



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA



# Visualizador de Variables para Laboratorio de Máquinas Eléctricas (VISLAB)

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE  
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Alejandro Bartel, Víctor Cartagena, Agustín Pérez

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS  
PARA LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO ELECTRICISTA.

## TUTOR

Pablo Toscano ..... Universidad de la República

## TRIBUNAL

Andrés Cardozo..... Universidad de la República

Mauricio Riera..... Universidad de la República

Pablo Toscano ..... Universidad de la República

Montevideo  
martes 27 octubre, 2015

*Visualizador de Variables para Laboratorio de Máquinas Eléctricas (VISLAB)*, Alejandro Bartel, Víctor Cartagena, Agustín Pérez.

Esta tesis fue preparada en L<sup>A</sup>T<sub>E</sub>X usando la clase iietesis (v1.1).  
Contiene un total de 187 páginas.  
Compilada el martes 27 octubre, 2015.  
<http://iie.fing.edu.uy/>

Sean los orientales tan ilustrados como valientes.

JOSÉ GERVASIO ARTIGAS

Esta página ha sido intencionalmente dejada en blanco.

# Agradecimientos

Muchas fueron las personas que de una forma u otra aportaron a la realización de VISLAB, a todas ellas les agradecemos profundamente. Esperando no olvidarnos de nadie a continuación listamos las más relevantes y les damos un especial agradecimiento.

- A nuestras familias, parejas y amigos por su apoyo constante e incondicional.
- A compañeros de trabajo por asesorarnos en las etapas iniciales, entre los cuales destacamos a Marcelo Choca, Carlos Iglesias, Jorge Coria, Jose Camejo.
- Al docente del IIE, Ing. Carlos Briozzo, por su colaboración y aportes en el diseño y depuración del hardware.
- A Matías Dumas quien nos dono una chapa de cobre para la implementación de los circuitos.
- A Ramiro Barron y Federico Lopez por habernos prestado accesorios de comunicación.
- A Jessica Monteiro y Marcelo Choca, por la realización del mímico del hardware.
- A nuestro tutor, Ing. Pablo Toscano, por la confianza depositada en el equipo, sus ideas y críticas, y el estímulo siempre positivo.

Esta página ha sido intencionalmente dejada en blanco.

# Tabla de contenidos

<b>Agradecimientos</b>	<b>III</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción de los equipos . . . . .	2
1.1.1. Par Acoplado Máquina sincrónica - Máquina de continua . . . . .	2
1.1.2. Fuente de Continua y Arrancador . . . . .	3
1.1.3. Cargas . . . . .	3
1.1.4. Panel de protección, comando y medidas . . . . .	3
1.2. Ensayos realizados en la asignatura Taller de Máquinas Eléctricas . . . . .	3
1.2.1. Ensayo de vacío . . . . .	3
1.2.2. Ensayo de cortocircuito . . . . .	4
1.2.3. Ensayo con carga resistiva . . . . .	4
1.2.4. Puesta en paralelo de dos MS . . . . .	4
1.3. Instrumentos y Medidas . . . . .	5
1.3.1. Tensiones y corrientes trifásicas . . . . .	5
1.3.2. Tensiones y corrientes continuas . . . . .	5
1.3.3. Medidas correspondientes a variables no-eléctricas . . . . .	5
1.4. Objetivos de Proyecto . . . . .	6
1.5. Resumen . . . . .	7
<b>2. Arquitectura del Sistema</b>	<b>9</b>
2.1. Elementos de sistema . . . . .	9
2.1.1. Equipos analizados . . . . .	10
2.2. Análisis de las arquitecturas del Sistema . . . . .	12
2.3. Evaluación de las arquitecturas a implementar del Sistema . . . . .	14
2.3.1. Selección de Sistema . . . . .	16
2.4. Resumen . . . . .	17
<b>3. Hardware</b>	<b>19</b>
3.1. Componentes del Hardware . . . . .	20
3.1.1. Arduino Mega 2560 . . . . .	20
3.1.2. Placa de comunicación Sistema-PC y alimentación de Arduino . . . . .	20
3.1.3. Placa IO (Input/Output) . . . . .	22
3.1.4. Fuente de 5 Vdc . . . . .	25
3.1.5. Sensores y Periféricos . . . . .	27

## Tabla de contenidos

3.2. Arquitectura . . . . .	32
3.2.1. Diagramas del Sistema . . . . .	32
3.2.2. Módulos de acondicionamiento de señales . . . . .	36
3.2.3. Mapeo de las señales de entradas . . . . .	38
3.2.4. Adquisición de las señales . . . . .	38
3.3. Diseño de protecciones del sistema . . . . .	39
3.4. Descripción exterior del Equipo . . . . .	40
3.4.1. Ubicación de los diferentes módulos dentro del Equipo . . . . .	42
3.4.2. Mímico del frente del equipo . . . . .	45
<b>4. Comunicación</b>	<b>47</b>
4.1. Desarrollo de comunicación . . . . .	47
4.2. Interfaces de comunicación y Puertos Seriales . . . . .	48
4.2.1. Puerto Serie . . . . .	48
4.2.2. RS-485 . . . . .	48
4.2.3. RS-232 . . . . .	49
4.2.4. Puerto USB . . . . .	49
4.2.5. Bus SMBus . . . . .	50
4.3. Protocolos de Comunicación . . . . .	50
4.3.1. Modbus . . . . .	50
4.3.2. Protocolo SMBus . . . . .	52
4.4. Pruebas de Comunicación Preliminares . . . . .	52
4.4.1. Pruebas de comunicación CVMk2 con PC . . . . .	52
4.4.2. Pruebas de comunicación Arduino-LabVIEW . . . . .	55
4.5. Comunicación Arduino-LabVIEW Implementada . . . . .	57
4.5.1. Protocolo Creado . . . . .	57
4.5.2. Driver de Comunicación Arduino-LabVIEW . . . . .	58
4.6. Resumen . . . . .	59
<b>5. Software</b>	<b>61</b>
5.1. Plataforma de desarrollo LabVIEW . . . . .	61
5.1.1. Introducción . . . . .	61
5.1.2. Principales características . . . . .	62
5.1.3. Programación en LabVIEW . . . . .	62
5.2. Software desarrollado en LabVIEW . . . . .	66
5.2.1. Desarrollo entorno gráfico . . . . .	66
5.2.2. Configuración parámetros del software . . . . .	79
5.3. Firmware desarrollado para placa Arduino . . . . .	83
<b>6. Pruebas y ajustes</b>	<b>85</b>
6.1. Introducción . . . . .	85
6.2. Calibración de Elementos . . . . .	85
6.2.1. Ajustes de las medidas de tensión DC . . . . .	85
6.2.2. Ajustes de sensores de corriente . . . . .	87
6.2.3. Sensor de temperatura . . . . .	91



<b>7. Conclusión</b>	<b>93</b>
7.1. Cumplimiento de Objetivos . . . . .	94
7.2. Limitaciones . . . . .	95
7.3. Posibles mejoras . . . . .	95
7.3.1. Software . . . . .	95
7.3.2. Mejoras generales del equipo . . . . .	96
7.4. Resumen . . . . .	96
<b>Apéndices</b>	<b>97</b>
<b>A. Desarrollo de programación LabVIEW</b>	<b>97</b>
A.1. Mapa del Software Principal . . . . .	97
A.1.1. Driver Arduino para LabVIEW . . . . .	99
A.1.2. Filtrado de señal . . . . .	100
A.1.3. Desarrollo para el ensayo de Máquinas en Paralelo . . . . .	101
A.1.4. Desarrollo Ensayo de Vacío . . . . .	102
A.1.5. Desarrollo Ensayo de Cortocircuito . . . . .	104
A.1.6. Desarrollo Ensayo de diagrama P-Q de Máquina Sincrónica . . . . .	105
A.1.7. Desarrollo Modelo de Potier . . . . .	106
A.1.8. Desarrollo Modelo Behn-Eschenburg . . . . .	112
A.1.9. Desarrollo de SubVI para la lectura de archivos de textos . . . . .	115
<b>B. Fundamentos teóricos</b>	<b>119</b>
B.1. Modelo de Behen-Eschenburg . . . . .	119
B.2. Modelo de Potier . . . . .	121
<b>C. Práctica de Máquina Sincrónica de la asignatura Taller de Máquinas Eléctricas del IIE</b>	<b>123</b>
<b>D. Manual de utilización del Sistema</b>	<b>135</b>
D.1. Introducción . . . . .	135
D.2. Descripción del equipo . . . . .	135
D.3. Instalación y configuraciones . . . . .	135
D.4. Conexionado . . . . .	136
D.5. Encendido del equipo . . . . .	137
D.6. Descripción del Software . . . . .	138
D.7. Configuraciones . . . . .	143
<b>E. Código del firmware de la placa Arduino</b>	<b>147</b>
<b>F. Gestión de tiempos</b>	<b>157</b>
F.1. Cambios en la planificación . . . . .	157
F.2. Dedicación mensual . . . . .	159
<b>G. Gestión de gastos</b>	<b>161</b>
G.1. Gastos realizados . . . . .	161

Tabla de contenidos

<b>H. Hoja de Datos de Equipo</b>	<b>163</b>
<b>Referencias</b>	<b>165</b>
<b>Glosario</b>	<b>166</b>
<b>Glosario</b>	<b>167</b>
<b>Índice de tablas</b>	<b>168</b>
<b>Índice de figuras</b>	<b>170</b>

# Capítulo 1

## Introducción

Las prácticas con máquinas eléctricas son parte fundamental dentro del Perfil Potencia de la carrera Ingeniería Eléctrica. Éstas se desarrollan en la asignatura "Taller de Máquinas Eléctricas", y tienen lugar en el Laboratorio de Máquinas Eléctricas (LME) del Instituto de Ingeniería Eléctrica (IIE).

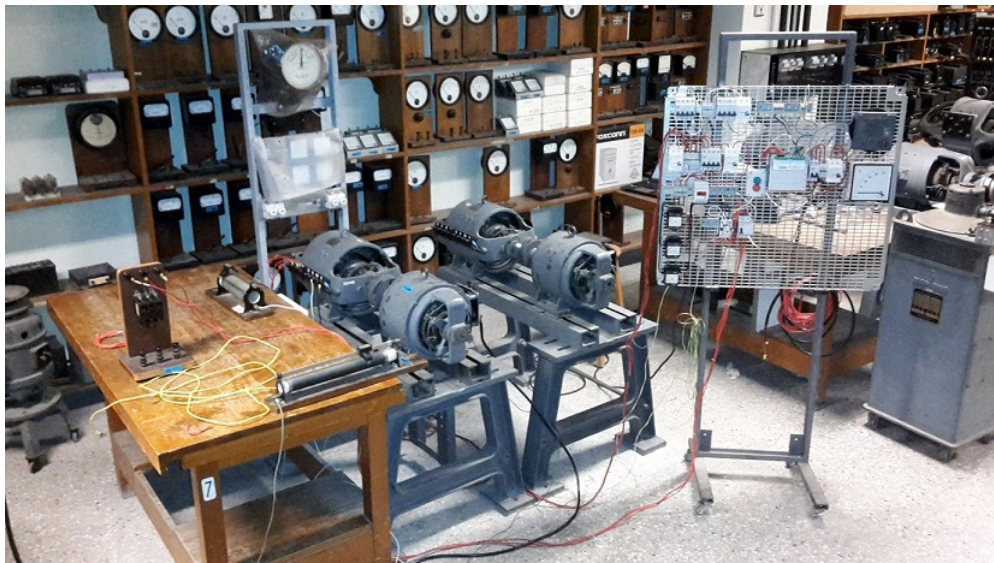


Figura 1.1: Fotografía del Laboratorio de Máquinas Eléctricas del IIE

## Capítulo 1. Introducción

En el transcurso de los ensayos correspondientes a dicha asignatura es necesario que los estudiantes (generalmente en grupos de 4) releven datos de distintos equipos y/o instrumentos de medida (analógicos, digitales, con display gráficos), distribuidos en la proximidad de la máquina/s ensayada/s, por un tiempo que se puede extender por varias horas; y en la mayoría de los casos para distintos estados electromecánicos en las máquinas. Es fácil ver que debido a la gran cantidad de instrumentación distribuida a la que se le debe prestar atención, es difícil para los alumnos entender como las ecuaciones y diagramas estudiados en el curso teórico se relacionan con las variables 'en vivo' que se observan durante la práctica.

A partir de esta problemática fue que surgió la idea de este proyecto. Durante el transcurso de este trabajo, se describen los elementos necesarios para la construcción de un dispositivo e implementación de interfaz gráfica (al conjunto de los dos lo denominaremos Sistema), que permite presentar todas la variables relevantes a los ensayos de manera amigable para el estudiante.

El dispositivo propuesto debe poder adquirir, guardar y desplegar todas las medidas relevantes durante los distintos ensayos, con una tasa de actualización menor o igual a la que los estudiantes puedan acceder con los instrumentos existentes, de manera gráfica, clara y concisa.

En próximas secciones se analizarán los equipos, instrumentos y ensayos realizados (o que se podrían realizar) en el curso "Taller de Máquinas", y con esto se dará una descripción detallada de los requerimientos que deberá cumplir el Sistema.

### 1.1. Descripción de los equipos

A continuación se describen algunos de los equipos de potencia existentes en el Laboratorio, que son utilizados en el transcurso del Taller.

#### 1.1.1. Par Acoplado Máquina sincrónica - Máquina de continua

En el Laboratorio se cuenta con máquinas síncronas trifásicas de diferentes valores nominales que son utilizadas generalmente para funcionar como generador. Como es necesario dar par mecánico para poder generar, las máquinas síncronas cuentan (en su mayoría) con máquinas de continua que funcionan como motor acopladas al mismo eje.

Una particularidad de los alternadores (otro nombre que se le da a los generadores síncronos) utilizados en el LME, es que el fabricante ha dispuesto los bobinados inductores en la carcasa fija de la máquina, de polos salientes, y el inducido giratorio construido sobre un tambor ranurado de hierro laminado. Cuentan con capacidad de ser autoexcitadas, pero esta característica no es utilizada siendo excitadas a través de una fuente de continua externa. La máquina de continua acoplada al alternador podrá ser conectada como shunt o de excitación independiente y funcionará como motor de arrastre.

El conexionado del motor y el alternador se indica en el esquema simplificado de la figura 7.1

## 1.2. Ensayos realizados en la asignatura Taller de Máquinas Eléctricas

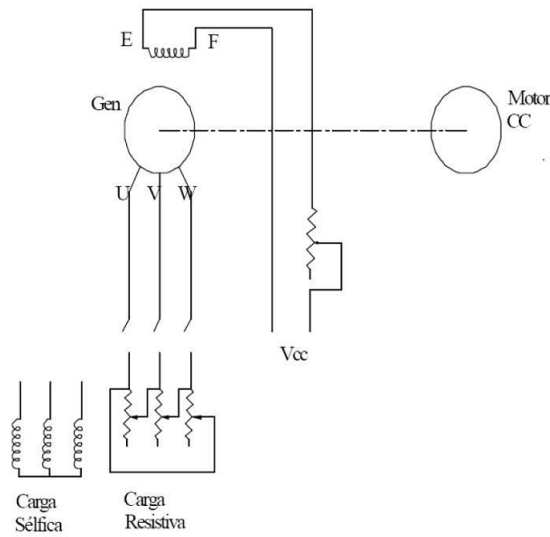


Figura 1.2: Diagrama simplificado del grupo MS-MCC utilizada en el Laboratorio

### 1.1.2. Fuente de Continua y Arrancador

El Laboratorio cuenta con un equipo de fabricación propia, el cual tiene una fuente de tensión continua variable basada en un variac y rectificador de onda completa utilizada para excitar las MS, un arrancador resistivo de tres pasos para los motores de continúa, y varios instrumentos analógicos y digitales de medida.

### 1.1.3. Cargas

La carga a conectar a las MS puede ser compuesta por: resistencias, bancos de inductancias y capacitores disponibles en laboratorio.

### 1.1.4. Panel de protección, comando y medidas

Se cuenta con un panel móvil que contiene borneras, protecciones, contactores y diversos instrumentos de medida para conectar la red de UTE a motores AC o la salida trifásica de la MS a su respectiva carga.

## 1.2. Ensayos realizados en la asignatura Taller de Máquinas Eléctricas

### 1.2.1. Ensayo de vacío

En este ensayo se determina la fuerza electromotriz de la MS en función de la corriente de excitación a velocidad nominal, para esto la máquina es arrastrada con el motor de continua descrito anteriormente. Con la máquina sin carga se la lleva

## Capítulo 1. Introducción

a velocidad nominal y se varía la corriente de excitación de la misma, produciendo un aumento gradual hasta alcanzar la tensión nominal de salida, descendiendo luego la excitación hasta el mínimo posible, normalizando así el ciclo de histéresis del hierro. Luego de esto se procede a realizar un nuevo ciclo en el que se toman las lecturas de excitación y tensión en bornes de la máquina. Se lleva la tensión hasta un 1,1 del valor nominal, cuidando el ajuste de la corriente de excitación, de tal forma que siempre se realice de forma creciente, pues si se retrocede al ajustar un punto se pasará de la curva ascendente de imantación del hierro a la descendente con imprecisión en el trazo de la curva relevada. Luego de llegar a dicha tensión se procede a obtener los datos de la curva descendente hasta un valor mínimo relevándose los valores de la FEM en el descenso.

Durante el transcurso de este ensayo los estudiantes deben ir registrando los datos en lápiz y papel, variando la corriente de excitación de a pasos y relevando los datos de las distintos instrumentos. A posteriori se realiza el gráfico donde se visualiza la curva de vacío.

### 1.2.2. Ensayo de cortocircuito

La realización de este ensayo consiste en cortocircuitar los bornes de salida de la máquina sincrónica, intercalando entre dos bornes un amperímetro con alcance adecuado para sensar la corriente nominal de la máquina. Luego se pone el grupo en marcha a la velocidad nominal mediante el arrastre del motor de continua y se va incrementando suavemente la excitación hasta obtener el valor corriente nominal en bornes de la máquina. Se toman valores intermedios menores que el nominal con lo que se obtendrá el gráfico de corriente de inducido en función de la de excitación.

Análogamente al caso anterior, los datos fueron primero relevados y luego graficados por los estudiantes.

### 1.2.3. Ensayo con carga resistiva

Se pone en funcionamiento la máquina sincrónica arrastrada por el motor de continua, y se fija el punto inicial del funcionamiento en que la corriente de excitación genere en la máquina la tensión nominal. Luego manteniendo la misma corriente de excitación y ajustando para cada variación de carga la velocidad a la nominal, se relevan los valores de tensión, corriente y potencia.

### 1.2.4. Puesta en paralelo de dos MS

Para este ensayo se consideran dos grupos generadores conformados por una máquina sincrónica y una máquina de arrastre continua cada uno, tal como los descritos en la sección 1.1.1. Se realiza el conexionado a fin de poder controlar ambos motores de arrastre de forma independiente. Luego se procede a realizar el conexionado entre ambos alternadores (por medio de un interruptor el cual cerrará el paralelo) y a un equipo de sincronización.

## 1.3. Instrumentos y Medidas

Teniendo realizadas las conexiones, se ponen ambos grupos en marcha, llevando uno de los grupos (grupo 1) a velocidad nominal. A continuación se pone en marcha el segundo grupo (grupo 2) en el mismo punto de funcionamiento que el primer grupo, comenzando luego el procedimiento de sincronización hasta asegurar que están dadas las condiciones para cerrar el interruptor que une ambos alternadores trifásicos. Una vez que se compruebe esto se cerrará el interruptor trifásico con lo que ambas máquinas comenzarán a operar en paralelo.

Luego de la puesta en paralelo se realizan variaciones con las variables disponibles (tensión de inducido y corriente de excitación en la máquina de continua y corriente de excitación en el alternador) y se registran los cambios que ocurren en el punto de funcionamiento del conjunto (tensión y frecuencia generadas por los alternadores en paralelo). También se registran las variaciones de potencia activa que fluye entre los alternadores al realizar estas variaciones.

## 1.3. Instrumentos y Medidas

### 1.3.1. Tensiones y corrientes trifásicas

Para valores trifásicos alternos (corrientes de línea, tensiones de fase/línea, etc), los estudiantes relevan los datos de un analizador de redes CVMk2 disponible en el Laboratorio (véase [4]). El mismo cuenta con una pantalla en la cual se pueden seleccionar los valores a visualizar, y se encuentra montado en el panel de protecciones, comando y medidas antes mencionado.

El equipo cuenta con varias funcionalidades, entre ellas, desplegar los fasores correspondientes a las tensiones y corrientes, visualizar contenido armónico, potencias instantáneas, entre otras. Además el CVMk2 puede transmitir los valores de las variables medidas a través de un puerto serial RS-485 utilizando protocolo Modbus-RTU

### 1.3.2. Tensiones y corrientes continuas

Los valores de tensiones y corrientes continuas pueden ser medidos por instrumentos analógicos (de hierro móvil), digitales o utilizando multímetros (pinza amperimétrica para el caso de las corrientes).

Como se dijo anteriormente, todos estos instrumentos están distribuidos entre la mesa de trabajo, panel móvil y fuente de continua; lo cual hace difícil su lectura simultánea.

### 1.3.3. Medidas correspondientes a variables no-eléctricas

Para valores correspondientes a medidas no-eléctricas, las dos variables más relevantes para las aplicaciones del Laboratorio son la temperatura del bobinado de las máquinas, y la velocidad de rotor.

## Capítulo 1. Introducción

Para las velocidades, se cuenta con tacómetros digitales y encoders, de los cuales estos últimos no se utilizaban para los ensayos del curso previo a la realización de este proyecto.

Para las temperaturas, se cuenta con termocuplas para medir temperatura por contacto y también con una cámara termográfica, la cual por el momento no se utiliza en el curso.

### 1.4. Objetivos de Proyecto

Del análisis de las secciones anteriores, se puede ver claramente que es difícil para los estudiantes comprender e interpretar lo que sucede durante el transcurso de una práctica. Al realizar las actividades de la forma antes descrita se pierden y pasan por alto eventos relevantes que ocurren durante el desarrollo de la misma, como puede ser la comparación de modelos teóricos con medidas reales, también el hecho de tener que concentrar la atención en relevar medidas en diferentes instrumentos de forma simultánea. Por otro lado, el analizar los valores obtenidos en tiempos diferentes de realizados los ensayos se pierde la continuidad del trabajo.

Para lograr que la práctica en el curso se realice de manera mas didáctica para el estudiante y en un menor tiempo, es que se propuso un software que presente todos los valores medidos durante los ensayos en un entorno que sea de fácil lectura. Se definieron entonces los ensayos y/o pruebas que están dentro del alcance de este proyecto y son:

- Ensayo de vacío de máquina síncrona
- Ensayo de cortocircuito de máquina síncrona
- Puesta en paralelo de dos MS
- Diagrama P-Q de máquina síncrona (actualmente no realizado por no contar con una aplicación que permita visualizarlo)

Además, se buscó que los estudiantes logaran asociar herramientas teóricas con el funcionamiento real de las máquinas, y para esto el programa calcula y despliega los diagramas fasoriales correspondientes a los modelos de Behn-Eschenburg y Potier de la MS. Los mismos varían acorde a los valores instantáneos de las tensiones y corrientes medidas, calculándose los fasores en base a parámetros del modelo previamente cargados en el software.

Para implementar el software de visualización, se eligió el entorno Labview por considerar que permite lograr lo requerido para la interfaz gráfica.

Para poder desplegar las variables en el software, es necesario antes que nada necesario un equipo que las adquiera. Primero se definieron las variables que debe medir dicho equipo:



- Tensiones y corrientes trifásicas, potencia activa y reactiva entregadas por la MS.
- Corrientes de excitación, corrientes y tensiones de armadura de la MCC.
- Velocidad rotacional de eje
- Temperatura de bobinados a definir

El Laboratorio cuenta con equipos existentes que tienen capacidad de comunicar los valores medidos hacia otros equipamientos que son el encoder para la velocidad rotacional y el CVMk2 para las variables trifásicas.

Los requerimientos para el hardware a implementar entonces:

- Poder comunicarse con el CVMk2 a través del protocolo Modbus
- Interpretar las señales de salida del encoder
- Sensar temperatura
- Medir corrientes/tensiones continuas (6 corrientes y 2 tensiones)

Esto permitió obtener todas las variables para ser enviadas al SW.

## 1.5. Resumen

En el presente capítulo se describieron los equipos e instrumentos utilizados, los ensayos realizados en el curso Taller de Máquinas Eléctricas y la problemática que surge a partir de los mismos.

Se definió los requerimientos que debe cumplir tanto el programa de visualización como el equipo adquirente de datos.

En los próximos capítulos se dará una descripción detallada de todo lo necesario para implementar el sistema aquí propuesto, y para que opere de la manera deseada.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 2

## Arquitectura del Sistema

### Introducción

En la realización de este proyecto, fue necesario diseñar un sistema en el cual las variables físicas medidas sean desplegadas en una pantalla.

La disposición en que los distintos elementos tanto físicos (hardware), como visuales y programáticos (software) que se conectan e intercambian información se denomina Arquitectura del Sistema.

En el transcurso de este capítulo se presentan las distintas arquitecturas que fueron evaluadas y que cumplen con los requerimientos del proyecto, en todos los casos se contempló la integración del analizador de red CVMK2.

A partir de estas evaluaciones se definió una arquitectura en particular sobre la cual se elaboró el proyecto.

### 2.1. Elementos de sistema

Para entender qué es necesario implementar, se comenzó el razonamiento a la inversa: desde la visualización final hacia atrás.

El primer elemento necesario es un software que permita presentar los datos de una manera amigable para el usuario final.

Se decidió programar el software en LabVIEW, una plataforma de diseño de sistemas en base a programación gráfica que permite crear pantallas y gráficos personalizados para despliegue de medidas. Internamente la aplicación desarrollada estará también encargada del tratamiento de los datos 'crudos' recibidos desde los distintos elementos físicos.

El siguiente paso es conseguir que los datos lleguen y se adquieran en la PC, para esto se eligió que el medio físico por el cual se transmiten sea un cable USB, mientras que el software desarrollado se encarga de la comunicación necesaria para la adquisición.

Los elementos restantes son los equipos que reciben las medidas directas (sensores) y equipos de comunicación hacia la PC. Para esto fue necesaria una evaluación del hardware existente en el Laboratorio del IIE y del disponible en plaza.

## Capítulo 2. Arquitectura del Sistema

### 2.1.1. Equipos analizados

#### Equipos en Laboratorio

El equipo existente en el Laboratorio de Máquinas Eléctricas del IIE que forma parte de las actividades mencionadas y formó parte del sistema a desarrollar es el Analizador de Red Circutor CVMk2.

Este equipo releva todas las variables trifásicas y cuenta con un puerto bajo el standard RS-485 el cual es capaz de comunicarse mediante protocolo Modbus. En el Capítulo (3) se realiza una descripción detallada del equipo.

#### Placa Arduino

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

El mismo se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un computador.

En la Figura 2.1 se puede ver la plataforma de desarrollo para Arduino y en Figura 2.2 la plaqueta física (ver referencia [1]).

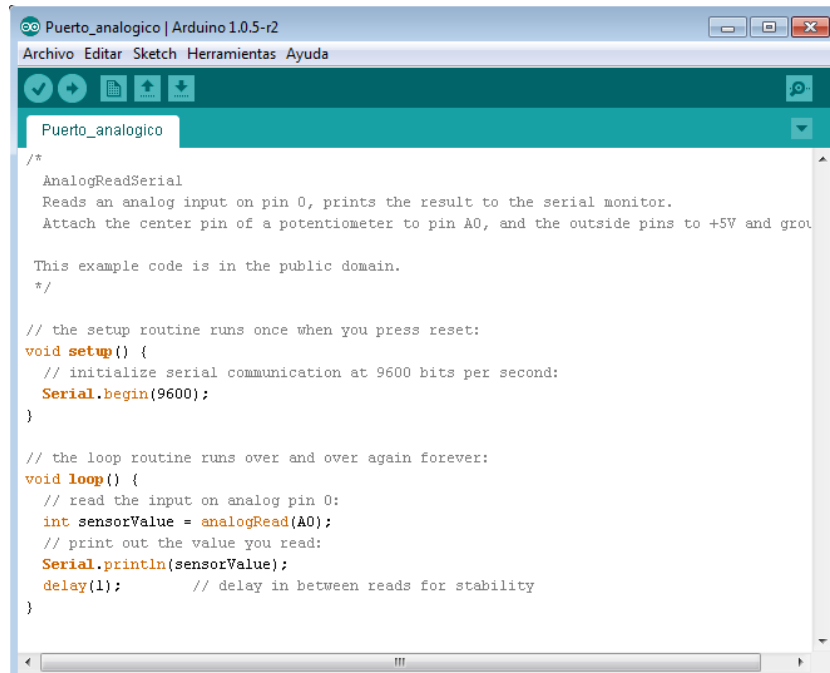
#### Placa Raspberry Pi

El Raspberry Pi es una computadora del tamaño de una tarjeta de crédito desarrollada en el Reino Unido por la Raspberry Pi Foundation, con el objetivo de enseñar ciencias de la computación en escuelas (ver referencia [6], [12]).

Existen varios modelos en producción pero la mayoría de las placas están basadas en un Broadcom BCM2835 system on a chip (SoC), que incluye un procesador ARM11 A6 de 700Mhz, GPU (Graphics processing unit) VideoCore IV, y dependiendo del modelo entre 256 y 512 Mb de RAM. También cuenta con un slot para tarjetas SD, donde se guarda el sistema operativo y la memoria funcional del computador.

La placa también cuenta con varios conectores, como pueden ser (para el caso del modelo B+) 4 conectores USB tipo A, un conector tipo micro-B para darle alimentación a la placa, un puerto Ethernet, una salida HDMI para el video y un

## 2.1. Elementos de sistema



```
Puerto_analogico | Arduino 1.0.5-r2
Archivo Editar Sketch Herramientas Ayuda

Puerto_analogico
/*
 AnalogReadSerial
 Reads an analog input on pin 0, prints the result to the serial monitor.
 Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

 This example code is in the public domain.
 */

// the setup routine runs once when you press reset:
void setup() {
 // initialize serial communication at 9600 bits per second:
 Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
 // read the input on analog pin 0:
 int sensorValue = analogRead(A0);
 // print out the value you read:
 Serial.println(sensorValue);
 delay(1);      // delay in between reads for stability
}
```

Figura 2.1: Entorno de desarrollo para Arduino

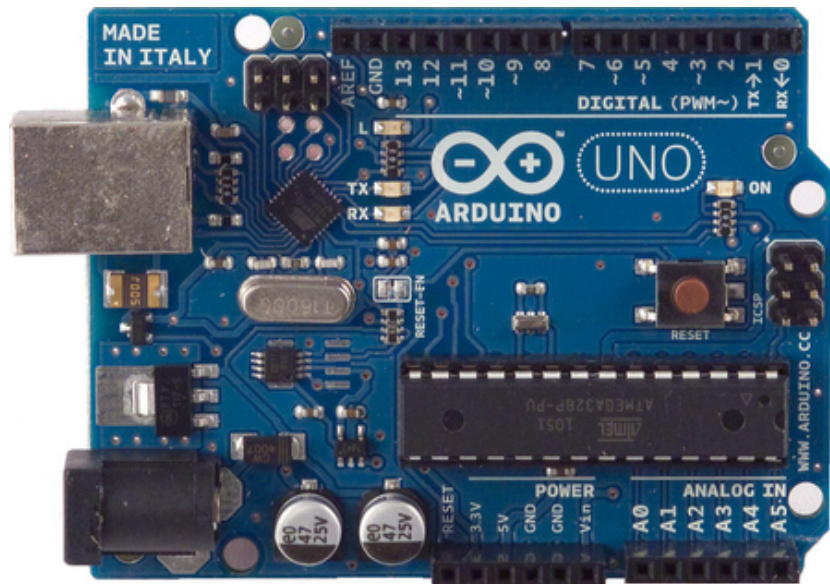


Figura 2.2: Vista de placa Arduino Uno



## 2.2. Análisis de las arquitecturas del Sistema

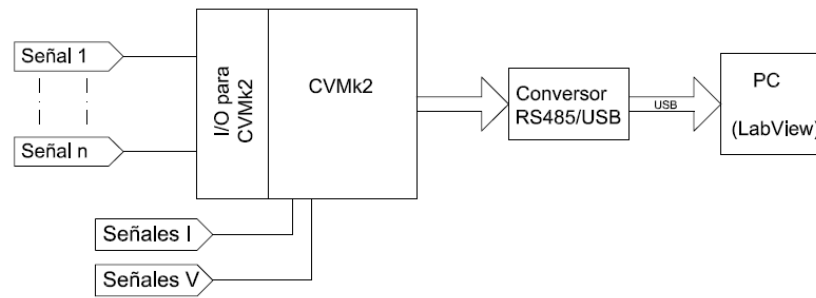


Figura 2.4: Esquema del Sistema 1

### Arquitectura 2 (Embebido)

En esta arquitectura, llamada coloquialmente embebida, se tiene un sistema totalmente independiente en donde la visualización del software solo requeriría una pantalla o proyector y periféricos como mouse y teclado para su funcionamiento.

El mismo consta de una PC basada en Linux (Raspberry Pi) la cual se comunica directamente con el CVMk2 a través RS-485 por MODBUS, y paralelamente con el resto de los sensores; a su vez sobre la misma placa en el sistema operativo corre el software de LabVIEW que se diseña para tal fin a medida.

De esta manera se tiene una implementación que solo necesita de periféricos (pantalla, teclado, etc) para interactuar con el exterior, y su arquitectura no depende de la conectividad o compatibilidades de futuros PCs.

Se muestra en Figura 2.5 el diagrama de bloques del Sistema.

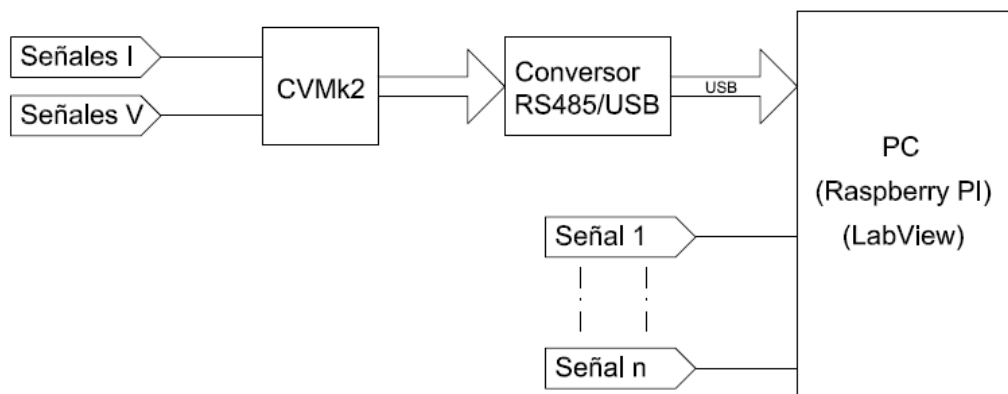


Figura 2.5: Esquema del Sistema 2

### Arquitectura 3

Esta arquitectura consiste en una placa receptora, compuesta por una placa Arduino Uno y un conversor RS-485/RS-232 que será capaz de adquirir señales

## Capítulo 2. Arquitectura del Sistema

analógicas, recibir la información que nos brinda el equipo CVMk2 a través de un puerto serie RS-485 y comunicarse con el PC por medio de un puerto USB. Obteniendo así toda la información requerida en la PC. El diagrama de bloques de este Sistema se observa en la Figura 2.6.

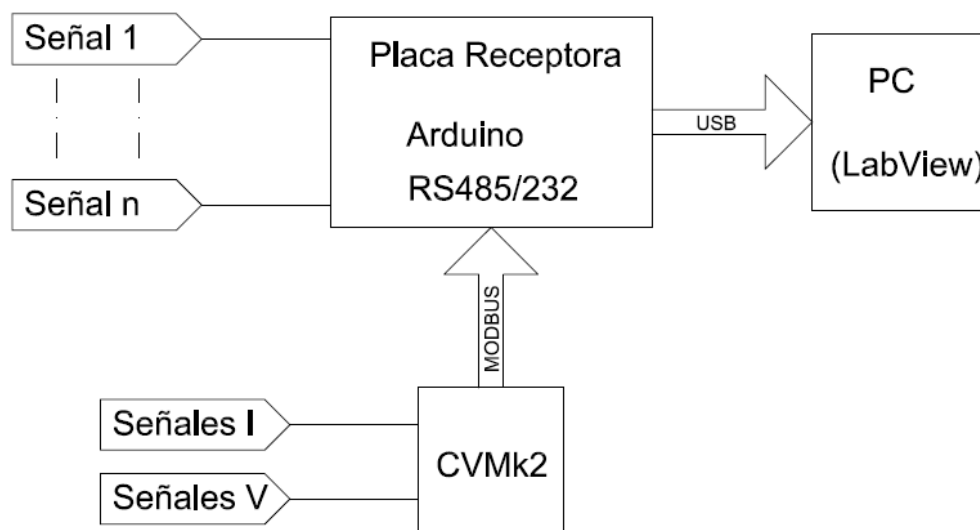


Figura 2.6: Esquema del Sistema 3

### Arquitectura 4

Esta arquitectura se compone de una placa Arduino Uno y el equipo CVMk2. La placa Arduino se encarga de sensar y relevar las señales analógicas extras necesarias y las transfiere mediante puerto USB al PC, por otro lado el CVMk2 comunica su información a la PC por medio de los conversores RS-485/RS-232/USB. Teniendo así toda la información requerida en el PC para ser procesada en el software.

El diagrama de bloques del Sistema se observa en Figura 2.7.

## 2.3. Evaluación de las arquitecturas a implementar del Sistema

A continuación se analizarán las distintas arquitecturas descritas anteriormente señalando las ventajas y desventajas que ofrecen cada una.

Este análisis se realizó luego de haber obtenido una comunicación fluida entre el equipo CVMk2 y el PC (ver sección:4.4.1).



## 2.3. Evaluación de las arquitecturas a implementar del Sistema

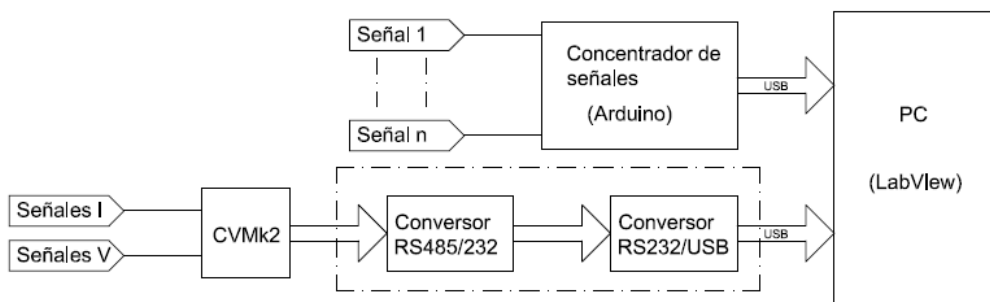


Figura 2.7: Esquema del Sistema 4

### Sistema 1

Como se mencionó anteriormente este sistema es el más sencillo de implementar desde el punto de vista del hardware, pero no es posible adquirir la placa expansora en plaza. En estas condiciones resulta inviable su implementación dado que esta pieza resulta fundamental para la adquisición de todas las señales necesarias, por lo que se terminó descartando esta opción.

### Sistema 2

Esta implementación del sistema es físicamente viable de construir, pero debido a que el procesador del Raspberry Pi usa una arquitectura con un conjunto de instrucciones del tipo ARM Av6, el sistema no es compatible con LabVIEW al no poder compilar programas para esa arquitectura. Por tal motivo se descartó esta implementación.

### Sistema 3

Esta arquitectura de sistema resultó con buena viabilidad con los elementos descritos, pero de implementación más compleja que los sistemas 1 y 4.

Esta arquitectura es la más cercana a lo que sería una de tipo embebido.

Como hardware adicional solo sería necesaria una PC completa con puerto USB.

### Sistema 4

Este Sistema fue probado casi en su totalidad por lo que se consideró factible en un primer momento.

Como se observa en Figura 2.7 éste necesita dos vías de comunicación, por un lado comunicar el equipo CVMk2 a un puerto USB de la PC (ver Sección 4.4.1) y por otro lado otro puerto USB para el Arduino (ver Sección 4.4.2), para luego ser monitoreados en ambos casos por LabVIEW.

## Capítulo 2. Arquitectura del Sistema

### 2.3.1. Selección de Sistema

Durante el análisis para la selección de la arquitectura a implementar se tuvieron en cuenta varios aspectos, como ser robustez del hardware, compatibilidad con periféricos que interactúa, simplicidad del conexionado y costos.

En cuanto a la robustez del hardware y simplicidad del conexionado fue tenido en cuenta que los usuarios del mismo serán estudiantes con lo que debe ser de conexionado y puesta en marcha de manera relativamente sencilla.

Para lograr compatibilidad razonable en el tiempo y así cumplir con uno de los objetivos del proyecto se decidió desarrollar el software en LabVIEW compilando para correr en SO Windows y utilizando para la comunicación puerto USB.

Teniendo en cuenta el aspecto de los costos, en el caso de adquirir la placa expansora para implementar por ejemplo el Sistema 1 no se tuvo respuesta por parte del representante y del fabricante, por lo que se tuvo que evaluar las otras arquitecturas.

Como segunda opción se tomó la decisión de trabajar con una placa Arduino cuyo costo es razonable dadas sus prestaciones y el alcance del Proyecto.

Teniendo en cuenta los aspectos mencionados, las arquitecturas que más se adecuaron a las necesidades planteadas del Proyecto son la de los Sistemas 3 y 4.

De los sistemas planteados como adecuados se optó por el 3, dado que el 4 tiene la desventaja de acaparar dos puertos USB de la PC para su comunicación.

Finalmente analizando la cantidad de variables a leer y la cantidad que le es posible concentrar el Arduino Uno, se debió optar por otra placa Arduino( Arduino Mega 2560) que se describirá en el Capítulo 3.

Al realizar este cambio ya no es necesaria la conversión RS-485 a RS-232 debido a las prestaciones de esta nueva placa desde el punto de vista de comunicación.

## 2.4. Resumen

En el transcurso del presente capítulo se estudiaron las distintas arquitecturas consideradas del sistema, cumpliendo cada una de ellas con los objetivos planteados.

Luego del análisis se optó por la arquitectura presentada como Sistema 3, la cual se muestra en figura 2.8, con la salvedad de reemplazar la plaqueta Arduino Uno por la Arduino Mega 2560.

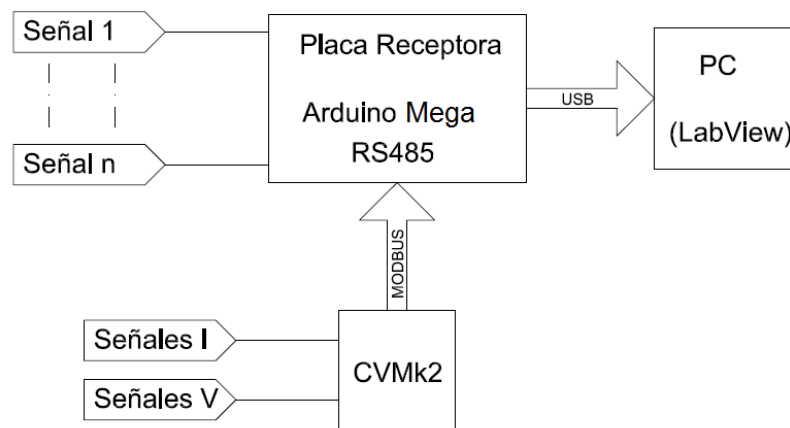


Figura 2.8: Esquema de la arquitectura definida para el Sistema

En próximos capítulos se detallarán todos los componentes necesarios para el correcto funcionamiento del dispositivo a crear y los resultados del testeado del mismo, basándose en la arquitectura elegida.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 3

## Hardware

### Introducción

Una vez definida la arquitectura del Sistema (Capítulo 2), se procede al diseño e implementación del mismo.

Como se definió anteriormente, se utilizó con una placa Arduino Mega 2560, la cual cumple el rol de concentradora de señales, tanto de las tensiones y corrientes alternas provenientes del CMV<sub>k</sub>2, como también las tensiones y corrientes continuas, variables físicas (velocidad del eje) y térmicas (temperatura bobinado).

Todos estos elementos combinados con una serie de borneras/terminales, circuitos impresos hechos a medida, fuente, entre otros, los cuales son esenciales para el correcto funcionamiento de todo el Sistema, se describen en el transcurso de este capítulo.

Del análisis de la arquitectura seleccionada, para realizar su implementación física surgen los siguientes elementos principales:

- Placa Arduino Mega 2560
- Placa IO
- Fuente de 5 V
- Caja PVC quien dará estructura y soporte al hardware

Estos elementos en conjunto con otros, como pueden ser el cableado y las borneras, componen la interfaz de hardware necesaria para el correcto funcionamiento del Software en LabVIEW.

## 3.1. Componentes del Hardware

### 3.1.1. Arduino Mega 2560

El Arduino Mega 2560 que se muestra en la figura 3.1 es un microcontrolador basado en el ATmega2560 de Atmel. La placa cuenta con 54 pines de I/O (Input/Output) digitales de los cuales 16 pueden ser usados como salidas PWM, 16 AI (Analog Input), 4 UART (Puertos seriales), power jack y conector USB.

La programación del dispositivo se realiza (al igual que con otras placas Arduino) con el Arduino IDE (Integrated Development Environment), el cual tiene capacidad de adicionarle 'librerías' creadas por terceros incorporando funciones no nativas al IDE original.

En la página web de Arduino se encuentra disponible toda la información sobre el mismo y se puede descargar de forma gratuita el software para la programación(ver referencia [1]).

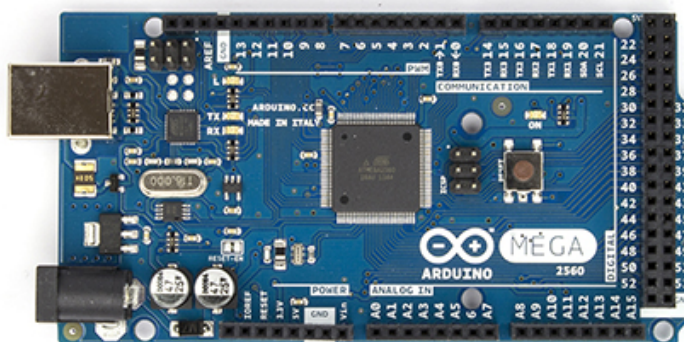


Figura 3.1: Placa Arduino Mega 2560

En el Sistema desarrollado cada pin de la placa, sea digital, analógico o de comunicación, tiene una tarea bien definida ya sea porque recibe la señal de un sensor directamente o recibe una medida por bus de comunicación (Serial o SMBus).

Para realizar la comunicación vía puerto serie RS-485 del analizador de red CVMk2 con la placa Arduino, se utilizó una Shield específica (ver figura 3.2), placa pre-preparada para conectarse sobre un Arduino. La misma realiza la conversión de RS-485 a UART, con esto se obtiene los datos del CVMk2.

### 3.1.2. Placa de comunicación Sistema-PC y alimentación de Arduino

Para la comunicación con el PC se desarrolló una placa la cual realiza el vínculo entre ambos. Ésta se compone de un puerto USB tipo B (hembra) donde se conecta la PC desde el exterior, y un puerto USB tipo A internamente donde se conecta la placa Arduino con la salvedad de que se aislaron los pines de potencia.

### 3.1. Componentes del Hardware

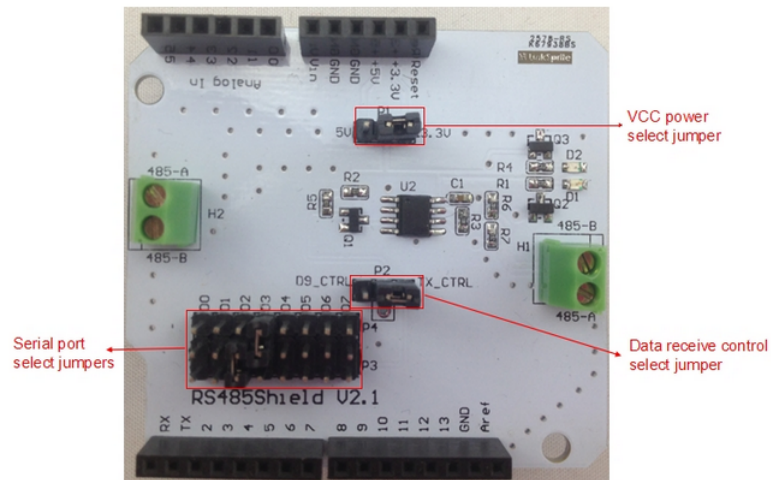


Figura 3.2: Shield RS485 para realizar la conversión de RS-485 a UART

La aislación de los pines de potencia se debe fundamentalmente a la necesidad de independizar la alimentación de la placa Arduino desde el puerto USB, con esto se logra mantener el hardware diseñado en funcionamiento independientemente de si está o no conectada la PC al puerto.

Por otro lado la placa Arduino admite ser alimentada de varias maneras, puede ser mediante un voltaje entre 7 y 12 Vdc tanto en su pin de Vin como en el conector tipo power Jack, o de lo contrario de la propia tensión de 5 Vdc proveniente del puerto USB.

Como los sensores de corriente y encoders utilizados funcionan con 5 Vdc de alimentación y se desarrolló una fuente para tal fin (ver sección 3.1.4), se decidió utilizar la misma fuente para alimentar la placa Arduino a través del puerto USB. Con esto se logró tener un sistema con una sola referencia de tensión (la fuente del Sistema) en todo el hardware.

En la figura 3.3 se muestra el circuito mediante el cual se realiza la comunicación y alimentación, donde se puede observar la aislación de los pines de puerto y el puente con la fuente de 5 Vdc desarrollada .

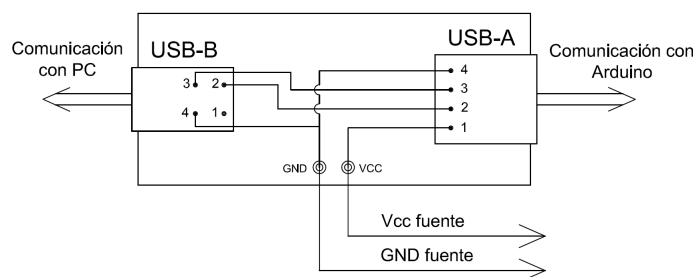


Figura 3.3: Esquemático de placa de Comunicación y Alimentación de Arduino. Referencias de pines: 1- Vcc (5 Vdc), 2- -Data, 3- +Data, 4- GND

### 3.1.3. Placa IO (Input/Output)

Esta placa surgió de la necesidad de concentrar y tratar las diferentes señales que adquiere el Sistema (no adquiridas por el CVMk2) para posteriormente ser enviadas a la placa Arduino. Algunas de ellas se envían de forma directa mientras que a otras se les realiza algún tratamiento según corresponda.

En ésta se implementaron los circuitos correspondientes a la recepción de las dos tensiones DC a medir, las cuales llegan a través del conector de 4 bornes, denominados en la figura 3.4 como VDC. Al ingresar las señales inmediatamente son rectificadas dado que se definió que el sensado de las tensiones DC no sean polarizados. Luego se tiene una resistencia de  $200\text{ k}\Omega$  (el rango de entrada a medir será de 50 a 250V) y finalmente el diodo foto-emisor del optoacoplador. Además se colocó un capacitor para estabilizar la tensión, con el cual se reduce el ruido proveniente de la propia señal a relevar y el inducido en los cables.

Como se puede observar en figura 3.1 los pines de la placa Arduino son de poca resistencia mecánica, para prevenir problemas en este aspecto se adquieren todas las señales desde esta placa mediante bornes atornillables, posteriormente se envían a la placa Arduino a través de un cableado de tipo bus (peine de varios hilos).

En la figura 3.5 se puede ver el circuito completo correspondiente a la Placa IO.

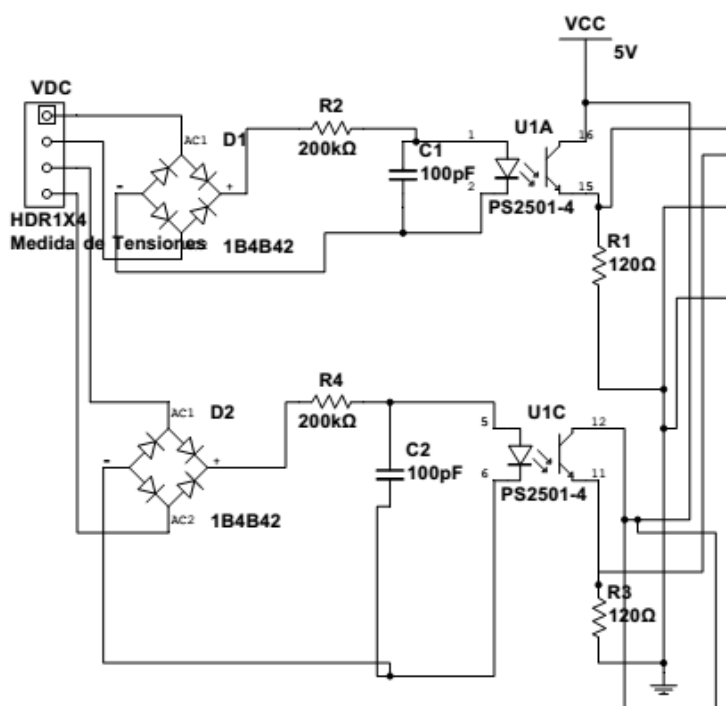


Figura 3.4: Circuito optoacoplado de sensado de corriente continua

Del lado de 5 V del optoacoplador (correspondiente a todo el circuito a la de-



### 3.1. Componentes del Hardware

recha del optoacoplador, ver figura 3.4) se coloca una resistencia adecuada (de  $120\ \Omega$ ) que dará el valor de tensión de la placa, el cual está controlado por la corriente que permite pasar el fototransistor. Físicamente los dos optoacopladores son parte del mismo encapsulado que cuenta con 4 optoacopladores en total (PS2501-4).

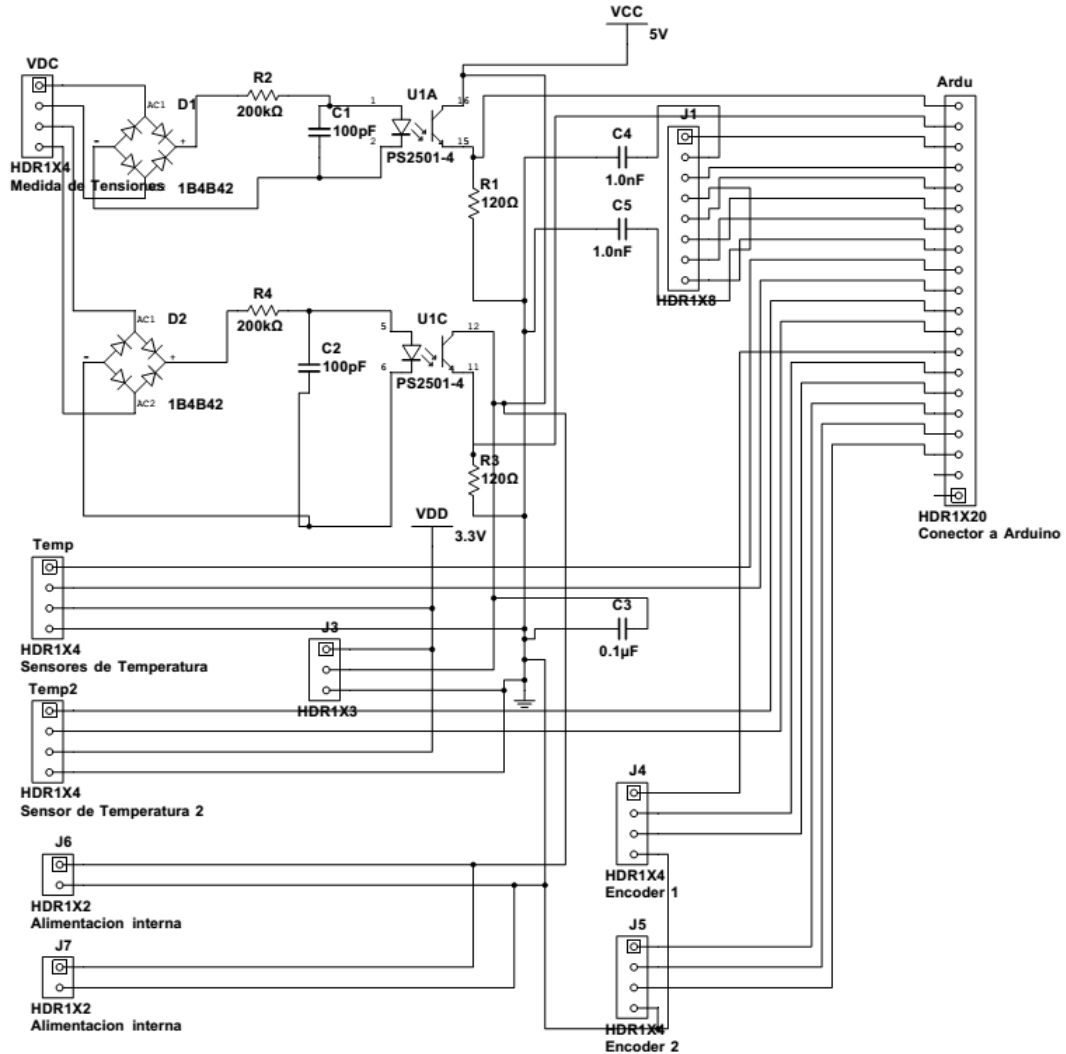


Figura 3.5: Circuito de Placa IO completo

A continuación se describen las restantes señales que adquiere la placa, las cuales constan del conexionado directo entre bornes atornillables y conectores para los cables del bus que comunica con la placa Arduino, con las correspondientes salvedades caso a caso:

- Encoders: se les distribuye tierra y tensión de alimentación (5 V).
- Sensor de temperatura: se le distribuye tensión de alimentación en 3,3 V recibida desde la placa Arduino y tierra, el resto de los componentes necesarios

### Capítulo 3. Hardware

(resistencias de pull-up del bus SMBus y capacitores de filtrado de ruido) se instalaron en la placa de montaje del sensor (ver 3.14).

- Sensores de corriente: se alimentan con Vcc (5 V) y GND (0 V) desde la fuente. Para el caso particular de los sensores de 5 A, los integrados requieren que se provea un capacitor entre tierra y su pin FILTER, el cual se lleva desde esta placa hasta el mismo.

En la figura 3.6 se muestra el circuito impreso de la placa.

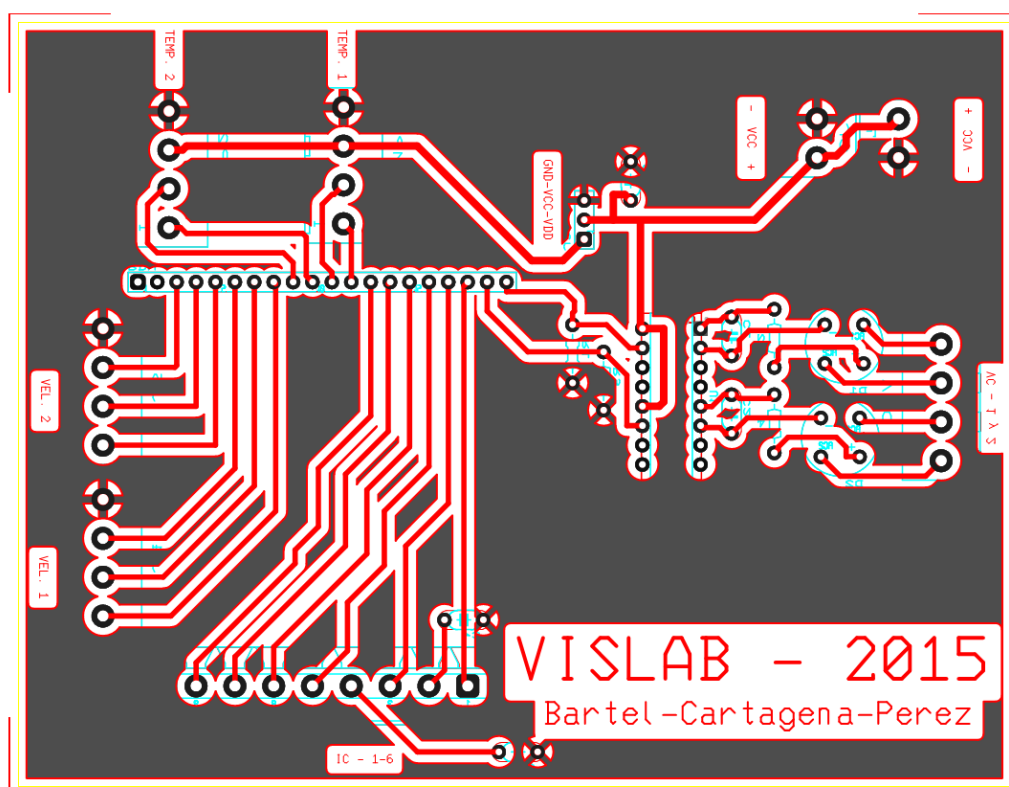


Figura 3.6: Circuito CAD de la Placa IO para armado de la misma

### 3.1.4. Fuente de 5 Vdc

Los circuitos internos del equipo necesitan una alimentación de 5 Vdc para operar, salvo el sensor de temperatura como se comentó en las secciones 3.5. Si bien la placa Arduino cuenta con un pin de 5 V y puede alimentarse desde el puerto USB, se optó que el sistema tenga una fuente interna para así hacerlo independiente del exterior y además no sobrecargar al Arduino.

Por lo antes mencionado se realizó el diseño de una fuente de tensión continua de 5 Vdc a partir de la tensión de red alterna de 230 V.

El circuito a la entrada cuenta con un transformador que reduce la tensión de 220/12 Vac de 5 W de potencia, luego se rectifica con un rectificador de onda completa (modelo KBPC1005) para luego suavizar el ripple de la tensión mediante un capacitor de  $2200\ \mu F$ , finalmente la misma se ajusta a 5 Vdc con un regulador lineal (LM7805).

En la figura 3.7 se puede ver el circuito correspondiente a la fuente.

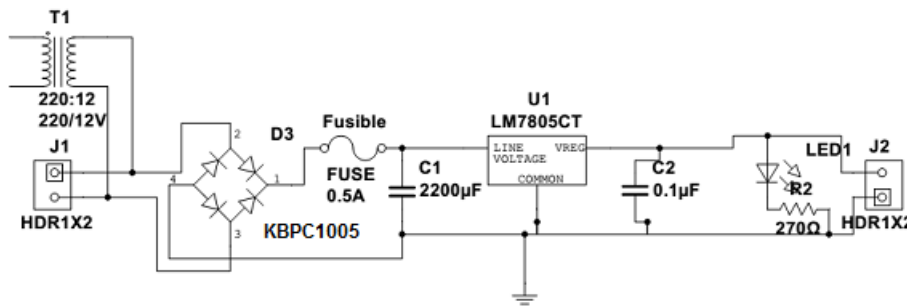


Figura 3.7: Circuito de la Fuente original

En un peor caso el circuito tendrá un consumo de 0.5 A. El transformador entrega un voltaje en su secundario de 12 Vrms, lo que corresponde a un pico de 17 V el cual es rectificado. Dado un capacitor de  $2200\ \mu F$  y un consumo de 0.5 A, se tiene un valor de ripple de 2.27 V a la entrada del regulador.

El regulador L7805 asegura una tensión típica de 5 V para valores de entrada que pueden variar entre 8 y 20 V, entonces el peor caso en estas condiciones es con una tensión de 9.73 V la cual está dentro del rango.

En todos los casos los componentes están sobredimensionados, dado que: el regulador puede entregar una corriente de hasta 1.5 A, el puente rectificador (modelo KBPC1005) es de 3 A con una tensión de 20 Vrms, y el transformador es de 5 W.

Además de los componentes antes mencionados, el circuito cuenta a la salida del regulador con un capacitor para filtrado de altas frecuencias en la tensión. Los datos del rectificador se encuevan en la figura 3.8.

### Capítulo 3. Hardware

Regulador de tensión LM7805CT	V <sub>mín</sub>	V <sub>typ</sub>	V <sub>máx</sub>
	4,65	5	5,35

Figura 3.8: Datos típicos del regulador de tensión de la fuente (@25 °C)

Finalmente se agregó un fotodiodo (led) y una resistencia en paralelo que funciona como indicador de encendido de la fuente.

En la implementación del circuito también se agregó un fusible de protección de 0.5 A entre el puente rectificador y el capacitor C1 para limitar el consumo del resto del circuito y así proteger aguas arriba de éste.

En el transcurso de las pruebas de todo el equipo, se encontraron problemas de ruido en la medida del Arduino. Para aplacar estos problemas, se instaló un capacitor a la salida del regulador (en paralelo con el capacitor de 0.1  $\mu F$  ya existente) de 2200 $\mu F$ , el cual redujo el ruido de 100 a 50 mVpp. Finalmente el circuito quedó como se muestra en la figura 3.9

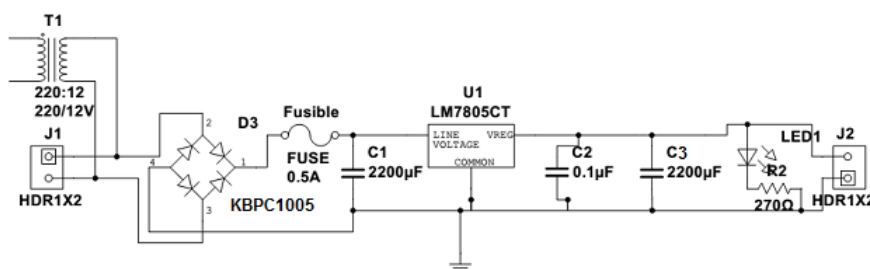


Figura 3.9: Circuito de la Fuente final implementado

Los valores relevados del circuito final en operación se indican en Tabla 3.1.

V de secundario del trafo	12.1 Vrms
V de puente de diodo	13.1 V
V de salida de regulador	5.0 V
I de carga (salida de regulador)	160 mA

Tabla 3.1: Valores de fuente medidos con el equipo en operación

Al momento del revelamiento de los datos, los Encoders no se estaban alimentado, pero según su hoja de datos el consumo máximo de los mismos es de 80 mA, por lo que la fuente es capaz de alimentar todo el sistema.

Para un valor típico de 400mA que es capaz de suministrar el Arduino, se puede ver que la fuente es necesaria para evitar que la placa trabaje cercana a su corriente máxima de régimen, dado que el valor esperado de consumo durante un ensayo sera 320mA (sumando aporte de Encoders).

#### 3.1.5. Sensores y Periféricos

##### Analizador de red CVMk2



Figura 3.10: Imagen frontal CVMk2

Este equipo mostrado en la figura 3.10 es un analizador de red el cual mide, calcula y muestra los principales parámetro eléctrico de redes industriales trifásicas equilibradas o desequilibradas. Las medidas se realizan en verdadero valor eficaz (true RMS) mediante tres entradas de tensión alterna, dos referencias de tensión (neutro y tierra), y cuatro entradas de corrientes para las medidas de secundarios .../1 A ó .../5 A, procedentes de los transformadores de corriente exteriores (ver referencias [4]).

A continuación se describen las características relevantes del dispositivo:

- Medición en verdadero valor eficaz.
- Clase 0.2 en Potencia y Energía.
- Valor instantáneo con fecha y hora.
- Comunicación RS-485 (Modbus/RTU).
- Ampliable mediante tarjeta de expansión de entrada/salida, memoria y comunicaciones.

Entre los parámetros que puede relevar se encuentran:

- Tensión simple y compuesta
- Corriente
- Frecuencia

### Capítulo 3. Hardware

- Potencia Activa, Inductiva, Capacitiva, Aparente (consumo y generación)
- factor de potencia
- THD de tensión y corriente (RMS y fundamental)
- Armónicos de tensión y corriente (2°...50°)

Condiciones de operación:

- Tensión nominal estándar 300/520  $V_{f-N}/V_{f-f}$
- Frecuencia nominal 45 a 65 Hz
- Secundarios .../5 A  $5A_{ca}$
- Secundarios .../1 A  $1A_{ca}$

<b>ENTRADAS DE TENSIÓN</b>	
Margen de medida	del 5 al 120% de $U_n$ para $U_n = 300$ V ac (f-N) del 5 al 120% de $U_n$ para $U_n = 520$ V ac (f-f)
Frecuencia	45...65 Hz
Tensión máxima medida	360 V c.a.
Sobretensión admisible	750 V c.a.
Consumo máximo (corriente limitada)	< 0,6 V·A
<b>ENTRADAS DE CORRIENTE</b>	
Margen de medida	del 1 al 120% de $I_n$ para $I_n = 5$ A
Secundario de los TC ( $I_n$ )	1 ó 5 A
Corriente primaria medida	Programable < 30.000 A
Sobrecarga admisible	6 A permanente, 100 A $t < 1$ s
Consumo	< 0,45 V·A
<b>ALIMENTACIÓN AUXILIAR</b>	
Alimentación	85 a 265 V c.a. (50...60Hz) (consumo < 30 VA) 100 a 300 V c.c. (consumo < 25 W)
<b>CONEXIONADO</b>	
Sección en conductor rígido	4.5 mm <sup>2</sup> (AWG 11)
Par de apriete del borne	0.8 Nm

Figura 3.11: Características Técnicas

### 3.1. Componentes del Hardware

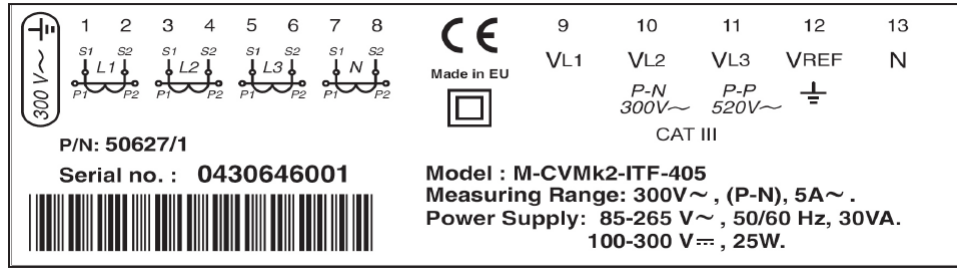


Figura 3.12: Etiqueta de conexión de Tensión y TI

BORNE	DESCRIPCIÓN
1	Conexión S1 de transformador de corriente de la fase L1
2	Conexión S2 de transformador de corriente de la fase L1
3	Conexión S1 de transformador de corriente de la fase L2
4	Conexión S2 de transformador de corriente de la fase L2
5	Conexión S1 de transformador de corriente de la fase L3
6	Conexión S2 de transformador de corriente de la fase L3
7	Conexión S1 de transformador de corriente de Neutro
8	Conexión S2 de transformador de corriente de Neutro
9	Entrada tensión de la fase L1
10	Entrada tensión de la fase L2
11	Entrada tensión de la fase L3
12	Entrada tensión $V_{REF}$ (GND)
13	Entrada tensión NEUTRO

Figura 3.13: Conexionado de medida en la red trifásica

#### Sensor de Temperatura MLX90614

Es un termómetro infrarrojo para realizar medidas de temperatura sin contacto.

Tanto el chip de la termopila sensible al IR como el acondicionador de señal integrado en el mismo encapsulado TO-39, cuenta con un amplificador de bajo ruido, un conversor analógico digital de 17bits y un poderoso DSP, que le permiten al termómetro alta precisión y resolución.

El mismo viene calibrado de fabrica con dos salidas, una digital PWM y una por SMBus. Además por defecto esta calibrado para una emisividad de 1 (correspondiente a un cuerpo negro), pero puede ser programado para otros valores entre 0.1 y 1 sin necesidad de recalibrado contra un cuerpo oscuro.

El encapsulado mostrado en figura 3.15a cuenta además con un filtro óptico que filtra la luz visible y las próximas al infrarrojo, lo que le permite ser inmune a la luz solar y de ambiente (el filtro pasabanda se encuentra entre los 5.5 y 14  $\mu\text{m}$  de longitud de onda).

Para la instalación del sensor se desarrolló una placa impresa que da soporte al encapsulado y a los componentes necesarios para su funcionamiento. En la figura

### Capítulo 3. Hardware

3.14 se muestra el diagrama de la misma y en la figura 3.15b la imagen del circuito ya montado donde se puede observar un acrílico que cumple función de protección mecánica.

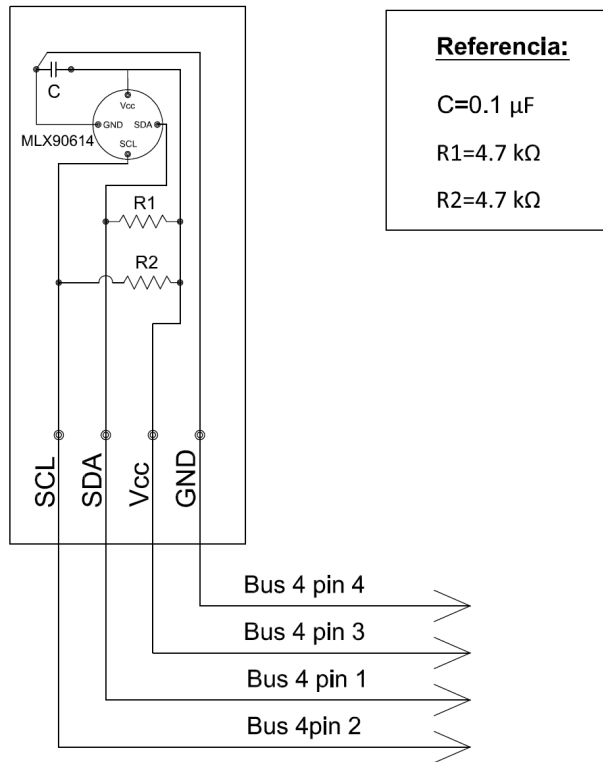
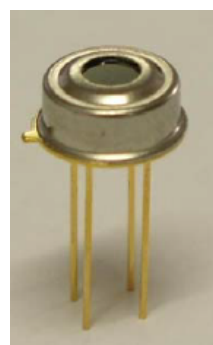


Figura 3.14: Diagrama de placa sensor de temperatura



(a)



(b)

Figura 3.15: 3.15a Encapsulado de sensor 3.15b Imagen del sensor montado en placa impresa



### 3.1. Componentes del Hardware

#### Sensor de Corriente de Efecto Hall ACS712

Estos sensores mostrados en figura 3.16 son dispositivos integrados económicos y precisos, empleados en aplicaciones industriales, comerciales y de sistemas de comunicación.

El dispositivo consiste de un circuito sensor Hall lineal, de precisión y de bajo offset, con un camino conductor de cobre próximo a la superficie del dado. La corriente aplicada a través del conductor de cobre genera un campo magnético que es sentido por el integrado y convertido en una señal de voltaje DC proporcional. La resistencia interna del camino de cobre es de  $1,2\text{m}\Omega$  típicamente, lo que le provee bajas pérdidas por disipación térmica, además el grosor del conductor de cobre le permite tener sobrecorrientes de hasta 5 veces su nominal.

Los terminales del sensor están eléctricamente aislados de las entradas de medida, lo que le permite ser utilizado directamente en aplicaciones en que se requiere aislamiento, sin ser necesario el uso de optoacopladores u otro método de aislación.

El ACS712 viene en un encapsulado tipo SOIC8 y calibrado de fábrica.

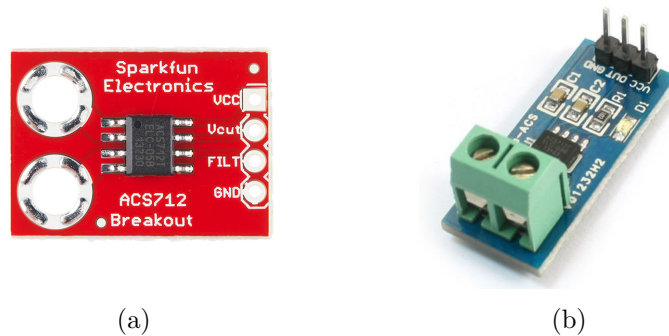


Figura 3.16: 3.16a Breakoutboard del sensor ACS712 de 5 A 3.16b Breakoutboard del sensor ACS712 de 20 y 30 A

#### Encoder E6B2-CWZ3E

El Encoder rotacional de tipo Incremental E6B2-C mostrado en figura 3.17, es un dispositivo de medida de velocidad rotacional de eje. Se instala acoplado al eje, y por cada revolución completa el dispositivo genera 2 trenes de pulsos desfasados entre si un cuarto de periodo en las salidas A y B. Este desfasaje indica el sentido de giro.

Por otra parte cuenta con una tercera salida que genera un pulso cada revolución completa de eje. El punto de la vuelta en el cual se genera este pulso es configurable durante el montaje del dispositivo.



Figura 3.17: Encoder E6B2-C

## 3.2. Arquitectura

En esta sección se describe el conexionado de los diferentes módulos que componen el Sistema como ser las placas impresas, periféricos descritos anteriormente y la interacción entre ellos.

### 3.2.1. Diagramas del Sistema

En la figura 3.19 se muestra el diagrama del Sistema compuesto por la placa concentradora (Placa IO), la placa de la Fuente DC y la placa de Comunicación con la PC.

### 3.2. Arquitectura

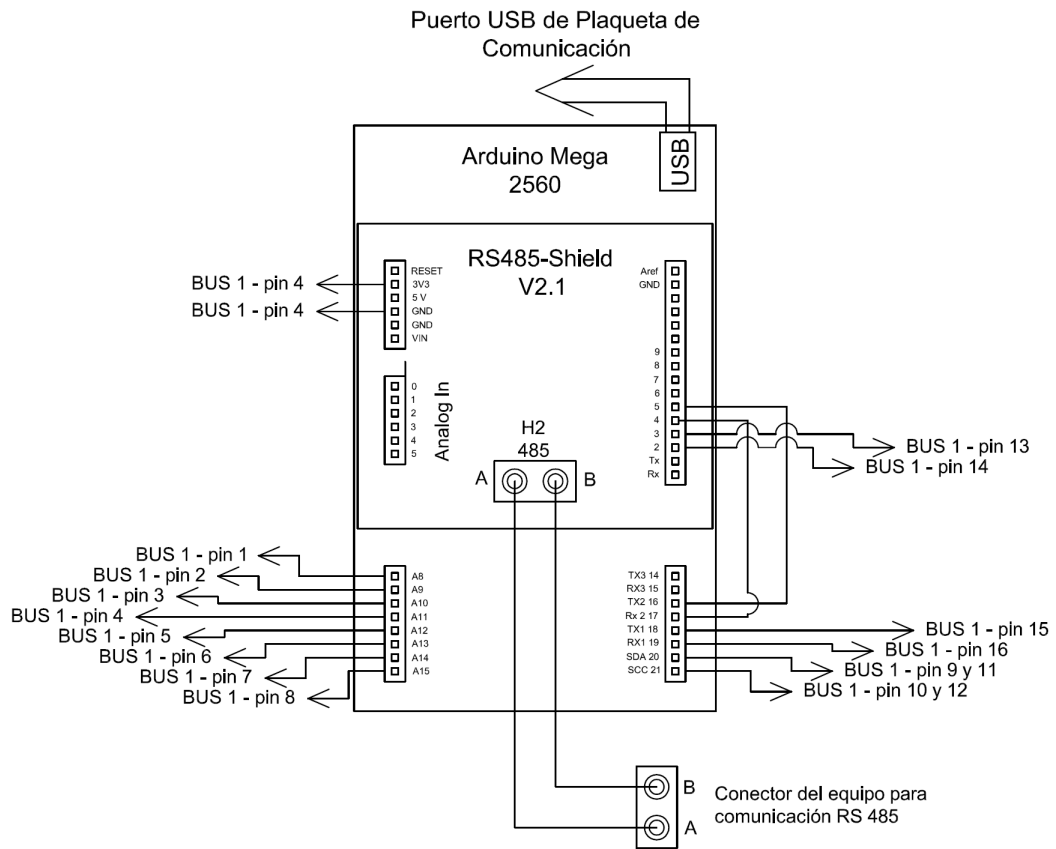


Figura 3.18: Diagrama de conexionado de la placa Arduino

En el esquemático de la figura: 3.18 se muestra el conexionado de la placa Arduino con el resto de los módulos que componen el Sistema, como ser el cableado con la placa concentradora de señales (Placa IO), placa de comunicación y a través del Shield RS485 para la comunicación con el equipo CVMk2.

La placa correspondiente a la Fuente de Potencia es la encargada de alimentar todo el Sistema, compuesta de un transformador relación 230/12 V, un circuito rectificador y regulador de tensión(ver sección 3.1.4).

Por otro lado la placa Concentradora (Placa IO) es la encargada de adquirir todas las señales proveniente de los bornes ubicados en el frente del equipo, desde donde se adquieren las tensiones y corrientes DC a medir, como también el bus de datos correspondiente de la temperatura y velocidades que se encuentran en el lateral derecho del equipo, para luego enviarlas a la placa Arduino. En la sección 3.5 se encuentra el diseño.

En la figura 3.18 se muestra la placa Arduino junto con la Shield RS485 la cual adquiere las señales de las medidas provenientes de la placa IO y a su vez se comunica con el CVMk2 solicitándole las medidas que éste realiza.

Luego de realizada la adquisición de los datos, éstos son enviados a la PC vía puerto USB. Esta comunicación se realiza desde la Placa de Comunicación (ver

### Capítulo 3. Hardware

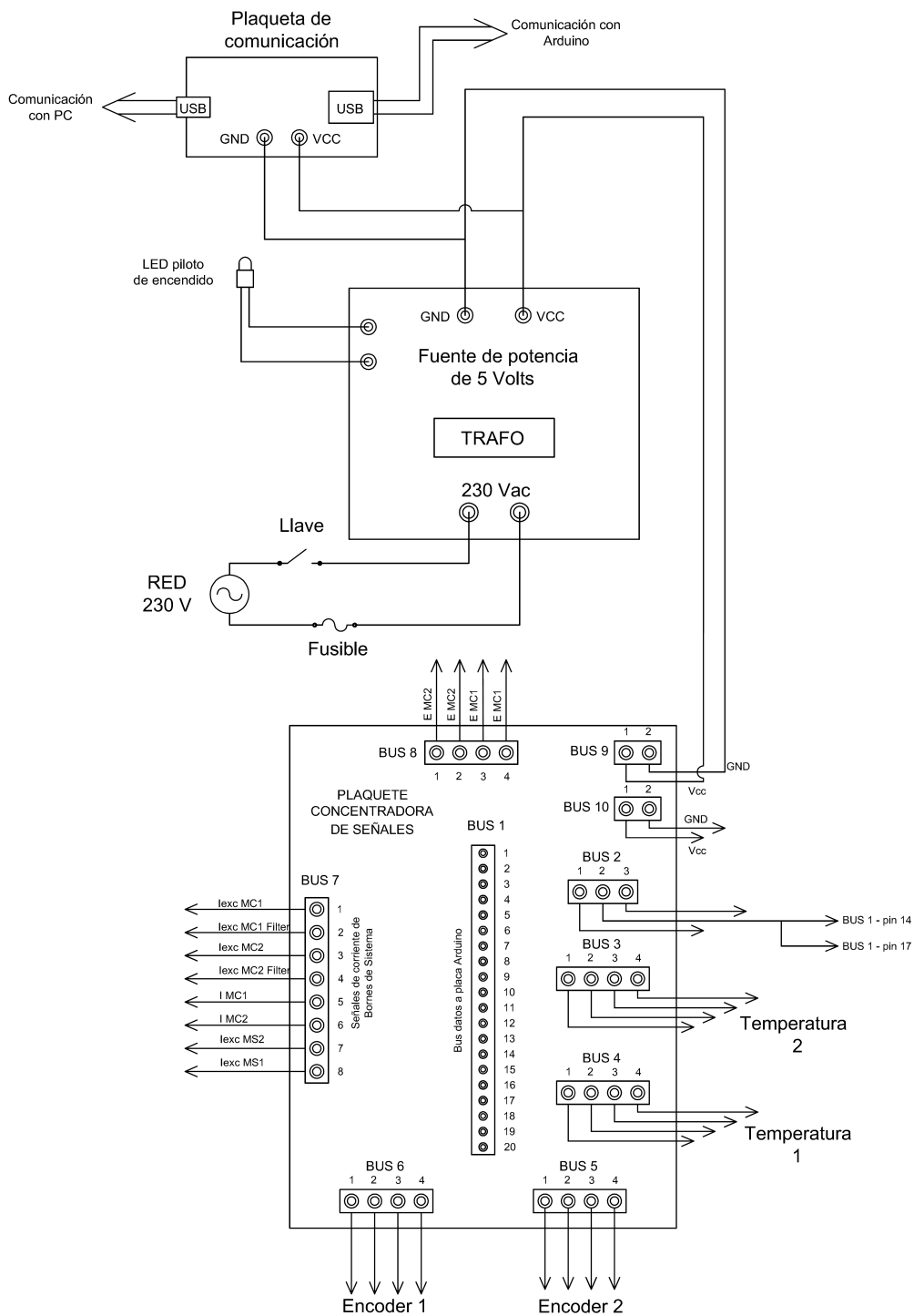


Figura 3.19: Diagrama del Sistema

### 3.2. Arquitectura

figura 3.19), mediante el puerto USB tipo B con que cuenta esta placa. La PC se comunica y recibe los datos a través del software desarrollado para el Sistema.

Esto hace que sea necesario un mapeo de las señales de entradas a cada pin de la placa Arduino y se detalle claramente a qué corresponde cada valor, esto se describe en la sección 3.2.3.

En el esquemático de la figura 3.20 se muestra el conexionado de los bornes correspondientes a las señales DC del frente del equipo.

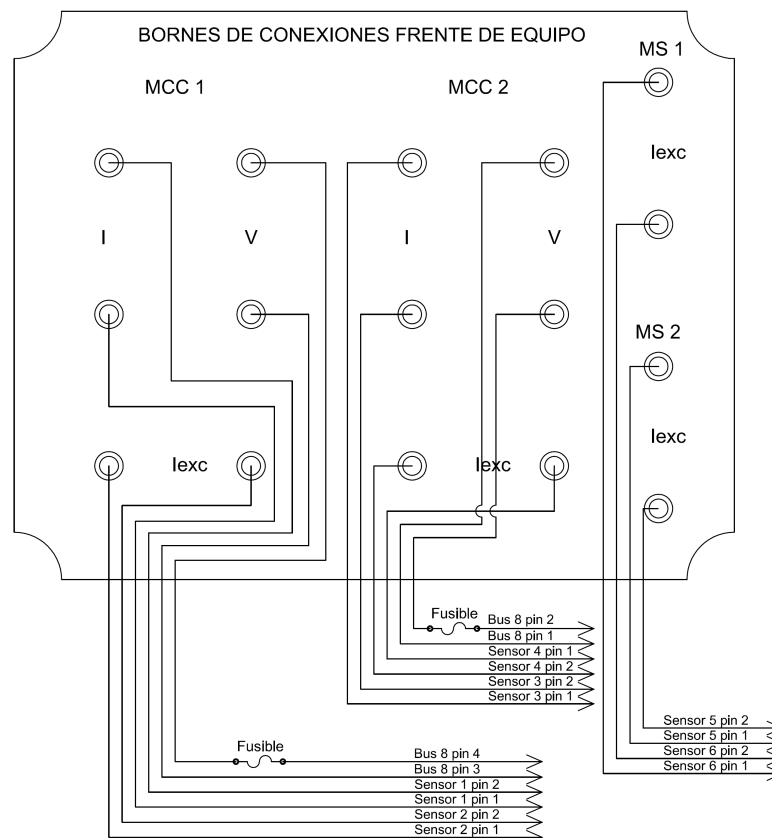


Figura 3.20: Diagrama de los bornes de medidas DC del frente del equipo

Las conexiones de los sensores de corriente se describen en el esquemático de la figura: 3.21.

## Capítulo 3. Hardware

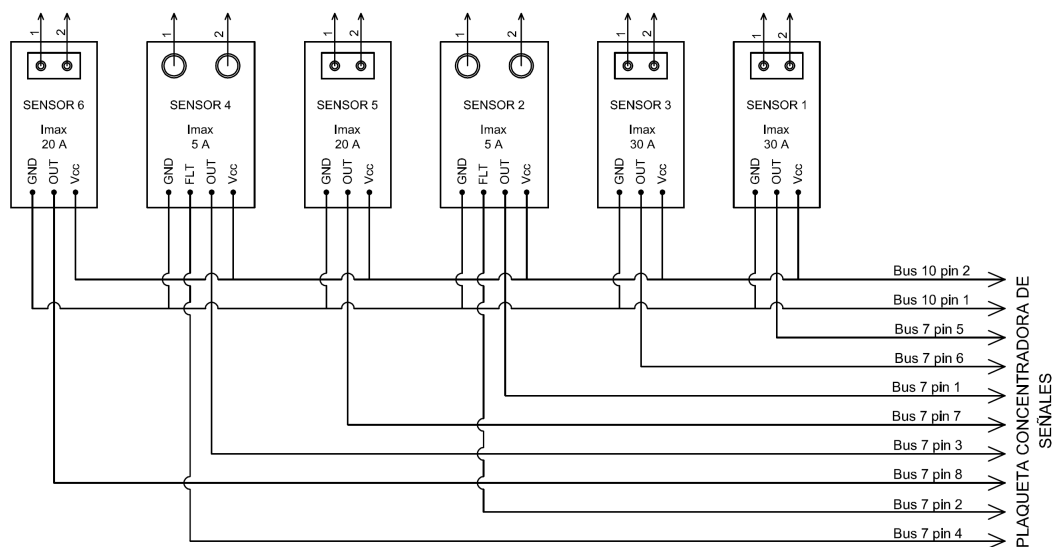


Figura 3.21: Diagrama de sensores de corriente DC

### 3.2.2. Módulos de acondicionamiento de señales

A continuación se describen las señales relevadas y acondicionadas directamente por el equipo desarrollado, es decir las que no son medidas por el analizador de red CVMk2. Las mismas se detallan a continuación con sus respectivos rangos nominales de operación:

1. Corriente excitación MS1 (0...7A)
2. Corriente excitación MS2 (0...7A)
3. Corriente armadura MCC1 (0...21A)
4. Corriente armadura MCC2 (0...21A)
5. Corriente excitación MCC1 (0...1,2A)
6. Corriente excitación MCC2 (0...1,2A)
7. Tensión armadura MCC1 (0...220V)
8. Tensión armadura MCC2 (0...220V)
9. Temperatura 1
10. Velocidad rotación 1
11. Velocidad rotación 2

## 3.2. Arquitectura

Los circuitos y componentes se desarrollaron bajo los siguientes criterios básicos de diseño:

- El diseño debe ser robusto, no dependiendo de polaridades definidas en las entradas (deben poder medir con signo en todo el rango de las variables en cuestión sin saturar).
- Para el diseño de los circuitos de medida de corrientes y tensiones DC se realizó tomando como criterio general valores sobredimensionados en 20 % del régimen permanente en todos los casos (valor de diseño =  $1.2 \cdot \text{valor nominal}$ ).

### Corriente de excitación MS

La medida de estas corrientes DC tanto para la máquina 1 como la 2 se obtienen con sensores de efecto Hall. Aplicando los criterios mencionados de diseño los mismo debieron ser capaces de sensar corrientes que varían en un rango de 0...8,4 A. En tal caso se utilizó el sensor ACS712ELCTR-20A-T (ver sección 3.1.5) y un conductor para el conexionado de  $2 \text{ mm}^2$ .

### Corriente de armadura de MCC

La medida de estas corrientes DC tanto para la máquina 1 como la 2 se obtienen con sensores de efecto Hall como en el caso anterior. Aplicando los criterios mencionados de diseño los mismo debieron ser capaces de sensar corrientes que varían en un rango de 0...25,2 A. En tal caso se utilizó el sensor ACS712ELCTR-30A-T (ver sección 3.1.5) y un conductor para el conexionado de  $3,5 \text{ mm}^2$ .

### Corriente de excitación de MCC

Esta corriente DC de ambas máquinas también se obtienen con sensores de efecto Hall. Aplicando los criterios mencionados de diseño los mismo debieron ser capaces de sensar corrientes que varían en un rango de 0...1,44 A. En tal caso se utilizó el sensores ACS712ELCTR-05B-T (ver sección 3.1.5) y conductores para el conexionado de  $1 \text{ mm}^2$ .

### Tensión de armadura de MCC

Estas tensiones DC tanto para la máquina 1 como la 2 se obtienen acondicionando la señal a un rango apropiado mediante un divisor de tensión resistivo el cual se describe su diseño en la sección 3.5. Los mismos son transductores que entregan una señal de tensión en rangos apropiados para ser sensadas por las entradas de la placa Arduino. Aplicando los criterios mencionados de diseño los mismos son capaces de relevar tensiones que varían en un rango de 0...264 V. En tal caso se utilizó conductores de  $1 \text{ mm}^2$  para el conexionado.

## Capítulo 3. Hardware

### Temperatura de MS

La temperatura se releva con un sensor Infrarrojo de Temperatura MLX90614. En cuanto a los rangos de operación no presenta problema dado que es muy superior al que sera sometido durante la operación del sistema ( $-40^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ ). Para el conexionado se utilizó cable UTP categoría 6. Ver sección 3.1.5.

### 3.2.3. Mapeo de las señales de entradas

En la tabla 3.2 se muestra una descripción detallada del mapeo de las señales de entrada al Arduino junto con la Shield RS485.

En la columna **Variable** se encuentra el nombre de la variable designada en el software que corre en la placa Arduino, en la columna **Dirección de Registro** se encuentra la dirección en hexadecimal del CVMk2 de las variables de interés que mide el mismo. En la columna **Pin en Arduino** se encuentran nombrados los pines correspondientes a la placa Arduino y finalmente en la columna **Periférico** se encuentra el periférico del cual proviene la señal.

Variable	Dirección de Registro	Pin en Arduino	Periférico
U_avg	3E-3F (HEX)	TX2-RX2 (Serial2)	CVMk2
L_avg	40-41 (HEX)	TX2-RX2 (Serial2)	CVMk2
P	42-43 (HEX)	TX2-RX2 (Serial2)	CVMk2
Ql	44-45 (HEX)	TX2-RX2 (Serial2)	CVMk2
Qc	46-47 (HEX)	TX2-RX2 (Serial2)	CVMk2
V1	-	A8	Sensor Opto
V2	-	A9	Sensor Opto
I1	-	A10	Sensor 5 A
I2	-	A11	Sensor 5 A
I3	-	A12	Sensor 30 A
I4	-	A13	Sensor 30 A
I5	-	A14	Sensor 20 A
I6	-	A15	Sensor 20 A
T	Por Defecto	SDA-SCL (SMBus)	MLX90614
Vel1	-	D2-D3	E6B2-CWZ3E
Vel2	-	D18-D19	E6B2-CWZ3E

Tabla 3.2: Tabla de Mapeo de Pines en Arduino-Shield vs. Variables Físicas

### 3.2.4. Adquisición de las señales

Para el caso de las corrientes se tienen 6 sensores de efecto hall lineales (basados en un ACS712) de diferentes fondos de escala y que permiten tener valores de corriente tanto positivos como negativos, mientras que para las tensiones se utiliza un circuito hecho a medida de alta impedancia de entrada utilizando optoacopladores (PS2501) operando en zona activa.

Cada sensor convierte el valor medido a un voltaje entre 0 y 5 Vdc, que luego la placa Arduino convierte mediante un conversor A/D de 10bits a un valor entero, donde 1 equivale a 4.9 mV ( $5\text{ V}/1024$ ).



### 3.3. Diseño de protecciones del sistema

Ésta es una de las razones por la cual es necesaria la existencia de una placa de interfaz (Placa IO) entre el Arduino y el resto del circuito, como se detalló en la sección 3.5.

Los valores de las señales AC llegan a través de un puerto serial mediante protocolo Modbus desde el CVMk2. El analizador de red cuenta con un puerto de comunicación bajo el standard RS-485, por lo que se usa un conversor de norma en formato Shield (ver sección 3.1.1). El puerto utilizado para esta comunicación serial es el número 2 (TX2, RX2) de la placa Arduino.

La temperatura del bobinado de la máquina bajo ensayo es medida desde el sensor infrarrojo MLX90614, el cual se comunica a través de un protocolo llamado SMBus, de 2 conductores los cuales se conectan a los pines SDA y SCL de la placa Shield del Arduino. El sensor además es el único elemento que requiere de una alimentación de 3.3 V por lo que se utiliza la salida de tensión regulada con que cuenta la placa Arduino para energizarse.

Por último, las velocidades angulares de los ejes de los grupos generadores (MS y MCC), son recibidas en pines digitales de la placa Arduino en forma de onda cuadrada dada por un Encoder. Estos pines cuentan con la particularidad de que con ellos se pueden manejar interrupciones al microcontrolador (ATmega2560), por lo cual permite tener un control más ajustado de los tiempos y así poder calcular la velocidad que envía el Encoder.

Una vez adquiridos los diferentes valores en la placa Arduino, dentro de la misma, el firmware acondiciona los valores medidos y realiza el ajuste según la curva relevada del sensor correspondiente. Luego se convierten los valores a string hexadecimales desde enteros (de 32bits) en complemento a 2, y se envían por el puerto serial (USB) hacia el software desarrollado en LabVIEW.

En el capítulo 6 se encuentra la descripción de las curvas de ajuste de los sensores, mientras que en la sección 5.3 se puede ver la descripción del firmware.

### 3.3. Diseño de protecciones del sistema

En esta sección se describen los criterios y metodologías utilizadas para proteger el Sistema ante eventos y situaciones externas al mismo.

#### Protecciones para la alimentación

Para la protección de la fuente se instaló un fusible en la entrada de alimentación del lado de 230 Vac con una corriente nominal de 0,5 A.

Por otra parte del lado DC de la fuente también se instaló un fusible de 0,5 A y un piloto implementado mediante un led color verde instalado en la cara superior del equipo a un lado del pulsador de encendido, con el cual se indica el estado del equipo (ON/OFF).

### Protecciones para las señales de corriente DC

A las señales de corriente que ingresan al equipo no se les instaló protecciones particulares. Se asume que la sobrecarga de los distintos bobinados serán controlados por medios externos al Sistema diseñado. Como medida de seguridad se colocaron advertencias visuales en el frente del equipo indicando en cada borne los rangos de operación.

Por otro lado los sensores tienen una aislación galvánica de 1500 V entre los bornes de entrada del sensor y la electrónica del otro lado del mismo. También el hecho de soportar durante un pulso de 100 ms una corriente de 100 A fue considerado a la hora de no implementar protecciones de sobrecorrientes internas.

### Protecciones para las señales de tensión DC

Las señales de tensión DC se relevan mediante optoacopladores que proporcionan una aislación galvánica de 5 kV entre las señales de entrada y la electrónica aguas abajo del mismo.

Por otro lado se instaló un fusible de 0.5 A como protección contra cortocircuitos que pudieran producirse dentro del gabinete del equipo, muy próximo a uno de los bornes de entrada de la señal.

## 3.4. Descripción exterior del Equipo

El Sistema fue conformado por un único hardware donde se concentran todas las señales a sensar y buses de comunicación.

Este hardware está conformado por una caja contenedora en cuyo interior se encuentran los módulos que lo componen, tales como el de comunicación y el de acondicionamiento de señales. En su frente se encuentran bornes que corresponden a las entradas de las señales a sensar (ver Figura 3.22) agrupados por máquina origen de la señal al cual pertenece.

En la figura:3.23 se muestra la vista lateral izquierda del equipo, donde se pueden ver los siguientes elementos:

- A) Puerto USB Tipo B para comunicación con PC
- B) Puerto de comunicación RS-485 para realizar a comunicación via Modbus con Analizador de red CVMk2
- C) Fusible de protección de entrada de potencia del equipo
- D) Entrada de potencia del equipo de 230 Vac

### 3.4. Descripción exterior del Equipo

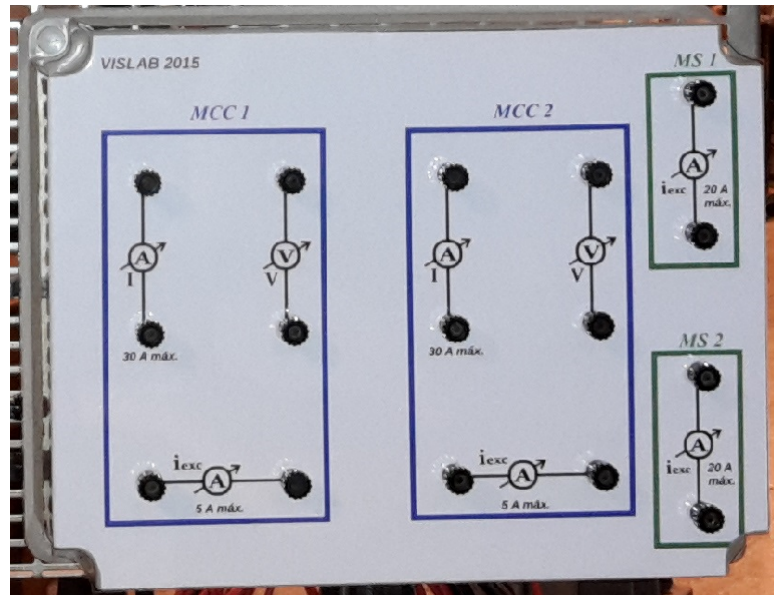


Figura 3.22: Diseño del frente del equipo

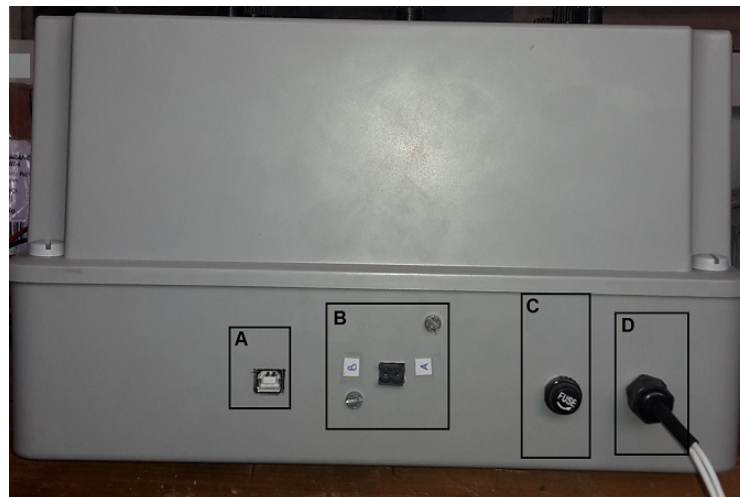


Figura 3.23: Vista lateral izquierda del equipo

Luego en su lateral derecho (ver figura 3.24) se encuentran conectores para el sensor de temperatura como también para los Encoder que relevan al velocidad de rotación de las máquinas.

En la figura:3.25 se muestra la vista lateral superior del equipo, donde se observa el pulsador de encendido y el led piloto indicador de encendido.

## Capítulo 3. Hardware



Figura 3.24: Vista lateral derecha del equipo: A)Puertos de adquisición de señales de Temperatura, B)Puertos de adquisición de señales de Velocidades



Figura 3.25: Vista lateral superior del equipo: A)Pulsador de encendido del equipo, B)Led indicador de estado del Equipo(ON/OFF)

### 3.4.1. Ubicación de los diferentes módulos dentro del Equipo

En la imagen 3.26 se puede observar la parte posterior del interior del Sistema donde se muestran las diferentes partes que lo componen y su ubicación:

### 3.4. Descripción exterior del Equipo

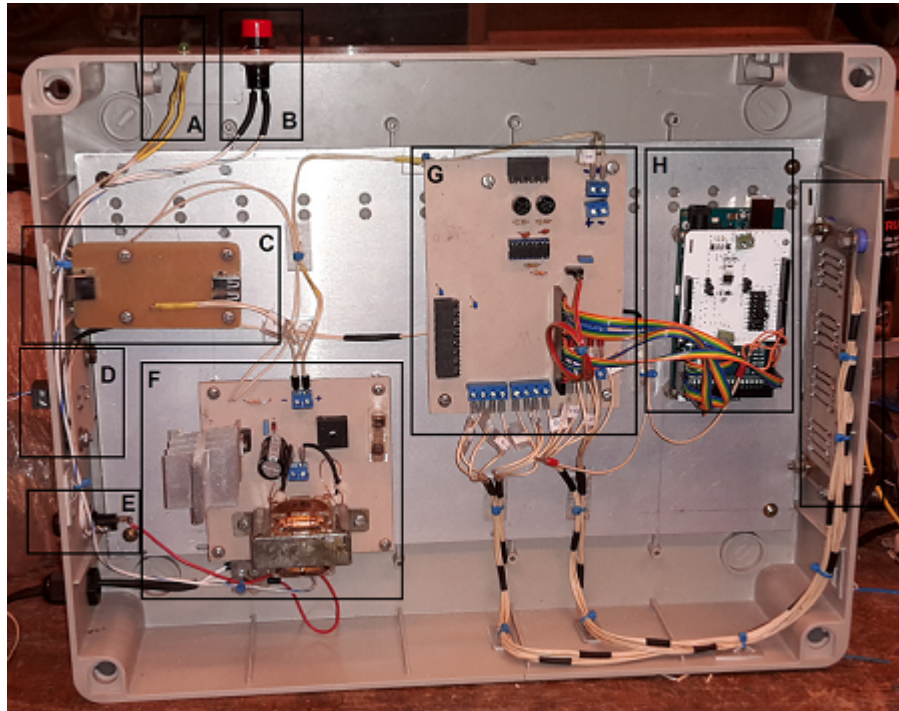


Figura 3.26: Parte de posterior del interior del Equipo

Con referencia en figura:3.26 se detallan a continuación las diferentes partes:

- A) Led piloto indicador de encendido de Sistema
- B) Pulsador de encendido del Sistema
- C) Placa de comunicación PC-Sistema vía puerto USB Tipo B
- D) Placa con bornes de comunicación Sistema-CVMk2 vía RS-485
- E) Fusible de protección alimentación de 230 V
- F) Placa fuente de alimentación 12 V del Sistema
- G) Placa concentradora de señales de medida DC relevadas
- H) Placa Arduino Mega 2560 con Shell de comunicación RS-485
- I) Placa con puertos de adquisición de señales de velocidad y temperatura

### Capítulo 3. Hardware

Por otro lado en la figura 3.27 se puede ver la tapa del equipo del lado interno donde se encuentran los bornes en los cuales se conectan las tensiones y corrientes DC a medir.

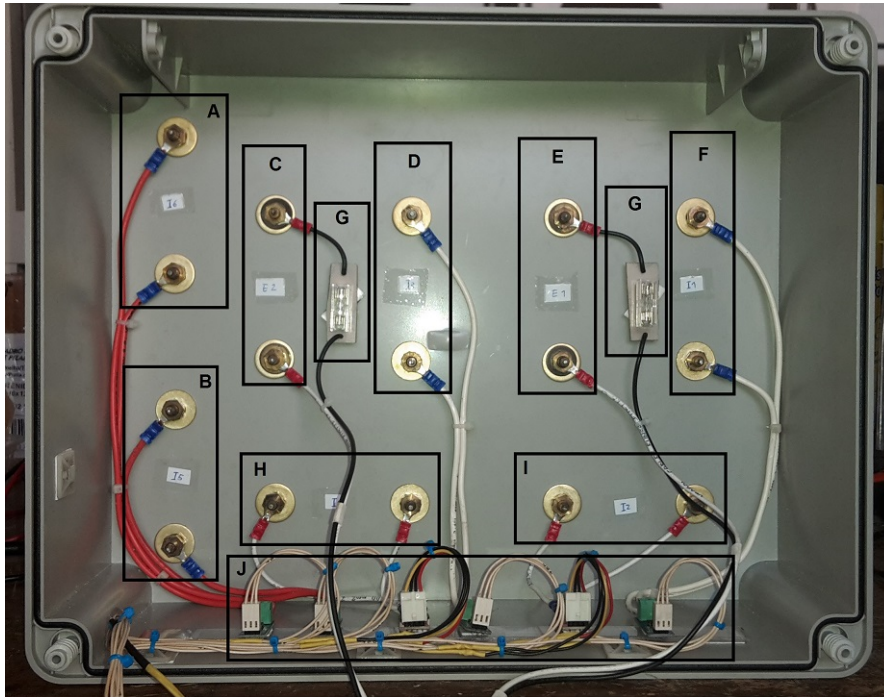


Figura 3.27: Vista de posterior de la tapa del gabinete del equipo

Referencias de la figura:3.27:

- A) Bornes medida de corriente de excitación de MS 1
- B) Bornes medida de corriente de excitación de MS 2
- C) Bornes medida tensión de armadura de máquina de DC 2
- D) Bornes medida de corriente de armadura de máquina de DC 2
- E) Bornes medida tensión de armadura de máquina de DC 1
- F) Bornes medida de corriente de armadura de máquina de DC 1
- G) Fusibles de protección de señales de tensión de máquina DC 1 y 2
- H) Bornes corriente de excitación máquina DC 2
- I) Bornes corriente de excitación máquina DC 1
- J) Sensores de efecto Hall de corriente

### 3.4.2. Mímico del frente del equipo

En el frente del equipo se encuentra un mímico donde están los nombres de los bornes de conexión de las señales de tensiones y corrientes continuas a medir que son relevadas directamente por el equipo. Los bornes están agrupados e identificados por máquina a la cual pertenecen. También se encuentra indicado los rangos de operación de las entradas de corriente.

En la figura 3.28 se puede ver el mismo.

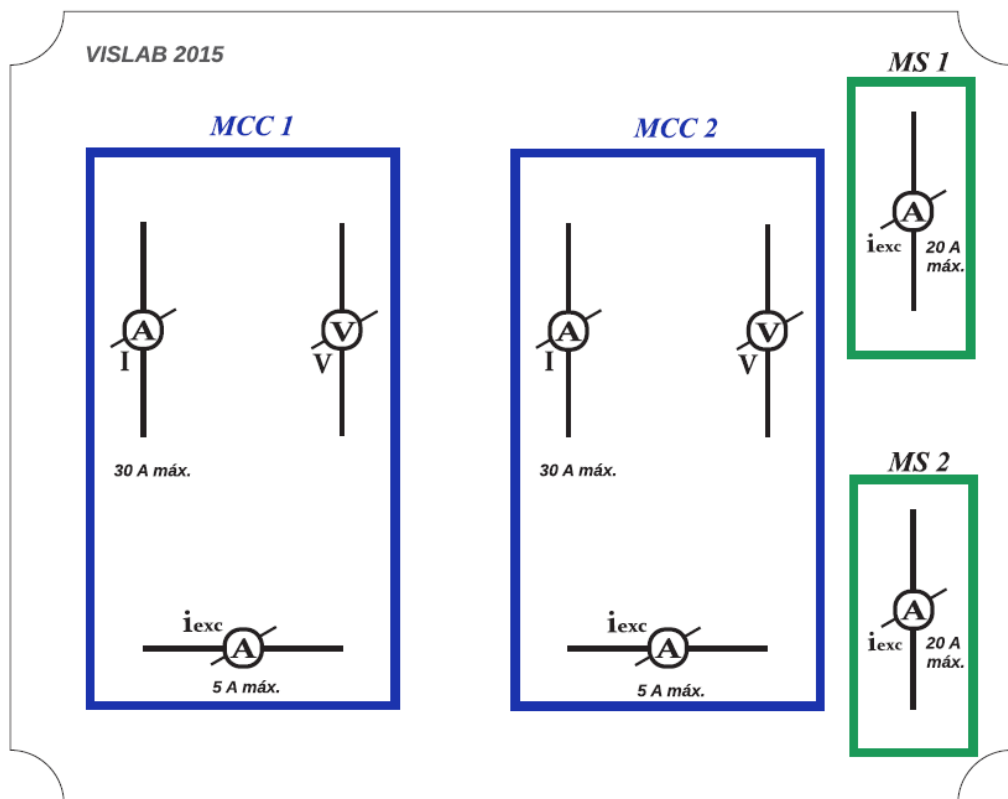


Figura 3.28: Mímico que se encuentra en el frente del equipo

El mismo se imprimió sobre un vinilo que posteriormente se pego en el frente del equipo. Sobre éste se instaló una lámina de acrílico transparente de 2 mm de espesor, creando con esto una barrera de protección mecánica al vinilo. Con esto se aisló de posibles daños que se provocarían durante su uso cotidiano, como ser rayados, suciedad y despegues en el momento de realizar las conexiones en los terminales.

Esta página ha sido intencionalmente dejada en blanco.



# Capítulo 4

## Comunicación

### Introducción

Como se describió en capítulos anteriores, el sistema implementado contiene varios elementos que intercambian datos entre sí mediante protocolos de comunicación.

El sistema consta de 3 tipos de elementos comunicados, primero el Analizador de Red CVMk2 se comunica a través del protocolo Modbus-RTU con la placa Arduino Mega 2560, el sensor de temperatura MLX90614 envía el valor medido por protocolo SMBus a la placa Arduino, y finalmente la PC recibe los valores medidos desde la placa Arduino con un protocolo serial creado específicamente para ese propósito utilizando un puerto USB.

En este capítulo se realizará una descripción de los protocolos mencionados y de los programas realizados para que los equipos puedan comunicarse entre sí.

### 4.1. Desarrollo de comunicación

De acuerdo con la arquitectura del sistema elegido, el microcontrolador Arduino Mega 2560 concentrará los datos de los periféricos/sensores y los enviará a la PC en que corre el software desarrollado en LabVIEW.

Esto se logró en primera instancia, para el caso de las variables trifásicas (tensiones y corrientes de fase, ángulos de fase) conectando a través de un bus RS-485 y comunicándose desde la placa Arduino con el analizador de redes CVMk2, usando el protocolo Modbus. A tal efecto se utilizó una librería para el IDE llamada “SimpleModbus” desarrollada por Juan Bester (ver referencia [9]), la cual construye la palabra de solicitud para los registros del analizador (donde se encuentran guardados los valores de las variables) y las guarda en registros ‘virtuales’ dentro de la placa Arduino.

Por otra parte, dentro del mismo código, se levantan las variables correspondientes a sensores directamente de los pines de entrada analógicos, y se ajustan los valores según sea pertinente. Para el caso particular del sensor de temperatura, se utilizó un bus de comunicación de protocolo SMBus.

## Capítulo 4. Comunicación

Finalmente, la placa Arduino envía los datos adquiridos, a través del puerto USB hacia la PC con un protocolo establecido previamente el cual es descrito en (4.5.1).

El programa de LabVIEW monitorea el puerto USB al cual esta conectado el equipo y actualizará las variables desplegadas en pantalla.

## 4.2. Interfaces de comunicación y Puertos Seriales

### 4.2.1. Puerto Serie

Un puerto serie o puerto serial es una interfaz de comunicación de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.

Tipos de comunicación en serie:

- Simplex: En este caso el emisor y el receptor están perfectamente definidos y la comunicación es unidireccional. Este tipo de comunicaciones se emplean, usualmente, en redes de radiodifusión, donde los receptores no necesitan enviar ningún tipo de dato al transmisor.
- Duplex, half duplex o semi-duplex: En este caso ambos extremos del sistema de comunicación cumplen funciones de transmisor y receptor y los datos se desplazan en ambos sentidos pero no de manera simultánea. Este tipo de comunicación se utiliza habitualmente en la interacción entre terminales y una computadora central.
- Full Duplex: El sistema es similar al duplex, pero los datos se desplazan en ambos sentidos simultáneamente. Para que sea posible ambos emisores poseen diferentes frecuencias de transmisión o dos caminos de comunicación separados, mientras que la comunicación semi-duplex necesita normalmente uno solo. Para el intercambio de datos entre computadores este tipo de comunicaciones son más eficientes que las transmisiones semi-dúplex.

### 4.2.2. RS-485

Está definido como un sistema en bus de transmisión multipunto diferencial, es ideal para transmitir a altas velocidades sobre largas distancias (35 Mbit/s hasta 10 metros y 100 kbit/s en 1200 metros) y a través de canales ruidosos, ya que reduce los ruidos que aparecen en los voltajes producidos en la línea de transmisión. El medio físico de transmisión es un par entrelazado que admite hasta 32 estaciones en 1 solo hilo, con una longitud máxima de 1200 metros operando entre 300 y 19200 bit/s y la comunicación half-duplex (semiduplex). Soporta 32 transmisiones y 32 receptores. La transmisión diferencial permite múltiples drivers dando la posibilidad de una configuración multipunto. Al tratarse de un estándar abierto

## 4.2. Interfaces de comunicación y Puertos Seriales

permite muchas y muy diferentes configuraciones y utilizaciones (ver referencia [14]).

Especificaciones:

- Interfaz diferencial
- Conexión multipunto
- Alimentación única de +5V
- Hasta 32 estaciones (ya existen interfaces que permiten conectar 256 estaciones)
- Velocidad máxima de 10 Mbit/s (a 12 metros)
- Longitud máxima de alcance de 1200 metros (a 100 kbit/s)
- Rango de bus de -7V a +12V

### 4.2.3. RS-232

RS-232 (Recommended Standard 232), en español: “Estándar Recomendado 232”, también conocido como EIA/TIA RS-232C, es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre un DTE (Data Terminal Equipment, “Equipo Terminal de Datos”) y un DCE (Data Communication Equipment, Equipo de Comunicación de Datos).

El RS-232 consiste en un conector tipo DB-25 (de 25 pines), aunque es normal encontrar la versión de 9 pines (DE-9, o popularmente mal denominados DB-9), más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie de la PC) (ver referencia [13]).

### 4.2.4. Puerto USB

El Bus Serial Universal o USB (por sus siglas en inglés Universal Serial Bus), es un estándar industrial desarrollado en los años 90, que define los conectores, protocolos de comunicación, y suministro de energía entre computadores y dispositivos electrónicos.

Fue diseñado para estandarizar la conexión de periféricos con PC, tanto para su intercomunicación como para suministro de alimentación en 5V de continua. Es de uso común, y utilizado por la gran mayoría de los dispositivos electrónicos modernos como: smartphones, consolas de videojuegos, cargadores de batería, mouse, teclados, etc.

El cable y conectores constan de cuatro hilos, un par trenzado de comunicación (Data+ y Data-) y dos de alimentación en 5V (+ y -).

### 4.2.5. Bus SMBus

El bus de manejo de sistema o SMBus (por sus siglas en inglés System Management Bus) es un protocolo simple de dos hilos de comunicación para aplicaciones de poco ancho de banda. Es más comúnmente utilizado para comunicar las placas madres de PC con su fuente de alimentación con instrucciones de tipo ON/OFF. Otros dispositivos que lo utilizan incluyen sensores de temperatura o voltaje, chips de reloj y fans (ventiladores de enfriamiento para PCs).

El SMBus fue definido por Intel en 1995. Envía datos, reloj e instrucciones y está basado en el protocolo de bus serial I2C de Philips. Su frecuencia de reloj es de 10 a 100kHz, y sus niveles de voltaje están definidos más estrictamente que los del I2C (ver referencia [15]).

## 4.3. Protocolos de Comunicación

### 4.3.1. Modbus

Es un protocolo de comunicaciones situado en el nivel 7 del Modelo OSI (Modelo de Interconexión de Sistemas Abiertos (ISO/IEC 7498-1)), basado en la arquitectura maestro/esclavo o cliente/servidor, diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLCs). Convertido en un protocolo de comunicaciones estándar de facto en la industria, es el que goza de mayor disponibilidad para la conexión de dispositivos electrónicos industriales. Las razones por las cuales el uso de Modbus es superior a otros protocolos son:

- Es público
- Su implementación es fácil y requiere poco desarrollo
- Maneja bloques de datos sin suponer restricciones

El mismo permite el control de una red de dispositivos, por ejemplo un sistema de medida de temperatura y humedad, y comunicar los resultados a un ordenador. Modbus también se usa para la conexión de un ordenador de supervisión con una unidad remota (RTU) en sistemas de supervisión y adquisición de datos (SCADA).

Existen dos variantes, con diferentes representaciones numéricas de los datos y detalles del protocolo ligeramente desiguales. Modbus/RTU es una representación binaria compacta de los datos. Modbus ASCII es una representación legible del protocolo pero menos eficiente. Ambas implementaciones son serie. El formato RTU finaliza la trama con un suma de control de redundancia cíclica (CRC), mientras que el formato ASCII utiliza una suma de control de redundancia longitudinal (LRC). La versión Modbus/TCP es muy semejante al formato RTU, pero estableciendo la transmisión mediante paquetes TCP/IP.

Para las comunicaciones internas de los equipos, o entre ellos, las RTU han adoptado el protocolo MODBUS, en la forma de MODBUS/RTU, que puede implementarse sobre una red RS-485 o sobre una red TCP/IP.

### 4.3. Protocolos de Comunicación

Cada dispositivo de la red Modbus posee una dirección única. Cualquier dispositivo puede enviar órdenes Modbus, aunque lo habitual es permitirlo sólo a un dispositivo maestro. Cada comando Modbus contiene la dirección del dispositivo destinatario de la orden. Todos los dispositivos reciben la trama pero sólo el destinatario la ejecuta (salvo un modo especial denominado "Broadcast"). Cada uno de los mensajes incluye información redundante que asegura su integridad en la recepción. Los comandos básicos Modbus permiten controlar un dispositivo RTU para modificar el valor de alguno de sus registros o bien solicitar el contenido de dichos registros (ver referencias: [10], [2]).

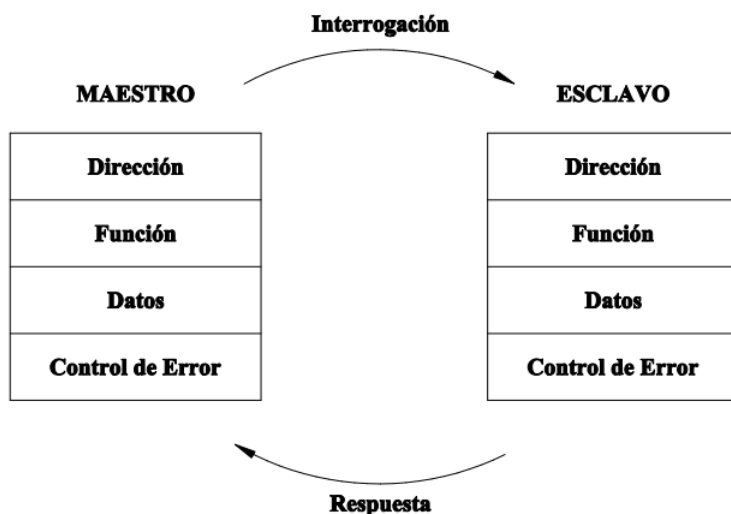


Figura 4.1: Formato general de las tramas

En la figura 4.1 se muestra como son armadas las tramas para la comunicación Modbus. En la figura 4.2 se muestra el formato para Modo ASCII y Modo RTU (las dos variantes Modbus).

Modo ASCII					
Comienzo de Trama	Dirección	Función	Datos	Control de Errores	Fin de Trama
:	2 bytes	2 bytes	N x 2 bytes	2 bytes	CR + LF

Modo RTU					
Comienzo de Trama	Dirección	Función	Datos	Control de Errores	Fin de Trama
Tiempo de 3 bytes	1 bytes	1 bytes	N x 1 bytes	2 bytes	

Figura 4.2: Formato trama MODBUS (ASCII y RTU)

## Capítulo 4. Comunicación

Descripción de los componentes de las tramas:

- **Dirección:** El maestro inicia la comunicación enviando un byte con la dirección del esclavo para el cual se destina el mensaje. Al enviar la respuesta, el esclavo también inicia el telegrama con su propia dirección, posibilitando que el maestro conozca cuál esclavo está enviándole la respuesta. El maestro también puede enviar un mensaje destinado a la dirección “0” (cero), lo que significa que el mensaje es destinado a todos los esclavos de la red (broadcast). En este caso, ningún esclavo irá a contestar al maestro.
- **Código de la función:** Este campo también contiene un único byte, donde el maestro especifica el tipo de servicio o función solicitada al esclavo (lectura, escritura, etc.). De acuerdo con el protocolo, cada función es utilizada para acceder un tipo específico de dato.
- **Campo de datos:** Campo con tamaño variable. El formato y el contenido de este campo dependen de la función utilizada y de los valores transmitidos. Este campo está descrito junto con la descripción de las funciones.
- **CRC:** La última parte del telegrama es el campo para el error del chequeo de transmisión. El método utilizado es el CRC-16 (Cycling Redundancy Check).

Descripción de código CRC:

Se utiliza un sistema de detección de errores diferente dependiendo del tipo de codificación utilizado ( ASCII o RTU ). En el caso de la codificación ASCII es el checksum ( o Longitud Redundancy Check LRC ) en módulo 16 expresado en ASCII ( 2 caracteres representan 1 byte ), sin considerar el ”:” ni el “CR LF” de la trama. En la codificación RTU se utiliza el método de CRC ( Cyclical Redundancy Check ) codificado en 2 bytes (16 bits) (ver referencia [16]).

### 4.3.2. Protocolo SMBus

El protocolo utilizado por el SMBus es del tipo multi-master/multi-slave y extremadamente similar al utilizado para I2C. Por esto se utilizará una librería de Arduino para I2C, para leer temperatura a través del sensor MLX. Dicha librería fue creada por Peter Fleury (ver referencia [15]).

## 4.4. Pruebas de Comunicación Preliminares

### 4.4.1. Pruebas de comunicación CVMk2 con PC

Se realizaron múltiples ensayos y pruebas para lograr ésta comunicación, en las siguientes secciones se describirán las mismas. Éstas se realizaron utilizando la aplicación MODBUS POLL.

## 4.4. Pruebas de Comunicación Preliminares

### Software MODBUS POLL

Es un simulador de Modbus diseñado principalmente para ayudar a los desarrolladores de dispositivos Modbus esclavos u otras personas que quieran probar y simular el protocolo Modbus (ver referencia [11]). Con las interfaces de documentos múltiples, se puede monitorizar varios esclavos Modbus y / o áreas de datos al mismo tiempo. Para cada ventana sólo tiene que especificar el ID de esclavo Modbus, función, dirección, tamaño y tipo de encuesta. Puede leer y escribir registros y bobinas (registros que contienen varias variables del tipo ON/OFF) desde cualquier ventana de manera individual o de a varios elementos. Múltiples formatos de datos tales como punto flotante, doble precisión y long (entero de 32bits) están disponibles. En Modbus Poll los errores de excepción se muestran en la línea de estado.

Variantes del protocolo soportados:

- Modbus RTU
- Modbus ASCII
- Modbus TCP / IP
- Modbus RTU sobre TCP / IP
- Modbus ASCII sobre TCP / IP
- Modbus UDP / IP
- Modbus RTU a través de UDP / IP
- Modbus ASCII a través de UDP / IP

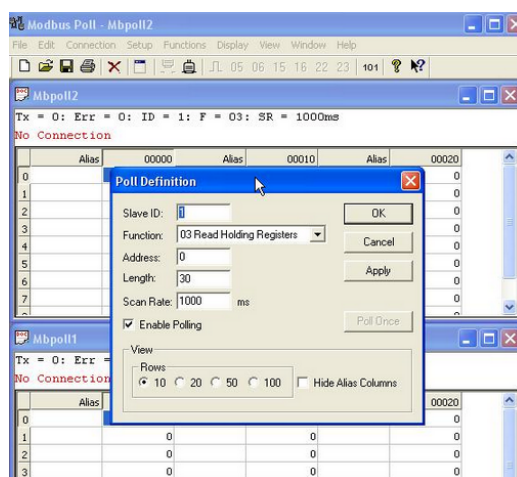


Figura 4.3: Modbus Poll

## Capítulo 4. Comunicación

### Comunicación con MODBUS POLL

Se probó la comunicación utilizando el software MODBUS POLL. Para esto se utilizó un conversor RS-485/RS-232 y otro RS-232/USB. Luego de realizadas varias pruebas se llegó a un conversor que funcionara correctamente y con esto se logró realizar consultas al equipo CVMk2 y obtener los valores guardados en los lugares de memoria correspondientes a las señales medidas.

Luego de estudiar el protocolo MODBUS y el software se configuró el equipo CVMk2 como esclavo de la siguiente manera:

- Núm. Perif 001
- Baudios 9600
- Paridad NO
- Bit de datos 8
- Bit stop 1
- Protocolo MODBUS

De forma similar se configuró el MODBUS POLL:

- Baudios 9600
- Paridad NO
- Bit de datos 8
- Bit stop 1

En la figura 4.4 se observan los parámetros de configuración para la comunicación, siendo estos: ID de esclavo, función a ejecutar, direcciones de memoria a leer, etc.

Corriendo el software con esta configuración se logró la comunicación correctamente.

Los datos recibidos en la PC, MODBUS POLL los presenta como se muestra en figura 4.5.



#### 4.4. Pruebas de Comunicación Preliminares

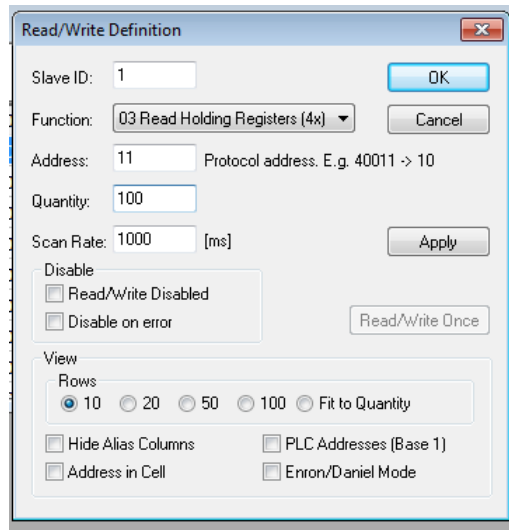


Figura 4.4: Configuración de direcciones en MODBUS POLL

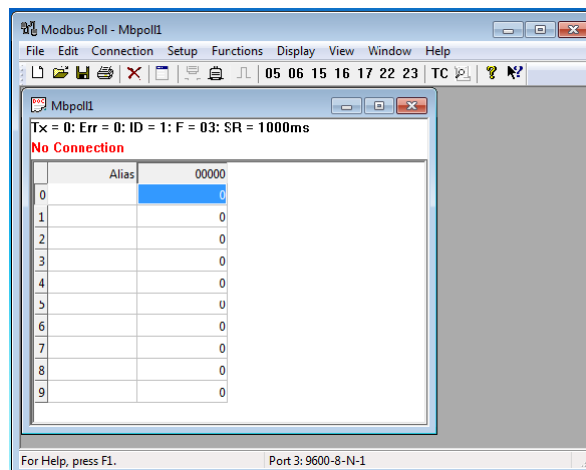


Figura 4.5: Datos presentados por MODBUS POLL

#### 4.4.2. Pruebas de comunicación Arduino-LabVIEW

Dado que el sistema analizado tiene una placa Arduino que se deberán comunicar con la PC, se estudió la interacción de estos dos componentes. Las pruebas que aquí se detallan fueron realizadas con un Arduino UNO, pero dado que todas las placas trabajan con el mismo IDE, se consideran equivalentes a realizarlas con un Arduino Mega 2560.

La comunicación Arduino se llevo adelante mediante un módulo que posee LabVIEW para desarrollar software específico para estas placas denominado LYNX. Usando esta herramienta se desarrolla la rutina mostrada en figura 4.6 a modo de prueba.

Utilizando la placa Arduino, se armó el circuito mostrado en figura 2.3 conec-

## Capítulo 4. Comunicación

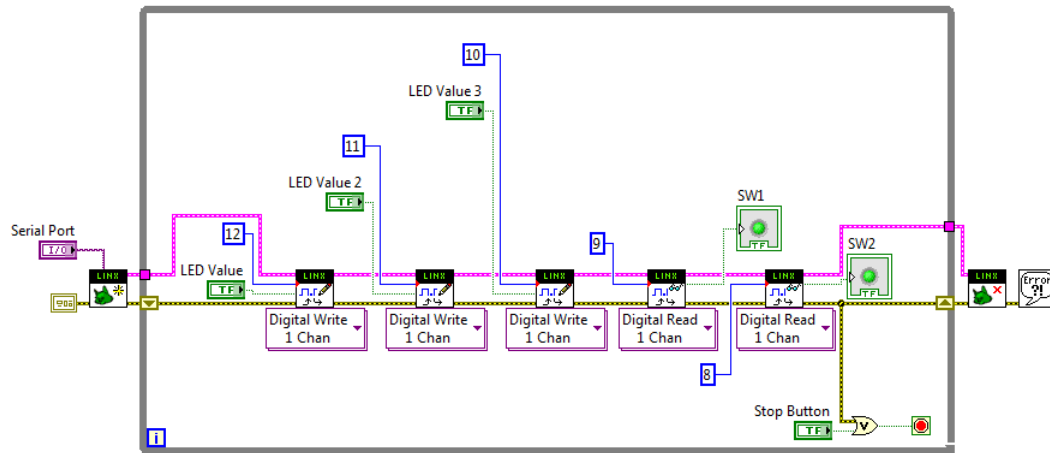


Figura 4.6: Diagrama de bloques del Software de Comunicación de testeo para Arduino

tado en los puertos de entrada/salida de la placa Arduino. Con esto se pudo lograr una comunicación bidireccional.

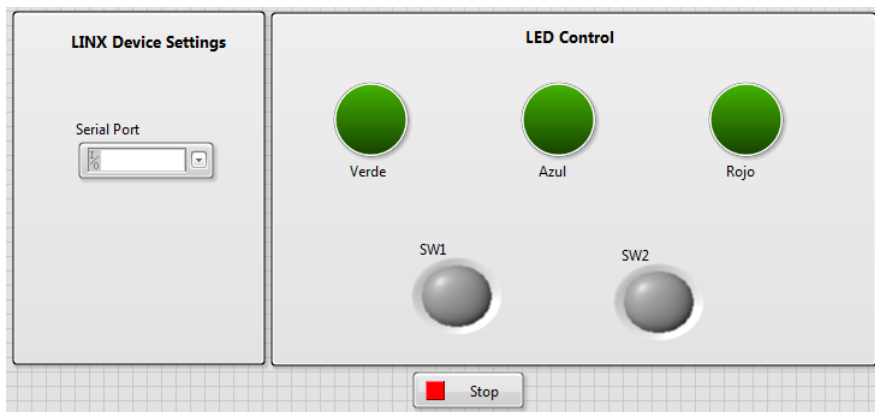


Figura 4.7: Panel frontal del Software de Comunicación de testeo para Arduino

## 4.5. Comunicación Arduino-LabVIEW Implementada

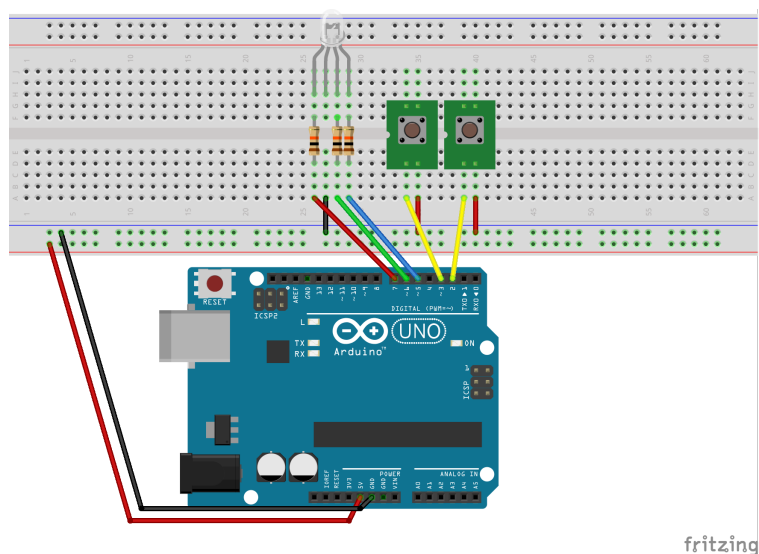


Figura 4.8: Esquemático del circuito de testeo de Comunicación con placa Arduino

Estas pruebas y experiencia obtenidas (tanto la de la comunicación CVMk2-PC como Arduino-LabVIEW) permitieron diseñar una buena implementación de la comunicación de la placa tanto, con el equipo CVMK2, como con el LabVIEW.

## 4.5. Comunicación Arduino-LabVIEW Implementada

### 4.5.1. Protocolo Creado

Como se estableció en capítulos anteriores, la comunicación entre la Placa Arduino Mega 2560 y la PC con el software desarrollado basado en LabVIEW será a través de puerto USB. Debido a que los datos serán enviados en tiempo real a la PC (por lo que el retardo existente entre la visualización y la realidad será el tiempo de ejecución del firmware de Arduino), es necesario establecer un protocolo a utilizar que minimice los errores introducidos por ruido o desincronización en la lectura de datos.

Para lograr esto, se estableció que la Placa Arduino envíe medidas de 8 bytes en codificadas como caracteres ASCII en Hexadecimal. Según el valor medido, los valores podrán ser U32 (unsigned integer 32 bits o entero sin signo de 32 bits) o I32 (también conocido como LONG o entero de 32bits en complemento a 2).

Cada valor medido estará precedido por dos bytes de dígito identificador (cada dígito estará asociado a una variable física real que recibe la placa Arduino) y antecedido por un carácter de fin de línea, como se detalla en figura 4.9.

La placa Arduino envía paquetes de datos de la manera antes mencionada, en orden secuencial y continuo por lo que se encuentra enviando los datos en un loop continuo y retransmitiendo cada valor una vez por ciclo. El dígito identificador y las variables asociadas a dicho numero se encuentran en tabla 4.1.

## Capítulo 4. Comunicación

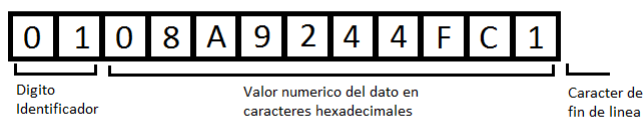


Figura 4.9: Paquete enviado de manera serial por placa Arduino a PC

Num. Idetificador en Comm Serial	Variable
00	U_avg
01	I_avg
02	P
03	Ql
04	Qc
05	V1
06	V2
07	I1
08	I2
09	I3
10	I4
11	I5
12	I6
13	T
15	Vel1
16	Vel2

Tabla 4.1: Tabla de Dígitos Identificadores

### 4.5.2. Driver de Comunicación Arduino-LabVIEW

La sección de programa en LabVIEW que se encarga del manejo de los datos de comunicación recibidos desde la placa Arduino se le denomina Driver Arduino.

Como se puede deducir de la sección anterior, es necesario que una parte del código principal del software esté monitoreando el puerto USB en intervalos regulares, de forma tal que se puedan actualizar los valores medidos en campo, hacia la pantalla donde se despliegan los mismos.

En la figura 4.10 se puede apreciar el backpanel (panel en donde se programa la parte lógica del LabVIEW) correspondiente al driver, que se explica detalladamente en capítulo 5 sección A.1.1.

El driver se encargará de, recibir los valores desde el puerto USB y procesar los datos recibidos ya que estos pueden venir de forma errónea (errores de comunicación).

En la tabla 3.2 se puede ver un resumen del mapeo de todas las variables contra

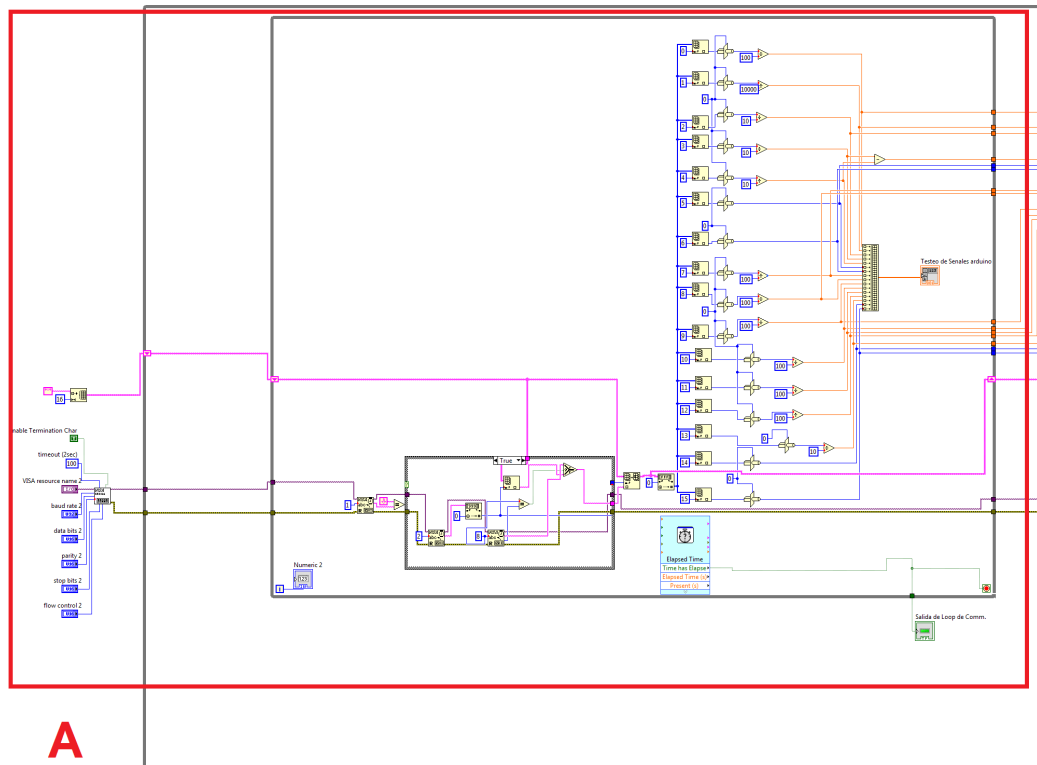


Figura 4.10: Backpanel de Driver Arduino en LabVIEW

las magnitudes y unidades correspondientes que el driver en LabVIEW recibe y debe entregar al resto del programa.

## 4.6. Resumen

En el transcurso de este capítulo se describieron las líneas de comunicación necesarias para que el programa desarrollado en LabVIEW pueda adquirir todas las medidas requeridas. Cabe destacar que la placa Arduino concentra en ella todas las comunicaciones, ya sea para recolectar medidas de otros dispositivos como para enviarlas a la PC.

El hecho que la placa Arduino sea el concentrador de comunicaciones necesariamente implica que el tiempo entre que efectivamente se realizan las medidas y que estas lleguen a desplegarse, estará dado por el tiempo de ejecución de código firmware en la placa. Resulta importante entonces el minimizar este tiempo de forma que las medidas desplegadas tengan un retraso despreciable con respecto a la velocidad de cambio de las variables relevadas.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 5

## Software

### Introducción

En este capítulo se abordará la implementación del software desarrollado que conforma el Sistema, siendo éste parte fundamental del mismo. Con éste se adquirirán, procesarán y desplegarán los datos de forma adecuada al usuario.

Toda la implementación se realizó utilizando LabVIEW como herramienta de programación debido a sus altas prestaciones como desarrollador de instrumentos virtuales.

Se presentará una introducción de la plataforma de desarrollo y luego se pasará a detallar las diferentes partes que componen el Software.

Como resultado se obtuvo una aplicación capaz de comunicarse con los periféricos que conforman el sistema, al mismo tiempo que procesa, despliega y guarda los datos adquiridos. A su vez es capaz de ser instalado en una PC de recursos moderados.

## 5.1. Plataforma de desarrollo LabVIEW

### 5.1.1. Introducción

LabVIEW (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench) es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido. El lenguaje que usa se llama lenguaje G, donde la G simboliza el hecho que es gráfico.

Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, MAC y GNU/Linux.

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica (Instrumentación electrónica) sino también a su programación embebida, comunicaciones,

## Capítulo 5. Software

matemáticas, etc. Un lema tradicional de LabVIEW es: "La potencia está en el Software", que con la aparición de los sistemas multinúcleo se ha hecho aún más potente. Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante (tarjetas de adquisición de datos, PAC, Visión, instrumentos y otro Hardware) como de otros fabricantes (ver referencias: [7], [8], [5]).

### 5.1.2. Principales características

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación pueden hacer programas relativamente complejos. También es muy rápido hacer programas con LabVIEW y cualquier programador, por experimentado que sea, puede beneficiarse de él. Los programas en LabVIEW son llamados instrumentos virtuales (VIs). Con LabVIEW se pueden crear programas de miles de VIs (equivalente a millones de páginas de código texto) para aplicaciones complejas como pueden ser programas de automatizaciones de decenas de miles de puntos de entradas/salidas, proyectos para combinar nuevos VIs con VIs ya creados, entre otros. Incluso existen buenas prácticas de programación para optimizar el rendimiento y la calidad de la programación. El LabVIEW 7.0 introduce un nuevo tipo de subVI llamado VIs Expreso (Express VIs). Estos son VIs interactivos que tienen una configuración de caja de diálogo que permite al usuario personalizar la funcionalidad del VI Expreso. El VIs estándar son VIs modulares y personalizables mediante cableado y funciones que son elementos fundamentales de operación de LabVIEW.

### 5.1.3. Programación en LabVIEW

#### Entorno de Desarrollo

Como se mencionó, la programación se realiza utilizando VI, estos constan de dos partes diferenciadas:

- Panel Frontal: El Panel Frontal es la interfaz con el usuario, la utilizamos para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real (como van fluyendo los datos). En esta interfaz se definen los controles (los usamos como entradas) e indicadores (los usamos como salidas).
- Panel Diagrama de Bloques: es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan íconos que realizan una determinada función y se interconectan (el código que controla el programa). Suele haber una tercera parte icono/conector que son los medios utilizados para conectar un VI con otros VIs.



## 5.1. Plataforma de desarrollo LabVIEW

Un panel frontal está formado por botones, pulsadores, gráficos, reguladores de magnitud, etc, como se observa en la figura 5.1. Cada uno de ellos puede estar definido como un control o un indicador. Los controles sirven para introducir parámetros al VI, mientras que los indicadores muestran los resultados producidos, ya sean datos adquiridos o resultados de alguna operación. En este lenguaje se refiere a las entradas del programa , como controles y a las salidas como indicadores. En el panel frontal el usuario puede modificar las entradas, y monitorear los datos de salida.



Figura 5.1: Panel Frontal

El diagrama de bloque contiene el código fuente en formato gráfico. Los objetos del panel frontal aparecen como terminales en el diagrama de bloques. Un terminal es aquel elemento del diagrama de bloques que representan una entrada o salida del programa. Adicionalmente, este diagrama contiene funciones y estructuras incorporadas en las bibliotecas de LabVIEW como se observa en la Figura 5.2. Los cables son los elementos que en este entorno de desarrollo cumplen la función de conectar cada uno de los nodos en el diagrama de bloques, incluyendo controles e indicadores, funciones y estructuras. Un nodo es un elemento del VI que ejecuta un código en su interior. Cuando un VI se ejecuta, los valores ingresados por el usuario en los controles fluyen a través de los cables en el diagrama de bloques y son usados en las distintas funciones del diagrama, los resultados son recogidos por otras funciones o indicadores.

## Capítulo 5. Software

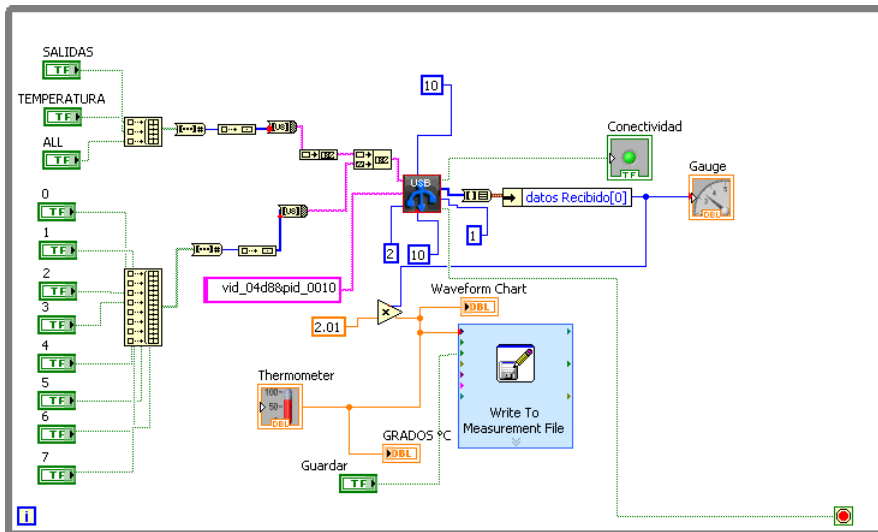


Figura 5.2: Diagrama de Bloques

## Comandos

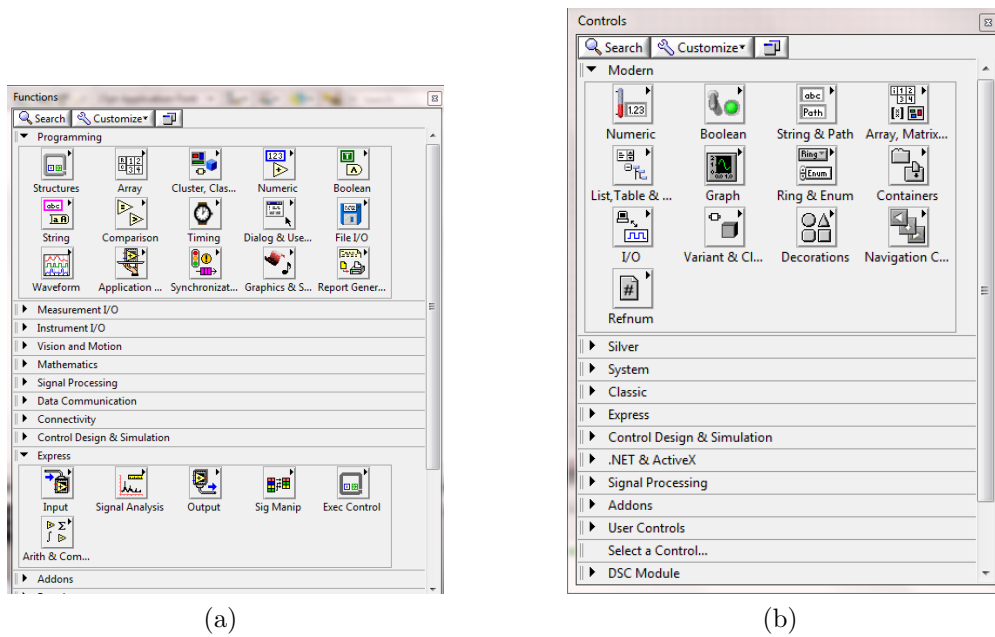


Figura 5.3: 5.3a: Paleta de Funciones 5.3b: Paleta de Control

La paleta de Funciones se despliega en el Panel de Bloques (ver Figura 5.3a) se utiliza en el diseño del diagrama de bloques. Contiene los objetos que se emplean en la implementación del VI, ya sean funciones aritméticas, de entrada/salida de

## 5.1. Plataforma de desarrollo LabVIEW

señales, entrada/salida de datos a fichero, adquisición de señales, temporización de la ejecución del programa.

La paleta de Control se despliega en el Panel Frontal (ver Figura 5.3b) se utiliza únicamente en éste, contiene todos los controles e indicadores que se pueden utilizar.

### Estructuras

Las estructuras controlan el flujo del programa mediante la secuenciación de acciones, ejecución de bucles, etc. Se comportan como cualquier nodo en el diagrama de bloques, es decir, ejecutan lo que está programado en su interior inmediatamente después de que tienen disponibles los datos de entrada. Una vez ejecutado el código en su interior suministran los correspondientes valores a los cables unidos a sus salidas. Sin embargo, cada estructura ejecuta su sub-diagrama de acuerdo con las reglas específicas que rigen su comportamiento.

Un sub-diagrama es una colección de nodos, cables, funciones y terminales situados en el interior del rectángulo que constituye la estructura. Tanto el loop FOR como el loop WHILE poseen un solo sub-diagrama, pero el CASE y el SEQUENCE pueden tener múltiples sub-diagramas pero únicamente será posible ejecutarlos de a uno. En el caso del CASE se ejecutará aquel que cumpla con la condición de entrada. Por otro lado la estructura SEQUENCE está diseñada para marcar un orden en la ejecución ya que LabVIEW no sigue un orden de ejecución sino que, ejecuta el código de un nodo inmediatamente después que tiene disponibles todas las entradas. Esto implica que puede ejecutar el código de varios nodos en paralelo. Tanto la estructura FOR, como la WHILE y el CASE cumplen las mismas funciones que en otros lenguajes.

LabVIEW maneja la jerarquía de los VI's mediante el uso de Sub-VIs. Estos son aquellos VIs que son utilizados como parte de otro VI. Es similar al concepto de una subrutina en otros lenguajes.

### Modulo VISA

El Software de Arquitectura de Instrumento Virtual (VISA por su sigla en inglés) es un estándar para configurar, programar y solucionar problemas de sistemas de instrumentación que incluye las interfaces GPIB, Serial, Ethernet, USB, entre otras. VISA brinda una interfaz de programación entre el hardware y entornos de programación, como por ejemplo LabVIEW. La paleta de LabVIEW de VISA mostrada en la Figura 8, permite realizar la comunicación con diferentes instrumentos. En la misma se encuentran VI's capaces de abrir y cerrar una comunicación con el instrumento, leer datos del mismo, y escribirle comandos que permitan configurarlo y darle órdenes. Un ejemplo de comunicación con un instrumento es mostrado en la Figura 9, donde en primer lugar se abre la comunicación con el instrumento mediante el sub VI OPEN, luego se lee un dato del mismo mediante el sub VI READ y luego se cierra la comunicación mediante CLOSE.

## Capítulo 5. Software

Cabe destacar que no siempre es necesario recurrir a los VI's de la paleta VISA para poder comunicarse con un instrumento, muchos fabricantes de éstos y usuarios de LabVIEW crean drivers para manejar los instrumentos más fácilmente. Estos drivers se encuentran en la página de National Instruments

### 5.2. Software desarrollado en LabVIEW

#### 5.2.1. Desarrollo entorno gráfico

En esta sección se describirá la composición y el desarrollo del entorno gráfico del Sistema. El mismo se compone de pestañas o solapas donde cada una de ellas corresponde a un determinado Ensayo o tarea a realizar. Es decir en cada solapa se encuentra lo necesario para realizar una determinada actividad como puede ser realizar Ensayo de Corto Circuito a una máquina sincrónica, entre otros.

Al ejecutar el programa el mismo se carga iniciando en la pantalla Principal que corresponde a la solapa de Inicio del mismo. Desde la misma se ve el conjunto de 'pestañas' correspondiente a las actividad que se pueden realizar, seleccionado la que se desee se navega hacia la solapa correspondiente.

En la Figura 5.4se puede ver el aspecto de la misma y la ubicación de los elementos.



Figura 5.4: Vista de solapa principal de Software

Para implementar esta estructura de programa se utilizó el bloque 'Tab Control' el cual cuenta LabVIEW. En la Figura 5.5a se muestra la forma de acceder a él en el diagrama de bloques del sistema. En la Figura 5.5b se muestra el icono que representa el bloque descrito.

## 5.2. Software desarrollado en LabVIEW

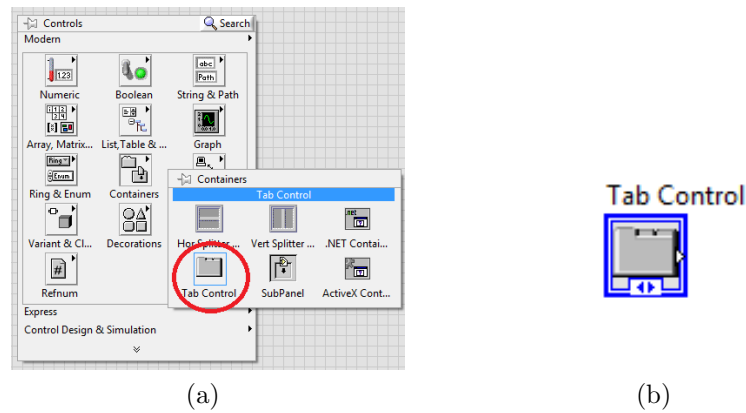


Figura 5.5: 5.5a: Ubicación de Tab Control 5.5b: Icono de Tab Control

Luego de insertar el bloque en el Diagrama de Bloques del entorno de desarrollo, en el Panel Frontal se configura agregando las solapas necesarias y asignándole los nombres que correspondan a cada una en función de la actividad que se realiza en ella. Luego utilizando la Paleta de Herramientas se configura el aspecto de la misma como puede ser efectos, color, dimensiones, entre otros.

Por otro lado se realizaron las imágenes correspondientes a los esquemáticos de las solapas de acuerdo a las necesidades de cada una, las cuales cumplen la función de fondo contextual en las diferentes magnitudes que son medidas y desplegadas.

La estructura del Sistema cuenta con ocho solapas desarrolladas para poder recorrer las diferentes funcionalidades/tareas que es capaz de realizar el mismo.

Las solapas son las siguientes:

- Inicio
- Comunicación
- Curva de Vacío de MS
- Curva de Cortocircuito de MS
- Diagrama fasorial Behn-Eschenburg
- Diagrama fasorial Potier
- Paralelo de MS
- Diagrama P-Q de MS

## Descripción de las solapas

### Solapa de Inicio

Al ejecutar el software se despliega la pantalla de Inicio que corresponde a la primer solapa utilizada como punto de partida del Sistema. La misma es la única con la opción de salir y cerrar el software mediante un botón de Cerrar. A continuación en la Figura 5.6 se observa el aspecto.



Figura 5.6: Pantalla de inicio de Software

### Solapa de Curva de Vacío de MS

En esta solapa se encuentra todo lo necesario para realizar el Ensayo de vacío de la MS. En la figura 5.7 se puede ver las diferentes partes que componen la misma.

Sobre la izquierda bajo el título de Gráfico Curva de Vacío MS se despliega el gráfico de la tensión de vacío  $E$  en función de la corriente de excitación  $i_{exc}$ .

Luego en la parte derecha de la pantalla se encuentra el esquemático del conexionado correspondiente al ensayo, desplegando las variables de interés sobre el mismo. En la parte inferior de la pantalla aparece la sección de grabado de datos, la cual se detalla en la sección 5.2.1.

## 5.2. Software desarrollado en LabVIEW

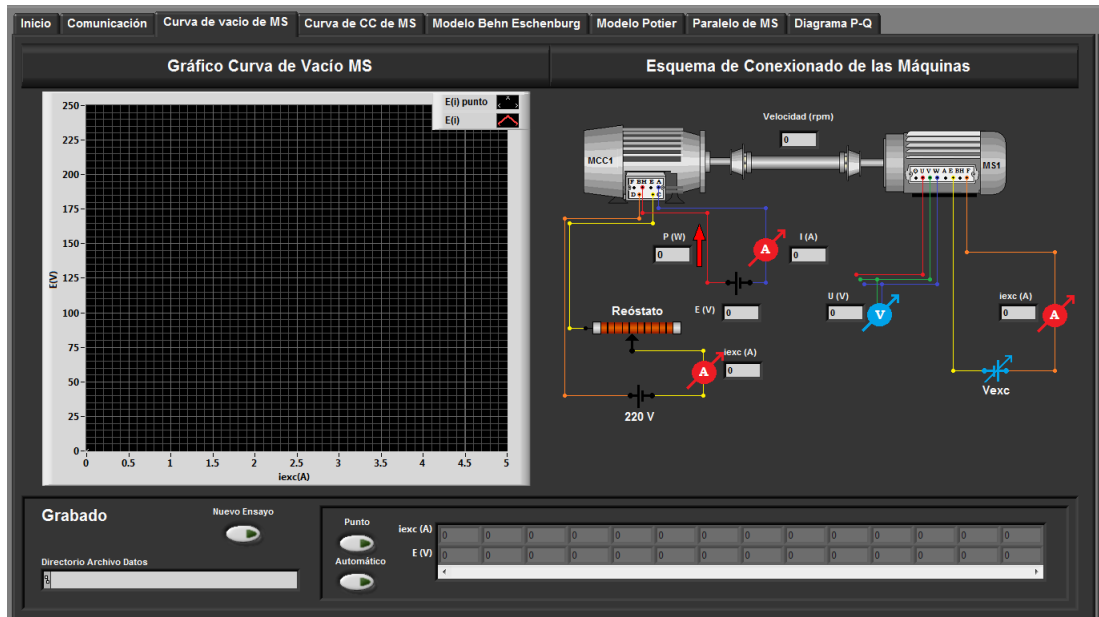


Figura 5.7: Solapa Curva de Vacío de MS

### Solapa de Curva de Cortocircuito de Máquina Síncrona

En esta solapa encuentra lo necesario para realizar el Ensayo de Cortocircuito de la MS. En la figura 5.8 se puede ver las diferentes partes que componen la misma.

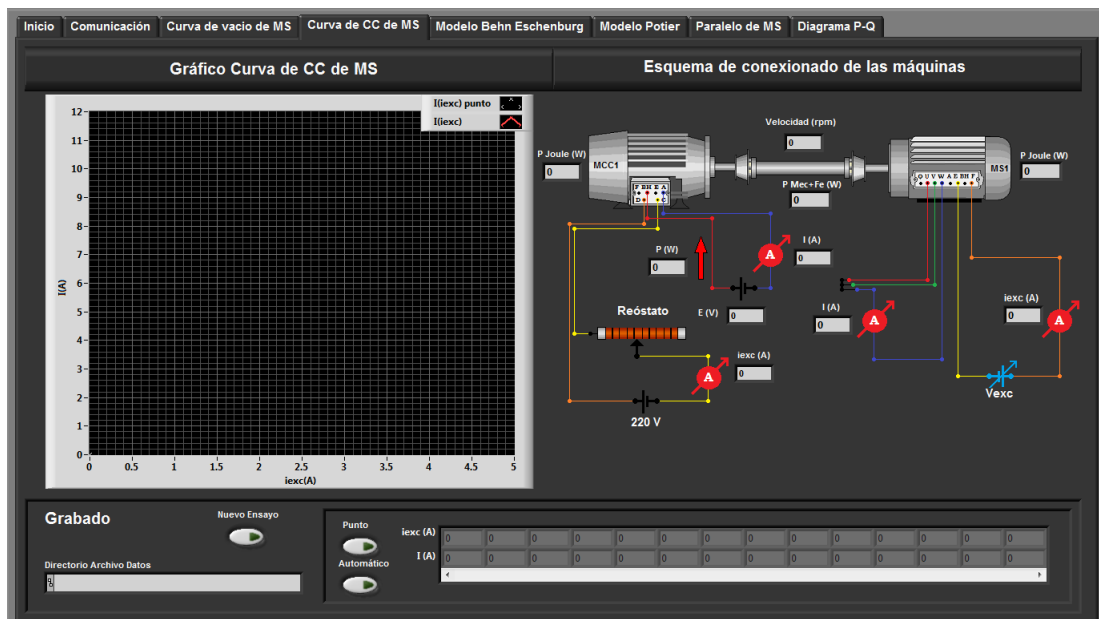


Figura 5.8: Solapa Curva de CC de MS

## Capítulo 5. Software

Sobre la izquierda de la solapa bajo el título de Gráfico Curva de CC de MS se encuentra el gráfico de la corriente de carga  $I$  en función de la corriente de excitación  $i_{exc}$  de la MS.

Luego en la parte derecha de la pantalla se encuentra el esquemático del conexionado correspondiente al ensayo, sobre el cual se despliegan las variables de interés.

En la parte inferior de la pantalla aparece la sección de grabado de datos cuyas características se describen en la sección 5.2.1.

### Solapa Modelo Behn Eschenburg

En esta solapa se encuentra el Modelo de Behn Eschenburg de la MS. En la figura 5.9 se muestra la pestaña donde se pueden ver las diferentes secciones con las que cuenta la misma.

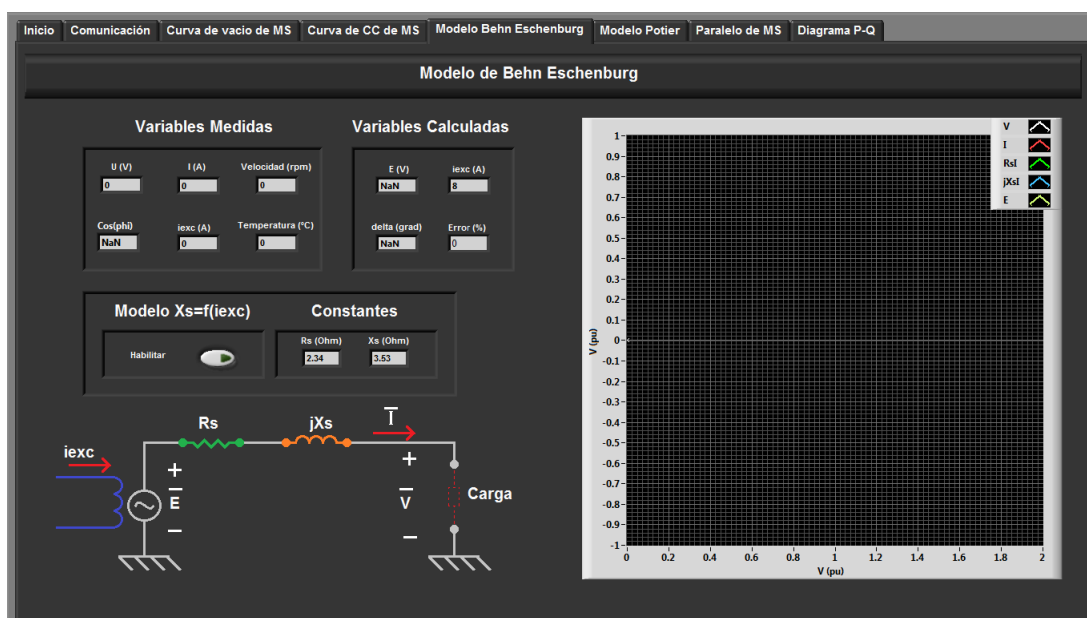


Figura 5.9: Solapa Modelo de Behn Eschenburg

En la parte izquierda se encuentran las variables de interés separadas en dos grupos, por un lado las que corresponden a mediciones que se están realizando en el momento del ensayo bajo el título de **Variables Medidas** y por otro las calculadas bajo el título de **Variables Calculadas**. Luego más abajo se encuentra un botón llamado 'Deshabilitar' bajo el título **Modelo Lineal**, presionándolo se deshabilita el modelo lineal en el cálculo. Luego se despliega  $R_s$  y  $X_s$  en el cuadro denominado **Constantes**.

El botón 'Deshabilitar' bajo el título **Modelo Lineal**, corresponde a que cuando se deshabilita este modelo en el software, lo que se está realizando es cambiar la constante  $X_s$  del modelo lineal por el valor que resulta de interpolar la curva  $X_s(i_{exc})$  según la corriente  $i_{exc}$  medida.



## 5.2. Software desarrollado en LabVIEW

En el gráfico a la derecha de la pantalla se despliegan todos los vectores que intervienen en el modelo de forma gráfica (ver B.1). Los módulos de los vectores en el gráfico son representados mediante valores en por unidad (pu), de esta forma se unificaron criterios de visualización respecto al modelo de Potier. Tomando como base los valores nominales de las MS, siendo estos:  $S = 3505 VA$  y  $U = 220 V$ .

### Solapa de Modelo Potier

En la parte superior izquierda de la solapa como se puede observar en la figura 5.11 se despliegan valores del modelo agrupados de la siguiente forma:

- Variables medidas
- Variables calculadas
- Constantes

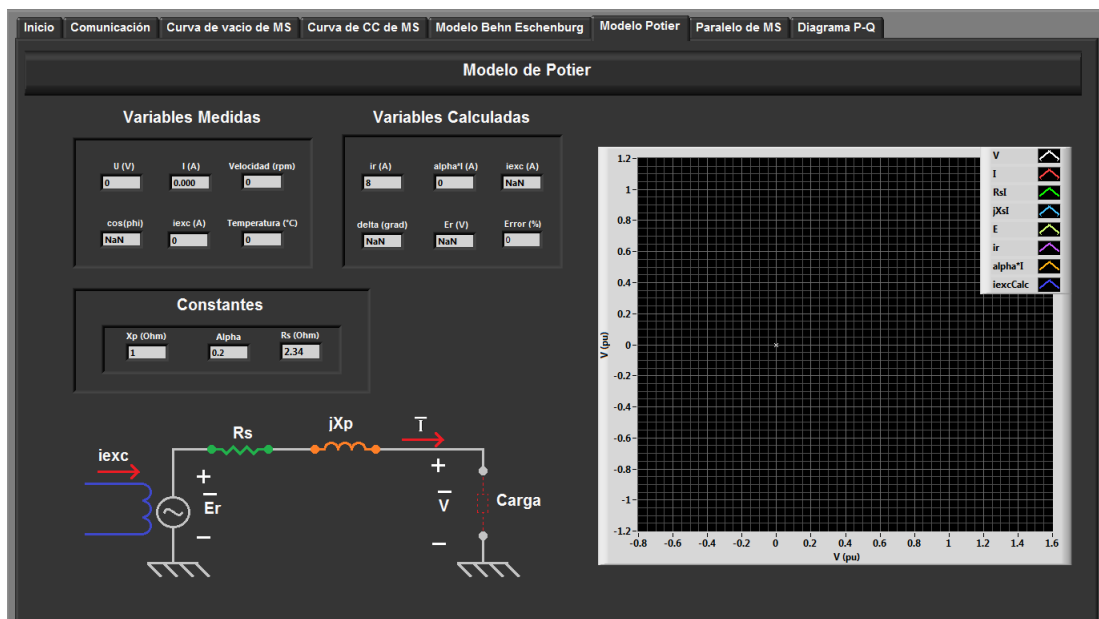


Figura 5.10: Solapa Modelo Potier

Sobre la parte inferior izquierda de la pantalla se encuentra representado el esquemático del modelo. Finalmente a la derecha se despliega el gráfico fasorial del modelo con todos los parámetros de interés (ver B.2). Los módulos de los vectores en el gráfico son representados mediante valores en por unidad (pu), de esta forma se logra ver los dos triángulos relevantes del modelo. Tomando como base los valores nominales de las MS, siendo estos:  $S = 3505 VA$  y  $U = 220 V$ .

### Solapa de Puesta en Paralelo de Máquinas Síncronas

En esta solapa se encuentra el esquemático del conexionado del ensayo de puesta en marcha dos MS en paralelo, están las MS que forman el paralelo junto con las máquinas de DC encargadas de arrastrarlas como se puede ver en la figura 5.11.

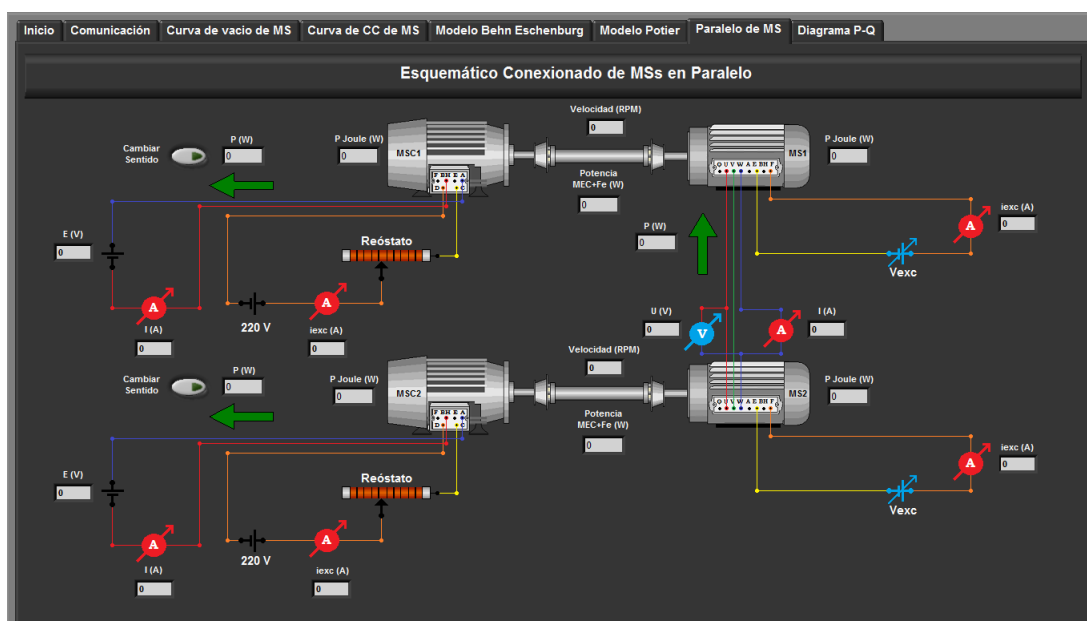


Figura 5.11: Solapa de Puesta en paralelo de MS

En la parte superior de la pantalla se encuentra el Grupo 1 compuesto de una MS y una máquina DC de arrastre, en la parte inferior el Grupo 2 con sus correspondientes máquinas. Sobre el esquemático se despliegan los valores de los parámetros de interés de cada máquina que interviene en el ensayo.

#### ■ Máquinas DC

- Tensiones
- Corrientes de excitación y de armadura
- Valor y sentido de potencia entregada
- Potencia correspondiente a pérdidas Joule. El cálculo se realizó mediante  $P_{MCC} = R_{MCC}I^2$ , siendo  $R_{MCC}$  e  $I$  la resistencia y corriente de armadura respectivamente.
- Velocidad

## 5.2. Software desarrollado en LabVIEW

### ■ Máquinas sincrónicas

- Tensiones de línea
- Corrientes de excitación y de línea
- Valor y sentido de potencia entregada
- Potencia correspondiente a pérdidas Joule. El calculo se realizó mediante  $P_s = 3R_s I_s^2$ , siendo  $R_s$  y  $I_s$  la resistencia y corriente de línea de la MS.
- Velocidad

### ■ Grupo generador

- Valor y sentido de potencia activa ver figura:5.12 bloque **A**
- Potencia correspondiente a pérdidas en el hierro y Joule. El calculo se realizó como  $P_{Fe} + P_{Mec} = P_{IN} - P_{OUT} - P_s - P_{MCC}$ , siendo  $P_{Fe}$  y  $P_{Mec}$  las pérdidas en el hierro y mecánicas respectivamente.  $P_{IN}$  y  $P_{OUT}$  corresponden a la potencia de entrada y salida despectivamente.
- Tensión y corriente de línea correspondientes al paralelo ver figura 5.12 bloque **B**

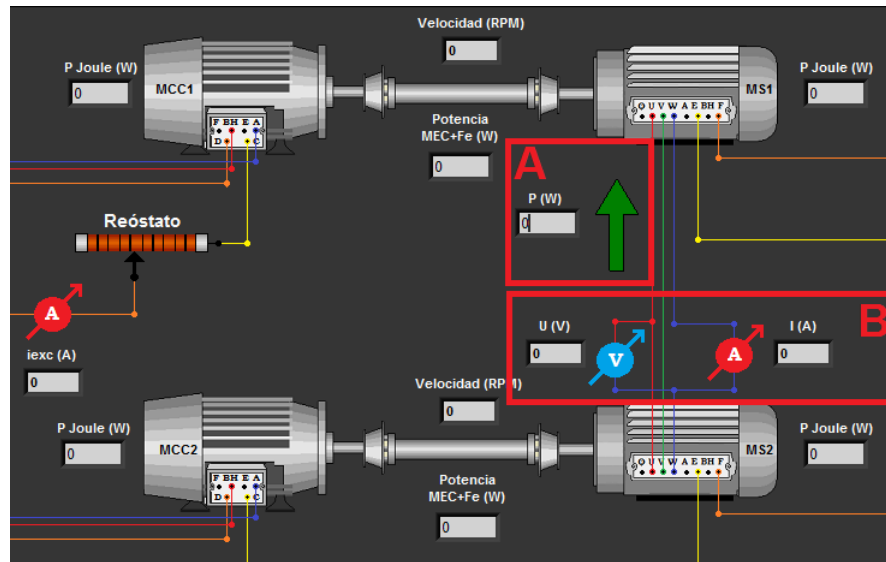


Figura 5.12: Puesta en paralelo de MS

### Solapa de Diagrama P-Q de MS

Esta solapa corresponde al estudio del diagrama P-Q y de los límites de funcionamiento de la MS, en el centro de la pantalla se despliegan las magnitudes correspondientes al punto en el que se encuentra funcionando.

En el gráfico se podrán ver los límites de potencia de operación de la MS.

El punto nominal de funcionamiento de la máquina es representado mediante la semirecta con un ángulo de  $\phi_n$ , correspondiendo éste al  $\cos(\phi_n)$  nominal de la máquina.

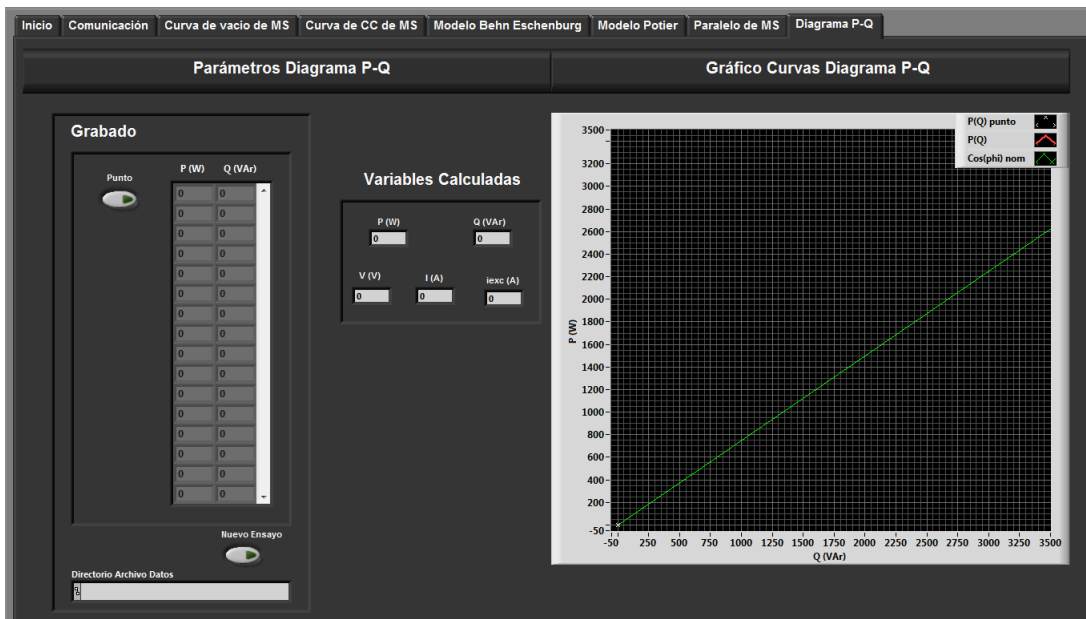


Figura 5.13: Solapa de Diagrama P-Q

En la parte izquierda de la pantalla aparece la sección de grabado de datos cuyas características se describen en la sección 5.2.1.

## 5.2. Software desarrollado en LabVIEW

### Grabado de datos durante los ensayos

Para los Ensayos de CC, Vacío y Diagrama P-Q de las máquinas sincrónicas se tiene la posibilidad de guardar los datos en un archivo .txt para su eventual estudio en un tiempo posterior a las pruebas. Para esto las solapas correspondientes a estos ensayos cuentan con una sección encargada de brindar las herramientas para poder hacerlo.

En la parte inferior de la pantalla se encuentra esta sección, compuesta por dos partes como se puede ver en la Figura 5.14.

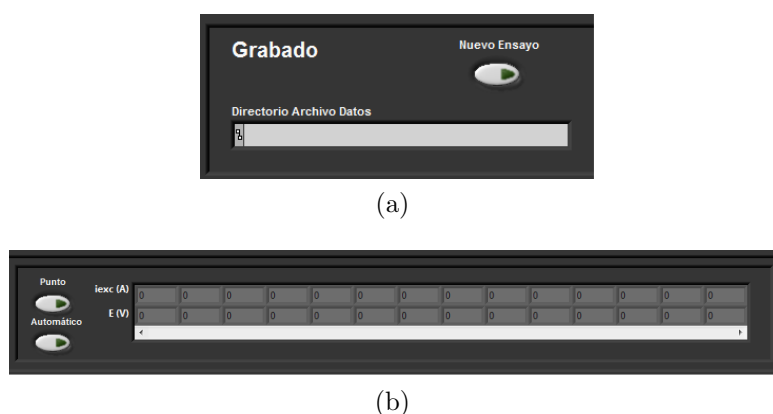


Figura 5.14: 5.14a: Directorio Archivo Datos 5.14b: Controles guardado de Datos

En la 5.14a se encuentra el campo Directorio Archivo Datos donde se muestra el nombre del archivo que se creó y la ubicación para guardar los datos relevados. El nombre del archivo como también el directorio donde se guarda, es preestablecido por el Sistema. También se encuentra un botón llamado Nuevo Ensayo el cual al presionarlo inicializa los parámetros en cero y crea un nuevo archivo de grabado, con los datos correspondientes al nuevo ensayo nombrado éste de la misma forma con la diferencia de un índice para diferenciarlo.

En la 5.14b se encuentran los controles de guardado de datos mediante dos botones, uno llamado Punto con el cual se pueden guardar un conjunto de valores seleccionándolos puntualmente uno a uno en el momento que se presiona. También se encuentra otro llamado Automático para los casos de Ensayo de CC y Vacío, donde al presionarlo el Sistema comienza a guardar datos automáticamente, de tal forma que solo lo hace cuando la variación de las variables relevadas se sale de una ventana de histéresis preestablecida (para el seteo de los parámetros de esta ventana referirse a 5.2.2), solo en ese momento se guarda un nuevo dato.

A medida que se graban los puntos relevados por cualquiera de los métodos, éstos son graficados y desplegados en el array que se muestra en 5.14b y pueden ser recorridos con la barra de desplazamiento horizontal existente.

En las figuras 5.15, 5.16 y 5.17 se muestran los diagramas de flujo del software para los ensayos de CC, Vacío y Diagrama P-Q respectivamente.

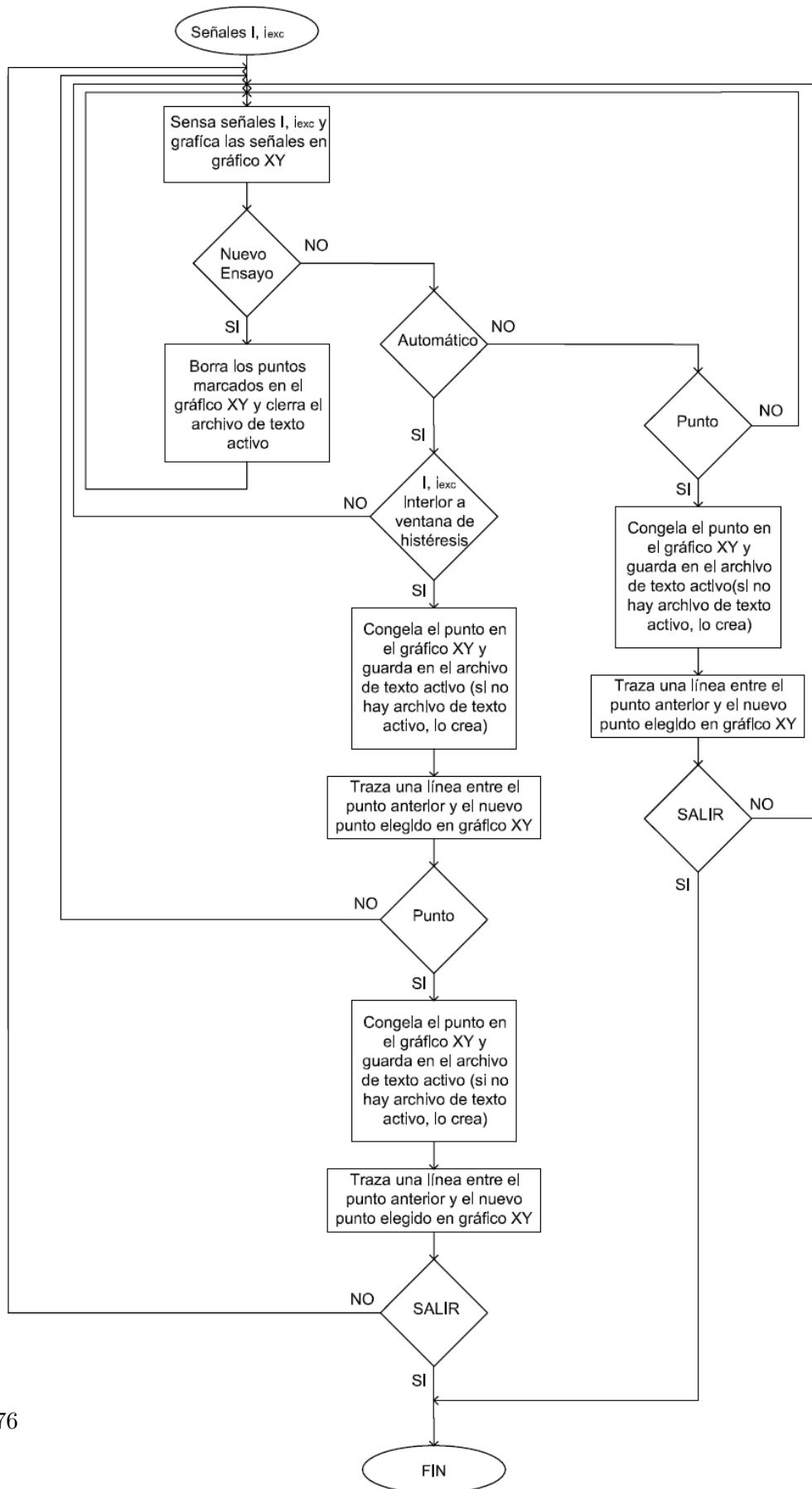


Figura 5.15: Diagrama de flujo para ensayo de CC

5.2. Software desarrollado en LabVIEW

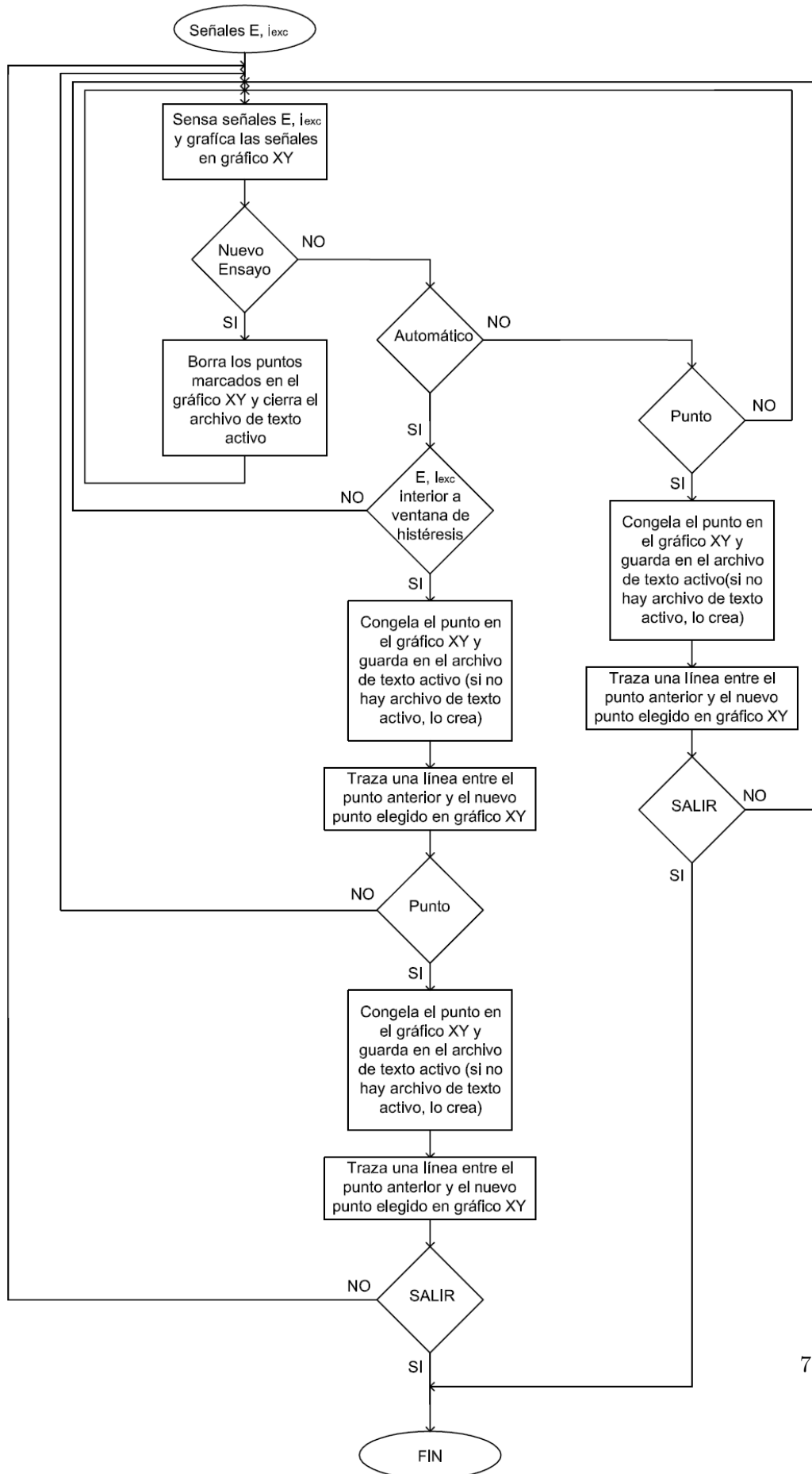


Figura 5.16: Diagrama de flujo para ensayo de Vacío

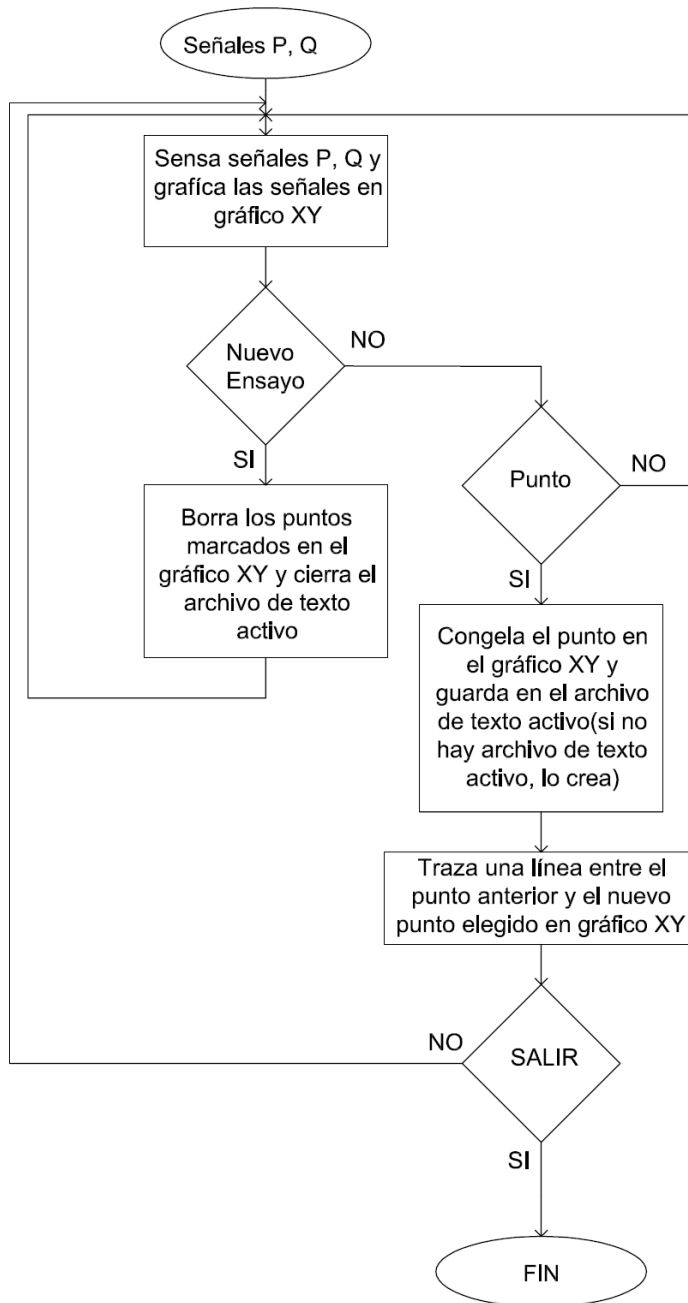


Figura 5.17: Diagrama de flujo para Diagrama P-Q



### Acondicionamiento de software en pantalla

Luego de desarrollar el aspecto, la estructura y dinámica, para la optimización del software en pantalla en el momento de la ejecución se incorporó un bloque encargado de utilizar toda la pantalla en tiempo de ejecución. Este bloque se muestra a continuación en la figura: 5.18.

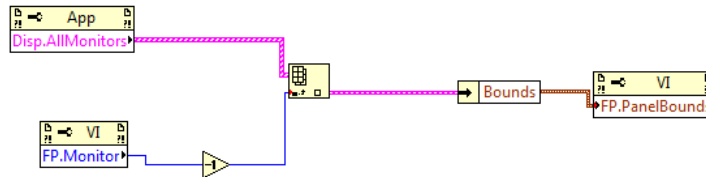


Figura 5.18: Bloque encargado del Full Screen

### 5.2.2. Configuración parámetros del software

Para realizar diferentes configuraciones de los parámetros del software se cuenta con una carpeta llamada 'ConstantesCurvas modelo BE'. La misma se encuentra ubicada junto a la aplicación desarrollada en LabVIEW, contiene archivos .txt con los seteos de constantes y curvas características de las MS y MCC. Luego de instalar la aplicación en la PC, se debe ubicar esta carpeta en el disco 'C :' de la PC. La misma contiene los siguientes archivos:

- Cte\_MS\_MCC.txt
- CtePotier Alpha\_Xp.txt
- CurvaXs\_iexc.txt
- CurvaVacioE\_iexc.txt
- LimHistSenal.txt
- ValoresNomMS.txt
- LimHistGrabadoAutoEVacio.txt
- LimHistGrabadoAutoECC.txt

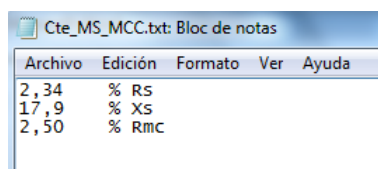
A continuación se describen los archivos .txt mencionados anteriormente:

**Cte\_MS\_MCC.txt:** en este archivo se encuentran las constantes correspondientes a la MS como MCC, estas constantes son:

- MS son las que corresponden a  $R_s$  y la  $X_s$ , siendo la  $X_s$  la de zona lineal
- MCC es la resistencia de armadura  $R_{mc}$

## Capítulo 5. Software

El seteo de las mismas se realiza como se muestra en la figura D.10, donde las constantes se encuentran ubicadas en una sola columna. El primer valor corresponde a la cte.  $R_s$ , el segundo corresponde a la cte.  $X_s$  y el último valor a la cte.  $R_{mc}$ .

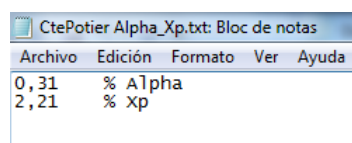


Archivo	Edición	Formato	Ver	Ayuda
2,34	%	RS		
17,9	%	Xs		
2,50	%	Rmc		

Figura 5.19: Seteo de constantes de MS y de MCC

**CtePotier Alpha\_Xp.txt:** en este archivo se encuentran las constantes de la MS correspondientes al modelo de Potier, siendo estas  $\alpha$  y  $X_p$ .

El seteo se realiza como se muestra en la figura D.11, donde las constantes se encuentran ubicadas en una sola columna, el primer valor corresponde a la cte.  $\alpha$  y el segundo corresponde a la cte.  $X_p$ .

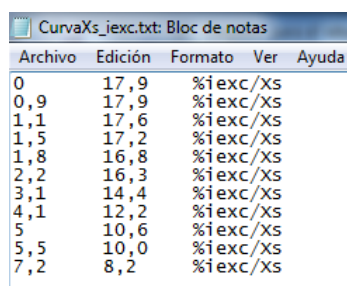


Archivo	Edición	Formato	Ver	Ayuda
0,31	%	Alpha		
2,21	%	xp		

Figura 5.20: Seteo de constantes de la MS modelo Potier

**CurvaXs\_iexc.txt:** este archivo contiene la curva  $X_s$  en función de  $i_{exc}$  ( $X_s(i_{exc})$ ), siendo  $X_s$  la reactancia sincrónica de la MS e  $i_{exc}$  la corriente de excitación de la MS. Los valores de la curva estarán ubicados en columnas. La primera columna corresponde al valor de  $i_{exc}$  y la segunda corresponde al valor de  $X_s$ .

El seteo se realiza como se muestra en la figura D.12, donde los valores de cada variable se encuentran ubicados en dos columnas, la primera corresponde al valor de  $i_{exc}$  y la segunda corresponde al valor de  $X_s$ .



Archivo	Edición	Formato	Ver	Ayuda
0	17,9	%iexc/Xs		
0,9	17,9	%iexc/Xs		
1,1	17,6	%iexc/Xs		
1,5	17,2	%iexc/Xs		
1,8	16,8	%iexc/Xs		
2,2	16,3	%iexc/Xs		
3,1	14,4	%iexc/Xs		
4,1	12,2	%iexc/Xs		
5	10,6	%iexc/Xs		
5,5	10,0	%iexc/Xs		
7,2	8,2	%iexc/Xs		

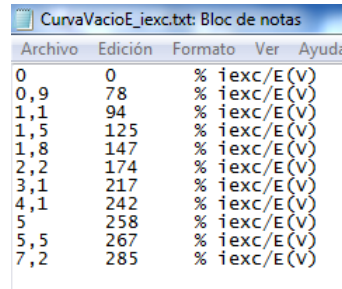
Figura 5.21: Curva  $X_s$  en función de  $i_{exc}$  correspondiente al modelo no lineal de la MS

**CurvaVacíoE\_iexc.txt:** este archivo contiene la curva de vacío  $E$  en función de  $i_{exc}$  ( $E(i_{exc})$ ), siendo  $E$  la tensión de vacío de la MS e  $i_{exc}$  la corriente de

## 5.2. Software desarrollado en LabVIEW

excitación de la MS. Los valores de la curva se encuentran ubicados en columnas, la primera corresponde al valor de  $i_{exc}$  y la segunda corresponde al valor de  $E$ .

El seteo se realiza como se muestra en la figura D.13, donde los valores de cada variable se encuentran ubicados en dos columnas, la primera corresponde al valor de  $i_{exc}$  y la segunda al valor de  $E$ .

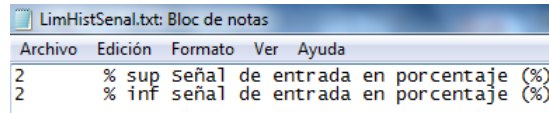


Archivo	Edición	Formato	Ver	Ayuda
0	0	% iexc/E(V)		
0,9	78	% iexc/E(V)		
1,1	94	% iexc/E(V)		
1,5	125	% iexc/E(V)		
1,8	147	% iexc/E(V)		
2,2	174	% iexc/E(V)		
3,1	217	% iexc/E(V)		
4,1	242	% iexc/E(V)		
5	258	% iexc/E(V)		
5,5	267	% iexc/E(V)		
7,2	285	% iexc/E(V)		

Figura 5.22: Curva de vacío de la MS,  $E$  en función de  $i_{exc}$

**LimHistSenal.txt:** en este archivo se setean las constantes para la banda de histéresis para realizar el filtro digital de la señal proveniente del equipo.

El seteo se realiza como se muestra en la figura D.14, donde las constantes se encuentran ubicadas en una sola columna, el primer valor corresponde a la cte. del margen superior de la histéresis y el segundo valor corresponde a la cte. del margen inferior de la misma.

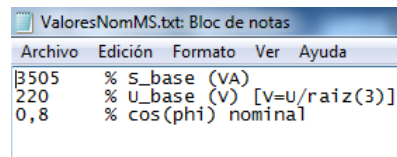


Archivo	Edición	Formato	Ver	Ayuda
2	% sup	Señal de entrada en porcentaje (%)		
2	% inf	señal de entrada en porcentaje (%)		

Figura 5.23: Limite banda de histéresis para filtrado de señales provenientes del equipo

**ValoresNomMS.txt:** en este archivo se realiza el seteo de los valores nominales de la MS, estos son potencia aparente nominal ( $S_n$ ), tensión nominal ( $U_n$ ) y  $\cos(\phi)$  nominal.

El seteo se realiza como se muestra en la figura D.15, donde las constantes se encuentran ubicadas en una sola columna, el primer valor corresponde a  $S_n$ , el segundo corresponde a  $U_n$  y el último valor al  $\cos(\phi)$  nominal.



Archivo	Edición	Formato	Ver	Ayuda
3505	% S_base	(VA)		
220	% U_base	(V) [v=u/raiz(3)]		
0,8	% cos(phi)	nominal		

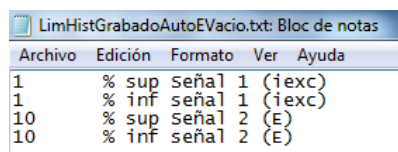
Figura 5.24: Valores nominales de la MS

**LimHistGrabadoAutoEVacio.txt:** en este archivo se setean las constantes de las bandas de histéresis para realizar el grabado de datos automáticos correspondientes al ensayo de vacío. El funcionamiento de la banda para cada señal es,

## Capítulo 5. Software

cuando la señal sale de la banda el software grabará y gráfica el nuevo dato, para luego centrar la banda en el nuevo dato esperando el próximo donde situarse.

El seteo se realiza como se muestra en la figura D.16, donde las constantes de la banda de histéresis se encuentran ubicadas en una sola columna, los dos primeros valores corresponden a la banda seteada para la corriente de excitación en el orden de margen superior e inferior, mientras que los últimos dos valores corresponden a la banda de la señal  $E$  en el orden de margen superior e inferior.

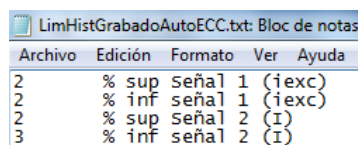


```
LimHistGrabadoAutoEVacio.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
1      % sup señal 1 (iexc)
1      % inf señal 1 (iexc)
10     % sup señal 2 (E)
10     % inf señal 2 (E)
```

Figura 5.25: Limite banda de histéresis para grabado automático de ensayo de vacío

**LimHistGrabadoAutoECC.txt:** en este archivo se setean las constantes para las bandas de histéresis para realizar el grabado de datos automáticos correspondientes al ensayos de vacío. El funcionamiento de la banda de para cada señal es, cuando la señal sale de la banda el software grabará y grafica el nuevo dato, para luego centrar la banda en el nuevo dato esperando el próximo donde situarse.

El seteo se realiza como se muestra en la figura D.17, donde las constantes de la banda de histéresis se encuentran ubicadas en una sola columna, los dos primeros valores corresponden a la banda seteada para la corriente de excitación en el orden de margen superior e inferior, y los últimos dos valores corresponden a la banda de la señal  $I$  en el orden de margen superior e inferior.



```
LimHistGrabadoAutoECC.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
2      % sup señal 1 (iexc)
2      % inf señal 1 (iexc)
2      % sup señal 2 (I)
3      % inf señal 2 (I)
```

Figura 5.26: Limite banda de histéresis para grabado automático de ensayo de cortocircuito

**Nota:** En todos los casos el seteo de las constantes en los .txt se debe realizar respetando la siguiente nomenclatura:

- los datos numéricos que tengan valores decimales deben de ser descritos con una coma (,)
- La separación entre dos valores debe ser con tabulación (TAB)
- los comentarios realizados en cada renglón se deberán separar por una tabulación (TAB) seguido de un signo de porcentaje (%)

## 5.3. Firmware desarrollado para placa Arduino

El firmware desarrollado que corre en la placa Arduino se describe a continuación mediante el siguiente pseudocódigo:

```

Incluir librería I2C

Incluir librería ModBus

Definición de contantes

Inicialización de variables

Seteo {
    Seteo – Comunicación – Serial – USB
    Seteo – Comunicación – Modbus
    Seteo – Comunicación – I2C
    Defino_pins_interrupciones
}

Main{
    Inicio variables de contadores (par encoders)
    Tiempo_inicio
    For  $i = 0$ :NumIteraciones{
        Acumulo_variables_continuas
    }
    Se divide por NumIteraciones+1
    Adquiero_Datos_Modbus
    Se ajusta tensiones por polinomios
    Se genera palabras para Com_USB (valores tensiones)
    Se ajusta corrientes (curvas lineales)
    Se genera palabras para Com_USB (valores corrientes)
    Leer_Sensor_Temp
    Se genera palbras para Com_USB (valor temperatura)
    Delay (para conteo encoders)
    Calculo_Vel

```

## Capítulo 5. Software

Genera palabras para Com\_USB (valores velocidades)

Envío\_Valores\_Com\_USB

} // Fin de Main

Función FillZeros(StringX){

Llena de caracteres 0 los valores en strings (para siempre tener el mismo ancho de palabra en comunicación USB)

}

Función count 1(){

Se incrementa contador (con una interrupción de procesador generador por encoder) para velocidad 1

}

Función count 2(){

Se incrementa contador (con una interrupción de procesador generador por encoder) para velocidad 2

}

En el apéndice E de encuentra descrito el mismo.

# Capítulo 6

## Pruebas y ajustes

### 6.1. Introducción

Una vez desarrolladas y ensambladas las diferentes partes que componen el hardware con el software finalizado, se estuvo en condiciones de comenzar a realizar las pruebas y ensayos sobre el mismo para así evaluar el desempeño y prestaciones en forma global.

Los primeros ensayos realizados, como es natural para un equipo de estas características fueron los correspondientes a las calibraciones de los sensores de medida. Para esto se le inyectó a cada sensor señales en los rangos donde van a operar.

Las calibraciones y ajuste de los sensores se realizó por intermedio de la placa centralizadora de medidas (placa Arduino). Con lo que se probó todo el Sistema de medidas y la velocidad con la cual responden los cambios de los periféricos de medida.

Finalmente se realizó un ensayo tipo de laboratorio para así probar el sistema funcionando en el ámbito para el cual fue desarrollado.

### 6.2. Calibración de Elementos

En las siguientes secciones se detallan las respectivas calibraciones realizadas para cada uno de los sensores que intervienen en las medidas.

#### 6.2.1. Ajustes de las medidas de tensión DC

Como se hace referencia en el capítulo 3 las medidas de tensión DC se realizan a través de un circuito diseñado para tal fin donde intervienen optoacopladores, estas medidas corresponden a la tensión de armadura de cada MCC. Por esto fue necesario que el rango de operación este entre 50 y 250 Vdc.

Para realizar la calibración de estos sensores en primera instancia se relevó la curva de valores medidos reales de tensión contra la medida directa de la placa Arduino.

## Capítulo 6. Pruebas y ajustes

Se constató que las curvas son relativamente lineales a partir de valores de campo superiores a los 40 V y carecen de histéresis. A pesar de ser posible ajustar la curva por una recta, se eligió parametrizar el comportamiento con un polinomio de 3<sup>er</sup> orden a partir de los datos relevados, para tener un ajuste mas preciso de la medida.

En el gráfico mostrado en la figura 6.1 y en la figura 6.2 se muestra el comportamiento del sensor de tensión 1 y 2 respectivamente.

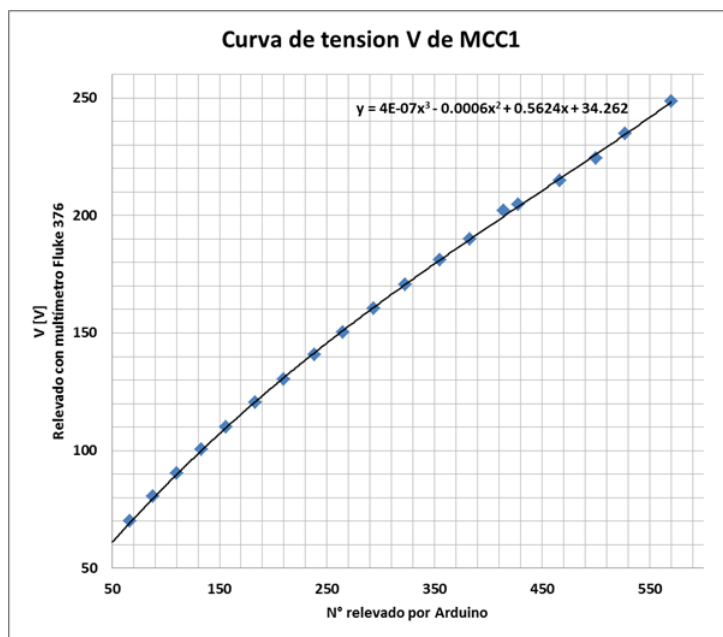


Figura 6.1: Curva de Medidas de Campo en función de los enteros medidos por Arduino

El polinomio que se muestra en cada gráfica es el que mejor se aproxima en cada caso, los cuales fueron utilizados en el firmware de la placa, para que la misma realice la conversión del valor obtenido con el valor de campo (real).



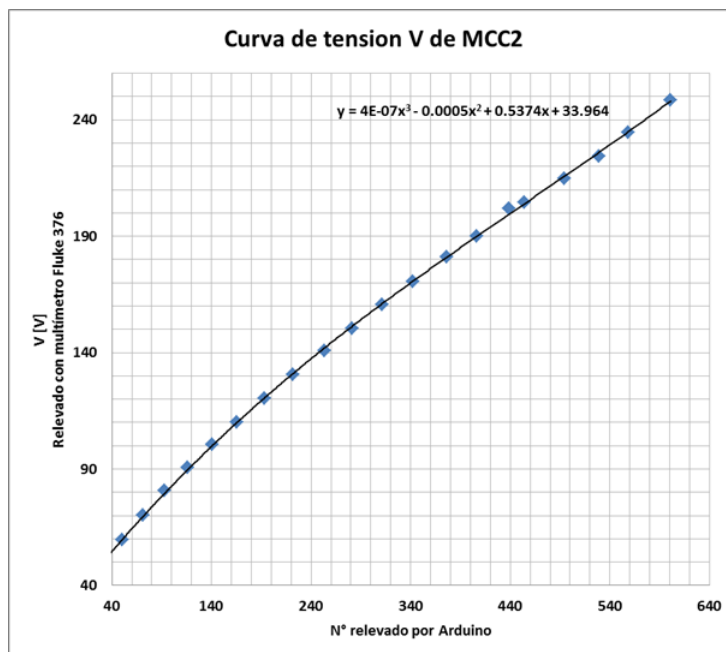


Figura 6.2: Curva de Medidas de Campo en función de los enteros medidos por Arduino

### 6.2.2. Ajustes de sensores de corriente

Los sensores de efecto Hall utilizados según datos del fabricante se comportan de forma lineal, es decir ante una variación lineal de la entrada, la salida varía de igual forma. Para comprobar su funcionamiento y relevar su transferencia se realizó el ensayo que se describe a continuación.

Como punto de partida, para todos los sensores se relevaron los extremos iniciales y finales de funcionamiento, con estos puntos se calcularon las curvas paramétricas lineales correspondientes a cada uno. Obtenidos los coeficientes de las parametrizaciones se ajustaron las transferencias de los sensores mediante software (firmware del Arduino) a ese comportamiento.

Luego de este ajuste, para comprobar el comportamiento de los mismos se les inyectó de forma gradual y creciente diferentes valores de corriente DC y se relevaron los resultados que fueron desprendiendo.

Obtenidos los puntos, se procedió a graficarlos y contrastarlos con las parametrizaciones lineales mencionadas anteriormente cuyos resultados se muestran a continuación, donde además en el propio gráfico se muestran las ecuaciones de las curvas.

#### Sensores de $i_{exc}$ de las MCC (sensores Hall de 5 A )

En las figuras 6.3 y 6.4 se encuentran el gráfico los puntos relevados vs la curva paramétrica del sensor encargado de medir la corriente DC de excitación de la MCC 1 y 2 respectivamente.

## Capítulo 6. Pruebas y ajustes

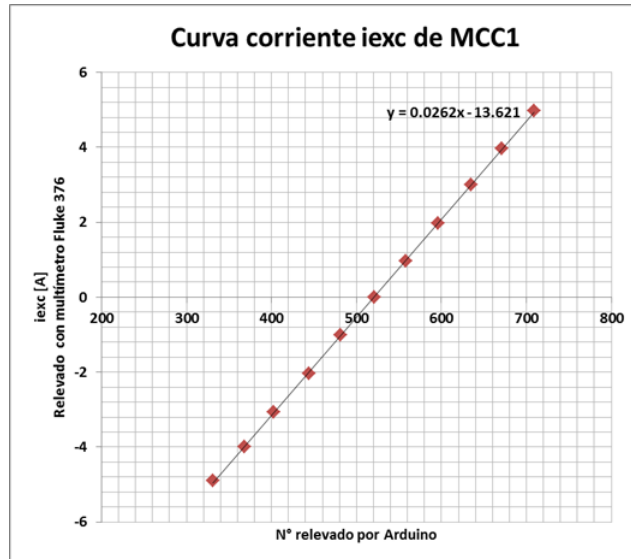


Figura 6.3: Curva de comportamiento de  $i_{exc}[A]$  de MCC 1

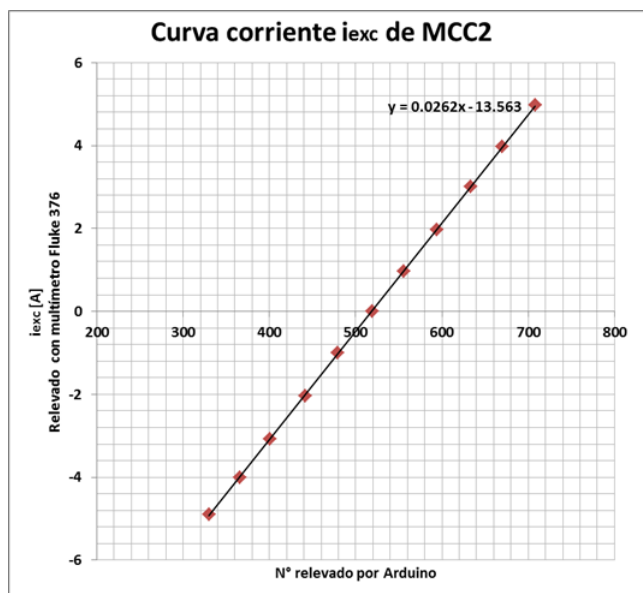


Figura 6.4: Curva de comportamiento de  $i_{exc}[A]$  de MCC 2

## 6.2. Calibración de Elementos

### Sensores de I de las MCC (sensores Hall de 30 A )

En las figuras 6.5 y 6.6 se encuentran graficados los puntos relevados del sensor encargado de medir la corriente I de DC de la MS 1 y 2 respectivamente.

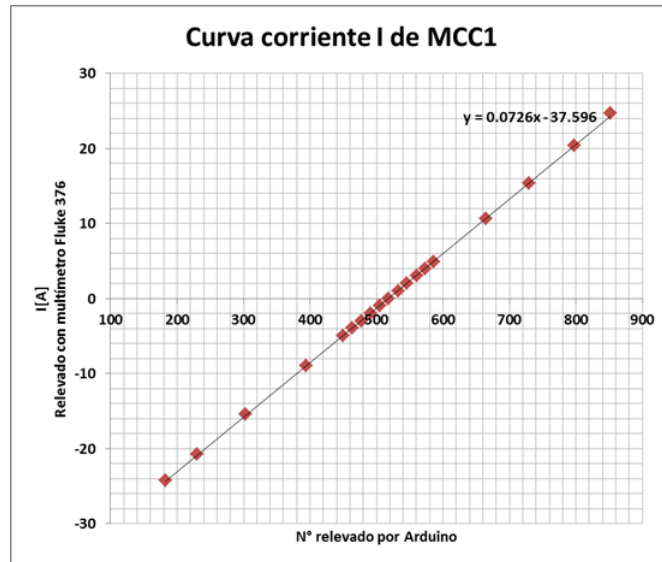


Figura 6.5: Curva de comportamiento de I[A] de MCC 1

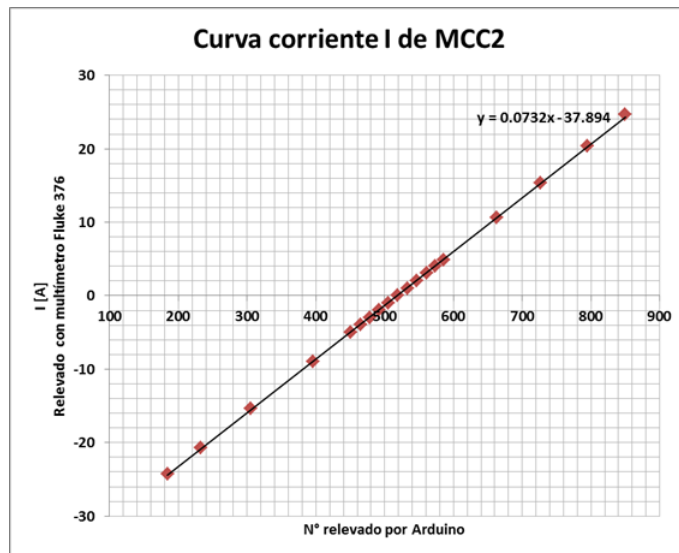


Figura 6.6: Curva de comportamiento de I[A] de MCC 2

## Capítulo 6. Pruebas y ajustes

### Sensores de $i_{exc}$ de las MS (sensores Hall de 20 A )

En las figuras 6.7 y 6.8 se encuentran graficados los puntos relevados del sensor encargado de medir la corriente DC de excitación de la MS 1 y 2 respectivamente.

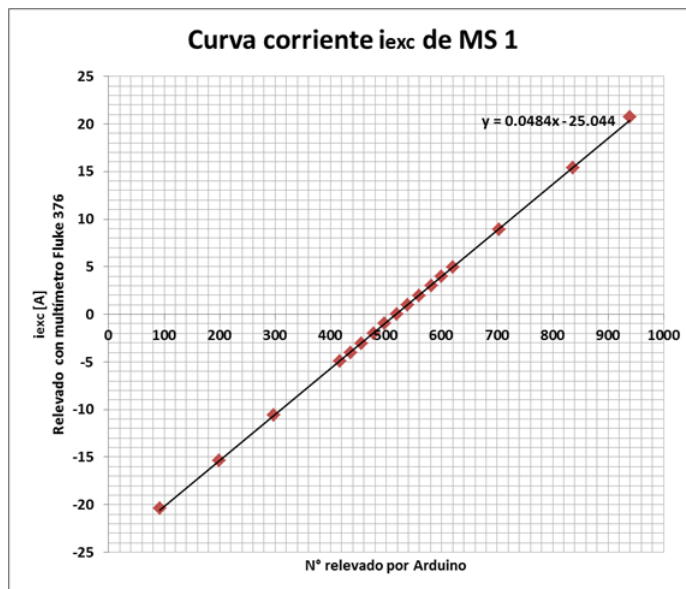


Figura 6.7: Curva de comportamiento de  $i_{exc}$ [A] de MS 1

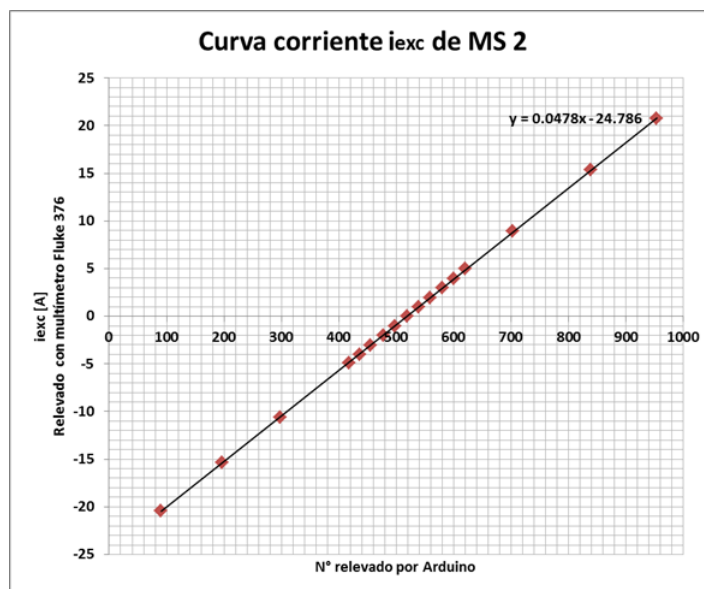


Figura 6.8: Curva de comportamiento de  $i_{exc}$ [A] de MS 2

## 6.2. Calibración de Elementos

Para todos los sensores, se midió el valor que relevaba el Arduino sin corriente los cuales fueron próximos al valor dado por el fabricante, pero no exactamente el mismo. El offset de la hoja de datos (Sensor Hall ACS712) es de 2.5 V lo cual corresponde al 512 de medida analógica en Arduino. Los valores de offset reales para estos sensores medidos se muestran en la tabla 6.1.

<b>Nom. Corriente</b>	<b>Offset [<math>N^\circ</math>]</b>
iexc MCC1	522
iexc MCC2	520
I MCC1	519
I MCC2	519
iexc MS1	519
iexc MS2	519

Tabla 6.1: Tabla con valores de offset relevados

Todas las medidas de corriente fueron contrastadas con los multímetro Fluke 376 y EXTECH MN35 dependiendo el caso, en los gráficos se encuentra especificado caso a caso el utilizado.

### 6.2.3. Sensor de temperatura

Como se describió en la sección 3.1.5 la calibración del mismo consiste en modificar el valor de emisibilidad del cuerpo que se quiere sensor. Para esto se tomó varios valores de temperatura del bobinado a sensor, paralelamente se hizo lo mismo con el sensor del equipo cotejando resultados y procediendo al ajuste.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 7

## Conclusión

### Introducción

En el transcurso de este capítulo se pretende describir tanto los logros obtenidos, así como las problemáticas que surgieron en la realización de este Proyecto. También se serán detalladas mejoras y/o ampliaciones posibles.

Durante el desarrollo del Proyecto, se analizaron distintas arquitecturas de Sistema, se diseñó y construyó hardware a la medida de los requerimientos anteriormente mencionados, se programó un software de visualización, y finalmente se realizó el testeo y debugeo de todo el Sistema.

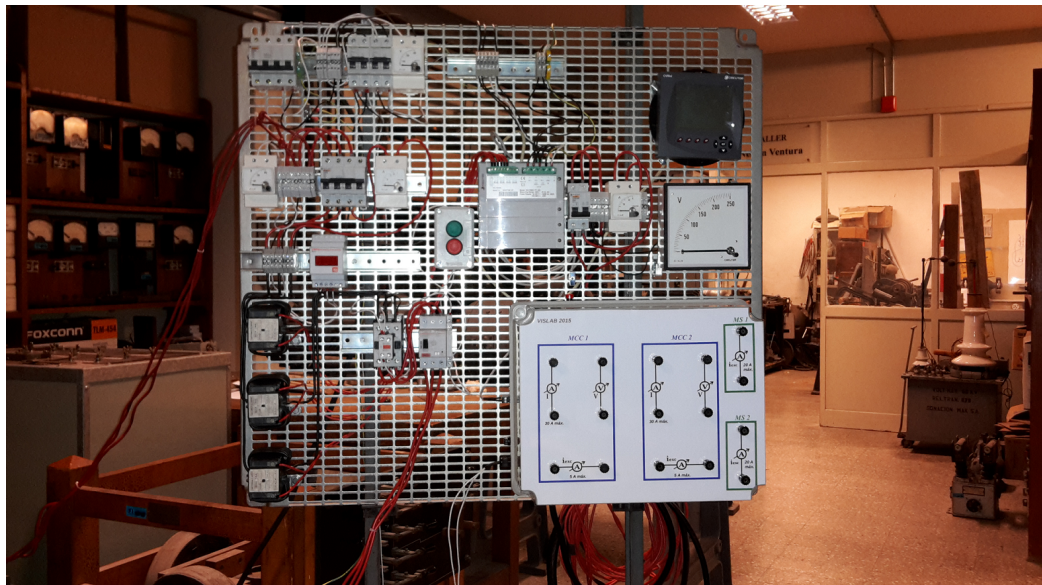


Figura 7.1: Fotografía del Equipo en su disposición final en panel

## Capítulo 7. Conclusión

Las temáticas necesarias para la implementación completa del Proyecto, incluyeron entre otras: electrónica (placas, fuente), telecomunicaciones (Modbus, I2C, normativas y cableado de buses) y máquinas eléctricas; de las cuales solamente la última estaba dentro de las conocidas previamente por los integrantes de este grupo, lo que implicó necesariamente un proceso de estudio y familiarización profundamente enriquecedor.

El resultado final, fue un conjunto de software y equipamiento que cumplen satisfactoriamente los objetivos planteados, contándose además con elementos que a posteriori podrán ser expandidos con otras características relevantes al Laboratorio (tanto en el caso del SW como el HW).

### 7.1. Cumplimiento de Objetivos

De su uso en ensayos de prueba, se vio que el Sistema simplifica la interpretación de los valores medidos y los concentra en una pantalla de manera tal que es fácil saber a qué medida corresponden. En particular se mejoraron las siguientes situaciones:

- En el ensayo de vacío, con el Sistema que se implementó se redujo considerablemente el tiempo de realización del mismo (de 20-25 min a menos de 5 min). También gracias a este Sistema fue posible la observación de la construcción del gráfico a medida que se muestrean los datos, así como guardar los resultados en formato digital.
- Para el ensayo de cortocircuito, si bien este ensayo es más breve al relevarse menor cantidad de puntos, se mejoró en los mismos aspectos que para el caso de ensayo de vacío.
- Para modelos de Behn-Eschenburg y Potier, se alcanzó la visualización de los fasores en tiempo real así como también de los valores medidos y calculados según corresponde a cada modelo; funcionalidad que hasta este Proyecto no se tenía en el Laboratorio. Con esto se puede contrastar los valores reales del circuito contra los esperados por los modelos teóricos estudiados, y en particular, para el caso de Behn-Eschenburg se dejó la opción de intercambiar entre valor de  $X_s$  contante o calculado dinámicamente en función de la corriente de excitación de la MS.
- Para los ensayos y puesta en paralelo de MS se reunieron todos los valores en los diagramas de conexionado correspondientes, logrando así una rápida y clara lectura e interpretación de estos, pudiéndose asociar los mismos a los elementos físicos según corresponda.
- El Sistema pudo integrar las velocidades de eje de las cuales sus valores son medidos continuamente, y también un sensor de temperatura sin contacto, que es móvil lo que permite medir cualquier objeto que se desee (teniendo en cuenta que se debe calibrar para la emisividad del objeto deseado).



- Con este desarrollo se pudo implementar un nuevo ensayo que hasta el momento no era realizado, siendo éste el diagrama P-Q de la MS. En la visualización de este ensayo se reunió todos los datos necesarios y su gráfico, el cual cuenta con características similares a los de ensayo de vacío y cortocircuito (construcción instantánea de gráfico y guardado de datos).

Todas estas mejoras hacen al curso de Taller de Máquinas Eléctricas más didáctico y dinámico, por lo que se considera que el Sistema cumple satisfactoriamente con los objetivos del Proyecto.

## 7.2. Limitaciones

Además de las limitaciones descritas en el anexo 8, que corresponden a los límites de medición de las entradas al hardware, y de los datos por algunas de sus protecciones, el Sistema está acotado por otras razones que se mencionan a continuación.

Para todo lo correspondiente a los tiempos de actualización de los datos en Software, los valores fueron actualizados aproximadamente cada 250 ms, lo que significó que todo cambio más rápido que este tiempo no será registrado por el software. En el caso particular de la adquisición de los valores de red trifásica a través del CVMk2 que actualiza sus registros internos cada un segundo, esto implicó, que todos estos valores fueran repetidos en cada ciclo de 250 ms y solo variarán cada un segundo.

Las incertidumbres correspondientes a los valores que se adquirieron a través de protocolos de comunicación, son los propios de los equipos comunicados (valores trifásicos y temperatura). Para las corrientes y tensiones continuas, las medidas se calibraron contra un multímetro Fluke modelo 376 y EXTECH MN35 respectivamente. Los encoder fueron calibrados aplicándose una onda cuadrada de un generador de ondas en la entrada correspondiente del equipo, por lo que se entiende que la incertidumbre en las medidas es una unidad de las doscientas cuentas enviadas por revolución. No se realizó un estudio formal de incertidumbres de las medidas por considerarse el mismo, fuera del alcance del proyecto.

Otra limitante intrínseca al equipamiento y software, es que no estuvo previsto la medición de valores transitorios, dado que lo propuesto para el Proyecto fue analizar el comportamiento en régimen. De ser deseables estas características sería necesario reformular todo el Sistema.

## 7.3. Posibles mejoras

### 7.3.1. Software

El software de visualización tiene capacidad para poder expandir las cantidades de ensayos representados (siempre y cuando no requieren nuevas medidas), así como implementar la visualización de otros modelos de máquinas y/o conexiones.

### 7.3.2. Mejoras generales del equipo

El hardware que se implementó cuenta con la capacidad de dar señales de comando, debido a que algunos de los pines de la placa Arduino pueden operar como salidas digitales en 5 V o salidas tipo PWM. Si entonces lo que se desea es operar y/o crear lazos de control entre el Sistema y las máquinas de Laboratorio, es necesario resolver los siguientes puntos:

- Formular un nuevo protocolo serial de comunicación entre Arduino y LabVIEW que sea bidireccional, donde se permita que los comandos dados en el software sean interpretados por la placa Arduino.
- Crear una placa de interfaz con relés comandables para realizar operaciones del tipo On/Off o implementar drivers de motor que acepten una señal tipo PWM como entrada de comando (para variar elementos de manera continua).
- Implementar en LabVIEW teclas, barras de comando o valor numérico editable de set-point, cuyos valores sean enviados e interpretados por el Arduino.
- Realizar el estudio formal del lazo de control deseado e implementar su código en LabVIEW o, de ser necesario un manejo de tiempos mas acotados, realizar la programación de lazo de control en la placa.

Otro agregado posible es el de una pequeña pantalla LCD que permita leer los valores medidos por el Arduino directamente para poder ver los datos desde la caja sin necesidad de conectar a la PC.

## 7.4. Resumen

En este capítulo final, se dieron conclusiones generales de este trabajo, así como la descripción detallada del cumplimiento de los objetivos propuestos al inicio de este Proyecto, y de las limitaciones propias del equipo diseñado y periféricos conectados al mismo.

Finalmente, se indicaron pautas las cuales pueden, tanto mejorar el comportamiento del equipo en condiciones operativas, como expandir las capacidades del Sistema y agregar nuevas funcionalidades aprovechando todo el HW y SW existente.

Como comentario final, el grupo de VisLab 2015 se siente agradecido de haber tenido la oportunidad de realizar un Proyecto de aplicación real e integrador de varias disciplinas de la ingeniería, y de haber aportado herramientas para que las próximas generaciones de estudiantes se vean beneficiados con experiencias de Laboratorio mas dinámicas, didácticas y enriquecedoras.

# Apéndice A

## Desarrollo de programación LabVIEW

### A.1. Mapa del Software Principal

Con los datos de corrientes y tensiones tanto de alterna como continua, desfases y potencias obtenido desde el hardware y software desarrollado, se realizan distintas rutinas en LabVIEW para obtener y comparar los modelos de las máquinas sincrónicas en estudio. Estos modelos son Behen-Eschenburg y Potier, también se realizaron rutinas de software para el relevamiento de los ensayos de vacío y cortocircuito de las máquinas y el diagrama P-Q de la máquina sincrónica.

La programación se realizó desde un programa principal con distintos subVI, el mapa del programa principal se presenta en figura A.1 y los bloques allí marcados corresponden a las rutinas creadas para el procesamiento de los datos según se indican en los siguientes items:

- Bloque A: Comunicación Arduino LabVIEW.
- Bloque B: Transmisión de señales y filtrado para las distintas rutinas creadas.
- Bloque C: Estudio de dos máquinas sincrónicas generando en paralelo.
- Bloque D: Ensayo de vacío de la máquina sincrónica.
- Bloque E: Ensayo de cortocircuito de la máquina sincrónica.
- Bloque F: Diagrama P-Q de la máquina sincrónica.
- Bloque G: Ensayo Modelo de Behen-Eschenburg máquina sincrónica.
- Bloque H: Ensayo Modelo de Potier máquina sincrónica.

## Apéndice A. Desarrollo de programación LabVIEW

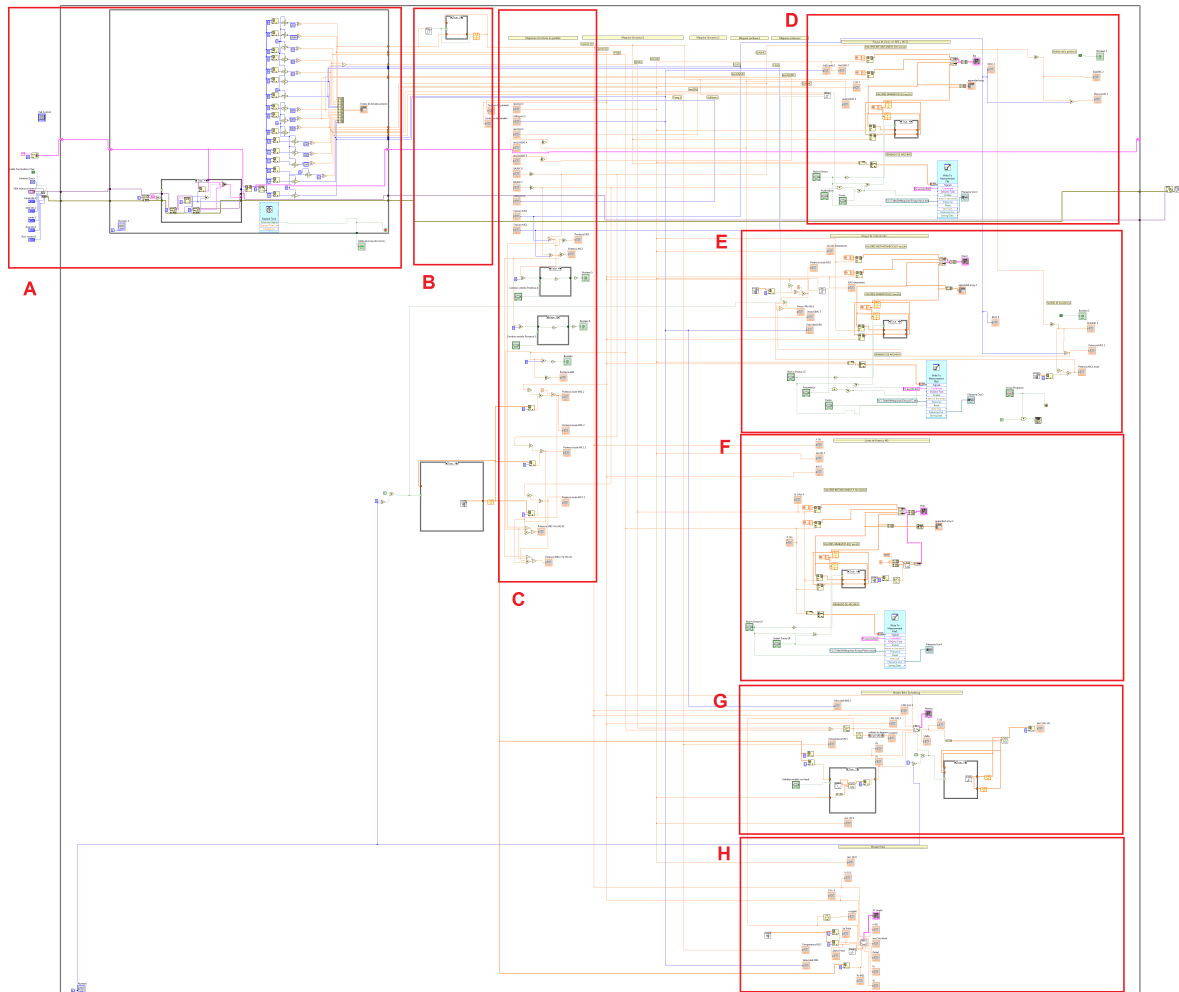


Figura A.1: Mapa del programa principal desarrollado en LabVIEW

En las siguientes secciones se detallan las diferentes rutinas creadas en el programa principal y los distintos subVI necesarios para el correcto funcionamiento del mismo.

## A.1. Mapa del Software Principal

### A.1.1. Driver Arduino para LabVIEW

El driver Arduino, como se menciona en capítulos anteriores, se encarga de recibir y acondicionar los datos que obtiene del puerto USB conectado al equipo.

En la figura A.2 se puede observar la rutina correspondiente al driver.

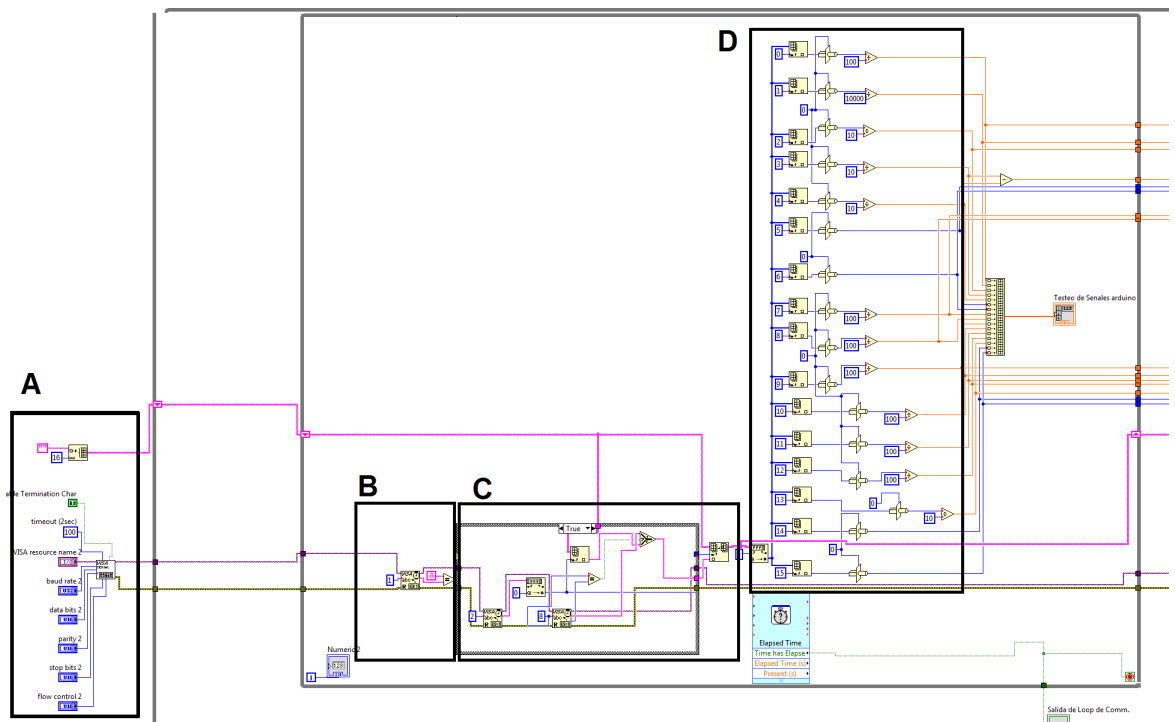


Figura A.2: Rutina de Driver Arduino para LabVIEW

Los bloques de esta rutina operan de la siguiente manera:

- Bloque A: Seteo de parámetros de comunicación serial, los mismos deben coincidir con los parámetros del firmware de Arduino para que funcione la comunicación correctamente
- Bloque B: Condición de inicio de un paquete de datos (valor medido). Cada medida es recibida como se explica en el Capítulo 3, por lo que una vez leído un carácter de fin de línea es que se comienza a realizar el procesamiento de los datos.
- Bloque C: Primero se leen los dos primeros bytes seriales, los cuales corresponden al dígito verificador, este se convierte a valor entero. Luego se lee el resto de los 8 bytes (se filtran los datos que no sean de exactamente 8 bytes), y se convierten en entero y se colocan en un arreglo, en la posición que fue dada por su dígito identificador.
- Bloque D: Se descompone el arreglo de datos, y se multiplican o dividen por constantes para obtenerse los valores en las unidades correctas.

## Apéndice A. Desarrollo de programación LabVIEW

### A.1.2. Filtrado de señal

La ubicación de esta rutina en el mapa del programa principal es el bloque B de figura A.1.

El bloque de filtrado de señal mostrado en figura A.3 con mas detalle, trabaja con un case, se crea un SubVI Ruido mostrado en figura A.4 (el archivo de este subVI es 'HisteresisSenalRuido (SubVI).vi') el mismo compara el valor de la señal que tenía en el ciclo de programa anterior y el nuevo valor, si se está fuera de los límites establecidos pondrá el nuevo valor sino seguirá con el valor anterior. Los límites establecidos se encuentran en el archivo 'LimHistSenal.txt' y son cargadas con el subVI 'LeerArchivoLimHistSeñal(SubVI).vi' (ver A.1.9).



Figura A.3: A.3a SubVI Ruido devuelve True A.3b SubVI Ruido devuelve Falso

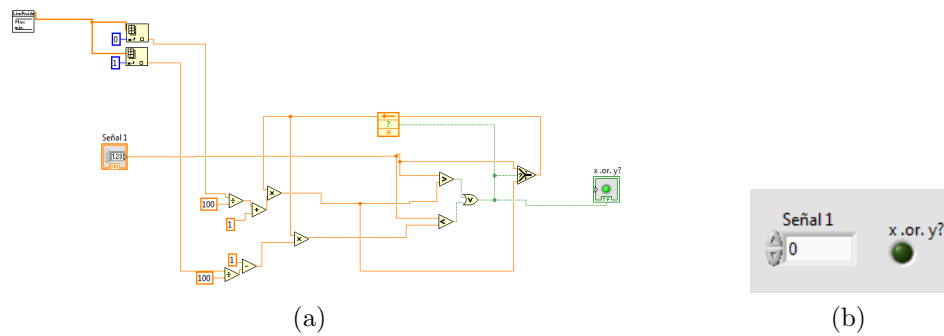


Figura A.4: A.4a SubVI Ruido A.4b SubVI Ruido panel frontal

### A.1.3. Desarrollo para el ensayo de Máquinas en Paralelo

La ubicación de esta rutina en el mapa del programa principal es el bloque C de figura A.1. La misma se muestra con mayor detalle en las figuras A.5 y A.6.

El desarrollo se realizó con las variables a tiempo real, con el especial cuidado en el indicado del sentido de la potencia que se logró con el bloque C.2.1 mostrado en figura A.6. Este bloque se realizó dándole una propiedad booleana a un objeto imagen (llamado objeto control), siendo en este caso la flecha que indica el sentido de la potencia. También un bloque de este tipo se encuentra en la figura A.5.

Para obtener las constantes de la máquina sincrónica, se utilizó el subVI que carga los datos desde los archivos .txt correspondientes. Para tal caso estas constantes se encuentran en el archivo 'Cte\_MS\_MCC.txt' y son cargadas con el subVI 'LeerArchivo cte MS MCC TXT (SubVI).vi' (ver A.1.9). Se le agregó un 'case' para asegurar que cargue los valores una única vez.

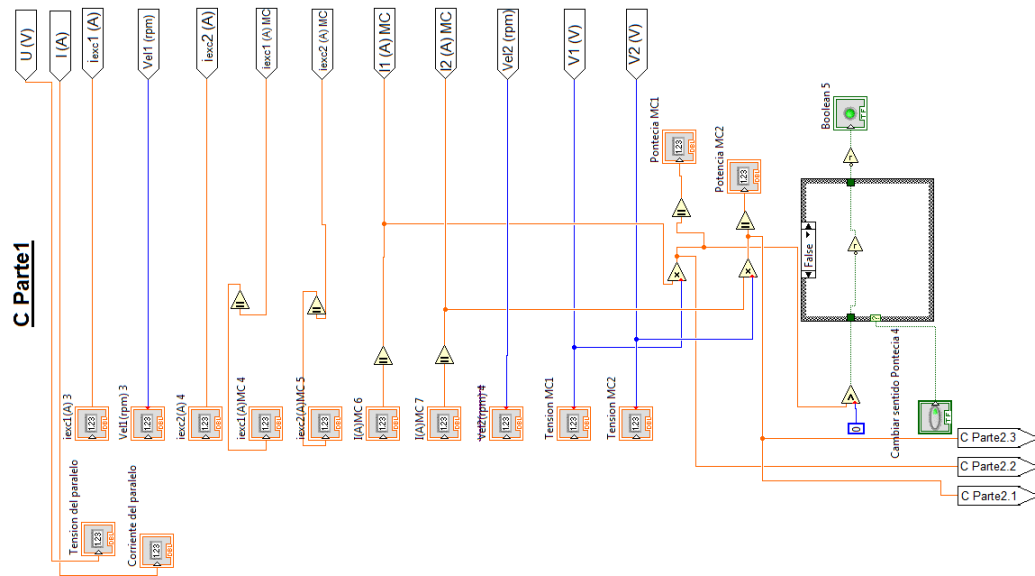


Figura A.5: C Parte 1: Rutina en programa principal para el estudio de las máquinas sincrónicas trabajando en paralelo

## Apéndice A. Desarrollo de programación LabVIEW

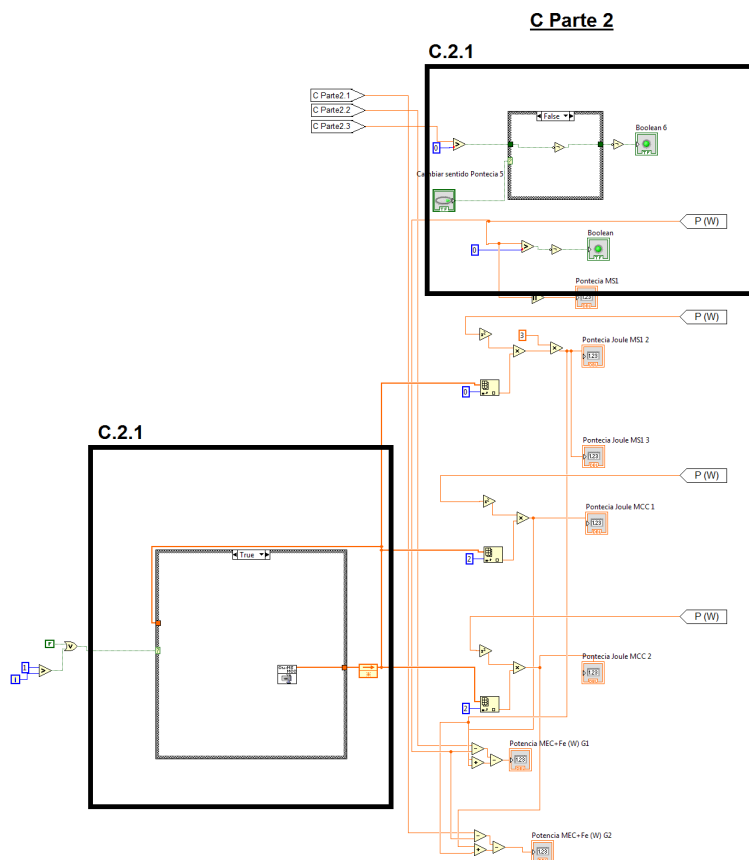


Figura A.6: C Parte 2: Rutina en programa principal para el estudio de las máquinas sincrónicas trabajando en paralelo

### A.1.4. Desarrollo Ensayo de Vacío

La ubicación de esta rutina en el mapa del programa principal es el bloque D de figura A.1. La misma se muestra con mayor detalle en las figuras A.7.

Se analiza la programación de esta rutina haciendo referencia a los bloques de la figura A.7. Se puede apreciar cuatro bloques diferenciados donde: el bloque A, prepara los puntos de interés  $U, i_{exc}$  para ser graficados en un diagrama XY, estos puntos graficados serán los instantáneos y los que se guardaron por interés del usuario que estará realizando el ensayo; el bloque B, guarda en un arreglo los puntos que le interesen al usuario que está haciendo el ensayo y desde este bloque se obtienen los puntos para dejarlos graficados en bloque A (puntos de interés); el bloque C, guarda los puntos con un bloque propio de LabVIEW, este es 'Write To Measurement File', estos datos elegidos para guardar son los mismos que se guardan en el bloque B. También se presenta en este bloque C la lógica para realizar un nuevo ensayo borrando todos los puntos obtenidos hasta ese momento en las variables y creando un nuevo archivo correspondiente al nuevo ensayo. Por último el bloque D, despliega los datos de potencia, tensión y corriente de la MCC.



## A.1. Mapa del Software Principal

La ventana de histéresis se carga con el subVI 'LeerArchivoLimHistSeñal(SubVI).vi' que tiene la misma programación del subVI presentado en la sección de filtrado de señal A.1.2, la programación se puede ver con detalle en figura A.4

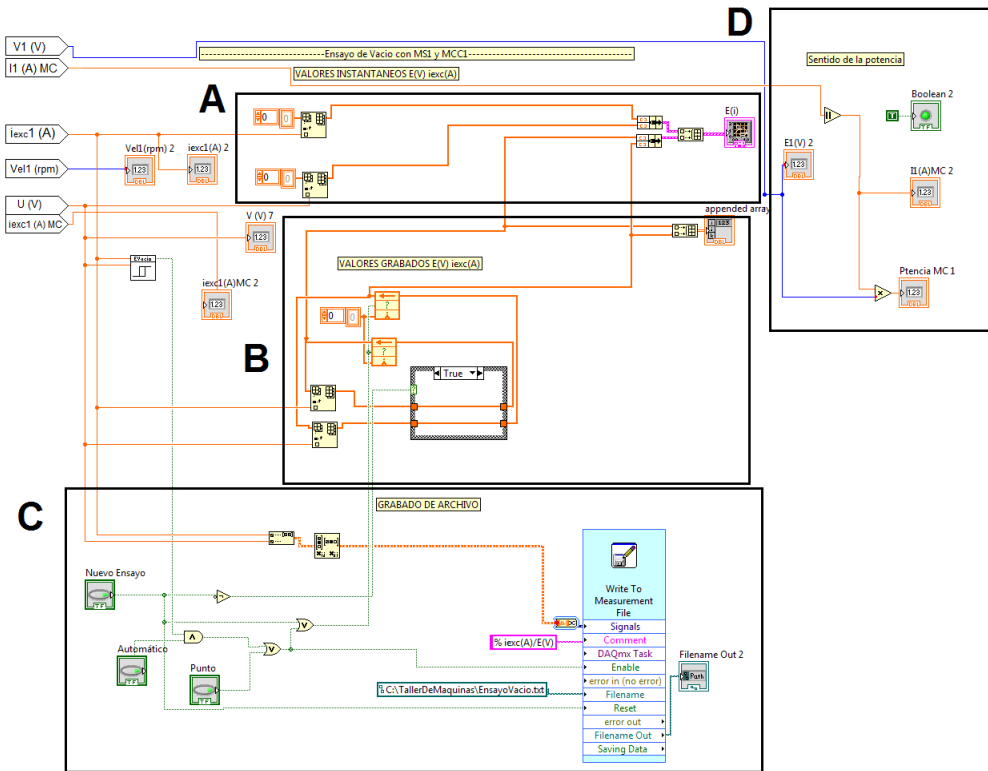


Figura A.7: Rutina en software principal para ensayo de vacío máquina sincrónica

## Apéndice A. Desarrollo de programación LabVIEW

### A.1.5. Desarrollo Ensayo de Cortocircuito

La ubicación de esta rutina en el mapa del programa principal es el bloque E de figura A.1. La misma se muestra con mayor detalle en las figuras A.8.

Se analiza la programación de esta rutina haciendo referencia a los bloques de la figura A.8. Se puede apreciar cinco bloques diferenciados donde: el bloque A, despliega las variables en panel frontal; el bloque B, prepara los puntos de interés  $I$ ,  $i_{exc}$  para ser graficados en un diagrama XY, estos puntos graficados serán los instantáneos y los que se guardaron por interés del usuario que estará realizando el ensayo; el bloque C, guarda en un arreglo los puntos que le interesen al usuario que está haciendo el ensayo y desde este bloque se obtienen los puntos para dejarlos graficados en bloque B (puntos de interés); bloque D, guarda los puntos con un bloque propio de LabVIEW, este es 'Write To Measurement File', estos datos elegidos para guardar son los mismos que se guardan en el bloque C, también se presenta en este bloque D la lógica para realizar un nuevo ensayo borrando todos los puntos obtenidos hasta ese momento en las variables y creando un nuevo archivo correspondiente al nuevo ensayo; el bloque E, realiza los cálculos para el despliegue de las potencias de las distintas máquinas que intervienen en el ensayo.

La ventana de histéresis se carga con el subVI 'LeerArchivoLimHistSeñal(SubVI).vi' que tiene la misma programación del subVI presentado en la sección de filtrado de señal A.1.2, la programación se puede ver con detalle en figura A.4

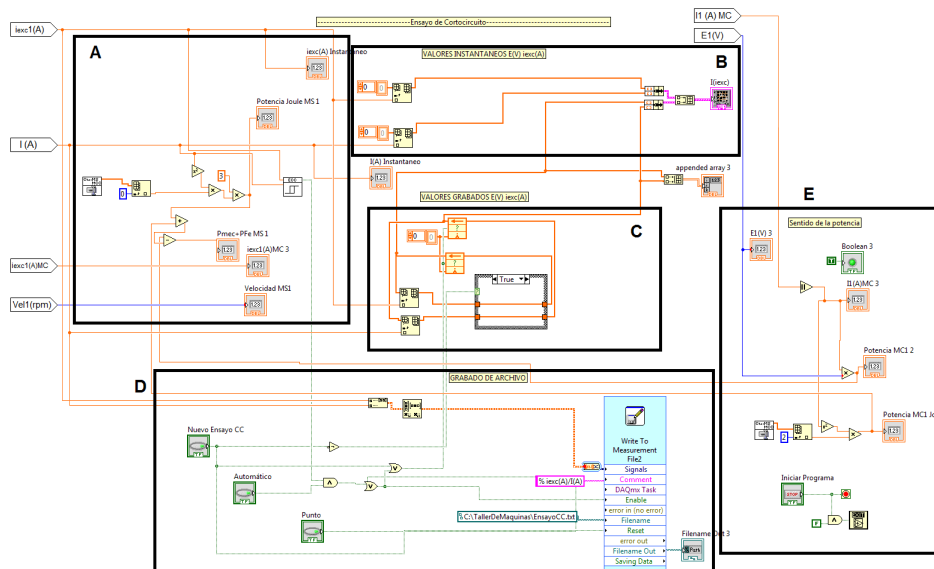


Figura A.8: Rutina en software principal para ensayo de cortocircuito para la máquina sincrónica

### A.1.6. Desarrollo Ensayo de diagrama P-Q de Máquina Sincrónica

La ubicación de esta rutina en el mapa del programa principal es el bloque F de figura A.1. La misma se muestra con mayor detalle en las figuras A.9.

Se analiza la programación de esta rutina haciendo referencia a los bloques de la figura A.9. Se puede apreciar cuatro bloques diferenciados donde: el bloque A, prepara los puntos de interés  $P(W)$ ,  $Q(VAr)$  para ser graficados en un diagrama XY, estos puntos graficados serán los instantáneos y los que se guardaron por interés del usuario que estará realizando el ensayo; el bloque B, guarda en un arreglo los puntos que le interesen al usuario que está haciendo el ensayo y desde este bloque se obtienen los puntos para dejarlos graficados en bloque A (puntos de interés); bloque C, guarda los puntos con un bloque propio de LabVIEW, este es 'Write To Measurement File', estos datos elegidos para guardar son los mismos que se guardan en el bloque C, también se presenta en este bloque C la lógica para realizar un nuevo ensayo borrando todos los puntos obtenidos hasta ese momento en las variables y creando un nuevo archivo correspondiente al nuevo ensayo; bloque D, Realiza los cálculos y prepara los valores para desplegar en gráfico XY el  $\cos(\phi)$  nominal de la máquina.

## Apéndice A. Desarrollo de programación LabVIEW

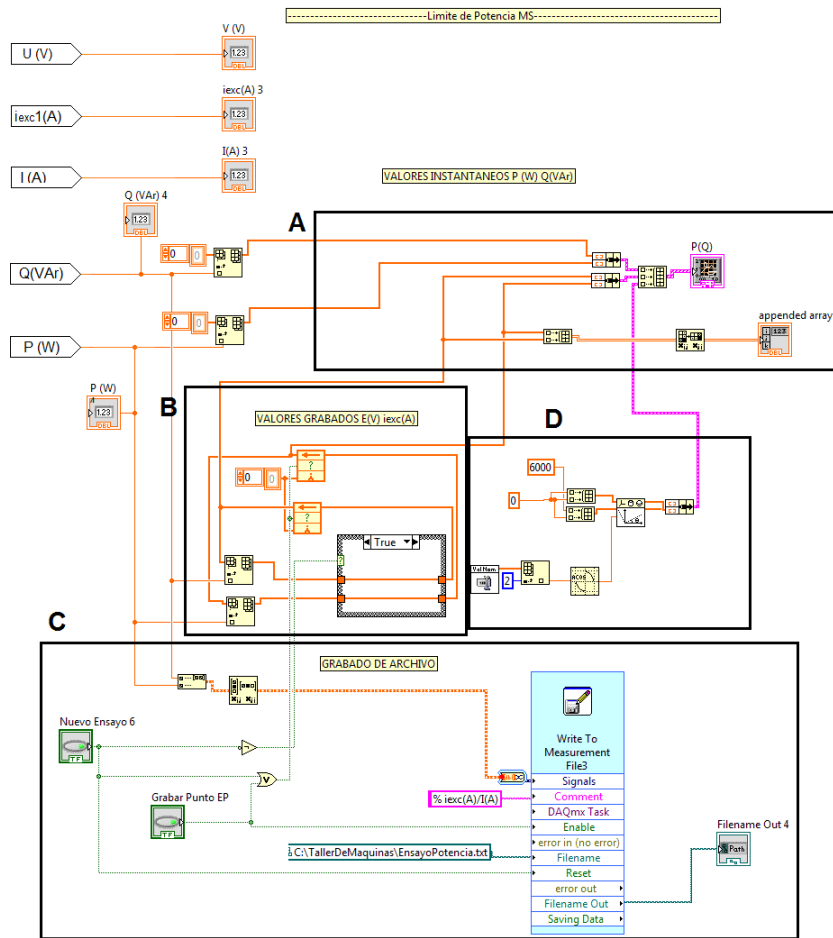


Figura A.9: Rutina en software principal para ensayo de diagrama P-Q de la máquina síncrona

### A.1.7. Desarrollo Modelo de Potier

El software creado en LabVIEW para obtener el modelo de Potier, se realizó con el método de cálculo (ver Apéndice B.2).

El método de cálculo se realiza a partir de las constantes de Potier ( $\alpha$  y  $X_p$ ) y la curva de ensayo de vacío de la máquina en cuestión  $E(i_r)$ , los cálculos a realizar son los siguientes con el modelo de la máquina presentado en figura A.10a

- De los datos relevados por el sistema creado se obtienen los datos  $I_s$ ,  $i_{exc}$  Relevada,  $V_s$ ,  $\cos(\phi)$  de la máquina síncrona.
- A partir de las señales relevadas se calcula el fasor  $\vec{E}_r = jX_p\vec{I} + R_s\vec{I}_s + \vec{V}_s$  donde se obtiene el modulo  $E_r$  y  $arg(\vec{E}_r) = \delta$  siendo  $\delta$  el ángulo interno de la máquina.
- Obtenido los fasores ( $\vec{E}_r$ ,  $R\vec{I}_s$ ,  $jX_p\vec{I}_s$ ) se grafican junto a los fasores  $\vec{V}_s$  e  $\vec{I}_s$

## A.1. Mapa del Software Principal

- De la curva de vacío dada, se interpolan los valores entrando con el valor  $E_r$ , teniendo así  $i_r$  que es la corriente de excitación cuando la máquina está operando en vacío.
- Luego se grafica en el diagrama fasorial el  $i_r$  calculado con un argumento igual a  $\delta + 90^\circ$  ver figura A.10b.
- Por último la corriente de excitación dada por el modelo es  $i_{exc}^- = \bar{i}_r - \alpha \bar{I}$  que será comparada con la relevada por el sistema diseñado ( $i_{exc} \text{ Relevada}$ ), y para completar el diagrama fasorial se ubican los fasores  $\bar{i}_r$ ,  $\alpha \bar{I}_s$  e  $i_{exc}^- = \bar{i}_r - \alpha \bar{I}_s$  quedando el mismo como se muestra en figura A.10b.

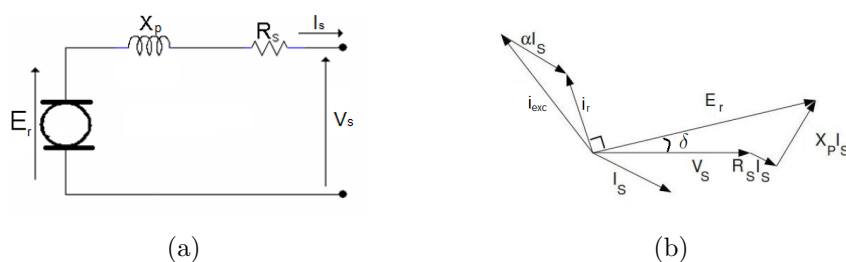


Figura A.10: A.10a Modelo Máquina sincrónica A.10b Diagrama fasorial modelo Potier

En la rutina creada para tal modelo en el programa principal intervienen distintos subVI, siendo mas importante el desarrollado para este modelo en particular "Potier (SubVI).vi". La ubicación de esta rutina en el mapa del programa principal es el bloque H de figura A.1

En el programa principal se tiene la rutina que se muestra en la figura A.11 con mas detalles, allí se observan las distintas señales y constantes que se le asignan al subVI principal de Potier que es necesario para su correcto funcionamiento.

Para obtener las constantes para el subVI principal de Potier, se utilizan los subVI que cargan los datos desde los archivos .txt correspondientes ver A.1.9.

Para este subVI los archivos necesarios son, constantes de Potier ( $X_p; \alpha$ ) que se encuentran en el archivo 'CtePotier Alpha\_ Xp.txt' y son cargadas con el subVI 'LeerArchivo cte Potier TXT (SubVI).vi'; curva de ensayo de vacío de la máquina sincrónica que se encuentran en el archivo 'CurvaVacíoE\_iexc.txt' y son cargadas con el subVI 'LeerArchivoCurvaVacíoE\_iexc (SubVI).vi'; constantes de la máquina sincrónica ( $R_s$ ) que se encuentran en el archivo 'Cte\_MS\_MCC.txt' y son cargadas con el subVI 'LeerArchivo cte MS MCC TXT (SubVI).vi'.

El subVI principal de Potier se muestra el modulo en figura A.12 y su rutina en la figura A.13. Donde las entradas al subVI son,  $R_s$ ,  $U$ ,  $I$ ,  $\phi$ ,  $X_p$ ,  $\alpha$ ,  $E$ ,  $i_{exc}$ ,  $i_{exc} \text{ vacio}$  y las salidas son, bus para gráfico XY,  $i_r$ ,  $i_{exc}$  modulo y ángulo,  $\delta$ ,  $E_r$  y  $\alpha I$  con notación de las referencias del modelo según figura A.10.

## Apéndice A. Desarrollo de programación LabVIEW

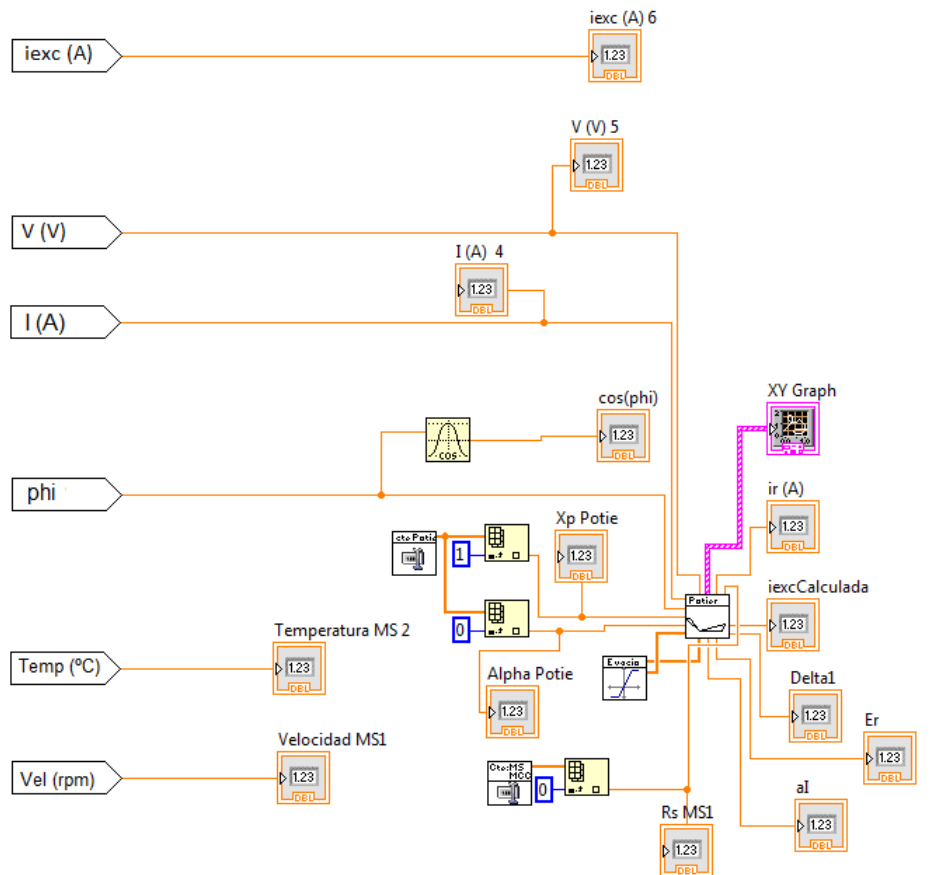


Figura A.11: Rutina del model de Potier en programa principal

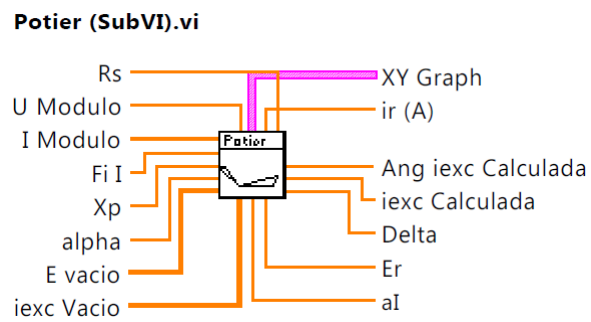


Figura A.12: Modulo SubVI Potier

## A.1. Mapa del Software Principal

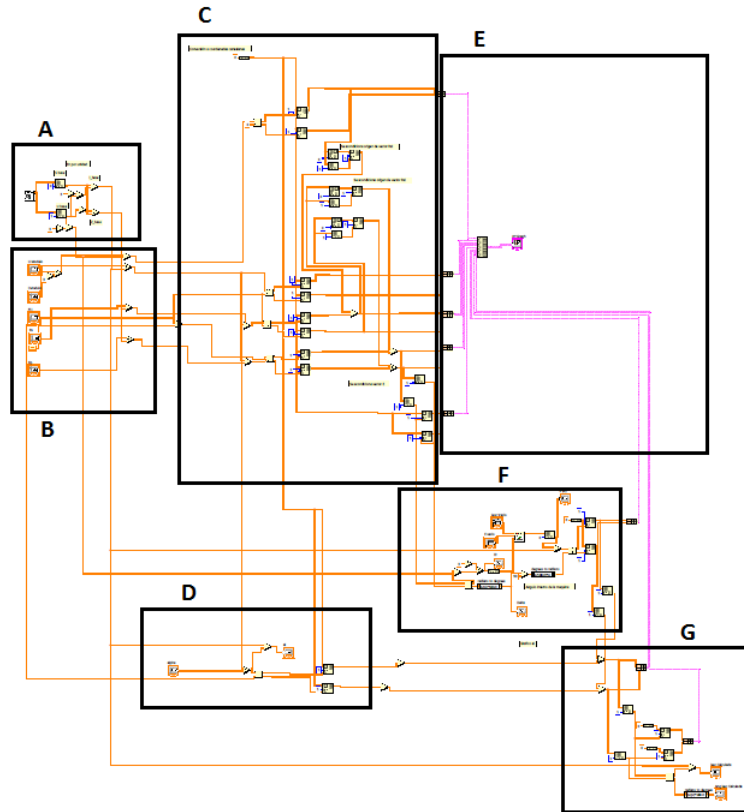


Figura A.13: Rutina subVI Potier

Se analiza el módulo del subVI Potier utilizando las referencias mostradas en figura A.13.

Bloque A y B (ver figura A.14) se setean las entradas del subVI y se calculan los valores bases para poder graficar los fasores de las magnitudes en 'por unidad' tomando para su cálculo los datos nominales de la máquina, estos son  $S = 3505 \text{ VA}$  y  $U = 220 \text{ V}$ .

Bloque C y E (ver figura A.15): en bloque C se acondicionan las señales de forma de obtener los vectores que corresponden a los fasores del diagrama fasorial del modelo de Potier y en el bloque E se pasa los vectores en un arreglo de 'cluster' para ser graficados en un gráfico XY de LabVIEW.

## Apéndice A. Desarrollo de programación LabVIEW

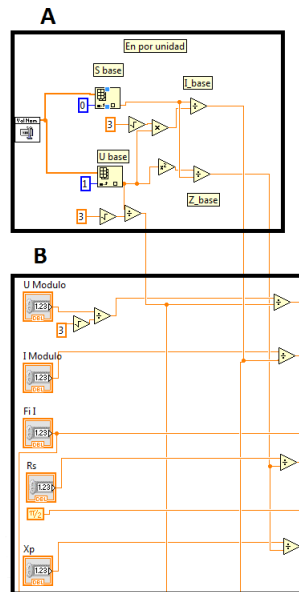


Figura A.14: Señales necesarias para el modelo de Potier y cálculo de las bases para obtener las magnitudes en 'por unidad'

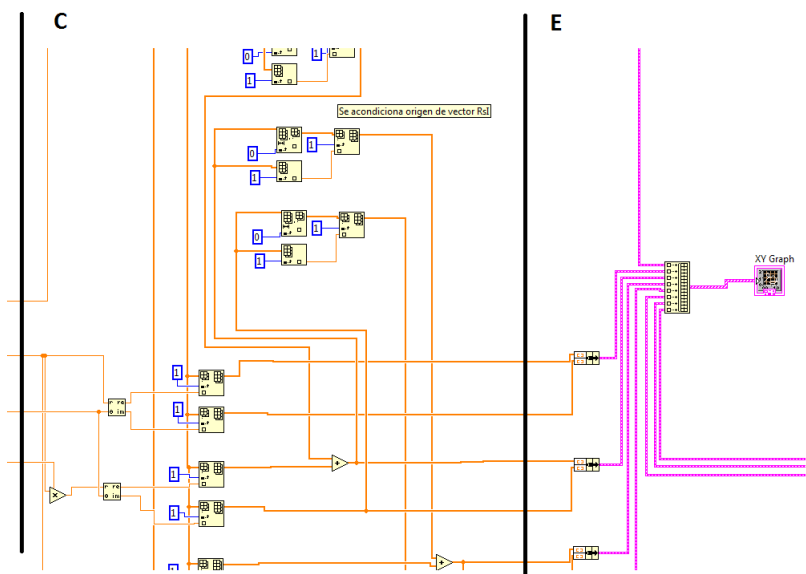


Figura A.15: Sector C: creación de vectores. Sector E: acondicionamiento de los vectores para ser graficados

Bloque F (ver figura A.16): en bloque F1 se prepara  $E_r$  y con el valor se interpola en la curva de ensayo de vacío de la máquina para obtener el valor de  $i_{exc}$  correspondiente a la máquina en vacío, en el bloque F3 se calcula el ángulo interno de la máquina y por último en bloque F2 se preparan los vectores con sus ángulos correspondientes, se los deja en un 'cluster' para graficarlos en el diagrama



## A.1. Mapa del Software Principal

fasorial.

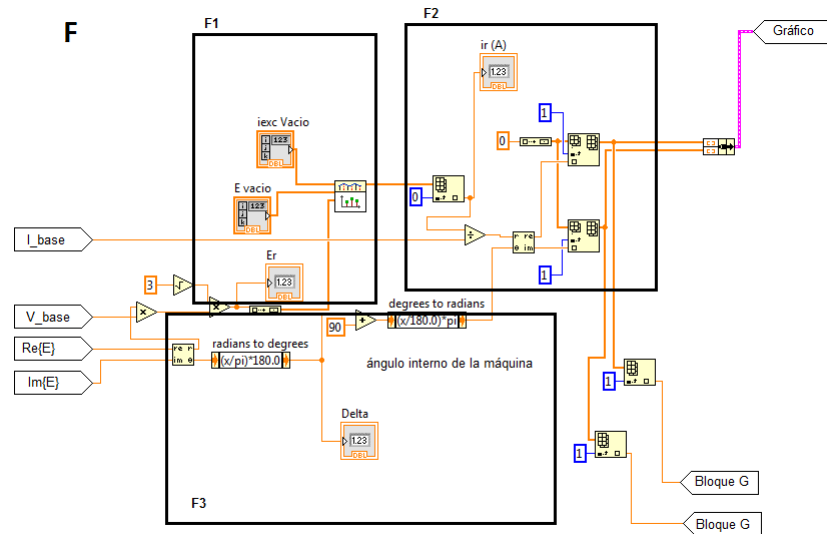


Figura A.16: Sector F: Cálculo de  $i_{exc}$  a partir de curva ensayo de vacío y prepara los fasores para ser graficados

Bloque D(ver figura A.17): en este bloque se prepara el fasor  $\alpha * I$  para su posterior gráfico fasorial, esto se muestra en bloque F ver figura A.16

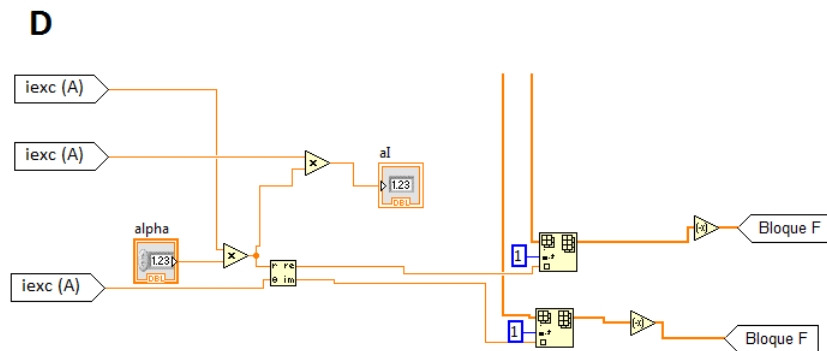


Figura A.17: Sector D: Cálculo  $\alpha * I$  y se prepara fasor

El software desarrollado para el modelo de Potier se obtiene un panel frontal de subVI como el que se presenta en figura A.18

## Apéndice A. Desarrollo de programación LabVIEW

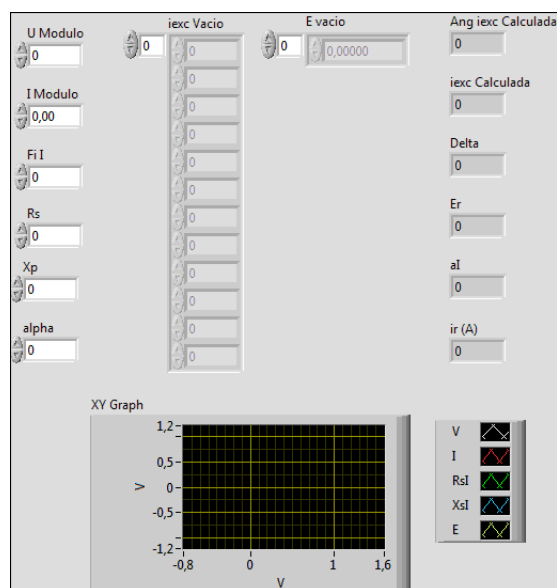


Figura A.18: Panel Frontal subVI Potier

### A.1.8. Desarrollo Modelo Behn-Eschenburg

El software creado en LabVIEW para obtener el modelo de Behen-Eschenburg, es realizado con el marco teórico que se encuentra en Apéndice (B.1).

En la rutina creada para este modelo en el programa principal intervienen distintos subVI, siendo el mas importante el desarrollado para este modelo en particular 'ModeloBehenEschenburg.vi'. La ubicación de esta rutina en el mapa del programa principal es el bloque G de figura A.1.

En el programa principal se tiene la rutina que se muestra en la figura A.19, allí se observan las distintas señales y constantes que se le asignan al subVI principal de Behen-Eschenburg que es necesario para su correcto funcionamiento.

Para obtener las constantes para el subVI principal de Behen-Eschenburg, se utilizan los subVI que cargan los datos desde los archivos .txt correspondientes (ver A.1.9).

Para la lectura de archivos para este subVI son de necesidad, la curva de ensayo de vacío de la máquina sincrónica que se encuentran en el archivo 'Curva-VacioE.iexc.txt' y son cargadas con el subVI 'LeerArchivoCurvaVacioE.iexc (SubVI).vi'; constantes de la máquina sincrónica ( $R_s$ ,  $X_s$ ) que se encuentran en el archivo 'Cte\_MS\_MCC.txt' y son cargadas con el subVI 'LeerArchivo cte MS MCC TXT (SubVI).vi' (ver... subVI); valor de  $X_s$  no lineal dependiendo de la  $i_{exc}$  relevado, para esto se carga la curva  $X_s(i_{exc})$  archivo que se encuentra en 'CurvaXs\_iexc.txt' y se carga con el subVI 'LeerArchivoCurvaXsNoLinBETXT (SubVI).vi'. con esto permitirá cambiar el modelo de la máquina de zona lineal a saturada y viceversa.

## A.1. Mapa del Software Principal

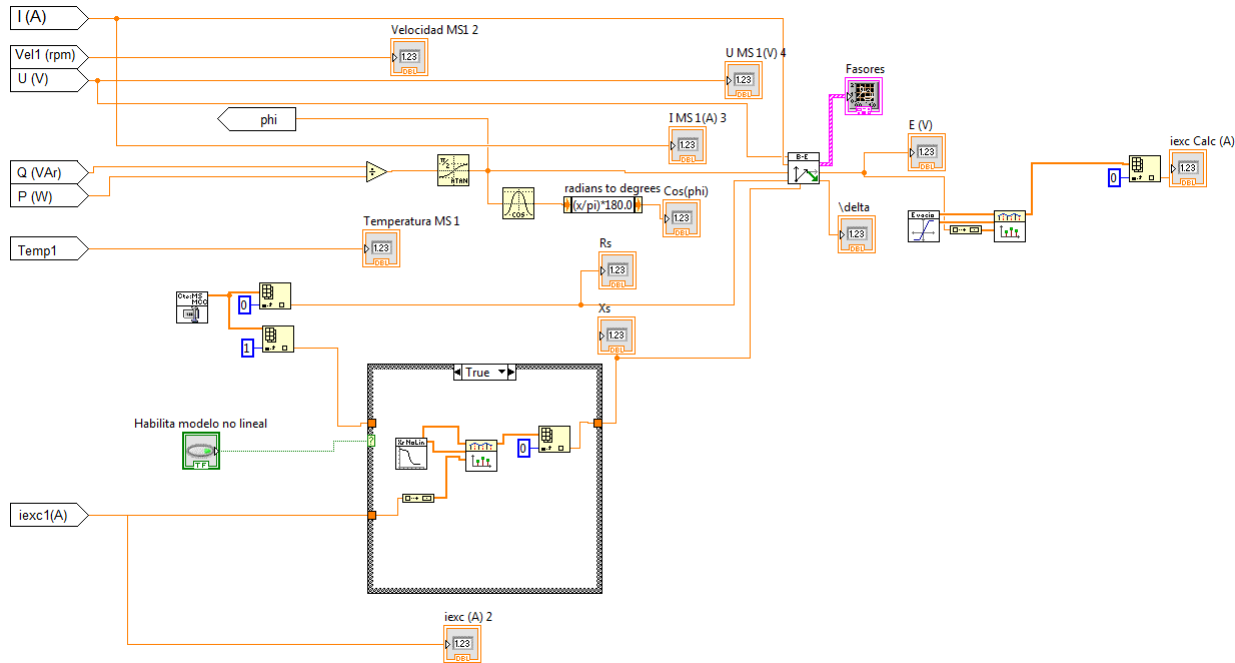


Figura A.19: Rutina del modelo de Behen-Eschenburg en programa principal

El subVI principal de Behen-Eschenburg se muestra el modulo en figura A.20 y su rutina en la figura A.21. Donde las entradas al subVI son,  $U$ ,  $I$ ,  $\phi$ ,  $R_s$ ,  $X_s$  y las salidas son, bus para gráfico XY,  $E_r$  y  $\delta$  con notación de las referencias del modelo según modelo de Behen Eschenburg (ver Apéndice B.1).

### ModeloBehenEschenburg.vi

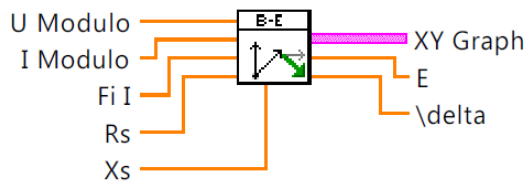


Figura A.20: Modulo SubVI Behen-Eschenburg

## Apéndice A. Desarrollo de programación LabVIEW

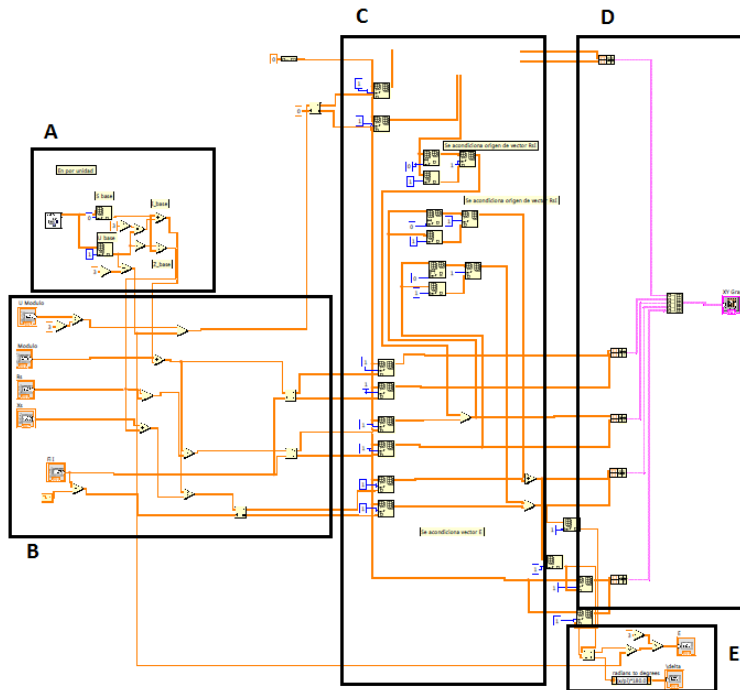


Figura A.21: Rutina SubVI para modelo de Behen-Eschenburg

Se analiza el módulo del subVI Behen-Eschenburg utilizando las referencias mostradas en figura A.21.

Bloque A y B (ver figura A.22) se setean las entradas del subVI y se calculan los valores bases para poder graficar los fasores de las magnitudes en 'por unidad' tomando para su cálculo los datos nominales de la máquina, estos son  $S = 3505 \text{ VA}$  y  $U = 220 \text{ V}$ .

Bloque C y D (ver figura A.23): en bloque C se acondicionan las señales de forma de obtener los vectores que corresponden a los fasores del diagrama fasorial del modelo de Potier y en el bloque D se pasa los vectores en un arreglo de 'cluster' para ser graficados en un gráfico XY de LabVIEW.

## A.1. Mapa del Software Principal

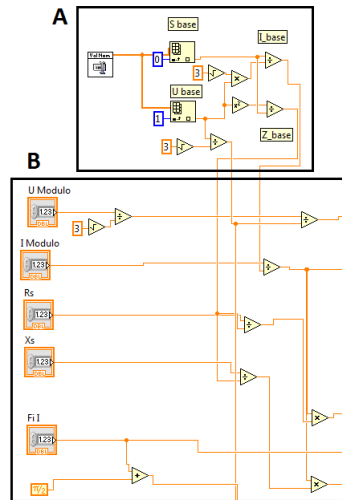


Figura A.22: Señales necesarias para el modelo de Behen-Eschenburg y cálculo de las bases para obtener las magnitudes en 'por unidad'

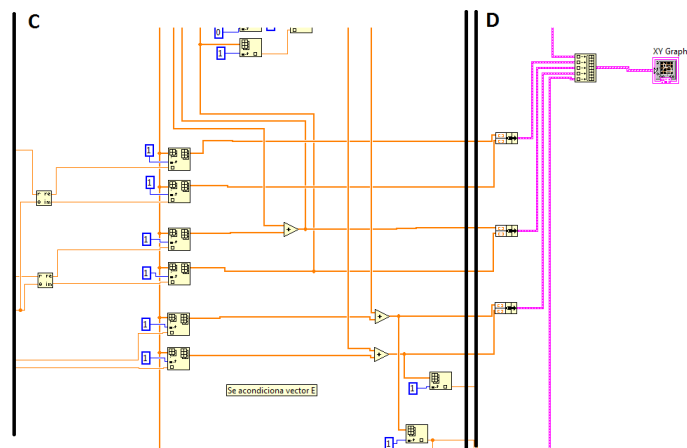


Figura A.23: Sector C: creación de vectores. Sector D: acondicionamiento de los vectores para ser graficados

Bloque E(ver figura A.24): en este bloque se preparan las señales de salida para el subVI en cuestión.

Para el software desarrollado para el modelo de Behen-Eschenburg se obtiene un panel frontal del subVI como el que se presenta en figura A.25

### A.1.9. Desarrollo de SubVI para la lectura de archivos de textos

Se programó una rutina para levantar las constantes necesarias para cada parte del programa desde un archivo de texto, se realiza de esta manera para obtener un dominio sobre las constantes luego de realizada la aplicación.

Los subVI desarrollados para tal fin son los siguientes:

## Apéndice A. Desarrollo de programación LabVIEW

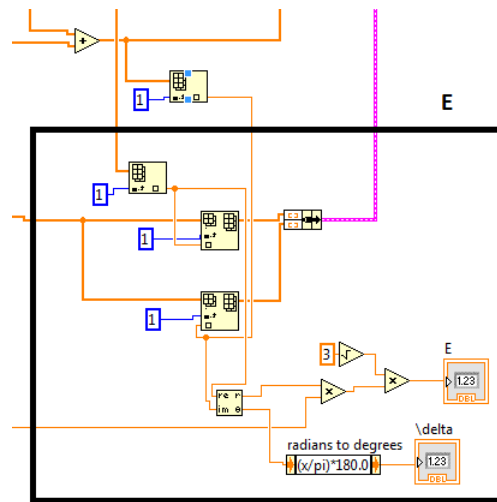


Figura A.24: Sector E: Señales de salida del subVI Behen-Eschenburg

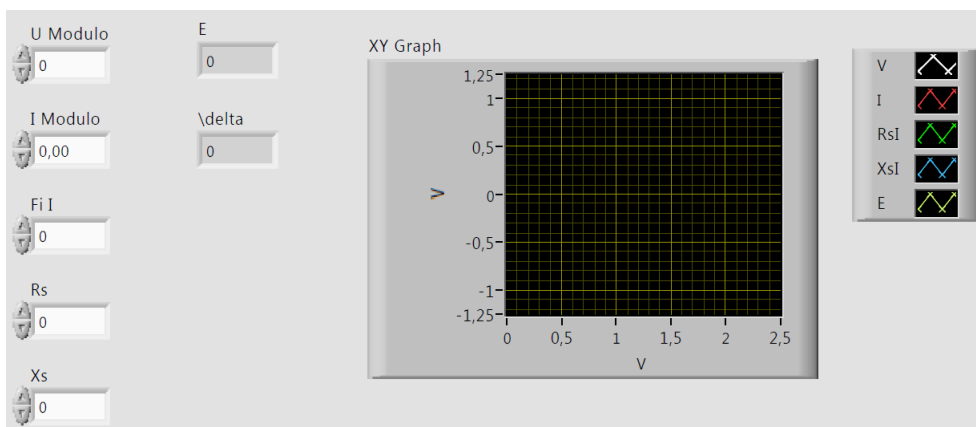


Figura A.25: Panel Frontal subVI Behen-Eschenburg

- LeerArchivo cte MS MCC TXT (SubVI).vi
- LeerArchivo cte Potier TXT (SubVI).vi
- LeerArchivoCurvaVacioE\_iexc (SubVI).vi
- LeerArchivoCurvaXsNoLinBETXT (SubVI).vi
- LeerArchivoLimHistGrabadoAutoECC (SubVI).vi
- LeerArchivoLimHistGrabadoAutoEVacio (SubVI).vi
- LeerArchivoLimHistSeñal(SubVI).vi
- LeerArchivoPorUnidadTXT (SubVI).vi

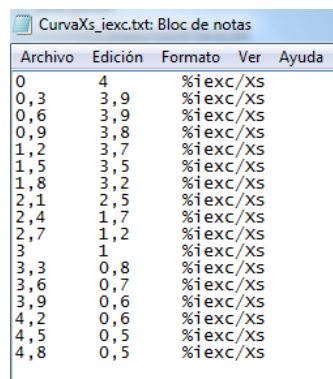
## A.1. Mapa del Software Principal

Se analiza la programación del SubVI 'LeerArchivoCurvaXsNoLinBETXT (SubVI).vi' como ejemplo pero todos los subVI presentados se programan de igual forma.

El modulo creado se muestra en figura A.27, donde las salidas son, dos arreglos con los datos numéricos del archivo de datos que levanta según corresponda, por otro lado su rutina se muestra en figura A.28.

Se analiza la rutina del SubVI según referencias de la figura A.28. El bloque A levanta el archivo .txt desde la ubicación que se le de, en este caso se programó para levantar los datos desde el disco C:\ConstantesCurvas modelo BE, el levantado de datos lo hace en forma de string, para convertirlo en número se hace un loop que se muestra en el bloque B, donde se convierte el string en número. La programación mostrada en el bloque B es un loop que guarda en un array los datos o número que se convierte de un string, la conversión se realiza desde los bloques propios del LabVIEW teniendo especial cuidado para los números que presentan decimal.

Los archivos de texto para ser leído tienen que tener el formato mostrado en figura A.26, donde tendrán a lo sumo dos columnas que serán las salidas del módulo del subVI mostradas en la figura A.27 y las columnas serán separadas por un símbolo de tabulado y los comentarios deben de comenzar con el signo de %.



Archivo	Edición	Formato	Ver	Ayuda
0	4	%iexc/Xs		
0,3	3,9	%iexc/Xs		
0,6	3,9	%iexc/Xs		
0,9	3,8	%iexc/Xs		
1,2	3,7	%iexc/Xs		
1,5	3,5	%iexc/Xs		
1,8	3,2	%iexc/Xs		
2,1	2,5	%iexc/Xs		
2,4	1,7	%iexc/Xs		
2,7	1,2	%iexc/Xs		
3	1	%iexc/Xs		
3,3	0,8	%iexc/Xs		
3,6	0,7	%iexc/Xs		
3,9	0,6	%iexc/Xs		
4,2	0,6	%iexc/Xs		
4,5	0,5	%iexc/Xs		
4,8	0,5	%iexc/Xs		

Figura A.26: Formato de los archivos de datos cte (archivos de textos)



Figura A.27: Módulo leer archivo módulo





# Apéndice B

## Fundamentos teóricos

En esta sección se describen los modelos de máquinas sincrónicas trabajados.

### B.1. Modelo de Behen-Eschenburg

Este modelo se utiliza para modelar la máquina sincrónica de polos lisos funcionando en régimen lineal (es decir, sin entrar en la zona de saturación).

Debido al giro del rotor respecto del estator, se producen inductancias mutuas entre los bobinados rotórico y estatóricos debido a las corrientes que circulan por ellos, las cuales dependen de la posición del rotor respecto del estator. Por lo tanto, se puede definir el flujo estatórico en una fase como:

$$\vec{\psi}_s = M_{rs}\vec{I}_r + L_s\vec{I}_s \quad (\text{B.1})$$

Siendo:

- $M_{rs}$ : Inductancia mutua rotor-estator
- $L_s$ : Inductancia cíclica directa

Los flujos correspondientes a las restantes fases del estator están desfasados  $\pm 120^\circ$  respecto del descrito en ec B.3 .

El circuito monofásico equivalente se puede observar en la figura B.1, siendo  $R_s$  la resistencia del bobinado estatórico y  $V_s$  la tensión en bornes del estator, siendo ésta generada debido a la corriente rotórica.

$$v_s(t) = e_s(t) - R_s i_s(t) \quad (\text{B.2})$$

Recordando la ley de Faraday:  $e_s(t) = -\frac{\partial \psi}{\partial t}$  y debido a que las ecuaciones son lineales y las tensiones/corriente sinusoidales, si se conecta una carga lineal en bornes del estator se obtiene en fasores los siguientes resultados:

## Apéndice B. Fundamentos teóricos

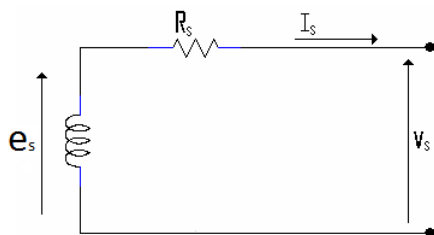


Figura B.1: Modelo monofásico de la máquina sincrónica

$$v_s(t) = e_s(t) - R_s i_s(t) \quad (\text{B.3})$$

$$\Rightarrow \begin{cases} v_s(t) \rightarrow \vec{V}_s \\ e_s(t) \rightarrow \vec{E}_s = -j\omega\vec{\psi}_s \end{cases} \quad (\text{B.4})$$

De la ley de Faraday y ecuaciones B.3) y B.4) se obtiene ec B.5)

$$\vec{V}_s = -j\omega\vec{\psi}_s - R_s\vec{I}_s \quad (\text{B.5})$$

De ecuaciones B.3) y B.5) se obtiene ec B.6), donde las variables son:

- $V_s$ : Tensión en bornes del estator.
- $R_s$ : Resistencia del bobinado estatórico.
- $X_s = \omega L_s$ : Reactancia Sincrónica.
- $E$ : Tensión inducida en el estator debida a la corriente rotórica.

$$\vec{V}_s + R_s\vec{I}_s + jX_s\vec{I}_s = -j\omega M_{rs}\vec{I}_r = \vec{E} \quad (\text{B.6})$$

De este modo, el circuito monofásico equivalente de la Máquina Sincrónica utilizando el Modelo de Behn-Eschenburg queda representado como se presenta en la figura B.2

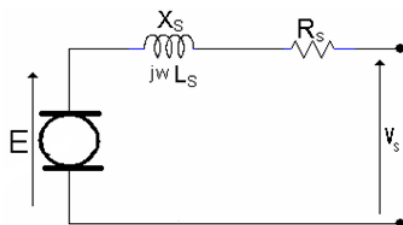


Figura B.2: Modelo Behen-Eschenburg, monofásico de la máquina sincrónica

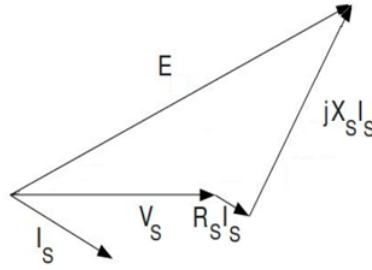


Figura B.3: Diagrama fasorial modelo Behn-Eschenburg

El diagrama fasorial para el modelo de la máquina síncrona representado en la figura B.2) se presenta en la figura B.3.

Para más información véase referencia [3].

## B.2. Modelo de Potier

La Máquina Síncrona al trabajar en zona de saturación, se pierde la linealidad del circuito magnético y por lo tanto, el Modelo de Behn-Eschenburg ya no representa correctamente el funcionamiento de la misma.

Este método no intenta describir el funcionamiento de la máquina en la zona de saturación, sino que permite calcular la corriente de excitación necesaria para que la Máquina Síncrona, de polos lisos y trabajando en la zona de saturación, alimente una carga dada, a tensión y corriente establecidas.

Mientras la MS trabaja en la zona lineal, la reactancia síncrona se mantiene constante (despreciando  $R_s$ ). Cuando la MS entra en zona de saturación, dicha reactancia comienza a decrecer, y se observa que tiende a un valor asintótico al alcanzar altos niveles de saturación. A ese valor asintótico se le llama Reactancia de Potier ( $X_p$ ).

Por lo tanto, cuando la MS trabaja en zona de saturación, se puede plantear la reactancia síncrona como la suma de dos reactancias, una constante (reactancia de Potier) y una variable, cuyo valor depende de la saturación de la máquina.

En figura B.4 se puede observar dicho comportamiento.

De ec. B.6 y separando la reactancia síncrona en dos reactancias, se obtienen la reactancia constante (reactancia de Potier) y la reactancia variable, siendo esta última la que depende de la saturación del hierro. Esto queda representado en las siguientes ecuaciones:

De ec. B.6 se tiene ec. B.7

$$\vec{V}_s + R_s \vec{I}_s = -j\omega M_{rs} \vec{I}_r - j\omega L_s \vec{I}_s \quad (\text{B.7})$$

Separando las reactancias síncronas en ec.B.7)se obtiene ec.B.8):

$$\vec{V}_s + R_s \vec{I}_s = -j\omega M_{rs} \vec{I}_r - jX_p \vec{I}_s - jX_v \vec{I}_s \quad (\text{B.8})$$

## Apéndice B. Fundamentos teóricos

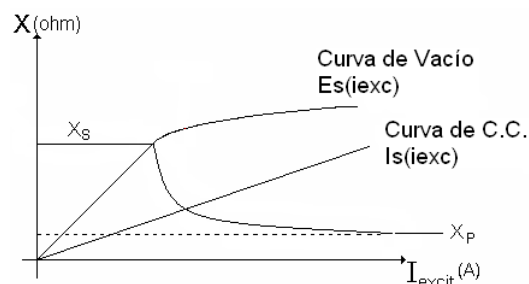


Figura B.4: Comportamiento Reactancia Sincrónica

Por lo que la reactancia sincrónica queda representada en la ec B.9.

$$X_p + X_v = X_s \quad (\text{B.9})$$

Finalmente de las ecuaciones B.8) y B.9) se obtiene ec.B.10). Siendo el primer miembro independiente de la saturación, mientras que el segundo miembro dependiente de la saturación.

$$\vec{V}_s + R_s \vec{I}_s + jX_p \vec{I}_s = -j\omega M_{rs} \vec{I}_r - jX_v \vec{I}_s = \vec{E}_r(\vec{I}_r, \vec{I}_s) \quad (\text{B.10})$$

Se define una corriente  $I_r$ , puesta en el rotor, que produce la misma fuerza magnetomotriz resultante de los efectos de  $I_s$  e  $I_r$ , la misma se representa en ec.B.11, siendo  $\alpha$  una constante ( $\alpha$  de Potier) que se determina mediante ensayos.

$$\vec{I}_{rr} = \vec{I}_r + \alpha \vec{I}_s \quad (\text{B.11})$$

En este caso, el diagrama fasorial de la MS de polos lisos trabajando en la zona de saturación se representa en la figura

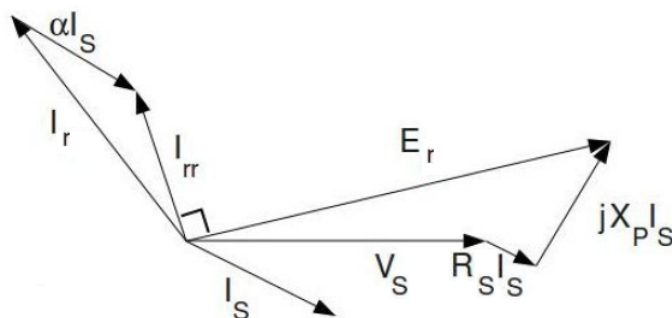


Figura B.5: Diagrama fasorial con la máquina en zona de saturación

Tanto  $\alpha$  como  $X_p$  se hallan realizando los ensayos de vacío, de cortocircuito y el ensayo en devatado.

## Apéndice C

# Práctica de Máquina Síncrona de la asignatura Taller de Máquinas Eléctricas del IIE

El documento que se anexa a continuación corresponde al empleado para realizar los ensayos a las MS que se realizaron en la asignatura hasta el año 2015. En él se describen uno a uno los pasos a seguir para la realización de los ensayos que comprende la práctica.

Se encuentra también una descripción detallada de las máquinas utilizadas para tales ensayos, como también procedimientos y sugerencias para la manipulación de las mismas.



INSTITUTO DE INGENIERÍA ELÉCTRICA  
FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE LA REPÚBLICA



# Taller de Máquinas Eléctricas

Práctica: Alternador  
Instructivo

Curso 2014

# Índice

<b>1. Objetivos de la práctica</b>	<b>3</b>
<b>2. Descripción de los equipos</b>	<b>3</b>
2.1. Máquina a ensayar - Generalidades . . . . .	3
2.2. Motor de arrastre . . . . .	4
2.3. Alternador . . . . .	4
2.3.1. Excitación . . . . .	4
2.3.2. Generación AC . . . . .	4
2.4. Cargas . . . . .	5
2.5. Equipo de sincronización . . . . .	5
<b>3. Ensayos a realizar</b>	<b>7</b>
3.1. Ensayo de vacío . . . . .	7
3.2. Ensayo de cortocircuito . . . . .	7
3.3. Ensayo con carga resistiva . . . . .	7
3.4. Ensayo de devatado . . . . .	7
3.5. Medida de resistencias . . . . .	8
3.6. Puesta en paralelo . . . . .	9
<b>4. Relevamiento previo y realización de ensayos</b>	<b>9</b>
<b>5. Documentación a entregar</b>	<b>9</b>
5.1. Preinforme . . . . .	9
5.2. Informe . . . . .	10
5.2.1. Modelo . . . . .	10
5.2.2. Gráficas . . . . .	10

## 1. Objetivos de la práctica

1. Determinación de las características de un alternador trifásico que permitan predecir sus condiciones de funcionamiento en diferentes regímenes, en base a los modelos de Potier y Behn Eschemburg.
2. Realización de un ensayo bajo carga resistiva.
3. Puesta en paralelo de dos alternadores.

## 2. Descripción de los equipos

### 2.1. Máquina a ensayar - Generalidades

Se trata de una máquina sincrónica proyectada para ser acoplada a un motor (motor de arrastre), produciendo energía eléctrica trifásica hasta una potencia aparente de  $3kVA$ . El fabricante ha dispuesto los bobinados inductores en la carcasa fija de la máquina, de polos salientes, con extremos de bobinados marcados en la bornera de conexiones como  $E$  y  $F$ . El inducido giratorio construido sobre un tambor ranurado de hierro laminado, contiene un doble bobinado:

- Uno trifásico compuesto de bobinados conectados entre sí en estrella y desplazado cada uno  $120^\circ$  eléctricos de las restantes, conectados a la parte fija mediante 4 anillos rozantes colocados en el extremo posterior de la máquina. Los anillos se encuentran accesibles en los bornes  $U, V, W$  y  $O$ , los tres primeros para las tensiones de fase y el último para el neutro de la estrella.
- Un segundo bobinado tipo generador de continua, con su colector en la parte anterior del generador, que por medio de las escobillas permite obtener en bornes  $A$  y  $BH$  una tensión continua capaz de excitar, intercalando medios de regulación, el campo inductor fijo dentro del cual gira, funcionando como un generador autoexcitado. Durante el desarrollo de la práctica, a los efectos de evitar la interacción de estos dos bobinados rotóricos sobre un único campo inductor fijo, prescindiremos del circuito autoexcitado, procediendo a alimentar el campo del alternador con una fuente independiente de baja tensión continua.

Se tiene por tanto una máquina síncrona construida de forma inversa a la comunmente estudiada (inductor con corriente continúa ubicado en el rotor e inducido trfásico en el estator), pero que en la práctica funciona de modo exactamente igual dado que el movimiento relativo entre inductor e inducido es el mismo en ambos casos.

Además del alternador a ensayar, se debe disponer de un motor de arrastre, para lo que se utilizará un motor shunt de corriente continua. El conexionado del motor y el alternador se indica en el esquema simplificado de la figura 1.



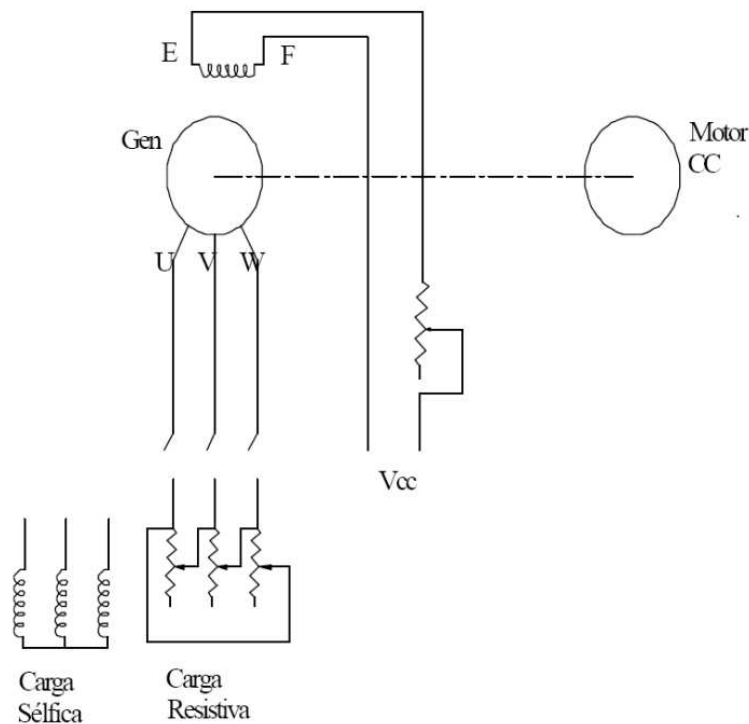


Figura 1: Diagrama simplificado del sistema.

## 2.2. Motor de arrastre

Se utilizará un motor de corriente tipo shunt. Debe siempre verificarse la existencia del campo inductor antes de alimentar corriente al rotor pues en ausencia de campo el motor tiende a velocidad máxima con corrientes muy altas en el rotor y riesgo de destrucción del rotor por fuerza centrífuga.

## 2.3. Alternador

### 2.3.1. Excitación

Debe disponerse un circuito que permita regular y medir la corriente de excitación a partir de una tensión de corriente, alimentando una tensión adecuada en bornes E y F conectados a los bobinados del campo. Verificar la corriente máxima de excitación.

### 2.3.2. Generación AC

La máquina tiene conectados sus bobinados en estrella con bornes U, V y W siendo también accesible el neutro en el borne O que no se utiliza (salvo para la puesta en paralelo). Para conocer el comportamiento del alternador en los diferentes ensayos se deben medir los valores eléctricos generados por lo que debe montarse un circuito de medida de potencia trifásica; tensión, corriente, potencia, frecuencia e indirectamente el  $\cos(\phi)$ . En ensayos en carga, se deberá llegar a la potencia aparente nominal de la máquina (3kVA) lo que dará corrientes de 9,2A (datos en placa) de manera que se deberán verificar los límites nominales de los instrumentos.

La llave trifásica a la salida del equipo de medición, permite cambiar sin tensión los bancos de carga, conectados con llave abierta sin necesidad de detener el grupo generador. Se deberá instalar

un amperímetro a la salida para poder detectar rápidamente corrientes excesivas al conectar la carga de cometerse errores o si se ha equivocado el conexionado, en cuyo caso debe cortarse la excitación inmediatamente (debe preverse un medio para hacerlo).

## 2.4. Cargas

La carga a conectar para obtener la potencia nominal en el ensayo a  $\cos(\phi)$  unitario se compone de resistencias trifásicas conectadas en paralelo. Cuando haya que obtener la carga para el ensayo en devatado, se utilizarán los bancos de 5 bobinados sélficos conectables en serie o paralelo según se operen las llaves de cuchilla correspondientes. Para obtener una intensidad en devatado del orden de  $I_n$  deben conectarse los bancos entre sí en conexión triángulo, pues en la conexión estrella no se obtiene una impedancia suficientemente baja.

## 2.5. Equipo de sincronización

En esta parte de la práctica se dispondrá de un segundo grupo alternador-motor de arrastre, de características idénticas al ya detallado, a fin de realizar la conexión en paralelo de las salidas trifásicas de ambas máquinas síncronas. En el laboratorio se tendrá asimismo un equipo especial para realizar esta conexión, cumpliendo las condiciones necesarias para la misma (las cuales deberán ser investigadas previamente y detalladas en el preinforme). En la figura 2 se muestra el esquema de conexionado del equipo referido.

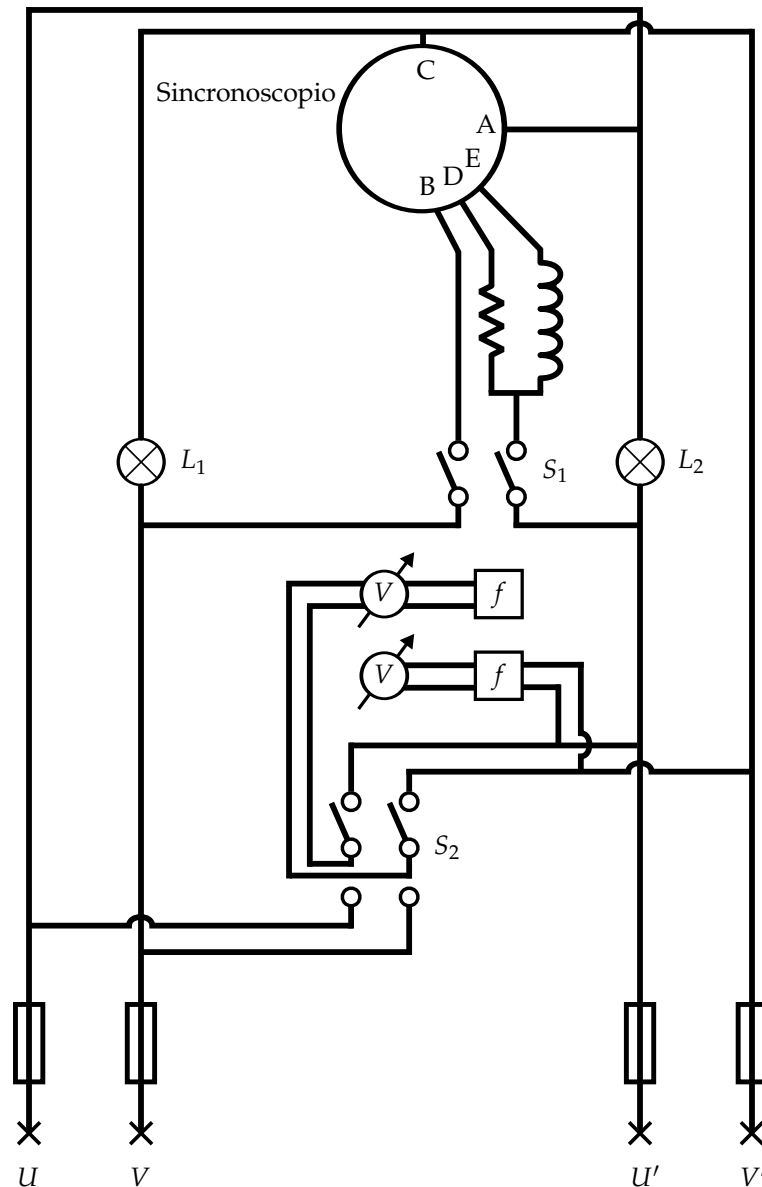


Figura 2: Diagrama del equipo de sincronización.

Se cuenta con dos pares de bornes destinados a dos fases correspondientes en cada máquina (por ejemplo  $U$ ,  $V$  en la máquina 1 y  $U'$ ,  $V'$  en la máquina 2).

El equipo cuenta con medidas de tensión y frecuencia para cada par de bornes, dos lámparas conectadas como se muestra en la figura y un sincronoscopio.

El sincronoscopio es un equipo capaz de mostrar a través del movimiento de una aguja (los hay también formados por *leds*) la relación de fase y frecuencia de dos tensiones sinusoidales. Es un instrumento monofásico en el cual se dispone de una escala circular de 360 grados y una aguja que puede girar libremente sobre la misma. La construcción del equipo implica que en caso que las dos tensiones tengan distinta frecuencia, la aguja girará a una velocidad proporcional a la diferencia entre ambas frecuencias (el sentido de giro indicará qué tensión es más lenta). Por el contrario si se aplican dos tensiones de igual frecuencia, la aguja quedará en una posición estática indicando la fase relativa entre ambas tensiones (siendo la misma el ángulo indicado por la aguja medido desde la posición de "las 12" tal como si se tratase de un reloj).

El equipo de sincronización tiene una llave ( $S_1$ ) para poder desconectar el sincronoscopio de las tensiones a sincronizar, dado que no es conveniente que permanezca conectado de forma continua (utilizándose solo en el momento de realizar el paralelo).

### 3. Ensayos a realizar

Todos a la velocidad nominal de la máquina (ver placa), controlable con lo que indica el frecuencímetro  $50\text{Hz}$  o con un medidor de velocidad de giro que será provisto.

#### 3.1. Ensayo de vacío

Determinación de la fuerza electromotriz de la máquina en función de la corriente de excitación a velocidad nominal. Con la llave de carga abierta se llevará el alternador a  $1500\text{rpm}$  y se actuará sobre el reóstato que regula la corriente de excitación produciendo un aumento gradual hasta alcanzar un valor de tensión de  $230\text{V}$ , descendiendo luego la excitación hasta el mínimo posible, normalizando así el ciclo de histéresis. Se procederá luego a un nuevo ciclo en que se tomarán las lecturas de excitación y tensión en bornes de máquina. Se llevará la tensión hasta  $1,1U_n$ , cuidando al ajustar corriente de excitación, haciéndolo siempre en forma creciente pues si se retrocede al ajustar un punto se pasará de la curva ascendente de imantación del hierro a la descendente con imprecisión en el trazado de la curva. Luego de llegar a  $1,1U_n$  se procederá a obtener los datos de la curva descendente hasta un valor mínimo anotándose los valores de la  $FEM$  en el descenso (los que serán ligeramente superiores a los de igual intensidad de excitación en período ascendente).

#### 3.2. Ensayo de cortocircuito

Se detendrá el grupo. Manteniendo el circuito de excitación del ensayo anterior se desconectará el circuito de medida de potencia en alterna realizándose en bornes del alternador el conexionado entre sí de los bornes, intercalando entre dos bornes un amperímetro con alcance adecuado para la  $I_n$  del generador.

Se pondrá el grupo en marcha a la velocidad nominal y se irá incrementando suavemente la excitación hasta obtener en el amperímetro el valor  $I_n$ . Se pueden tomar valores intermedios menores que  $I_n$  con lo que se podrá verificar que la gráfica en que se representa  $I_{cc}$  en función de  $I_{exc}$  resulta ser una recta que pasa por el origen (no hay saturación en el hierro).

Se anotarán los valores  $I_n$  y los correspondientes a  $I_{exc}$  para cada caso.

#### 3.3. Ensayo con carga resistiva

Sin modificar el circuito utilizado para el ensayo en vacío, con cerrar la llave trifásica quedará preparado el circuito para el ensayo en carga. Los parámetros que deberán mantenerse constantes son la corriente de excitación y la velocidad de giro (frecuencia).

Con la carga desconectada, debe fijarse el punto inicial en que la corriente de excitación genere en la máquina la tensión nominal ( $220\text{V}$ ). Luego, sin modificar esta corriente de excitación y ajustando para cada variación de carga la velocidad a la nominal de  $1500\text{rpm}$ , se anotarán los valores de tensión, corriente y potencia.

#### 3.4. Ensayo de devatado

Se sustituirá la carga resistiva utilizada en el ensayo anterior por el banco de inductancias que ofrecen un factor de potencia del orden de  $0,1$ . Se excita desde valores mínimos el alternador y por accionamiento de las llaves cuchillas inversoras se busca obtener una corriente próxima a  $I_n$ . Deben

accionarse las llaves cuchillas en los tres carros de manera similar de forma de obtener una carga equilibrada en las tres fases.

Se deberá anotar la corriente de excitación requerida para cada punto de esta curva. Si se deseara obtener más puntos de la curva en devatado, se podrá, manteniendo velocidad de giro constante, operar sobre las conexiones posibles en los bancos de inductancias manteniendo la intensidad  $I_n$  para diferentes valores de la corriente de excitación.

### **3.5. Medida de resistencias**

La medida de la resistencia del circuito inductor se realizará por el método voltamperimétrico.

### 3.6. Puesta en paralelo

Se detendrá el grupo que se utilizó para los ensayos anteriores (en adelante, grupo 1). Se realizará el conexionado a fin de poder controlar el segundo grupo alternador-motor de arrastre (se debe lograr alimentar la máquina de continua, excitación e inducido, y se debe proveer excitación para el alternador), teniendo en cuenta que las corrientes y tensiones involucradas son iguales que para el grupo 1.

Una vez comprobado el funcionamiento del grupo 2, se detendrá el mismo y se procederá a realizar el conexionado entre ambos alternadores (por medio de un interruptor el cual cerrará el paralelo) y al equipo de sincronización.

Teniendo realizadas las conexiones, se pondrá ambos grupos en marcha, llevando el grupo 1 a  $1500\text{rpm}$ . A continuación se realizará el procedimiento de sincronización (que deberá ser analizado en el preinforme) hasta asegurarse que están dadas las condiciones para cerrar el interruptor que une ambos alternadores trifásicos. Una vez que se compruebe esto se cerrará el interruptor trifásico con lo que ambas máquinas comenzarán a operar en paralelo.

Realizar variaciones con las variables disponibles (tensión de inducido y corriente de excitación en la máquina de continua y corriente de excitación en el alternador) y registrar los cambios en el punto de funcionamiento del conjunto (tensión y frecuencia generadas por los alternadores en paralelo).

Registrar las variaciones en la potencia activa que fluye entre los alternadores al realizar estas variaciones.

## 4. Relevamiento previo y realización de ensayos

Previo a la realización de las prácticas, se deberá relevar los datos de chapa de las máquinas así como de los instrumentos y dispositivos a ser utilizados en la práctica. Cada grupo deberá prever un plan de trabajo adecuado de forma de poder registrar todas las medidas de las magnitudes de interés para la realización del informe final.

## 5. Documentación a entregar

### 5.1. Preinforme

En el preinforme se deben tratar los puntos y contestar las preguntas planteadas a continuación.

Sobre la máquina síncrona en general:

- Describir los modelos de Potier y de Behn Eschemburg para el alternador trifásico. Explicar qué representa físicamente cada parámetro, qué ensayos es necesario realizar para calcularlos, qué magnitudes se deben relevar en cada ensayo y cómo se calculan los parámetros de los modelos a partir de las magnitudes relevadas.
- ¿Cómo será la curva de reactancia sincrónica? Justifique.

Sobre la puesta en paralelo:

- ¿Cuáles son las condiciones que se deben cumplir a la hora de conectar en paralelo dos máquinas síncronas trifásicas?
- ¿Qué función cumplen las dos lámparas en el equipo de sincronización? (ver figura 2)
- Dado el equipo del que dispone, elabore una lista que detalle el orden de las acciones a realizar a fin de realizar la conexión en paralelo de forma exitosa.

- ¿Qué sucede si se realiza la puesta en paralelo de manera incorrecta? (por ejemplo si se cierra el interruptor en un momento inadecuado).

Para todas las partes de la práctica:

- Elaborar el esquema de conexión exacto (no un simple unifilar) para cada ensayo donde figuren: las máquinas, los instrumentos, los reóstatos, los elementos de corte, los bornes de cada uno de los elementos recién mencionados, etc. Recordar que los cables conectan bornes de distintos elementos. Deben figurar los rangos de operación de todos los instrumentos y de los reóstatos junto con los cálculos e hipótesis que justifican su elección.
- Elaborar tablas o planillas para cada ensayo, donde figuren claramente los datos a relevar.

## 5.2. Informe

El informe se compone del preinforme más todo lo pedido para el informe.

### 5.2.1. Modelo

De los datos obtenidos en los ensayos se deberán obtener los parámetros correspondientes a cada modelo.

### 5.2.2. Gráficas

Se deberá graficar:

- Tensión de vacío -  $V$  vs  $I_{exc}$
- Curva de devatado -  $V$  vs  $I_{exc}$
- Diagrama fasorial del alternador mostrando los puntos relevados en el ensayo de carga resistiva
- Triángulo de Potier
- Curva de reactancia síncrona

Esta página ha sido intencionalmente dejada en blanco.



# Apéndice D

## Manual de utilización del Sistema

### D.1. Introducción

En el transcurso de este documento se realiza una descripción del equipo, sus partes, conexiones y los pasos a seguir para ser puesto en marcha. También se describen las diferentes partes de software y sus funciones para que el usuario pueda utilizarlo y sacarle el mayor provecho.

### D.2. Descripción del equipo

Este Sistema fue desarrollado a medida para realizar ensayos y actividades con máquinas sincrónicas y máquinas DC con que cuenta el Laboratorio de Máquinas Eléctricas del IIE.

El mismo está conformado por un hardware compuesto por una caja en cuyo interior se encuentran todos los elementos físicos/electrónicos que lo componen acompañado de un software desarrollado en LabVIEW para correr en una PC.

En la figura D.1 y D.2 se muestra el equipo desde diferentes ángulos donde se pueden observar los bornes y puertos por donde se conectan los periféricos con los que interactúa el equipo.

Se comenzará describiendo el conexionado del mismo, luego se indicarán los pasos a seguir para la puesta en marcha, finalizando con la descripción de las pantallas del software y de las diferentes partes que lo componen.

### D.3. Instalación y configuraciones

Para su funcionamiento el Sistema interactúa con un PC mediante puerto USB, ésta para ser utilizada conjuntamente con el equipo se le debe instalar el software del Sistema. Para realizar la instalación se debe copiar la carpeta que contiene el mismo en la PC, paso siguiente ejecutar el .exe dando comienzo a la instalación, luego de realizar algunos pasos que el instalador irá indicando, se dará por finalizada la misma.

## Apéndice D. Manual de utilización del Sistema

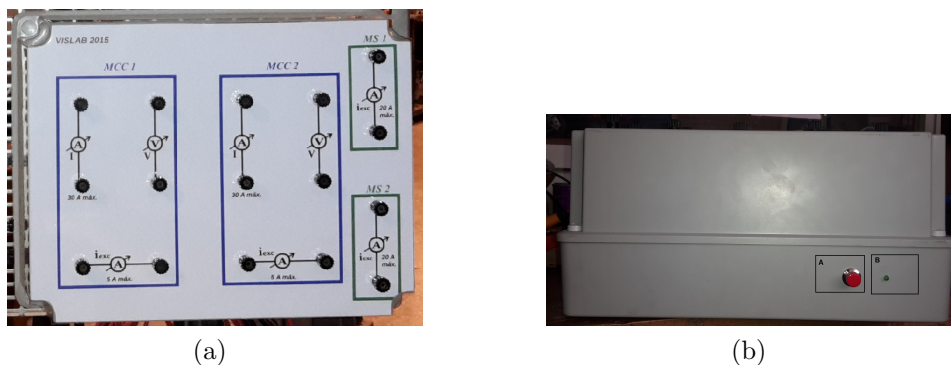


Figura D.1: D.1a Vista frontal del equipo donde se observan los bornes para realizar las conexiones de las señales a medir agrupadas por máquina. D.1b Vista lateral superior del equipo donde se observa el botón de encendido (A) y el led indicador de encendido del equipo(B)



Figura D.2: D.2a Vista lateral derecha del equipo, se observan los puertos correspondientes a la comunicación con el sensor de temperatura(A) y los encoder para relevar la velocidad de las máquinas (B). D.2b Vista lateral izquierda, se observa el puerto USB (A) y el RS485 (B) correspondientes a la comunicación con la PC y el CVMk2 respectivamente. También se encuentra la entrada de alimentación de equipo (D) junto con el fusible de protección(C)

Por último se debe copiar la carpeta llamada **ConstantesCurvas modelo BE** directamente en el disco C: de la PC, esta carpeta contiene archivos con parámetros y constantes que levantará el software en tiempo de ejecución (ver D.7).

Por otro lado la primera vez que se conecta el sistema a una determinada PC es necesario tener conexión a Internet, esto se debe a que el mismo descargará el driver correspondiente al manejo de la placa Arduino. Luego de realizar la descarga en el futuro no será necesario tener conexión a Internet para el funcionamiento del Sistema.

### D.4. Conexionado

A continuación se describen las conexiones a realizar para la puesta en marcha del equipo:

1. Conectar el equipo a la red 230 Vac para su alimentación

## D.5. Encendido del equipo

2. Conectar la PC al equipo por medio del puerto USB tipo B
3. Conectar el Analizador de Red CVMk2 al equipo mediante el puerto de comunicación RS485
4. Conectar las tensiones DC provenientes de las máquinas DC en los bornes para ese fin que se encuentran en el frente del equipo.
5. Realizar las conexiones de de las corrientes de armadura de las las máquinas DC en los bornes identificados para tal fin que se encuentran en el frente del equipo.
6. Realizar las conexiones de las corrientes de excitación de las máquinas DC que se encuentran sobre la parte inferior del frente del equipo.
7. Conectar las corrientes de excitación de las máquinas sincrónicas en los bornes identificados para tal motivo los cuales se encuentran en el frente del equipo sobre la derecha.
8. Conectar el/los Encoder para sensar la velocidad de las máquinas, los puertos se encuentran en el lateral derecho del equipo con la identificación correspondiente.
9. Conectar el sensor de temperatura en el puerto que se encuentran en el lateral derecho del equipo.

Sobre el equipo se encuentran identificados todos los puertos y bornes con su nombre correspondiente. Es importante destacar que la conexión mediante el puerto USB a la PC y la alimentación del equipo son las únicas necesarias para que el mismo opere, las restantes son opcionales. Tener en cuenta que en caso de realizar ensayos que comprenden la utilización de solo un conjunto de máquinas DC y Sincrónica, las conexiones se deben realizar en los bornes identificados como máquinas MS 1 y MCC 1. En todos los casos las conexiones de las señales DC es irrelevante la polaridad con la cual se realice la misma.

## D.5. Encendido del equipo

Para comenzar a trabajar con el equipo se debe encender con el botón de encendido que se encuentra en el lateral superior del equipo de color rojo, junto a un led verde que indica el estado del mismo. Luego se ejecuta el software desde el icono que se creó cuando fue instalado el mismo, al ejecutarlo se despliega la solapa principal donde se debe configurar la comunicación seleccionando el puerto de la PC en el cual se encuentra conectado el equipo. Después de realizar estos pasos se esta en condiciones de trabajar con el mismo.

## D.6. Descripción del Software

La estructura del software se compone de 8 solapas donde en cada una se encuentra todo lo necesario para realizar una actividad con el equipo. A continuación se describe la funcionalidad y las partes que las componen:

### Solapa Principal

Ésta, como el nombre lo indica, es la solapa principal que inicia el programa luego de su ejecución (ver D.3).



Figura D.3: Solapa Principal

Luego de realizar la comunicación con la PC se esta en condiciones de comenzar a trabajar con el equipo seleccionando la solapa de la tarde a que se dese realizar.

## Solapa Ensayo de Vacío de MS

En esta solapa se encuentra el unifilar de las máquinas que participan en este ensayo como también se visualizan los parámetros de interés para el mismo. En la figura D.4 se muestra la misma, donde se puede ver a la izquierda el gráfico de la tensión de vacío de la MS en función de la corriente de excitación de la misma. A la derecha se encuentra el esquemático del conexionado del ensayo, donde se pueden ver los valores de las corrientes, tensiones y velocidad de rotación así como también el valor de la potencia activa consumida por la máquina continua y la dirección de la misma.

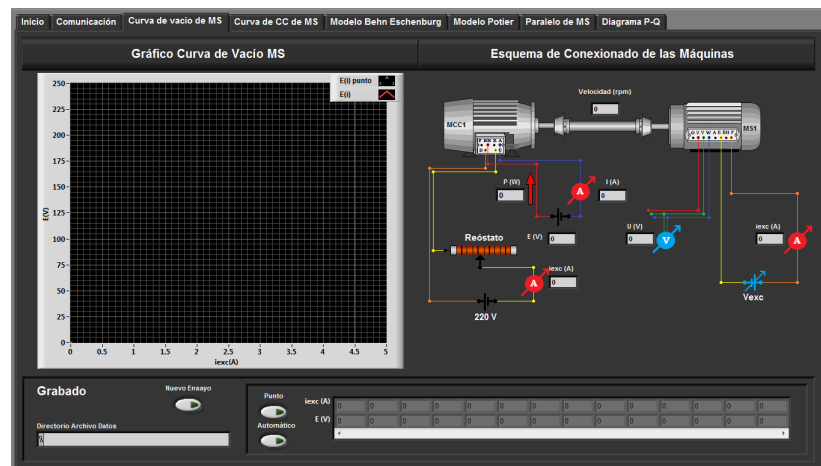


Figura D.4: Solapa de Ensayo de Vacío de MS

En la parte inferior de la solapa se encuentra la sección de grabado de datos (ver: D.6)

## Solapa Ensayo de Cortocircuito

En esta solapa se encuentra el unifilar de las máquinas que participan en este ensayo como también se visualizan los parámetros de interés para el mismo. En la figura D.5 se muestra la misma, donde se puede ver a la izquierda el gráfico de la corriente de línea durante el CC de la MS en función de la corriente de excitación de la misma. A la derecha se encuentra el esquemático del conexionado del ensayo, donde se pueden ver los valores de las corrientes, tensiones y velocidad de rotación así como también el valor de la potencia activa consumida por la máquina continua y la dirección de la misma.

En la parte inferior de la solapa se encuentra la sección de grabado de datos (ver: D.6)

## Solapa Modelo de Behn Eschenburg

En esta solapa se encuentran las herramientas para realizar el modelo de Behn Eschenburg de la máquina sincrónica. Sobre la parte derecha se encuentra un gráfico

## Apéndice D. Manual de utilización del Sistema

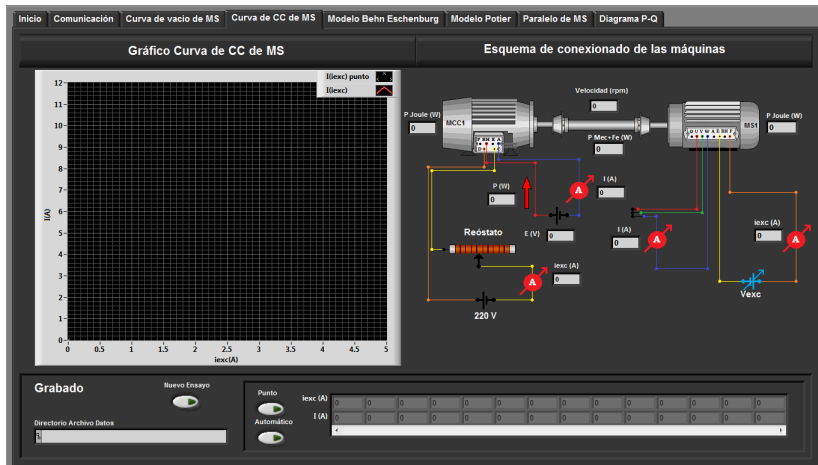


Figura D.5: Solapa de Ensayo de Cortocircuito de MS

donde se despliega el diagrama fasorial del modelo. A la izquierda de la pantalla se encuentran tres cuadros, los dos de arriba se despliegan las variables calculadas y medidas respectivamente que son relevantes al modelo en cuestión. Luego en el cuadro de más abajo se despliegan las constantes  $R_s$  y  $X_s$  y también se encuentra un botón para habilitar y deshabilitar el modelo  $X_s = F(i_{exc})$ . Finalmente en la parte inferior izquierda de la solapa se encuentra el modelo esquemático como se ve en la figura: D.6.

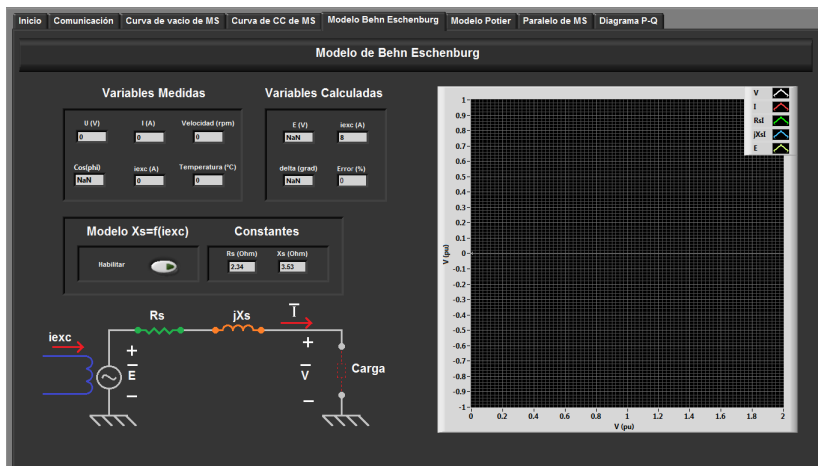


Figura D.6: Solapa de Modelo de Behn Eschenburg

## Solapa Modelo Potier

En esta solapa se encuentra representado el Modelo de Potier de las máquinas síncronas. Sobre la derecha se encuentra una gráfica donde se despliegan los fasores correspondiente al modelo de acuerdo al punto de funcionamiento en que se encuentre la máquina conectada en ese momento. Sobre la derecha se encuentran tres cuadros, los dos primeros corresponden a Variables Medidas y Calculadas respectivamente, en la tercera se despliegan las constantes del modelo. Finalmente en la parte inferior se encuentra el esquemático del modelo de la MS.

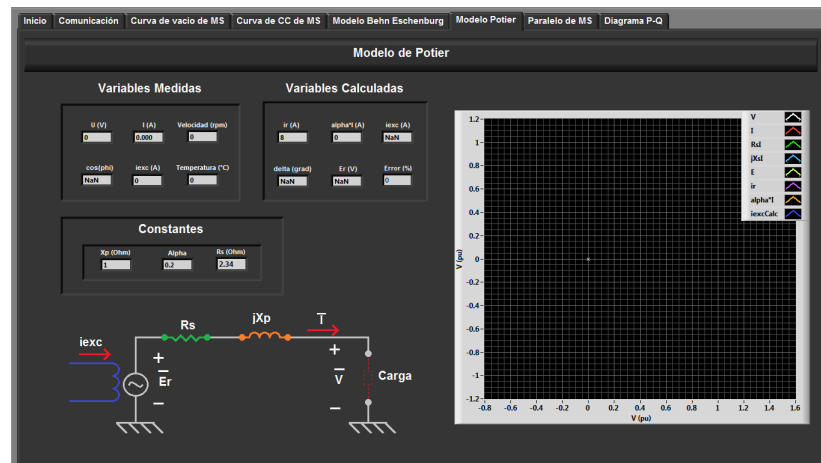


Figura D.7: Solapa de Modelo de Potier

## Solapa de Paralelo de MS

En esta solapa se encuentran las herramientas para visualizar los valores de las variables de interés durante la realización de la puesta en paralelo de dos máquinas síncronas. Como se puede ver en la figura: D.8 se encuentra el esquemático del conexionado de este ensayo junto con las tensiones, corrientes, velocidades y potencias que intervienen en el mismo. La flecha que se encuentra al costado de la potencia que están intercambiando ambas MS marca la dirección en la que esta fluyendo la potencia en cada momento dado que ésta puede variar de acuerdo a cual de ellas este entregando y cual absorbiendo potencia alternativamente. Las flechas que se encuentran sobre la tensión y corriente de campo de las máquinas DC también indican la dirección en la cual esta fluyendo la potencia, pero en el instante de iniciar el ensayo se deben corregir una única vez luego del conexionado en caso de ser necesario mediante el botón que se encuentra al lado de ellas.

## Solapa de Límites de Potencia de MS

Como se puede ver en la la figura: D.9 a la derecha de esta solapa se encuentra una gráfica en la cual se indica el punto de funcionamiento desde el punto de vista de potencia activa o reactiva que la MS este consumiendo o entregando en cada

## Apéndice D. Manual de utilización del Sistema

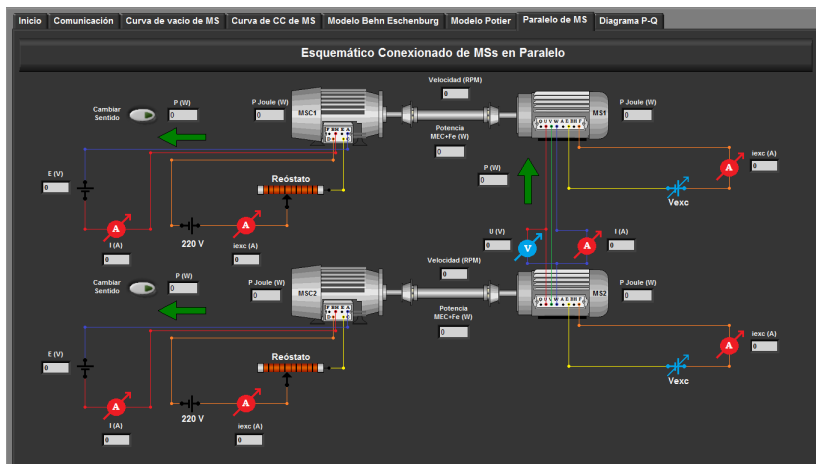


Figura D.8: Solapa de Paralelo de MS

momento la ir variando sus parámetros. En el cuadro de **Variables Calculadas** se muestran los valores de las variables de interés para el ensayo en cada momento, luego a la izquierda de la solapa se encuentra el bloque de grabado de datos cuyos elementos se encuentran descritos en la sección: D.6 a menos de la ubicación relativa de unos con otros.

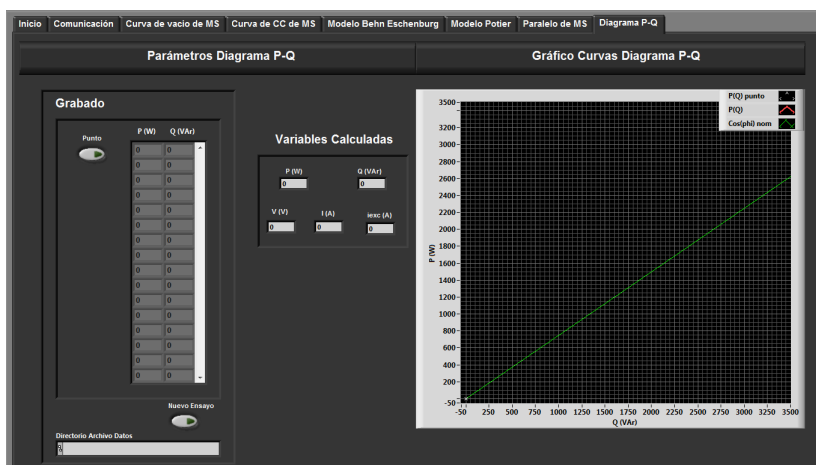


Figura D.9: Solapa de Límites de Potencia de MS

## Grabado de Datos

En el recuadro llamado Grabado de las solapas, se encuentran las herramientas con que cuenta el Sistema para grabar valores del ensayo mientras se está realizando. El campo **Directorio Archivo Datos** muestra el directorio y el nombre del archivo donde se están guardando los valores relevados. Luego se tiene el botón **Nuevo Ensayo** con el cual se puede comenzar un nuevo ensayo desde cero, al



presionarlo se crea un nuevo archivo vacío donde se guardarán los nuevos valores relevados. También se tiene un array en el cual se despliegan los valores que se van reelevando durante el ensayo pudiendo recorrerlo mediante la barra que tiene debajo. Los datos se pueden relevar de forma manual o automática, presionando el botón **Punto** se guardan punto a punto los datos que se quiera. La otra forma es presionando el botón **Automático** con el cual los datos se van guardando automáticamente a medida que el ensayo transcurre, relevando un dato cada vez que la medida se salga de una ventana predefinida calculada sobre el punto en el cual se encuentra el punto de operación en ese momento, de esta forma se evita guardar valores irrelevantes. Los archivos generados de los diferentes ensayos quedarán disponibles para ir a buscarlos luego de los ensayos en el directorio que se muestra en esa sección.

## D.7. Configuraciones

Para realizar diferentes configuraciones de los parámetros del software se cuenta con una carpeta llamada 'ConstantesCurvas modelo BE'. La misma se encuentra ubicada junto a la aplicación desarrollada en LabVIEW, contiene archivos .txt con los seteos de constantes y curvas características de las MS y MCC. Luego de instalar la aplicación en la PC, se debe ubicar esta carpeta en el disco 'C :' de la PC. La misma contiene los siguientes archivos:

- Cte\_MS\_MCC.txt
- CtePotier Alpha\_Xp.txt
- CurvaXs\_iexc.txt
- CurvaVacioE\_iexc.txt
- LimHistSenal.txt
- ValoresNomMS.txt
- LimHistGrabadoAutoEVacio.txt
- LimHistGrabadoAutoECC.txt

A continuación se describen los archivos .txt mencionados anteriormente:

**Cte\_MS\_MCC.txt:** en este archivo se encuentran las constantes correspondientes a la MS como MCC, estas constantes son:

- MS son las que corresponden a  $R_s$  y la  $X_s$ , siendo la  $X_s$  la de zona lineal
- MCC es la resistencia de armadura  $R_{mc}$

## Apéndice D. Manual de utilización del Sistema

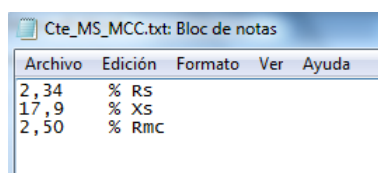


Figura D.10: Seteo de constantes de MS y de MCC

El seteo de las mismas se realiza como se muestra en la figura D.10, donde las constantes se encuentran ubicadas en una sola columna. El primer valor corresponde a la cte.  $R_s$ , el segundo corresponde a la cte.  $X_s$  y el último valor a la cte.  $R_{mc}$ .

**CtePotier Alpha\_Xp.txt:** en este archivo se encuentran las constantes de la MS correspondientes al modelo de Potier, siendo estas  $\alpha$  y  $X_p$ .

El seteo se realiza como se muestra en la figura D.11, donde las constantes se encuentran ubicadas en una sola columna, el primer valor corresponde a la cte.  $\alpha$  y el segundo corresponde a la cte.  $X_p$ .

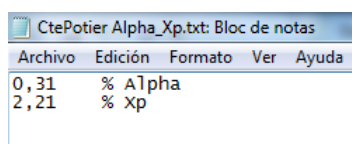


Figura D.11: Seteo de constantes de la MS modelo Potier

**CurvaXs\_iexc.txt:** este archivo contiene la curva  $X_s$  en función de  $i_{exc}$  ( $X_s(i_{exc})$ ), siendo  $X_s$  la reactancia sincrónica de la MS e  $i_{exc}$  la corriente de excitación de la MS. Los valores de la curva estarán ubicados en columnas. La primera columna corresponde al valor de  $i_{exc}$  y la segunda corresponde al valor de  $X_s$ .

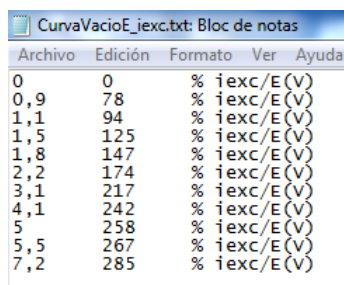
El seteo se realiza como se muestra en la figura D.12, donde los valores de cada variable se encuentran ubicados en dos columnas, la primera corresponde al valor de  $i_{exc}$  y la segunda corresponde al valor de  $X_s$ .

Archivo	Edición	Formato	Ver	Ayuda
0	17,9	%iexc/Xs		
0,9	17,9	%iexc/Xs		
1,1	17,6	%iexc/Xs		
1,5	17,2	%iexc/Xs		
1,8	16,8	%iexc/Xs		
2,2	16,3	%iexc/Xs		
3,1	14,4	%iexc/Xs		
4,1	12,2	%iexc/Xs		
5	10,6	%iexc/Xs		
5,5	10,0	%iexc/Xs		
7,2	8,2	%iexc/Xs		

Figura D.12: Curva  $X_s$  en función de  $i_{exc}$  correspondiente al modelo no lineal de la MS

**CurvaVacíoE\_iexc.txt:** este archivo contiene la curva de vacío  $E$  en función de  $i_{exc}$  ( $E(i_{exc})$ ), siendo  $E$  la tensión de vacío de la MS e  $i_{exc}$  la corriente de excitación de la MS. Los valores de la curva se encuentran ubicados en columnas, la primera corresponde al valor de  $i_{exc}$  y la segunda corresponde al valor de  $E$ .

El seteo se realiza como se muestra en la figura D.13, donde los valores de cada variable se encuentran ubicados en dos columnas, la primera corresponde al valor de  $i_{exc}$  y la segunda al valor de  $E$ .

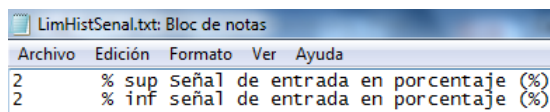


Archivo	Edición	Formato	Ver	Ayuda
0	0	% iexc/E(V)		
0,9	78	% iexc/E(V)		
1,1	94	% iexc/E(V)		
1,5	125	% iexc/E(V)		
1,8	147	% iexc/E(V)		
2,2	174	% iexc/E(V)		
3,1	217	% iexc/E(V)		
4,1	242	% iexc/E(V)		
5	258	% iexc/E(V)		
5,5	267	% iexc/E(V)		
7,2	285	% iexc/E(V)		

Figura D.13: Curva de vacío de la MS,  $E$  en función de  $i_{exc}$

**LimHistSenal.txt:** en este archivo se setean las constantes para la banda de histéresis para realizar el filtro digital de la señal proveniente del equipo.

El seteo se realiza como se muestra en la figura D.14, donde las constantes se encuentran ubicadas en una sola columna, el primer valor corresponde a la cte. del margen superior de la histéresis y el segundo valor corresponde a la cte. del margen inferior de la misma.

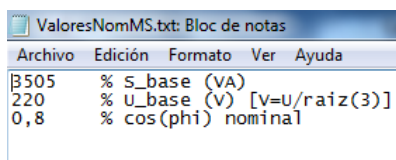


Archivo	Edición	Formato	Ver	Ayuda
2	% sup señal de entrada en porcentaje (%)			
2	% inf señal de entrada en porcentaje (%)			

Figura D.14: Limite banda de histéresis para filtrado de señales provenientes del equipo

**ValoresNomMS.txt:** en este archivo se realiza el seteo de los valores nominales de la MS, estos son potencia aparente nominal ( $S_n$ ), tensión nominal ( $U_n$ ) y  $\cos(\phi)$  nominal.

El seteo se realiza como se muestra en la figura D.15, donde las constantes se encuentran ubicadas en una sola columna, el primer valor corresponde a  $S_n$ , el segundo corresponde a  $U_n$  y el último valor al  $\cos(\phi)$  nominal.



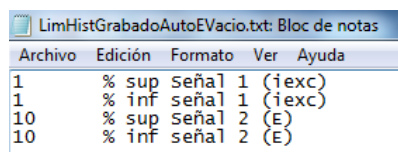
Archivo	Edición	Formato	Ver	Ayuda
3505	% S_base (VA)			
220	% U_base (V) [V=U/raiz(3)]			
0,8	% cos(phi) nominal			

Figura D.15: Valores nominales de la MS

**LimHistGrabadoAutoEVacio.txt:** en este archivo se setean las constantes de las bandas de histéresis para realizar el grabado de datos automáticos correspondientes al ensayo de vacío. El funcionamiento de la banda para cada señal es, cuando la señal sale de la banda el software grabará y gráfica el nuevo dato, para luego centrar la banda en el nuevo dato esperando el próximo donde situarse.

## Apéndice D. Manual de utilización del Sistema

El seteo se realiza como se muestra en la figura D.16, donde las constantes de la banda de histéresis se encuentran ubicadas en una sola columna, los dos primeros valores corresponden a la banda seteada para la corriente de excitación en el orden de margen superior e inferior, mientras que los últimos dos valores corresponden a la banda de la señal  $E$  en el orden de margen superior e inferior.

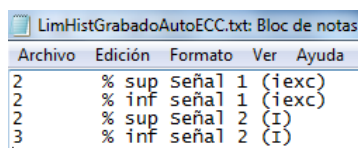


Archivo	Edición	Formato	Ver	Ayuda
1	% sup	señal 1	(iexc)	
1	% inf	señal 1	(iexc)	
10	% sup	señal 2	(E)	
10	% inf	señal 2	(E)	

Figura D.16: Limite banda de histéresis para grabado automático de ensayo de vacío

**LimHistGrabadoAutoECC.txt:** en este archivo se setean las constantes para las bandas de histéresis para realizar el grabado de datos automáticos correspondientes al ensayos de vacío. El funcionamiento de la banda de para cada señal es, cuando la señal sale de la banda el software grabará y grafica el nuevo dato, para luego centrar la banda en el nuevo dato esperando el próximo donde situarse.

El seteo se realiza como se muestra en la figura D.17, donde las constantes de la banda de histéresis se encuentran ubicadas en una sola columna, los dos primeros valores corresponden a la banda seteada para la corriente de excitación en el orden de margen superior e inferior, y los últimos dos valores corresponden a la banda de la señal  $I$  en el orden de margen superior e inferior.



Archivo	Edición	Formato	Ver	Ayuda
2	% sup	señal 1	(iexc)	
2	% inf	señal 1	(iexc)	
2	% sup	señal 2	(I)	
3	% inf	señal 2	(I)	

Figura D.17: Limite banda de histéresis para grabado automático de ensayo de cortocircuito

**Nota:** En todos los casos el seteo de las constantes en los .txt se debe realizar respetando la siguiente nomenclatura:

- los datos numéricos que tengan valores decimales deben de ser descritos con una coma (,)
- La separación entre dos valores debe ser con tabulación (TAB)
- los comentarios realizados en cada renglón se deberán separar por una tabulación (TAB) seguido de un signo de porcentaje (%)

## Apéndice E

### Código del firmware de la placa Arduino

```

#include<SimpleModbusMaster.h>
#include <I2C.h>

/*
    The example will use packet1 to read a register from address 0 (the adc ch0 v
    from the arduino slave (id=1). It will then use this value to adjust the brig
    of an led on pin 9 using PWM.
    It will then use packet2 to write a register (its own adc ch0 value) to addre
    on the arduino slave (id=1) adjusting the brightness of an led on pin 9 using
*/

//////////////////// Port information //////////////////////
#define baud 57600
#define timeout 1000
#define polling 200 // the scan rate
#define retry_count 10

// used to toggle the receive/transmit pin on the driver
#define TxEnablePin 9

// The total amount of available memory on the master to store data
#define TOTAL_NO_OF_REGISTERS 42

// This is the easiest way to create new packets
// Add as many as you want. TOTAL_NO_OF_PACKETS
// is automatically updated.
enum
{
    PACKET1,
    TOTAL_NO_OF_PACKETS // leave this last entry
};

// Create an array of Packets to be configured
Packetpackets[TOTAL_NO_OF_PACKETS];

// Masters register array
unsigned int reg[TOTAL_NO_OF_REGISTERS];

//Defino constantes de curvas
#define offsetI1 521
#define offsetI2 519
#define offsetI3 518
#define offsetI4 519
#define offsetI5 519
#define offsetI6 519

```

```
//Definicion de valores auxiliares
#define NumIter 50
#define MLX90614 0x5A

// Inicializacion de variables auxiliares
long V1r = 0;
float V1f = 0;
long V2r = 0;
float V2f = 0;
long I1r = 0;
float I1f = 0;
long I2r = 0;
float I2f = 0;
long I3r = 0;
float I3f = 0;
long I4r = 0;
float I4f = 0;
long I5r = 0;
float I5f = 0;
long I6r = 0;
float I6f = 0;

// Inicializacion de variables de campo
long V1 = 0;
long V2 = 0;
float I1 = 0;
float I2 = 0;
float I3 = 0;
float I4 = 0;
float I5 = 0;
float I6 = 0;
float T = 0;
double Vel1f;
double Vel2f;
String V1s;
String V2s;
String I1s;
String I2s;
String I3s;
String I4s;
String I5s;
String I6s;
String Ts;
String Vel1s;
String Vel2s;
String regaux;
```

```

// Contadores para Medida de velocidades
volatile int CNT1;
volatile int CNT2;

void setup()
{

    Serial.begin(115200);
    // Initialize each packet
    modbus_construct(&packets[PACKET1], 1, READ_HOLDING_REGISTERS, 62, 10, 0);
    // Initialize the Modbus Finite State Machine
    modbus_configure(&Serial2, baud, SERIAL_8N2, timeout, polling, retry_count,
    TxEnablePin, packets, TOTAL_NO_OF_PACKETS, reg);

    I2c.begin(); //Initialise the i2c bus
    I2c.timeOut(10);
    PORTD = (1 << PORTD1) | (1 << PORTD0); //enable pullups

    attachInterrupt(digitalPinToInterrupt(2), count1, CHANGE);
    attachInterrupt(digitalPinToInterrupt(18), count2, CHANGE);

}

void loop()
{
    // Inicio de conteo de interrupciones
    CNT1=0;
    CNT2=0;
    long start = micros();

    // Inicacion de muestreo de Variables de Tensiones/Corrientes continuas
    for (int i=0; i<NumIter; i++){
        V1r += analogRead(8);
        V2r += analogRead(9);
        I1r += analogRead(10);
        I2r += analogRead(11);
        I3r += analogRead(12);
        I4r += analogRead(13);
        I5r += analogRead(14);
        I6r += analogRead(15);
    }

    V1r /= NumIter+1;
    V2r /= NumIter+1;
    I1r /= NumIter+1;

```



```

I2r /= NumIter+1;
I3r /= NumIter+1;
I4r /= NumIter+1;
I5r /= NumIter+1;
I6r /= NumIter+1;

// Se actualizan medidas en CVMk2 por modbus
modbus_update();

V1f = float(V1r);
V1 = long((0.0000004)*pow(V1f,3) + (-0.0006)*pow(V1f,2) +
(0.5624)*V1f + 34.262); // Curva de ajuste de orden 3 del sensor
V1s = String(V1,HEX);
V1s.toUpperCase();

V2f = float(V2r);
V2 = long((0.0000004)*pow(V2f,3) + (-0.0005)*pow(V2f,2) +
(0.5374)*V2f + 33.964); // Curva de ajuste de orden 3 del sensor
V2s = String(V2,HEX);
V2s.toUpperCase();

// Se miden corrientes de los sensores de efecto hall

I1f = float(I1r);
I1 = 100*(0.0262*I1f - 13.621);
I1s = String((long)I1,HEX);
I1s.toUpperCase();

I2f = float(I2r);
I2 = 100*(0.0262*I2f - 13.563);
I2s = String((long)I2,HEX);
I2s.toUpperCase();

I3f = float(I3r);
I3 = 100*(0.0726*I3f-37.596);
I3s = String((long)I3,HEX);
I3s.toUpperCase();

I4f = float(I4r);
I4 = 100*(0.0732*I4f-37.894);
I4s = String((long)I4,HEX);
I4s.toUpperCase();

I5f = float(I5r);
I5 = 100*(0.0484*I5f-25.044);
I5s = String((long)I5,HEX);

```

```

I5s.toUpperCase();

I6f = float(I6r);
I6 = 100*(0.0478*I6f-24.786);
I6s = String((long)I6,HEX);
I6s.toUpperCase();

/////////////////////////////////Medicion de sensor de temperatura/////////////////////////////////
int data_low = 0;
int data_high = 0;
I2c.read(MLX90614, 0x07, 2);

data_low = I2c.receive(); //Read 1 byte and then send ack
data_high = I2c.receive(); //Read 1 byte and then send ack

//This converts high and low bytes together and processes temperature,
//MSB is a error bit and is ignored for temps
double tempFactor = 0.02; // 0.02 degrees per LSB (measurement resolution of t
double tempData = 0x0000; // zero out the data
int frac; // data past the decimal point

// This masks off the error bit of the high byte, then moves it left 8 bits an
tempData = (double)((data_high & 0x007F) << 8) + data_low);
tempData = (tempData * tempFactor)-0.01;

T = 10*(tempData - 273.15);

Ts = String((long)T,HEX);
Ts.toUpperCase();
/////////////////////////////////Fin de codigo de sensor de temperatura/////////////////////////////////

// Retraso 150 para contar mas veces
delay(160);

// Calculo las velocidades
long finish = micros() - start;
double Vel1f = ((double)CNT1/(double)finish)*1000000*60/200;
double Vel2f = ((double)CNT2/(double)finish)*1000000*60/200;

Vel1s = String((long)Vel1f,HEX);
Vel1s.toUpperCase();
Vel2s = String((long)Vel2f,HEX);
Vel2s.toUpperCase();

Serial.print("00");

```

```

regaux = String(reg[0],HEX);
regaux.concat(String(reg[1],HEX));
regaux.toUpperCase();
if (reg[0]>=0) {
    regaux = FillZeros(regaux);
}
Serial.println(regaux);
Serial.print("01");
regaux = String(reg[2],HEX);
regaux.concat(String(reg[3],HEX));
regaux.toUpperCase();
if (reg[2]>=0) {
    regaux = FillZeros(regaux);
}
Serial.println(regaux);
Serial.print("02");
regaux = String(reg[4],HEX);
regaux.concat(String(reg[5],HEX));
regaux.toUpperCase();
if (reg[4]>=0) {
    regaux = FillZeros(regaux);
}
Serial.println(regaux);
Serial.print("03");
regaux = String(reg[6],HEX);
regaux.concat(String(reg[7],HEX));
regaux.toUpperCase();
if (reg[6]>=0) {
    regaux = FillZeros(regaux);
}
Serial.println(regaux);
Serial.print("04");
regaux = String(reg[8],HEX);
regaux.concat(String(reg[9],HEX));
regaux.toUpperCase();
if (reg[8]>=0) {
    regaux = FillZeros(regaux);
}
Serial.println(regaux);
Serial.print("05");
if (V1>=0) {
    V1s = FillZeros(V1s);
}
Serial.println(V1s);
Serial.print("06");
if (V2>=0) {

```

```
V2s = FillZeros(V2s);
}
Serial.println(V2s);
Serial.print("07");
if (I1>=0) {
  I1s = FillZeros(I1s);
}
Serial.println(I1s);
Serial.print("08");
if (I2>=0) {
  I2s = FillZeros(I2s);
}
Serial.println(I2s);
Serial.print("09");
if (I3>=0) {
  I3s = FillZeros(I3s);
}
Serial.println(I3s);
Serial.print("10");
if (I4>=0) {
  I4s = FillZeros(I4s);
}
Serial.println(I4s);
Serial.print("11");
if (I5>=0) {
  I5s = FillZeros(I5s);
}
Serial.println(I5s);
Serial.print("12");
if (I6>=0) {
  I6s = FillZeros(I6s);
}
Serial.println(I6s);
Serial.print("13");
if (T>=0) {
  Ts = FillZeros(Ts);
}
Serial.println(Ts);
Serial.print("14");
if (Vel1f>=0) {
  Vel1s = FillZeros(Vel1s);
}
Serial.println(Vel1s);
Serial.print("15");
if (Vel2f>=0) {
  Vel2s = FillZeros(Vel2s);
```

```

}
Serial.println(Vel2s);

}

String FillZeros(String X) {
    String Aux1;
    String Aux2;
    if (X.length() < 8) {
        int i = 8 - X.length();
        Aux2 = X;
        for (int i = 8 - X.length(); i > 0; i--) {
            Aux1 = Aux2;
            Aux2 = "0";
            Aux2.concat(Aux1);
        }
        X = Aux2;
        return X;
    }
}

void count1() {
    CNT1++;
}

void count2() {
    CNT2++;
}

```

Esta página ha sido intencionalmente dejada en blanco.

# Apéndice F

## Gestión de tiempos

### F.1. Cambios en la planificación

Durante el transcurso del proyecto el cronograma planteado en un inicio sufrió alguna modificaciones. Se había estipulado en un principio la programación de tiempos del proyecto hasta el Hito 1 (04/03/15) con las tareas siguientes:

1. Estudio del analizador de red CVMk2
2. Estudio protocolos de comunicación que interviene en el Sistema
3. Estudio de plataforma de desarrollo LabVIEW
4. Establecer comunicación entre CVMk2 y PC vía LabVIEW
5. Obtención en PC de variables de interés para el Sistema
6. Definir equipamiento necesario para acondicionamiento de variable y señales eléctricas de forma adecuada

En ésta primera etapa del proyecto se tuvo la necesidad de reprogramar las tareas dado que no se lograba la comunicación de forma adecuada del CVMk2 con la PC. Esto significo el no poder definir la arquitectura del Sistema en tiempo y forma con lo cual trajo consigo la definición del equipamiento a adquirir.

Próximo al Hito 1 se logro la comunicación antes mencionada, llegado así al mismo con los puntos del 1 al 4 concluidos y el 5 en parte. El haber concluido de forma satisfactoria el punto 4 nos permitió seguir con la realización del proyecto dentro de los objetivos generales propuestos en un principio.

La segunda etapa del proyecto marcada por la fecha del culminación del Hito 2 (16/06/15) se compone de las siguientes tareas:

1. Desarrollo de entorno gráfico y procesamiento de datos adquiridos en LabVIEW
2. Definir equipamientos para el acondicionamiento de variables y señales eléctricas de forma adecuada

## Apéndice F. Gestión de tiempos

### 3. Familiarización con la Norma IEE std 15-2009

En esta segunda etapa del proyecto la programación de las tareas a realizar fue afectada por la reprogramación realizada al finalizar el Hito 1. La tarea 2 descrita en el esquema anterior pertenecía en un principio a la primer etapa pero a consecuencia de la reprogramación paso a formar parte de de los objetivo de este Hito.

Durante esta etapa se comenzó a trabajar en paralelo en lo que refiere a software, hardware y comunicación. Culminando esta etapa con los puntos 1 y 2 con un buen avance. Se realizo la definición de la arquitectura definitiva con lo cual dio lugar a realizar las compras de los componentes y avanzar en el desarrollo del software y comunicación de forma especifica para la arquitectura definida. En tanto que el punto 3 correspondiente a la norma se replanifico para el final del proyecto.

Al finalizar esta etapa se tenía prácticamente culminado el tratamiento de datos y un alto avance en lo correspondiente al entorno gráfico en LabVIEW. El hardware se tenía prácticamente definida la arquitectura especifica y se fue avanzando mientras se esperaban los componentes adquiridos del exterior.

Finalmente en la última etapa del proyecto se trabajo en la ejecución y finalización del hardware paralelamente con el software tanto en tratamiento de datos como comunicación. La ejecución del hardware y pruebas implicó una gran dedicación horaria no planificada de forma correcta, explicada por la falta de experiencia en trabajos de esta índole por parte del grupo, tanto en aspecto de manualidad como de gestión de materiales.

En lo correspondiente al estudio de la Norma IEE std 15-2009, este tema fue incluido para realizar ensayos no practicados anteriormente en el Taller de Máquinas Eléctricas, pero a raíz de las demoras por los imprevistos mencionados se decide dejar sin efecto su estudio. Esto no comprometió los objetivos centrales del proyecto dado que su realización solamente extiende las funcionalidades del mismo.



## F.2. Dedicación mensual

En la tabla F.1 se encuentran las horas de dedicación del grupo de proyecto para la realización del mismo discriminadas por mes.

Mes	Dedicación (hs)
Agosto 2014	27
Setiembre 2014	68
Octubre 2014	24
Noviembre 2014	12
Diciembre 2014	39
Enero 2015	63
Febrero 2015	132
Marzo 2015	117
Abril 2015	70
Mayo 2015	128
Junio 2015	153
Julio 2015	81
Agosto 2015	303
Septiembre 2015	550
Octubre 2015	290
<b>Total</b>	<b>2057</b>

Tabla F.1: Dedicación horaria mensual del equipo del proyecto.

Esta página ha sido intencionalmente dejada en blanco.

# Apéndice G

## Gestión de gastos

### G.1. Gastos realizados

Para el proyecto se planteo como uno de los objetivos de diseño utilizar componentes que se encuentren en plaza. Éste se llevo a cavo pese a haber realizado compras en el exterior cuya justificación se debió al hecho de reducir costos, dado que al adquirirlos mediante compras vía Internet del exterior los mismos son significativamente menores.

En la tabla G.1 se detallan las compras realizadas con sus costo.

Hardware	Unidades	Costo[USD]	Costo total [\$U]
Placa Arduino Mega 2560 R3	1	46,00	1357,00
Caja PVC	1	24,14	712,00
Shield RS-485 Arduino	1	21,00	619,50
Transductor de Corriente efecto Hall ACS712 5 Amp	6	47,70	1407,00
Transductor de Corriente efecto Hall ACS712 20 Amp	3	5,46	161,07
Transductor de Corriente efecto Hall ACS712 30 Amp	3	5,46	161,07
Optoacoplador COM-00784 ROHS	6	13,50	398,25
Sensor de temperatura	2	39,90	1177,50
Vinilo y Acrílico	1	21,50	640,00
Gastos varios	1	135,08	3985,00
		Total	10618,40

Tabla G.1: Costos de insumos para la realización del proyecto(cotización USD a 29.5 \$U)

Habiendo sido el costo estimado sin la mano de obra de \$ 15840 y siendo el real de \$ 10618 se puede ver que la estimación estuvo por sobre de la real. Un posible

## Apéndice G. Gestión de gastos

motivo fue el hecho de no haber comprado las tarjetas expansoras del analizador de red CVMk2.

# Apéndice H

## Hoja de Datos de Equipo

En la presente tabla se resumen las señales a medir junto a sus características tanto en rango de operación como incertidumbre.

Símbolo	Características	Notas	Valor	Unidad	Incertidumbre
VMCC1	Rango de Medidas	Tensión de armadura	50..250	V	±1 V
VMCC1	Rango de Medidas	Tensión de armadura	50..250	V	±1 V
iexc MCC1	Rango de Medidas	iexc MCC1	-5..5	A	±0,1 A
iexc MCC2	Rango de Medidas	iexc MCC2	-5..5	A	±0,1 A
iexc MS1	Rango de Medidas	iexc MS1	-20..20	A	±0,1 A
iexc MS2	Rango de Medidas	iexc MS2	-20..20	A	±0,1 A
IMCC1	Rango de Medidas	IMCC1	-30..30	A	±0,1 A
IMCC2	Rango de Medidas	IMCC2	-30..30	A	±0,1 A
Tobj	Rango de Medidas	véase hoja de datos MLX90614	--	--	--
Vvel	Rango de Medidas	Velocidad de eje Encoder	1100..1600	rpm	±2 rpm
IV1	Máx Corriente de fuga en régimen	limitado por fusible de protección	0,5	A	--
IV2	Máx Corriente de fuga en régimen	limitado por fusible de protección	0,5	A	--
VAC	Tensión de Alimentación		180..250	Vrms	--

Figura H.1: Hoja de Datos del Equipo

Esta página ha sido intencionalmente dejada en blanco.

# Referencias

- [1] Arduino. <https://www.arduino.cc/>. *Web*, 2015.
- [2] Lammert Bies. <http://www.lammertbies.nl/comm/info/crc-calculation.html>. *Web*, 1997-2015.
- [3] Stephen J. Chapman. Maquinas eléctricas. *Libro*, 2015.
- [4] CIRCUTOR. Manual de instrucciones (m98206501-01-13a) - cvmk2 -. *Manual*, 2015.
- [5] Monografía de Medidas Eléctricas LabView Año 2013. <https://eva.fing.edu.uy/enrol/index.php?id=579>. *Monografía*, 2013.
- [6] RASPBERRY PI FOUNDATION. <http://www.raspberrypi.org/>. *Web*, 2015.
- [7] National Instruments. <http://latam.ni.com/>. *Web*, 2015.
- [8] LabVIEW MaderHub. <https://labviewhacker.com/>. *Web*, 2015.
- [9] Simply Modbus. <http://www.simplymodbus.ca/>. *Web*, 2015.
- [10] Modbus Organization. <http://www.modbus.org/>. *Web*, 2005-2015.
- [11] Witte Software. [http://www.modbustools.com/modbus\\_poll.html](http://www.modbustools.com/modbus_poll.html). *Web*, 2015.
- [12] Wikipedia. [http://es.wikipedia.org/wiki/raspberry\\_pi](http://es.wikipedia.org/wiki/raspberry_pi). *Web*, 2015.
- [13] Wikipedia. <http://es.wikipedia.org/wiki/rs-232>. *Web*, 2015.
- [14] Wikipedia. <http://es.wikipedia.org/wiki/rs-485>. *Web*, 2015.
- [15] Wikipedia. [https://es.wikipedia.org/wiki/system\\_management\\_bus](https://es.wikipedia.org/wiki/system_management_bus). *Web*, 2015.
- [16] WordPress. <http://www.tolaemon.com>. *Web*, 2015.

Esta página ha sido intencionalmente dejada en blanco.



# Glosario

- **MS** Máquina Sincrónica
- **MCC** Máquina de Corriente Continua
- **PWM** Pulse-width modulation
- **AI** Analog Input
- **UART** Universal Asynchronous Receiver-Transmitter
- **IDE** Integrated Development Environment
- **SMBus** System Management Bus
- **DSP** Digital Signal Processor
- **SW** Software
- **HW** Hardware

Esta página ha sido intencionalmente dejada en blanco.

# Índice de tablas

3.1. Valores de fuente medidos con el equipo en operación . . . . .	26
3.2. Tabla de Mapeo de Pines en Arduino-Shield vs. Variables Físicas .	38
4.1. Tabla de Dígitos Identificadores . . . . .	58
6.1. Tabla con valores de offset relevados . . . . .	91
F.1. Dedicación horaria mensual del equipo del proyecto. . . . .	159
G.1. Costos de insumos para la realización del proyecto(cotización USD a 29.5 \$U) . . . . .	161

Esta página ha sido intencionalmente dejada en blanco.

# Índice de figuras

1.1. Fotografía del Laboratorio de Máquinas Eléctricas del IIE . . . . .	1
1.2. Diagrama simplificado del grupo MS-MCC utilizada en el Laboratorio . . . . .	3
2.1. Entorno de desarrollo para Arduino . . . . .	11
2.2. Vista de placa Arduino Uno . . . . .	11
2.3. Diagrama esquemático de la placa Raspberry Pi . . . . .	12
2.4. Esquema del Sistema 1 . . . . .	13
2.5. Esquema del Sistema 2 . . . . .	13
2.6. Esquema del Sistema 3 . . . . .	14
2.7. Esquema del Sistema 4 . . . . .	15
2.8. Esquema de la arquitectura definida para el Sistema . . . . .	17
3.1. Placa Arduino Mega 2560 . . . . .	20
3.2. Shield RS485 para realizar la conversión de RS-485 a UART . . . . .	21
3.3. Esquemático de placa de Comunicación y Alimentación de Arduino. Referencias de pines: 1- Vcc (5 Vdc), 2- -Data, 3- +Data, 4- GND . . . . .	21
3.4. Circuito optoacoplado de sensado de corriente continua . . . . .	22
3.5. Circuito de Placa IO completo . . . . .	23
3.6. Circuito CAD de la Placa IO para armado de la misma . . . . .	24
3.7. Circuito de la Fuente original . . . . .	25
3.8. Datos típicos del regulador de tensión de la fuente (@25 °C) . . . . .	26
3.9. Circuito de la Fuente final implementado . . . . .	26
3.10. Imagen frontal CVMk2 . . . . .	27
3.11. Características Técnicas . . . . .	28
3.12. Etiqueta de conexión de Tensión y TI . . . . .	29
3.13. Conexión de medida en la red trifásica . . . . .	29
3.14. Diagrama de placa sensor de temperatura . . . . .	30
3.15. 3.15a Encapsulado de sensor 3.15b Imagen del sensor montado en placa impresa . . . . .	30
3.16. 3.16a Breakoutboard del sensor ACS712 de 5 A 3.16b Breakout- board del sensor ACS712 de 20 y 30 A . . . . .	31
3.17. Encoder E6B2-C . . . . .	32
3.18. Diagrama de conexionado de la placa Arduino . . . . .	33
3.19. Diagrama del Sistema . . . . .	34
3.20. Diagrama de los bornes de medidas DC del frente del equipo . . . . .	35

## Índice de figuras

3.21. Diagrama de sensores de corriente DC . . . . .	36
3.22. Diseño del frente del equipo . . . . .	41
3.23. Vista lateral izquierda del equipo . . . . .	41
3.24. Vista lateral derecha del equipo: A)Puertos de adquisición de señales de Temperatura, B)Puertos de adquisición de señales de Velocidades . . . . .	42
3.25. Vista lateral superior del equipo: A)Pulsador de encendido del equipo, B)Led indicador de estado del Equipo(ON/OFF) . . . . .	42
3.26. Parte de posterior del interior del Equipo . . . . .	43
3.27. Vista de posterior de la tapa del gabinete del equipo . . . . .	44
3.28. Mímico que se encuentra en el frente del equipo . . . . .	45
4.1. Formato general de las tramas . . . . .	51
4.2. Formato trama MODBUS (ASCII y RTU) . . . . .	51
4.3. Modbus Poll . . . . .	53
4.4. Configuración de direcciones en MODBUS POLL . . . . .	55
4.5. Datos presentados por MODBUS POLL . . . . .	55
4.6. Diagrama de bloques del Software de Comunicación de testeo para Arduino . . . . .	56
4.7. Panel frontal del Software de Comunicación de testeo para Arduino . . . . .	56
4.8. Esquemático del circuito de testeo de Comunicación con placa Arduino . . . . .	57
4.9. Paquete enviado de manera serial por placa Arduino a PC . . . . .	58
4.10. Backpanel de Driver Arduino en LabVIEW . . . . .	59
5.1. Panel Frontal . . . . .	63
5.2. Diagrama de Bloques . . . . .	64
5.3. 5.3a: Paleta de Funciones 5.3b: Paleta de Control . . . . .	64
5.4. Vista de solapa principal de Software . . . . .	66
5.5. 5.5a: Ubicación de Tab Control 5.5b: Icono de Tab Control . . . . .	67
5.6. Pantalla de inicio de Software . . . . .	68
5.7. Solapa Curva de Vacío de MS . . . . .	69
5.8. Solapa Curva de CC de MS . . . . .	69
5.9. Solapa Modelo de Behn Eschenburg . . . . .	70
5.10. Solapa Modelo Potier . . . . .	71
5.11. Solapa de Puesta en paralelo de MS . . . . .	72
5.12. Puesta en paralelo de MS . . . . .	73
5.13. Solapa de Diagrama P-Q . . . . .	74
5.14. 5.14a: Directorio Archivo Datos 5.14b: Controles guardado de Datos . . . . .	75
5.15. Diagrama de flujo para ensayo de CC . . . . .	76
5.16. Diagrama de flujo para ensayo de Vacío . . . . .	77
5.17. Diagrama de flujo para Diagrama P-Q . . . . .	78
5.18. Bloque encargado del Full Screen . . . . .	79
5.19. Seteo de constantes de MS y de MCC . . . . .	80
5.20. Seteo de constantes de la MS modelo Potier . . . . .	80
5.21. Curva $X_s$ en función de $i_{exc}$ correspondiente al modelo no lineal de la MS . . . . .	80
5.22. Curva de vacío de la MS, $E$ en función de $i_{exc}$ . . . . .	81

5.23. Limite banda de histéresis para filtrado de señales provenientes del equipo . . . . .	81
5.24. Valores nominales de la MS . . . . .	81
5.25. Limite banda de histéresis para grabado automático de ensayo de vacío . . . . .	82
5.26. Limite banda de histéresis para grabado automático de ensayo de cortocircuito . . . . .	82
6.1. Curva de Medidas de Campo en función de los enteros medidos por Arduino . . . . .	86
6.2. Curva de Medidas de Campo en función de los enteros medidos por Arduino . . . . .	87
6.3. Curva de comportamiento de $i_{exc}[A]$ de MCC 1 . . . . .	88
6.4. Curva de comportamiento de $i_{exc}[A]$ de MCC 2 . . . . .	88
6.5. Curva de comportamiento de $I[A]$ de MCC 1 . . . . .	89
6.6. Curva de comportamiento de $I[A]$ de MCC 2 . . . . .	89
6.7. Curva de comportamiento de $i_{exc}[A]$ de MS 1 . . . . .	90
6.8. Curva de comportamiento de $i_{exc}[A]$ de MS 2 . . . . .	90
7.1. Fotografía del Equipo en su disposición final en panel . . . . .	93
A.1. Mapa del programa principal desarrollado en LabVIEW . . . . .	98
A.2. Rutina de Driver Arduino para LabVIEW . . . . .	99
A.3. A.3a SubVI Ruido devuelve True A.3b SubVI Ruido devuelve Falso	100
A.4. A.4a SubVI Ruido A.4b SubVI Ruido panel frontal . . . . .	100
A.5. C Parte 1:Rutina en programa principal para el estudio de las máquinas sincrónicas trabajando en paralelo . . . . .	101
A.6. C Parte 2:Rutina en programa principal para el estudio de las máquinas sincrónicas trabajando en paralelo . . . . .	102
A.7. Rutina en software principal para ensayo de vacío máquina sincrónica	103
A.8. Rutina en software principal para ensayo de cortocircuito para la máquina sincrónica . . . . .	104
A.9. Rutina en software principal para ensayo de diagrama P-Q de la máquina sincrónica . . . . .	106
A.10.A.10a Modelo Máquina sincrónica A.10b Diagrama fasorial modelo Potier . . . . .	107
A.11.Rutinda del model de Potier en programa principal . . . . .	108
A.12.Modulo SubVI Potier . . . . .	108
A.13.Rutina subVI Potier . . . . .	109
A.14.Señales necesarias para el modelo de Potier y cálculo de las bases para obtener las magnitudes en 'por unidad' . . . . .	110
A.15.Sector C: creación de vectores. Sector E: acondicionamiento de los vectores para ser graficados . . . . .	110
A.16.Sector F: Cálculo de $i_{exc}$ a partir de curva ensayo de vacío y prepara los fasores para ser graficados . . . . .	111
A.17.Sector D: Cálculo $\alpha * I$ y se prepara fasor . . . . .	111

## Índice de figuras

A.18. Panel Frontal subVI Potier . . . . .	112
A.19. Rutina del modelo de Behen-Eschenburg en programa principal . . . . .	113
A.20. Modulo SubVI Behen-Eschenburg . . . . .	113
A.21. Rutina SubVI para modelo de Behen-Eschenburg . . . . .	114
A.22. Señales necesarias para el modelo de Behen-Eschenburg y cálculo de las bases para obtener las magnitudes en 'por unidad' . . . . .	115
A.23. Sector C: creación de vectores. Sector D: acondicionamiento de los vectores para ser graficados . . . . .	115
A.24. Sector E: Señales de salida del subVI Behen-Eschenburg . . . . .	116
A.25. Panel Frontal subVI Behen-Eschenburg . . . . .	116
A.26. Formato de los archivos de datos cte (archivos de textos) . . . . .	117
A.27. Módulo leer archivo módulo . . . . .	117
A.28. Rutina SubVI para leer Archivo txt . . . . .	118
B.1. Modelo monofásico de la máquina sincrónica . . . . .	120
B.2. Modelo Behen-Eschenburg, monofásico de la máquina sincrónica . . . . .	120
B.3. Diagrama fasorial modelo Behen-Eschenburg . . . . .	121
B.4. Comportamiento Reactancia Sincrónica . . . . .	122
B.5. Diagrama fasorial con la máquina en zona de saturación . . . . .	122
D.1. D.1a Vista frontal del equipo donde se observan los bornes para realizar las conexiones de las señales a medir agrupadas por máquina. D.1b Vista lateral superior del equipo donde se observa el botón de encendido (A) y el led indicador de encendido del equipo(B) . . . . .	136
D.2. D.2a Vista lateral derecha del equipo, se observan los puertos correspondientes a la comunicación con el sensor de temperatura(A) y los encoder para relevar la velocidad de las máquinas (B). D.2b Vista lateral izquierda, se observa el puerto USB (A) y el RS485 (B) correspondientes a la comunicación con la PC y el CVMk2 respectivamente. También se encuentra la entrada de alimentación de equipo (D) junto con el fusible de protección(C) . . . . .	136
D.3. Solapa Principal . . . . .	138
D.4. Solapa de Ensayo de Vacío de MS . . . . .	139
D.5. Solapa de Ensayo de Cortocircuito de MS . . . . .	140
D.6. Solapa de Modelo de Behn Eschenburg . . . . .	140
D.7. Solapa de Modelo de Potier . . . . .	141
D.8. Solapa de Paralelo de MS . . . . .	142
D.9. Solapa de Límites de Potencia de MS . . . . .	142
D.10. Seteo de constantes de MS y de MCC . . . . .	144
D.11. Seteo de constantes de la MS modelo Potier . . . . .	144
D.12. Curva $X_s$ en función de $i_{exc}$ correspondiente al modelo no lineal de la MS . . . . .	144
D.13. Curva de vacío de la MS, $E$ en función de $i_{exc}$ . . . . .	145
D.14. Limite banda de histéresis para filtrado de señales provenientes del equipo . . . . .	145
D.15. Valores nominales de la MS . . . . .	145



## Índice de figuras

D.16.Limite banda de histéresis para grabado automático de ensayo de vacío . . . . .	146
D.17.Limite banda de histéresis para grabado automático de ensayo de cortocircuito . . . . .	146
H.1. Hoja de Datos del Equipo . . . . .	163



Esta es la última página.  
Compilado el martes 27 octubre, 2015.  
<http://iie.fing.edu.uy/>