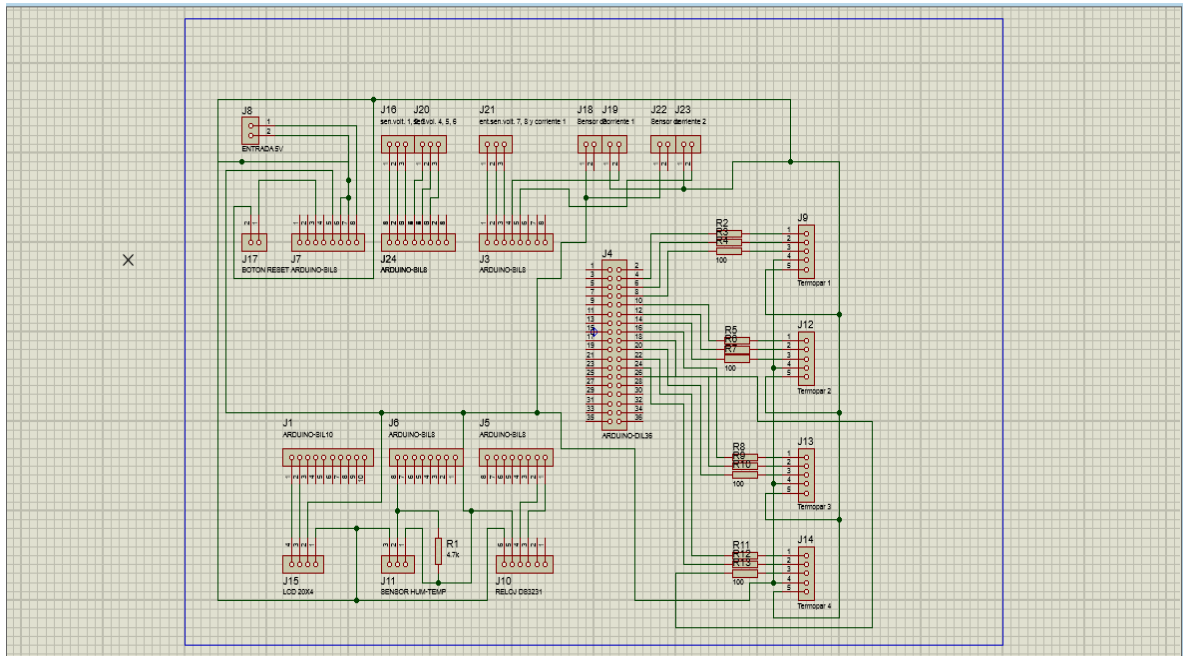


Figura 3.17 Integración de los Componentes

3.7 Diseño de la Placa de etapa Adquisición de Datos en PCB y en 3D

En esta etapa de adquisición de datos podemos apreciar el diseño que se hizo en PCB Wizard y como se ve en 3D.

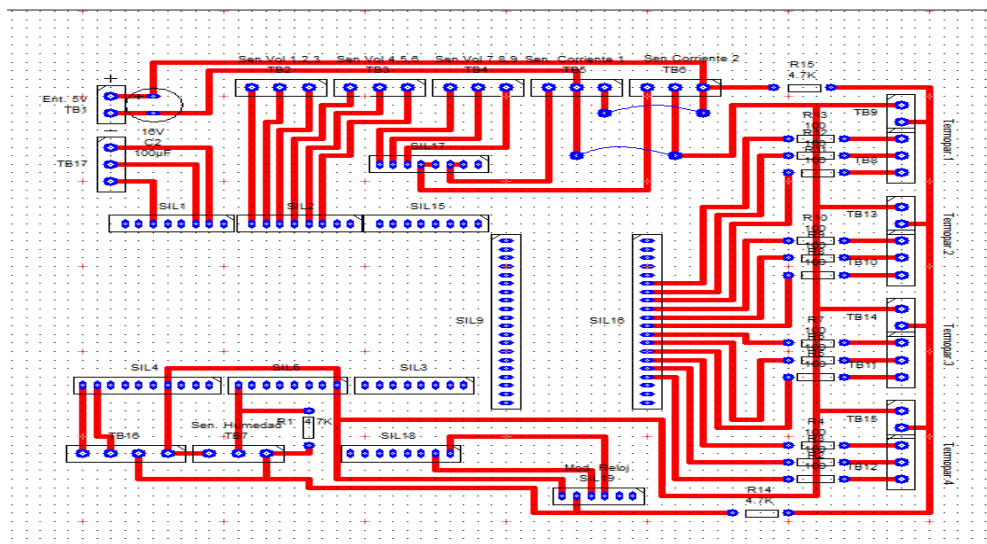


Figura 3.18 Diseño en PCB WIZARD Adquisición de Datos

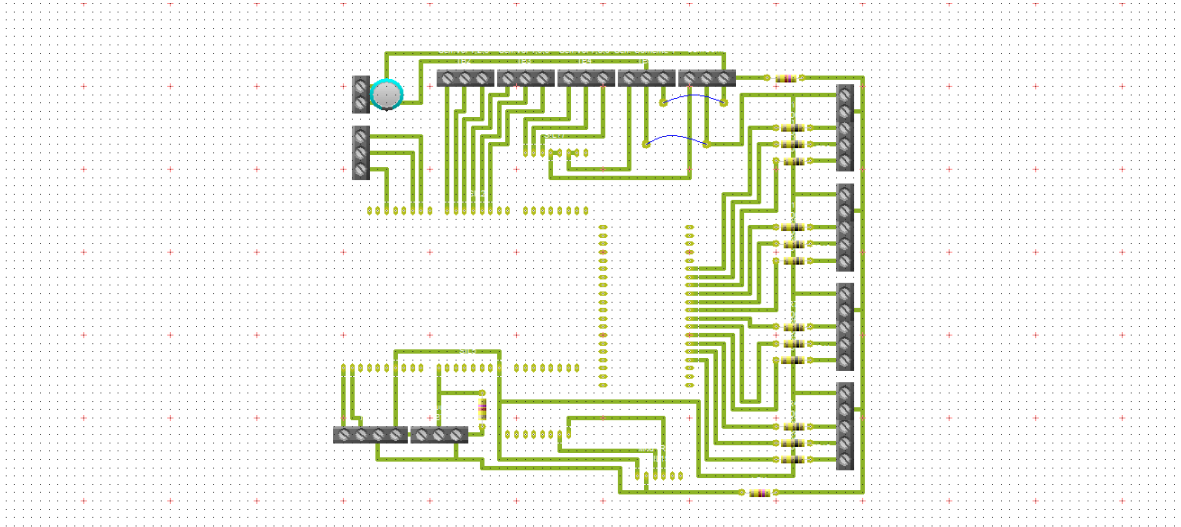


Figura 3.19 vista en 3D Adquisición de Datos

3.8 Materiales para Construcción de las Placas Fenólicas

Para la creación de las placas de los diseños se usó los siguientes materiales:

- Plancha
- Hoja Transfer para diseño de circuito
- Placa fenólica
- Cloruro férrico
- Recipiente de plástico
- Cautín
- Estaño
- Grasa
- Taladro
- Broca 0.9mm
- Lija
- Thinner
- Mesa de madera

3.9 Construcción de la Placa de Adquisición de Datos, Divisor de Tensión (Diferencia de Potencial) y Fuente Regulable de 48-54 volts

Construcción de la placa adquisición de datos

Es importante que la impresión de los diseños PCB Wizard en las hojas transfer sea con impresora a laser, lo cual garantiza que, al aplicar calor sobre ellas, se transfiera más rápido la tinta a la placa fenólica y quede al 100%.

En la figura 3.20 se aprecia el proceso de planchado de la placa fenólicas sobre la hoja transfer.

Como se observa para hacer la placa que se diseñó en PCB hay que colocar bien la impresión de la hoja transfer en la placa fenólica y esta a su vez se plancha aproximadamente por dos minutos y se pone a remojar en un recipiente con agua para levantar el papel y dejar solo la tinta de las pistas del diseño.



Figura 3.20 Proceso de planchado adquisición de

Después del proceso anterior, se procede a colocar la placa fenólicas ya planchadas a un recipiente con cloruro férrico, con la finalidad de quitar las partes de cobre que no se necesitan y solo dejar las pistas de cobre que van a servir para alimentar todos los componentes y pasen los datos que se enviaran, después se lija las partes que quedaron con tinta ósea las pistas, se lava con agua toda la placa, después de eso se limpia con thinner para que quede limpia la placa, después de eso, se checa la continuidad de las pistas para ver que todas están bien y correctas a donde van, al terminar eso, se comienza con las siguientes perforación donde irán los componentes soldados como se observa en las siguientes figuras.



Figura 3.21 quitado de cobre con cloruro férrico

Este es el proceso donde se deja la placa con el cloruro férrico para quitar el cobre que no necesitamos y quede el cobre de las pistas que va a servir.

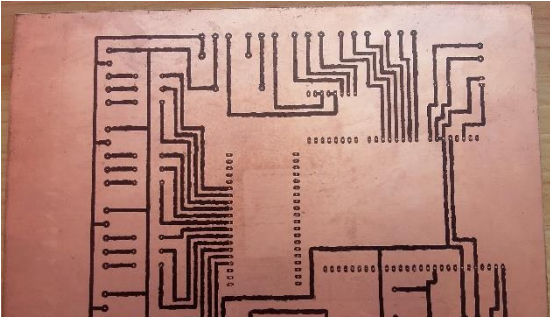


Figura 3.22 Vista de las pistas después del cloruro férrico

En la figura 3.22, se muestran las pistas deseadas con el cobre, donde posteriormente se lijará para un mejor acabado.

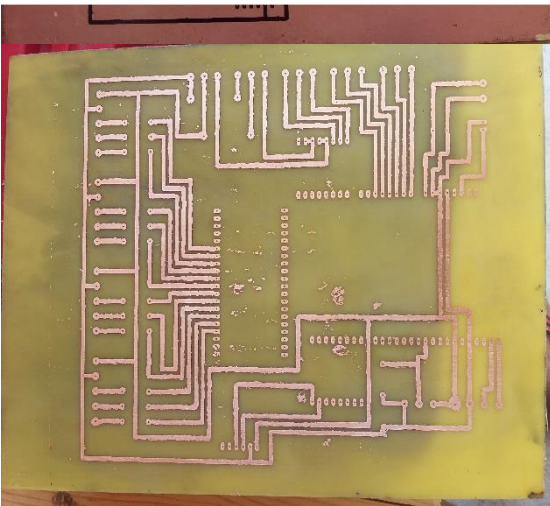


Figura 3.23 Vista de las pistas después de lijarlo

En este tercer paso, se observa después de lijar la tinta que quedo en la figura anterior 3.23, el cobre que nos servirá para darle funcionamiento a nuestros sensores y demás componentes.

Construcción del Proceso de adaptación a los pines sobrantes del Arduino ATmega2560 y Ethernet Shield

En este proceso se hizo la parte de la adaptación de los pines digitales y analógicos sobrantes del Arduino ATmega2560 donde irán los sensores de temperatura, voltaje y el módulo de reloj, esta placa se hizo a doble cara como se muestra en la siguiente figura 3.24

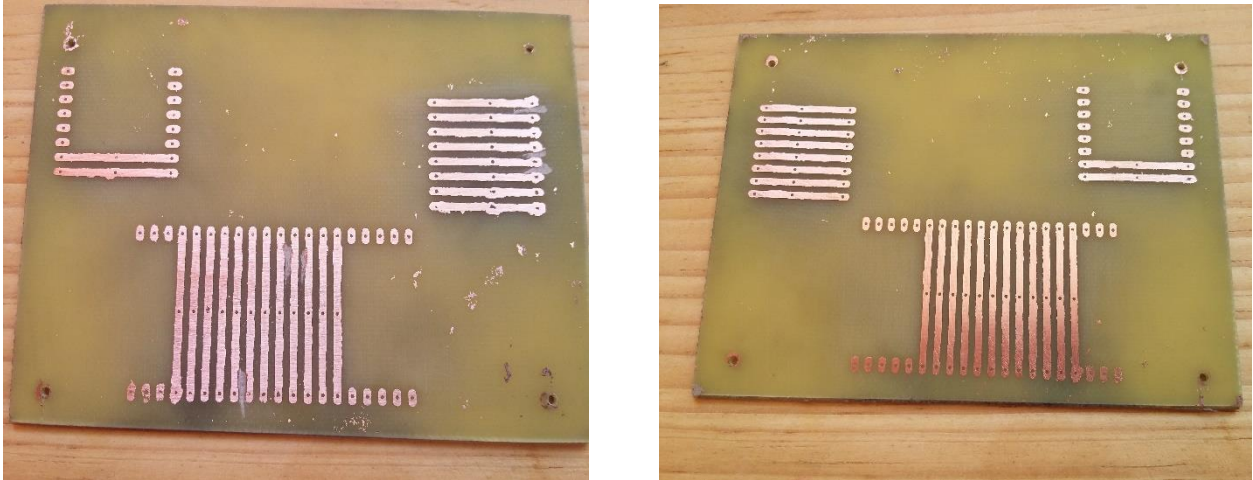


Figura 3.24 Placa a doble cara, adaptación al Arduino con el ethernet

Construcción de la placa Divisor de Tensión (Diferencia de Potencial)

A continuación, se realiza el mismo proceso anterior, donde solo se mostrará las etapas del cloruro férrico, el lijado, lavado con agua y después thinner sobre la placa y como queda al finalizar el circuito con sus pistas, como se observa en las siguientes figuras.

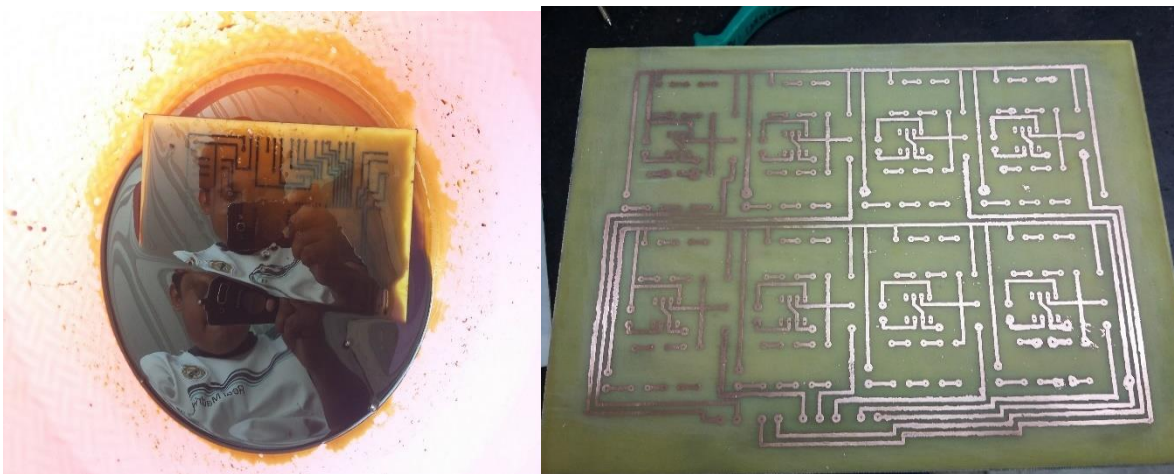


Figura 3.25 Procesos quitado de cobre y lijado Divisor de Tensión (sensor de voltaje)

Construcción de la Placa Fuente Regulable 48-54 Volt

En este paso se hace lo mismo que los dos pasos anteriores, proceso del cloruro férrico, lavado con agua y después thinner, y el finalizado de cómo queda la placa.

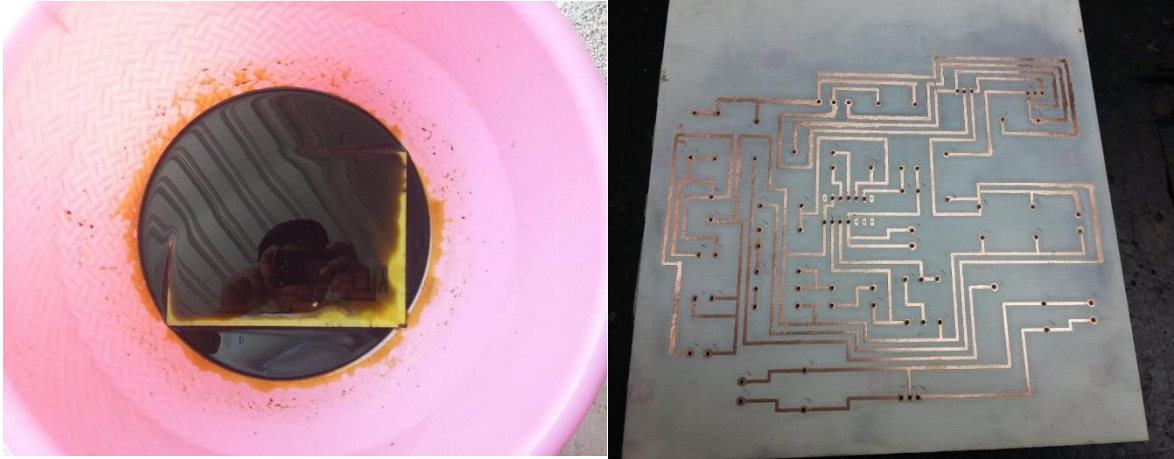


Figura 3.26 Procesos quitado de cobre y lijado fuente regulable

Después de haber realizado el procedimiento de todas las palcas, se continua con el proceso de taladrar y abrir los agujeros a las placas, terminando de hacer eso, se empieza a soldar cada componente con los agujeros correspondientes como se puede apreciar en la siguiente figura 3.27.

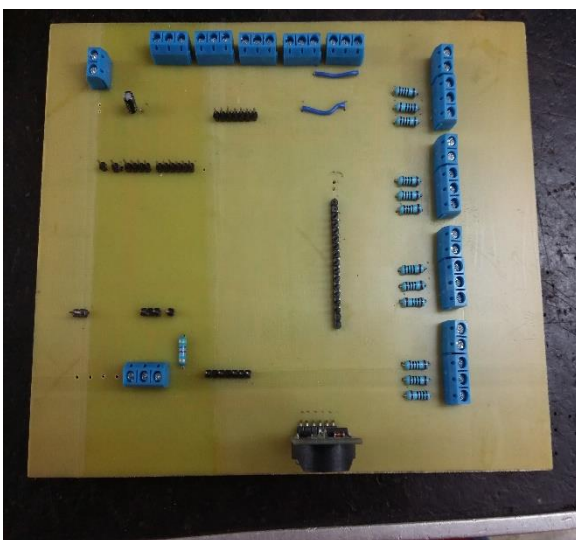
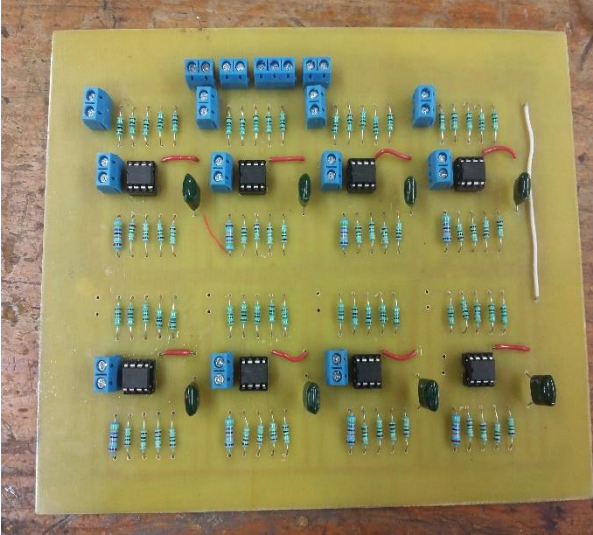


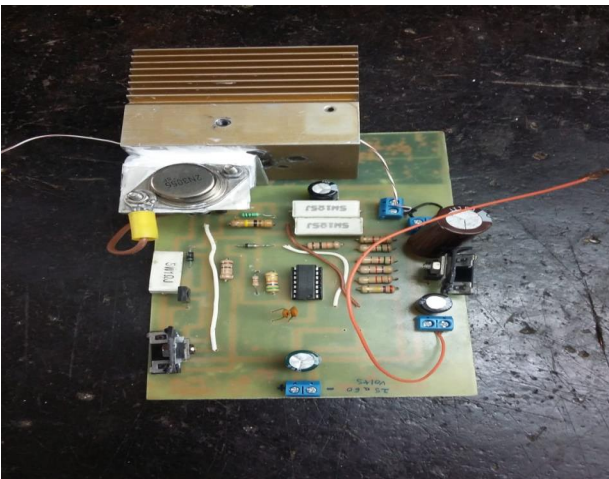
Figura 3.27 Colocación de las piezas soldado

Etapa de adquisición de datos con todos los componentes soldados.



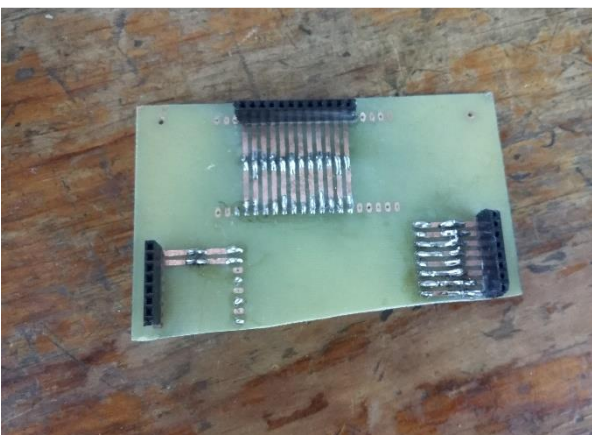
Etapa del divisor de voltaje (sensor de voltaje) con todos los componentes soldados.

Figura 3.28 Colocación de los componentes y piezas ya soldados



Etapa de la fuente de tensión regulable de 48-54 VOLT con todas las piezas y componentes soldados

Figura 3.29 Colocación de los componentes y piezas soldadas



Etapa de adaptación de los pines digitales y analógicos junto con i2c del módulo del reloj del Arduino ATmega2560.

Figura 3.30 Colocación de piezas soldadas

3.10 Vistas de las Placas de los Diseños

A continuación, se presenta en la figura 3.31 la vista de las cuatro placas que se diseñaron para las mediciones de las variables de voltaje, corriente, temperatura y humedad, como se observa la primera placa del lado izquierdo es el divisor de tensión (sensor de voltaje), es la que recibirá la medición de voltaje de cada batería en el sistema de fuerza del banco de baterías, la segunda placa es la de adquisición de datos donde contiene los sensores para la detección de las variables antes mencionadas junto con la pantalla, el módulo de reloj y el sensor de humedad y la tercera placa es la fuente regulable donde regulará el voltaje de 48-54 a 12 Volt y de 12 a 5 Volt.

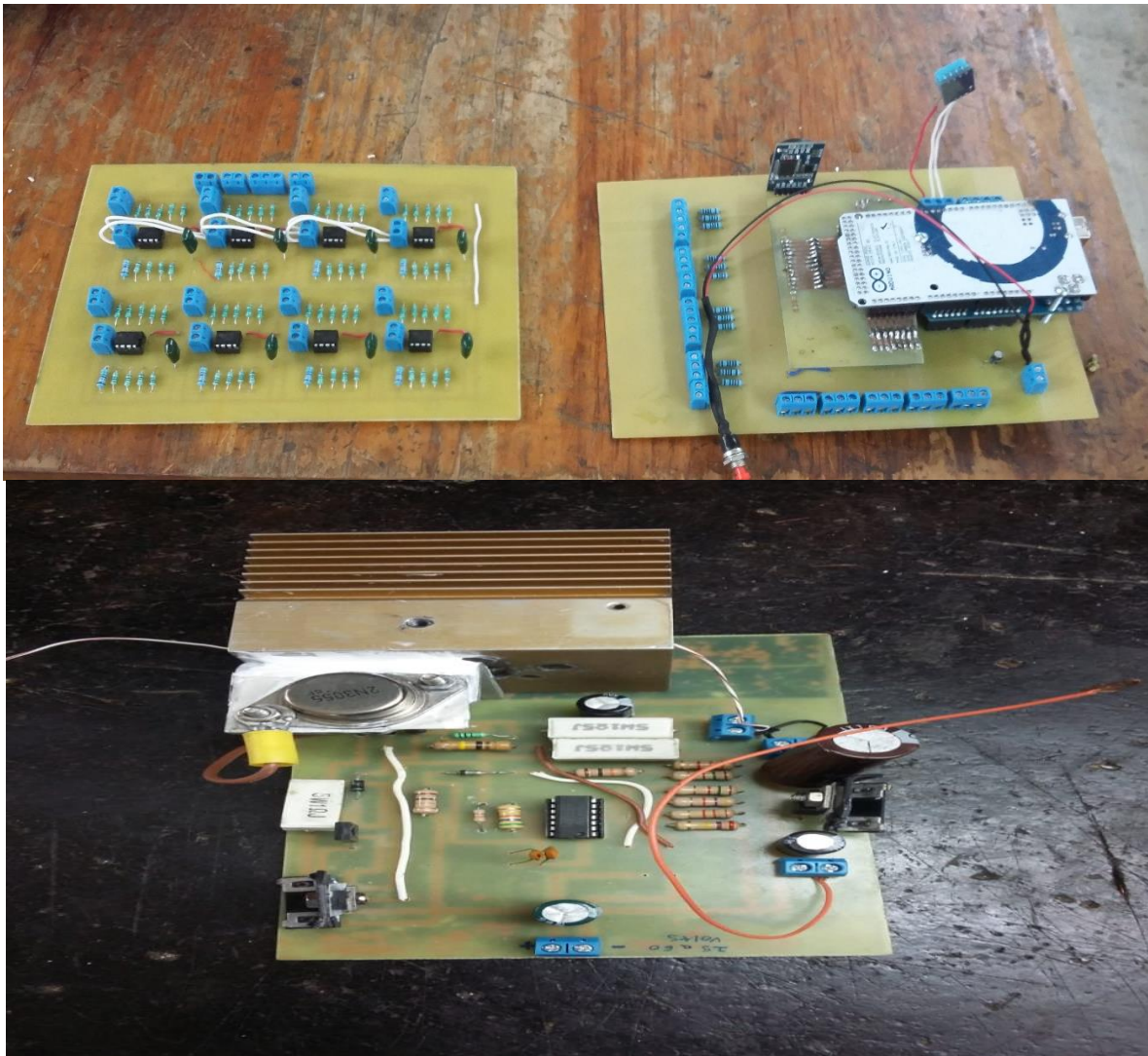


Figura 3.31 Vista de las placas para Arduino ATmega2560

3.11 Prueba de Funcionamiento del Proyecto

Para la verificación, se realizaron las pruebas correspondientes hacia el banco de baterías en la Zona de Transmisión Tuxtla, poniendo todos los sensores a las baterías para ver el censado adecuado. Como se muestra en las siguientes figuras.

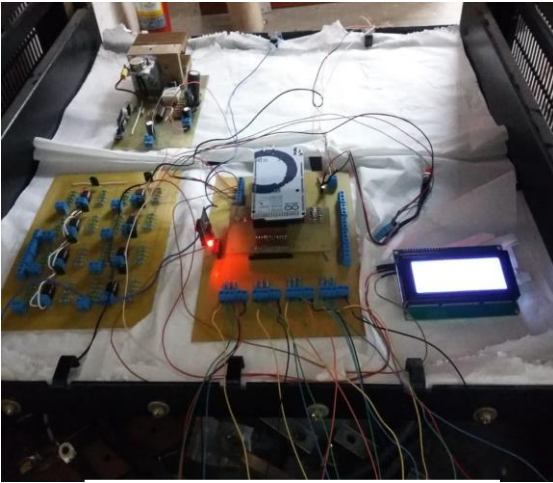


Figura 3.32 Prueba de funcionamiento

En la figura 3.32 se muestra las tres placas conectadas, la fuente regulable, la de divisor de voltaje y la de adquisición de datos, donde podremos observar las siguientes mediciones en las siguientes figuras 3.33, 3.34, 3.35 y 3.36



Figura 3.33 Medición de Humedad y Temperatura

En esta etapa se muestra la medición de la Humedad y Temperatura que hay en el ambiente.

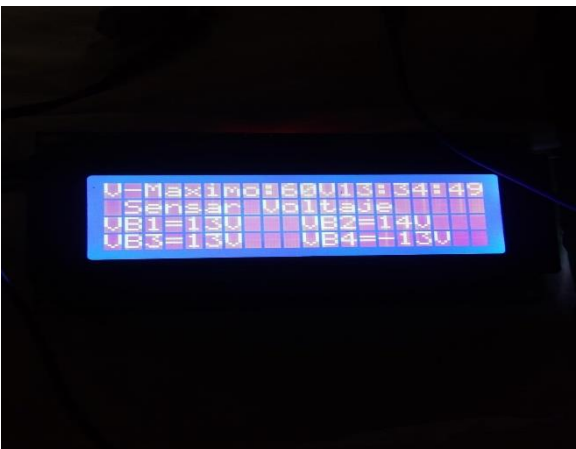


Figura 3.34 Medición de Voltaje

En esta etapa se muestra la medición de los voltajes de cada Batería que hay en el Sistema de Fuerza.



Figura 3.35 Medición de Corriente

En esta etapa se muestra la medición de corriente del sensor general de las baterías sin suministrarle carga al Sistema de Fuerza.



Figura 3.36 Medición de Temperatura

En esta etapa se muestra la medición de Temperatura de cada Batería.

3.12 Implementación del Proyecto en Sistema de Fuerza (Banco de Baterías)

Para la implementación del proyecto primero fue necesario hacer pruebas hacia el Banco de Baterías para ver su comportamiento que tendría cada sensor y demás componentes.

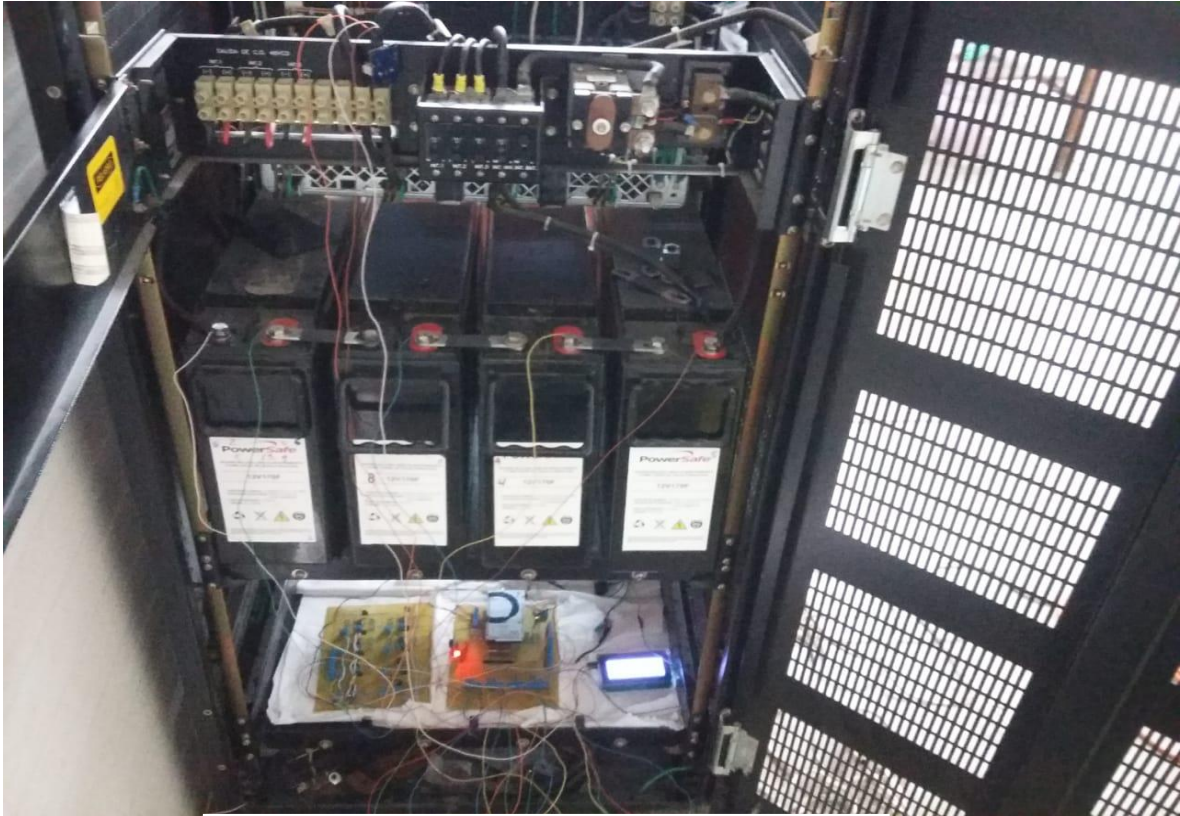


Figura 3.37 Implementación de las placas en el Banco de Baterías

En esta figura se muestra todos los sensores conectados, en la parte de arriba se muestra el sensor de corriente que es un sensor no invasivo, en la baterías están conectados los cables hacia la placa del divisor de tensión que son los sensores de voltaje para tener la medición de cada batería, los sensores de temperatura de cada batería se encuentran en la parte de la espalda y el sensor de humedad y temperatura ambiente esta en la parte de la placa de adquisición por que solo ese mide la humedad y temperatura que hay en el ambiente.

3.13 Diagramas de las Conexiones de los Circuitos de la Placa de Adquisición de Datos y Diferencia de Potencial.

Adquisición de Datos

En este diagrama se muestra las conexiones de todos los sensores que se utilizaron en el Sistema de Monitoreo del Banco de Baterías Figura 3.38 Diagrama Adquisición de Datos.

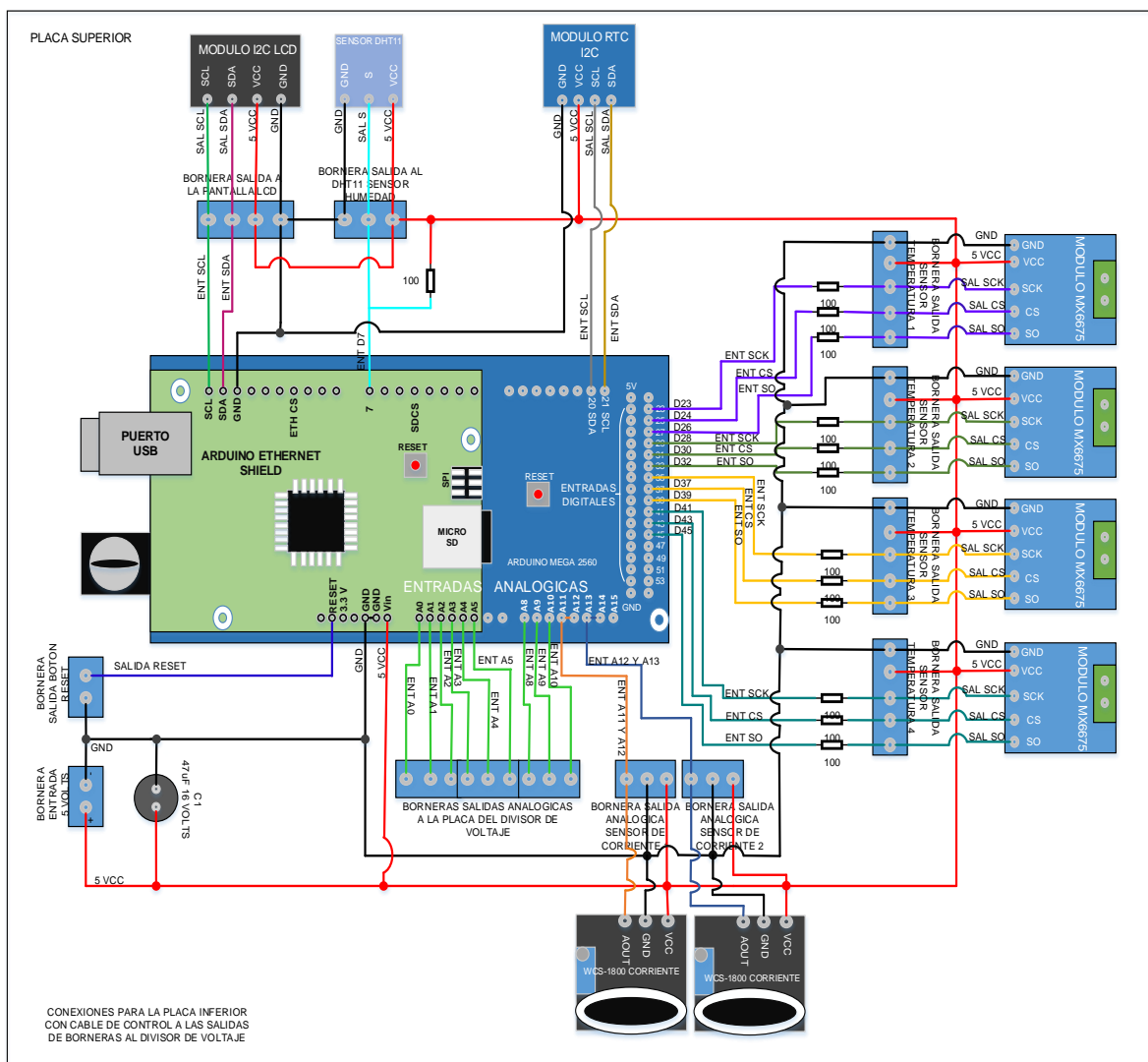


Figura 3.38 Diagrama Adquisición de Datos.

Diferencia de Potencial

En este diagrama podemos observar las conexiones de todos los componentes con los cuales se utilizaron para el diseño de cada sensor de voltaje que estará en cada batería de cada celda hacia el Banco de Baterías como se muestra en la Figura 3.39 Diagrama de Sensor de Voltaje.

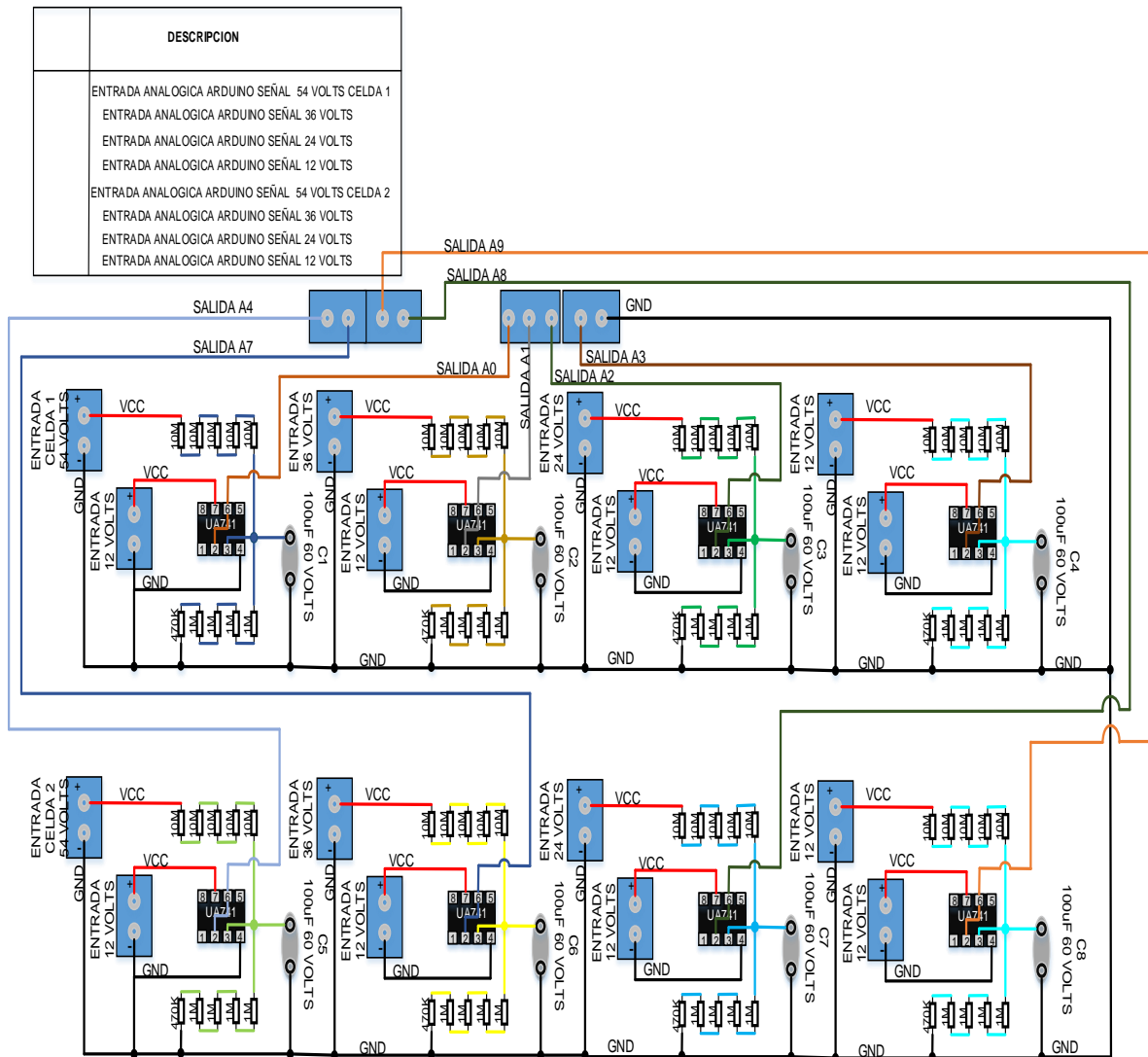


Figura 3.39 Diagrama de Sensor de Voltaje.

Programacion del Nuevo Sistema de Fuerza (Banco de Baterias)

```

#include <LiquidCrystal_I2C.h> //librería del lcd con I2C

#include <Wire.h>

#include <Sodaq_DS3231.h> //librería del módulo reloj

#include <max6675.h> // librería del módulo max6675

#include "DHT.h" // librería del módulo DHT11

#define DHTPIN1 7

#define DHTTYPE DHT11

#include <SD.h> //librería de la MicroSD

#include <SPI.h> // librería de la comunicación con la shield ethernet

#include <Ethernet.h> //librería de la shield ethernet

#include <EthernetClient.h>

#include <EthernetServer.h>

//////////comunicacion al servidor////////////////////////////////////

byte mac[] = {0x90,0xA2,0xDA,0x00,0x00,0x00}; // Declaramos la MAC (Los 3
primeros octetos son de arduino, los otros inventados)

byte ip[] = {192,168,0,2}; // Declaramos la IP de la shield

byte gateway[] = {192,168,0,1}; // Declaramos la IP del router

byte subnet[] = {255,255,255,0}; // Declaramos la máscara de subred

EthernetServer servidor(1080);

EthernetServer server(1080);

```

```
File myFile;

File webFile;

const int chipSelect = 4;

////////función de los días, fechas y horas////////

char DiaSemana[][4] = {"Dom", "Lun", "Mar", "Mier", "Jue", "Vier", "Sab"};

//DateTime dt(19, 06, 12, 9, 17, 0, 2);

DHT dht1(DHTPIN1, DHTTYPE); // Lector del sensor de humedad

LiquidCrystal_I2C lcd(0x27,20,4); // lector del lcd con i2c

////////lectura de los pines analógicos de Voltaje////////

int analogInput0 = A0;

float Vinput = 4.924;

float vout1 = 0.0;

float vin1 = 0.0;

float m4;

float R1 = 49940000;

float R2 = 4400000;

int value1 = 0.0;
```



```
int v1,v2,v3,v4;
```

```
int analogInput1 = A1;
```

```
float vout2 = 0.0;
```

```
float vin2 = 0.0;
```

```
float m3;
```

```
float R3 = 48000000;
```

```
float R4 = 4400000;
```

```
int value2 = 0.0;
```

```
int analogInput2 = A2;
```

```
float vout3 = 0.0;
```

```
float vin3 = 0.0;
```

```
float m2;
```

```
float R5 = 48880000;
```

```
float R6 = 4400000;
```

```
int value3 = 0.0;
```

```
int analogInput3 = A3;
```

```
float vout4 = 0.0;
```

```
float vin4 = 0.0;
```

```
float m1;
```

```
float R7 = 48880000;
```



```

float R8 = 4400000;

int value4 = 0.0;

//////////Lectura de los pines Digitales Temperatura//////////

float temperatura=0;

int CSK1 = 27, CS1 = 25, SO1 = 23, CSK2 = 33, CS2 = 31, SO2 = 29, CSK3 = 39, CS3
= 37, SO3 = 35, CSK4 = 45, CS4 = 43, SO4 = 41;

//int CSK5 = 26, CS5 = 24, SO5 = 22, CSK6 = 32, CS6 = 30, SO6 = 28, CSK7 = 38, CS7
= 36, SO7 = 34, CSK8 = 44, CS8 = 42, SO8 = 40;

MAX6675 sensor1(CSK1, CS1, SO1), sensor2(CSK2, CS2, SO2), sensor3(CSK3, CS3,
SO3), sensor4(CSK4, CS4, SO4);

//////////CONFIGURABLE//////////

int poleoVOL = 15; // Voltaje Poleo Cada 15 Segundos

int poleoCOR = 15; // Corriente Poleo Cada 15 Segundos// Voltaje Poleo Cada 10
Segundos

int poleoTEM = 15; // Corriente Poleo Cada 15 Segundos

int poleoHUM = 15; // Corriente Poleo Cada 15 Segundos

int poleoPantalla = 10; // Pantalla cada 10 segundos

//////////NO CONFIGURABLE//////////

int poleoCLOCK = 1;

int CLOCK = 0;

int pantallaC = 4;

int CVOL = 1;

```

```
int CCOR = 2;

int CTEM = 3;

int CHUM = 4;

////////////////////////////////////muestras de corriente////////////////////////////////////

int muestra1 = 0;

int muestra2 = 0;

int l1;

bool inicio = true;

bool SDactivo = false;

////////////////////////////////////

void setup(){

  Serial.begin(9600);

  while(!Serial){

    //espere a que el puerto serial se conecte. Necesario solo para puerto USB nativo

  }

  if (!SD.begin(4)) {

    Serial.println("inicializando MicroSD fallida!");

    return;

  }

}
```

```
Serial.println("MicroSD inicializada.");

if (!SD.exists("Banco.htm")) {      // Comprueba en la SD si esta el archivo Banco.html

    Serial.println("ERROR - No encuentro index.htm!"); // Caso que no encuentra el
archivo Banco.html

    return;          // Volvemos a intentar inicializar la SD card

}

Serial.println("EXITO - Banco.html encontrado."); // Caso que si encuentra el archivo
Banco.html

lcd.begin(20, 4);

lcd.clear();

lcd.init();      // inicializar la lcd

lcd.init();

lcd.backlight(); // imprimir mensaje en la LCD.

dht1.begin(); // llamar al sensor DHT11

rtc.begin();

Wire.begin();

//rtc.setDateTime (dt);

Ethernet.begin(mac, ip);

server.begin();
```

```
poleoCLOCK = 10 * poleoCLOCK;

poleoVOL = 10* poleoVOL;

poleoCOR = 10 * poleoCOR;

poleoTEM = 10 * poleoTEM;

poleoHUM = 10 * poleoHUM;

poleoPantalla = 10 * poleoPantalla;

{

pinMode(analogInput0, INPUT);

pinMode(analogInput1, INPUT);

pinMode(analogInput2, INPUT);

pinMode(analogInput3, INPUT);

}

}

void loop(){

if (inicio){

    leerClock();

    leerVoltaje();

    leerCorriente();

    leerTemperatura();

    leerHumedad();
```

```
    inicio=false;
}

if(CLOCK >= poleoCLOCK){
    leerClock();
    CLOCK=0;
}CLOCK++;

if (CVOL >= poleoVOL){

leerVoltaje();
CVOL=1;
}CVOL++;

if (CCOR >= poleoCOR){
leerCorriente();
CCOR=2;
}CCOR++;

if (CTEM >= poleoTEM){
leerTemperatura();
CTEM=3;
}CTEM++;
```

```
if (CHUM >= poleoHUM){  
  leerHumedad();  
  CHUM=4;  
}CHUM++;  
delay(100);  
}  
  
void leerVoltaje(){  
  
  value1 = analogRead(analogInput0);  
  vout1 = (value1 * (4.315 / 1024.0));  
  vin1 = vout1 * (R1 + R2) / R2 ;  
  m4 = vin1;  
  
  value2 = analogRead(analogInput1);  
  vout2 = (value2 * (4.315 / 1024.0));  
  vin2 = vout2 * (R3 + R4) / R4 ;  
  m3 = vin2;  
  
  value3 = analogRead(analogInput2);  
  vout3 = (value3 * (4.315 / 1024.0));  
  vin3 = vout3 * (R5 + R6) / R6 ;
```

```
m2 = vin3;
```

```
value4 = analogRead(analogInput3);
```

```
vout4 = (value4 * (4.315 / 1024.0));
```

```
vin4 = vout4 * (R7 + R8) / R8 ;
```

```
m1 = vin4 ;
```

```
float v1=m2-m1;
```

```
float v2=m2-m1;
```

```
float v3=m3-m2;
```

```
float v4=m4-m3;
```

```
if (m4<0.0043){
```

```
    m4 = 0.0;
```

```
}
```

```
if (m3<0.0043){
```

```
    m3 = 0.0;
```

```
}
```

```
if (m2<0.0043){
```

```
    m2 = 0.0;
```

```
}
```

```
if (m1<0.0043){
```

```
    m1 = 0.0;
```

```
}
```

```
////////////////////////////////////lecturas de los voltajes////////////////////////////////////7
```

```
if (pantallaC == 1){  
    myFile = SD.open("Banco.txt", FILE_WRITE);  
if (myFile){  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("V-Maximo:");  
    lcd.print(((int)(Vinput / (R2 / (R1 + R2)))));  
    lcd.print("V");  
    lcd.setCursor(1, 1);  
    lcd.print("Sensor Voltaje");  
    lcd.setCursor(0, 2);  
    lcd.print("VB1=");  
    lcd.print(v4);  
    lcd.print("V");  
    lcd.setCursor(10, 2);  
    lcd.print("VB2=");  
    lcd.print(v3);  
    lcd.print("V");  
    lcd.setCursor(0, 3);  
    lcd.print("VB3=");  
    lcd.print(v2);
```



```
lcd.print("V");
```

```
lcd.setCursor(10, 3);
```

```
lcd.print("VB4=");
```

```
lcd.print(v1);
```

```
lcd.print("V");
```

```
DateTime now = rtc.now();
```

```
myFile.print(' ');
```

```
myFile.print(now.date(), DEC);
```

```
myFile.print('/');
```

```
myFile.print(now.month(), DEC);
```

```
myFile.print('/');
```

```
myFile.print(now.year(), DEC);
```

```
myFile.print(' ');
```

```
myFile.print(now.hour());
```

```
myFile.print(':');
```

```
myFile.print(now.minute());
```

```
myFile.print(':');
```

```
myFile.println(now.second());
```

```
myFile.print(' ');
```

```
myFile.print("Voltaje de cada Bateria");
```

```
myFile.print(' ');
```

```
myFile.println(" VB1=");
myFile.print(v4);
myFile.print("V");
myFile.println(" VB2=");
myFile.print(v3);
myFile.print("V");
myFile.println(" VB3=");
myFile.print(v2);
myFile.print("V");
myFile.println(" VB4=");
myFile.print(v1);
myFile.print("V");
myFile.print(' ');

    if (myFile){
myFile.close();

}

}

}

}

//////////lectura de la corriente//////////

void leerCorriente(){
```

```
float sensorC1 = analogRead(A4);

float a1 = 1.25*(1.015873*60*sensorC1/1024-30);

float c1= a1/2;

float I1 = c1;

muestra1++;

{

  if (pantallaC == 2){

    myFile = SD.open("Banco.txt", FILE_WRITE);

    if (myFile){

      lcd.clear();

      lcd.setCursor(0, 1);

      lcd.print("Sensor Corriente");

      lcd.setCursor(0, 2);

      lcd.print("IBG=");

      lcd.print(I1,2);

      lcd.print("A");

      myFile.print(' ');

      myFile.print("Corriente de cada Cama Bateria");

      myFile.print(' ');
```

```
myFile.println("IBG=");
```

```
myFile.print(a1,4);
```

```
myFile.print("A");
```

```
if (myFile){
```

```
myFile.close();
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
//////////lectura de las temperaturas//////////
```

```
void leerTemperatura()
```

```
{
```

```
if (pantallaC == 3){
```

```
    myFile = SD.open("Banco.txt", FILE_WRITE);
```

```
if (myFile){
```

```
    lcd.clear();
```

```
lcd.setCursor(0, 1);
```

```
lcd.print("Sensor Temperatura");
```

```
lcd.setCursor(0, 2);
```

```
lcd.print("TB1=");  
lcd.print(sensor1.readCelsius());  
lcd.print("C");  
lcd.setCursor(10, 2);  
lcd.print("TB2=");  
lcd.print(sensor2.readCelsius());  
lcd.print("C");  
lcd.setCursor(0, 3);  
lcd.print("TB3=");  
lcd.print(sensor3.readCelsius());  
lcd.print("C");  
lcd.setCursor(10, 3);  
lcd.print("TB4=");  
lcd.print(sensor4.readCelsius());  
lcd.print("C");  
  
myFile.print(' ');  
  
myFile.print("Temperatura de cada Bateria");  
myFile.print(' ');  
myFile.println("TB1=");  
myFile.print(sensor1.readCelsius());  
myFile.print("C");
```

```
myFile.println("TB2=");
myFile.print(sensor2.readCelsius());
myFile.print("C");
myFile.println("TB3=");
myFile.print(sensor3.readCelsius());
myFile.print("C");
myFile.println("TB4=");
myFile.print(sensor4.readCelsius());
myFile.print("C");
myFile.print(' ');

if (myFile){
myFile.close();

}

}

}

}

//////////lectura de la humedad y temperatura ambiente//////////
void leerHumedad()
{

float H1 = dht1.readHumidity();
```

```
float t1 = dht1.readTemperature();

if (pantallaC == 4){
    myFile = SD.open("Banco.txt", FILE_WRITE);
    if (myFile){
        lcd.clear();
        lcd.setCursor(0, 1);
        lcd.print("SISTEMA DE FUERZA");
        lcd.setCursor(0, 2);
        lcd.print("HUM-TEMP AMBIENTE");
        lcd.setCursor(0, 3);
        lcd.print("Ha=");
        lcd.print(H1);
        lcd.print("%");
        lcd.setCursor(10, 3);
        lcd.print("Ta=");
        lcd.print(t1);
        lcd.print("C");

        myFile.print(' ');
```

```
myFile.print("HUM-TEMP AMBIENTE");

myFile.print(' ');

myFile.print("Ha=");
myFile.print(H1);
myFile.print("%");
myFile.println("Ta=");
myFile.print(t1);
myFile.print("C");

myFile.print(' ');

myFile.close();}
}
if (pantallaC > 4){
    pantallaC = 0;
}pantallaC++;
return;
}

////////////////////////////////lectura del reloj////////////////////////////////

void leerClock(){
```



```
DateTime now = rtc.now();

lcd.setCursor(12, 0);

lcd.print(now.hour());

lcd.print(':');

lcd.print(now.minute());

lcd.print(':');

lcd.print(now.second());

int hora = now.hour();

int minuto = now.minute();

int segundo = now.second();

}

//////////lectura de la shiel etherenet//////////

void leerEthernet(){

EthernetClient cliente = server.available();

if (cliente){

    boolean currentLineIsBlank = true;

    while (cliente.connected()){

        char c = cliente.read();

        if (c == '\n' && currentLineIsBlank){

            cliente.println("HTTP/1.1 200 OK");

            cliente.println("Content-Type: text/html");

            cliente.println("connection: close");

            cliente.println("Refresh: 10");
```

```
cliente.println("");

cliente.println("<!DOCTYPE HTML>");

cliente.println("<html>");

cliente.println("<HEAD>");

cliente.println("<TITLE>Ethernet Monitor</TITLE>");

cliente.println("<HEAD>");

cliente.println("<BODY>");

cliente.println("<hr />");

cliente.println("<H1>Banco de Baterias</H1>");

cliente.println("<br />");

cliente.println("<H2>Sistema de Monitoreo TEM-VOL-COR y HUM-TEM
Ambiente</H2>");

cliente.println("<br />");

}

webFile = SD.open("Banco.htm"); // Abre el archivo Banco.html

if (webFile) {

    while(webFile.available()) {

        cliente.write(webFile.read()); // Manda una página web al cliente

    }

    webFile.close();

}

break;}}
```

3.14 Comunicación a la Plataforma ZABBIX

La comunicación a ZABBIX, primeramente, se tuvo que instalar el programa virtual box, y posteriormente iniciar sesión con una máquina Virtual llamada CentOS ya que esta permite el ingreso a la plataforma.

Para el ingreso a la plataforma primeramente se obtenía la dirección IP de la red de CFE y posteriormente se ingresaba la dirección obtenida en nuestro navegador, lo cual nos llevaba a la página que había sido creada para las pruebas, agregábamos el link de ZABBIX y esto nos llevaba a la página de inicio de la plataforma, como se observa en la figura 3.40



Figura 3.40 Inicio de sesión a ZABBIX

Después de iniciar sesión, vamos a verificar nuestro código en el Arduino ATmega2560 junto con nuestro Ethernet para ingresarle la IP ya que los dos están conectados y listos

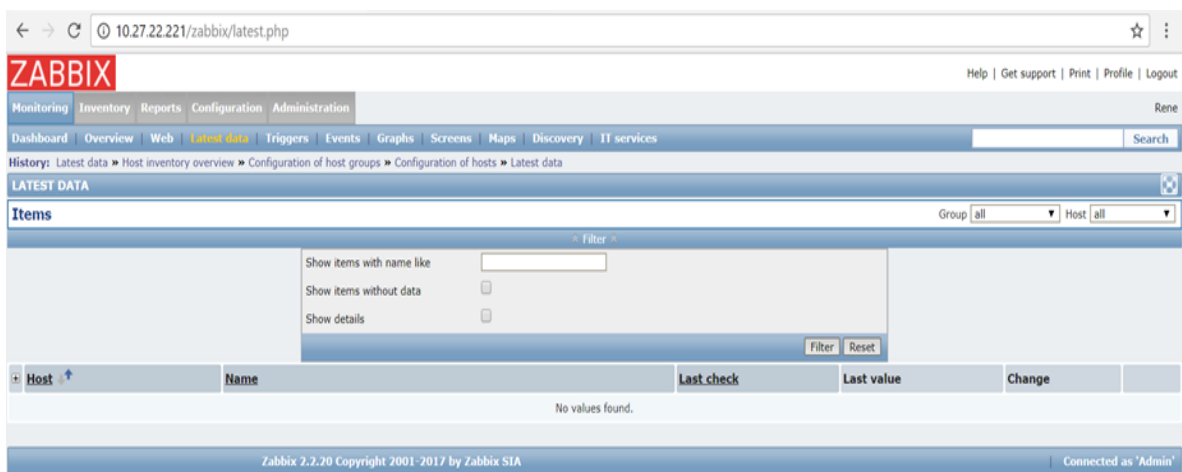


Figura 3.41 Comunicación a ZABBIX

para enviar las variables a medir e ingresarle los datos medidos a nuestra plataforma ZABBIX como lo observaremos en la siguiente figura 3.41

En esta parte podemos observar como reconoce al Arduino la plataforma ZABBIX y listo para enviar las variables correspondientes a monitorear hacia el Banco de Baterías. Figura 3.42 Reconocimiento del Arduino ATmega2560 a Plataforma ZABBIX

| Name | Applications | Items | Triggers | Graphs | Discovery | Web | Interface | Templates | Status | Availability |
|---------------|-------------------|------------|---------------|-------------|---------------|---------|--------------------|---|---------------|--|
| arduino2 | Applications (0) | Items (1) | Triggers (0) | Graphs (0) | Discovery (0) | Web (0) | 10.27.18.23: 10050 | - | Monitored | <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |
| Zabbix server | Applications (11) | Items (63) | Triggers (42) | Graphs (10) | Discovery (2) | Web (0) | 127.0.0.1: 10050 | Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent) | Not monitored | <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> |

Figura 3.42 Reconocimiento del Arduino ATmega2560 y Ethernet a ZABBIX

3.15 Datos Guardados de la Micro SD del Arduino Ethernet Shiel y enviados a Plataforma ZABBIX

Esta etapa de los datos enviados a plataforma ZABBIX, primero se guardan en la micro SD en el Arduino Shield las variables correspondientes a medir, y después se envían a la plataforma ZABBIX con un Unix time, donde se guardan con, el día, mes, año, hora, minuto y segundos de cada medición que son la temperatura y humedad del ambiente, voltaje de cada batería, corriente general y total y las temperaturas de cada batería; con la finalidad de guardar y llevar una estadística del comportamiento del Banco de Baterías.

Los datos se guardan en el servidor ZABBIX como se muestra en la figura 3.43

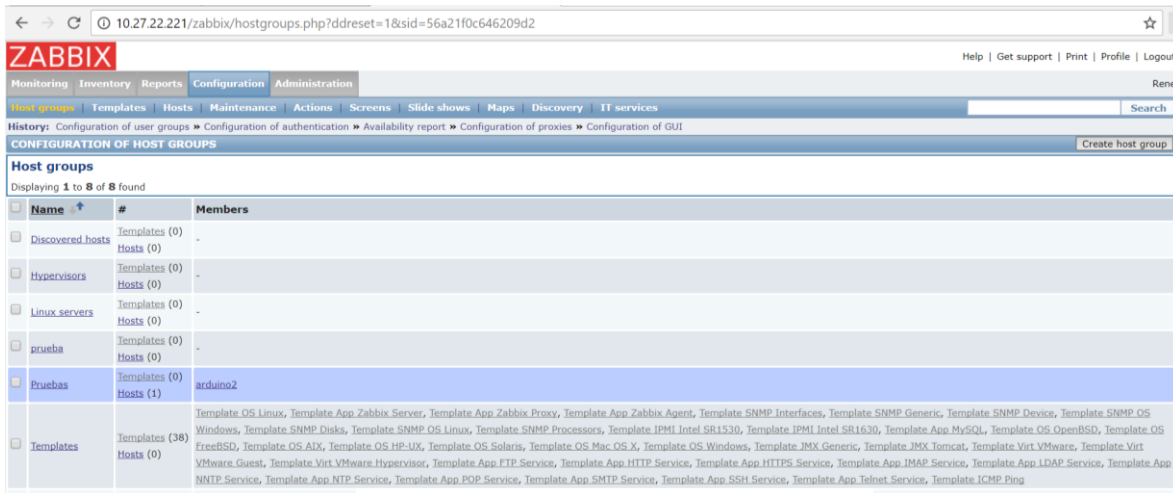


Figura 3.43 Datos enviados al Servidor ZABBIX

3.16 Diagrama de los Datos Obtenidos.

En las siguientes figuras se muestran los valores medidos que son, el voltaje, la corriente y temperatura que hay en el Banco de Baterías enviadas a plataforma ZABBIX.

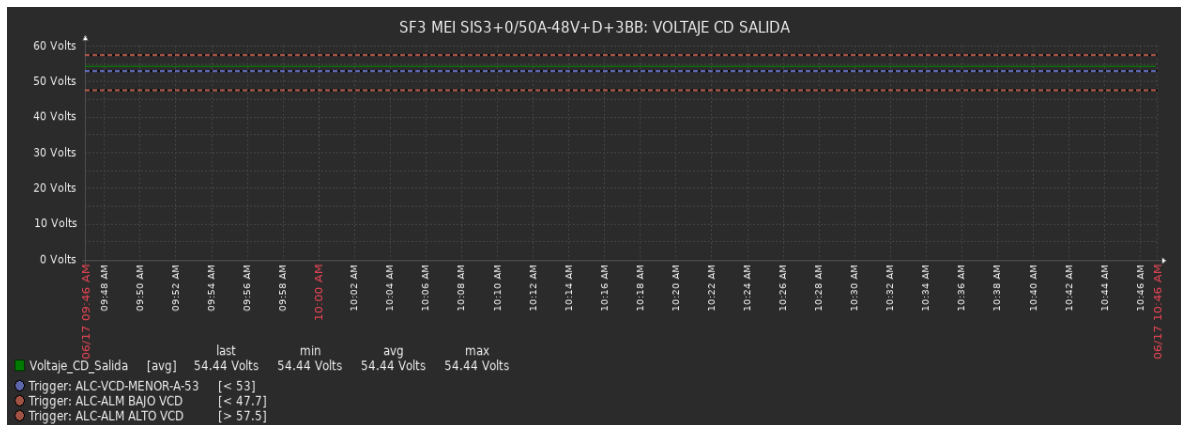


Figura 3.44 Diagrama de los Valores de Voltaje

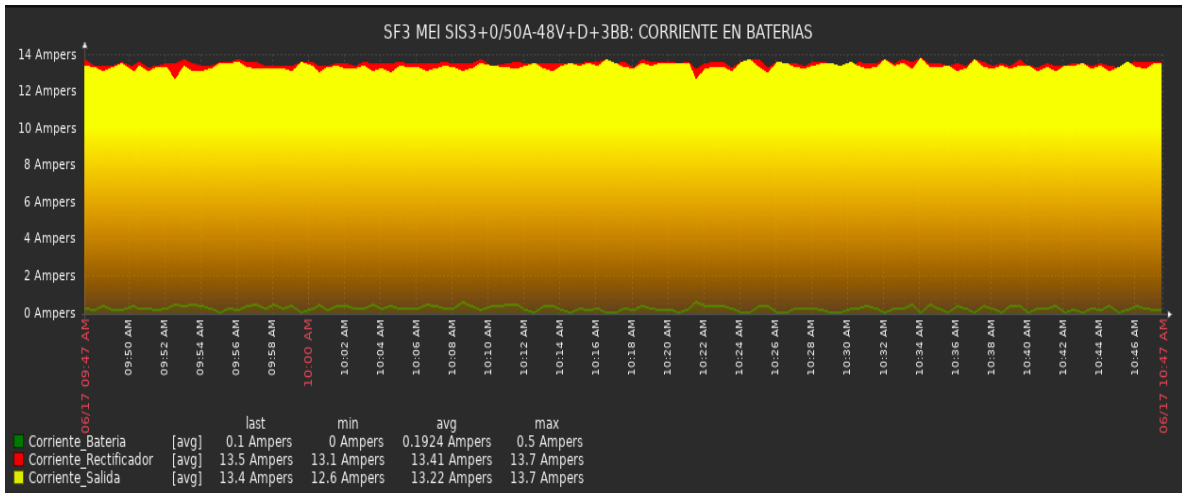


Figura 3.45 Diagrama de los Valores de Corriente

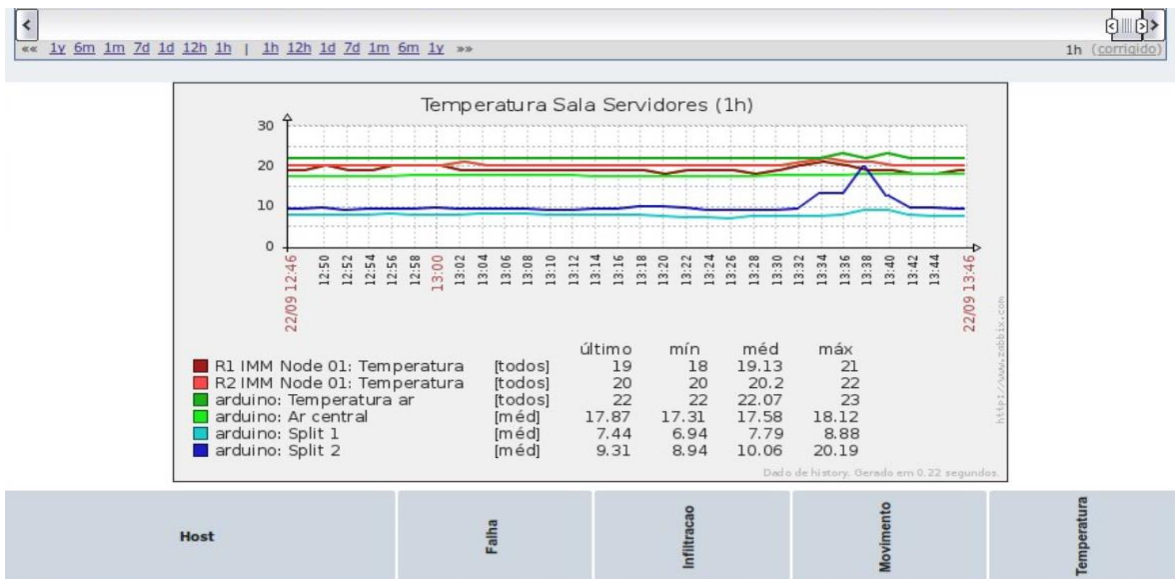


Figura 3.46 Diagrama de los Valores de Temperatura

Observaciones y Sugerencias

Las observaciones que se hicieron en el Sistema de Fuerza del Banco de Baterías son las siguientes:

- Cada batería presenta diferentes voltajes por las cargas y descargas que tienen y hace que su temperatura suba más de lo normal, por tal motivo se requiere de tenerlas monitoreadas para que no sufran daños y así se puedan recuperar las baterías de dichas cargas que sufren y no provoque un daño mayor al Sistema de Fuerza de 48 VCD.
- En un principio se hizo un estudio en el Banco de Baterías, para ver que sensores más sofisticados se iban a usar y así tener una mayor precisión en las mediciones que se iban hacer hacia cada batería. Para los valores de voltaje, se utilizó un Divisor de Tensión utilizando una alta impedancia para tener una mejor medición del voltaje. El de corriente se utilizó uno no invasivo para tener una mejor medición y así no poder dañar al Sistema de Fuerza cuando haya un fallo en ellas y no dañar propiamente al sensor. En la medición de temperatura se utilizó sensores más precisos que es el termopar tipo k que es industrial para poder tener mediciones a altas temperaturas y tener una mejor medición en cada batería. Respecto a la humedad se utilizó solamente un sensor para registrar la humedad del site debido a la existencia de aires acondicionados.

Conclusión

El departamento de comunicaciones de EPS Zona de Transmisión Tuxtla, lleva acabo el monitoreo de diferentes equipos, uno de ellos, son los Sistemas de Fuerza de 48 VCD, los cuales están conformado por un Banco de Baterías para el respaldo de energía o alimentación. Estos equipos de fuerza son los encargados de energizar de manera constante, en caso de falla de la alimentación de los servicios propios en el lugar, por lo que el monitoreo hacia el Banco de Baterías es indispensable para tener un control sobre las baterías: cuanto voltaje tiene cada batería, la corriente que suministran las baterías, su temperatura si es la adecuada y así poder observar que cada batería esté en buenas condiciones y determinar en un momento dado cuáles presentan problemas; con la finalidad de prevenir cualquier anomalía o daño físico en cada celda, ya que estas se van deteriorando y físicamente presenta protuberancias por altas temperaturas que pudiera afectar su voltaje de operación de acuerdo a sus especificaciones por el fabricante.

Bibliografía

- [1] Academia UNAM de Instrumentación Amplificadores tutorial por Jorge Márquez Flores.<http://www.academicos.ccadet.unam.mx/jorge.marquez/cursos/Instrumentacion/AmplificadoresOperacionales.pdf>.
- [2] Sistema de Gestión de carga para Batería de ION-LITIO, Universidad Javeriana, Facultad de Ingeniería Electrónica Bogotá.
- [3] candy-ho.com/Drivers/Manual%20de%20Arduino%20560%20CANDY-Ho.pdf
- [4] https://www.jmi.com.mx/catalogo_sensores/Tablas_Termopares.pdf, <https://cdn-shop.adafruit.com/datasheets/MAX6675.pdf>
- [5] microprocesadores.unam.mx/assets/documentohumedadtemperatura2-1.pdf, <http://www.agspecinfo.com/pdfs/M/MD0025.PDF>
- [6] Sistema modular de Fuerza 3KVA,» de Instructivo de Operaciones Sistema de fuerza, MEI Multieléctrica Industrial, S.A. DE C.V., 2017.
- [7] Curso Básico de Baterías, MEI Multieléctrica Industrial, S.A. DE C.V. , 2017.
- [8] <http://domoticamayca.blogspot.com/2016/11/tutorial-arduino-sd-crear-leer-eliminar.html>
- [9] <https://ardubasic.wordpress.com/2017/05/14/escudo-de-ethernet-w5100-iv-lector-de-tarjetas-sd/>
- [10] <https://www.prometec.net/sdcard/>
- [11] Educa chip-Com-Arduino-Ethernet-Shield