

**DISEÑO E IMPLEMENTACIÓN DE UN EQUIPO DE DIAGNÓSTICO Y
VISUALIZACIÓN DE LAS SEÑALES ANALÓGICAS Y DIGITALES EN
LOS VARIADORES DE FRECUENCIA ABB DE LAS LÍNEAS ACS550,
ACS800, ACS850, ACS880 PARA LA EMPRESA COL CONTROLES &
INTEGRACIÓN S.A.S.**

PEDRO FABIÁN TEATINO SUÁREZ

**UNIVERSIDAD PONTIFICA BOLIVARIANA
FACULTAD DE INGENIERÍA ELECTRONICA
SECCIONAL BUCARAMANGA
2018**

**DISEÑO E IMPLEMENTACIÓN DE UN EQUIPO DE DIAGNÓSTICO Y
VISUALIZACIÓN DE LAS SEÑALES ANALÓGICAS Y DIGITALES EN
LOS VARIADORES DE FRECUENCIA ABB DE LAS LÍNEAS ACS550,
ACS800, ACS850, ACS880 PARA LA EMPRESA COL CONTROLES &
INTEGRACIÓN S.A.S.**

**ESTUDIANTE
PEDRO FABIÁN TEATINO SUÁREZ**

PROYECTO DE GRADO

**DIRECTOR
PhD. OMAR PINZÓN ARDILA**

**UNIVERSIDAD PONTIFICA BOLIVARIANA
FACULTAD DE INGENIERÍA ELECTRÓNICA
SECCIONAL BUCARAMANGA
2018**

CONTENIDO

1	INTRODUCCIÓN	14
2	OBJETIVOS	16
2.1	OBJETIVO GENERAL	16
2.2	OBJETIVOS ESPECIFICOS	16
3	MARCO TEÓRICO	17
3.1	FUNDAMENTOS TEÓRICOS.....	17
3.1.1	Variador de frecuencia.....	17
3.1.2	Motor.....	18
3.1.3	Tarjeta de desarrollo Arduino.....	19
3.1.4	Antecedentes	20
4	METODOLOGÍA	21
5	CARACTERIZACION DE LOS VARIADORES DE VELOCIDAD ABB22	
6	LAS CARACTERISTICAS ELECTRICAS PRINCIPALES DE LAS TARJETAS DE CONTROL.....	23
7	DISEÑO E IMPLEMENTACIÓN DEL ACONDICIONAMIENTO DE SEÑALES.....	24
7.1	ETAPAS DE DESARROLLO	24
7.1.1	TARJETA DE FUENTES REGULADAS	27
7.2	TARJETA DE SEÑALES.....	31
7.3	TARJETA DE SIMULACIÓN DE ENCODER	36
7.4	TARJETA DE PRUEBA DE SEMICONDUCTORES.....	39
8	DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE.....	45
9	CONCLUSIONES	71
10	BIBLIOGRAFÍA	72
11	ANEXOS	73
11.1	Anexo No 1: Características Generales de los variadores ABB..	73
11.1.1	ACS550 Convertidores de frecuencia ABB de propósito general	73
	Aplicaciones	73
	Características principales	73

Datos técnicos.....	74
11.1.2 ACS800 Convertidores de frecuencia industriales.....	75
Convertidores de frecuencia montados en pared, ACS800-01	76
Convertidores regenerativos montados en pared, ACS800-11	77
Convertidores montados en pared para armónicos reducidos, ACS800-31	78
Convertidores de frecuencia autoportantes, ACS800-02	79
Convertidores de frecuencia instalados en armario, ACS800-07	80
Convertidores de frecuencia con refrigeración líquida, ACS800-07LC	80
Especificaciones técnica.	82
11.1.3 Convertidores de frecuencia ABB para maquinaria ACS850	83
Amplia gama de opciones	83
Diseño robusto	83
Especificaciones técnicas	84
11.1.4 Convertidores de frecuencia industriales ABB- ACS880	85
Convertidores de frecuencia únicos montados en pared, ACS880-01	85
Convertidores de frecuencia únicos instalados en armario, ACS880-07	86
11.2 Anexo No. 2. CARACTERISTICAS ELÉCTRICAS DE TARJETAS ELECTRÓNICAS DE CONTROL DE LOS VARIADORES ABB.....	88
11.2.1 TARJETA DE CONTROL ACS550	88
11.2.2 TARJETA DE CONTROL ACS800	89
11.2.3 TARJETA DE CONTROL ACS850	92
11.2.4 TARJETA DE CONTROL ACS880	95
11.3 Anexo No. 3 CÓDIGO DE PROGRAMACIÓN	97
11.3.1 Código Inclusión de las librerías a utilizar	97
11.3.2 Código Tamaño de fuente	97
11.3.3 Código Configuración de modelo de pantalla	97
11.3.4 Código Declaración de constantes globales	97
11.3.5 Código recuadro que indica selección	98

11.3.6	Código menú principal	98
11.3.7	Código Encoder principal	100
11.3.8	Código Encoder 512 ppr	102
11.3.9	Código 3 velocidades 512 ppr horario	104
11.3.10	Código 512 ppr 600 rpm en sentido horario	106
11.3.11	Código ON 512 ppr 600 rpm en sentido horario.....	108
11.3.12	Código 512 ppr 1800 rpm en sentido horario	108
11.3.13	Código ON 512 ppr 1800 rpm en sentido horario.....	110
11.3.14	Código 512 ppr 3000 rpm en sentido horario	111
11.3.15	Código ON 512 ppr 3000 rpm en sentido horario.....	113
11.3.16	Código 3 velocidades 512 ppr horario.....	114
11.3.17	Código 512 ppr 600 rpm en sentido antihorario	116
11.3.18	Código ON 512 ppr 600 rpm en sentido antihorario	118
11.3.19	Código 512 ppr 1800 rpm en sentido antihorario	118
11.3.20	Código ON 512 ppr 1800 rpm en sentido antihorario	120
11.3.21	Código 512 ppr 3000 rpm en sentido antihorario	121
11.3.22	Código ON 512 ppr 3000 rpm en sentido antihorario	123
11.3.23	Código Encoder 1024 ppr	124
11.3.24	Código 3 velocidades 1024 ppr horario.....	125
11.3.25	Código 1024 ppr 300 rpm en sentido horario	127
11.3.26	Código ON 1024 ppr 300 rpm en sentido horario.....	129
11.3.27	Código 1024 ppr 900 rpm en sentido horario	130
11.3.28	Código ON 1024 ppr 900 rpm en sentido horario.....	132
11.3.29	Código 1024 ppr 1500 rpm en sentido horario	133
11.3.30	Código ON 1024 ppr 1500 rpm en sentido horario.....	135
11.3.31	Código 3 velocidades 1024 ppr antihorario	135
11.3.32	Código 1024 ppr 300 rpm en sentido antihorario	137
11.3.33	Código ON 1024 ppr 300 rpm en sentido antihorario	139
11.3.34	Código 1024 ppr 900 rpm en sentido antihorario	140
11.3.35	Código ON 1024 ppr 900 rpm en sentido antihorario	142

11.3.36	Código 1024 ppr 1500 rpm en sentido antihorario	143
11.3.37	Código ON 1024 ppr 1500 rpm en sentido antihorario ...	145
11.3.38	Código Prueba semiconductores	145
11.3.39	Código Prueba tiristor.....	147
11.3.40	Código Prueba de IGBT	150
11.3.41	Código señal digital	156
11.3.42	Código Fuentes	158
11.3.43	Código salida digital	159
11.3.44	Código salida rele.....	161
11.3.45	Código Leer señal	163
11.3.46	Código Señal de drive	165
11.3.47	Código Escribir señal	165
11.3.48	Código Señal de salida auto.....	167
11.3.49	Código Señal auto mA	169
11.3.50	Código Suma mA	170
11.3.51	Código Resta mA	171
11.3.52	Código Pause suma mA.....	172
11.3.53	Código Pause resta mA	173
11.3.54	Código Señal auto V	174
11.3.55	Código Suma V	175
11.3.56	Código Resta V	177
11.3.57	Código Pause suma V.....	178
11.3.58	Código Pause resta V	179
11.3.59	Código Señal manual	180
11.3.60	Código Señal manual V.....	182
11.3.61	Código Señal up down V	183
11.3.62	Código Señal manual mA.....	185
11.3.63	Código Señal up down mA.....	186
11.3.64	Código Configuración de pines	188
11.3.65	Código Bienvenida	188

INDICE DE FIGURAS

Figura No. 1 Diagrama de bloques de un variador de frecuencia.	17
Figura No. 2 Interior de un motor de inducción.	19
Figura No. 3 Tarjeta de desarrollo Arduino DUE	19
<i>Figura No. 4 Diseño de bloques del equipo</i>	21
Figura No. 5 Diagrama de flujo. Procedimiento para la implementación del proyecto	25
Figura No. 6 Regulador de voltaje con proteccion de diodos.	27
Figura No. 7 Vista superior de modelo 3D en KICAD de tarjeta de fuentes reguladas.	29
Figura No. 8 Vista isométrica de modelo 3D en KICAD de tarjeta de fuentes reguladas.	29
Figura No. 9 Top o mascararas de PCB tarjeta de fuentes reguladas desarrollada en KICAD.....	30
Figura No. 10 Button de PCB tarjeta de fuentes reguladas desarrollada en KICAD.	30
Figura No. 11 Vista superior de tarjeta fuente terminada.	31
Figura No. 12 Análisis del circuito no inductor.....	32
Figura No. 13 Vista superior de modelo 3D en KICAD de tarjeta de señales.	34
Figura No. 14 Vista isométrica de modelo 3D en KICAD de tarjeta de señales.....	34
Figura No. 15 Top o mascararas de PCB tarjeta de señales desarrollada en KICAD.	35
Figura No. 16 Button de PCB tarjeta de señales desarrollada en KICAD.	35
Figura No. 17 Vista superior de tarjeta de señales terminada.....	36
Figura No. 18 Vista superior de modelo 3D en KICAD de tarjeta de simulación de encoder.	37
Figura No. 19 Vista isométrica de modelo 3D en KICAD de tarjeta de simulación de encoder.	37
Figura No. 20 Top o mascararas de PCB tarjeta de señales desarrollada en KICAD.	38

Figura No. 21 Button de PCB tarjeta de simulación de encoder desarrollada en KICAD.	38
Figura No. 22 Vista superior de tarjeta Simulación de Encoder Terminada.	39
Figura No. 23 Vista superior de modelo 3D en KICAD de tarjeta de pruebas de semiconductores.	40
Figura No. 24 Vista isométrica de modelo 3D en KICAD de tarjeta de pruebas de semiconductores.	41
Figura No. 25 Top o mascararas de PCB tarjeta de prueba de semiconductores desarrollada en KICAD.....	41
Figura No. 26 Button de PCB tarjeta de prueba de semiconductores desarrollada en KICAD.....	42
Figura No. 27 Vista superior de tarjeta de pruebas de semiconductores. .	42
Figura No. 28 Envolverte del equipo de pruebas.....	43
Figura No. 29 Equipo de pruebas terminado.....	44
Figura No. 30 Pantalla de bienvenida.	45
Figura No. 31 Pantalla de menú principal.	46
Figura No. 32 Pantalla encoder principal	46
Figura No. 33 Sentido de giro.....	47
Figura No. 34 Velocidades 512 ppr	47
Figura No. 35 Velocidades 1024 ppr	48
Figura No. 36 Encoder 512ppr horario 600rpm.....	48
Figura No. 37 Generacion señal encoder 51 ppr horario 600rpm	48
Figura No. 38 Encoder 512ppr antihorario 600rpm	49
Figura No. 39 Generacion señal encoder 512ppr antihorario 600rpm.....	49
Figura No. 40 Encoder 512ppr horario 1800rpm.....	50
Figura No. 41 Generacion señal encoder 512ppr horario 1800rpm.	50
Figura No. 42 Encoder 512ppr antihorario 1800rpm.	51
Figura No. 43 Generacion señal encoder 512ppr antihorario 1800rpm.....	51
Figura No. 44 Encoder 512ppr horario 3000rpm	52
Figura No. 45 Generacion señal encoder 512ppr horario 3000rpm	52
Figura No. 46 Encoder 512ppr antihorario 3000rpm	53

Figura No. 47 Generacion señal encoder 512ppr antihorario 3000rpm.....	53
Figura No. 48 Encoder 1024ppr horario 300rpm.....	54
Figura No. 49 Generacion señal encoder 1024ppr horario 300rpm.....	54
Figura No. 50 Encoder 1024ppr antihorario 300rpm.....	55
Figura No. 51 Generacion señal encoder 1024ppr antihorario 300rpm.....	55
Figura No. 52 Encoder 1024ppr horario 900rpm.....	56
Figura No. 53 Generacion señal encoder 1024ppr horario 900rpm.....	56
Figura No. 54 Encoder 1024ppr antihorario 900rpm.....	57
Figura No. 55 generacion señal encoder 1024ppr antihorario 900rpm.....	57
Figura No. 56 Encoder 1024ppr horario 1500rpm.....	58
Figura No. 57 Generacion señal encoder 1024ppr horario 1500rpm.....	58
Figura No. 58 Encoder 1024ppr antihorario 1500rpm.....	59
Figura No. 59 Generacion señal encoder 1024ppr antihorario 1500rpm...	59
Figura No. 60 Selección de señal.....	60
Figura No. 61 Generacion de señal.....	61
Figura No. 62 Play-pause de señal automática V.....	61
Figura No. 63 Play-pause de señal automática mA.....	61
Figura No. 64 Señal de salida automática.....	62
Figura No. 65 Señal de salida automática con pausas.....	62
Figura No. 66 Arriba-abajo voltaje.....	62
Figura No. 67 Arriba-abajo corriente.....	63
Figura No. 68 Señal de salida manual.....	63
Figura No. 69 Señal digital.....	64
Figura No. 70 Fuentes.....	64
Figura No. 71 On-off señal digital.....	65
Figura No. 72 Leer señal.....	65
Figura No. 73 Pantalla prueba de semiconductores.....	66
Figura No. 74 Pantalla Tiristor.....	66
Figura No. 75 Tiristor_1.....	67
Figura No. 76 Tiristor_2.....	67

Figura No. 77 Tiristor_3.....	67
Figura No. 78 Generacion de prueba tiristor	68
Figura No. 79 Tiristor OK.....	68
Figura No. 80 Tiristor Falla.....	68
<i>Figura No. 81 Prueba de IGBT.....</i>	<i>69</i>
Figura No. 82 Generacion prueba de IGBT	69
Figura No. 83 IGBT OK	70
Figura No. 84 IGBT Falla	70
Figura No. 85 Variador ACS800 para montaje en pared.....	77
Figura No. 86 Variador ACS800 regenerativo para montaje en pared	78
Figura No. 87 Variador ACS800 bajo armónicos monatado en pared.....	79
<i>Figura No. 88 Variador ACS800 Autoportante</i>	<i>79</i>
<i>Figura No. 89 Variador ACS800 instalado en armario</i>	<i>80</i>
Figura No. 90 Variador ACS800 con refrigeración líquida.....	81
<i>Figura No. 91 Variador ACS850 para maquinaria</i>	<i>84</i>
Figura No. 92 Convertidores ACS880 montados en pared.	86
Figura No. 93 Convertidores ACS880 instalados en armario.....	87

INDICE DE TABLAS

Tabla 1: Tabla de terminales de control ACS550	88
Tabla 2: Tabla de terminales de control ACS550	89
Tabla 3 Especificaciones de la tarjeta RMIO.....	89
Tabla 4: Tabla de terminales de control ACS800	91
Tabla 5: Tabla de terminales de control ACS850	92
Tabla 6 Especificaciones de la tarjeta JCU	93
Tabla 7: Tabla de terminales de control ACS880	95

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: DISEÑO E IMPLEMENTACIÓN DE UN EQUIPO DE DIAGNÓSTICO Y VISUALIZACIÓN DE LAS SEÑALES ANALÓGICAS Y DIGITALES EN LOS VARIADORES DE FRECUENCIA ABB DE LAS LÍNEAS ACS550, ACS800, ACS850, ACS880 PARA LA EMPRESA COL CONTROLES & INTEGRACIÓN S.A.S.

AUTOR(ES): Pedro Fabián Teatino Suárez

PROGRAMA: Facultad de Ingeniería Electrónica

DIRECTOR(A): PhD. Omar Pinzón Ardila

RESUMEN

El diagnóstico temprano para las señales analógicas y digitales de los variadores de frecuencia ABB de las series ACS550, ACS800, ACS850 y ACS880 es una herramienta innovadora que suple la necesidad existente en la empresa COL CONTROLES & INTEGRACIÓN SAS, porque realiza un diagnóstico en campo, del variador sin necesidad de traslado, lo cual reduce el tiempo de toma de decisiones.

PALABRAS CLAVE:

Adquisición señales, Arduino, detección fallas, variadores frecuencia, visualización señales.

V° B° DIRECTOR DE TRABAJO DE GRADO

GENERAL SUMMARY OF WORK OF GRADE

TITLE: DESIGN AND IMPLEMENT AN EARLY DIAGNOSIS EQUIPMENT FOR THE ANALOG AND DIGITAL SIGNALS OF THE ABB FREQUENCY CONVERTERS OF THE ACS550, ACS800, ACS850 AND ACS880 SERIES FOR THE COMPANY COL CONTROLES & INTEGRACIÓN S.A.S.

AUTHOR(S): Pedro Fabián Teatino Suárez

FACULTY: Facultad de Ingeniería Electrónica

DIRECTOR: PhD. Omar Pinzón Ardila

ABSTRACT

The early diagnosis for the analog and digital signals of the ABB frequency inverters of the ACS550, ACS800, ACS850 and ACS880 series is an innovative tool that meets the existing need in the company COL CONTROLES & INTEGRACIÓN SAS, because it makes a diagnosis in the field, of the inverter without the need for transfer, which reduces the decision-making time.

KEYWORDS:

Acquisition signals, Arduino, fault detection, frequency variators, visualization signals.

V° B° DIRECTOR OF GRADUATE WORK

1 INTRODUCCIÓN

La inclusión de los motores de corriente alterna en la industria, ha facilitado de muchas formas el proceso de producción y distribución de bienes y servicios, pero con el transcurrir del tiempo surgieron otro tipo de necesidades para obtener una mejor calidad del producto y un mayor desarrollo en cuanto a la eficiencia industrial.¹ Allí se empieza a desarrollar la idea de mantener el control sobre los motores que involucran el proceso. En lugar de controlar el proceso con sistemas mecánicos que son ineficientes y presentan grandes pérdidas energéticas y desperdicios de material, controlar el comportamiento del motor para que se adecúe a las necesidades de torque y velocidad del proceso.²

Con la inclusión de la automatización industrial y el desarrollo de nuevos métodos y procesos en las empresas, se hizo casi que indispensable la instalación de los variadores de frecuencia con el fin de mantener un control más preciso y dinámico en la industria.

Esta tecnificación mejoro la calidad e hizo más competitivo el mercado, pero dicha incorporación de dispositivos y maquinaria, también hizo necesario un acompañamiento técnico especializado con el fin de alargar y mejorar la vida útil de dichos mecanismos. Este servicio técnico para buen funcionamiento de los variadores de frecuencia producidos por la empresa ABB, es el que provee y en el que se enfoca la empresa Col Controles & Integración en la ciudad de Bogotá.

Con el desarrollo e implementación de este proyecto, se quiere brindar un servicio más rápido y eficiente a las empresas que lo solicitan; buscando reducir el tiempo de diagnóstico de un variador en campo, lo cual se traduce en un ahorro económico para las empresas que lo solicitan, debido a que su planta de producción o proceso involucrado estará improductivo un lapso mucho menor al utilizado normalmente por el personal de servicio, ya que se podría diagnosticar el equipo directamente en planta evitando tiempos muertos por transporte del mismo a las instalaciones del laboratorio de Col Controles. Este ahorro de tiempo y costos se ve directamente reflejado tanto en el cliente como en la empresa prestadora del servicio.

¹ WEG. *Motores Eléctricos. Guía de Especificaciones.* [En Línea]. <http://ecatalog.weg.net/files/wegnet/WEG-guia-de-especificacion-50039910-manual-espanol.pdf> [Tomado en 2015].

² CENTRO NACIONAL DE INVESTIGACION Y DESARROLLO TECNOLÓGICO. México. *Documento de grado.* [En Línea]. http://www.cenidet.edu.mx/subaca/web-elec/tesis_mc/220MC_carl.pdf. [Tomado en 2015].

2 OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar e Implementar un equipo de diagnóstico temprano para las señales analógicas y digitales de los variadores de frecuencia ABB de las series ACS550, ACS800, ACS850 y ACS880.

2.2 OBJETIVOS ESPECIFICOS

- Documentar las diferentes series en las que se implementará el sistema de detección de fallas tanto en el funcionamiento como en los módulos de comunicación.
- Caracterizar los diferentes variadores de frecuencia con el fin de encontrar similitudes y diferencias entre las distintas series.
- Diseñar e implementar el acondicionamiento de señales que del variador se entregan al sistema de detección.
- Diseñar e implementar el software para el sistema.

3 MARCO TEÓRICO

3.1 FUNDAMENTOS TEÓRICOS

3.1.1 Variador de frecuencia

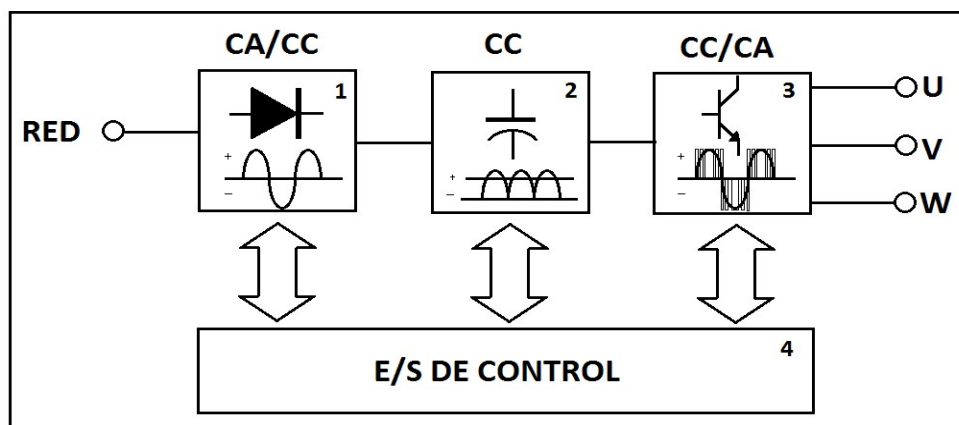
El variador de frecuencia, que también es llamado inversor o DRIVES, es un tipo especial de variadores de velocidad que permiten un completo y preciso control de los motores de inducción de corriente ac.³

Este dispositivo está comprendido o formado por 4 bloques principales que se muestran en la figura 1 y se especifican a continuación.

El bloque 1 es el rectificador, el cual se encarga de convertir el suministro de alimentación ac monofásico o trifásico en una señal cc por medio de un arreglo de diodos.

El bloque 2 es denominado el bus de continua, donde se almacena y filtra la señal rectificada con el fin de hacer estable su valor y mantener almacenada la energía necesaria para el funcionamiento del motor, está conformada por una serie de condensadores, y algunas veces bobinas.

Figura No. 1 Diagrama de bloques de un variador de frecuencia.



Fuente propia

³ SIEMENS INDUSTRY. Variadores de velocidad SINAMICS. [EnLínea]. <https://www.industry.siemens.com/home/aan/es/ecuador/Documents/Articulo%20variador%20de%20velocidad%20Siemens.pdf>. [Tomado en 2015].

El bloque 3 es la salida de la señal del variador hacia el motor, el cual hace una nueva conversión de señales, pero ahora pasa de ser cc a ac por medio de una modulación PWM en la cual se obtiene un buen control de la frecuencia, tensión e intensidad entrante al motor.

El bloque 4 es el encargado del control de los 3 bloques anteriores, y también de la protección, regulación y otros aspectos de las entradas y salidas de las señales del variador. Incluyendo también una interfaz de comunicación con el usuario y otros dispositivos como lo puede ser el encoder.

3.1.2 Motor

El motor eléctrico es una máquina que se encarga de transformar la energía eléctrica en energía mecánica. El motor de inducción es el elemento electromecánico más utilizado en la industria para un sinnúmero de procesos.⁴

La utilización de los motores de inducción ha ido en acenso en las últimas décadas, debido a que es una máquina simple, resistente, económica y fáciles de mantener. La velocidad de trabajo de motor depende de la frecuencia de alimentación que recibe, y con este concepto nace también la idea de mantener un control sobre él.

Los motores de inducción trifásicos están comprendidos de 2 partes principales, que son el estator y el rotor giratorio⁵, junto a otros elementos adicionales que se pueden ver en la figura 2.

El estator consta de una armazón de acero que soporta un núcleo cilíndrico hueco y de varias ranuras equidistantes entre sí, hechas en la circunferencia interna para adicionar su devanado. El rotor también posee una serie de ranuras especiales y está hecho de una serie de láminas que los conforman, sobre sus ranuras se encuentra su devanado.

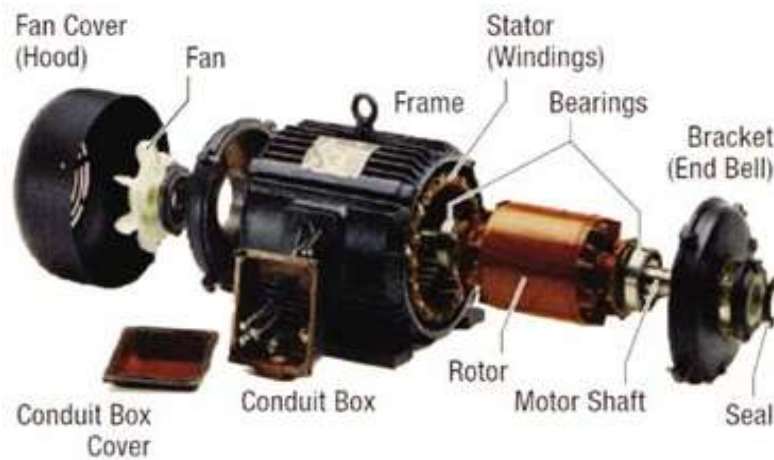
El funcionamiento de este tipo de motor está basado en la producción de un campo magnético rotatorio en el estator, luego de que reciba excitación. Este campo genera una serie de corrientes en el rotor, que al interactuar con dicho campo, producen un par mecánico.

⁴ UNIVERSIDAD TECNOLÓGICA NACIONAL. Argentina. Facultad de ingeniería eléctrica. [En Línea].

http://www.fra.utn.edu.ar/download/carreras/ingenierias/electrica/materias/planestudio/quin_tonivel/electronicall/apuntes/variadores_de_frecuencia.pdf. [Tomado en 2015].

⁵ ¿Que es Arduino? [En Línea]. <https://www.arduino.cc/en/Guide/Introduction>. [Tomado en 2015].

Figura No. 2 Interior de un motor de inducción.



Fuente: <http://www.emb.cl/electroindustria/articulo.mvc?xid=1411&tip=7>

3.1.3 Tarjeta de desarrollo Arduino

Es una placa electrónica simple con su propio entorno de desarrollo y su control se ejerce mediante un código de programación abierto.

Figura No. 3 Tarjeta de desarrollo Arduino DUE



Fuente: <http://www.seeedstudio.com/depot/Arduino-Due-p-1333.html>

El Arduino es principalmente una tarjeta de desarrollo o computación física, que significa que mediante su implementación se puede construir sistemas interactivos relacionando software y hardware. Dentro de sus componentes principales se encuentra un procesador y periféricos de entradas y salidas de señales analógicas y digitales dependiendo de su modelo. Estos periféricos son diseñados para poder ejercer comunicación con gran

cantidad de sensores y dispositivo. En la figura No 3 se puede ver la disposición de una placa de desarrollo Arduino DUE similar a la utilizada para el desarrollo de este proyecto.

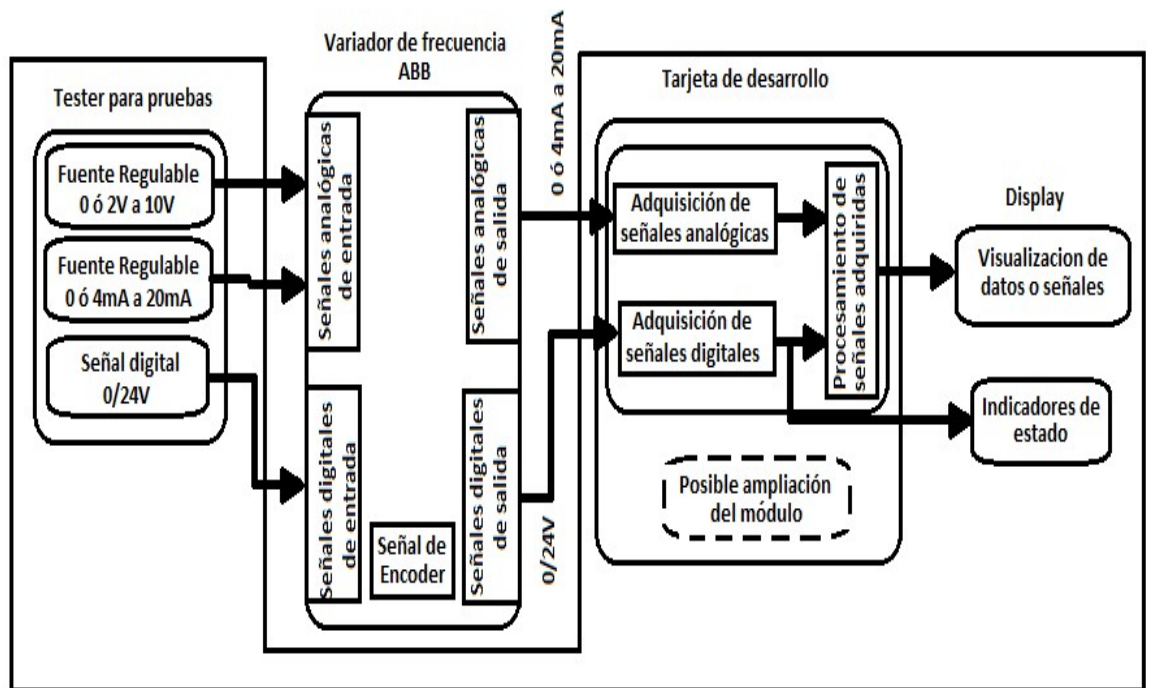
3.1.4 Antecedentes

En la búsqueda de instrumentos o módulos de diagnóstico para variadores de velocidad ABB, la única información que se maneja es del programa que facilita el proveedor llamado *DriveWindows*, que hace una monitorización de señales, pero no integra totalmente los parámetros y tampoco es aplicable para todos los variadores debido a que se comunica con ellos mediante el sistema de control distribuido, por lo cual muchas veces se dificulta el servicio técnico por no tener acceso total al sistema, En otra serie de variadores su comunicación es mediante fibra óptica e implementos que se tienen únicamente en laboratorio y no permiten su traslado a campo.

4 METODOLOGÍA

Este proyecto se basó en investigación aplicada y experimental, donde se hizo la descripción del equipo de diagnóstico, para las diversas fallas que pueden presentar los variadores de frecuencia ABB en las entradas y salidas analógicas y digitales, mediante la experimentación en el laboratorio de Col Controles y la documentación proporcionada por ABB quienes son los fabricantes. Adicionalmente se realizó la investigación respectiva de los módulos y elementos electrónicos que se utilizaron para el desarrollo del equipo de diagnóstico.

Figura No. 4 Diseño de bloques del equipo



Fuente: Propia

5 CARACTERIZACION DE LOS VARIADORES DE VELOCIDAD ABB

Para ser lo más asertivos frente a la documentación de las señales analógicas y digitales de los variadores de frecuencia ABB de las series ACS550, ACS800, ACS850 y ACS880, fue necesario acudir a los manuales del usuario y catálogos técnicos de cada convertidor de frecuencia, los cuales se exponen a continuación y de manera general en los anexos.

ABB como fabricante de variadores hace una división de su producto de baja tensión de acuerdo al ambiente de trabajo, estas divisiones son llamada variadores de:

Propósito general llamados así por su fácil instalación y puesta en servicio (anexo “ACS550 Convertidores de frecuencia ABB de propósito general”).

Variadores para maquinaria que se caracterizan por la versatilidad en su programación mediante las cuales puede cubrirse las exigencias específicas solicitada por diferentes máquinas (anexo “ACS850 Convertidores de frecuencia ABB para maquinaria”).

Variadores de frecuencia industriales que son flexibles en su programación, pero también están diseñados para soportar los ambientes más agresivos teniendo un excelente desempeño en el control a motor, también tienen características de construcción especiales que debes pedir como adicionales de fábrica dentro de las cuales se tienen equipos de bajo armónicos y regenerativos. En esta división se encuentran los equipos ACS800 (anexo “ACS800 Convertidores de frecuencia industriales”) y ACS880 (anexo “ACS880 Convertidores de frecuencia industriales”).

6 LAS CARACTERISTICAS ELECTRICAS PRINCIPALES DE LAS TARJETAS DE CONTROL

Luego de realizar la revisión en cada uno de los manuales y documentos de los variadores, los cuales facilitó el fabricante ABB, se puede ver que en los terminales de entradas analógicas todos los equipos manejan una impedancia de $200\text{ k}\Omega$ cuando se configura para la recepción de señal de voltaje, y una impedancia de $100\ \Omega$ cuando la entrada se configura en corriente, por lo cual se estandarizan estos valores de impedancia para el desarrollo de cálculos en el equipo de pruebas.

La carga que se conecta en los terminales de salidas analógicas, los cuales entregan señales de corriente $0(4)$ a 20 mA , ABB hace salvedad que debe ser una impedancia menor a $500\ \Omega$ en todos los variadores.

Las entradas digitales reciben un voltaje de 24 V , dejando como estándar los niveles lógicos de la siguiente manera: "0" $< 5\text{ V}$ y "1" $> 15\text{ V}$. En todos los variadores pueden ser configurables como PNP o NPN para recibir señal de fuente interna del drive o una señal 24 V externa.

Las salidas relé de los variadores ABB soportan voltaje de corriente continua o alterna, pero en el equipo de pruebas se hará pruebas de conmutación de las salidas relé con voltaje de 24 V en corriente continua y una corriente menor a 500 mA para garantizar que pueda conmutar sin inconvenientes. Toda esta información descrita anteriormente puede ser ampliada y revisada en los anexos "Tarjetas de control variadores ABB".

7 DISEÑO E IMPLEMENTACIÓN DEL ACONDICIONAMIENTO DE SEÑALES

7.1 ETAPAS DE DESARROLLO

En el Diagrama de flujo 1. Se pudo observar el procedimiento propuesto para llevar a buen término el desarrollo del proyecto, basado en la información obtenida de los variadores a diagnosticar, e intentando hacer el mejor uso de la información suministrada por los desarrolladores de los variadores, en este caso ABB.

El control principal es basado en la placa de desarrollo Arduino, para la cual se hizo una realimentación de conceptos para generar una fácil y mejor programación, que permite que el módulo se acople lo mejor posible a las necesidades de la empresa.

I. Necesidad de la empresa

La empresa Col Controles busca hacer mucho más eficiente su proceso de servicio en campo, buscando reducir al máximo los tiempos de diagnóstico de los variadores, para así aumentar su productividad tanto en la empresa, como en la planta de su cliente.

II. Documentación sobre los variadores a trabajar

Col Controles tiene acceso a los manuales de los variadores de frecuencia de la marca ABB, los cuales facilitan para poder tener una mirada más amplia sobre el tipo de señales a trabajar, y la forma en cómo se debe implementar el instrumento de diagnóstico.

III. Familiarización con los variadores y su trabajo en campo

Luego de tener la información respectiva del funcionamiento, del software y del hardware de los variadores, se harán visitas a campo para observar de primera mano cómo trabajan y la forma tradicional en la que se hace el servicio de diagnóstico.

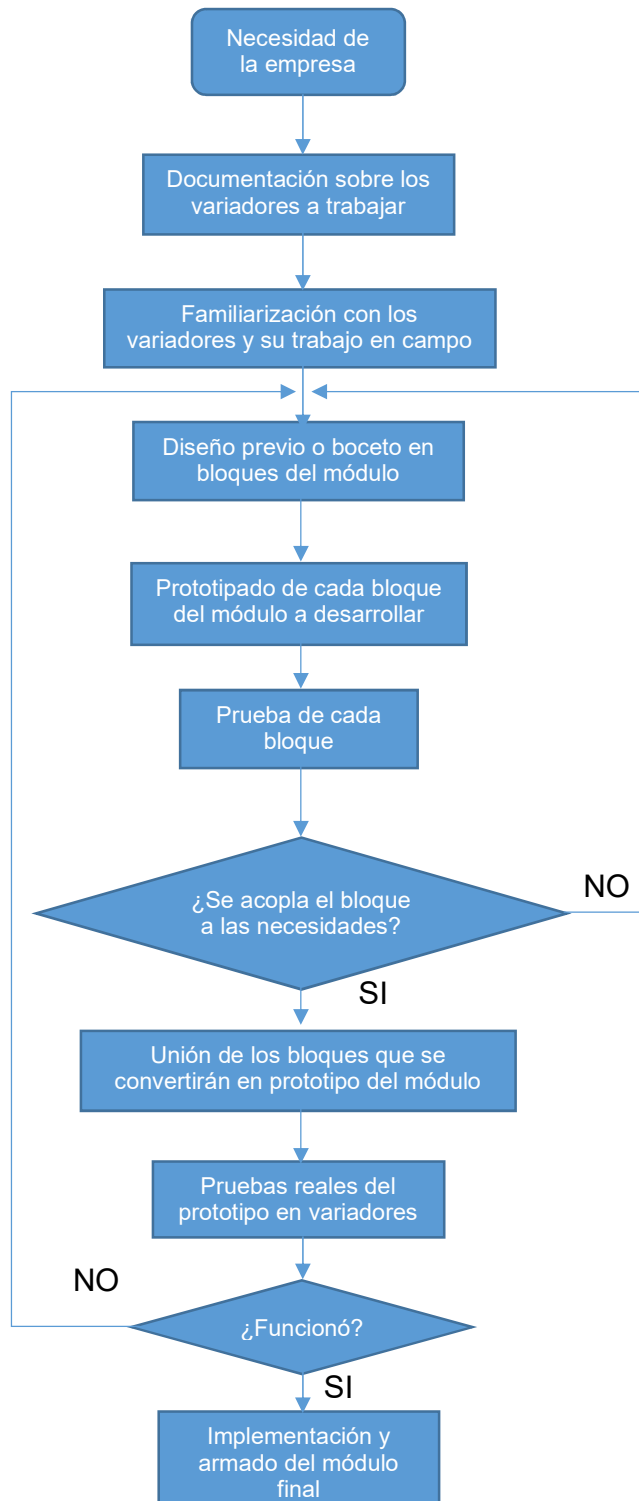
IV. Diseño previo o boceto en bloques del módulo

Teniendo una base de cómo trabaja el servicio de la forma tradicional y observando las señales y el tipo de prueba como se diagnostica, se hace un boceto del módulo a desarrollar, con el fin de saber que necesitamos y de qué manera trabajaremos el módulo, con el fin de hacerlo lo más fiel posible al diagnóstico hecho de forma tradicional.

V. Prototipado de cada bloque del módulo a desarrollar

Como el módulo está dividido en bloques de fuente de señales o tester, alimentación del módulo, acople de señales de salida del variador, bloque de procesamiento de señales y visualización de datos y señales de interés.

Figura No. 5 Diagrama de flujo. Procedimiento para la implementación del proyecto



Fuente: Propia

- VI. Prueba de cada bloque desarrollado**
Cada bloque se prueba por separado, con el fin de observar que cumple su función esencial sin llegar a afectar los demás bloques por algún error de señal, con el fin de no retrasar el trabajo adelantado.
- VII. ¿Se acopla el bloque a las necesidades?**
De acuerdo al resultado de las pruebas efectuadas a cada bloque, se toma la decisión si el bloque es idóneo para su función, de lo contrario se vuelve a la fase de diseño previo o boceto en bloques del módulo con el fin de desarrollar una mejor solución.
- VIII. Unión de los bloques que se convertirán en el prototipo del módulo**
Luego de comprobar el correcto funcionamiento de cada bloque del módulo, se procede a unirlos para que el prototipo del módulo quede completo en una pieza.
- IX. Pruebas reales del prototipo en variadores**
El módulo se conecta a un variador para iniciar un diagnóstico de fallas, haciendo una comprobación de algunas señales de la forma tradicional como se trabaja actualmente el servicio.
- X. Funciona**
Según el resultado obtenido de las pruebas con los variadores, se decide si el prototipo sufre las necesidades para convertirlo en el módulo real, o si el proyecto vuelve a la etapa de diseño.
- XI. Implementación y armado del módulo final**
El prototipo desarrollado para las pruebas, lo rearmamos con las especificaciones finales de diseño de hardware para obtener el producto final.

El circuito o hardware del dispositivo está dividido en 4 grandes partes que a su vez las convertimos en 4 PCB. El motivo principal por el cual se hizo dicha división es para reducir costos en caso de falla de alguna parte del dispositivo porque se haría el cambio de la tarjeta que contiene el componente dañado o reemplazarlo por otra tarjeta igual para verificar su funcionamiento.

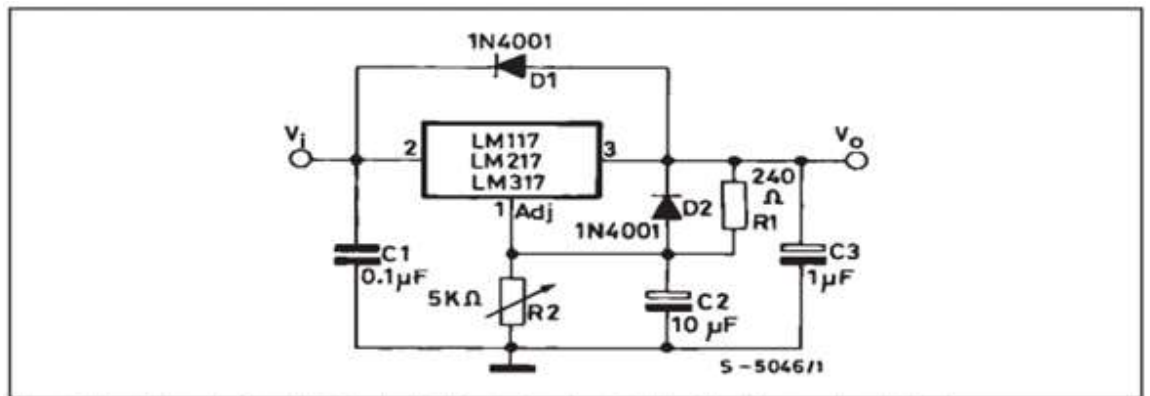
Las 4 tarjetas o PCB en las cuales se divide el hardware son llamadas: Tarjeta de fuentes reguladas, Tarjeta de señales, Tarjeta de simulación de encoder y tarjeta de prueba de semiconductores. Las mismas van conectadas entre sí por medio de la tarjeta de desarrollo Arduino Due la cual a su vez se conecta a una pantalla táctil mediante un shield. La pantalla táctil de 7" hará las veces de HMI por sus siglas en inglés (Human Machine Interface).

7.1.1 TARJETA DE FUENTES REGULADAS

Es la encargada de tomar la alimentación de 24 V de corriente directa que provee una fuente y genera por medio de dos LM317 con su respectivo arreglo de resistencias una fuente de 12 V de corriente directa y una fuente de 9 V de corriente directa, adicionalmente tiene conectores en paralelo a la fuente principal de 24 voltios para obtener más puntos de alimentación 24 V los cuales se necesitan para algunas pruebas. La fuente 12 VDC se usa principalmente para alimentar los circuitos integrados en las otras tarjetas como lo son el LM358 y la compuerta 7404. La fuente 9 VDC se usa principalmente para producir los disparos en la puerta de los dispositivos de potencia a probar como lo son los IGBT y tiristores.

Se selecciona el dispositivo LM317 para control de fuente regulada debido a que cuenta con un rango de voltaje a la salida bastante amplio ($V_{in}-V_{out}=40V$), corriente de salida de 1,5 Amperios, temperatura de trabajo de 0 °C a 150 °C y para tal fin tiene una forma de configuración bastante básica y sencilla. Los condensadores son utilizados para tener una mayor estabilidad en los voltajes de entrada y salida. Los diodos generan protección y ayudan a descargar los condensadores, el diseño a utilizar en cada una de las fuentes está basado en la figura No. 6 y los componentes utilizados son calculados bajo la ecuación No. 1 que provee el fabricante del LM317 en su datasheet.

Figura No. 6 Regulador de voltaje con protección de diodos.



Fuente:

<http://pdf1.alldatasheet.com/datasheetpdf/view/22749/STMICROELECTRONICS/LM317.html>

$$V_o = V_{ref} \left(1 + \frac{R_2}{R_1} \right) + I_{Adj} R_2 \quad (1)$$

Para la fuente de 12 V y 9 V se toman como valores estándares basados en las indicaciones del fabricante del LM317 lo siguiente:

$$R_1 = 220 \Omega; V_{ref} = 1,25 V; V_o = 12 V; I_{Adj} = 100 \mu A$$

La corriente I_{Adj} es bastante pequeña para tenerla en cuenta, por lo cual al despejar tendremos la ecuación No 2.

$$R_2 = \left(\frac{R_1}{V_{ref}} \right) (V_{out} - V_{ref}) \quad (2)$$

Al resolver la ecuación No 2 para la fuente con voltaje de salida de 12 V, obtenemos el siguiente valor de resistencia R_2 .

$$\Rightarrow R_2 = \left(\frac{220}{1,25} \right) (12 - 1,25) \Rightarrow R_2 = 1892 \Omega$$

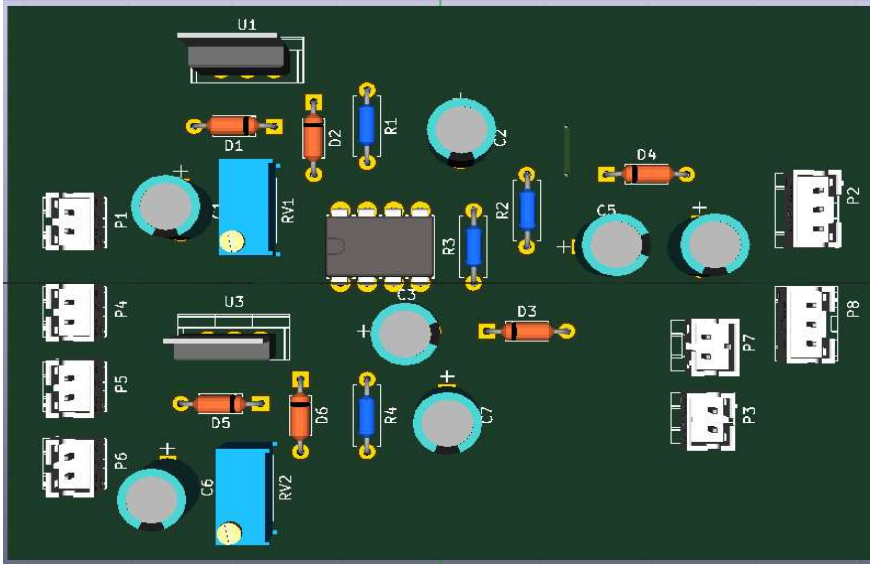
De igual manera para la fuente con voltaje de salida de 9 V, se resuelve la ecuación No 2 y se obtiene el valor respectivo de resistencia para la generación de los 9 V.

$$\Rightarrow R_2 = \left(\frac{220}{1,25} \right) (9 - 1,25) \Rightarrow R_2 = 1364 \Omega$$

Luego de obtener los valores de resistencias necesarios para generar los 12 y 9 voltios respectivamente, se procede a hacer el diseño de la tarjeta PCB la cual se puede ver una vista superior en la figura No. 7, una vista isométrica 3D en la figura No. 8.

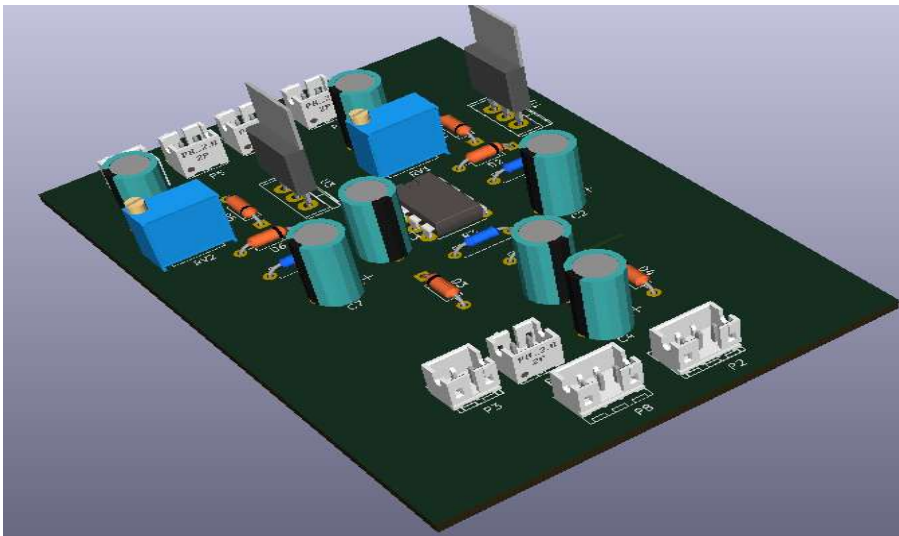
La capa TOP de la PCB que se observa en la figura No. 9 utilizada para facilitar la incorporación de los elementos en la tarjeta y la capa Bottom de la PCB que se ve en la figura No. 10 y se utiliza para obtener el patron los caminos de cobre.

Figura No. 7 Vista superior de modelo 3D en KICAD de tarjeta de fuentes reguladas.



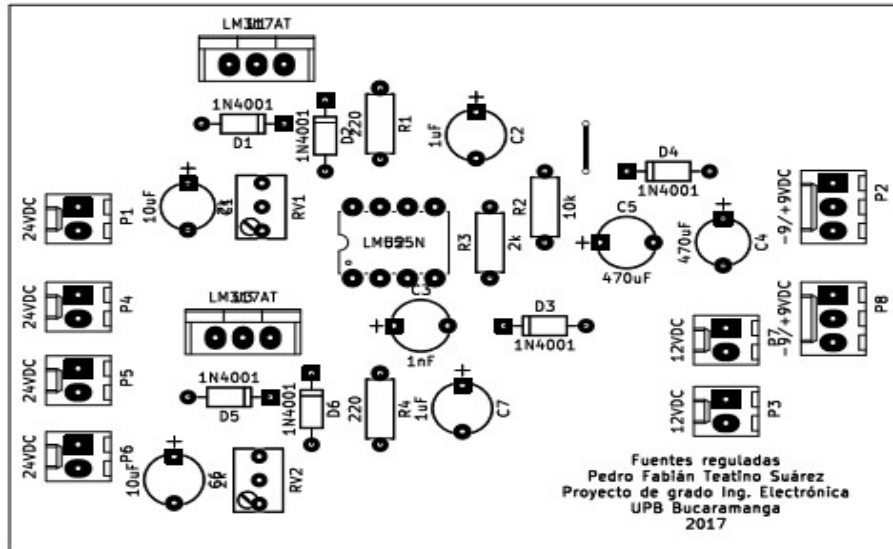
Fuente: Propia

Figura No. 8 Vista isométrica de modelo 3D en KICAD de tarjeta de fuentes reguladas.



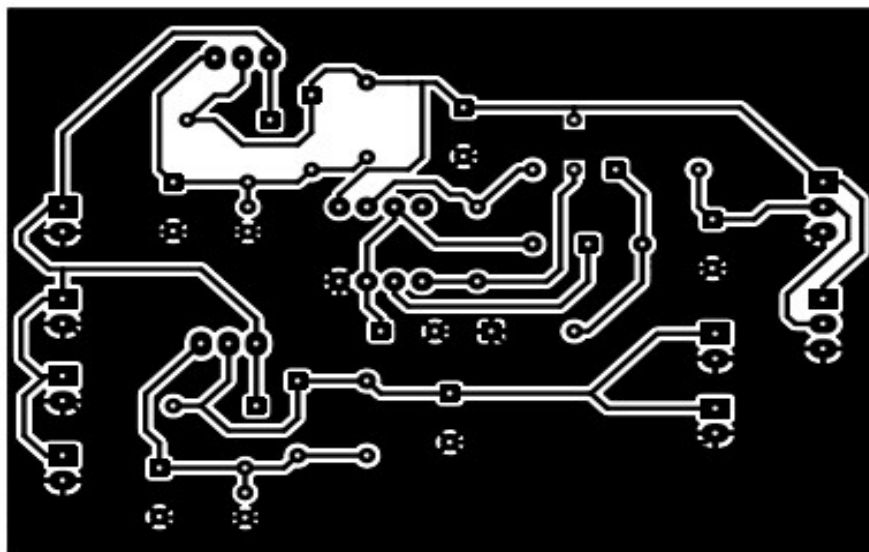
Fuente: Propia

Figura No. 9 Top o mascarar de PCB tarjeta de fuentes reguladas desarrollada en KICAD.



Fuente: Propia

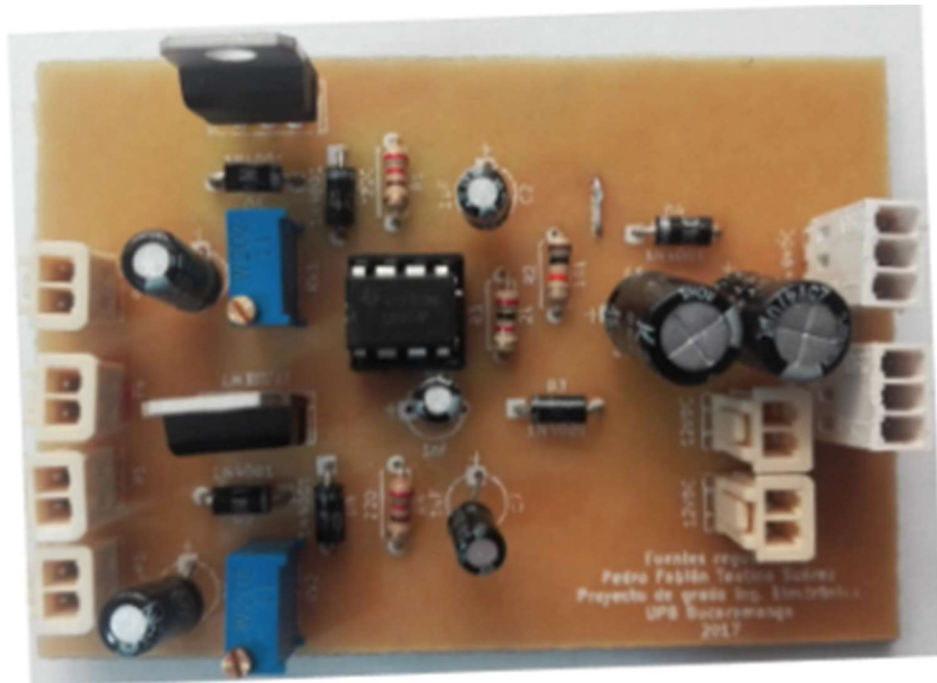
Figura No. 10 Bottom de PCB tarjeta de fuentes reguladas desarrollada en KICAD.



Fuente: Propia

Luego del diseño de la tarjeta para las fuentes reguladas, se procede a la realización de la PCB en baquelita tratada con ácido y posteriormente se incorporan los elementos electrónicos para obtener la tarjeta terminada que se aprecia en la figura No. 11.

Figura No. 11 Vista superior de tarjeta fuente terminada.



Fuente: Propia

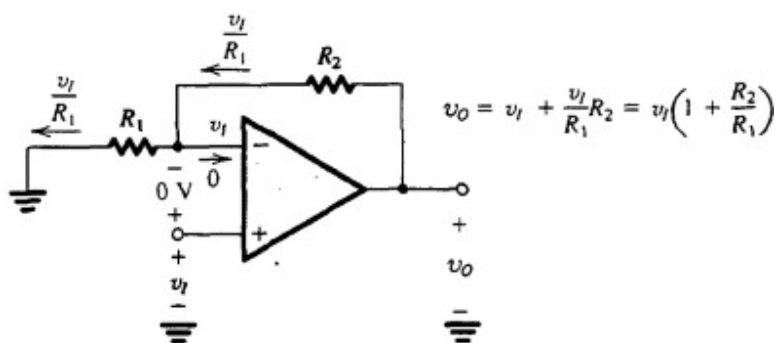
7.2 TARJETA DE SEÑALES

Es la encargada de acoplar eléctricamente las señales analógicas proveídas por el Arduino para que puedan ser reconocidas por las diferentes familias de variadores de velocidad para los cuales fue diseñado el dispositivo, este acoplamiento se hace con ayuda de un amplificador operacional y su respectivo arreglo de resistencias basado en la figura No. 12, buscando que se encuentren en los rangos de voltaje y corriente estándares, de 0(2) a 10 V y 0(4) a 20 mA.

Para el acople de la señal que provee el variador y que debe ser reconocida por el Arduino se hace un divisor de tensión de una señal de corriente 0 a 20 mA en una señal 0 a 3,2 V con resistencias.

La ecuación No 3 es la que nos deja calcular los elementos que se usan en un amplificador no inversor para obtener la señal requerida a la salida, que para este caso es de 10 Voltios.

Figura No. 12 Análisis del circuito no inversor.



Fuente: Circuitos microelectrónicos 4ta edic. (Adel S Sedra, Kenneth C Smith) pag. 100 editorial Oxford University Press, Inc. ISBN 0-19-511663-1.

$$V_o = V_i + \left(1 + \frac{R_2}{R_1}\right) \quad (3)$$

$V_i = 2,75 V$ Valor máximo de salida del Arduino por DAC

$V_o = 10 V$ Valor máximo en la entrada analógica del variador

$$\frac{V_o}{V_i} = A = \left(1 + \frac{R_2}{R_1}\right) \quad (4)$$

$$\Rightarrow A = \frac{10}{2,75} \approx 3,6$$

Al despejar la ecuación No 3 para hallar la ganancia (A) obtenemos la ecuación No 4. Queriendo mantener la relación de A con valores de resistencia comunes, se propone usar los siguientes valores para R_1 y R_2 .

$$\text{Si } R_2 = 3k\Omega \text{ y } R_1 = 1,2k\Omega \Rightarrow \left(1 + \frac{R_2}{R_1}\right) = 3,5 \Rightarrow A \approx 3,6$$

$$\Rightarrow V_o = 2,75 V \left(1 + \frac{3000\Omega}{1200\Omega}\right) \Rightarrow V_o = 9,62 V$$

Para simular la señal de corriente de 0 a 20 mA se pone a la salida del Arduino una resistencia que al conectar a la entrada analógica del variador el cual tiene una impedancia de entrada de 100Ω forman una impedancia

total de 138Ω aproximadamente, la cual fue calculada mediante la ecuación No. 5.

$$R = \frac{V}{I} \quad (5)$$

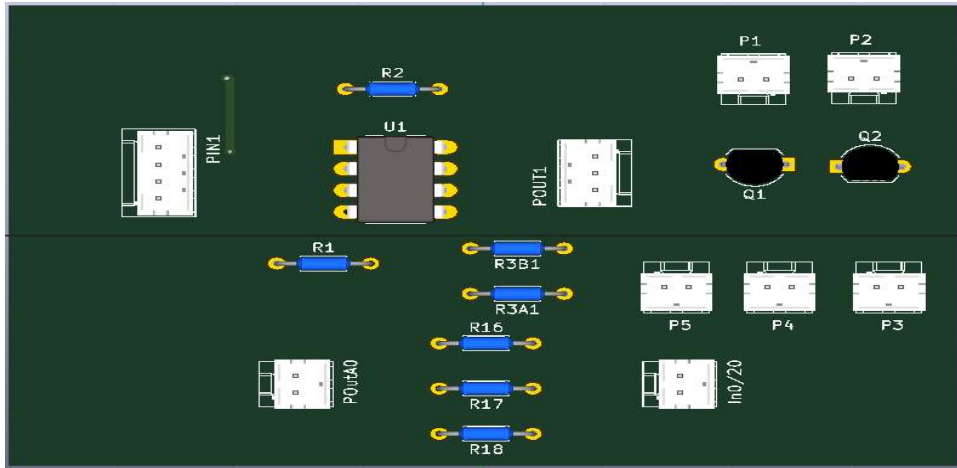
Como la salida de voltaje máxima del Arduino es de $2,75 \text{ V}$, y la señal de corriente que queremos es de 20 mA aproximadamente, entonces verificamos que:

$$R = \frac{2,75 \text{ V}}{0,02 \text{ A}} \Rightarrow R = 137,5 \Omega$$

Esta tarjeta también provee una señal digital de 0 o 24 V , la cual es usada para probar la respuesta de las entradas digitales del variador y también la conducción de las salidas de rele internas del drive, esto se hace generando o quitando una señal desde el arduino a la base de los transistores $2\text{N}2222$ que dejan paso o interrumpen la señal 24 V . El planteamiento de la distribución de esta PCB llamada tarjeta de señales, se ve en la figura No. 13 y una vista isométrica en la figura No. 14.

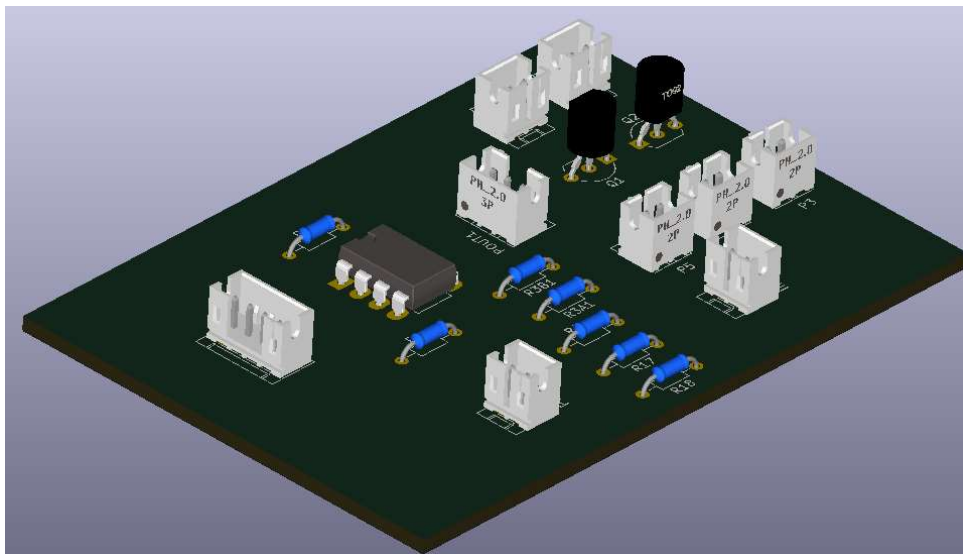
En la figura No. 15 y la figura No. 16 se puede ver la capa TOP y Bottom respectivamente utilizadas para la incorporación de elementos electrónicos y caminos de cobre con los cuales se desarrolla la PCB final que se muestra en la figura No. 17.

Figura No. 13 Vista superior de modelo 3D en KICAD de tarjeta de señales.



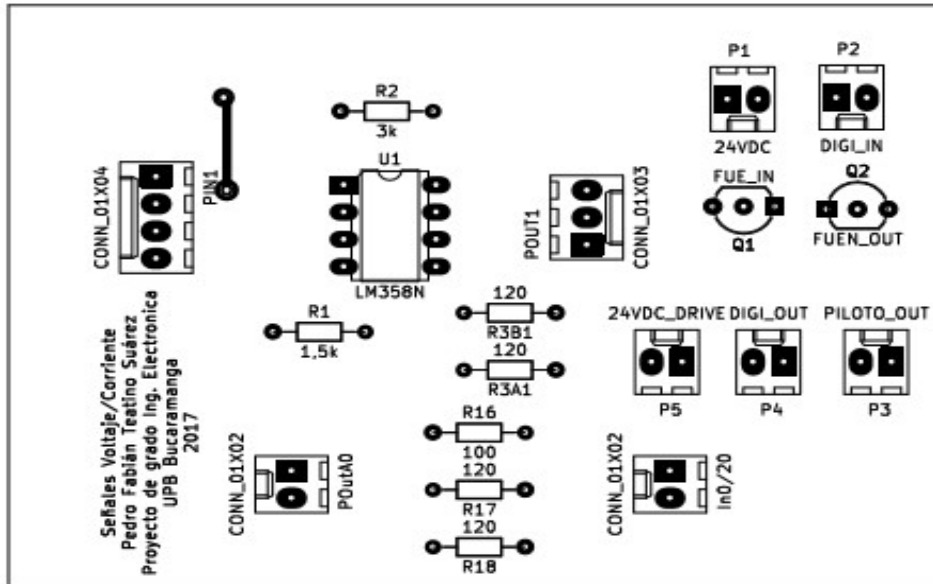
Fuente: Propia

Figura No. 14 Vista isométrica de modelo 3D en KICAD de tarjeta de señales.



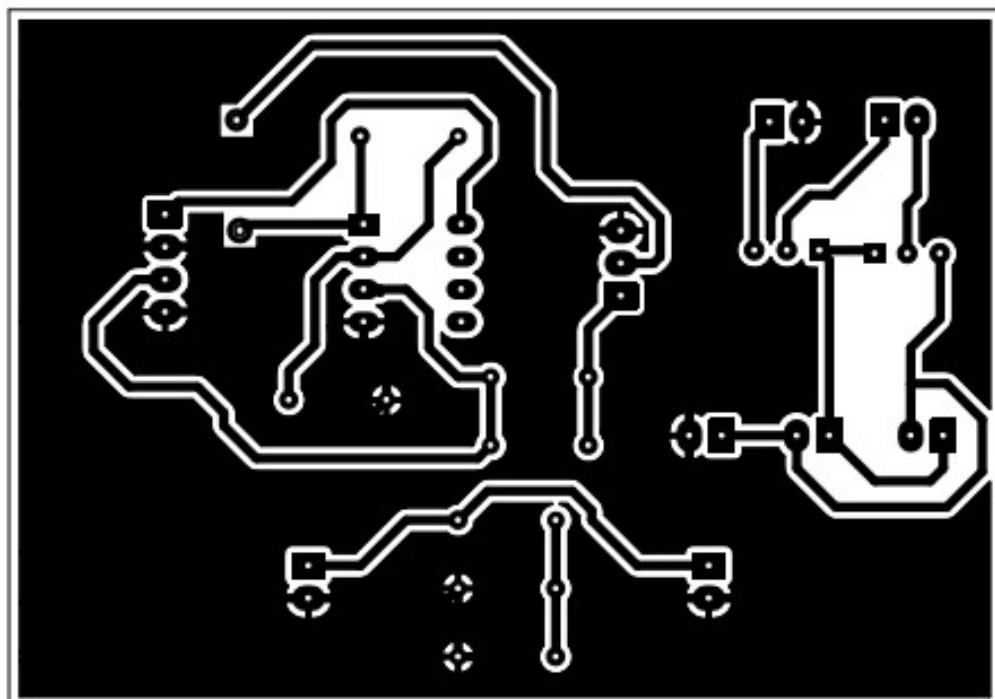
Fuente: Propia

Figura No. 15 Top o mascarás de PCB tarjeta de señales desarrollada en KICAD.



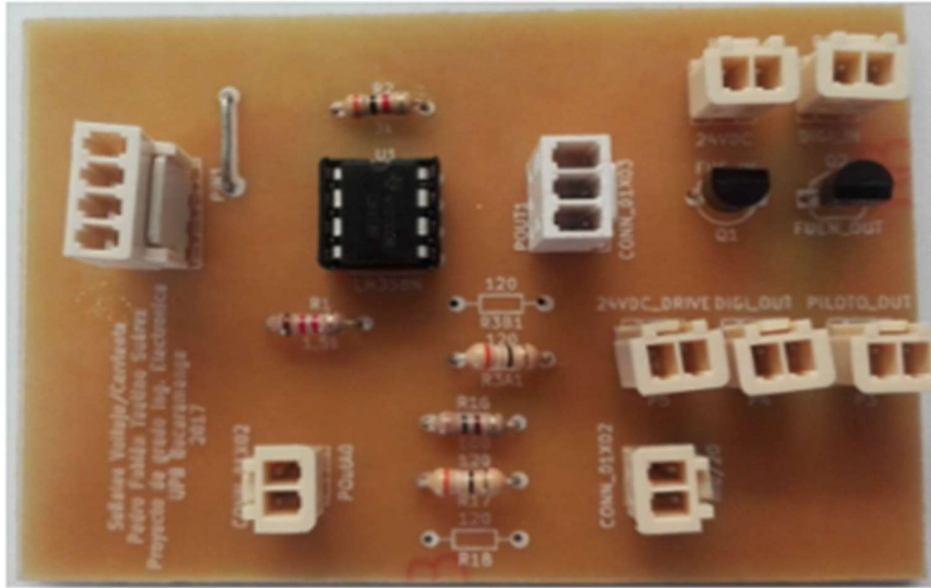
Fuente: Propia

Figura No. 16 Button de PCB tarjeta de señales desarrollada en KICAD.



Fuente: Propia

Figura No. 17 Vista superior de tarjeta de señales terminada.



Fuente: Propia

7.3 TARJETA DE SIMULACIÓN DE ENCODER

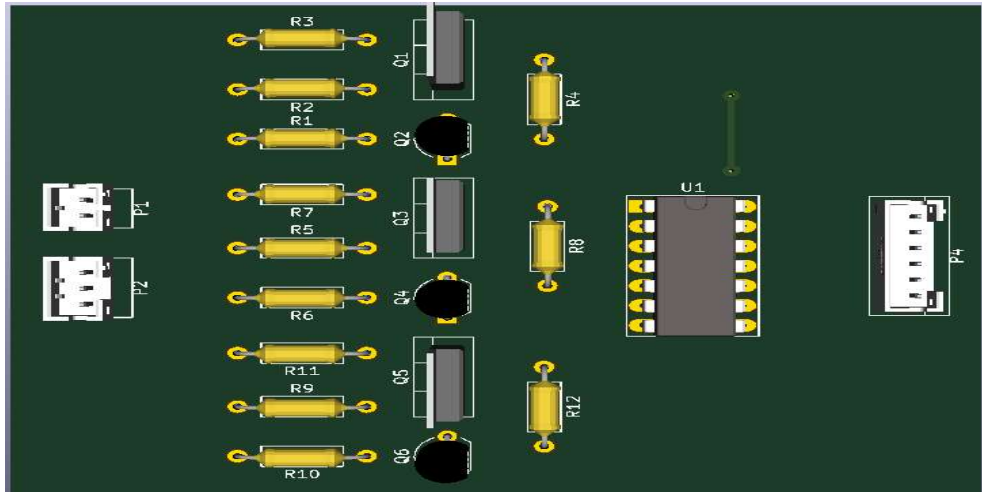
Las señales que simulan el encoder son señales 24 V que se generan con un desfase de 90° por medio de un arreglo de transistores los cuales activa y desactiva el Arduino.

Esta señal de Arduino activa un transistor 2N2222 que a su vez activa un TIP41C que en configuración PULL-DOWN deja circular las señales de 24 V de corriente continua necesarias en cada canal para ser interpretado por el variador mediante su respectivo modulo o accesorio para lectura de encoder (OTAC-01, RTAC-01, entre otros). Adicional mente en paralelo se encuentran los otros 3 canales negados respectivamente por medio de una compuerta lógica 7404. Las 6 salidas son A, A negado, B, B negado, Z y Z negado.

La generación de esta señal se encuentra limitada por la frecuencia de conmutación que tienen los dispositivos que allí intervienen. La frecuencia de conmutación del 2N2222 es de 250 MHz según el datasheet y la frecuencia de conmutación del TIP41C es de 3 MHz.

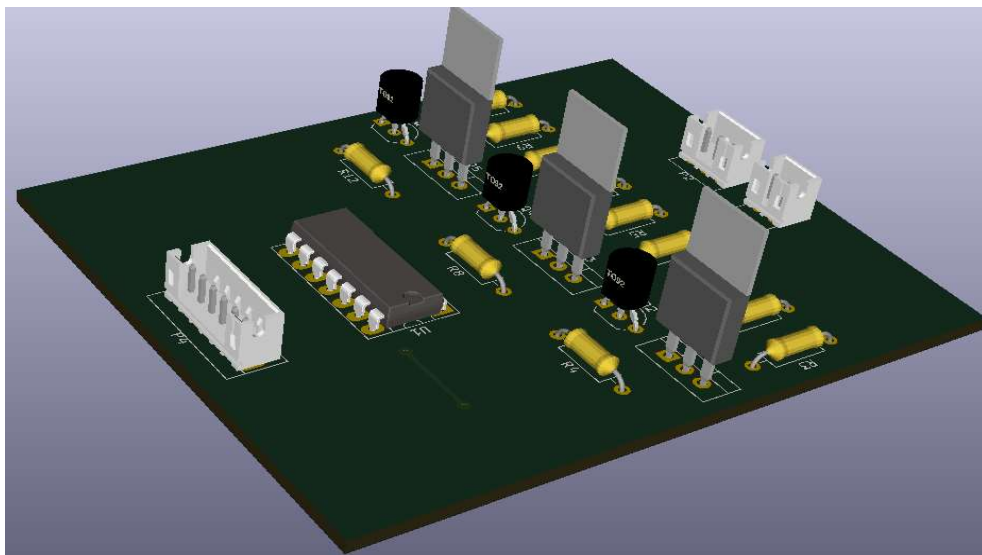
En la figura No. 18 se ve la simulación de la disposición de los elementos electrónicos de la tarjeta, teniendo también una vista isométrica en la figura No. 19.

Figura No. 18 Vista superior de modelo 3D en KICAD de tarjeta de simulación de encoder.



Fuente: Propia

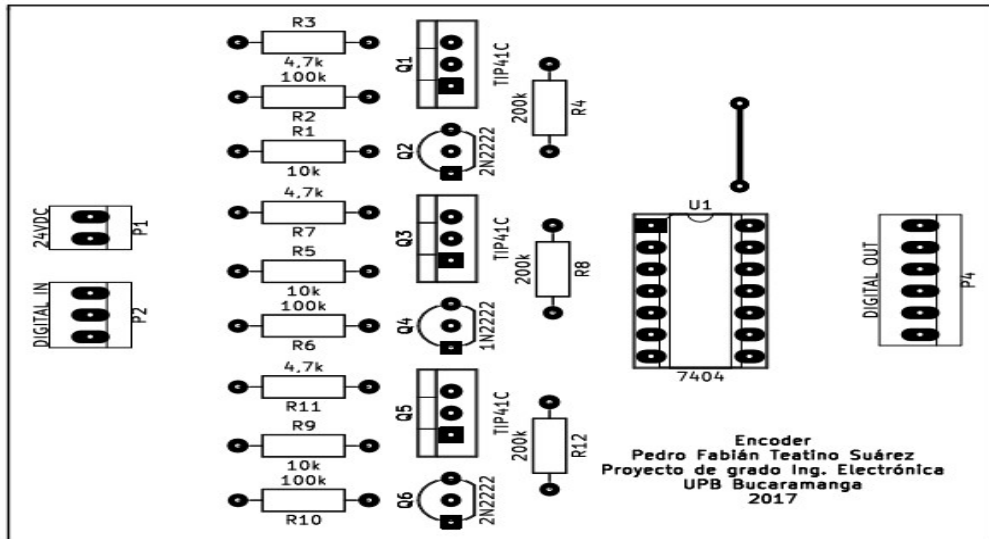
Figura No. 19 Vista isométrica de modelo 3D en KICAD de tarjeta de simulación de encoder.



Fuente: Propia

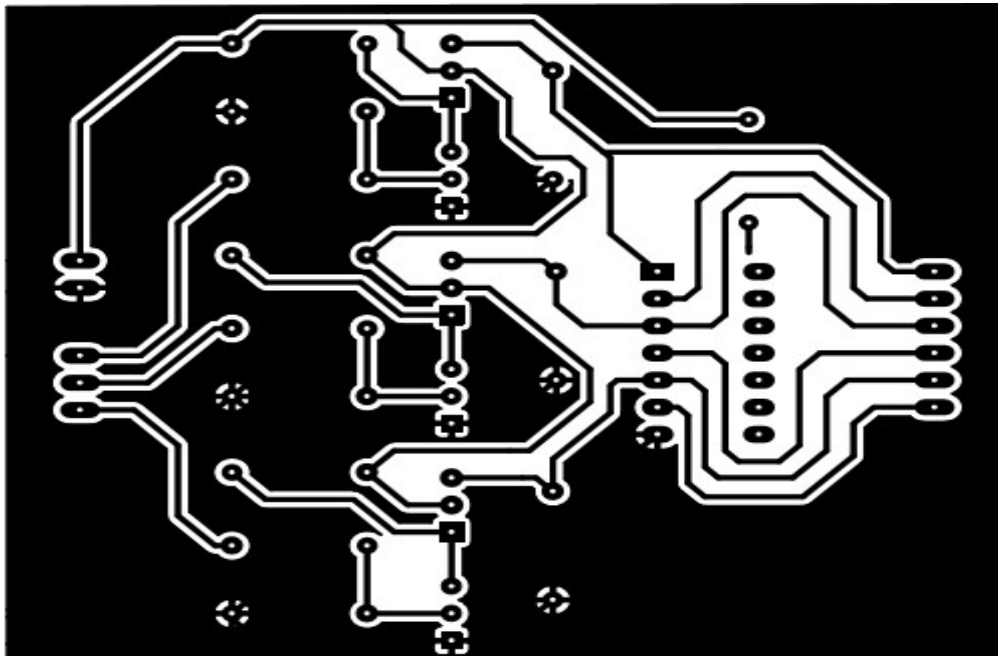
Las figuras que se utiliza para la impresión de la PCB son la figura No. 20 que muestra la disposición de los elementos electrónicos, y la figura No. 21 que muestra los caminos de cobre. En la figura No. 22 se ve la PCB terminada.

Figura No. 20 Top o mascarás de PCB tarjeta de señales desarrollada en KICAD.



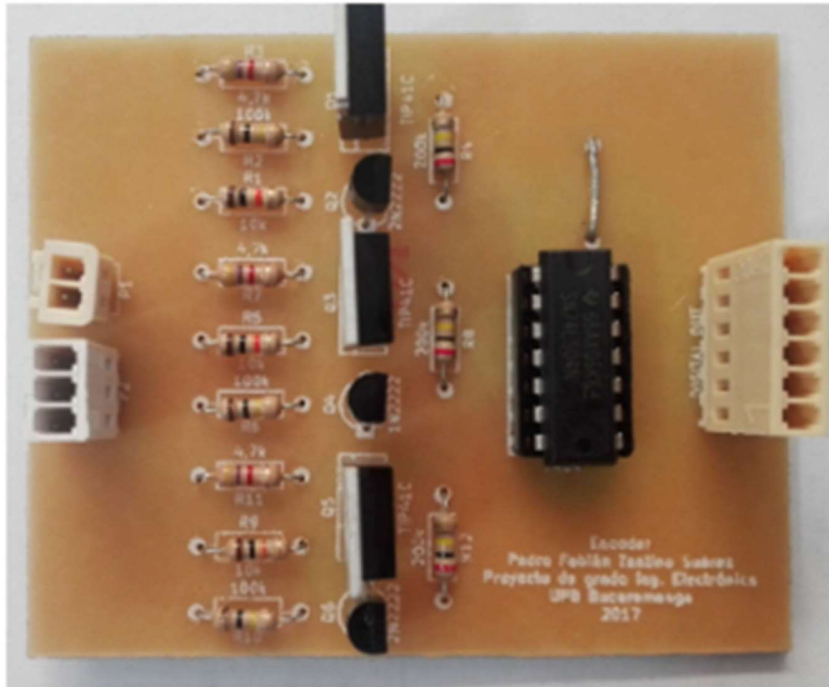
Fuente: Propia

Figura No. 21 Button de PCB tarjeta de simulación de encoder desarrollada en KICAD.



Fuente: Propia

Figura No. 22 Vista superior de tarjeta Simulación de Encoder Terminada.



Fuente: Propia

7.4 TARJETA DE PRUEBA DE SEMICONDUCTORES

Esta **funciona** bajo el principio de paso y corte de un voltaje 24 V de corriente continua que simula la potencia que circula por el semiconductor (IGBT o Tiristor) y una señal en la puerta que simula el control de la conmutación del mismo.

Para el diseño de esta etapa, se tienen en cuenta los principios básicos de funcionamiento de los tiristores SCR los cuales son:

Aplicar una corriente de puerta (I_{GT}) mínima que es de 100 mA en los dispositivos a diagnosticar, acompañado de un voltaje de puerta (V_{GT}) mínima de 2,5 V, para generarla conducción de ánodo a cátodo cuando la tensión entre ellos sea positiva. Una vez allá entrado en conducción el dispositivo, esta corriente de puerta generada (I_{GT}) deja de ser necesaria para mantener la conducción del dispositivo y empieza a ser esencial mantener un valor de corriente ánodo a cátodo por encima del valor mínimo, la cual es llamada corriente de mantenimiento (I_H) y depende de las características de construcción de cada dispositivo, en los dispositivos a diagnosticar varía entre 150 y 200 mA, por lo cual se dimensiona una corriente estándar de 352 mA aproximadamente para garantizar la conducción siempre, por ello esta tarjeta tiene una resistencia de 68 Ω que

se pone en serie con el tiristor y fue calculada utilizando la ecuación No 5, sabiendo que el voltaje que allí se aplica es de 24 V.

El apagado del SCR se hace por medio del corte de tensión positiva entre el ánodo y cátodo por medio de la apertura del relé que permite su paso.

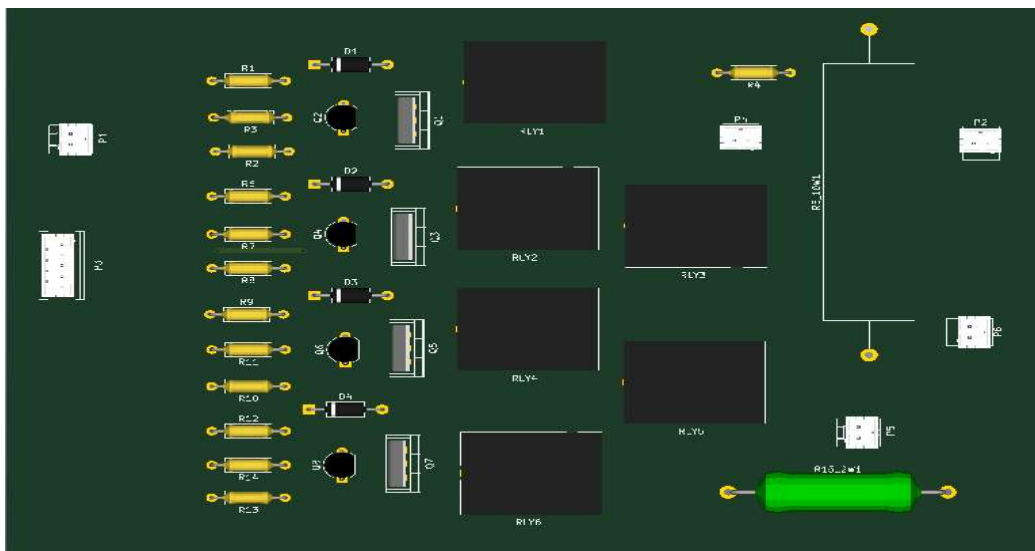
Los principios de funcionamiento de los transistores IGBT también son importantes, ya que para su funcionamiento varían en cuanto al de los tiristores SCR.

En los IGBT podemos tener un control de activación y desactivación por medio de la señal en puerta, mientras que en los SCR solo podíamos controlar la activación por puerta pero la desactivación si es por línea o cruce por cero.

Los IGBT son controlador por señal de tensión en puerta (V_{GE}) que al ser positiva y superior a su valor mínimo, garantiza la conducción siempre y cuando el voltaje colector-emisor sea positivo, y al ser negativa e inferior a su valor máximo de conmutación garantiza el apagado del dispositivo, este voltaje de puerta (V_{GE}) de los semiconductores a trabajar acepta como rangos máximos +/- 20 V, pero su valor típico de funcionamiento es entre +/- 6 V y +/- 15 V por lo cual se decide aplicarle una tensión de +/- 9 V en puerta la cual se encuentra entre los valores aceptados, esta señal positiva o negativa se provee mediante el arreglo de transistores y reles.

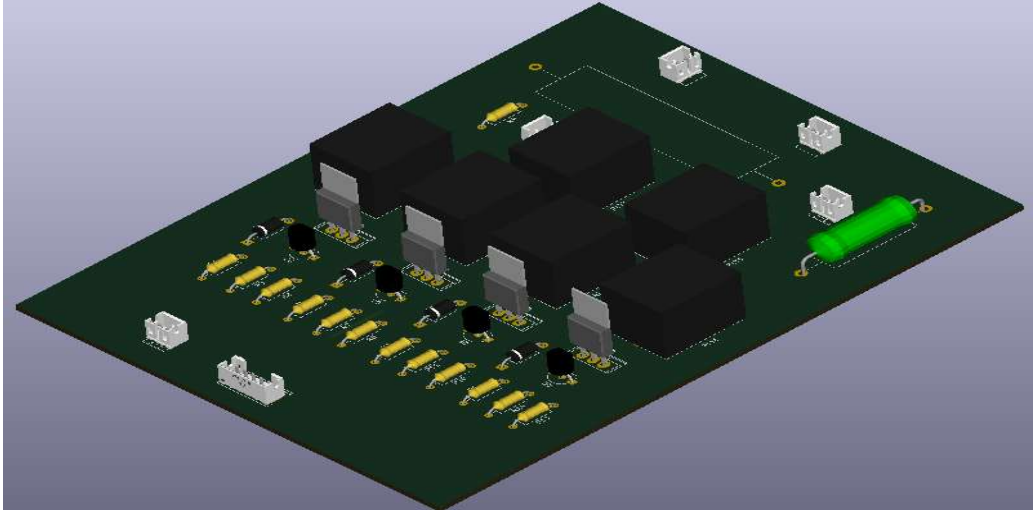
En la figura No. 23 se muestra la disposición de los elementos electrónicos, y en la figura No. 24 se ve una vista isométrica 3D de la misma.

Figura No. 23 Vista superior de modelo 3D en KICAD de tarjeta de pruebas de semiconductores.



Fuente: Propia

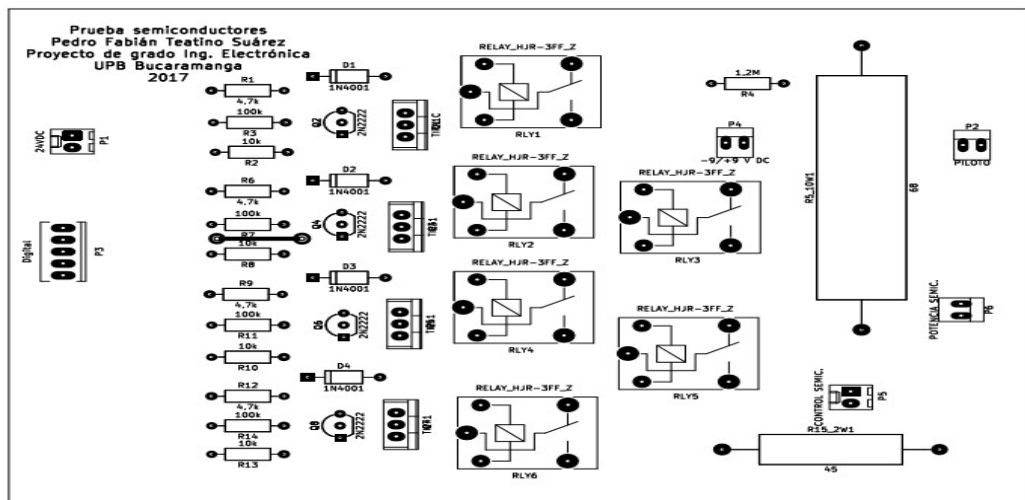
Figura No. 24 Vista isométrica de modelo 3D en KICAD de tarjeta de pruebas de semiconductores.



Fuente: Propia

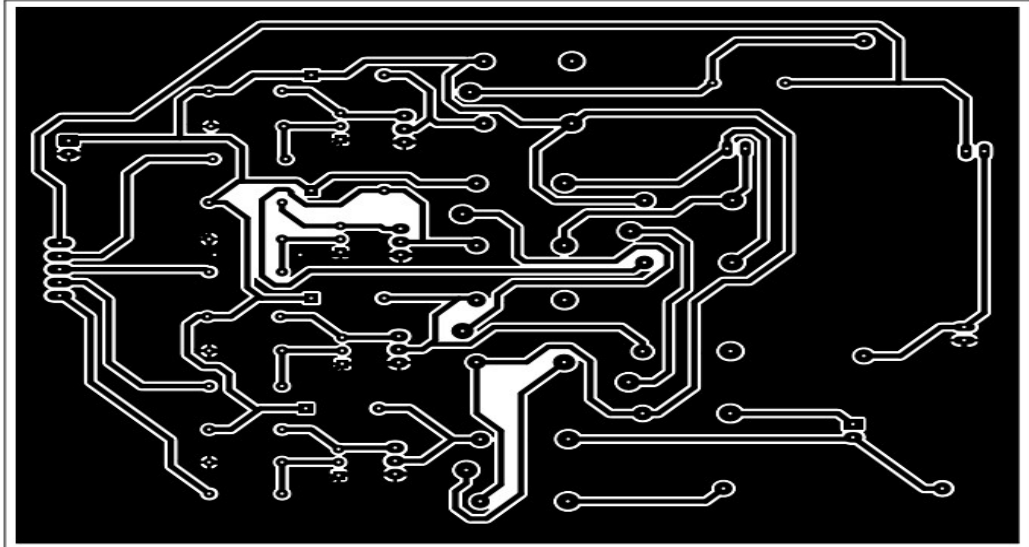
En la figura No. 25 y la figura No. 26 se ven las capas TOP y Bottom de la PCB tarjeta de prueba semiconductores que se utiliza para generar la placa final que se muestra en la figura No. 28.

Figura No. 25 Top o mascarás de PCB tarjeta de prueba de semiconductores desarrollada en KICAD.



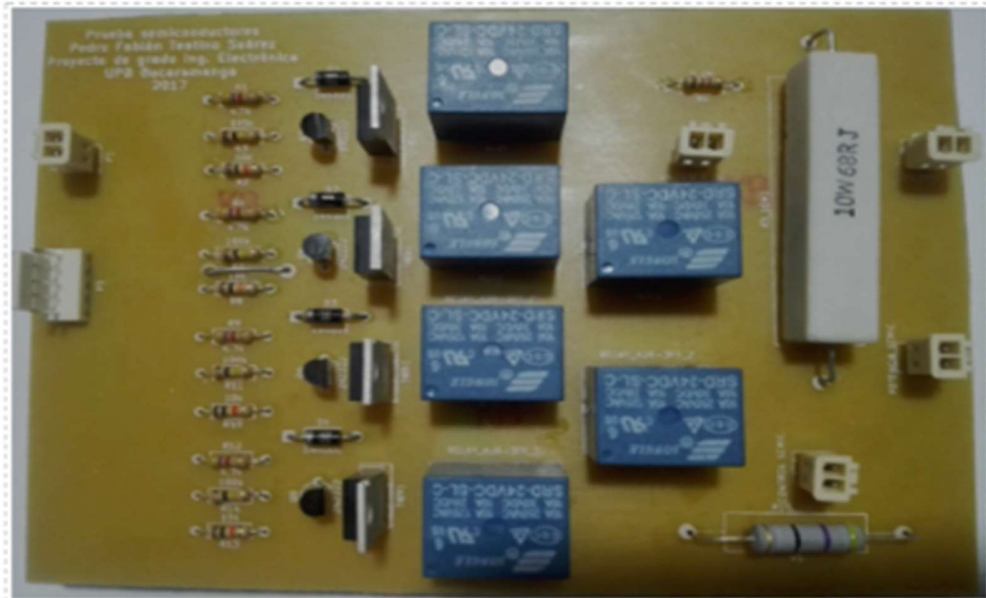
Fuente: Propia

Figura No. 26 Button de PCB tarjeta de prueba de semiconductores desarrollada en KICAD.



Fuente Propia

Figura No. 27 Vista superior de tarjeta de pruebas de semiconductores.



Fuente: Propia

Una vez teniendo las PCB terminas, se hace las conexiones pertinentes al arduino y se realiza la disposición dentro de un envoltente IP65 que provee de gran protección al equipo como se muestra en la figura No. 28.

Figura No. 28 Envoltente del equipo de pruebas.



Fuente: Propia

Una vez conectados todos sus elementos, se hace la disposición interna en la envoltente recubriendo con un acrílico de 4mm de espesor en donde van dispuestos todos los terminales de conexión y luces piloto, en su parte central la pantalla con la cual el usuario interactua como se ve en la figura No. 29.

Figura No. 29 Equipo de pruebas terminado.



Fuente: Propia

8 DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

El código de programación del proyecto se desarrolló en la plataforma IDE de Arduino.

En esta programación se utilizan librerías preestablecidas para lograr la óptima comunicación entre el Arduino DUE y la pantalla touch que hace las veces de HMI (human machine interface)

El programa consta de varias subrutinas desarrolladas para obtener la mejor respuesta del módulo según el requerimiento del técnico al momento de utilizarlo, las cuales se describen a continuación.

Al iniciar el equipo, lo primero que vera el usuario en pantalla es un saludo de bienvenida donde se ve el logo y nombre de la empresa Col Controles, junto con una leyenda de departamento de servicios como se ve en la figura No 30, su programación se puede ver en el anexo “código bienvenida”. El recuadro de bienvenido es un botón que al tocarlo nos lleva a la pantalla de menú principal.

Figura No. 30 Pantalla de bienvenida.



Fuente: Propia

- Menú principal: muestra las diferentes opciones de funcionamiento del dispositivo dentro de las cuales el operador puede seleccionar escribir señal, Leer señal, señal digital, encoder y prueba de semiconductores, tal como se ve en la figura No 41 y se genera mediante el código del anexo “menú principal”. Cada uno de estos recuadros al tocarlo nos llevara a otra pantalla.

Figura No. 31 Pantalla de menú principal.



Fuente: Propia

- Encoder: dentro de esta subrutina el arduino genera pulsos por 2 de sus puertos digitales, en ellos la señal se desfasa 90 grados adelante o atrás según la dirección de motor que desees simular. allí se debe seleccionar la cantidad de pulsos por revolución a generar 512 ppr, 1024 ppr (ver figura No 32).

Figura No. 32 Pantalla encoder principal



Fuente: Propia

Luego de realizar la selección de la cantidad de pulsos por revolución, se selecciona la dirección de motor que desees simular (horario o antihorario) como se muestra en la figura No. 33.

Figura No. 33 Sentido de giro.



Fuente: Propia

Al seleccionar la dirección, se puede escoger entre 3 velocidades diferentes que dependen de los pulsos por revolución tomados anteriormente, las velocidades son 600 rpm, 1800 rpm, 3000 rpm para encoder de 512 ppr (ver figura No. 34) o 300 rpm, 900 rpm, 1500 rpm para encoder de 1024 ppr (ver figura No. 35).

Figura No. 34 Velocidades 512 ppr



Fuente: Propia

Una vez se haga la selección de velocidad requerida por el usuario, se visualiza la pantalla que permite activar la generación de señales de simulación.

Al escoger las opciones: 512ppr > horario > 600 rpm, se visualiza la pantalla que se muestra en la figura No. 36 la cual generara las señales de 24 V con atraso de 90° del canal B con respecto al A a una frecuencia de 5.120 HZ aproximadamente (ver figura No. 37). La programación se presenta en el anexo "Codigo 512 ppr 600 rpm en sentido horario" y anexo "Codigo ON 512 ppr 600 rpm en sentido horario".

Figura No. 35 Velocidades 1024 ppr



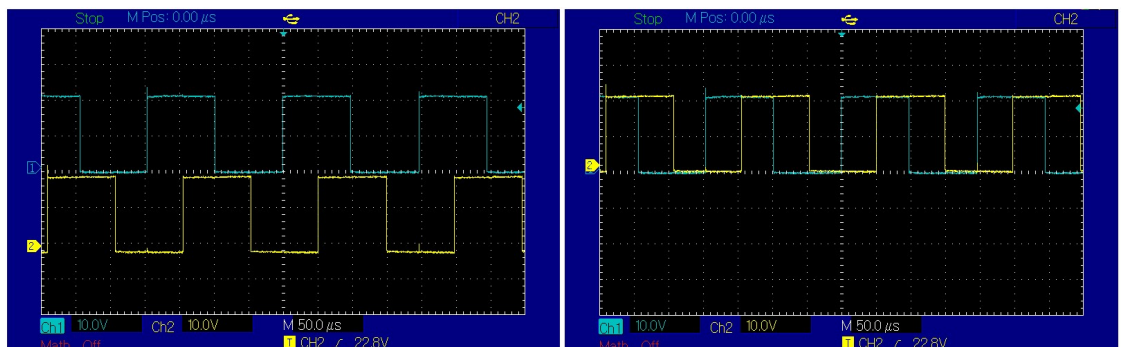
Fuente: Propia

Figura No. 36 Encoder 512ppr horario 600rpm



Fuente: Propia

Figura No. 37 Generacion señal encoder 51 ppr horario 600rpm



Fuente: Propia

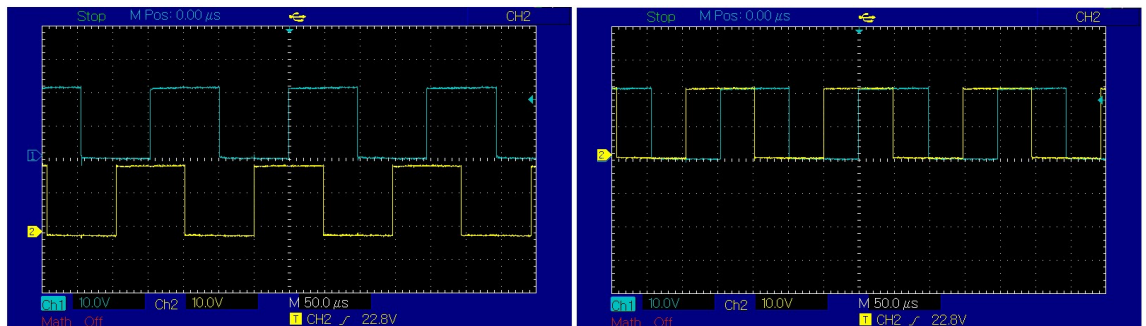
Al escoger las opciones: 512ppr > antihorario > 600 rpm, se visualiza la pantalla que se muestra en la figura No. 38 la cual generara las señales de 24 V con adelanto de 90° del canal B con respecto al A a una frecuencia de 5.120 HZ aproximadamente (ver figura No. 39). La programación se presenta en el anexo “Codigo 512 ppr 600 rpm en sentido antihorario” y anexo “Codigo ON 512 ppr 600 rpm en sentido antihorario”.

Figura No. 38 Encoder 512ppr antihorario 600rpm



Fuente: Propia

Figura No. 39 Generacion señal encoder 512ppr antihorario 600rpm



Fuente: Propia

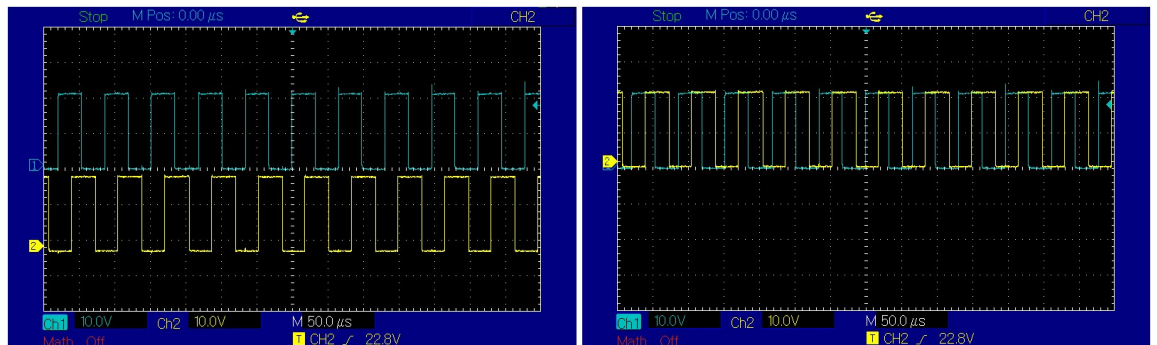
Al escoger las opciones: 512ppr > horario > 1800 rpm, se visualiza la pantalla que se muestra en la figura No. 40 la cual generara las señales de 24 V con retraso de 90° del canal B con respecto al A a una frecuencia de 15.360 HZ aproximadamente (ver figura No. 41). La programación se presenta en el anexo “Codigo 512 ppr 1800 rpm en sentido horario” y anexo “Codigo ON 512 ppr 1800 rpm en sentido horario”.

Figura No. 40 Encoder 512ppr horario 1800rpm



Fuente: Propia

Figura No. 41 Generacion señal encoder 512ppr horario 1800rpm.



Fuente: Propia

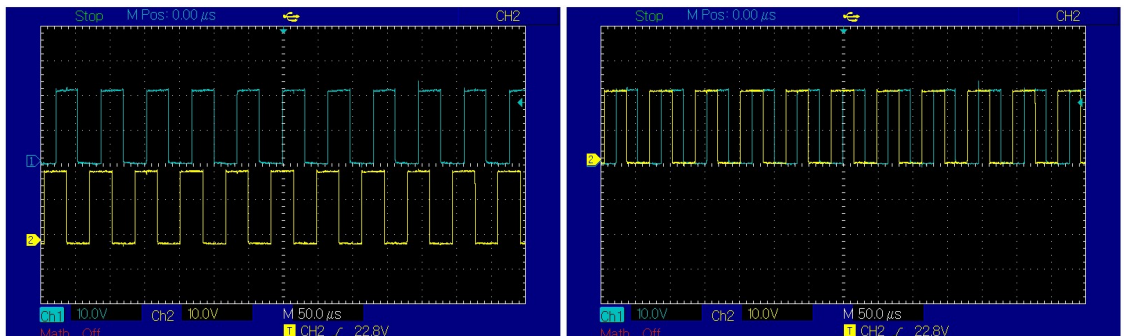
Al escoger las opciones: 512ppr > antihorario > 1800 rpm, se visualiza la pantalla que se muestra en la figura No. 42 la cual generara las señales de 24 V con adelanto de 90° del canal B con respecto al A a una frecuencia de 15.360 HZ aproximadamente (ver figura No. 43). La programación se presenta en el anexo “Codigo 512 ppr 1800 rpm en sentido antihorario” y anexo “Codigo ON 512 ppr 1800 rpm en sentido antihorario”.

Figura No. 42 Encoder 512ppr antihorario 1800rpm.



Fuente: Propia

Figura No. 43 Generacion señal encoder 512ppr antihorario 1800rpm



Fuente: Propia

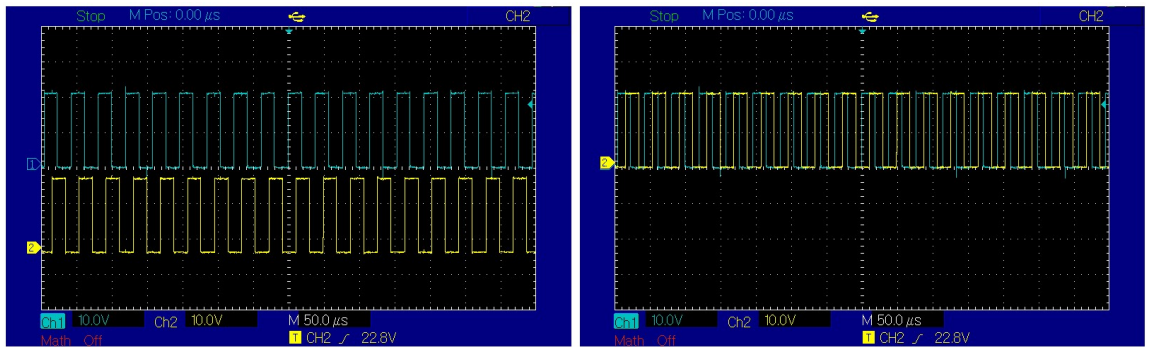
Al escoger las opciones: 512ppr > horario > 3000 rpm, se visualiza la pantalla que se muestra en la figura No. 44 la cual generara las señales de 24 V con retraso de 90° del canal B con respecto al A a una frecuencia de 25.600 HZ aproximadamente (ver figura No. 45). La programación se presenta en el anexo “Codigo 512 ppr 3000 rpm en sentido horario” y anexo “Codigo ON 512 ppr 3000 rpm en sentido horario”.

Figura No. 44 Encoder 512ppr horario 3000rpm



Fuente: Propia

Figura No. 45 Generacion señal encoder 512ppr horario 3000rpm



Fuente: Propia

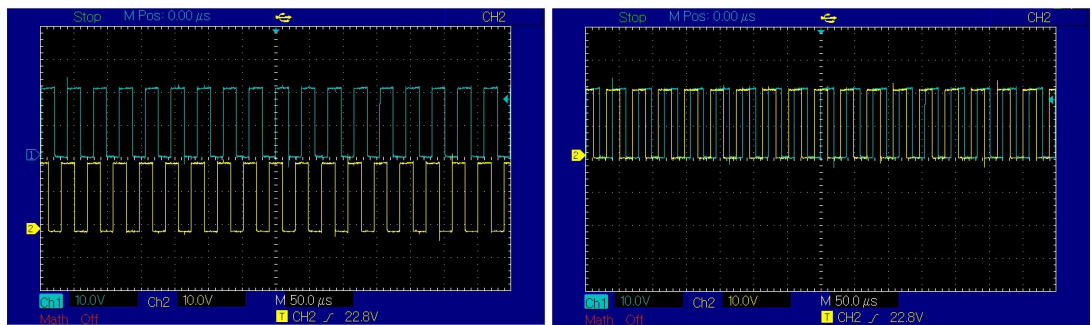
Al escoger las opciones: 512ppr > antihorario > 3000 rpm, se visualiza la pantalla que se muestra en la figura No. 46 la cual generara las señales de 24 V con adelanto de 90° del canal B con respecto al A a una frecuencia de 25.600 HZ aproximadamente (ver figura No. 47). La programación se presenta en el anexo “Codigo 512 ppr 3000 rpm en sentido antihorario” y anexo “Codigo ON 512 ppr 3000 rpm en sentido antihorario”.

Figura No. 46 Encoder 512ppr antihorario 3000rpm



Fuente: Propia

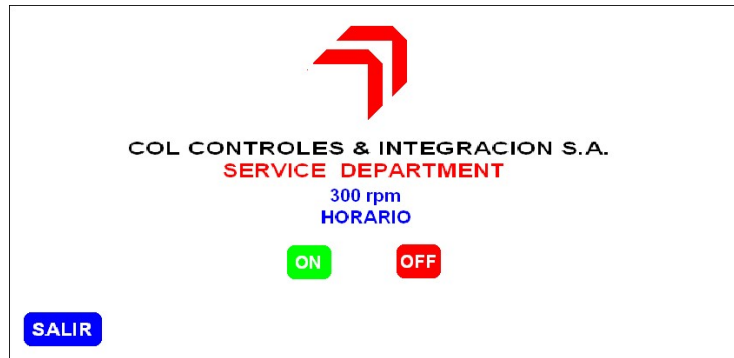
Figura No. 47 Generacion señal encoder 512ppr antihorario 3000rpm



Fuente: Propia

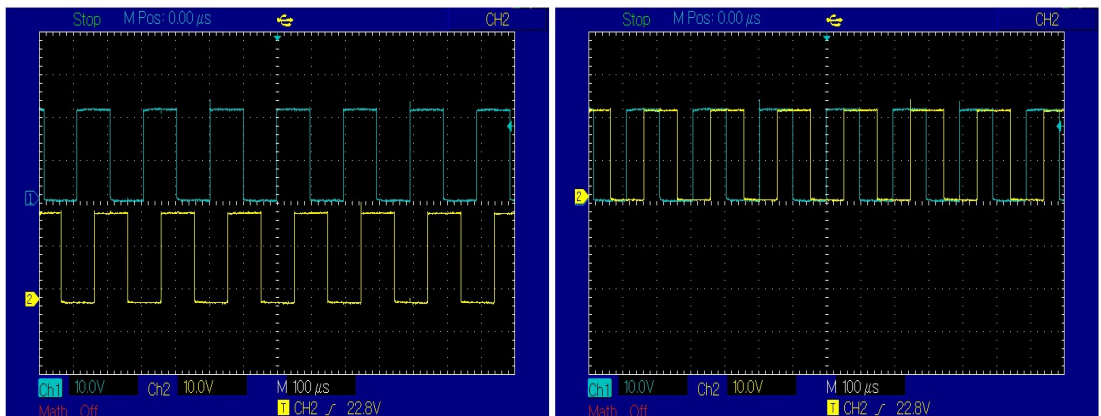
Al seleccionar las opciones: 1024ppr > horario > 300 rpm, se visualiza la pantalla que se muestra en la figura No. 48 la cual generara las señales de 24 V con retraso de 90° del canal B con respecto al A a una frecuencia de 5.120 HZ aproximadamente (ver figura No. 49). La programación se presenta en el anexo “Codigo 1024 ppr 300 rpm en sentido horario” y anexo “Codigo ON 512 ppr 300 rpm en sentido horario”.

Figura No. 48 Encoder 1024ppr horario 300rpm.



Fuente: Propia

Figura No. 49 Generacion señal encoder 1024ppr horario 300rpm.



Fuente: Propia

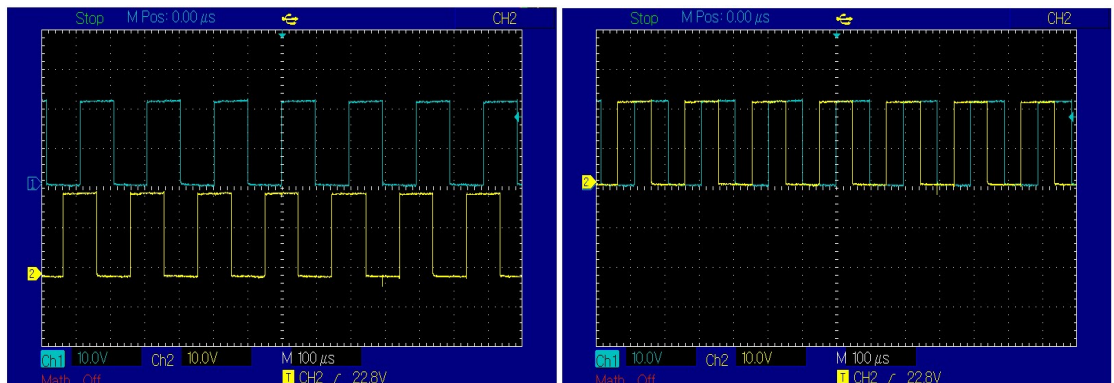
Al seleccionar las opciones: 1024ppr > antihorario > 300 rpm, se visualiza la pantalla que se muestra en la figura No. 50 la cual genera las señales de 24 V con adelanto de 90° del canal B con respecto al A a una frecuencia de 5.120 HZ aproximadamente (ver figura No. 51). La programación se presenta en el anexo "Codigo 1024 ppr 300 rpm en sentido antihorario" y anexo "Codigo ON 512 ppr 300 rpm en sentido antihorario",

Figura No. 50 Encoder 1024ppr antihorario 300rpm



Fuente: Propia

Figura No. 51 Generacion señal encoder 1024ppr antihorario 300rpm



Fuente: Propia

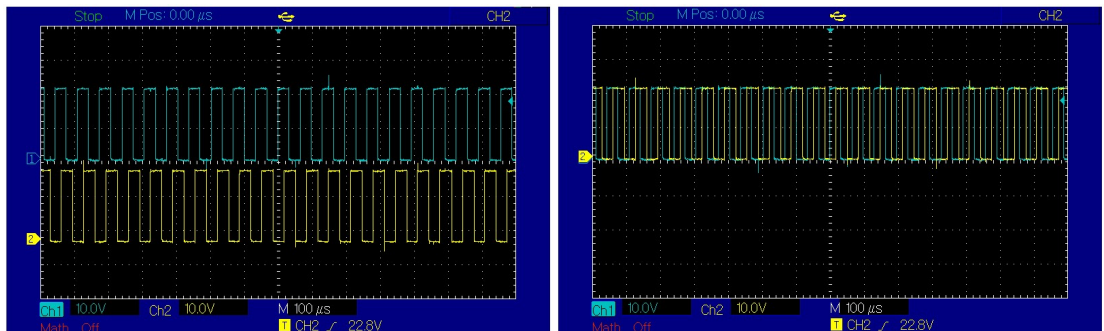
Al seleccionar las opciones: 1024ppr > horario > 900 rpm, se visualiza la pantalla que se muestra en la figura No. 52 la cual generara las señales de 24 V con retraso de 90° del canal B con respecto al A a una frecuencia de 15.360 HZ aproximadamente (ver figura No.53). La programación se presenta en el anexo “Codigo 1024 ppr 900 rpm en sentido horario” y anexo “Codigo ON 512 ppr 900 rpm en sentido horario”.

Figura No. 52 Encoder 1024ppr horario 900rpm



Fuente: Propia

Figura No. 53 Generacion señal encoder 1024ppr horario 900rpm



Fuente: Propia

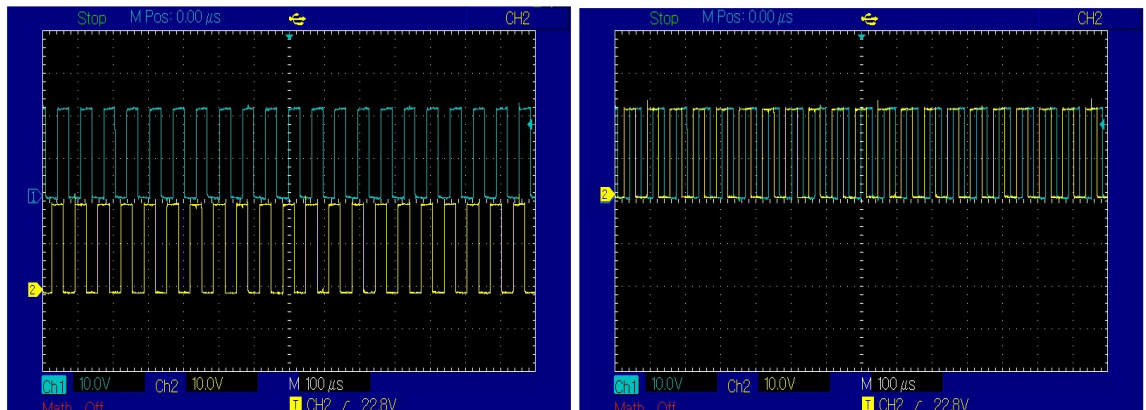
Al seleccionar las opciones: 1024ppr > antihorario > 900 rpm, se visualiza la pantalla que se muestra en la figura No. 54 la cual generara las señales de 24 V con adelanto de 90° del canal B con respecto al A a una frecuencia de 15.360 HZ aproximadamente (ver figura No. 55). La programación se presenta en el anexo “Codigo 1024 ppr 900 rpm en sentido antihorario” y anexo “Codigo ON 512 ppr 900 rpm en sentido antihorario”,

Figura No. 54 Encoder 1024ppr antihorario 900rpm



Fuente: Propia

Figura No. 55 generacion señal encoder 1024ppr antihorario 900rpm



Fuente: Propia

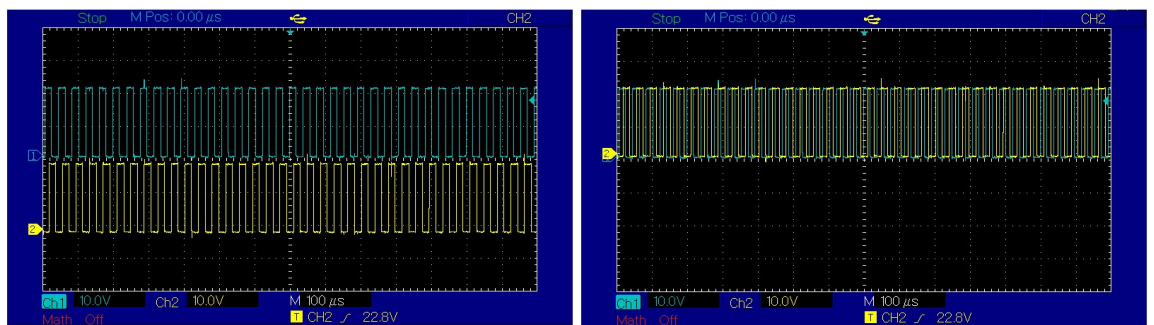
Al seleccionar las opciones: 1024ppr > horario > 1500 rpm, se visualiza la pantalla que se muestra en la figura No. 56 la cual generara las señales de 24 V con adelanto de 90° del canal B con respecto al A a una frecuencia de 25.600 HZ aproximadamente (ver figura No. 57). La programación se presenta en el anexo “Codigo 1024 ppr 1500 rpm en sentido horario” y anexo “Codigo ON 512 ppr 1500 rpm en sentido horario”

Figura No. 56 Encoder 1024ppr horario 1500rpm



Fuente: Propia

Figura No. 57 Generacion señal encoder 1024ppr horario 1500rpm



Fuente: Propia

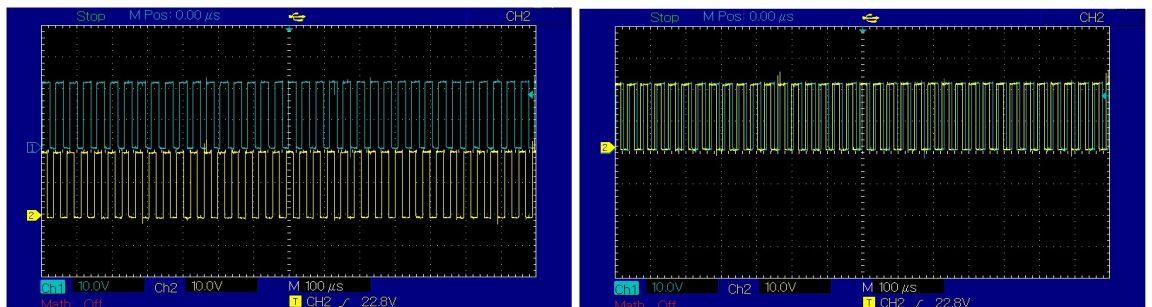
Al seleccionar las opciones: 1024ppr > antihorario > 1500 rpm, se visualiza la pantalla que se muestra en la figura No. 58 la cual generara las señales de 24 V con adelanto de 90° del canal B con respecto al A a una frecuencia de 25.600 HZ aproximadamente (ver figura No. 59). La programación se presenta en el anexo “Codigo 1024 ppr 1500 rpm en sentido antihorario” y anexo “Codigo ON 512 ppr 1500 rpm en sentido antihorario”.

Figura No. 58 Encoder 1024ppr antihorario 1500rpm



Fuente: Propia

Figura No. 59 Generacion señal encoder 1024ppr antihorario 1500rpm



Fuente: Propia

- Escribir señal: Esta subrutina está compuesta de 2 opciones principales, el operador puede escoger entre generar una señal de corriente 0 a 20mA y una señal de voltaje 0 a 10V (ver figura No. 60 y anexos “generacion de señal”).

Figura No. 60 Selección de señal



Fuente: Propia

Una vez seleccionado el tipo de señal, se puede escoger una generación automática o manual (ver figura No. 61). La opción automática que consiste en que el Arduino empieza generando el 0% de la señal y la va llevando gradualmente hasta el 100% para luego devolverla a 0% luego de oprimir la tecla play que se muestra en la figura No. 62 y la figura No. 63, en voltaje o corriente según corresponda (ver figura No. 64), en esta opción el operador puede pausar en cualquier momento la señal, lo cual hace que el equipo de pruebas quede generando una señal constante hasta que se vuelva a activar la tecla play como se puede ver en la figura No. 65.

NOTA: Las subrutinas de generación de señales son iguales para la señal de voltaje y de corriente, lo único que se tuvo en cuenta fue el acoplamiento de señal por medio del amplificador operacional y sus resistencias para el voltaje, y un arreglo de impedancia para el uso de señal de corriente. Aunque en el arduino se genera en el mismo puerto, en el equipo de pruebas tienen puertos de salida distintos.

Figura No. 61 Generacion de señal



Fuente: Propia

Figura No. 62 Play-pause de señal automática V



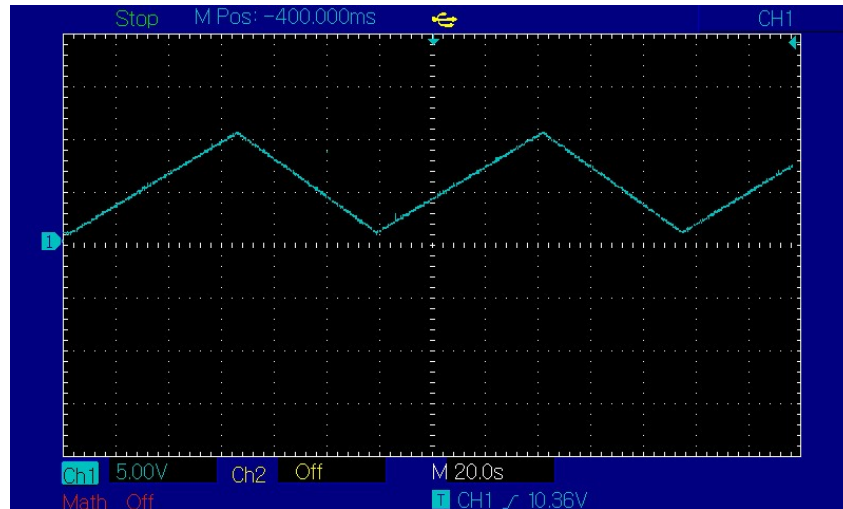
Fuente: Propia

Figura No. 63 Play-pause de señal automática mA



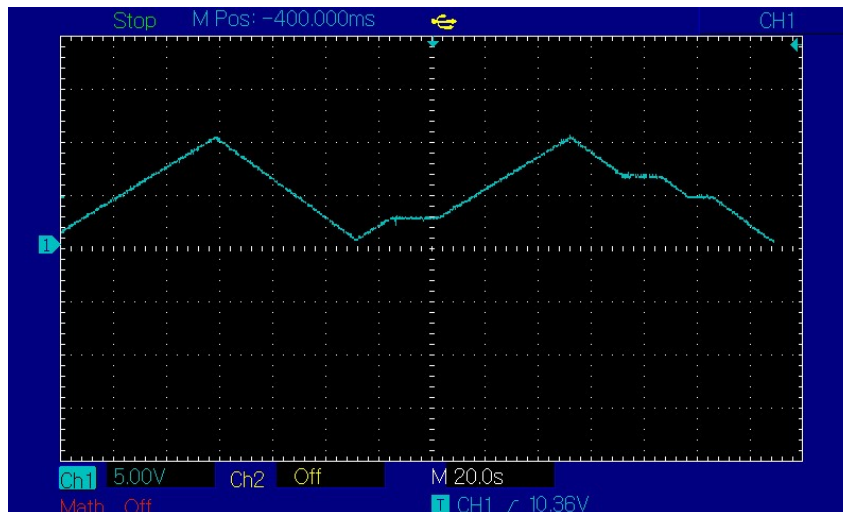
Fuente: Propia

Figura No. 64 Señal de salida automática



Fuente: Propia

Figura No. 65 Señal de salida automática con pausas



Fuente: Propia

La otra opción consiste en que el operador ingrese un valor a su antojo entre el 0% y el 100% subiendo o bajando en pasos del 5% de la señal mediante el control que provee la HMI (ver figura No 66 y figura No. 67) como puede apreciarse en la figura No. 68.

Figura No. 66 Arriba-abajo voltaje



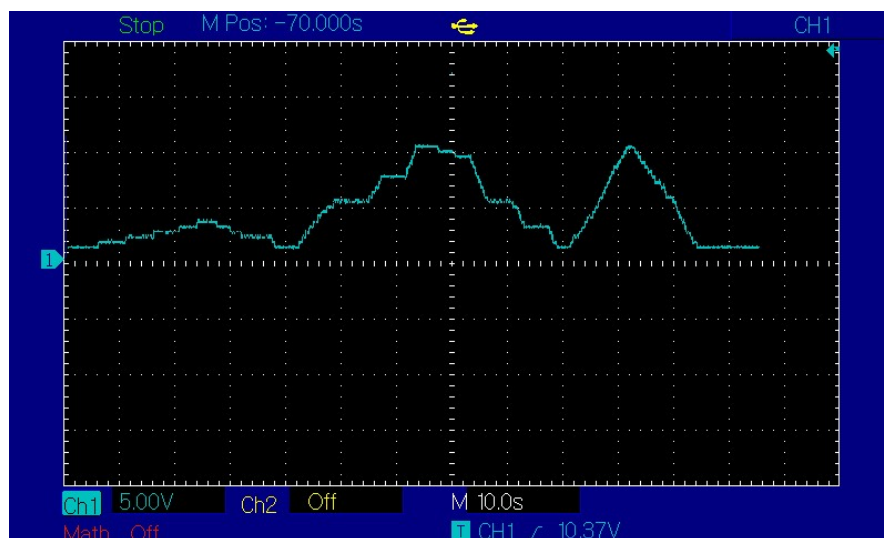
Fuente: Propia

Figura No. 67 Arriba-abajo corriente



Fuente: Propia

Figura No. 68 Señal de salida manual



Fuente: Propia

- Señal digital: esta subrutina es la encargada de aplicar una señal de voltaje de 24V en corriente continua a una de las entradas digitales del variador para así poder comprobar su lectura y respuesta, o a una de las salidas de relé para comprobar su conmutación (ver figura No. 69), en en la figura No. 70 se observa que el equipo de pruebas puede trabajar esta subrutina utilizando su fuente interna de 24 V o utilizando la fuente 24 V que tiene el variador (anexo “Fuentes”).

Figura No. 69 Señal digital



Fuente: Propia

Figura No. 70 Fuentes



Fuente: Propia

Una vez especificado donde se aplicará la señal (sea puerto digita o relé) y seleccionando la fuente con la cual se va a trabajar aparece la pantalla de activación o desactivación de la señal 24 V (ver figura No. 71) que se hace a voluntad del usuario como lo deja ver la figura “Generacion de señal digital” y su programación se encuentra en el anexo “código salida digital” y “código salida rele”.

Figura No. 71 On-off señal digital



Fuente: Propia

- Lectura señal: Esta señal es recibida del variador y varia en el rango de 0 a 20mA, lo que se hace es hacerla pasar por una resistencia de 165 Ohm para tener un divisor de tension y asi poder leer un voltaje tolerable para el Arduino, lo que se busca es corroborar que el variador esté produciendo dicha señal de una buena manera, como es una subrutina de solo lectura, es generada mediante el anexo “código leer señal” y se visualiza como se muestra en la figura No. 72.

Figura No. 72 Leer señal



Fuente: Propia

- Prueba semiconductores: esta subrutina consiste en la simulación de señales para pruebas de los semiconductores de potencia que hacen parte del variador de velocidad (IGBT/Tiristor). Lo que allí se hace es aplicar un voltaje 24V en la parte de potencia del semiconductor y generar una subrutina donde se aplican señales o disparos en la puerta del semiconductor bajo ciertas particularidades y de esta forma probar su conmutacion entre estado de conducción y corte, esta selección entre prueba de IGBT o tiristor se hace

en la pantalla que se muestra en la figura No. 73 y su programación se puede ver en el anexo “código prueba semiconductores”.

Figura No. 73 Pantalla prueba de semiconductores



Fuente: Propia

Dependiendo del tipo de semiconductor a probar se realiza una rutina diferente, al seleccionar la opción de tiristor se genera la pantalla de la figura No. 74, en la cual al presionar la tecla ON muestra en la pantalla el momento de activación de la señal 24 V que simula la potencia (ver figura No. 75), y la activación y desactivación de la señal de control (ver figuras No. 76 y figura No. 77), en la figura No, 78 se aprecia la señal azul que son los 24 V, y la señal amarilla que simula el disparo en la puerta para el control, una vez se quita la señal amarilla, el tiristor debe quedar en conducción hasta que se quite la señal azul que simula el cruce por cero, una vez completada la rutina el dispositivo dice al usuario si el tiristor se encuentra en buen estado o en falla de acuerdo a las lecturas que va haciendo mientras la prueba se lleva a cabo (ver figura No. 79 y figura No. 80). La programación de esta rutina se puede ver en el anexo “código prueba tiristor”.

Figura No. 74 Pantalla Tiristor



Fuente: Propia

Figura No. 75 Tiristor_1



Fuente: Propia

Figura No. 76 Tiristor_2



Fuente: Propia

Figura No. 77 Tiristor_3



Fuente: Ppropia

Figura No. 78 Generacion de prueba tiristor



Fuente: Propia

Figura No. 79 Tiristor OK



Fuente: Propia

Figura No. 80 Tiristor Falla



Fuente: Propia

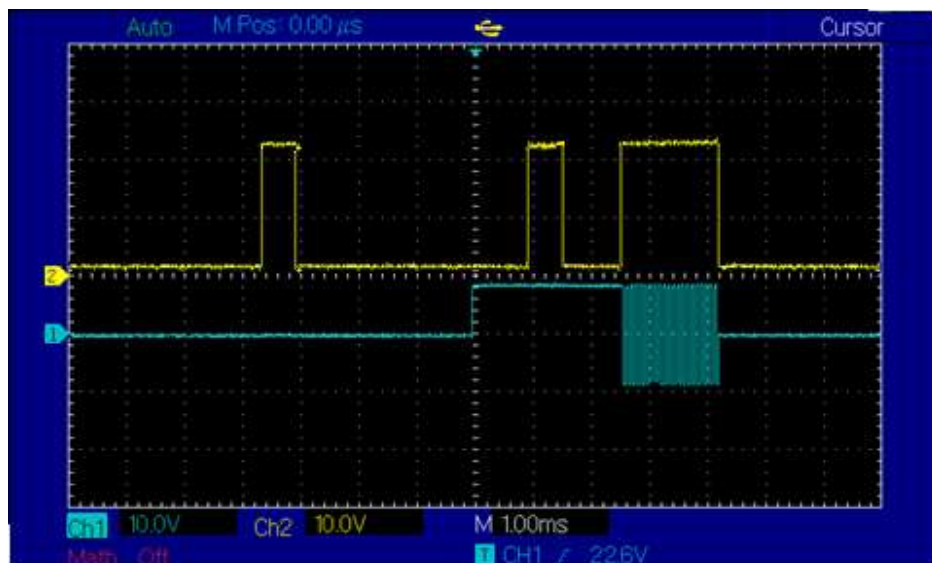
Para la prueba de IGBT, luego de seleccionar la opción correspondiente se genera la pantalla que se muestra en la figura No. 81, donde al oprimir el botón ON el Arduino inicia una rutina donde activa y desactiva la señal 24 V que simula la potencia, y también genera en puesta una señal de +9 V y -9 V para comprobar la conmutación entre conducción y corte del mismo, esta rutina se puede apreciar en la figura No. 82. Mientras el Arduino va generando estas señales de pruebas, también va haciendo lectura de la respuesta del IGBT (si se activa y se desactiva según lo las señales), para así poder dar un resultado de diagnostico final que se puede ver en la figura No. 83 y la figura No. 84.

Figura No. 81 Prueba de IGBT



Fuente: Propia

Figura No. 82 Generacion prueba de IGBT



Fuente: Propia

Figura No. 83 IGBT OK



Fuente: Propia

Figura No. 84 IGBT Falla



Fuente: Propia

9 CONCLUSIONES

En este capítulo de conclusiones, es muy grato para el investigador, terminar afirmando que el 100% del proyecto fue una satisfacción y realidad, culminada en su totalidad con los resultados esperados y un poco mas, pues frente al Objetivo General, el cual expresa diseñar e Implementar un equipo de diagnóstico temprano para las señales analógicas y digitales de los variadores de frecuencia ABB de las series ACS550, ACS800, ACS850 y ACS880 se cumplió a cabalidad ya que el dispositivo quedó completamente funcional y realiza la detección de fallas en las entradas analógicas y digitales de los variadores, y adicionalmente realiza pruebas de funcionamiento de IGBT y Tiristores con excelentes resultados.

Se puede concluir frente al Objetivos específicos No 1, donde se documenta las diferentes series en las que se implementará el sistema de detección de fallas tanto en el funcionamiento, como en los módulos de comunicación si se pudieron documentar las diferentes series de variadores ABB, aunque los módulos de comunicaciones no se incorporó la información, porque mientras se fue desarrollando el proyecto dicha información no me pareció relevante en el diseño e implementación del dispositivo.

Caracterizar los diferentes variadores de frecuencia con el fin de encontrar similitudes y diferencias entre las distintas series. Si se pudieron caracterizar y gracias a ello se encontraron similitudes en las impedancias de entradas y salidas de las tarjetas de control de cada una de las series de variadores ABB, lo cual facilitó el cumplimiento del siguiente objetivo porque con sus impedancias iguales, solo se debía hacer un circuito de acondicionamiento de señales que funcionaba en todas las series de variadores.

Diseñar e implementar el acondicionamiento de señales que del variador se entregan al sistema de detección, en este objetivo se pudieron acondicionar las señales que del drive se entregan al dispositivo de detección de fallas, pero adicionalmente se pudo realizar un acondicionamiento de señales que el dispositivo de detección de fallas entrega al variador para pruebas de respuestas del mismo, lo cual durante el desarrollo del proyecto, tomó más importancia que la misma lectura de señales del drive hacia el sistema detector.

Diseñar e implementar el software para el sistema, si se cumplió, porque la programación que se realizó para la placa de desarrollo arduino, generó como resultado cada una de las señales esperadas, las cuales creo que son las idóneas para realizar las pruebas de diagnóstico.

Mientras se realiza el proyecto, se contempla la posibilidad de generar una serie de señales simuladoras de encoder, por lo cual se hace la adición de esta etapa de pruebas.

10 BIBLIOGRAFÍA

- ARIZA O. William F. *Servo Mecanismo Digital: PWM y Amplificador Diferencial*. Bucaramanga, 2008, 96 p. Trabajo de grado (Ingeniero Electrónico). Obtenido el 11 de Febrero de 2015. Universidad Pontificia Bolivariana, Facultad de Ingeniería, Escuela de Ingeniería Electrónica. Disponible en:
http://repository.upb.edu.co:8080/jspui/bitstream/123456789/291/1/digital_15945.pdf
- CALDERON S. Juan C. *Análisis de fallas causadas por accionamientos de frecuencia variables en los rodamientos de motores de inducción*. Santiago de Cali, 2009, 104 p. Trabajo de grado (Ingeniero Electricista). Obtenido el 12 de Febrero de 2015. Universidad Autónoma de Occidente, Facultad de Ingeniería, Departamento de Energética y Mecánica. Disponible en:
<http://bdigital.uao.edu.co/bitstream/10614/1364/3/TEL00498.pdf>
- HART. Daniel W. *Electrónica de potencia*. Madrid, Pearson Educación S.A., 2001.
- MUHAMMAD H. Rashid. *Electrónica de potencia. Circuitos, dispositivos y aplicaciones*. Naucalpan de Juárez. Prentice Hall Hispanoamérica S.A., 1995.
- *Sistemas de Regulación y Controles Automáticos: Variadores de frecuencia* (n.d.). Obtenido el 17 de Febrero de 2015. Ingenieros.es, Portal de Ingenieros Españoles. Disponible en:
http://ingenieros.es/files/proyectos/Variadores_de_frecuencia.pdf
- *Variadores de frecuencia* (n.d.). Obtenido el 10 de Febrero de 2015. Página web de la Universidad Tecnológica Nacional, Facultad regional Avellaneda. Disponible en:
http://www.fra.utn.edu.ar/download/carreras/ingenierias/electrica/materias/planestudio/quintonivel/electronicall/apuntes/variadores_de_frecuencia.pdf
- VILLAMIZAR RINCÓN. Juan C. *Máquinas Eléctricas*. Bucaramanga, Sic Editorial Ltda., 2008.

11 ANEXOS

11.1 Anexo No 1: Características Generales de los variadores ABB

11.1.1 ACS550 Convertidores de frecuencia ABB de propósito general

Los convertidores de frecuencia ABB de propósito general son fáciles de comprar, instalar, configurar y utilizar, lo que permite ahorrar mucho tiempo; de ahí el uso del término “estándar”. Estos convertidores tienen una interfaz común de usuario y de proceso con bus de campo, así como herramientas de software comunes para el dimensionamiento, la puesta en marcha, el mantenimiento, y recambios comunes.

Aplicaciones

Los convertidores de frecuencia ABB estándar pueden emplearse en un amplio conjunto de industrias. Se usan frecuentemente en bombas, ventiladores y en aplicaciones de par constante, como por ejemplo las cintas transportadoras. Los convertidores ABB estándar resultan perfectos si se desea sencillez en la instalación, la puesta en marcha y el manejo, siendo igualmente apropiados en aquellos casos en los que no se requiera tecnología personalizada o específica para productos especiales.

Características principales

- Herramienta FlashDrop
- Panel de control asistente para un uso intuitivo
- Reactancia de auto inductancia variable para la reducción de los armónicos superiores
- Control vectorial
- Tarjetas barnizadas para entornos difíciles
- Filtro EMC integral de categoría C2 (1er entorno) como estándar
- Sistema de bus de campo flexible con Modbus integrado y numerosos adaptadores de bus de campo que se pueden montar internamente.

Datos técnicos

Conexión de red	
Rango de potencia y tensión	Trifásica, 380 a 480 V, +10/-15%, 0,75 a 355 kW Trifásica, 208 a 240 V, +10/-15%, 0,75 a 75 kW Autoidentificación de la línea de entrada
Frecuencia	48 a 63 Hz
Factor de potencia	0,98
Conexión del motor	
Tensión	Trifásica, de 0 a U_{ALIM}
Frecuencia	0 a 500 Hz
Capacidad de carga continua (par constante a una temperatura ambiente máxima de 40 °C)	Intensidad de salida nominal I_{2N}
Capacidad de sobrecarga (a una temperatura ambiente máxima de 40 °C)	En uso normal: $1,1 \times I_{2N}$ durante 1 minuto cada 10 minutos En uso en trabajo pesado: $1,5 \times I_{2nd}$ durante 1 minuto cada 10 minutos Independientemente del uso: $1,8 \times I_{2nd}$ durante 2 segundos cada 60 segundos
Frecuencia de conmutación	Por defecto 4 kHz Seleccionable 1 kHz, 2 kHz, 4 kHz, 8 kHz, 12 kHz
Tiempo de aceleración	0,1 a 1800 s
Tiempo de deceleración	0,1 a 1800 s
Control de la velocidad	
Bucle abierto	20% del deslizamiento del motor nominal
Bucle cerrado	0,1% de la velocidad nominal del motor
Bucle abierto	< 1% con escalón de par del 100%
Bucle cerrado	0,5% con escalón de par del 100%
Control del par	
Bucle abierto	< 10 ms con par nominal
Bucle cerrado	< 10 ms con par nominal
Bucle abierto	±5% con par nominal
Bucle cerrado	±2% con par nominal
Límites ambientales	
Temperatura ambiente	
De -15 a 50 °C	No se permite escarcha. De 40 a 50 °C con derrateo.
Altitud	Intensidad nominal disponible entre 0 y 1000 m.
Intensidad de salida	En altitudes de 1000 a 4000 m (3300 a 13 200 ft) por encima del nivel del mar, el derrateo es del 1% por cada 100 m (330 ft). Si el lugar de instalación está a una altitud superior a 2000 m (6600 ft) sobre el nivel del mar, contacte con su distribuidor u oficina de ABB local para más información.
Humedad relativa	5 a 95%, sin condensación
Grado de protección	IP 21 o IP 54 (≤ 160 kW)
Color del armario	NCS 1502-Y, RAL 9002, PMS 420 C
Niveles de contaminación	IEC721-3-3
Transporte	No se permite polvo conductor Clase 1C2 (gases químicos) Clase 1S2 (partículas sólidas)
Almacenamiento	Clase 2C2 (gases químicos) Clase 2S2 (partículas sólidas)
Funcionamiento	Clase 3C2 (gases químicos) Clase 3S2 (partículas sólidas)

Conexiones de control programables	
Dos entradas analógicas	
Señal de tensión	0 (2) a 10 V, $R_{in} > 312 \text{ k}\Omega$ sin diferencial
Señal de intensidad	0 (4) a 20 mA, $R_{in} = 100 \Omega$ sin diferencial
Valor de referencia del potenciómetro	10 V ± 2% máx. 10 mA, $R < 10 \text{ k}\Omega$
Máxima demora de tiempo	12 a 32 ms
Resolución	0,1%
Precisión	±1%
Dos salidas analógicas	0 (4) a 20 mA, carga < 500 Ω
Precisión	±3%
Tensión auxiliar	24 V CC ±10%, máx. 250 mA
Seis entradas digitales	12 a 24 V CC con alimentación interna o externa, PNP y NPN
Impedancia de entrada	2,4 k Ω
Máxima demora de tiempo	5 ms ± 1 ms
Tres salidas de relé	
Tensión máxima de conmutación	250 V CA/30 V CC
Intensidad máxima de conmutación	6 A/30 V CC; 1.500 V A/230 V CA
Intensidad continua máxima	2 A rms
Comunicación serie	
EIA-485	Protocolo Modbus
Cumplimiento de normativas del producto	
Directiva Europea de Baja Tensión 2006/95/CE	
Directiva Europea sobre Maquinaria 2006/42/CE	
Directiva EMC 2004/108/CE	
Sistema de control de calidad ISO 9001	
Sistema medioambiental ISO 14001	
Certificaciones UL, cUL, CE, C-Tick y GOST R	
Cumple la directiva RoHS	

Fuente: Manual Técnico, ACS 550

11.1.2 ACS800 Convertidores de frecuencia industriales.

Los convertidores de frecuencia industriales ABB se han diseñado para aplicaciones industriales y, en especial, para aplicaciones de industrias de procesos como las industrias de la pulpa y el papel, metalúrgica, minera, cementera, energética, química, petrolífera y del gas. Los convertidores de frecuencia industriales ABB están disponibles tanto como accionamientos de CA completos como en forma de módulos para satisfacer las demandas de los usuarios, los fabricantes de maquinaria y los integradores de sistemas. Estos convertidores son convertidores de CA de alta flexibilidad que pueden configurarse para satisfacer las necesidades específicas de las aplicaciones industriales, por lo que la configuración ajustada a cada pedido constituye una parte integral de la oferta. Los convertidores completos y los módulos de accionamiento cubren una amplia gama de potencias y tensiones, incluidas tensiones industriales de hasta 690 V. Los convertidores de frecuencia industriales ABB se entregan con una amplia gama de opciones integradas. Una característica clave de estos convertidores es su capacidad de programación, que facilita la adaptación a distintas aplicaciones⁶.

Diseño industrial

Los convertidores de frecuencia industriales ABB se han diseñado con unas especificaciones de intensidad que permiten utilizarlos en entornos industriales con aplicaciones que requieran una elevada capacidad de sobrecarga. El corazón del accionamiento figura el DTC, o control directo del par, que aporta un rendimiento elevado y ventajas significativas: por ejemplo, un control estático y dinámico preciso de la velocidad y el par, un elevado par de arranque y cables a motor largos. Las opciones integradas del accionamiento facilitan y agilizan las tareas de instalación. Los armarios y bastidores robustos, con una amplia gama de clases de protección, así como los terminales de alimentación, se han diseñado para su uso en entornos duros.

Uno de los criterios de diseño más significativos de los convertidores de frecuencia industriales ABB es su prolongada vida de servicio. De esta manera, las piezas sometidas a desgaste como los ventiladores y los condensadores se han seleccionado en consonancia. En combinación con las amplias características de protección, ello aporta una excelente fiabilidad en un mercado tan exigente como el industrial.

⁶Convertidores de frecuencia industriales ABB ACS800, convertidores únicos | Catálogo

Convertidores de frecuencia únicos

La configuración de convertidor de frecuencia único contiene un rectificador, un bus de CC y un inversor en una única unidad de accionamiento de CA.

Los convertidores de frecuencia únicos son accionamientos de CA completos que pueden instalarse sin necesidad de ningún armario o cerramiento adicional. Dichos convertidores de frecuencia están disponibles en forma de estructuras montadas en pared, autoportantes e instaladas en un armario. El grado de protección mínimo de los convertidores únicos es IP21, y están disponibles opcionalmente con grados de protección superiores⁷.

Seguridad funcional

La solución de seguridad funcional de ABB cumple los requisitos de la nueva Directiva sobre Maquinaria 2006/42/CE de la Unión Europea. Esta directiva está relacionada con normas como

IEC 62061 (Safety Integrity Level) e ISO 13849-1 (Performance Level), que exigen un grado de seguridad documentado

y probado y un concepto de seguridad en el ciclo de vida completo. La función safe torque off (STO) es una solución certificada que ofrece niveles de seguridad SIL 2 y PL d (Cat. 2)⁸.

Los convertidores de frecuencia ABB pueden equiparse opcionalmente con la función safe torque off (STO). Esta función puede utilizarse para prevenir una puesta en marcha imprevista y constituye una solución rentable y certificada para una seguridad básica. Otras funciones de seguridad que están incluidas en las unidades drives de armario son Safe Stop (SS1) y Safely-Limited Speed (SLS), las que se pueden certificar en los niveles de seguridad SIL 2 o PL d (Cat.2).

Convertidores de frecuencia montados en pared, ACS800-01

El convertidor de frecuencia montado en pared ACS800-01 ofrece todo lo necesario hasta un máximo de 200 kW. Todas las opciones y características importantes están integradas dentro del convertidor: reactancia de red, filtro EMC, chopper de frenado, etc. El usuario disfruta de todas estas características en un único y completo paquete IP21 o IP55. No obstante, el accionamiento también tiene un tamaño muy reducido. Una amplia gama de alternativas de software hacen que este accionamiento sea adecuado para cualquier aplicación.

⁷ Convertidores de frecuencia industriales ABB ACS800, convertidores únicos | Catálogo

⁸ Convertidores de frecuencia industriales ABB ACS800, convertidores únicos | Catálogo

Figura No. 85 Variador ACS800 para montaje en pared.



Fuente: Catalogo Técnico ACS800

Convertidores regenerativos montados en pared, ACS800-11

El convertidor regenerativo montado en pared ACS800-11 está equipado con una unidad de alimentación activa. Ofrece un convertidor regenerativo de pleno rendimiento en un único paquete compacto. Todas las características y opciones importantes están integradas en el convertidor, incluidos una línea LCL y un filtro EMC. Las especificaciones de potencia van de 5,5 kW hasta 110 kW. Está disponible con grado de protección IP21.

Figura No. 86 Variador ACS800 regenerativo para montaje en pared



Fuente: Catalogo Técnico ACS800

Convertidores montados en pared para armónicos reducidos, ACS800-31

El convertidor montado en pared para armónicos reducidos ACS800-31 ofrece una solución única integrada para los armónicos. Tiene un contenido en armónicos excepcionalmente reducido y cumple los requisitos sobre armónicos más estrictos sin necesidad de instalar filtros externos ni transformadores multi-pulsos.

El ACS800-31 montado en pared es un convertidor para armónicos reducidos en un paquete completo de hasta 110 kW. Al igual que otros convertidores montados en pared, cuenta con todas las características y opciones integradas. Está disponible con grado de protección IP21.

Figura No. 87 Variador ACS800 bajo armónicos monatado en pared



Fuente: Catalogo Técnico ACS800

Convertidores de frecuencia autoportantes, ACS800-02

El convertidor de frecuencia autoportante ACS800-02 es un e innovador bastidor de tipo estantería. Las especificaciones de potencia parten de 45 kW y llegan hasta 560 kW. El ACS800-02 está disponible en un armario IP21 muy compacto y ofreceen exclusiva dos direcciones de montaje. También ofrece un gran rango de opciones integradas incluyendo los filtros EMC, unidades de frenado o módulos de bus de campo.

Figura No. 88 Variador ACS800 Autoportante



Fuente: Catalogo Técnico ACS800

Convertidores de frecuencia instalados en armario, ACS800-07

El convertidor de frecuencia instalado en armario ACS800-07 ofrece configuraciones estandarizadas que pueden adaptarse a cualquier aplicación. Cubre una amplia gama de potencia hasta 2800 kW y es muy compacto: el accionamiento de mayor tamaño sólo tiene 3,2 metros de ancho. Está disponible con grado de protección IP21, IP22, IP42, IP54 e IP54R. Se dispone de una amplia gama de opciones integradas, y se ofrecen servicios de diseño de aplicaciones si se requiere adaptación.

Figura No. 89 Variador ACS800 instalado en armario



Fuente: Catalogo Técnico ACS800

Convertidores de frecuencia con refrigeración líquida, ACS800-07LC

El convertidor de frecuencia con refrigeración líquida ACS800 cuenta con un diseño robusto para aplicaciones de media y alta potencia. El armario, totalmente cerrado y de tamaño compacto, está optimizado para condiciones ambientales duras. La serie de producto ACS800 con refrigeración líquida ofrece una fiabilidad avanzada tanto para el sector industrial como el marítimo. La refrigeración líquida reduce el nivel de ruido al mínimo, mejora la transferencia de calor y hace innecesario un equipo de aire acondicionado.

Figura No. 90 Variador ACS800 con refrigeración líquida



Fuente: Catalogo Técnico ACS800

Especificaciones técnica.

Conexión de red	
Rango de potencia y tensión	trifásico, $U_{2IN} = 208$ a 240 V, $\pm 10\%$, excepto -07, -07LC, -17, -17LC, -37, -37LC trifásico, $U_{3IN} = 380$ a 415 V, $\pm 10\%$ trifásico, $U_{5IN} = 380$ a 500 V, $\pm 10\%$ trifásico, $U_{7IN} = 525$ a 690 V, $\pm 10\%$ (600 V UL, CSA)
Frecuencia	48 a 63 Hz
Factor de potencia	$\cos\phi_f = 0.98$ (fundamental) $\cos\phi = 0.93$ a 0.95 (total)
Factor de potencia (ACS800-11/-31/-17/-17LC/-37/-37LC)	$\cos\phi_f = 1$ (fundamental) $\cos\phi = 0.99$ (total)
Rendimiento (a potencia nominal)	
ACS800-0x	98%
ACS800-1x/-3x	97%

Conexión del motor	
Tensión para unidades > 500 V	Tensión de salida trifásica de 0 a $U_{2IN}/U_{3IN}/U_{5IN}/U_{7IN}$, véase "Tabla de selección del filtro para el ACS800" en la página 46, bajo los filtros du/dt.
Frecuencia	0 a ± 300 Hz (0 a ± 120 Hz con filtros du/dt opcionales)
Punto inicio debil. campo	8 a 300 Hz
Control del motor	Control directo del par de ABB (DTC)
Control del par:	Tiempo de incremento de par:
Bucle abierto	< 5 ms con par nominal
Bucle cerrado	< 5 ms con par nominal
	No linealidad:
Bucle abierto	$\pm 4\%$ con par nominal
Bucle cerrado	$\pm 3\%$ con par nominal
Control de velocidad:	Precisión estática:
Bucle abierto	10% del deslizamiento del motor
Bucle cerrado	0,01% de la velocidad nominal
	Precisión dinámica:
Bucle abierto	De 0,3 a 0,4%/segundo con incremento de par del 100%
Bucle cerrado	De 0,1 a 0,2%/segundo con incremento de par del 100%

Cumplimiento de normativas del producto	
CE	
Directiva Europea de Baja Tensión 2006/95/CE	
Directiva Europea sobre Maquinaria 2006/42/CE	
Directiva EMC 2006/108/CE	
Sistema de control de calidad ISO 9001 y sistema de gestión medioambiental ISO 14001	
UL, cUL 508A o 508C y CSA C22.2 N° 14-95, C-Tick, GOST R	

EMC conforme a EN 61800-3/A11 (2000), EN 61800-3 (2004)	
2° ambiente, distribución no restringida, categoría C3 - estándar en -07 (bastidor nXR8i), -07LC, -17, -17LC, -37 y -37LC (bastidor R7i-nXR8i), opcional en los otros.	
1° entorno, distribución restringida (categoría C2) como opción hasta una intensidad de entrada de 1000 A.	

Límites ambientales	
Temperatura ambiente	
Transporte	-40 a +70 °C
Almacenamiento	-40 a +70 °C
Funcionamiento	
Refrigeración por aire	-15 a +50 °C, no se permite escarcha +40 a 50 °C a intensidad de salida reducida (1% / 1 °C)
Refrigeración líquida	0 a +55 °C, no se permite escarcha +45 a 55 °C a intensidad de salida reducida (0,5% / 1 °C)
Método de refrigeración	
Refrigeración por aire	Aire limpio seco
Refrigeración líquida	Refrigeración líquida directa
Altitud	
0 a 1000 m	sin derrateo
1000 a 4000 m	con derrateo ~ (1% / 100 m) (unidades de 690 V: 1000 a 2000 m con derrateo)
Humedad relativa	5 a 95%, sin condensación
Categoría de protección	
IP21	estándar para -01, -11, -31, -02, -07, -17, -37
IP22	opción para -07, -17, -37
IP42	estándar para -07LC, -17LC, -37LC, opción para -07, -17, -37
IP54	opción para -07, -07LC, -17, -17LC, -37, -37LC
IP54R	opción para -07, -17, -37
IP55	opción para -01
R = conexión al conducto de salida de aire	
Pintura	-07, -07LC, -17, -17LC, -37, -37LC: RAL 7035 -01, -11, -31, -02: NCS 1502-Y (RAL 9002, PMS 420 C)
Niveles de contaminación	No se permite polvo conductor
Almacenamiento	IEC60721-3-1, Clase 1C2 (gases químicos), Clase 1S2 (partículas sólidas)
Transporte	IEC60721-3-2, Clase 2C2 (gases químicos), Clase 2S2 (partículas sólidas)
Funcionamiento	IEC60721-3-3, Clase 3C1/3C2* (gases químicos), Clase 3S2 (partículas sólidas)
Clasificación de vibración marítima	3 a 13,2 Hz: ± 1 mm amplitud (pico) 13,2 a 100 Hz: aceleración de 0,7 g

C = sustancias químicamente activas
S = sustancias mecánicamente activas
* tarjetas de circuitos barnizadas

Las opciones disponibles se detallan en el Resumen de características y la tabla de opciones. Véanse las páginas 62-63.

Fuente: Catalogo técnico, ACS800

11.1.3 Convertidores de frecuencia ABB para maquinaria ACS850

Los convertidores de frecuencia ABB para maquinaria han sido diseñados para satisfacer los requisitos de producción y rendimiento de fabricantes de maquinaria, integradores de sistemas, cuadristas y usuarios finales en una amplia gama de aplicaciones. Los convertidores ACS850 son idóneos para aplicaciones como grúas, extrusoras, cintas transportadoras, bobinadoras, bombas, ventiladores y mezcladores en industrias tales como las de manipulación de materiales, plásticos y caucho, alimentación y bebidas, textiles y metalurgia⁹.

Estos convertidores pueden ser configurados para satisfacer las necesidades precisas de cada sector, por lo que la configuración mediante pedido forma parte integral de esta oferta. Los convertidores abarcan una extensa gama de potencias y tensiones y, gracias a sus características de serie y opcionales, pueden programarse sin dificultad, lo que facilita su adaptación a diferentes aplicaciones.

Amplia gama de opciones

Los convertidores ACS850 ofrecen una gran variedad de opciones integradas, como distintas E/S y comunicaciones. Cuentan también con una amplia selección de accesorios externos. La flexibilidad y facilidad de programación de los módulos los hace aptos para numerosas aplicaciones en distintos ámbitos industriales.

Diseño robusto

Las especificaciones actuales de los convertidores de frecuencia ABB para maquinaria están pensadas para aplicaciones con elevados requisitos de sobrecarga. El corazón del convertidor es la plataforma de control motor DTC (control directo de par), que proporciona un control preciso de la velocidad y del par incluso sin realimentación. El convertidor está diseñado para tener una vida útil prolongada y, por esta razón, ciertos componentes como los ventiladores y los condensadores han sido seleccionados para ofrecer la máxima duración. Esto, junto con sus numerosas características de protección y detalles de diseño como el recubrimiento de las tarjetas de circuito impreso, permite ofrecer una fiabilidad excelente para el exigente mercado industrial.

⁹ Convertidores de frecuencia ABB para maquinaria, ACS850 | Catálogo Técnico

Figura No. 91 Variador ACS850 para maquinaria



Fuente: Catalogo técnico ACS850

Especificaciones técnicas

Conexión de red		Condiciones de funcionamiento	
Tensión de alimentación	Trifásica de 380 a 500 V CA +10/-15% Trifásica de 200 a 240 V CA ±10%	Grado de protección	IP20 conforme a EN 60529 (IP00 en bastidor G) De tipo abierto conforme a UL 508
Frecuencia	De 50 a 60 Hz ±5%	Temperatura ambiente	De -10 a +55 °C (14 a 131 °F), con derrateo por encima de 40 °C (104 °F) Sin congelación
Conexión de CC		Altitud de instalación	De 0 a 4000 m (0 a 13 000 ft) (red IT: 2000 m [6560 ft]), con derrateo por encima de 1000 m (3280 ft): 1%/100 m (328 ft)
Nivel de tensión de CC	De 485 a 675 V CC ±10% (tipos -5) De 270 a 324 V CC ±10% (tipos -2)	Humedad relativa	Máx. 95%, sin condensación
Circuito de carga	Interno en los bastidores de A a D Externo en los bastidores de E0 a G2	Condiciones atmosféricas/ ambientales	Clases 3K3 y 3C2 conforme a EN 60721-3-3. No admite neblinas de aceite, formación de hielo, condensación de humedad ni gotas, aspersion, salpicaduras y chorros de agua (EN 60204, Parte 1)
Conexión del motor		Vibración	Clase 3M4 conforme a EN 60721-3-3
Tipos de motor	Motores de inducción de CA, motores de imanes permanentes y motores síncronos de reluctancia	EMC (conforme a EN 61800-3)	Categorías C2 y C3 con filtro opcional (conforme a EN 61800-3)
Frecuencia de salida	De 0 a 500 Hz	Seguridad funcional	Safe Torque Off (STO) conforme a EN 61800-5-2) IEC 61508: SIL 3 IEC 61511: SIL3 EN 62061: SILCL 3 EN ISO 13849-1: PL e Homologación TÜV
Control del motor	Control DTC o escalar	Homologaciones	Bastidores de A a D: CE, GOST R, UL, cUL, CSA, C-Tick Bastidores de E0 a G: CE, GOST R; pendientes: UL, cUL, CSA, C-Tick
Control del par:	Tiempo de incremento de par:		
Bucle abierto	< 5 ms con par nominal		
Bucle cerrado	< 5 ms con par nominal		
	No linealidad:		
Bucle abierto	±4% con par nominal		
Bucle cerrado	±3% con par nominal		
Control de velocidad:	Precisión estática:		
Bucle abierto	10% del deslizamiento del motor		
Bucle cerrado	0,01% de la velocidad nominal		
	Precisión dinámica:		
Bucle abierto	De 0,3 a 0,4%/s con incremento de par del 100%		
Bucle cerrado	De 0,1 a 0,2%/s con incremento de par del 100%		
Conexión de potencia de frenado			
Chopper de frenado	De serie en los bastidores de A a D, opción integrada en el resto		
Resistencia de frenado	Resistencia externa conectada al convertidor		

Fuente: Catalogo Técnico ACS850

11.1.4 Convertidores de frecuencia industriales ABB- ACS880

Los convertidores de frecuencia de la gama ACS880 forman parte de la familia de convertidores de Compatibilidad Total de ABB. Son compatibles con prácticamente cualquier tipo de proceso, sistema de automatización, usuario y requisito de negocio, y han sido diseñados para trabajar con cualquier aplicación con motores, en cualquier sector e independientemente del rango de potencia. La innovación tras la Compatibilidad Total es la nueva arquitectura de nuestros convertidores de frecuencia que simplifica el funcionamiento, optimiza la eficiencia energética y contribuye a maximizar el rendimiento de los procesos. La gama ACS880 consiste en convertidores únicos, convertidores múltiples y módulos de convertidor.

Control directo de par (DTC)

La tecnología de control de motores propia de ABB ofrece un control preciso del par y la velocidad para todas las aplicaciones y para prácticamente cualquier tipo de motor de CA.

Unidad de memoria extraíble

Almacena todas las configuraciones de software y parámetros en un módulo fácil de sustituir e instalar.

Eficiencia energética

El convertidor de frecuencia ofrece características tales como un optimizador de energía e información de eficiencia energética que le ayudan a monitorizar y ahorrar la energía utilizada en los procesos.

Convertidores de frecuencia únicos montados en pared,

ACS880-01

Los convertidores de frecuencia únicos montados en pared están diseñados con la nueva arquitectura de la nueva familia de convertidores de Compatibilidad Total de ABB. Su diseño está preparado para cumplir con las necesidades de industrias como petróleo y gas, minería, metales, productos químicos, cemento, plantas de energía, manipulación de materiales, pulpa y

papel, madera e industrias navales y para controlar un amplio rango de aplicaciones como grúas, cabrestantes, extrusoras, transportadores, compresores, bombas y ventiladores. Los convertidores están disponibles en 9 diferentes tamaños de bastidor (R1 to R9) para una fácil instalación y puesta en marcha.

En el corazón del convertidor se encuentra el control directo de par (direct torque control, DTC), es la tecnología de control de motores de primer nivel de ABB. El vasto rango de opciones incluye los filtros EMC, encoders, resolver, filtros du/dt, filtros senoidales, reactancias y resistencias de frenado, herramienta de monitorización remota, así como software específico de aplicación. Las funciones de seguridad incorporadas reducen la necesidad de contar con componentes de seguridad externos.

Comunicación serie convertidor a convertidor para la sincronización de varios convertidores. Protección ambiental IP21 e IP55 para ambientes sucios y húmedos.

Figura No. 92 Convertidores ACS880 montados en pared.



Fuente Manual técnico, ACS880

Convertidores de frecuencia únicos instalados en armario, ACS880-07

Los convertidores de frecuencia únicos instalados en armario se fabrican bajo pedido, satisfaciendo las necesidades del cliente a pesar de los

desafíos técnicos. Diseñado bajo el concepto de arquitectura común ABB, estos convertidores compactos están disponibles en cuatro tamaños (de R6 a R9) para un ensamblaje y puesta en marcha simple.

Estos equipos están adaptados a las necesidades específicas de industrias como pulpa y papel, minería, metalurgia, cementera, energética, manipulación de materiales, madera, marina, química, petrolífera y del gas. Las aplicaciones típicas incluyen grúas, extrusoras, cabrestantes, cintas transportadoras, mezcladoras, compresores, bombas y ventiladores. El equipo contiene el rectificador, el bus de CC y el inversor, todo ello en un armario compacto. Las características y opcionales incluyen extensiones de entradas y salidas, opcionales de buses de campo, filtros du/dt, filtros EMC, resistencias de frenado, fusibles e interruptor principal.

De forma estándar, sin necesidad de ningún firmware adicional, el convertidor de frecuencia es capaz de controlar motores de inducción, motores síncronos y servomotores de inducción. Puede controlarlos tanto en lazo abierto como en lazo cerrado gracias a su plataforma de control de alta precisión, control directo de par (DTC). La incorporación de características de seguridad reduce la necesidad de elementos de seguridad externos.

Figura No. 93 Convertidores ACS880 instalados en armario



Fuente Catalogo técnico ACS880


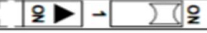

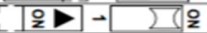
11.2 Anexo No. 2. CARACTERÍSTICAS ELÉCTRICAS DE TARJETAS ELECTRÓNICAS DE CONTROL DE LOS VARIADORES ABB.

11.2.1 TARJETA DE CONTROL ACS550

Tabla 1: Tabla de terminales de control ACS550

Tabla de terminales de control

A continuación se facilita información para conectar el cableado de control en X1 en el convertidor de frecuencia.

	X1	Descripción del hardware		
E/S analógica	1	SCR Terminal para la pantalla del cable de señal (apantallamiento). (Conectado internamente a tierra del chasis).		
	2	EA1 Canal 1 de entrada analógica, programable. Por defecto ² = referencia de frecuencia. Resolución 0,1%, exactitud ±1%. Es posible utilizar dos tipos distintos de conmutador DIP. J1: EA1 OFF: 0...10 V ($R_i = 312 \text{ kohmios}$)  J1: EA1 ON: 0...20 mA ($R_i = 100 \text{ ohmios}$) 		
			3	AGND Circuito de entrada analógica común (conectado internamente a tierra del chasis a través de 1 Mohmio).
			4	+10 V Fuente de referencia de potenciómetro: 10 V ±2%, max. 10 mA ($1 \text{ kohmio} \leq R \leq 10 \text{ kohmios}$).
	5	EA2 Canal 2 de entrada analógica, programable. Por defecto ² = sin usar. Resolución 0,1%, exactitud ±1%. Es posible utilizar dos tipos distintos de conmutador DIP. J1: EA2 OFF: 0...10 V ($R_i = 312 \text{ kohmios}$)  J1: EA2 ON: 0...20 mA ($R_i = 100 \text{ ohmios}$) 		
			6	AGND Circuito de entrada analógica común (conectado internamente a tierra del chasis a través de 1 Mohmio).
			7	AO1 Salida analógica, programable. Por defecto ² = frecuencia. 0...20 mA (carga < 500 ohmios). Precisión ±3%.
	8	AO2 Salida analógica, programable. Por defecto ² = intensidad. 0...20 mA (carga < 500 ohmios). Precisión ±3%.		
	9	AGND Circuito de salida analógica común (conectado internamente a tierra del chasis a través de 1 Mohmio).		
Entradas digitales ¹	10	+24V Salida de tensión auxiliar de 24 V CC / 250 mA (referencia a GND), protegida contra cortocircuito.		
	11	GND Salida de tensión auxiliar común (conectada internamente como flotante).		
	12	DCOM Entrada digital común. Para activar una entrada digital, debe haber $\geq +10 \text{ V}$ ($0 \leq -10 \text{ V}$) entre esta entrada y DCOM. Los 24 V pueden ser proporcionados por el ACS550 (X1-10) o por la fuente externa a 12...24 V de cualquier polaridad.		
	13	ED1 Entrada digital 1, programable. Por defecto ² = marcha/paro.		
	14	ED2 Entrada digital 2, programable. Por defecto ² = avance/retroceso.		
	15	ED3 Entrada digital 3, programable. Por defecto ² = sel. velocidad constante (código).		
	16	ED4 Entrada digital 4, programable. Por defecto ² = sel. velocidad constante (código).		
	17	ED5 Entrada digital 5, programable. Por defecto ² = sel. par de rampas (código).		
	18	ED6 Entrada digital 6, programable. Por defecto ² = sin usar.		

¹ Impedancia de entrada digital 1,5 koh míos. La tensión máxima para las entradas digitales es de 30 V.

Tabla 2: Tabla de terminales de control ACS550

		X1	Descripción del hardware
Salidas de relé	19	SR1C	 Salida de relé 1, programable. Por defecto ² = Listo Máximo: 250 V CA / 30 V CC, 2 A Mínimo: 500 mW (12 V, 10 mA)
	20	SR1A	
	21	SR1B	
	22	SR2C	 Salida de relé 2, programable. Por defecto ² = En marcha Máximo: 250 V CA / 30 V CC, 2 A Mínimo: 500 mW (12 V, 10 mA)
	23	SR2A	
	24	SR2B	
	25	SR3C	 Salida de relé 3, programable. Por defecto ² = Fallo (-1) Máximo: 250 V CA / 30 V CC, 2 A Mínimo: 500 mW (12 V, 10 mA)
	26	SR3A	
	27	SR3B	

Fuente: Manual del hardware, ACS550-01/U1

² Los valores por defecto dependen de la macro utilizada. Los valores especificados corresponden a la macro por defecto.

Nota: los terminales 3, 6 y 9 se encuentran al mismo potencial.

11.2.2 TARJETA DE CONTROL ACS800

Tabla 3 Especificaciones de la tarjeta RMIO

<i>Entradas Analógicas</i>	
	<i>Dos entradas de intensidad diferenciales programables (0 mA / 4 mA... 20 mA, Rin = 100 ohmios) y una entrada de tensión diferencial programable (-10 V / 0 V / 2 V ... +10 V, Rin = 200 kohmios). Las entradas analógicas están aisladas galvánicamente como un grupo.</i>
<i>Tensión de prueba de aislamiento</i>	500VCA, 1min
<i>Tensión de modo común máx. entre los canales</i>	±15 V CC

<i>Tasa de rechazo de modo común</i>	> 60 dB a 50 Hz
<i>Resolución</i>	0,025% (12 bits) para la entrada de -10 V... +10 V. 0,5% (11 bits) para las entradas de 0... +10 V y 0... 20 mA.
<i>Imprecisión</i>	±0,5% (rango de escala completa) a 25 °C (77 °F). Coeficiente de temperatura: ±100 ppm/°C (±56 ppm/°F), máx.
Salida de tensión constante	
<i>Tensión</i>	+10 V CC, 0, -10 V CC ± 0,5% (rango de escala completa) a 25 °C (77 °F). Coeficiente de temperatura: ±100 ppm/°C (±56 ppm/°F) máx.
<i>Carga máxima Potenciómetro aplicable</i>	10 mA
<i>Tensión</i>	1 kohmio a 10 kohmios
Salida de alimentación auxiliar	
<i>Tensión</i>	24 V CC ± 10%, a prueba de cortocircuito
<i>Tensión Intensidad máxima</i>	250 mA (compartida entre esta salida y los módulos opcionales instalados en la tarjeta RMIO)
Salidas analógicas	
	Dos salidas de intensidad programables: 0 (4) a 20 mA, $R_L < 700$ ohmios
<i>Resolución</i>	0,1% (10 bits)
<i>Imprecisión</i>	±1% (rango de escala completa) a 25 °C (77 °F). Coeficiente de temperatura: ±200 ppm/°C (±111 ppm/°F) máx.
Entradas Digitales	
	Seis entradas digitales programables (tierra común: 24 V CC, -15% al +20%) y una entrada de bloqueo de marcha. Aisladas en grupo, pueden dividirse en dos grupos aislados. Entrada de termistor: 5 mA, < 1,5 kohmios "1" (temperatura normal), > 4 kohmios "0" (temperatura alta), circuito abierto "0" (temperatura alta). Alimentación interna para entradas digitales (+24 V CC): a prueba de cortocircuito. Puede usarse una alimentación externa de 24 V CC en lugar de la alimentación interna.
<i>Tensión de prueba de aislamiento</i>	500VCA, 1min
<i>Umbrales lógicos</i>	< 8 V CC ... "0", > 12 V CC ... "1"
<i>Intensidad de entrada</i>	DI1 a DI5: 10 mA, DI6: 5 mA
<i>Constante de tiempo de filtro</i>	1 ms
Salidas de relé	

	Tres salidas de relé programables
Capacidad de conmutación	8 A a 24 V CC o 250 V CA, 0,4 A a 120 V CC
Intensidad continua mínima	5 mA rms a 24 V CC
Intensidad continua máxima	2 A rms
Tensión de prueba de aislamiento	4 kV CA, 1 minuto

A continuación se muestran las conexiones del cable de control externo a la tarjeta RMIO para el programa de control estándar del ACS800 (macro de Fábrica). Para las conexiones de control externo de otros programas y otras macros de control.

Tabla 4: Tabla de terminales de control ACS800

	1	1	VREF-	Tensión de referencia -10 V CC,
	2	2	AGND	$1\text{ kohmio} \leq R_L \leq 10\text{ kohmios}$
	X21			
	1	1	VREF+	Tensión de referencia 10 V CC,
	2	2	AGND	$1\text{ kohmio} \leq R_L \leq 10\text{ kohmios}$
	3	3	AI1+	Referencia de velocidad 0(2) ... 10 V,
	4	4	AI1-	$R_{in} = 200\text{ kohmios}$
	5	5	AI2+	Por defecto no se usan. 0(4) ... 20 mA,
	6	6	AI2-	$R_{in} = 100\text{ ohmios}$
	7	7	AI3+	Por defecto no se usan. 0(4) ... 20 mA,
	8	8	AI3-	$R_{in} = 100\text{ ohmios}$
	9	9	AO1+	Velocidad del motor 0(4)...20 mA \approx
	10	10	AO1-	0...veloc. nom. motor, $R_L \leq 700\text{ ohmios}$
	11	11	AO2+	Intensidad de salida 0(4)...20 mA \approx
	12	12	AO2-	0...intens. nom. motor, $R_L \leq 700\text{ ohmios}$
	X22			
	1	1	DI1	Paro/Marcha
	2	2	DI2	Avance/Retroceso ¹⁾
	3	3	DI3	No se usa
	4	4	DI4	Selección de aceleración y deceleración ²⁾
	5	5	DI5	Selección de velocidad constante ³⁾
	6	6	DI6	Selección de velocidad constante ³⁾
	7	7	+24 VD	+24 V CC máx. 100 mA
	8	8	+24 VD	
	9	9	DGND1	Tierra digital
	10	10	DGND2	Tierra digital
	11	11	DIIL	Bloqueo de marcha (0 = paro) ⁴⁾
	X23			
	1	1	+24 V	Salida y entrada de tensión auxiliar,
	2	2	GND	no aislada, 24 V CC, 250 mA ⁵⁾
	X25			
	1	1	RO1	Salida de relé 1: listo
	2	2	RO1	
	3	3	RO1	
	X26			
	1	1	RO2	Salida de relé 2: en marcha
	2	2	RO2	
	3	3	RO2	
	X27			
	1	1	RO3	Salida de relé 3: fallo (-1)
	2	2	RO3	
	3	3	RO3	

Fuente: Manual de Hardware. Convertidor de frecuencia ACS800-01

11.2.3 TARJETA DE CONTROL ACS850

Tabla 5: Tabla de terminales de control ACS850

XPOW		
Entrada de alimentación externa 24 V CC, 1,6 A	+24 V I	1
	GND	2
XRO1, XRO2, XRO3		
Salida de relé RO1 [listo] 250 V CA / 30 V CC 2 A	NO	1
	COM	2
	NC	3
Salida de relé RO2 [modulando] 250 V CA / 30 V CC 2 A	NO	4
	COM	5
	NC	6
Salida de relé RO3 [fallo (-1)] 250 V CA / 30 V CC 2 A	NO	7
	COM	8
	NC	9
XD24		
+24 V CC*	+24 VD	1
Tierra de entrada digital	DIGND	2
+24 V CC*	+24 VD	3
Tierra de entrada/salida digital	DIOGND	4
Puente de selección de tierra		A11
XDI		
Entrada digital DI1 [paro/arranque]	DI1	1
Entrada digital DI2	DI2	2
Entrada digital DI3 [restauración]	DI3	3
Entrada digital DI4	DI4	4
Entrada digital DI5	DI5	5
Entrada digital DI6 o entrada de termistor	DI6	6
Bloqueo de marcha (0 = paro)	D'IL	A
XDIO		
Entrada/salida digital DIO1 [salida: lista]	DIO1	1
Entrada/salida digital DIO2 [salida: en marcha]	DIO2	2
XAI		
Tensión de referencia (+)	+VREF	1
Tensión de referencia (-)	-VREF	2
Tierra	AGND	3
Entrada analógica AI1 (intensidad o tensión, seleccionable con el puente A11) [referencia de velocidad 1]	AI1+	4
	AI1-	5
Entrada analógica AI2 (intensidad o tensión, seleccionable con el puente A12)	AI2+	6
	AI2-	7
Puente de selección de intensidad/tensión AI1		A11
Puente de selección de intensidad/tensión AI2		A12
XAO		
Salida analógica AO1 [intensidad en %]	AO1+	1
	AO1-	2
Salida analógica AO2 [velocidad en %]	AO2+	3
	AO2-	4
XD2D		
Puente de terminación de enlace de convertidor a convertidor		T
Enlace de convertidor a convertidor	B	1
	A	2
	BGND	3
XSTO		
Función Safe torque off. Ambos circuitos deben estar cerrados para que el convertidor pueda ponerse en marcha.	OUT1	1
	OUT2	2
	IN1	3
	IN2	4

Fuente: Manual de Hardware. ACS850-04

Tabla 6 Especificaciones de la tarjeta JCU

<i>Alimentación</i>	24 V ($\pm 10\%$) CC, 1,6 A Suministrados desde la unidad de alimentación del convertidor o desde una fuente de alimentación externa a través del conector XPOW (paso 5 mm, tamaño del cable 2,5 mm ²)
<i>Salidas de relé RO1...RO3</i>	Paso del conector de 5 mm, tamaño del cable 2,5 mm ² 250 V CA / 30 V CC, 2 A Protegido por varistores Nota: Las salidas de relés del convertidor no cumplen los requisitos de protección para tensión ultrabaja (PELV) en instalaciones situadas a más de 4000 m (13 123 ft) si se utilizan con una tensión superior a los 48 V. En lugares de instalación situados entre 2000 m (6562 ft) y 4000 m (13 123 ft), los requisitos PELV no se cumplen si una o dos salidas de relés se utilizan con una tensión superior a los 48 V y las salidas de relés restantes se utilizan con una tensión inferior a los 48 V.
<i>Salida de +24 V</i>	Paso del conector de 5 mm, tamaño del cable de 2,5 mm ²
<i>Entradas digitales DI1...DI6</i>	Paso del conector de 3,5 mm, tamaño del cable de 1,5 mm ² Niveles lógicos a 24 V: "0" < 5 V, "1" > 15 V R _{in} : 2,0 kohmios Tipo de entrada: NPN/PNP (DI1...DI5), NPN (DI6) Filtrado: 0,25 ms DI6 (XDI: 6) puede utilizarse alternativamente como entrada para 1...3 termistores PTC. "0" > 4 kohmios, "1" < 1,5 kohmios I _{max} : 15 mA
<i>Entrada de bloqueo de marcha DIIL</i>	Tamaño de cable 1,5 mm ² Niveles lógicos a 24 V: "0" < 5 V, "1" > 15 V R _{in} : 2,0 kohmios Tipo de entrada: NPN/PNP Filtrado: 0,25 ms
<i>Entradas/salidas digitales DIO1 y DIO2</i>	Como entradas: Niveles lógicos a 24 V: "0" < 5 V, "1" > 15 V R _{in} : 2,0 kohmios Filtrado: 0,25 ms Como salidas: Intensidad de salida total limitada por las salidas de tensión auxiliares a 200 mA Tipo de salida: Emisor abierto
<i>Tensión de referencia para las entradas analógicas</i>	Paso del conector de 3,5 mm, tamaño del cable de 1,5 mm ² 10V $\pm 1\%$ y-10V $\pm 1\%$, R _{carga} > 1 kohmio

<i>+VREF y -VREF</i>	
<i>Entradas analógicas AI1 y AI2</i>	<i>Paso del conector de 3,5 mm, tamaño del cable de 1,5 mm² Intensidad de entrada: -20 ... 20 mA, R_{in}: 100 ohmios Tensión de entrada: -10 ... 10 V, R_{in}: 200 kohmios Entradas diferenciales, modo común ±20 V Intervalo de muestreo por canal: 0,25 ms Filtrado: 0,25 ms Resolución: 11 bits + bit de signo Imprecisión: 1% del intervalo de escala total</i>
<i>Salidas analógicas AO1 y AO2</i>	<i>Paso del conector de 3,5 mm, tamaño del cable de 1,5 mm² 0...20 mA, R_{carga} < 500 ohmios Intervalo de frecuencia: 0 ... 800 Hz Resolución: 11 bits + bit de signo Imprecisión: 2% de intervalo de escala total</i>
<i>Enlace de convertidor a convertidor</i>	<i>Paso del conector de 3,5 mm, tamaño del cable de 1,5 mm² Capa física: RS-485 Terminación mediante puente</i>

Fuente: Manual de Hardware. ACS850-04

11.2.4 TARJETA DE CONTROL ACS880

Tabla 7: Tabla de terminales de control ACS880

XPOW Entrada de alimentación externa		
1	+24V	24 V CC, 2 A
2	GND	
XAI Tensión de referencia y entradas analógicas		
1	+VREF	10 V CC, R_i 1...10 kohmios
2	-VREF	-10 V CC, R_i 1...10 kohmios
3	AGND	Tierra
4	AI1+	Ref. de velocidad 0(2)...10 V, $R_{en} > 200$ kohmios ¹⁾
5	AI1-	
6	AI2+	Por defecto no se usa. 0(4)...20 mA, $R_{en} > 100$ ohmios ²⁾
7	AI2-	
J1	J1	Puente de selección de intensidad/tensión AI1
J2	J2	Puente de selección de intensidad/tensión AI2
XAO Salidas analógicas		
1	AO1	Régimen de motor rpm 0...20 mA, $R_i < 500$ ohmios
2	AGND	
3	AO2	Intensidad de motor 0...20 mA, $R_i < 500$ ohmios
4	AGND	
XD2D Enlace de convertidor a convertidor		
1	B	Enlace de convertidor a convertidor
2	A	
3	BGND	
J3	J3	Terminador de enlace de convertidor a convertidor
XRO1, XRO2, XRO3 Salidas de relé		
1	NC	Listo 250 V CA / 30 V CC 2 A
2	COM	
3	NO	
1	NC	En marcha 250 V CA / 30 V CC 2 A
2	COM	
3	NO	
1	NC	Fallo(-1) 250 V CA / 30 V CC 2 A
2	COM	
3	NO	
XD24 Bloqueo de marcha		
1	DI1L	Por defecto no se usa.
2	+24 VD	+24 V CC 200 mA ³⁾
3	DICOM	Tierra de entrada digital
4	+24 VD	+24 V CC 200 mA ³⁾
5	DIOGND	Tierra de entrada/salida digital
J6		Interruptor de selección de tierra
XDIO Entradas/salidas digitales		
1	DIO1	Salida: Listo
2	DIO2	Salida: En marcha
XDI Entradas digitales		
1	DI1	Paro (0) / Marcha (1)
2	DI2	Avance (0) / Retroceso (1)
3	DI3	Restaurar
4	DI4	Selección de aceleración y deceleración ⁴⁾
5	DI5	Velocidad constante 1 (1 = activado)
6	DI6	Por defecto no se usa.
XSTO Safe Torque Off		
1	OUT1	Función "Safe Torque Off". Ambos circuitos deben estar cerrados para que el convertidor pueda ponerse en marcha.
2	SGND	
3	IN1	
4	IN2	
X12 Conexión de módulo de funciones de seguridad		
X13 Conexión del panel de control		

Fuente: Manual de Hardware. Convertidores ACS880-01

Notas:

- 1) Entrada de intensidad [0(4)...20 mA, $R_{en} > 100$ ohmios] o tensión [0(2)...10 V, $R_{en} > 200$ kohmios] seleccionada con el puente J1. El cambio de los ajustes requiere el reinicio de la unidad de control.
- 2) Entrada de intensidad [0(4)...20 mA, $R_{en} > 100$ ohmios] o tensión [0(2)...10 V, $R_{en} > 200$ kohmios] seleccionada con el puente J2. El cambio de los ajustes requiere el reinicio de la unidad de control.
- 3) La capacidad de carga total de estas salidas es de 4,8 W (200 mA / 24 V) menos la potencia consumida por DIO1 y DIO2.
- 4) 0 = abierto, 1 = cerrado

11.3 Anexo No. 3 CÓDIGO DE PROGRAMACIÓN

11.3.1 Código Inclusión de las librerías a utilizar

```
#include <UTFT.h> //Llama a la librería de la UTFT
#include <UTouch.h> //Llama a la librería UTouch
#include <UTFT_Geometry.h>
```

Fuente propia

11.3.2 Código Tamaño de fuente

```
extern uint8_t SmallFont[]; //Declara el tamaño de la fuente a utilizar
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];
```

Fuente propia

11.3.3 Código Configuración de modelo de pantalla

```
UTFT myGLCD(CTE70,25,26,27,28); //El modelo y los parámetros de la
pantalla 7" RS,WR,CS,RST
UTouch myTouch( 6, 5, 32, 3, 2); //comunica la touch con el Arduino
UTFT_Geometry geo(&myGLCD);
```

Fuente propia

11.3.4 Código Declaración de constantes globales

```
int x, y;
char stCurrent[20]="";
int stCurrentLen=0;
char stLast[20]="";
int k;
int h;
int m;
int p;
int a=8; // Digital Output
int s=9; // Digital Output
int d=10; // Digital Output
int f=11; // Digital Output
int g=12;
int readSi=A0;
int i;
```

```
int lectura;  
int lectura1;
```

Fuente propia

11.3.5 Código recuadro que indica seleccion

```
void indicadorSeleccion(int x1, int y1, int x2, int y2)  
{  
myGLCD.setColor(255, 0, 0);  
myGLCD.drawRoundRect (x1, y1, x2, y2);  
while (myTouch.dataAvailable())  
    myTouch.read();  
myGLCD.setColor(20, 155, 75);  
myGLCD.drawRoundRect (x1, y1, x2, y2);  
}
```

Fuente propia

11.3.6 Código menú principal

```
void menu()  
{  
    myGLCD.clrScr();  
    myGLCD.fillScr(255,255,255);  
    myGLCD.setFont(BigFont);  
    myGLCD.setColor(0, 0, 255);  
  
    myGLCD.setColor(0, 0, 0); //Color de fuente  
    myGLCD.setBackColor(255, 255, 255); //Color tras texto  
    myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,  
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y  
    myGLCD.setBackColor(255, 255, 255);  
    myGLCD.setColor(255,0,0);  
    myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);  
    geo.fillTriangle(377,43,389,43,389,31);  
    geo.fillTriangle(413,31,413,43,425,43);  
    geo.fillTriangle(413,67,413,79,425,67);  
    myGLCD.fillRect (389, 31, 413, 43);  
    myGLCD.fillRect (413, 43, 425, 67);  
  
    geo.fillTriangle(356,64,368,64,368,52);  
    geo.fillTriangle(392,52,392,64,404,64);  
    geo.fillTriangle(392,88,392,100,404,88);  
    myGLCD.fillRect (368, 52, 392, 64);  
    myGLCD.fillRect (392, 64, 404, 88);
```

```

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (20, 260, 218, 310);
myGLCD.fillRoundRect (226, 260, 408, 310);
myGLCD.fillRoundRect (414, 260, 598, 310);
myGLCD.fillRoundRect (607, 260, 790, 310);
myGLCD.fillRoundRect (321, 353, 474, 400);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (20, 260, 218, 310);
myGLCD.drawRoundRect (226, 260, 408, 310);
myGLCD.drawRoundRect (414, 260, 598, 310);
myGLCD.drawRoundRect (607, 260, 790, 310);
myGLCD.setBackColor(0,0,255);
myGLCD.print("ESCRIBIR SEÑALL", 22, 277);
myGLCD.print("LEER SEÑAL", 230, 277);
myGLCD.print("SEÑAL DIGITAL", 418, 277);
myGLCD.print("ENCODER", 610, 277);
myGLCD.print("PRUEBA SEMICONDUCTORES", CENTER, 370);

```

```

myGLCD.setColor(0,100,255);
myGLCD.fillCircle(50,438,30);
myGLCD.setColor(255,255,255);
myGLCD.fillRect (48, 416, 52, 433);
geo.drawArc(50,438,18,25,335,4);

```

```

myGLCD.setColor(75,100,255);
myGLCD.fillCircle(770,40,20);
myGLCD.setColor(255,255,255);
geo.drawArc(770,35,8,-75,180,2);
myGLCD.fillRect (770, 43, 772, 48);
myGLCD.fillRect (770, 53, 772, 55);

```

```

analogWrite(DAC1, 0);

```

```

while (true)
{
  if (myTouch.dataAvailable())
  {
    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=353) && (y<=400)) //encoder
    {
      if ((x>=321) && (x<=474))
      {
        indicadorSeleccion(321, 353, 474, 400);
      }
    }
  }
}

```

```
    encoder();
  }
}

if ((y>=260) && (y<=310)) //escribir señal
{
  if ((x>=20) && (x<218))
  {
    indicadorSeleccion(20, 260, 218, 310);
    menuSenalOut();
  }
  if ((x>=226) && (x<=408)) //leer señal
  {
    indicadorSeleccion(226, 260, 408, 310);
    lecturaDigitalSenal();
  }
  if ((x>=414) && (x<=598)) //señal digital
  {
    indicadorSeleccion(414, 260, 598, 310);
    senalRele();
  }
  if ((x>=607) && (x<=790)) //prueba semiconductores
  {
    indicadorSeleccion(607, 260, 790, 310);
    shotSignal();
  }
}
if ((x>=20) && (x<=80))
{
  if((y>=408) && (y<=468))
  {
    loop();
  }
}
}
```

Fuente propia

11.3.7 Código Encoder principal

```
void encoder()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
}
```

```

myGLCD.setFont(BigFont);
myGLCD.setColor(0, 0, 255);

myGLCD.setColor(0, 0, 0); //Color de fuente
myGLCD.setBackColor(255, 255, 255); //Color tras texto
myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
myGLCD.setBackColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

geo.fillTriangle(377,43,389,43,389,31);
geo.fillTriangle(413,31,413,43,425,43);
geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(75,100,255);
myGLCD.fillCircle(770,40,20);
myGLCD.setColor(255,255,255);
geo.drawArc(770,35,8,-75,180,2);
myGLCD.fillRect (770, 43, 772, 48);
myGLCD.fillRect (770, 53, 772, 55);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.fillRect (257, 260, 348, 310);
myGLCD.fillRect (452, 260, 543, 310);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackColor(0,0,255);
myGLCD.print("SALIR", 50, 417);
myGLCD.print("1024", 466, 277);
myGLCD.print("512", 279, 277);

while (true)
{
  if (myTouch.dataAvailable())
  {
    myTouch.read();
    x=myTouch.getX();

```

```
y=myTouch.getY();

if ((y>=400)&&(y<=450))
{
  if ((x>=40)&&(x<=125))
  {
    indicadorSeleccion(40, 400, 125, 450);
    menu();
  }
}
if ((y>=260) && (y<=310))
{
  if ((x>=257) && (x<=348))
  {
    indicadorSeleccion(257, 260, 348, 310);
    encoder512();
  }
  if ((x>=452) && (x<=543))
  {
    indicadorSeleccion(452, 260, 543, 310);
    encoder1024();
  }
}
}
}
}
```

Fuente propia

11.3.8 Código Encoder 512 ppr

```
void encoder512()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackgroundColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
  120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackgroundColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);
}
```

```

geo.fillTriangle(377,43,389,43,389,31);
geo.fillTriangle(413,31,413,43,425,43);
geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (181, 210, 392, 260);
myGLCD.fillRoundRect (400, 210, 608, 260);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (181, 210, 392, 260);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("HORARIO", 210, 228);
myGLCD.print("ANTIHORARIO", 401, 228);
myGLCD.print("SALIR", 50, 417);

while (true)
{
  if (myTouch.dataAvailable())
  {
    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=210) && (y<=260)) // Upper row
    {
      if ((x>=181) && (x<392))
      {
        indicadorSeleccion(181, 210, 392, 260);
        horario512();
      }
      if ((x>=400) && (x<=608)) // Button: 1
      {
        indicadorSeleccion(400, 210, 608, 260);
        antihorario512();
      }
    }
    if ((y>=400)&&(y<=450))
    {
      if ((x>=40)&&(x<=125))

```


11.3.10 Código 512 ppr 600 rpm en sentido horario

```
void horario512_600rpm()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
  geo.fillTriangle(392,88,392,100,404,88);
  myGLCD.fillRect (368, 52, 392, 64);
  myGLCD.fillRect (392, 64, 404, 88);

  myGLCD.setColor(0,0,255);
  myGLCD.print("600 rpm", CENTER, 333);
  myGLCD.print("HORARIO", CENTER, 355);

  myGLCD.setColor(0, 255, 0);
  myGLCD.fillRoundRect (310, 260, 375, 310);
  myGLCD.setColor(0,0,0);
  myGLCD.drawRoundRect (310, 260, 375, 310);
  myGLCD.setBackColor(0,255,0);
  myGLCD.print("ON", 325, 279);

  myGLCD.setColor(255, 0, 0);
  myGLCD.fillRoundRect (430, 260, 495, 310);
  myGLCD.setColor(0,0,0);
  myGLCD.drawRoundRect (430, 260, 495, 310);
  myGLCD.setBackColor(255,0,0);
  myGLCD.print("OFF", 438, 279);
```

```

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
        horario512();
      }
    }
    if ((y>=260) && (y<=310))
    {
      if ((x>=310) && (x<=375))
      {
        indicadorSeleccion(310, 260, 375, 310);
        onHorario512_600rpm();
      }
      if ((x>=430) && (x<=495))
      {
        indicadorSeleccion(430, 260, 495, 310);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
      }
    }
  }
}

```

```
}  
}  
}
```

Fuente propia

11.3.11 Código ON 512 ppr 600 rpm en sentido horario

```
void onHorario512_600rpm()  
{  
  for(k=k; k<80095; k=k+1)  
  {  
    digitalWrite(16, LOW);  
    digitalWrite(15, HIGH);  
    delayMicroseconds(44);  
  
    digitalWrite(16, LOW);  
    digitalWrite(15, LOW);  
    delayMicroseconds(44);  
  
    digitalWrite(16, HIGH);  
    digitalWrite(15, LOW);  
    delayMicroseconds(44);  
  
    digitalWrite(16, HIGH);  
    digitalWrite(15, HIGH);  
    delayMicroseconds(44);  
  }  
  horario512_600rpm();  
}
```

Fuente propia

11.3.12 Código 512 ppr 1800 rpm en sentido horario

```
Void horario512_1800rpm()  
{  
  myGLCD.clrScr();  
  myGLCD.fillScr(255,255,255);  
  myGLCD.setFont(BigFont);  
  myGLCD.setColor(0, 0, 255);  
  
  myGLCD.setColor(0, 0, 0); //Color de fuente  
  myGLCD.setBackColor(255, 255, 255); //Color tras texto  
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,  
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
```

```

myGLCD.setBackgroundColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

    geo.fillTriangle(377,43,389,43,389,31);
    geo.fillTriangle(413,31,413,43,425,43);
    geo.fillTriangle(413,67,413,79,425,67);
    myGLCD.fillRect (389, 31, 413, 43);
    myGLCD.fillRect (413, 43, 425, 67);

    geo.fillTriangle(356,64,368,64,368,52);
    geo.fillTriangle(392,52,392,64,404,64);
    geo.fillTriangle(392,88,392,100,404,88);
    myGLCD.fillRect (368, 52, 392, 64);
    myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0,0,255);
myGLCD.print("1800 rpm", CENTER, 333);
myGLCD.print("HORARIO", CENTER, 355);

myGLCD.setColor(0, 255, 0);
myGLCD.fillRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)

```



```

digitalWrite(16, LOW);
digitalWrite(15, HIGH);
delayMicroseconds(12);

digitalWrite(16, LOW);
digitalWrite(15, LOW);
delayMicroseconds(11);

digitalWrite(16, HIGH);
digitalWrite(15, LOW);
delayMicroseconds(11);

digitalWrite(16, HIGH);
digitalWrite(15, HIGH);
delayMicroseconds(12);
}
horario512_1800rpm();
}

```

Fuente propia

11.3.14 Código 512 ppr 3000 rpm en sentido horario

```

void horario512maxima()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
  120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
}

```

```

    geo.fillTriangle(392,88,392,100,404,88);
    myGLCD.fillRect (368, 52, 392, 64);
    myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0,0,255);
myGLCD.print("3000 rpm", CENTER, 333);
myGLCD.print("HORARIO", CENTER, 355);

myGLCD.setColor(0, 255, 0);
myGLCD.fillRoundRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
    if (myTouch.dataAvailable())
    {

        myTouch.read();
        x=myTouch.getX();
        y=myTouch.getY();

        if ((y>=400) && (y<=450)) // Upper row
        {
            if ((x>=40) && (x<=125)) // Button: 1
            {

```



```
    indicadorSeleccion(40, 400, 125, 450);
    digitalWrite(14, HIGH);
    digitalWrite(15, HIGH);
    digitalWrite(16, HIGH);
    horario512();
  }
}
if ((y>=260) && (y<=310))
{
  if ((x>=310) && (x<=375))
  {
    indicadorSeleccion(310, 260, 375, 310);
    onHorario512_3011rpm();
  }
  if ((x>=430) && (x<=495))
  {
    indicadorSeleccion(430, 260, 495, 310);
    digitalWrite(14, HIGH);
    digitalWrite(15, HIGH);
    digitalWrite(16, HIGH);
  }
}
}
}
}
}
```

Fuente propia

11.3.15 Código ON 512 ppr 3000 rpm en sentido horario

```
void onHorario512_3011rpm()
{
  for(k=k; k<400095; k=k+1)
  {
    digitalWrite(16, LOW);
    digitalWrite(15, HIGH);
    delayMicroseconds(4);

    digitalWrite(16, LOW);
    digitalWrite(15, LOW);
    delayMicroseconds(6);

    digitalWrite(16, HIGH);
    digitalWrite(15, LOW);
    delayMicroseconds(4);
  }
}
```

```

digitalWrite(16, HIGH);
digitalWrite(15, HIGH);
delayMicroseconds(5);
}
horario512maxima();
}

```

Fuente propia

11.3.16 Código 3 velocidades 512 ppr horario

```

void antihorario512()
{
myGLCD.clrScr();
myGLCD.fillScr(255,255,255);
myGLCD.setFont(BigFont);
myGLCD.setColor(0, 0, 255);

myGLCD.setColor(0, 0, 0); //Color de fuente
myGLCD.setBackColor(255, 255, 255); //Color tras texto
myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
myGLCD.setBackColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

geo.fillTriangle(377,43,389,43,389,31);
geo.fillTriangle(413,31,413,43,425,43);
geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (104, 210, 275, 260);
myGLCD.fillRoundRect (307, 210, 478, 260);
myGLCD.fillRoundRect (508, 210, 679, 260);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.setBackColor(0,0,255);
myGLCD.print("600 rpm", 129, 228);
}

```


11.3.17 Código 512 ppr 600 rpm en sentido antihorario

```
void antihorario512_600rpm()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackgroundColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackgroundColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
  geo.fillTriangle(392,88,392,100,404,88);
  myGLCD.fillRect (368, 52, 392, 64);
  myGLCD.fillRect (392, 64, 404, 88);

  myGLCD.setColor(0,0,255);
  myGLCD.print("600 rpm", CENTER, 333);
  myGLCD.print("ANTIHORARIO", CENTER, 355);

  myGLCD.setColor(0, 255, 0);
  myGLCD.fillRoundRect (310, 260, 375, 310);
  myGLCD.setColor(0,0,0);
  myGLCD.drawRoundRect (310, 260, 375, 310);
  myGLCD.setBackgroundColor(0,255,0);
  myGLCD.print("ON", 325, 279);

  myGLCD.setColor(255, 0, 0);
  myGLCD.fillRoundRect (430, 260, 495, 310);
  myGLCD.setColor(0,0,0);
  myGLCD.drawRoundRect (430, 260, 495, 310);
  myGLCD.setBackgroundColor(255,0,0);
  myGLCD.print("OFF", 438, 279);
}
```

```

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
        antihorario512();
      }
    }
    if ((y>=260) && (y<=310))
    {
      if ((x>=310) && (x<=375))
      {
        indicadorSeleccion(310, 260, 375, 310);
        onantiHorario512_600rpm();
      }
      if ((x>=430) && (x<=495))
      {
        indicadorSeleccion(430, 260, 495, 310);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
      }
    }
  }
}

```

```
}  
}  
}
```

Fuente propia

11.3.18 Código ON 512 ppr 600 rpm en sentido antihorario

```
void onantiHorario512_600rpm()  
{  
  for(k=k; k<80095; k=k+1)  
  {  
    digitalWrite(16, LOW);  
    digitalWrite(15, HIGH);  
    delayMicroseconds(44);  
  
    digitalWrite(15, HIGH);  
    digitalWrite(16, HIGH);  
    delayMicroseconds(44);  
  
    digitalWrite(16, HIGH);  
    digitalWrite(15, LOW);  
    delayMicroseconds(44);  
  
    digitalWrite(15, LOW);  
    digitalWrite(16, LOW);  
    delayMicroseconds(44);  
  }  
  antihorario512_600rpm();  
}
```

Fuente propia

11.3.19 Código 512 ppr 1800 rpm en sentido antihorario

```
void antihorario512_1800rpm()  
{  
  myGLCD.clrScr();  
  myGLCD.fillScr(255,255,255);  
  myGLCD.setFont(BigFont);  
  myGLCD.setColor(0, 0, 255);  
  
  myGLCD.setColor(0, 0, 0); //Color de fuente  
  myGLCD.setBackColor(255, 255, 255); //Color tras texto  
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,  
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
```

```

myGLCD.setBackgroundColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

    geo.fillTriangle(377,43,389,43,389,31);
    geo.fillTriangle(413,31,413,43,425,43);
    geo.fillTriangle(413,67,413,79,425,67);
    myGLCD.fillRect (389, 31, 413, 43);
    myGLCD.fillRect (413, 43, 425, 67);

    geo.fillTriangle(356,64,368,64,368,52);
    geo.fillTriangle(392,52,392,64,404,64);
    geo.fillTriangle(392,88,392,100,404,88);
    myGLCD.fillRect (368, 52, 392, 64);
    myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0,0,255);
myGLCD.print("1800 rpm", CENTER, 333);
myGLCD.print("ANTIHORARIO", CENTER, 355);

myGLCD.setColor(0, 255, 0);
myGLCD.fillRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)

```



```

digitalWrite(16, LOW);
digitalWrite(15, HIGH);
delayMicroseconds(12);

digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
delayMicroseconds(11);

digitalWrite(16, HIGH);
digitalWrite(15, LOW);
delayMicroseconds(11);

digitalWrite(15, LOW);
digitalWrite(16, LOW);
delayMicroseconds(12);
}
antihorario512_1800rpm();
}

```

Fuente propia

11.3.21 Código 512 ppr 3000 rpm en sentido antihorario

```

void antihorario512maxima()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
  120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
}

```

```

geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0,0,255);
myGLCD.print("3000 rpm", CENTER, 333);
myGLCD.print("ANTIHORARIO", CENTER, 355);

myGLCD.setColor(0, 255, 0);
myGLCD.fillRoundRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {

```



```

digitalWrite(15, LOW);
digitalWrite(16, LOW);
delayMicroseconds(5);
}
antihorario512maxima();
}

```

Fuente propia

11.3.23 Código Encoder 1024 ppr

```

void encoder1024()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackgroundColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackgroundColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
  geo.fillTriangle(392,88,392,100,404,88);
  myGLCD.fillRect (368, 52, 392, 64);
  myGLCD.fillRect (392, 64, 404, 88);

  myGLCD.setColor(0, 0, 255);
  myGLCD.fillRoundRect (181, 210, 392, 260);
  myGLCD.fillRoundRect (400, 210, 608, 260);
  myGLCD.fillRoundRect (40, 400, 125, 450);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (181, 210, 392, 260);
  myGLCD.setBackgroundColor(0,0,255);
  myGLCD.print("HORARIO", 210, 228);
}

```



```

myGLCD.setColor(0, 0, 255);

myGLCD.setColor(0, 0, 0); //Color de fuente
myGLCD.setBackgroundColor(255, 255, 255); //Color tras texto
myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
myGLCD.setBackgroundColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

geo.fillTriangle(377,43,389,43,389,31);
geo.fillTriangle(413,31,413,43,425,43);
geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (104, 210, 275, 260);
myGLCD.fillRect (307, 210, 478, 260);
myGLCD.fillRect (508, 210, 679, 260);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("300 rpm", 129, 228);
myGLCD.print("900 rpm", 332, 228);
myGLCD.print("1500 rpm", 528, 228);
myGLCD.print("SALIR", 50, 417);

while (true)
{
  if (myTouch.dataAvailable())
  {
    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=210) && (y<=260)) // Upper row
    {
      if ((x>=101) && (x<=272))
      {
        indicadorSeleccion(104, 210, 275, 260);
      }
    }
  }
}

```



```

geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0,0,255);
myGLCD.print("300 rpm", CENTER, 333);
myGLCD.print("HORARIO", CENTER, 355);

myGLCD.setColor(0, 255, 0);
myGLCD.fillRoundRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();

```



```

delayMicroseconds(44);

digitalWrite(16, HIGH);
digitalWrite(15, LOW);
delayMicroseconds(44);

digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
delayMicroseconds(44);
}

```

Fuente propia

11.3.27 Código 1024 ppr 900 rpm en sentido horario

```

void horario1024_900rpm()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
  geo.fillTriangle(392,88,392,100,404,88);
  myGLCD.fillRect (368, 52, 392, 64);
  myGLCD.fillRect (392, 64, 404, 88);

  myGLCD.setColor(0,0,255);
  myGLCD.print("900 rpm", CENTER, 333);
  myGLCD.print("HORARIO", CENTER, 355);
}

```

```

myGLCD.setColor(0, 255, 0);
myGLCD.fillRoundRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
        horario1024();
      }
    }
  }
}

```


11.3.29 Código 1024 ppr 1500 rpm en sentido horario

```
void horario1024maxima()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackgroundColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackgroundColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
  geo.fillTriangle(392,88,392,100,404,88);
  myGLCD.fillRect (368, 52, 392, 64);
  myGLCD.fillRect (392, 64, 404, 88);

  myGLCD.setColor(0,0,255);
  myGLCD.print("1500 rpm", CENTER, 333);
  myGLCD.print("HORARIO", CENTER, 355);

  myGLCD.setColor(0, 255, 0);
  myGLCD.fillRoundRect (310, 260, 375, 310);
  myGLCD.setColor(0,0,0);
  myGLCD.drawRoundRect (310, 260, 375, 310);
  myGLCD.setBackgroundColor(0,255,0);
  myGLCD.print("ON", 325, 279);

  myGLCD.setColor(255, 0, 0);
  myGLCD.fillRoundRect (430, 260, 495, 310);
  myGLCD.setColor(0,0,0);
  myGLCD.drawRoundRect (430, 260, 495, 310);
  myGLCD.setBackgroundColor(255,0,0);
  myGLCD.print("OFF", 438, 279);
```

```

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
        horario1024();
      }
    }
    if ((y>=260) && (y<=310))
    {
      if ((x>=310) && (x<=375))
      {
        indicadorSeleccion(310, 260, 375, 310);
        onHorario1024_1500rpm();
      }
      if ((x>=430) && (x<=495))
      {
        indicadorSeleccion(430, 260, 495, 310);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
      }
    }
  }
}

```

```
}  
}  
}
```

Fuente propia

11.3.30 Código ON 1024 ppr 1500 rpm en sentido horario

```
void onHorario1024_1500rpm()  
{  
  for(k=k; k<400095; k=k+1)  
  {  
    digitalWrite(16, LOW);  
    digitalWrite(15, HIGH);  
    delayMicroseconds(4);  
  
    digitalWrite(16, LOW);  
    digitalWrite(15, LOW);  
    delayMicroseconds(6);  
  
    digitalWrite(16, HIGH);  
    digitalWrite(15, LOW);  
    delayMicroseconds(4);  
  
    digitalWrite(16, HIGH);  
    digitalWrite(15, HIGH);  
    delayMicroseconds(5);  
  }  
  horario1024maxima();  
}
```

Fuente propia

11.3.31 Código 3 velocidades 1024 ppr antihorario

```
void antihorario1024()  
{  
  myGLCD.clrScr();  
  myGLCD.fillScr(255,255,255);  
  myGLCD.setFont(BigFont);  
  myGLCD.setColor(0, 0, 255);  
  
  myGLCD.setColor(0, 0, 0); //Color de fuente  
  myGLCD.setBackColor(255, 255, 255); //Color tras texto  
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,  
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
```

```

myGLCD.setBackgroundColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

    geo.fillTriangle(377,43,389,43,389,31);
    geo.fillTriangle(413,31,413,43,425,43);
    geo.fillTriangle(413,67,413,79,425,67);
    myGLCD.fillRect (389, 31, 413, 43);
    myGLCD.fillRect (413, 43, 425, 67);

    geo.fillTriangle(356,64,368,64,368,52);
    geo.fillTriangle(392,52,392,64,404,64);
    geo.fillTriangle(392,88,392,100,404,88);
    myGLCD.fillRect (368, 52, 392, 64);
    myGLCD.fillRect (392, 64, 404, 88);

/*myGLCD.setColor(185,120,115);
myGLCD.print("SELECCIONE LA", CENTER, 280);
myGLCD.print("VELOCIDAD DESEADA", CENTER, 295);*/

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (104, 210, 275, 260);
myGLCD.fillRect (307, 210, 478, 260);
myGLCD.fillRect (508, 210, 679, 260);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(255, 255, 255);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("300 rpm", 129, 228);
myGLCD.print("900 rpm", 332, 228);
myGLCD.print("1500 rpm", 528, 228);
myGLCD.print("BACK", 50, 417);

while (true)
{
    if (myTouch.dataAvailable())
    {
        myTouch.read();
        x=myTouch.getX();
        y=myTouch.getY();

        if ((y>=210) && (y<=260)) // Upper row
        {
            if ((x>=101) && (x<=272))
            {
                indicadorSeleccion(104, 210, 275, 260);
                antihorario1024_300rpm();
            }
        }
    }
}

```



```

myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0,0,255);
myGLCD.print("300 rpm", CENTER, 333);
myGLCD.print("ANTIHORARIO", CENTER, 355);

myGLCD.setColor(0, 255, 0);
myGLCD.fillRoundRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

```

```
if ((y>=400) && (y<=450)) // Upper row
{
  if ((x>=40) && (x<=125)) // Button: 1
  {
    indicadorSeleccion(40, 400, 125, 450);
    digitalWrite(14, HIGH);
    digitalWrite(15, HIGH);
    digitalWrite(16, HIGH);
    antihorario1024();
  }
}
if ((y>=260) && (y<=310))
{
  if ((x>=310) && (x<=375))
  {
    indicadorSeleccion(310, 260, 375, 310);
    onantiHorario1024_300rpm();
  }
  if ((x>=430) && (x<=495))
  {
    indicadorSeleccion(430, 260, 495, 310);
    digitalWrite(14, HIGH);
    digitalWrite(15, HIGH);
    digitalWrite(16, HIGH);
  }
}
}
}
}
```

Fuente propia

11.3.33 Código ON 1024 ppr 300 rpm en sentido antihorario

```
void onantiHorario1024_300rpm()
{
  for(k=k; k<80095; k=k+1)
  {
    digitalWrite(16, LOW);
    digitalWrite(15, HIGH);
    delayMicroseconds(44);

    digitalWrite(15, HIGH);
    digitalWrite(16, HIGH);
    delayMicroseconds(44);
  }
}
```

```

digitalWrite(16, HIGH);
digitalWrite(15, LOW);
delayMicroseconds(44);

digitalWrite(15, LOW);
digitalWrite(16, LOW);
delayMicroseconds(44);
}
antihorario1024_300rpm();
}

```

Fuente propia

11.3.34 Código 1024 ppr 900 rpm en sentido antihorario

```

void antihorario1024_900rpm()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackgroundColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
  120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackgroundColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
  geo.fillTriangle(392,88,392,100,404,88);
  myGLCD.fillRect (368, 52, 392, 64);
  myGLCD.fillRect (392, 64, 404, 88);

  myGLCD.setColor(0,0,255);
  myGLCD.print("900 rpm", CENTER, 333);
  myGLCD.print("ANTIHORARIO", CENTER, 355);
}

```

```

myGLCD.setColor(0, 255, 0);
myGLCD.fillRoundRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
        antihorario1024();
      }
    }
  }
  if ((y>=260) && (y<=310))

```


11.3.36 Código 1024 ppr 1500 rpm en sentido antihorario

```
void antihorario1024maxima()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
  geo.fillTriangle(392,88,392,100,404,88);
  myGLCD.fillRect (368, 52, 392, 64);
  myGLCD.fillRect (392, 64, 404, 88);

  myGLCD.setColor(0,0,255);
  myGLCD.print("1500 rpm", CENTER, 333);
  myGLCD.print("ANTIHORARIO", CENTER, 355);

  myGLCD.setColor(0, 255, 0);
  myGLCD.fillRoundRect (310, 260, 375, 310);
  myGLCD.setColor(0,0,0);
  myGLCD.drawRoundRect (310, 260, 375, 310);
  myGLCD.setBackColor(0,255,0);
  myGLCD.print("ON", 325, 279);

  myGLCD.setColor(255, 0, 0);
  myGLCD.fillRoundRect (430, 260, 495, 310);
  myGLCD.setColor(0,0,0);
  myGLCD.drawRoundRect (430, 260, 495, 310);
  myGLCD.setBackColor(255,0,0);
  myGLCD.print("OFF", 438, 279);
```

```

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

digitalWrite(14, HIGH);
digitalWrite(15, HIGH);
digitalWrite(16, HIGH);
k=0;

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
        antihorario1024();
      }
    }
    if ((y>=260) && (y<=310))
    {
      if ((x>=310) && (x<=375))
      {
        indicadorSeleccion(310, 260, 375, 310);
        onantiHorario1024_1500rpm();
      }
      if ((x>=430) && (x<=495))
      {
        indicadorSeleccion(430, 260, 495, 310);
        digitalWrite(14, HIGH);
        digitalWrite(15, HIGH);
        digitalWrite(16, HIGH);
      }
    }
  }
}

```



```
}  
}  
}
```

Fuente propia

11.3.37 Código ON 1024 ppr 1500 rpm en sentido antihorario

```
void onantiHorario1024_1500rpm()  
{  
  for(k=k; k<400095; k=k+1)  
  {  
    digitalWrite(16, LOW);  
    digitalWrite(15, HIGH);  
    delayMicroseconds(4);  
  
    digitalWrite(16, HIGH);  
    digitalWrite(15, HIGH);  
    delayMicroseconds(6);  
  
    digitalWrite(16, HIGH);  
    digitalWrite(15, LOW);  
    delayMicroseconds(4);  
  
    digitalWrite(15, LOW);  
    digitalWrite(16, LOW);  
    delayMicroseconds(5);  
  }  
  antihorario1024maxima();  
}
```

Fuente propia

11.3.38 Código Prueba semiconductores

```
Void shotSignal()  
{  
  myGLCD.clrScr();  
  myGLCD.fillScr(255,255,255);  
  myGLCD.setFont(BigFont);  
  myGLCD.setColor(0, 0, 255);  
  
  digitalWrite(8, HIGH);  
  digitalWrite(9, HIGH);  
  
  myGLCD.setColor(0, 0, 0); //Color de fuente
```

```

myGLCD.setBackColor(255, 255, 255); //Color tras texto
myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); myGLCD.setBackColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

geo.fillTriangle(377,43,389,43,389,31);
geo.fillTriangle(413,31,413,43,425,43);
geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (191, 210, 392, 260);
myGLCD.fillRoundRect (405, 210, 603, 260);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (191, 210, 392, 260);
myGLCD.drawRoundRect (405, 210, 603, 260);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackColor(0,0,255);
myGLCD.print("TIRISTOR", 220, 228);
myGLCD.print("IGBT", 470, 228);
myGLCD.print("SALIR", 50, 417);

while (true)
{
if (myTouch.dataAvailable())
{
myTouch.read();
x=myTouch.getX();
y=myTouch.getY();

if ((y>=210) && (y<=260)) // Upper row
{
if ((x>=191) && (x<392))
{
indicadorSeleccion(191, 210, 392, 260);
tiristor();
}
}
if ((x>=405) && (x<=603)) // Button: 1

```



```

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(8, HIGH);
        digitalWrite(9, HIGH);
        digitalWrite(9, LOW);
      }
    }
    if ((y>=260) && (y<=310))
    {
      if ((x>=310) && (x<=375))
      {
        indicadorSeleccion(310, 260, 375, 310);
        digitalWrite(8, LOW);
        myGLCD.setColor(255, 255, 255);
        myGLCD.fillRect (280, 128, 515, 153);
        myGLCD.setColor(0, 0, 0); //Color de fuente
        myGLCD.setBackgroundColor(255, 255, 255);
        myGLCD.print("POTENCIA ACTIVA", CENTER, 130);
        delay(1000);
        digitalWrite(9, LOW);
        myGLCD.setColor(255, 255, 255);
        myGLCD.fillRect (195, 178, 595, 205);
        myGLCD.setColor(20, 190, 0); //Color de fuente
        myGLCD.setBackgroundColor(255, 255, 255);
        myGLCD.print("SEÑAL DE CONTROL ACTIVA", CENTER, 180);
        delay(1000);
        digitalWrite(9, HIGH);
        myGLCD.setColor(0, 110, 0); //Color de fuente

```

```

myGLCD.print("SEÑAL DE CONTROL INACTIVA", CENTER, 180);
h=digitalRead(11);
if (digitalRead(11)==LOW)
{
  myGLCD.setColor(255, 255, 255);
  myGLCD.fillRect (195, 335, 585, 363);
  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.print("TIRISTOR IS A", 230, 340);
  myGLCD.setColor(190, 20, 0);
  myGLCD.print("FALLA", 455, 340);
  myGLCD.print(":", CENTER, 85);
  myGLCD.setColor(0, 0, 0);
  myGLCD.print("POTENCIA INACTIVA", CENTER, 130);
  digitalWrite(9, HIGH);
}
myGLCD.printNum1(digitalRead(11), CENTER, 375);
}
if ((x>=430) && (x<=495))
{
  h=digitalRead(11);
  indicadorSeleccion(430, 260, 495, 310);
  digitalWrite(8, HIGH);
  myGLCD.setColor(0, 0, 0);
  myGLCD.print("POTENCIA INACTIVA", CENTER, 130);
  digitalWrite(9, HIGH);

  if (h=HIGH)
  {
    if (digitalRead(11)==LOW)
    {
      myGLCD.setColor(255, 255, 255);
      myGLCD.fillRect (195, 335, 585, 363);
      myGLCD.setColor(0, 0, 0); //Color de fuente
      myGLCD.print("TIRISTOR ESTA", 285, 340);
      myGLCD.setColor(20, 190, 0);
      myGLCD.print("OK", 480, 340);
    }
    if (digitalRead(11)==HIGH)
    {
      myGLCD.setColor(255, 255, 255);
      myGLCD.fillRect (195, 335, 585, 363);
      myGLCD.setColor(0, 0, 0); //Color de fuente
      myGLCD.print("TIRISTOR ESTA", 230, 340);
      myGLCD.setColor(230, 20, 0);
      myGLCD.print("FALLA", 455, 340);
    }
  }
}
}

```



```

myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

myGLCD.setBackgroundColor(255,255,255);

while (true)
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    digitalWrite(8, HIGH);

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(8, HIGH);
        digitalWrite(9, HIGH);
        shotSignal();
      }
    }
    if ((y>=260) && (y<=310))
    {
      if ((x>=310) && (x<=375))
      {
        indicadorSeleccion(310, 260, 375, 310);
        digitalWrite(8, HIGH);
        delay(100);
      }
    }
  }
}

```

```

digitalWrite(8, LOW);
delay(500);
digitalWrite(9, LOW);
delay(500);
digitalWrite(9, HIGH);
delay(500);
digitalWrite(8, HIGH);
delay(500);
digitalWrite(9, LOW);
delay(1700);
digitalWrite(9, HIGH);
delay(500);
digitalWrite(8, LOW);
delay(500);
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);

```



```
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
```

```

{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
digitalWrite(9, LOW);
delay(2);
if (digitalRead(11)==HIGH)
{m=m+1;}
digitalWrite(9, HIGH);
delay(2);
if (digitalRead(11)==LOW)
{p=p+1;}
if ((m==20) && (p==20))
{
  myGLCD.setColor(255, 255, 255);
  myGLCD.fillRoundRect (195, 335, 585, 363);
  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.print("IGBT ESTA", 297, 340);
}

```



```
geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (178, 210, 412, 260);
myGLCD.fillRoundRect (435, 210, 658, 260);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (178, 210, 412, 260);
myGLCD.drawRoundRect (435, 210, 658, 260);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SEÑAL DIGITAL", 182, 228);
myGLCD.print("SEÑAL RELE", 458, 228);
myGLCD.print("SALIR", 50, 417);

while (true)
{
  if (myTouch.dataAvailable())
  {
    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=210) && (y<=260))
    {
      if ((x>=210) && (x<412))
      {
        indicadorSeleccion(178, 210, 412, 260);
        fuentes();
      }
      if ((x>=450) && (x<=640))
      {
        indicadorSeleccion(435, 210, 658, 260);
        fuentes();
      }
    }
    if ((y>=400)&&(y<=450))
    {
      if ((x>=40)&&(x<=125))
      {
        indicadorSeleccion(40, 400, 125, 450);
        menu();
      }
    }
  }
}
```

```

    }
  }
}
}

```

Fuente propia

11.3.42 Código Fuentes

```

void fuentes()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackgroundColor(255, 255, 255); //Color tras texto
  myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
  myGLCD.setBackgroundColor(255, 255, 255);
  myGLCD.setColor(255,0,0);
  myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

  geo.fillTriangle(377,43,389,43,389,31);
  geo.fillTriangle(413,31,413,43,425,43);
  geo.fillTriangle(413,67,413,79,425,67);
  myGLCD.fillRect (389, 31, 413, 43);
  myGLCD.fillRect (413, 43, 425, 67);

  geo.fillTriangle(356,64,368,64,368,52);
  geo.fillTriangle(392,52,392,64,404,64);
  geo.fillTriangle(392,88,392,100,404,88);
  myGLCD.fillRect (368, 52, 392, 64);
  myGLCD.fillRect (392, 64, 404, 88);

  myGLCD.setColor(0, 0, 255);
  myGLCD.fillRoundRect (178, 210, 412, 260);
  myGLCD.fillRoundRect (435, 210, 658, 260);
  myGLCD.fillRoundRect (40, 400, 125, 450);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (178, 210, 412, 260);
  myGLCD.drawRoundRect (435, 210, 658, 260);
  myGLCD.drawRoundRect (40, 400, 125, 450);
  myGLCD.setBackgroundColor(0,0,255);
  myGLCD.print("FUENTE DRIVE", 198, 228);
}

```

```
myGLCD.print("FUENTE TESTER", 445, 228);
myGLCD.print("BACK", 50, 417);

while (true)
{
  if (myTouch.dataAvailable())
  {
    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    if ((y>=210) && (y<=260)) // Upper row
    {
      if ((x>=1780) && (x<412))
      {
        indicadorSeleccion(178, 210, 412, 260);
        digitalWrite(13, LOW);
        digitalWrite(12, LOW);
        digitalWriteOutput();
      }
      if ((x>=435) && (x<=658)) // Button: 1
      {
        indicadorSeleccion(435, 210, 658, 260);
        digitalWrite(12, LOW);
        digitalWrite(13, LOW);
        releOutput();
      }
    }
    if ((y>=400)&&(y<=450))
    {
      if ((x>=40)&&(x<=125))
      {
        indicadorSeleccion(40, 400, 125, 450);
        senalRele();
      }
    }
  }
}
```

Fuente propia

11.3.43 Código salida digital

```
void digitalOutput()
{
```

```

myGLCD.clrScr();
myGLCD.fillScr(255,255,255);
myGLCD.setFont(BigFont);
myGLCD.setColor(0, 0, 255);

myGLCD.setColor(0, 0, 0); //Color de fuente
myGLCD.setBackColor(255, 255, 255); //Color tras texto
myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
myGLCD.setBackColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

geo.fillTriangle(377,43,389,43,389,31);
geo.fillTriangle(413,31,413,43,425,43);
geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 255, 0);
myGLCD.fillRoundRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

while (true)

```



```
{
  if (myTouch.dataAvailable())
  {

    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();

    digitalWrite(a, a);

    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        digitalWrite(13, LOW);
        fuentes();
      }
    }
    if ((y>=260) && (y<=310))
    {
      if ((x>=310) && (x<=375))
      {
        indicadorSeleccion(310, 260, 375, 310);
        digitalWrite(13, HIGH);
      }
      if ((x>=430) && (x<=495))
      {
        indicadorSeleccion(430, 260, 495, 310);
        digitalWrite(13, LOW);
      }
    }
  }
}
```

Fuente propia

11.3.44 Código salida rele

```
void releOutput()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);
```

```

myGLCD.setColor(0, 0, 0); //Color de fuente
myGLCD.setBackgroundColor(255, 255, 255); //Color tras texto
myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); myGLCD.setBackgroundColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

geo.fillTriangle(377,43,389,43,389,31);
geo.fillTriangle(413,31,413,43,425,43);
geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 255, 0);
myGLCD.fillRoundRect (310, 260, 375, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (310, 260, 375, 310);
myGLCD.setBackgroundColor(0,255,0);
myGLCD.print("ON", 325, 279);

myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect (430, 260, 495, 310);
myGLCD.setColor(0,0,0);
myGLCD.drawRoundRect (430, 260, 495, 310);
myGLCD.setBackgroundColor(255,0,0);
myGLCD.print("OFF", 438, 279);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

while (true)
{
  if (myTouch.dataAvailable())
  {
myTouch.read();
  x=myTouch.getX();

```



```

myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120);
myGLCD.setBackgroundColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

    geo.fillTriangle(377,43,389,43,389,31);
    geo.fillTriangle(413,31,413,43,425,43);
    geo.fillTriangle(413,67,413,79,425,67);
    myGLCD.fillRect (389, 31, 413, 43);
    myGLCD.fillRect (413, 43, 425, 67);

    geo.fillTriangle(356,64,368,64,368,52);
    geo.fillTriangle(392,52,392,64,404,64);
    geo.fillTriangle(392,88,392,100,404,88);
    myGLCD.fillRect (368, 52, 392, 64);
    myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);
senalin();

while (true)
{
    if (myTouch.dataAvailable())
    {
        myTouch.read();
        x=myTouch.getX();
        y=myTouch.getY();
        int lectura5;

        lectura=analogRead(readSi);
        lectura5=(lectura*20)/4095;
        myGLCD.printNuml(lectura5, CENTER, 300);

        delay(200);

        if ((y>=400) && (y<=450)) // Upper row
        {
            if ((x>=40) && (x<=125)) // Button: 1
            {
                indicadorSeleccion(40, 400, 125, 450);
                menu();
            }
        }
    }
}

```

```

    }
  }
}
else
{
  int lectura5;
  lectura=analogRead(readSi);
  lectura5=(lectura*20)/4095;
  myGLCD.printNuml(lectura5, CENTER, 300);
  delay(200);
}
}
delay(50);
}

```

Fuente propia

11.3.46 Código Señal de drive

```

void senalin()
{
  int lectura5;
  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackColor(255, 255, 255); //Color tras texto
  myGLCD.print("SEÑAL DE DRIVE", CENTER, 250);

  lectura=analogRead(readSi);
  lectura5=(lectura*20)/4095;
  myGLCD.print("mA", 450, 300);
  myGLCD.printNuml(lectura5, CENTER, 300);
  delay(50);
}

```

Fuente propia

11.3.47 Código Escribir señal

```

void menuSenalOut()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 0); //Color de fuente
  myGLCD.setBackColor(255, 255, 255); //Color tras texto

```

```

myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
myGLCD.setBackgroundColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

    geo.fillTriangle(377,43,389,43,389,31);
    geo.fillTriangle(413,31,413,43,425,43);
    geo.fillTriangle(413,67,413,79,425,67);
    myGLCD.fillRect (389, 31, 413, 43);
    myGLCD.fillRect (413, 43, 425, 67);

    geo.fillTriangle(356,64,368,64,368,52);
    geo.fillTriangle(392,52,392,64,404,64);
    geo.fillTriangle(392,88,392,100,404,88);
    myGLCD.fillRect (368, 52, 392, 64);
    myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (123, 210, 315, 260);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (123, 210, 315, 260);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SIGNAL AUTO", 132, 227);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (464, 210, 683, 260);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (464, 210, 683, 260);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SIGNAL MANUAL", 470, 227);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

i=0;

analogWrite(DAC1, 0);

while (true)
{
    if (myTouch.dataAvailable())
    {

```



```

geo.fillTriangle(377,43,389,43,389,31);
geo.fillTriangle(413,31,413,43,425,43);
geo.fillTriangle(413,67,413,79,425,67);
myGLCD.fillRect (389, 31, 413, 43);
myGLCD.fillRect (413, 43, 425, 67);

geo.fillTriangle(356,64,368,64,368,52);
geo.fillTriangle(392,52,392,64,404,64);
geo.fillTriangle(392,88,392,100,404,88);
myGLCD.fillRect (368, 52, 392, 64);
myGLCD.fillRect (392, 64, 404, 88);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (40, 400, 125, 450);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (40, 400, 125, 450);
myGLCD.setBackColor(0,0,255);
myGLCD.print("SALIR", 50, 417);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (200, 210, 285, 260);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (200, 210, 285, 260);
myGLCD.setBackColor(0,0,255);
myGLCD.print("VOL.", 212, 227);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (520, 210, 605, 260);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (520, 210, 605, 260);
myGLCD.setBackColor(0,0,255);
myGLCD.print("AMP.", 533, 227);

i=0;

analogWrite(DAC1, 0);

  while (true)
  {
    if (myTouch.dataAvailable())
    {

      myTouch.read();
      x=myTouch.getX();
      y=myTouch.getY();
    }
  }

```



```
if ((y>=400) && (y<=450)) // Upper row
{
  if ((x>=40) && (x<=125)) // Button: 1
  {
    indicadorSeleccion(40, 400, 125, 450);
    menuSenalOut();
  }
}
if ((y>=210) && (y<=260))
{
  if ((x>=520) && (x<=605)) //VOLTAJE
  {
    indicadorSeleccion(520, 210, 605, 260);
    senalOutAutomA();
  }
  if ((x>=200) && (x<=285)) //mAMPERE
  {
    indicadorSeleccion(200, 210, 285, 260);
    senalOutAutoV();
  }
}
}
}
}
```

Fuente propia

11.3.49 Código Señal auto mA

```
void senalOutAutomA()
{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 255);
  myGLCD.fillRoundRect (40, 400, 125, 450);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (40, 400, 125, 450);
  myGLCD.setBackColor(0,0,255);
  myGLCD.print("BACK", 50, 417);

  myGLCD.setColor(0, 0, 255);
  myGLCD.fillRoundRect (200, 100, 285, 150);
  myGLCD.setColor(255, 255, 255);
```

```
myGLCD.drawRoundRect (200, 100, 285, 150);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("PLAY", 212, 120);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (520, 100, 605, 150);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (520, 100, 605, 150);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("PAUSE", 523, 120);

analogWrite(DAC1, i);

if(i<=20){
    sumamA();
}
else if(i>=4090){
    restamA();
}
}
```

Fuente propia

11.3.50 Código Suma mA

```
void sumamA()
{
    myGLCD.setColor(0, 0, 0);
    myGLCD.setBackgroundColor(255,255,255);
    for(i=i; i<4095; i=i+12){
        int lectura5;
        myGLCD.setBackgroundColor(255,255,255);
        int i2;
        analogWrite(DAC1, i);
        i2=(i*20)/4095; // Envio de datos a DAC1
        myGLCD.print("SEÑAL DE ENTRADA AL DRIVE", CENTER, 175);
        myGLCD.print("mA", 450, 200);
        myGLCD.printNuml(i2, CENTER, 200);

        myGLCD.print("SEÑAL DE SALIDA DRIVE", CENTER, 300);
        lectura=analogRead(readSi);
        lectura5=(lectura*20)/4096;
        myGLCD.print("mA", 450, 325);
        myGLCD.printNuml(lectura5, CENTER, 325);

        if(i>=4087){
```



```

lectura=analogRead(readSi);
lectura5=(lectura*20)/4095;
myGLCD.print("mA", 450, 325);
myGLCD.printNum1(lectura5, CENTER, 325);

if (i<=12)
{
  sumamA();
}
myTouch.read();
x=myTouch.getX();
y=myTouch.getY();
if ((y>=400) && (y<=450)) // Upper row
{
  if ((x>=40) && (x<=125)) // Button: 1
  {
    indicadorSeleccion(40, 400, 125, 450);
    senalOutAuto();
  }
}
if ((y>=100) && (y<=150))
{
  if ((x>=520) && (x<=605)) //PAUSE
  {
    indicadorSeleccion(520, 100, 605, 150);
    pauseRestamA();
  }
  if ((x>=200) && (x<=285)) // PLAY
  {
    indicadorSeleccion(200, 100, 285, 150);
  }
}
delay(35);
}
}

```

Fuente propia

11.3.52 Código Pause suma mA

```

void pauseSumamA()
{
  myGLCD.setColor(0, 0, 0);
  myGLCD.setBackColor(255,255,255);

  while (true){

```

```

i=i;
int i2;
int lectura5;
analogWrite(DAC1, i);
i2=(i*20)/4095; // Envio de datos a DAC1
myGLCD.printNuml(i2, CENTER, 200);

lectura=analogRead(readSi);
lectura5=(lectura*20)/4095;
myGLCD.printNuml(lectura5, CENTER, 325);

myTouch.read();
x=myTouch.getX();
y=myTouch.getY();
if ((y>=100) && (y<=150)) // Upper row
{
  if ((x>=200) && (x<=285)) // Button: 1
  {
    indicadorSeleccion(200, 100, 285, 150);
    sumamA();
  }
  if ((x>=520) && (x<=605))
  {
    indicadorSeleccion(520, 100, 605, 150);
  }
}
if ((y>=400) && (y<=450)) // Upper row
{
  if ((x>=40) && (x<=125)) // Button: 1
  {
    indicadorSeleccion(40, 400, 125, 450);
    senalOutAuto();
  }
}
}
delay(100);
}

```

Fuente propia

11.3.53 Código Pause resta mA

```

void pauseRestamA()
{
  myGLCD.setColor(0, 0, 0);

```

```
myGLCD.setBackgroundColor(255,255,255);

while (true){
  i=i;
  int lectura5;
  int i2;
  analogWrite(DAC1, i);
  i2=(i*20)/4095; // Envio de datos a DAC1
  myGLCD.printNuml(i2, CENTER, 200);

  lectura=analogRead(readSi);
  lectura5=(lectura*20)/4095;
  myGLCD.printNuml(lectura5, CENTER, 325);

  myTouch.read();
  x=myTouch.getX();
  y=myTouch.getY();
  if ((y>=100) && (y<=150)) // Upper row
  {
    if ((x>=200) && (x<=285)) // Button: 1
    {
      indicadorSeleccion(200, 100, 285, 150);
      restamA();
    }
    if ((x>=520) && (x<=605))
    {
      indicadorSeleccion(520, 100, 605, 150);
    }
  }
  if ((y>=400) && (y<=450)) // Upper row
  {
    if ((x>=40) && (x<=125)) // Button: 1
    {
      indicadorSeleccion(40, 400, 125, 450);
      senalOutAuto();
    }
  }
}
delay(100);
}
```

Fuente propia

11.3.54 Código Señal auto V

```
void senalOutAutoV()
```

```

{
  myGLCD.clrScr();
  myGLCD.fillScr(255,255,255);
  myGLCD.setFont(BigFont);
  myGLCD.setColor(0, 0, 255);

  myGLCD.setColor(0, 0, 255);
  myGLCD.fillRoundRect (40, 400, 125, 450);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (40, 400, 125, 450);
  myGLCD.setBackgroundColor(0,0,255);
  myGLCD.print("BACK", 50, 417);

  myGLCD.setColor(0, 0, 255);
  myGLCD.fillRoundRect (200, 100, 285, 150);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (200, 100, 285, 150);
  myGLCD.setBackgroundColor(0,0,255);
  myGLCD.print("PLAY", 212, 120);

  myGLCD.setColor(0, 0, 255);
  myGLCD.fillRoundRect (520, 100, 605, 150);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (520, 100, 605, 150);
  myGLCD.setBackgroundColor(0,0,255);
  myGLCD.print("PAUSE", 523, 120);

  analogWrite(DAC1, i);

  if(i<=20){
    sumaV();
  }
  else if(i>=4090){
    restaV();
  }
}
}

```

Fuente propia

11.3.55 Código Suma V

```

void sumaV()
{
  myGLCD.setColor(0, 0, 0);
  myGLCD.setBackgroundColor(255,255,255);
  for(i=i; i<4095; i=i+12){

```

```

int i2;
int lectura5;
analogWrite(DAC1, i);
i2=(i*2)+2100; // Envio de datos a DAC1
myGLCD.print("SEÑAL DE ENTRADA AL DRIVE", CENTER, 175);
myGLCD.print("mV", 450, 200);
myGLCD.printNuml(i2, CENTER, 200);

myGLCD.print("SEÑAL DE SALIDA DRIVE", CENTER, 300);
lectura=analogRead(readSi);
lectura5=(lectura*20)/4095;
myGLCD.print("mA", 450, 325);
myGLCD.printNuml(lectura5, CENTER, 325);

if(i>=4087){
  restaV();
}

myTouch.read();
x=myTouch.getX();
y=myTouch.getY();

if ((y>=400) && (y<=450)) // Upper row
{
  if ((x>=40) && (x<=125)) // Button: 1
  {
    indicadorSeleccion(40, 400, 125, 450);
    senalOutAuto();
  }
}
if ((y>=100) && (y<=150))
{
  if ((x>=520) && (x<=605)) //PAUSE
  {
    indicadorSeleccion(520, 100, 605, 150);
    pauseSumaV();
  }
  if ((x>=200) && (x<=285)) // PLAY
  {
    indicadorSeleccion(200, 100, 285, 150);
  }
}
}
delay (150);
}
}

```

Fuente propia

11.3.56 Código Resta V

```
void restaV()
{
  for(i=i; i>12; i=i-12){
    int i2;
    int lectura5;
    myGLCD.setColor(0, 0, 0);
    myGLCD.setBackColor(255,255,255);
    analogWrite(DAC1, i); // Envio de datos a DAC1
    i2=(i*2)+2100;
    myGLCD.printNuml(i2, CENTER, 200);

    lectura=analogRead(readSi);
    lectura5=(lectura*20)/4095;
    myGLCD.print("mA", 450, 325);
    myGLCD.printNuml(lectura5, CENTER, 325);

    if (i<=12)
    {
      sumaV();
    }
    myTouch.read();
    x=myTouch.getX();
    y=myTouch.getY();
    if ((y>=400) && (y<=450)) // Upper row
    {
      if ((x>=40) && (x<=125)) // Button: 1
      {
        indicadorSeleccion(40, 400, 125, 450);
        senalOutAuto();
      }
    }
    if ((y>=100) && (y<=150))
    {
      if ((x>=520) && (x<=605)) //PAUSE
      {
        indicadorSeleccion(520, 100, 605, 150);
        pauseRestaV();
      }
      if ((x>=200) && (x<=285)) // PLAY
      {
        indicadorSeleccion(200, 100, 285, 150);
      }
    }
  }
}
```

```

    delay(150);
  }
}

```

Fuente propia

11.3.57 Código Pause suma V

```

void pauseSumaV()
{

myGLCD.setColor(0, 0, 0);
myGLCD.setBackColor(255,255,255);

while (true){
i=i;
int i2;
int lectura5;
analogWrite(DAC1, i);
i2=(i*2)+2100; // Envio de datos a DAC1
myGLCD.printNuml(i2, CENTER, 200);

lectura=analogRead(readSi);
lectura5=(lectura*20)/4095;
myGLCD.printNuml(lectura5, CENTER, 325);

myTouch.read();
x=myTouch.getX();
y=myTouch.getY();
if ((y>=100) && (y<=150)) // Upper row
{
if ((x>=200) && (x<=285)) // Button: 1
{
indicadorSeleccion(200, 100, 285, 150);
sumaV();
}
if ((x>=520) && (x<=605))
{
indicadorSeleccion(520, 100, 605, 150);
}
}
if ((y>=400) && (y<=450)) // Upper row
{
if ((x>=40) && (x<=125)) // Button: 1
{
indicadorSeleccion(40, 400, 125, 450);
}
}
}
}

```

```

        senalOutAuto();
    }
}
}
}

```

Fuente propia

11.3.58 Código Pause resta V

```

void pauseRestaV()
{
    myGLCD.setColor(0, 0, 0);
    myGLCD.setBackColor(255,255,255);

    while (true){
        i=i;
        int lectura5;
        int i2;
        analogWrite(DAC1, i);
        i2=(i*2)+2100; // Envio de datos a DAC1
        myGLCD.printNuml(i2, CENTER, 200);

        lectura=analogRead(readSi);
        lectura5=(lectura*20)/4095;
        myGLCD.printNuml(lectura5, CENTER, 325);

        myTouch.read();
        x=myTouch.getX();
        y=myTouch.getY();
        if ((y>=100) && (y<=150)) // Upper row
        {
            if ((x>=200) && (x<=285)) // Button: 1
            {
                indicadorSeleccion(200, 100, 285, 150);
                restaV();
            }
            if ((x>=520) && (x<=605))
            {
                indicadorSeleccion(520, 100, 605, 150);
            }
        }
        if ((y>=400) && (y<=450)) // Upper row
        {
            if ((x>=40) && (x<=125)) // Button: 1
            {

```

```

        indicadorSeleccion(40, 400, 125, 450);
        senalOutAuto();
    }
}
}
}
}

```

Fuente propia

11.3.59 Código Señal manual

```

void senalOutManual()
{
    myGLCD.clrScr();
    myGLCD.fillScr(255,255,255);
    myGLCD.setFont(BigFont);
    myGLCD.setColor(0, 0, 255);

    myGLCD.setColor(0, 0, 0); //Color de fuente
    myGLCD.setBackColor(255, 255, 255); //Color tras texto
    myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
120); //Escribe lo que hay entre comillas en el punto x(centrado) y y
    myGLCD.setBackColor(255, 255, 255);
    myGLCD.setColor(255,0,0);
    myGLCD.print("SERVICE DEPARTMENT", CENTER, 155);

    geo.fillTriangle(377,43,389,43,389,31);
    geo.fillTriangle(413,31,413,43,425,43);
    geo.fillTriangle(413,67,413,79,425,67);
    myGLCD.fillRect (389, 31, 413, 43);
    myGLCD.fillRect (413, 43, 425, 67);

    geo.fillTriangle(356,64,368,64,368,52);
    geo.fillTriangle(392,52,392,64,404,64);
    geo.fillTriangle(392,88,392,100,404,88);
    myGLCD.fillRect (368, 52, 392, 64);
    myGLCD.fillRect (392, 64, 404, 88);

    myGLCD.setColor(0, 0, 255);
    myGLCD.fillRoundRect (40, 400, 125, 450);
    myGLCD.setColor(255, 255, 255);
    myGLCD.drawRoundRect (40, 400, 125, 450);
    myGLCD.setBackColor(0,0,255);
    myGLCD.print("BACK", 50, 417);

    myGLCD.setColor(0, 0, 255);

```

```

myGLCD.fillRoundRect (200, 210, 285, 260);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (200, 210, 285, 260);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("VOL.", 212, 227);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (520, 210, 605, 260);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (520, 210, 605, 260);
myGLCD.setBackgroundColor(0,0,255);
myGLCD.print("AMP.", 533, 227);

i=0;

analogWrite(DAC1, 0);

while (true)
{
if (myTouch.dataAvailable())
{

myTouch.read();
x=myTouch.getX();
y=myTouch.getY();

if ((y>=400) && (y<=450)) // Upper row
{
if ((x>=40) && (x<=125)) // Button: 1
{
indicadorSeleccion(40, 400, 125, 450);
menuSenalOut();
}
}
if ((y>=210) && (y<=260))
{
if ((x>=520) && (x<=605)) //VOLTAJE
{
indicadorSeleccion(520, 210, 605, 260);
senalOutManualmA();
}
if ((x>=200) && (x<=285)) //mAMPERE
{
indicadorSeleccion(200, 210, 285, 260);
senalOutManualV();
}
}
}
}

```

```
}  
}  
}
```

Fuente propia

11.3.60 Código Señal manual V

```
Void senalOutManualV()  
{  
    myGLCD.clrScr();  
    myGLCD.fillScr(255,255,255);  
    myGLCD.setFont(BigFont);  
    myGLCD.setColor(0, 0, 255);  
  
    myGLCD.setColor(0, 0, 255);  
    myGLCD.fillRoundRect (40, 400, 125, 450);  
    myGLCD.setColor(255, 255, 255);  
    myGLCD.drawRoundRect (40, 400, 125, 450);  
    myGLCD.setBackgroundColor(0,0,255);  
    myGLCD.print("SALIR", 50, 417);  
  
    myGLCD.setColor(0, 0, 255);  
    myGLCD.fillRoundRect (206, 100, 309, 150);  
    myGLCD.setColor(255, 255, 255);  
    myGLCD.drawRoundRect (206, 100, 309, 150);  
    myGLCD.setBackgroundColor(0,0,255);  
    myGLCD.print("ARRIBA >", 230, 119);  
  
    myGLCD.setColor(0, 0, 255);  
    myGLCD.fillRoundRect (206, 160, 309, 210);  
    myGLCD.setColor(255, 255, 255);  
    myGLCD.drawRoundRect (206, 160, 309, 210);  
    myGLCD.setBackgroundColor(0,0,255);  
    myGLCD.print("ABAJO <", 208, 179);  
  
    myGLCD.setColor(VGA_BLACK);  
    myGLCD.setBackgroundColor(VGA_WHITE);  
    myGLCD.print("SEÑAL DE ENTRADA AL DRIVE", 350, 75);  
    myGLCD.print("SEÑAL DE SALIDA DRIVE", 340, 280);  
  
    int lectura5;  
  
    myGLCD.setColor(255, 255, 255);  
    myGLCD.fillRoundRect (428, 300, 465, 325);
```

```
myGLCD.setColor(VGA_BLACK);
myGLCD.setBackgroundColor(VGA_WHITE);
lectura=analogRead(readSi);
lectura5=(lectura*20)/4095;
myGLCD.print("mA", 530, 305);
myGLCD.printNum1(lectura5, 430, 305);

i=0;
analogWrite(DAC1, 0);
upDownV();
}
```

Fuente propia

11.3.61 Código Señal up down V

```
void upDownV()
{
  while(true)
  {
    if (myTouch.dataAvailable())
    {

      int i1;
      int lectura5;
      myGLCD.setColor(255, 255, 255);
      myGLCD.fillRoundRect (428, 300, 465, 325);
      myGLCD.setColor(255, 255, 255);
      myGLCD.fillRoundRect (448, 92, 535, 123);
      myGLCD.setColor(VGA_BLACK);
      myGLCD.setBackgroundColor(VGA_WHITE);
      i1=(i*2)+2100; // Envio de datos a DAC1
      myGLCD.printNum1(i1, 450, 100);
      myGLCD.print("mV", 545, 100);

      lectura=analogRead(readSi);
      lectura5=(lectura*20)/4095;
      myGLCD.print("mA", 530, 305);
      myGLCD.printNum1(lectura5, 430, 305);

      analogWrite(DAC1, i);

      myTouch.read();
      x=myTouch.getX();
      y=myTouch.getY();
    }
  }
}
```

```

delay(150);

if ((y>=400) && (y<=450)) // Upper row
{
  if ((x>=40) && (x<=125)) // Button: 1
  {
    indicadorSeleccion(40, 400, 125, 450);
    senalOutManual();
  }
}

if ((x>=206) && (x<=309)) // Upper row
{
  if ((y>=100) && (y<=150)) // Button: 1
  {
    indicadorSeleccion(206, 100, 309, 150);
    i=i+205;
    if (i>4090)
    {
      i=4095;
    }
  }
  if ((y>=160) && (y<=210))
  {
    indicadorSeleccion(206, 160, 309, 210);
    i=i-205;
    if (i<190)
    {
      i=0;
    }
  }
}
}
else
{
  int lectura5;
  myGLCD.setColor(255, 255, 255);
  myGLCD.fillRoundRect (428, 300, 465, 325);
  lectura=analogRead(readSi);
  myGLCD.setColor(VGA_BLACK);
  myGLCD.setBackColor(VGA_WHITE);
  lectura5=(lectura*20)/4095;
  myGLCD.printNumI(lectura5, 430, 305);
  analogWrite(DAC1, i);
  delay(150);
}
}

```



```
}
```

Fuente propia

11.3.62 Código Señal manual mA

```
void senalOutManualmA()
{
    myGLCD.clrScr();
    myGLCD.fillScr(255,255,255);
    myGLCD.setFont(BigFont);
    myGLCD.setColor(0, 0, 255);

    myGLCD.setColor(0, 0, 255);
    myGLCD.fillRoundRect (40, 400, 125, 450);
    myGLCD.setColor(255, 255, 255);
    myGLCD.drawRoundRect (40, 400, 125, 450);
    myGLCD.setBackgroundColor(0,0,255);
    myGLCD.print("SALIR", 50, 417);

    myGLCD.setColor(0, 0, 255);
    myGLCD.fillRoundRect (206, 100, 309, 150);
    myGLCD.setColor(255, 255, 255);
    myGLCD.drawRoundRect (206, 100, 309, 150);
    myGLCD.setBackgroundColor(0,0,255);
    myGLCD.print("ARRIBA >", 230, 119);

    myGLCD.setColor(0, 0, 255);
    myGLCD.fillRoundRect (206, 160, 309, 210);
    myGLCD.setColor(255, 255, 255);
    myGLCD.drawRoundRect (206, 160, 309, 210);
    myGLCD.setBackgroundColor(0,0,255);
    myGLCD.print("ABAJO <", 208, 179);

    myGLCD.setColor(VGA_BLACK);
    myGLCD.setBackgroundColor(VGA_WHITE);
    myGLCD.print("SEÑAL DE ENTRADA AL DRIVE", 350, 75);
    myGLCD.print("SEÑAL DE SALIDA DRIVE", 340, 280);

    int lectura5;

    myGLCD.setColor(255, 255, 255);
    myGLCD.fillRoundRect (428, 300, 465, 325);

    myGLCD.setColor(VGA_BLACK);
    myGLCD.setBackgroundColor(VGA_WHITE);
```

```
lectura=analogRead(readSi);
lectura5=(lectura*20)/4095;
myGLCD.print("mA", 530, 305);
myGLCD.printNuml(lectura5, 430, 305);

i=0;
analogWrite(DAC1, 0);
upDownmA();
}
```

Fuente propia

11.3.63 Código Señal up down mA

```
void upDownmA()
{
  while(true)
  {
    if (myTouch.dataAvailable())
    {

      int i1;
      int lectura5;
      myGLCD.setColor(255, 255, 255);
      myGLCD.fillRoundRect (428, 300, 465, 325);
      myGLCD.setColor(255, 255, 255);
      myGLCD.fillRoundRect (448, 92, 535, 123);
      myGLCD.setColor(VGA_BLACK);
      myGLCD.setBackColor(VGA_WHITE);
      i1=(i*20)/4095; // Envio de datos a DAC1
      myGLCD.printNuml(i1, 450, 100);
      myGLCD.print("mA", 545, 100);

      lectura=analogRead(readSi);
      lectura5=(lectura*20)/4095;
      myGLCD.print("mA", 530, 305);
      myGLCD.printNuml(lectura5, 430, 305);

      analogWrite(DAC1, i);

      myTouch.read();
      x=myTouch.getX();
      y=myTouch.getY();

      delay(150);
    }
  }
}
```

```
if ((y>=400) && (y<=450)) // Upper row
{
  if ((x>=40) && (x<=125)) // Button: 1
  {
    indicadorSeleccion(40, 400, 125, 450);
    senalOutManual();
  }
}

if ((x>=206) && (x<=309)) // Upper row
{
  if ((y>=100) && (y<=150)) // Button: 1
  {
    indicadorSeleccion(206, 100, 309, 150);
    i=i+205;
    if (i>4090)
    {
      i=4095;
    }
  }
  if ((y>=160) && (y<=210))
  {
    indicadorSeleccion(206, 160, 309, 210);
    i=i-205;
    if (i<190)
    {
      i=0;
    }
  }
}
}
else
{
  int lectura5;
  myGLCD.setColor(255, 255, 255);
  myGLCD.fillRoundRect (428, 300, 465, 325);
  lectura=analogRead(readSi);
  myGLCD.setColor(VGA_BLACK);
  myGLCD.setBackColor(VGA_WHITE);
  lectura5=(lectura*20)/4095;
  myGLCD.printNuml(lectura5, 430, 305);
  analogWrite(DAC1, i);
  delay(150);
}
}
```

Fuente propia

11.3.64 Código Configuración de pines

```
void setup()
{
  pinMode(14, OUTPUT);
  pinMode(15, OUTPUT);
  pinMode(16, OUTPUT);
  pinMode(11, INPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(A0, INPUT);
  Serial.begin(9600); // Configura el Puerto serie.
  analogReadResolution(12); //Resolucion del ADC de 12 bits
  analogWriteResolution(12); // Configura resolución DAC de 12 bits

  myGLCD.InitLCD(); //Inicializa la LCD
  myGLCD.setFont(BigFont); //Define el tamaño de la fuente
  myTouch.InitTouch(); //Inicializa el Touch
  myTouch.setPrecision(PREC_HI); //Presicion de touch

  // Clear the screen and draw the frame
  myGLCD.clrScr();
}
```

Fuente propia

11.3.65 Código Bienvenida

```
Void loop()
{
  int buf[798];
  int x, x2;
  int y, y2;
  int r;

  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);
  digitalWrite(10, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);
```

```

myGLCD.setColor(255, 255, 255); //Color de fondo de rectangulo
descrito a continuación
myGLCD.fillRect(0, 0, 799, 479); //Dibuja rectangulo rellene entre 2
puntos

myGLCD.setColor(0, 0, 0); //Color de fuente
myGLCD.setBackColor(255, 255, 255); //Color tras texto
myGLCD.print("COL CONTROLES & INTEGRACION S.A.", CENTER,
250);
myGLCD.setBackColor(255, 255, 255);
myGLCD.setColor(255,0,0);
myGLCD.print("SERVICE DEPARTMENT", CENTER, 275);

geo.fillTriangle(372,135,387,135,387,120);
geo.fillTriangle(417,120,417,135,432,134);
geo.fillTriangle(417,165,417,180,432,165);
myGLCD.fillRect (387, 120, 417, 135);
myGLCD.fillRect (417, 135, 432, 165);

geo.fillTriangle(347,162,362,162,362,147);
geo.fillTriangle(392,147,392,162,407,162);
geo.fillTriangle(392,192,392,207,407,192);
myGLCD.fillRect (362, 147, 392, 162);
myGLCD.fillRect (392, 162, 407, 192);

myGLCD.setColor(0, 0, 255);
myGLCD.fillRoundRect (308, 340, 488, 390);
myGLCD.setColor(255, 255, 255);
myGLCD.drawRoundRect (308, 340, 488, 390);
myGLCD.setBackColor(0,0,255);
myGLCD.print("BIENVENIDO", 340, 357);

while (true)
{
if (myTouch.dataAvailable())
{
myTouch.read();
x=myTouch.getX();
y=myTouch.getY();

if ((y>=340) && (y<=390)) // Upper row
{
if ((x>=308) && (x<=488)) // Button: 1
{
indicadorSeleccion(308, 340, 488, 390);
menu();
}
}
}
}

```

```
    }  
  }  
  delay(100);  
}
```

Fuente propia