



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
REDES INDUSTRIALES**

**“EVALUACIÓN DE LA EFICIENCIA DE LOS
CONTROLADORES ARDUINO MEGA Y SIEMENS LOGO
230RC EN PROCESOS INDUSTRIALES”.**

TESIS DE GRADO

Previa a la obtención del título de:

INGENIERO EN ELECTRÓNICA CONTROL Y REDES INDUSTRIALES

**PRESENTADO POR:
JOSÉ LUIS TINAJERO LEÓN**

RIOBAMBA – ECUADOR

2014

AGRADECIMIENTO

A Dios por ser el eje sobre el cual gira mi vida, por brindarme cada día la oportunidad de ser mejor, y poder alcanzar todos los objetivos planteados.

A mis padres y hermana por la confianza, fuerza y entrega que me han brindado, y el apoyo incondicional durante toda mi existencia.

A mi novia que ha estado junto a mí en todo momento, y muchas gracias por sus palabras de aliento ya que con ellas he podido superar lo inconvenientes que se presentaron.

A mi tutor y docentes que aportaron sus conocimientos durante toda mi vida estudiantil.

Y a todas las personas que me rodean y que día adía me enseñan algo diferente, muchas gracias por su apoyo y conocimientos. Que aunque no las mencione han formado parte de este gran logro.

DEDICATORIA

Este trabajo está dedicado a mis padres, novia, familia y amigos.

A mi madre por ser un claro ejemplo de lucha, perseverancia y superación.

A mi padre por haber sido el mejor hombre que pude haber conocido, humilde, sencillo y muy capaz.

A mi novia por estar a mi lado y ayudarme en todo momento de manera incondicional.

A mi primo Juan Pablo por ser una gran persona y que siempre has estado a mi lado y ahora más que nunca.

José Luis

FIRMAS RESPONSABLES Y NOTA

NOMBRE	FIRMA	FECHA
Ing. Gonzalo Samaniego DECANO DE LA FACULTAD DE INFORMATICA Y ELECTRÓNICA	_____	_____
Ing. Alberto Arellano DIRECTOR DE LA ESCUELA ING. ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES	_____	_____
Ing. Jorge Luis Paucar DIRECTOR DE TESIS	_____	_____
Ing. Janeth Arias MIEMBRO DEL TRIBUNAL	_____	_____
DIRECTOR CENTRO DE DOCUMENTACIÓN	_____	_____

NOTA DE LA TESIS: _____

~ v ~

RESPONSABILIDAD DEL AUTOR

Yo José Luis Tinajero León, soy el único autor del presente trabajo de tesis titulado: “EVALUACIÓN DE LA EFICIENCIA DE LOS CONTROLADORES ARDUINO MEGA Y SIEMENS LOGO 230RC EN PROCESOS INDUSTRIALES”, siendo el responsable de las ideas, fundamentos y resultados expuestos en el presente trabajo, y el patrimonio intelectual de la misma pertenecen a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

José Luis Tinajero León

ÍNDICE DE ABREVIATURAS

SISO: Simple In, Simple Out (Entrada Simple, Salida Simple)

SIMO: Simple In, MultipleOut (Entrada Simple, Salida Múltiple)

MISO: Multiple In, Simple Out (Entradas Múltiples, Salida Simple)

MIMO: Multiple In, MultipleOut(Entradas Múltiples, Salidas Múltiples)

PD: Control Proporcional Derivativo

PI: Control Proporcional Integral

PID: Control Proporcional Integral Derivativo

NA: Contacto Normalmente Abierto

NC: Contacto Normalmente Cerrado

Termopar: termocupla

mV/ V: milivoltios / voltios

mW: miliWatts

mA: miliAmperios

°C: grados centígrados

PLC: Controlador Lógico Programable

PC: computador personal

Vcc/VDC: Voltaje en corriente continua

Vca: Voltaje en corriente alterna

V AC/DC: Voltaje de corriente alterna o continua

ms: milisegundos

UARTS: Transmisor- receptor Asíncrono Universal

ISR: interruptServiceRoutine (rutina de interrupción de servicio)

GND: Tierra

LabView: Laboratory Virtual Instrument Engineering Workbench

Vis: Instrumento Virtual

Led: Diodo emisor de luz

ASCII: Código estándar Estadounidense para el intercambio de información

cm: centímetros

HMI: Interfaz Humano Máquina

uV/°C: microVoltio por cada grado centígrado

FUP: Diagrama de Funciones

KOP: Esquema de Contactos

USB: Universal Serial Bus

NI: National Instruments

I/O pin: pin entrada o salida

ÍNDICE GENERAL

PORTADA	i
AGRADECIMIENTO	ii
DEDICATORIA	iii
FIRMAS RESPONSABLES Y NOTA	iv
RESPONSABILIDAD DEL AUTOR.....	v
ÍNDICE DE ABREVIATURAS	vi
ÍNDICE GENERAL	viii
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS.....	xv
ÍNDICE DE ANEXOS.....	xvi
INTRODUCCIÓN.....	xvii
CAPÍTULO I:	
MARCO REFERENCIAL	19
1.1. ANTECEDENTES.....	20
1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS	22
1.3. OBJETIVOS	24
1.3.1. OBJETIVO GENERAL.....	24
1.3.2. OBJETIVOS ESPECÍFICOS.....	24
1.4. HIPÓTESIS	25
CAPITULO II	
MARCO TEORICO	26
2. CONCEPTOS.....	27
2.1. PROCESO.....	27
2.1.1. PROCESO INDUSTRIAL.....	27
2.2. SISTEMAS DE CONTROL	28
2.2.1. INTRODUCCION	28
2.2.2. SISTEMAS DE CONTROL LAZO ABIERTO.....	29
2.2.3. SISTEMAS DE CONTROL LAZO CERRADO	29
2.2.4. DISEÑO DE SISTEMAS DE CONTROL.....	30
2.2.5. VARIABLES.....	31

2.2.6.	SISTEMAS DE CONTROL.....	32	
2.3.	APARATOS DE MANDO, REGULACIÓN Y CONTROL.....	35	
2.3.1.	SISTEMA DE MANDO.....	35	
2.3.2.	SISTEMAS DE SEÑALIZACIÓN	37	
2.3.3.	INTERRUPTORES DE CONTROL	38	
2.3.4.	ELEMENTOS DE PROTECCION.....	40	
2.3.5.	SENSORES DE TEMPERATURA: BIMETALES	43	
2.3.6.	AD595	46	
2.3.7.	DEFINICION DE PLC	47	
2.3.8.	DEFINICIÓN DE ARDUINO	52	
2.3.9.	SOFTWARE	61	
CAPITULO III			
DISEÑO E IMPLEMENTACIÓN DEL HARDWARE			72
3.1.	DISEÑO DEL TABLERO DE CONTROL DE UNA MAQUINA AGITADORA MEDIANTE SIEMENS LOGO!	73	
3.2.	DISEÑO DEL TABLERO DE CONTROL DE UNA MAQUINA AGITADORA MEDIANTE ARDUINO MEGA 2560	78	
3.3.	DISEÑO DE UN CONTROL DE TEMPERATURA PARA LA COCCION DE BALDOSAS MEDIANTE ARDUINO Y MONITOREADO POR MEDIO DE UN HMI.....	88	
CAPITULO IV			
DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE			94
4.1.	LOGO! SOFT COMFORT	95	
4.1.1.	ELEMENTOS EMPLEADOS PARA LA PROGRAMACION KOP	96	
4.2.	ARDUINO	98	
4.3.	LABVIEW-ARDUINO.....	102	
4.3.1.	PANEL FRONTAL.....	103	
4.3.2.	DIAGRAMA DE BLOQUES.....	103	
4.4.	CONEXIÓN ARDUINO- LABVIEW	104	
4.4.1.	INTERFAZ PARA ARDUINO. (INTERFACE FOR ARDUINO TOOLKIT)....	105	
4.4.2.	BLOQUES PRINCIPALES DE CONEXIÓN PALETA DE CONEXIÓN LABVIEW	110	
4.4.3.	CONFIGURACIÓN NI MAX LABVIEW	113	

4.4.4.	CREACION BASE DE DATOS.....	115
4.4.5.	ESTRUCTURAS EMPLEADAS.....	117
4.4.6.	PASOS QUE SE SIGUIERON PARA CREAR EL HMI.....	118
CAPITULO V		
PRUEBAS Y RESULTADOS 122		
5.1.	TABLEROS DE CONTROL DE LA MAQUINA AGITADORA.....	123
5.1.1.	Cuadro de resultados Tablero de control Arduino	127
5.1.2.	Análisis de resultados Tablero de Control Arduino.....	128
5.2.	CONTROL DE TEMPERATURA PARA LA COCCION DE BALDOSAS MEDIANTE ARDUINO.....	128
5.2.1.	Cuadro de Resultados Tablero de Temperatura Arduino.....	128
5.2.2.	Análisis de resultados Tablero de Temperatura Arduino	130
CONCLUSIONES		
RECOMENDACIONES		
RESUMEN		
ABSTRACT		
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE FIGURAS

<i>Fig. II. 1 Representación Sistema de Control</i>	28
<i>Fig. II. 2 Sistema De Control Lazo Abierto</i>	29
<i>Fig. II. 3 Sistema De Control Lazo Cerrado</i>	29
<i>Fig. II. 4 Sistema De Control Lazo Cerrado Multivariable</i>	30
<i>Fig. II. 5 Diseño Del Sistema de Control</i>	31
<i>Fig. II. 6 Sistemas De Mando Permanente</i>	35
<i>Fig. II. 7 Sistema Mando Instantáneo</i>	36
<i>Fig. II. 8 Pulsador De Emergencia</i>	36
<i>Fig. II. 9 Finales De Carrera</i>	37
<i>Fig. II. 10 Indicadores Luminosos</i>	38
<i>Fig. II. 11 Relé Térmico</i>	41
<i>Fig. II. 12 Relé Termomagnético</i>	42
<i>Fig. II. 13 Termocupla</i>	43
<i>Fig. II. 14 Termopar Tipo K</i>	44
<i>Fig. II. 15 Curvas Características Termopares</i>	45
<i>Fig. II. 16 AD595</i>	46
<i>Fig. II. 17 Diagrama De Bloque Funcional AD595</i>	47
<i>Fig. II. 18 SIEMENS LOGO 230RC 0BA6</i>	50
<i>Fig. II. 19 Perspectivas Del LOGO! 230 RC</i>	50
<i>Fig. II. 20 Módulo De Ampliación</i>	51
<i>Fig. II. 21 Estructura Programa Arduino</i>	53
<i>Fig. II. 22 Arduino Mega 2560</i>	56
<i>Fig. II. 23 Configuración Pull Down</i>	60
<i>Fig. II. 24 Configuración Pull Up</i>	60
<i>Fig. II. 25 Software LOGO! SOFT</i>	62
<i>Fig. II. 26 Pantalla Principal Arduino</i>	63
<i>Fig. II. 27 Panel Frontal</i>	64
<i>Fig. II. 28 Barra De Herramientas Panel Frontal</i>	66
<i>Fig. II. 29 Diagrama De Bloques</i>	66
<i>Fig. II. 30 Barra De Herramientas Diagrama De Bloques</i>	68

<i>Fig. II. 31 Sonómetro.....</i>	<i>69</i>
<i>Fig. II. 32 Vibrómetro.....</i>	<i>69</i>
<i>Fig. III. 1 Relé Termomagnético LOGO! 230 RC.....</i>	<i>73</i>
<i>Fig. III. 2 SIEMENS LOGO! 230 RC.....</i>	<i>74</i>
<i>Fig. III. 3 Instalación Switch.....</i>	<i>75</i>
<i>Fig. III. 4 Actuadores Del LOGO!.....</i>	<i>75</i>
<i>Fig. III. 5 Cable Multifilar.....</i>	<i>75</i>
<i>Fig. III. 6 Borneras De Conexión LOGO! 230RC.....</i>	<i>76</i>
<i>Fig. III. 7 Canaleta Montaje LOGO! 230 RC.....</i>	<i>76</i>
<i>Fig. III. 8 Riel DIN LOGO! 230 RC.....</i>	<i>77</i>
<i>Fig. III. 9 Esquema De Conexión LOGO! 230 RC.....</i>	<i>77</i>
<i>Fig. III. 10 Tablero LOGO! 230 RC.....</i>	<i>78</i>
<i>Fig. III. 11 Fuente De Poder Arduino.....</i>	<i>79</i>
<i>Fig. III. 12 Relé Termomagnético Arduino.....</i>	<i>80</i>
<i>Fig. III. 13 Arduino Mega Para El Agitador.....</i>	<i>81</i>
<i>Fig. III. 14 Simulación Tablero De Control Arduino.....</i>	<i>82</i>
<i>Fig. III. 15 Diseño De Placa En Ares.....</i>	<i>82</i>
<i>Fig. III. 16 Placa De Potencia Terminada.....</i>	<i>83</i>
<i>Fig. III. 17 Instalación Accionadores Tablero Arduino.....</i>	<i>83</i>
<i>Fig. III. 18 Actuadores Tablero Arduino.....</i>	<i>84</i>
<i>Fig. III. 19 Cable UTP Multifilar.....</i>	<i>84</i>
<i>Fig. III. 20 Bornera Tablero Arduino.....</i>	<i>85</i>
<i>Fig. III. 21 Canaleta Tablero Arduino.....</i>	<i>85</i>
<i>Fig. III. 22 Riel DIN Tablero Arduino.....</i>	<i>86</i>
<i>Fig. III. 23 Caja Protectora Arduino.....</i>	<i>87</i>
<i>Fig. III. 24 Tablero Arduino Terminado.....</i>	<i>87</i>
<i>Fig. III. 25 AD595 Termocupla K.....</i>	<i>88</i>
<i>Fig. III. 26 Relé Etapa de Potencia.....</i>	<i>89</i>
<i>Fig. III. 27 Termopar Tipo K.....</i>	<i>89</i>

<i>Fig. III. 28 LEDs Sistema de Control De Temperatura</i>	89
<i>Fig. III. 29 Borneras Sistema de Control De Temperatura</i>	90
<i>Fig. III. 30 Arduino Mega Control de Temperatura</i>	91
<i>Fig. III. 31 Simulación Control de Temperatura</i>	92
<i>Fig. III. 32 Diseño de Placa en Ares</i>	92
<i>Fig. III. 33 Etapa de Potencia Control de Temperatura</i>	93
<i>Fig. III. 34 Controlador Arduino Control de Temperatura</i>	93
<i>Fig. IV. 1 Panel de Programación FUP</i>	95
<i>Fig. IV. 2 Panel de Programación UDF</i>	95
<i>Fig. IV. 3 Panel de Programación KOP</i>	96
<i>Fig. IV. 4 Contacto NA. LOGO! SOFT COMFORT</i>	96
<i>Fig. IV. 5 Contacto NC LOGO! SOFT COMFORT</i>	97
<i>Fig. IV. 6 Temporizador LOGO! SOFT COMFORT</i>	97
<i>Fig. IV. 7 Pantalla Configuración Temporizador</i>	97
<i>Fig. IV. 8 Memorias LOGO! SOFT COMFORT</i>	98
<i>Fig. IV. 9 Bobina LOGO! SOFT COMFORT</i>	98
<i>Fig. IV. 10 Void Setup IDE Arduino</i>	100
<i>Fig. IV. 11 Void Loop IDE Arduino</i>	101
<i>Fig. IV. 12 Interrupción IDE Arduino</i>	101
<i>Fig. IV. 13 Panel Frontal Arduino-LabView</i>	103
<i>Fig. IV. 14 Diagrama de Bloques Arduino-LabView</i>	104
<i>Fig. IV. 15 Pantalla Principal Package Manager</i>	107
<i>Fig. IV. 16 ToolKit Para LabView</i>	107
<i>Fig. IV. 17 Instalación ToolKit Para LabView</i>	108
<i>Fig. IV. 18 Sketch para Conexión Arduino-Labview</i>	109
<i>Fig. IV. 19 Librería Arduino en LabView</i>	109
<i>Fig. IV. 20 Bloque INIT</i>	110
<i>Fig. IV. 21 Bloque CLOSE</i>	111
<i>Fig. IV. 22 Bloque SET DIGITAL PIN MODE</i>	111

<i>Fig. IV. 23 Bloque ANALOG READ PIN</i>	112
<i>Fig. IV. 24 Bloque DIGITAL WRITE PIN</i>	112
<i>Fig. IV. 25 Administrador de Dispositivos Arduino</i>	113
<i>Fig. IV. 26D Dispositivos E Interfaces NI MAX</i>	113
<i>Fig. IV. 27 Configuración Arduino NI MAX</i>	114
<i>Fig. IV. 28 Configuración de Puerto Arduino NI MAX</i>	114
<i>Fig. IV. 29 Datos de fecha Y Hora LabView</i>	115
<i>Fig. IV. 30 Transformar Dato a Cadena LabView</i>	115
<i>Fig. IV. 31 Construir Matriz en LabView</i>	116
<i>Fig. IV. 32 Hoja de Datos en LabView</i>	117
<i>Fig. IV. 33 Paleta de Estructuras de Programación en LabView</i>	117
<i>Fig. IV. 34 Control de la Estructura While</i>	118
<i>Fig. IV. 35 Control Booleano de la Estructura While</i>	118
<i>Fig. IV. 36 Diagrama de Bloques Estructura While</i>	118
<i>Fig. IV. 37 Inicio de Comunicación Arduino-LabView</i>	119
<i>Fig. IV. 38 Declaración de Pines Arduino</i>	119
<i>Fig. IV. 39 Estructura del Programa de Arduino</i>	120
<i>Fig. IV. 40 Lectura Pin Analógico</i>	120
<i>Fig. IV. 41 Representación del valor Análogo Gráficamente</i>	120
<i>Fig. IV. 42 Control de Salidas Digitales</i>	121
<i>Fig. IV. 43 Cierre de Comunicación Arduino-LabView</i>	121
<i>Fig. V. 1 Test Student</i>	130
<i>Fig. V. 2 Test Wilcoxon</i>	131
<i>Fig. V. 3 Índice de Spearman</i>	131
<i>Fig. V. 4 Test De Varianzas</i>	132

ÍNDICE DE ANEXOS

ANEXO 1. PROGRAMACIÓN ARDUINO MEGA

ANEXO 2. FOTOS DE LAS PRUEBAS REALIZADAS

INTRODUCCIÓN

El presente trabajo de investigación titulado como “EVALUACIÓN DE LA EFICIENCIA DE LOS CONTROLADORES ARDUINO MEGA Y SIEMENS LOGO 230RC EN PROCESOS INDUSTRIALES”, tiene como finalidad demostrar la capacidad y funcionalidad que posee la tarjeta Arduino Mega para ser empleada en procesos industriales.

El presente trabajo pretende demostrar que Arduino Mega es capaz de realizar las acciones y actividades que realizaría un PLC Siemens LOGO! 230RC, en procesos industriales, ya que hasta el momento dicha tarjeta solo ha sido empleada en actividades académicas, y no ha existido ningún estudio para comprobar el funcionamiento de dicha tarjeta en el ámbito industrial, donde priman factores que podrían alterar el correcto desempeño y funcionamiento de este dispositivo.

Para realizar esta investigación se ha tenido en cuenta los factores que influyen en los procesos industriales, los mismos que son: ruido, vibraciones, interferencias electromagnéticas, golpes, polvo y temperatura.

De manera adicional se pretende crear una alternativa para sustituir un PLC, debido al costo de inversión que conlleva la implementación de un tablero de control con dicho dispositivo, la implementación con el dispositivo en estudio conllevaría un costo de inversión menor y se podría obtener los mismos resultados que brinda un PLC.

Para lo cual es necesario conocer la forma de programación de cada tecnología, y las características que posee cada uno de los dispositivos tales como número de entradas y salidas digitales, número de entradas analógicas, valores de los voltajes de alimentación, grados de protección IP que poseen. Para lograr los objetivos planteados en este trabajo de investigación se realizarán varias pruebas, donde se podrá comprobar el correcto funcionamiento de cada uno de los tableros, de cada prueba se obtendrán datos de su funcionamiento.

Para obtener los datos de las pruebas, los tableros serán expuestos a ambientes de trabajo industrial, ubicándolos en zonas donde existe presencia de ruido, interferencias electromagnéticas, vibraciones, polvo y manipulación.

El presente trabajo de investigación está conformado por 5 capítulos los cuales se los detalla a continuación:

El capítulo I está estructurado por el Marco Referencial en donde se destaca los antecedentes y la respectiva justificación del trabajo, además de estos temas en este capítulo se encuentra definidos los objetivos que se pretende alcanzar con la investigación y el planteamiento de la hipótesis.

El capítulo II está conformado por el Marco Teórico, en donde se puede encontrar toda la información necesaria para tener una idea clara de las tecnologías que se utilizaran en el desarrollo del presente trabajo, además de términos y elementos que serán empleados para la implementación del mismo.

El capítulo III de título: “Diseño e Implementación de Hardware”, se destaca cada uno de los elementos y pasos que se desarrollaron para la elaboración de los tableros de control para cada una de los propósitos planteados. En este capítulo se detallan los elementos empleados indicando las características de cada uno y la función y funcionalidad que tiene dentro del tablero.

En el capítulo IV denominado “Diseño e Implementación de Software” se explica de manera detalla y legible los programas que se emplearon para el desarrollo del tema de investigación, haciendo énfasis en cada uno de los elementos y palabras reservadas que se usaron para realizar los respectivas programas.

El capítulo V se detalla las “Pruebas y Resultados”, este capítulo se analizan los datos obtenidos y se muestra el respectivo análisis realizado para verificar la hipótesis planteada.

~ 19 ~

CAPÍTULO I:

MARCO REFERENCIAL

1.1. ANTECEDENTES

Los PLC fueron creados debido a muchas necesidades que surgieron en el afán de automatizar la industria automotriz, antes de que se crearan los PLC la etapa de control se la realizaba mediante relés, controladores y contadores que se usaban para este fin específico, por lo que se les consideraba como sistemas electrónicos cableados.

Los controladores lógicos programables desde su creación hasta el día de hoy se los emplea en muchos procesos industriales que puede tener una o varias necesidades a la vez de esta manera dando el grado de complejidad del proceso. Las aplicaciones más generales de un Autómata Programable son:

- Lugar de aplicación con espacio muy reducido.
- En procesos donde existen cambios en su línea de producción de manera continuo o periódica.
- Procesos que siguen una secuencia y orden lógico.
- Cuando un proceso es muy complejo y se lo puede ampliar de manera continua.
- Se necesita monitorear o controlar la ejecución del proceso desde un mismo lugar.

De manera general cualquier tipo de proceso industrial que se desea automatizar se lo puede realizar con un PLC por lo cual las investigaciones y estudios realizados sobre éstos dispositivos son muy amplias.

Arduino fue concebido en un instituto italiano, en el cual se buscaba tener un dispositivo que se lo pudiera utilizar en las aulas de clases, que no tuviera un costo elevado, que fuese compatible con todos los sistemas operativos y que sea muy sencillo de uso para personas que incluso poseen muy poco conocimiento sobre este dispositivo. La tarjeta de hardware libre se lo puede utilizar en cualquier tipo de aplicación donde el valor monetario y la complejidad del proyecto juegan un papel importante. Las aplicaciones más comunes de este hardware son varias pero dentro de estas destacan:

- Sistemas automatizados de encendido y apagado de luces y calefacción en hogares. (ARDUINO)
- Dispositivos cuya función sirve para mejorar la vida de personas con algún tipo de discapacidad.
- Llegando a un alto grado de complejidad de proyectos desarrollados por investigadores y especialistas en distintas universidades.

Arduino es un dispositivo que ofrece muchas prestaciones y ejemplos que facilitan su desarrollo. Por lo cual su campo de aplicación ha aumentado de manera considerable dando solución a muchos problemas de manera económica y eficiente.

1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS

El presente estudio comparativo se lo realiza para conocer los beneficios, riesgos, costos-beneficios del uso de un PLC y una tarjeta de hardware y software libre denominada Arduino. Poniendo en manifiesto las deficiencias como las fortalezas de cada uno de los elementos en estudio.

Por medio de diferentes pruebas entre los dos dispositivos obtener datos como tiempo de respuesta, tiempos de ejecución, funciones especiales, etc. Verificar las ventajas y desventajas que pueden presentar cada uno de los elementos al realizar la ejecución de la actividad dentro del mismo lugar de trabajo.

PLC es un controlador lógico programable, es un dispositivo que se lo aplica en automatización industrial, un autómata programable tiene muchas aplicaciones dentro del campo industrial, éste está diseñado con varias entradas y salidas, un rango de temperatura extenso para soportar el trabajo, presenta un alto grado de inmunidad al ruido eléctrico y soporta vibraciones y son resistentes a los golpes. Siemens LOGO! 230RC posee dos terminales de alimentación de AC DC 115v~230V, tiene 8 entradas Digitales/Analógicas que se las puede ampliar hasta 24, las mismas que se pueden alimentar con DC 12V, DC 24V, AC 24V, ACDC 115~230V. Tiene 4 salidas digitales ampliables hasta 16 que pueden soportar relés de hasta 10 A AC 230V o 0,3 A DC 24V. Display LCD que ayuda a editar el programa, mostrar estados de las entradas, memorias, mostrar mensajes, etc. Además posee una interfaz para módulos de ampliación.

La tarjeta de hardware libre Arduino consiste en una placa que tiene un microcontrolador que permite el desarrollo de proyectos de electrónica, la tarjeta Arduino Mega posee un microcontrolador ATmega1280, el voltaje de funcionamiento es de 5V, posee 54 pines de entradas/salidas digitales de las cuales 14 de estas generan una salida PWM, 16 pines de entradas analógicas, una intensidad de 40mA por cada pin, con un reloj de 16MHz.

(Arduino Home, 2014)

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Evaluar la eficiencia de los controladores Arduino Mega y SIEMENS LOGO! 230RC en procesos industriales.

1.3.2. OBJETIVOS ESPECÍFICOS

- Fundamentar científicamente las características técnicas de los controladores Arduino Mega y Siemens LOGO 230 RC.
- Diseñar una Interfaz de Usuario Maquina (HMI) en LabView para monitorear y almacenar las variables de los controladores usando el protocolo de comunicación de cada dispositivo.
- Reemplazar el sistema actual de control por el controlador de mayor eficiencia seleccionado para la preparación de pasta para la elaboración de baldosas.
- Implementar los controladores en el proceso de cocción de la baldosa.
- Realizar una comparación de los datos obtenidos de los elementos en estudio (temperatura, activación de motores, tiempo de respuestas, adquisición de datos).

1.4. HIPÓTESIS

No existe diferencia significativa de la eficiencia entre los parámetros de la tarjeta Arduino Mega y Siemens Logo 230RC.

~ 26 ~

CAPITULO II

MARCO TEORICO

2. CONCEPTOS

2.1. PROCESO.

Conjunto de fases sucesivas de un fenómeno natural o de una operación artificial. (Enciclopedia Salvat Diccionario)

2.1.1. PROCESO INDUSTRIAL.

Se entiende por proceso a toda aplicación que se desarrolla de manera sistemática porque tiene una cantidad determinada de pasos ordenados y que poseen una secuencia simultanea o alternada entre sí, para llegar a un resultado deseado y preciso de acuerdo a las necesidades.

Se define como proceso industrial porque el campo de aplicación está destinado al área industrial, esto quiere decir que las instalaciones donde se desarrolla el proceso están destinadas a obtener, modificar o producir productos.

Y se los puede considerar de acuerdo al número de entradas y salidas que posee el sistema.

➤ PROCESO INDUSTRIAL SIMPLE

Se puede considerar un proceso industrial simple aquel cuyas entradas o salidas son pocas o únicas, en este caso se podría señalar los siguientes casos:

- SISO (Simple In, Simple Out): Una entrada, una salida.
- SIMO (Simple In Multiple Out): Una entrada, múltiples salidas.
- MISO (Multiple In Simple Out): Múltiples entradas una salida

➤ PROCESO INDUSTRIAL COMPLEJO

A un proceso industrial complejo se considera aquel que tiene muchas entradas y salidas a la vez, el mismo que conlleva una gran cantidad de control para que el proceso se lo pueda llevar a cabo de manera correcta.

- MIMO (Multiple In MultipleOut): Múltiples entradas múltiples salidas.

2.2. SISTEMAS DE CONTROL

2.2.1. INTRODUCCION

La ingeniería de control se basa en todos los fundamentos sobre la teoría de la retroalimentación y el análisis de sistemas lineales.

Un sistema de control es una interconexión de componentes que forman una configuración del sistema que proporcionará una respuesta deseada.



Fig. II. 1 Representación Sistema de Control

De esta manera se representa en un diagrama de bloques.

La relación entre la entrada y salida del sistema es el producto del proceso, en el cual la señal que es ingresada al sistema es procesada para generar la salida deseada o estimada, de manera general en todos los sistemas de control la señal de salida es amplificada, para ser utilizada, analizada o archivada.

2.2.2. SISTEMAS DE CONTROL LAZO ABIERTO

“Un sistema de control en lazo abierto utiliza un dispositivo de actuación para controlar el proceso directamente sin emplear retroalimentación”. (DORF Richard, 2005)

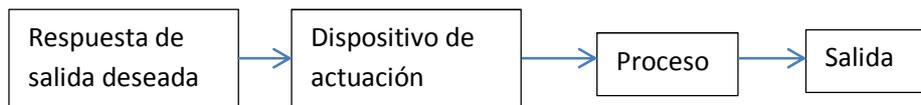


Fig. II. 2 Sistema De Control Lazo Abierto
Fuente: DORF. SISTEMAS DE CONTROL

Un sistema de control de lazo abierto es aquel donde la salida no es monitoreada, simplemente se ejecuta el proceso y la respuesta o salida se obtiene de acuerdo al funcionamiento del sistema y las condiciones que rodean a este proceso.

2.2.3. SISTEMAS DE CONTROL LAZO CERRADO

“Un sistema de control en lazo cerrado usa una medida de la salida y la retroalimentación de esta señal para compararla con la salida deseada (referencia u orden)”. (DORF Richard, 2005)

El grafico representa un sistema de control con una variable.

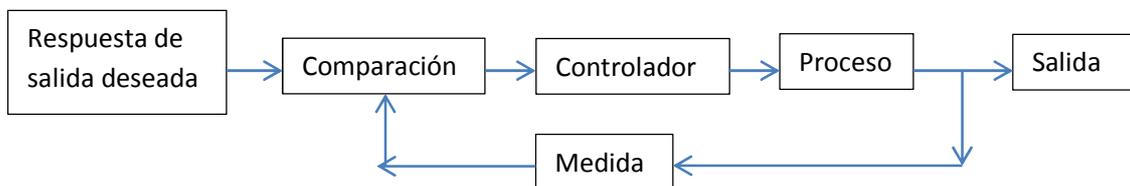


Fig. II. 3 Sistema De Control Lazo Cerrado
Fuente: DORF. SISTEMAS DE CONTROL

Un esquema de control multivariable se lo muestra a continuación.

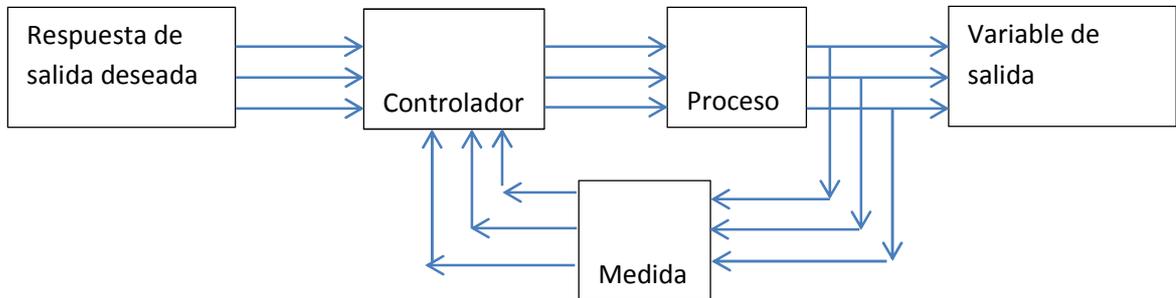


Fig. II. 4 Sistema De Control Lazo Cerrado Multivariable

Fuente: DORF. SISTEMAS DE CONTROL

En un sistema de control de lazo cerrado permite controlar la salida y obtener los resultados deseados, mediante una comparación entre la entrada y la medición que se obtiene en la retroalimentación del sistema, el margen de error que produce es el que ingresa al proceso, de acuerdo al valor se toma una acción y se logra los resultados deseados.

2.2.4. DISEÑO DE SISTEMAS DE CONTROL

El objetivo principal del diseño en la ingeniería de control es conseguir de manera eficiente y adecuada la configuración, especificaciones e identificación de los principales parámetros dentro de un sistema de control para satisfacer la o las necesidades reales que puede presentar el sistema.

Para lograr estas metas se debe plantear el diseño de la siguiente manera:

- Establecer los objetivos del sistema, lo que se desea solucionar o las metas a las cuales se desea llegar. Por ejemplo control de la temperatura de un horno, control de la velocidad de un motor, etc.
- Se debe identificar las variables que se desean controlar. Por ejemplo la temperatura de un horno, la velocidad del motor.

- Se debe escribir las especificaciones en función de la precisión que desea alcanzar. Se debe identificar un sensor que nos permita medir la variable controlada.
- Se debe elegir un actuador que cumpla todos los requisitos y requerimientos del proceso, para que cumpla de manera eficiente y efectiva con la tarea asignada a dicho dispositivo.
- Se debe seleccionar un controlador, el mismo que tiene como función comparar la respuesta deseada con la respuesta real, para que luego de esta comparación sea transferida a un amplificador.
- Finalmente, se debe ajustar los parámetros que el sistema posee para lograr un comportamiento adecuado, y que sea el deseado.

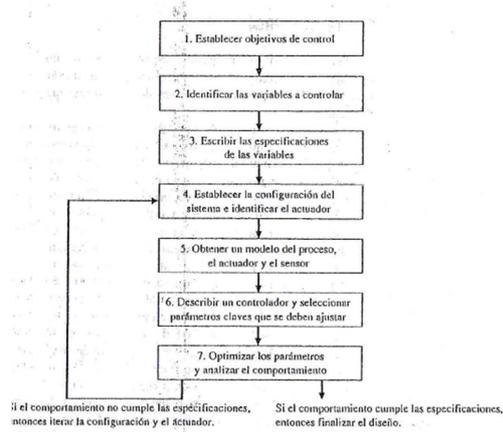


Fig. II. 5 Diseño Del Sistema de Control

Fuente: DORF. SISTEMAS DE CONTROL MODERNO

2.2.5. VARIABLES

Son todos los fenómenos que se los puede medir y controlar, y que pueden afectar a las entradas o salidas de los sistemas o procesos.

➤ VARIABLE DINAMICA

Variable que se desea mantener constante, también se la conoce como variable controlada.

➤ **VARIABLE MANIPULADA**

Es la variable que se puede cambiar para regular el proceso. Esta variable va a cambiar en la ejecución del proceso para obtener los resultados deseados, siendo esta la que permite realizar el control.

➤ **PERTURBACIONES**

Son aquellas que afectan o modifican la variable controlada. También son conocidas como interferencias, y existen dos tipos:

- *Interferencias externas.*

Son todas aquellas que afectan el sistema de medida como consecuencia del principio usado para medir la señal deseada.

- *Interferencias internas.*

Son aquellas que afectan de manera indirecta a la salida del sistema debido a los efectos sobre todas las características.

2.2.6. SISTEMAS DE CONTROL

➤ **SISTEMA DE CONTROL ON-OFF**

Es un control muy común y popular en la industria, debido a su bajo costo y fácil implementación, es muy fácil de controlar debido a que tiene dos posiciones, lo que implica que el elemento de control final o actuador puede encontrarse encendido o apagado.

En este sistema de control se debe tener muy en cuenta la histéresis que es la diferencia entre los tiempos de apagado y encendido del elemento final.

➤ **SISTEMA DE CONTROL PROPORCIONAL**

El elemento actuador permanece en una posición intermedia, la salida del controlador final siempre es proporcional a la magnitud del error, para reducir el error existente en el proceso se tiene un Offset,

la ganancia del sistema se la obtiene mediante la relación entre la variable controlada y la variable manipulada.

Este tipo de control posee una ecuación característica.

$$y = Kc \cdot e + y_0$$

De donde:

y: Salida del controlador

Kc: Ganancia Proporcional

e: error

y₀: Condiciones Iniciales

Como ventajas de este tipo de control se encuentra: que los datos se los puede comprobar de manera instantánea y los datos pueden ser comprobados con facilidad.

Las desventajas de este tipo de control es que presenta muy poca inmunidad al ruido, y de manera adicional es imposible corregir los errores en un régimen constante y permanente.

➤ **SISTEMA DE CONTROL PROPORCIONAL INTEGRAL PI**

La característica principal de este control consiste en integrar el área bajo la curva cuando el error es diferente de cero, y esta suma se la realiza en un intervalo de tiempo. Se logra eliminar el error en estado estable, el mismo que se define como la acción de control permanece constante cuando el error es de igual manera constante. Se ve reducido la velocidad de respuesta de manera considerable, pero a la vez si la velocidad del sistema es muy pequeña afecta de manera directa a la estabilidad del sistema.

Algo muy destacable respecto a los sistemas de control integral es que no existen controladores que realicen control Proporcional Integral.

➤ **SISTEMA DE CONTROL PROPORCIONAL DERIVATIVO PD**

Este tipo de control no hace la corrección del error, sino trabaja con la derivada del error con respecto al tiempo, el sistema de control trabaja con la velocidad con la que varía el error, la señal generada es siempre proporcional a la derivada del error en el tiempo, y su principal característica es que tiene un efecto de acción anticipativa, lo cual representa que se corrige el error antes de que esta señal sea excesiva.

➤ **SISTEMA DE CONTROL PROPORCIONAL INTEGRAL Y DERIVATIVO PID**

Es uno de los principales sistemas de control por retroalimentación, y este conjuga los tres tipos de control:

- Proporcional: determina la reacción del error en el momento en que se produce.
- Integral: genera una corrección proporcional a la integral del error que se produjo.
- Derivativo: porque determina la reacción del tiempo en el que el error se generó.

Aplicar un control PID no asegura que exista un control óptimo del proceso o del correcto funcionamiento del mismo debido a la estabilidad.

2.3. APARATOS DE MANDO, REGULACIÓN Y CONTROL

2.3.1. SISTEMA DE MANDO

➤ **Sistemas de mando permanente.**

Como característica principal de este tipo de mando es que al accionar el elemento de puesta en marcha este conserva su posición hasta el que operador vuelva a accionar dicho elemento.

Dentro de este grupo constan: interruptores, conmutadores, selectores, pulsador de llave, etc.



Fig. II. 6 Sistemas De Mando Permanente

Fuente: Siemens Lista De Precios Ecuador 2010

➤ **Sistema de mando instantáneo.**

Dentro de este sistema de mando se puede considerar a los dispositivos que recuperan su forma original, o regresan a su posición de reposo una vez que se deja de ejercer una fuerza sobre estos.

Para poner en marcha cualquier proceso se debe usar un contacto normalmente abierto (NA) y para detener el proceso un contacto normalmente cerrado (NC), es necesario emplear contactos auxiliares ya que una vez que se deja de realizar una fuerza sobre el dispositivo este retornara a su posición original y el proceso se verá afectado.

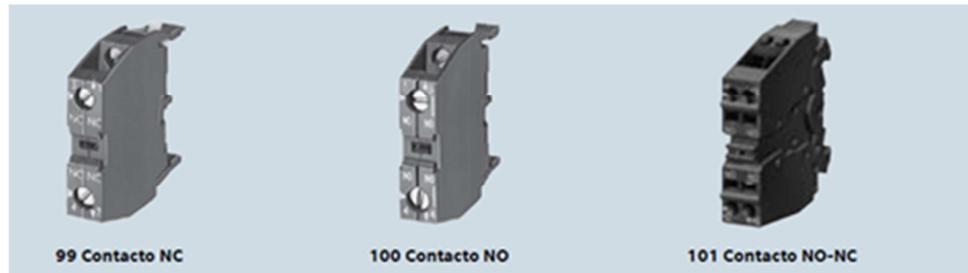


Fig. II. 7 Sistema Mando Instantáneo

Fuente: Siemens Lista De Precios Ecuador 2010

➤ **Teclados.**

Se los usa para comunicarnos con otros dispositivos, para lograr este fin es necesario usar una cantidad considerable de pulsadores, los mismos que estarán conectados en matrices para poder transmitir las ordenes.

➤ **Parada de emergencia.**

Es una medida que se adopta para prevenir peligros y daños a personas o instalaciones de la empresa. Este tipo de dispositivo es capaz de afectar de manera escalable a la industria, ya que es capaz de producir la para de una máquina, un proceso, una zona e incluso llegar a detener toda la fábrica.



Fig. II. 8 Pulsador De Emergencia

Fuente: Siemens Lista De Precios Ecuador 2010

➤ **Finales de Carrera**

Son interruptores de posición electromecánicos, y a estos se los puede dividir en dos grandes grupos, los mismos que se les detalla a continuación:

- Interruptores de control, el papel principal de estos es, detectar el paso de un objeto, generalmente se los conecta a las entradas de una unidad para procesar los datos.
- Interruptores de potencia, estos se los emplea para seguridad de los equipos e instalaciones, ya que se los coloca en las fases de alimentación.



Fig. II. 9 Finales De Carrera

Fuente: Siemens Lista De Precios Ecuador 2010

2.3.2. SISTEMAS DE SEÑALIZACIÓN

La señalización de los equipos consiste en colocar dispositivos de manera muy general de tipo luminoso, las mismas que sirven para indicar el estado de funcionamiento de la maquinaria y de los procesos que ocurren dentro de las instalaciones de la empresa o lugar donde se realiza la actividad.

Los colores que poseen cada uno de los dispositivos tienen su propio significado, y este puede ser considerado de muchas formas, aunque una manera muy general es que el color rojo sirve para indicar que existe peligro, el amarillo indica un mal funcionamiento dentro del proceso, y un color verde indica que la maquinaria está funcionando correctamente.



Fig. II. 10 Indicadores Luminosos

Fuente: Siemens Lista De Precios Ecuador 2010

2.3.3. INTERRUPTORES DE CONTROL

Son dispositivos que se encargan de monitorear las magnitudes físicas, químicas que intervienen dentro de un proceso de producción, las mismas que deben estar parametrizadas para que el proceso tenga un correcto funcionamiento.

➤ ***Control de nivel.***

Son dispositivos que permiten realizar un control de líquidos o fluidos, estos nos indican el nivel que existe en un recipiente, permitiendo de esta manera que el proceso se detenga o se ponga en marcha.

➤ ***Control de presión.***

Estos nos permiten controlar el aumento o disminución de la presión del proceso, abriendo o cerrando contactos los mismos que permiten el paso o el cierre de válvulas. Para poder hacer uso de estos dispositivos se debe tener en cuenta lo siguiente:

- A qué proceso está dirigido.
- En sentido de trabajo en uno solo o en ambos.
- Si se deben tener consideraciones de los niveles.
- El fluido con el que va a trabajar.
- La precisión del elemento.
- Los factores ambientales, lugar donde se colocara el dispositivo.

➤ **Control de temperatura.**

Son dispositivos cuya función es controlar la temperatura del proceso, poseen contactos los mismos que van cambiando de posición de acuerdo al valor que censa.

Un uso específico dentro de las industrias es para mantener una temperatura deseada sea de cocción o de refrigeración. Una consideración importante que se debe tener es que estos dispositivos se los selecciona de acuerdo al rango de temperatura que se desea controlar.

➤ **Detectores inductivos y capacitivos.**

Son dispositivos estáticos y no poseen piezas de mando o elementos móviles, gracias a su gran resistencia a ambientes hostiles que presentan humedad, polvo o ambientes corrosivos.

De manera adicional la frecuencia de trabajo no acorta la vida útil de uno de estos dispositivos, tiene un tiempo de respuesta muy corto lo que reduce la posibilidad de fallas o errores.

Estos detectores disponen de un potenciómetro que regula la sensibilidad del dispositivo.

Existen dos tipos de detectores:

- **Inductivos**, estos dispositivos son capaces de detectar materiales conductores, presenta una variación que sufre un circuito electromagnético cuando a este se le aproxima un objeto metálico.
- **Capacitivos**, este grupo de elementos son capaces de detectar cualquier tipo de objetos, pueden ser estos: líquidos, plástico, vidrio, caucho aun cuando cualquiera de estos este expuesto a condiciones indeseadas.

➤ **Detectores fotoeléctricos.**

Son elementos que constan de dos partes, de un emisor y un receptor de luz, la detección es efectiva cuando el objeto ingresa en el haz luminoso y altera la cantidad de luz que percibe o recibe el receptor para provocar un cambio en la salida del circuito.

2.3.4. ELEMENTOS DE PROTECCION

Estos elementos tienen como función principal proteger todos los dispositivos que se encuentran en una instalación industrial y también proteger a los usuarios que se encuentren dentro de la misma área.

Esto quiere decir que la finalidad de emplear estos elementos es proteger al circuito y los actuadores que complementan al mismo.

Los sistemas de protección más comunes son los fusibles y los interruptores de protección, la forma de escoger cuál de estos se emplea en una determinada aplicación son las condiciones del circuito, esto implica la intensidad de corriente, el número de polos que se necesita en el circuito si es monofásico, bifásico o trifásico.

➤ **RELE TERMICO**

Son elementos de protección que protegen a los actuadores en este caso motores contra sobrecargas que pueden producirse en el arranque o durante el trabajo del actuador.

El principio de funcionamiento de este dispositivo es la deformación de un bimetálico por efectos del calor, que se deforma de acuerdo a las características que presenta cada una de las líneas conectadas a sus fases. Lo más relevante de estos instrumentos es que se los puede usar con elementos de corriente alterna y continua.

Tipos de relés:



Fig. II. 11 Relé Térmico

Fuente: Siemens Lista De Precios Ecuador 2010

➤ **RELE TERMOMAGNÉTICO**

Es un aparato capaz de interrumpir o abrir un circuito eléctrico cuando la intensidad de la corriente eléctrica que por él circula excede de un determinado valor, o, en el que se ha producido un cortocircuito, con el objetivo de no causar daños a los equipos eléctricos.

Se fabrican disyuntores de diferentes tamaños y características lo cual hace que sea ampliamente utilizado en viviendas, industrias y comercios.

Características

Los parámetros más importantes que definen un disyuntor son:

- **Calibre o corriente nominal:** Corriente de trabajo para la cual está diseñado el dispositivo. Existen desde 5A hasta 64A.
- **Tensión de trabajo:** Tensión para la cual está diseñado el disyuntor. Existen monofásico (230 V) o trifásico (400 V).
- **Poder de corte:** Intensidad máxima que el disyuntor puede interrumpir.
- **Poder de cierre:** Intensidad máxima que puede circular por el dispositivo en el momento de cierre sin que éste sufra daños por choque eléctrico.

- **Número de polos:** Número máximo de conductores que se pueden conectar. Existen disyuntores de uno, dos, tres y cuatro polos.



Fig. II. 12 Relé Termomagnético

Fuente: Autor

➤ **RELE ELECTROMAGNÉTICO**

Estos elementos pueden ser de uno o varios contactos de acuerdo a las necesidades que presenta el circuito eléctrico, de esta manera se puede conectar o desconectar cuando este dispositivo se acciona.

Se los utiliza para proteger circuitos de sobrecargas grandes, el tiempo de desconexión es inmediato.

Su funcionamiento se encuentra basado en un electroimán que es atraído sobre una armadura metálica, generando la conexión o desconexión.

2.3.5. SENSORES DE TEMPERATURA: BIMETALES

Se considera bimetalo a los elementos o piezas que están formados por dos metales cuyo coeficiente de dilatación térmica es diferente, y que se encuentran unidos de manera firme y consistente.

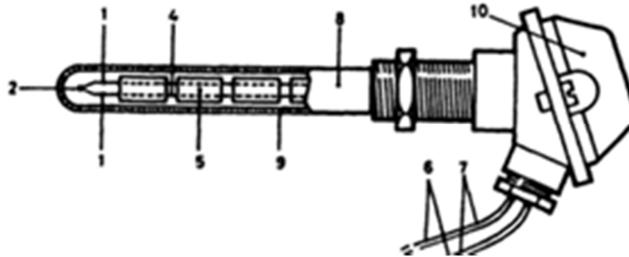


Fig. II. 13 Termocupla

Fuente: Pallas. Sensores y acondicionadores de señal

➤ **TERMOCUPLA**

Las termocuplas son elementos muy comunes que miden la temperatura en forma de señal eléctrica que generalmente se encuentra en el orden de los milivoltios (mV) que varían de acuerdo a la temperatura. En muchas bibliografías también se los conoce como termopares.

La aplicación de los termopares es muy extensa debido a varios factores entre los cuales se puede destacar: conectores estándar, relativamente económicos y con un amplio rango de medida de temperatura además de ser intercambiables.

De acuerdo al rango de temperatura que mide varía las características de construcción de cada termopar, de manera general la estructura física de estos sensores son una capsula o vaina de metal, dentro de esta capsula se encuentran los metales.

Para una correcta aplicación de los termopares se debe tener en cuenta las siguientes consideraciones:

- La temperatura máxima que alcance el termopar debe ser inferior a la temperatura de fusión. (PALLAS, 2003)
- La corriente que circula por el circuito de los termopares debe ser mínima. Ya que debido a los efectos que están expuestas los sensores generarían valores diferentes a los del entorno.
- Se debe mantener una de las dos uniones a una temperatura fija si se desea medir la otra unión, toda variación que se produzca en la unión será considerada una fuente de error.

➤ **TERMOCUPLA TIPO K**

Este termopar está construido de Cromel que es una aleación de Níquel y Cromo, y Alumel que es una aleación de Níquel y Aluminio.

El rango de operación se encuentra entre los -200°C a $+1372^{\circ}\text{C}$ con una variación respecto a la sensibilidad de $41 \mu\text{V}/^{\circ}\text{C}$.

Debido a su característica constructiva son muy resistentes a altas temperaturas y a la oxidación.



Fig. II. 14 Termopar Tipo K

Fuente: Autor

- **CURVA CARACTERISTICA TERMOCUPLA TIPO K**

La salida de las termocuplas o termopares no son lineales, estas varían de la temperatura que censan, lo que implican que la salida en el termopar no es constante.

En la fig.II.15 se muestra la Fuerza termo- electromotriz que depende de los metales y de la diferencia de temperatura entre las dos uniones.

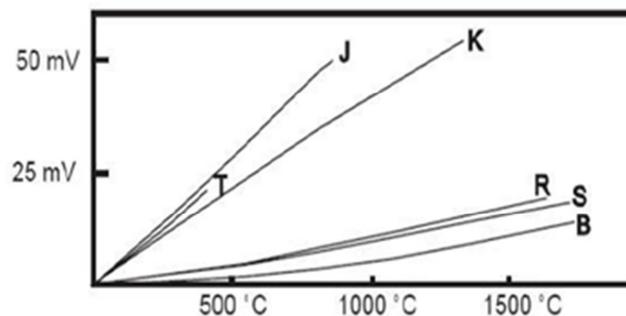


Fig. II. 15 Curvas Características Termopares

Fuente: pallas. Sensores y acondicionadores de señal

- **APLICACIÓN DE LA TERMOCUPLA TIPO K**

Se emplean en atmosferas no reductoras y en su margen de medida son mejores que lo de tipo E, J y T cuando se las pone a medir en ambientes oxidantes.

El termopar tipo K generalmente se lo emplea en la industria de la fundición y en hornos cuya temperatura máxima no supera los 1300°C, por ejemplo se lo utiliza en hornos de cocción de baldosas, hornos de cocción de alimentos, etc.

- **LINEALIZACIÓN**

Debido a que los termopares generan una señal no lineal se requiere o es necesario linealizar la salida para que esta pueda ser manipulada de mejor manera. Para llevar a cabo este fin en este trabajo de investigación se ha utilizado el integrado AD595.

2.3.6. AD595

El AD595 es un instrumento de amplificación completo y un compensador de unión fría para termocuplas en un chip monolítico. En este dispositivo se combina un punto de referencia frío con un amplificador pre calibrado para generar una señal de alto nivel de 10mV/°C de salida directamente desde la señal de la termocupla.

Este chip se encuentra ensamblado con una alarma de fallo, que indica cuando el termopar se encuentra defectuoso o cuando uno o los dos cables del sensor se encuentran abiertos, la salida puede ser usada de varias formas incluyendo un accionamiento de nivel TTL.



Fig. II. 16 AD595

Fuente: Autor

➤ **DATOS TÉCNICOS DEL AD595**

- De manera adicional se lo puede usar con termocupla tipo T como entrada.
- Baja impedancia voltaje de salida: 10mV/°C.
- Punto de compensación de juntura fría.
- Amplio Rango de alimentación: de +5V a +15V.
- Bajo consumo: <1mW.
- Alarma por falla de termocupla.
- SetPoint como modo de operación.

➤ **DIAGRAMA DE BLOQUE FUNCIONAL DEL AD595**

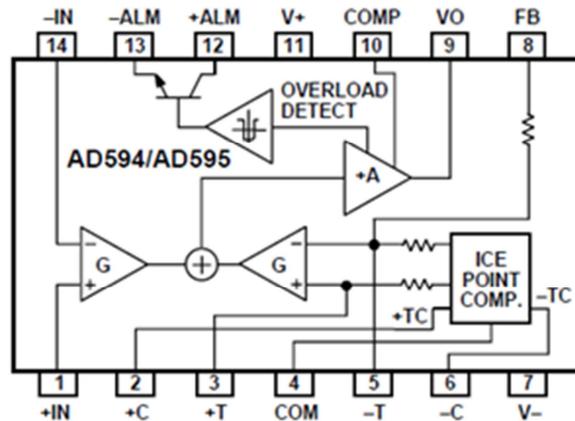


Fig. II. 17 Diagrama De Bloque Funcional AD595

Fuente: Hoja Técnica Del AD595

2.3.7. DEFINICION DE PLC

PLC es un controlador lógico programable, es un dispositivo que se lo aplica en automatización industrial, un autómata programable tiene muchas aplicaciones dentro del campo industrial, éste está diseñado con varias entradas y salidas, un rango de temperatura extenso para soportar el trabajo, presenta un alto grado de inmunidad al ruido eléctrico y soporta vibraciones y son resistentes a los golpes. Las salidas de un autómata programable deben ser una respuesta adecuada a las condiciones de entrada en un tiempo determinado para obtener los resultados deseados.

Los autómatas programables se los emplea en procesos industriales que poseen una o varias necesidades en las cuales el espacio de aplicación es de vital importancia, los procesos en los que existen cambios periódicos, instalaciones con procesos amplios y complejos.

Se los aplica de manera general para realizar maniobras de máquinas, maniobra de instalaciones y para señalar y controlar procesos.

➤ **VENTAJAS**

- Se reduce de manera considerable el tiempo en la elaboración de los proyectos.
- Se puede añadir modificaciones sin que estas conlleven más gastos, ya sea añadiendo componentes que sirvan para la utilización propia del PLC.
- Se los puede ubicar en espacios muy reducidos.
- Se puede controlar varias máquinas con un mismo controlador.
- El tiempo de repuesta es menor a otros controladores, lo que reduce el tiempo de puesta en funcionamiento.
- Se lo puede reutilizar cuando el autómeta queda pequeño en relación al proceso.
- El mantenimiento es muy económico.

➤ **DESVENTAJAS**

Entre los principales inconvenientes se encuentran:

- La manipulación debe ser realizada por personal capacitado.
- Se necesita de software específicos para la creación y manipulación de los programas
- La inversión inicial para conseguir un PLC puede ser muy costosa o elevada.

➤ **SIEMENS LOGO**

Siemens LOGO! Es un módulo lógico programable el cual lleva integrado:

- Elementos de control.
- Unidad de operación y visualización.
- Fuente de alimentación.
- Interface para módulos de programa y cable de PC.
- Reloj Temporizador.

- Marcas Binarias.
- Un determinado número de entradas y salidas de acuerdo al tipo de equipo que se posee.
- Funciones básicas con activación y desactivación retardada y un relé de impulsos.

Mediante log se pueden dar solución a varios problemas que se pueden presentar ya sean estos en el ambiente doméstico, en invernaderos y para aplicaciones industriales que poseen maquinas pequeñas.

Se puede dar solución a cualquier problema siempre y cuando el relé programable cumpla con las condiciones de entrada y salida previstas para dar solución al problema.

▪ **EQUIPOS EXISTENTES**

Existen varios modelos de los cuales se pueden destacar:

- 12/24 V c.c
- 24 V c.c
- 24 V c.a
- 230 V c.a

Cada uno de estos modelos presentan sus variantes, las mismas que son mayor o menor cantidad en el número de entradas y salidas que posee cada dispositivo, de manera adicional se debe tener en cuenta el tipo de salida que posee cada dispositivo, ya que algunos cuentan con salidas de relé y otros con salida de transistores.

I. EQUIPO EMPLEADO

LOGO! 230RC, Módulo Lógico, tiene Display, voltaje de alimentación a las entradas y salidas del LOGO 115V/ 230V/ RELAY.

Tiene 8 entradas digitales, 4 salidas digitales, memoria de hasta 200 bloques, módulos de expansión extra con alimentación de 230 V AC/DC.



Fig. II. 18 SIEMENS LOGO 230RC 0BA6
Fuente: Hoja Técnica 6ED1052-1FB00-0BA6



DATOS TÉCNICOS

- Montaje: sobre rail DIN de 35mm.
- Voltaje de alimentación: 115/230 V AC/DC.
- Número de entradas: 8
- Número de salidas: 4 ;relays.
- Grado de protección: IP20.
- Temperatura de operación: 0°C a 55°C.

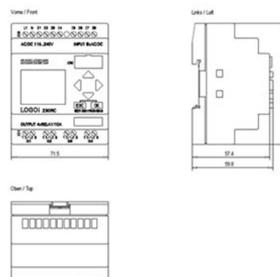


Fig. II. 19 Perspectivas Del LOGO! 230 RC

Fuente:<https://mall.industry.siemens.com/mall/es/ec/Catalog/Product/6ED1052-1FB00-0BA6>

II. MÓDULOS DE AMPLIACIÓN

Existen una amplia gama de módulos de expansión de los cuales se encuentran los siguientes:

- Módulos digitales LOGO! Con voltajes de alimentación de 12V DC, 24V AC/DC y 115...240V AC/DC, en la presentación con 4 entradas y 4 salidas.
- Módulos analógicos LOGO! Con voltajes de alimentación de 12V DC, 24V DC, cuya presentación tiene 2 entradas analógicas o con 2 entradas Pt100.



MÓDULO EMPLEADO

LOGO! AM2 Módulo de Expansión, DC 12/24V, 0 – 10V o 0-20mA.



Fig. II. 20 Módulo De Ampliación

Fuente: Hoja Tecnica Del Producto 6ED1055-1MA00-0BA0

a) DATOS TECNICOS

- Montaje: sobre rail DIN de 35mm.
- Voltaje de alimentación: 12/24 V DC.
- Número de entradas analógicas: 2
- Rango de voltaje de entrada: 0 a +10 V.
- Rango de corriente de entrada: 0 a 20mA.
- Grado de protección: IP20.
- Temperatura de operación: 0°C a 55°C.

2.3.8. DEFINICIÓN DE ARDUINO

Arduino es una plataforma de hardware y software libre, su estructura se encuentra basada en un microcontrolador de la familia Atmel AVR. Además Arduino tiene un entorno de programación que permite desarrollar los proyectos electrónicos de manera más sencilla y eficiente.

Debido a la gran variedad que poseen dentro de los microcontroladores y el bajo costo se convierten en una gran opción para realizar grandes y múltiples diseños electrónicos.

Arduino puede detectar el medio ambiente mediante la recepción de la entrada de una gran variedad de sensores. El microcontrolador se programa mediante un IDE de programación de Arduino el mismo que se encuentra basado en Wiring.

De manera adicional Arduino brinda muchas prestaciones en las cuales podemos destacar la comunicación con ordenadores o simplemente ser proyectos completamente autónomos. (ARDUINO)

Para programar Arduino se debe tener en cuenta las siguientes consideraciones:

- **Estructura del programa**

La programación de Arduino es bastante sencilla y muy semejante a la programación en lenguaje C.

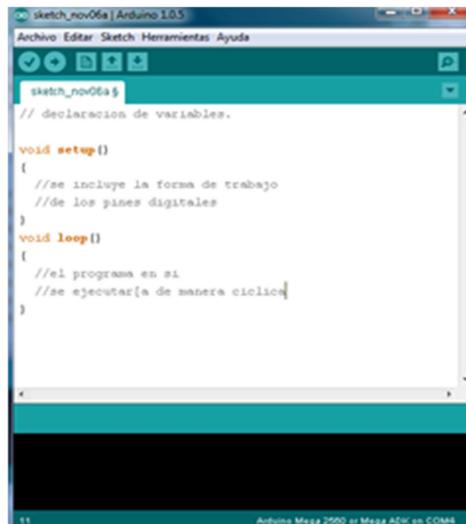
Un programa realizado en el IDE de Arduino debe contener dos partes importantes y necesarias en todo programa, las funciones voidsetup y voidloop.

La función voidsetup es aquella que tiene como función la configuración de todos los parámetros con los que va a trabajar la placa de Arduino, esta función se ejecuta una sola vez al iniciar el programa, en ella debe constar los modos de trabajo de los pines digitales y la configuración del puerto serial en caso de ser necesario.

Aun no exista ninguna línea de programación dentro de esta función, ésta debe ser declarada.

La función voidloop todo lo que se encuentra dentro de esta función se ejecutara de manera cíclica hasta que la placa sea desconectada. Dentro de esta función se declarará todas las acciones que ejecutara la placa de manera intermitente por lo cual realizara las acciones necesarias de acuerdo a los cambios que se presenten en el hardware.

Estructura de un programa de Arduino, en el IDE de programación:



```
sketch_nov06a | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
sketch_nov06a $
// declaración de variables.
void setup()
{
  //se incluye la forma de trabajo
  //de los pines digitales
}
void loop()
{
  //el programa en si
  //se ejecutara de manera ciclica
}
```

Fig. II. 21 Estructura Programa Arduino

Fuente: Autor

- **Variables**

Es la manera en la que se nombrarán a los valores que se desean usar en la ejecución del programa, las variables irán cambiando de valor de acuerdo a la ejecución del programa. Las variables deben ser declaradas considerando el tipo de valor con el que va a trabajar y un nombre para que sea de fácil manejo de manera opcional se las puede asignar un valor.

Las variables pueden ser declaradas en cualquier parte del programa, por lo que se les puede considerar como variables globales y locales, aunque lo más recomendable es declararlas antes de la función setup.

Ejemplos de la forma de declarar:

```
intaut=3; // señal del selector en modo automatico
```

```
intest_aut=0;
```

- **Tipos de datos**

Se puede usar varios tipos de datos, dependiendo de las características y condiciones que sean necesarias para el programa de Arduino.

Como más importantes se destacan los siguientes:

- *Byte*.- valores numéricos de hasta 8 bits sin punto decimal.
- *Int*.- valores numéricos de hasta 16 bits sin números decimales.
- *Long*.-valores numéricos de hasta 32 bits sin números decimales.
- *Float*.- números con punto decimal.
- *Boolean*.- valores de verdadero o falso.
- *Arrays*.-se usa para crear matrices.

- **Operadores matemáticos**

Se usan todos los operadores que permitan obtener un valor como resultado de la operación. Se tienen varios tipos de operadores entre los cuales existen los operadores matemáticos, operadores de comparación y operadores lógicos o booleanos.

- **Valores constantes**

Son aquellos que durante la ejecución del programa no van a cambiar nunca de valor.

- **Estructuras de control**

Permiten realizar el control de datos de manera deseada para obtener los resultados deseados con la programación. Al igual que la mayoría de los lenguajes de programación C, las estructuras son:

- If
- If- else
- For
- While
- Do- while

- **Entradas/ salidas digitales**

Los pines digitales de la placa de Arduino pueden ser usados como entradas y salidas, las mismas para ser usadas de la manera anteriormente mencionada es necesaria usar las palabras reservadas, OUTPUT o INPUT.

- **Entradas/ salidas analógicas.**

Los pines analógicos son capaces de leer un valor análogo conectado a cualquier pin, con una resolución de 10 bits, de manera adicional se pueden usar las salidas PWM, dando un valor intermedio entre 0 y 5 voltios.

- **Tiempos de espera**

Existen dos formas de poner contabilizar el tiempo en Arduino.

El comando `delay(ms)`; permite detener el programa durante el tiempo que se encuentra entre paréntesis.

El comando `millis()`; devuelve el valor transcurrido desde que comienza la ejecución del programa, o desde el momento en que fue pulsado el botón RESET.

➤ **MODELOS DE PLACAS DE ARDUINO**

Arduino posee una gran variedad de placas las mismas que han sido creadas con fines específicos, aunque otras pueden ser consideradas como placas multifunción.

Dentro de la gran gama de placas existentes se pueden destacar las siguientes:

- Arduino Uno
- Arduino Leonardo
- Arduino Micro
- Arduino Robot
- Arduino Ethernet
- Arduino Mega ADK/ 2560
- Arduino Nano

▪ **PLACA EMPLEADA ARDUINO MEGA 2560**

Arduino Mega 2560 es placa basada en un microcontrolador ATmega2560. Posee 54 entradas/salidas digitales de las cuales 14 tienen la opción de ser usadas como PWM, 16 entradas analógicas, 4 UARTs que permiten realizar comunicación serial, un cristal oscilador de 16MHz, un conector USB, un botón de reseteo. (Arduino Home, 2014)



Fig. II. 22 Arduino Mega 2560

Fuente: Autor

➤ **ESPECIFICACIONES TECNICAS DE ARDUINO MEGA 2560**

Tabla II. I Datos Técnicos Arduino Mega 2560

DETALLE	
Microcontrolador	ATmega2560
Voltaje de Operación	5V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (limite)	6-20 V
I/O pines digitales	54
I/O pines PWM	14
Entradas analógicas	16
I/O Pin dc corriente	40Ma
I/O Pin dc corriente a 3.3V	50mA
Flash Memory	256KB
Peso	37g

➤ **MÓDULOS DE AMPLIACIÓN**

Para una mayor facilidad de desarrollo y aplicación de Arduino se han creado interfaces, las cuales se montan sobre la placa que se tenga para ampliar las características de Arduino.

Existen varios Shield que permiten mejorar las características de Arduino entre los cuales se destacan:

- Arduino GSM
- Arduino Ethernet
- Arduino WiFi
- Arduino Wireless
- Arduino USB Host
- Arduino Motor
- Arduino Proto

➤ **INTERRUPCIONES**

Las interrupciones sirven para interrumpir la ejecución del programa, generalmente se las usa para realizar un salto en la ejecución del programa una vez que ejecuta la interrupción regresa al lugar donde fue interrumpido el programa.

Las placas de Arduino poseen dos interrupciones externas en los pines digitales 2 y 3.

Se las conoce como ISR (interruptServiceRoutine), la rutina de intervención de servicio.

En el caso de la tarjeta Arduino Mega posee 5 pines de interrupción.

Tabla II. // Interrupciones Arduino Mega 2560

N° Inter.	Inter. 0	Inter. 1	Inter. 2	Inter. 3	Inter.4	Inter.5
N° Pin	2	3	21	20	19	18

Para emplear una interrupción se debe emplear el siguiente código:
`attachInterrupt(N° Interrupción, nombre de la función, tipo de interrupción);`

Este código se lo declara en el voidsetup del programa.

En momentos donde es imposible ejecutar la interrupción se la debe enmascarar, para lo cual se emplea el siguiente código:
`detachInterrupt(N° de Interrupción);`

• **Tipos de Interrupción.**

Existen 4 formas definidas en las que se puede realizar la interrupción.

- LOW.- Salta a la ISR cada vez que el pin digital se encuentra en modo o nivel bajo.

- CHANGE.- Ejecutará la ISR cada vez que se presenta un cambio de nivel en el pin de interrupción, sea de un estado de nivel Alto a Bajo o de Bajo a Alto.
- RISING.- Ejecutará la ISR si y solo si el nivel de entrada varía de un nivel Bajo a Alto.
- FALLING.- Saltará a la rutina de interrupción de servicio cuando el nivel de entrada cambia de Alto a Bajo.

➤ **CONFIGURACIÓN DE RESITENCIAS.**

Arduino es un dispositivo electrónico cuyos niveles de voltaje en sus pines digitales varían en dos estados:

- HIGH: también considerado como un 1 lógico, es cuando en la entrada digital tiene un voltaje igual a +5V.
- LOW: se considera a un 0 lógico, cuando en la entrada digital se tiene un valor de 0 voltios.

Los valores en los pines digitales de la placa de Arduino se pueden ver afectados por varios factores que pueden alterar el verdadero estado de la entrada, uno de los principales factores que influye en falsos estados es el ruido eléctrico.

De acuerdo a los valores con los que se desea trabajar en los pines digitales se puede usar una de las siguientes configuraciones, de manera adicional cabe destacar que al realizar esto se disminuye el ruido eléctrico que se puede generar:

- **Configuración Pull Down**

En esta configuración la resistencia se conecta a tierra (GND), en este caso los interruptores o pulsadores se colocaran en antes de la resistencia, así el funcionamiento de este circuito se lo describe de la siguiente manera: al cerrarse el interruptor, toda la corriente circula por el lado que presenta menos resistencia eso implica que se dirige a la

entrada digital, pero al abrirse el circuito la corriente se dirige hacia la resistencia lo que genera un valor lógico de 0 voltios en dicha entrada.

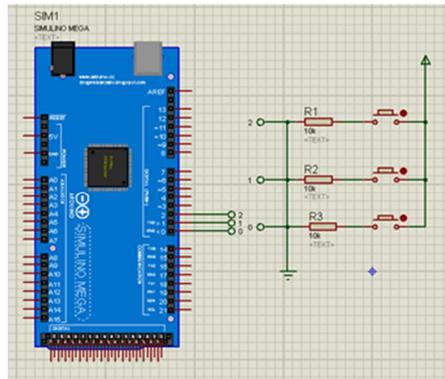


Fig. II. 23 Configuración Pull Down

Fuente: Autor

- **Configuración Pull Up**

En esta configuración la resistencia se la ubica después de la alimentación o a los 5V, y el interruptor después de la resistencia.

El funcionamiento de este circuito es opuesto a la configuración Pull Down, esto quiere decir que cuando el interruptor está abierto toda la corriente circula hacia la entrada de la tarjeta, y cuando el interruptor o pulsador es accionado toda la corriente se desplaza hacia tierra porque encuentra menos oposición al paso de corriente esto implica que se obtiene un 0 lógico en la entrada.

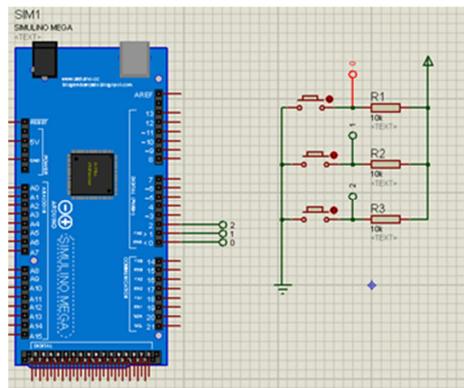


Fig. II. 24 Configuración Pull Up

Fuente: Autor

2.3.9. SOFTWARE

Es el programa o conjunto de programas que nos permiten realizar rutinas para que estas realicen una función o tarea específica.

La mayoría de dispositivos programables poseen sus propios grupos de programas que permiten realizar las actividades necesarias para programarlos, siendo éstas únicas con formas y diseños propios.

➤ **SOFTWARE SIEMENS LOGO!**

El software que se emplea es el paquete denominado

➤ LOGO! Soft Comfort.

Este software incluye las siguientes facilidades:

- Crear programas OFFLINE para la aplicación, se lo puede crear en la PC y no de manera directa en el LOGO.
- Simulación de su circuito en la PC, para comprobar el funcionamiento del diseño realizado.
- Generar e imprimir el esquema generado. Se puede imprimir el circuito de manera completa o en varios diagramas.
- Se puede proteger el programa, porque se lo puede almacenar en discos externos o propios de la PC, lo que permitirá volver a cargar el programa en caso de pérdida o realizar modificaciones al programa cuando este lo requiera.
- Transferencia del programa desde LOGO! Al PC y viceversa.

En LOGO!SoftComfort se pueden elaborar los programas de conmutación de forma eficiente, confortable y transparentes.

LOGO!SoftComfort se lo puede ejecutar a partir de Windows 95, en la actualidad no presenta ningún inconveniente en ejecutarlo sobre Windows 7.

➤ **LOGO!SoftComfort V7.1**

Es una de las versiones más recientes de este software, el mismo que puede trabajar con LOGO de la versión OBA6, esta actualización permite trabajar de manera normal con versiones anteriores de LOGO!. Se debe comprar las licencias para hacer uso de este software, pero por fines educativos las licencias son gratuitas.

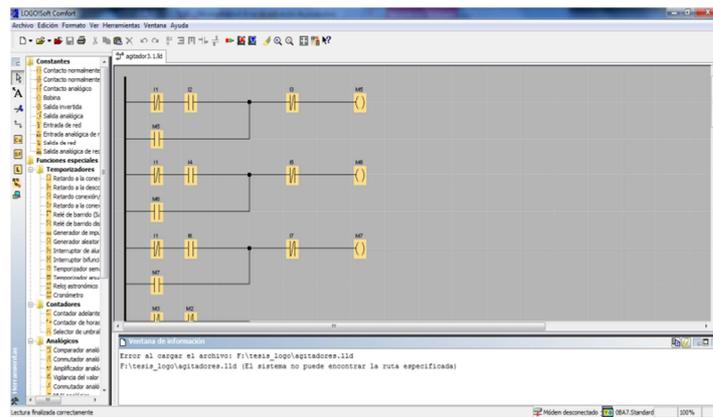


Fig. II. 25 Software LOGO! SOFT

Fuente: Autor

➤ **SOFTWARE ARDUINO.**

Arduino es software libre, por lo cual no se debe pagar nada por descargarlo y usarlo.

El código abierto de Arduino lo convierte en un programa que tiene un código fácil de escribir y cargar a la placa detallando las entradas y salidas de la misma. Se ejecuta sobre Windows, MAC OS y Linux.

El entorno de programación de Arduino se encuentra escrito sobre Java y basado en Processing, avr-gcc, y otro software de código abierto.

Para descargar el software se debe dirigir a la página oficial de Arduino y de acuerdo a las características del sistema operativo que se tiene en la PC se debe descargar la versión de Arduino.

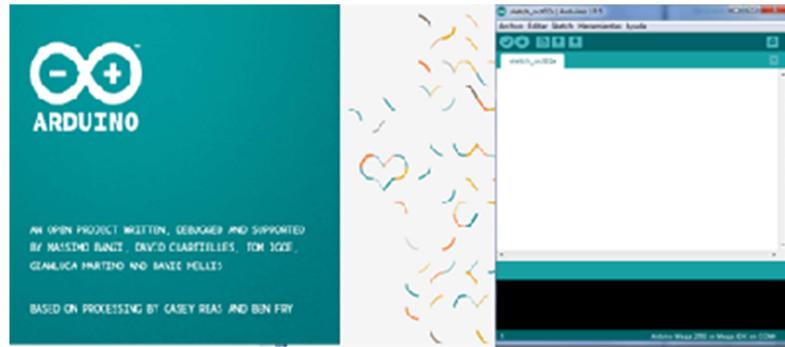


Fig. II. 26 Pantalla Principal Arduino

Fuente: Autor

➤ **LABVIEW**

LabView (Laboratory Virtual Instrument Engineering Workbench) es un entorno de desarrollo que se basa en una programación gráfica. Este tipo de programación utiliza íconos, terminales y cables para reemplazar al lenguaje textual, describiendo de esta manera las acciones de programación.

Tiene muchos tipos de comunicación con Hardware entre los principales se encuentran el GPIB, VXI, RS-232, RS-485 y tarjetas de adquisición de datos plug-in.

LabView permite resolver varios problemas, es necesario seguir un método en la programación con LabView, para realizar un código que sea más entendible, escalable y modificable. Tiene mucha aplicación en sistemas de medida, procesamiento digital de señales, manipulación de audio y video, problemas de automatización, generador de señales, entre otras funciones.

▪ **INSTRUMENTOS VIRTUALES (VIs)**

Todos los programas que se crean en LabView son denominados como instrumentos virtuales, o Vis, ya que su apariencia y operación se asemejan o imitan a los instrumentos físicos.

LabView tiene un conjunto comprensivo de Vis y funciones que permiten adquirir, analizar, desplegar y almacenar datos, de manera adicional tiene herramientas que permiten solucionar problemas en el código creado.

▪ **PARTES DE UN VI**

Todos los VIs de LabView contienen tres partes principales:

- Panel Frontal
- Diagrama de bloques
- Panel de icono/conector

I. **PANEL FRONTAL**

El panel frontal es la interfaz de usuario para el VI. En el panel frontal se pueden colocar controles e indicadores los mismos que representarán los terminales interactivos de entrada y salida del VI. (National Instruments, 2006)



Fig. II. 27 Panel Frontal

Fuente: Autor



CONTROLES E INDICADORES

El panel frontal se construye con controladores e indicadores, los cuales son los terminales interactivos de entrada y salida del VI, respectivamente. (National Instruments, 2006)

Los controles son los dispositivos de entrada entre los cuales podemos destacar perillas, termómetros, botones y mucho más dispositivos, los controladores se encargan de suministrar los datos al diagrama de bloques del VI.

Los indicadores son gráficos tales como Leds y otros dispositivos de despliegue, estos se encargan de simular los dispositivos de salida de instrumentos y despliegan los datos que se adquirieron o generaron en el diagrama de bloques.



CONTROLES E INDICADORES NUMERICOS

El tipo de dato numérico se lo puede representar de muchas maneras, como por ejemplo números enteros o reales. Los objetos más comunes de LabView son el Control numérico y el indicador numérico aunque existen más formas de representar estos datos como por medio de diales o medidores.



CONTROLES E INDICADORES BOOLEANOS

Los tipos de datos booleanos se encuentran representados de dos maneras en sus estados: VERDADERO o FALSO u ON y OFF. Estos objetos se los utiliza para ingresar y mostrar valores de True o False.



CONTROLES E INDICADORES DE CADENA

El tipo de cadena es una secuencia de caracteres ASCII (National Instruments, 2006). Se usan controles de cadena para recibir texto ingresado por el usuario, y los indicadores de cadena sirven para mostrar texto al usuario.



BARRA DE HERRAMIENTAS

Esta barra de herramientas está asociada al panel frontal, las funciones más destacadas en esta barra sirven para correr y editar los VI.



Fig. II. 28 Barra De Herramientas Panel Frontal

Fuente: Autor

II. DIAGRAMA DE BLOQUES

El diagrama de bloques contiene el código fuente gráfico, en este existen todos los elementos que se colocaron en el panel frontal. En esta ventana todos los objetos del panel frontal aparecen como terminales.

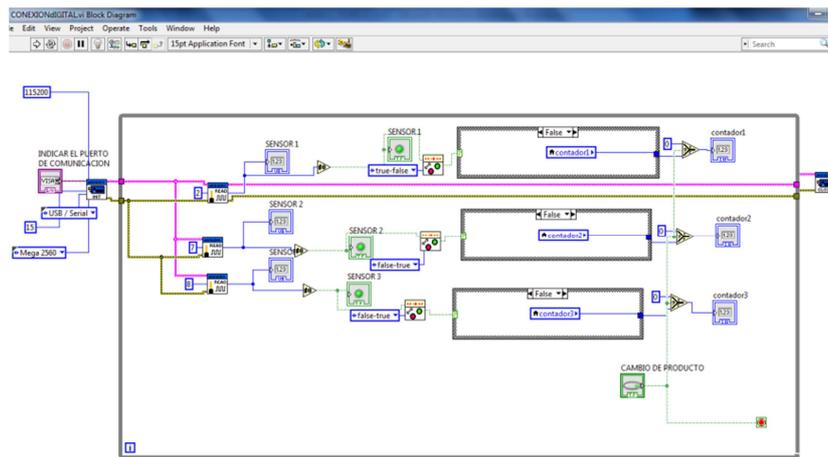


Fig. II. 29 Diagrama De Bloques

Fuente: Autor



TERMINALES

Los objetos del panel frontal se muestran en el diagrama de bloques. Los terminales son puertos de entrada y salida que comparten información entre el panel frontal y el diagrama de bloques (National Instruments, 2006).

Existen varios tipos de terminales entre los que se pueden destacar los de control o indicador y terminales de nodo.

Los terminales muestran el tipo de dato del control o del indicador.



NODOS DEL DIAGRAMA DE BLOQUES

Los nodos son objetos que poseen entradas y/o salidas y realizan operaciones cuando un VI se está ejecutando.

a) FUNCIONES

Son elementos muy fundamentales de operación para LabView, estos objetos no poseen paneles frontales o diagramas de bloques pero tienen paneles conectores.

b) SubVIs

Los SubVIs son instrumentos virtuales que se construyen previamente para ser usados dentro de otros.

Cualquier VI puede ser usado como un SubVI.

c) ESTRUCTURAS

Son elementos de control del proceso, entre los cuales se puede destacar las estructuras:

- Case
- ForLoops
- WhileLoops



CABLES

Sirven para transmitir o transferir los datos entre los objetos mediante estos elementos.

Los cables varían de color, estilo y espesor de acuerdo al tipo de dato con el que se trabaja.



BARRA DE HERRAMIENTAS

Cuanto se ejecuta un VI aparecen los botones de la barra de herramientas los que se usan para depurar el VI.



Fig. II. 30 Barra De Herramientas Diagrama De Bloques

Fuente: Autor

III. PANEL DE ICONO/CONECTOR

Se puede usar un VI dentro de otro VI, a esto se lo conoce como un subVI

Lo que se asemeja a una función en el lenguaje de programación en texto.

Para hacer uso de un VI como subVI se debe tener un ícono y un panel conector.

➤ SONÓMETRO

Los sonómetros son instrumentos de medida, que se usan para medir el nivel de ruido que se produce o genera en un determinado lugar. La unidad de medida del ruido son los decibelios.

Para el presente estudio se empleó la aplicación Sonómetro – Sound Meter Lite. Es parte del paquete Smart Tools. Esta aplicación se la puede descargar de Google Play e instalar en teléfonos que tienen sistema operativo Android.

Dicha aplicación usa el micrófono del teléfono para medir el nivel de ruido existente en un lugar, la medición se la realiza en decibelios. La calibración

con la que trabaja está realizada con un medidor de ruido real. Posee un indicador gráfico en tiempo real, un indicador analógico y digital que muestra el valor del ruido.

Se empleó esta aplicación para tener un valor estimado del ruido al cual se encuentra sometida la placa de Arduino.



Fig. II. 31 Sonómetro

Fuente: Google Play

➤ **VIBRÓMETRO**

Los vibrómetros son instrumentos de medida, que se usan para cuantificar las oscilaciones mecánicas que se produce o genera en un determinado lugar.

Para el presente estudio se empleó la aplicación Vibrómetro – Vibration Meter. Es parte del paquete Smart Tools.

Esta aplicación se la puede descargar de Google Play e instalar en teléfonos que tienen sistema operativo Android.

Mediante esta aplicación se puede medir la vibración de movimientos telúricos usando los sensores que posee el dispositivo.

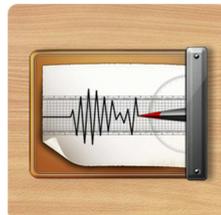


Fig. II. 32 Vibrómetro

Fuente: Google Play

➤ HMI

DEFINICIÓN

Interfaz Usuario Maquina (Human Machine Interface), se usan para lograr una comunicación entre el usuario y una máquina, para monitorear, supervisar y controlar equipos o maquinaria industrial.

En estas aplicaciones existen muchos elementos que la conforman, tales como botones, controladores, menús, indicadores, etc.

De acuerdo a la forma en la que interactúa con el usuario con el HMI se la puede clasificar de tres tipos:

- **Interfaz alfanumérica:** todos los comandos de interacción son mediante texto.
- **Interfaz gráfica de usuario:** los comandos se los realiza de manera gráfica.
- **Interfaz táctil:** sistemas touch los mismos que tiene paneles que permiten interactuar al usuario.

APLICACIÓN

Un sistema HMI se lo puede emplear en cualquier parte y situación de la empresa, en la que sea necesario tener monitoreo y control en tiempo real. Además si se desea monitorear todo un proceso o planta de manera centralizada.

VENTAJAS

- Monitoreo en tiempo real.
- Facilidad para operar o monitorear un proceso.
- Fácil manejo, adquisición y almacenamiento de datos.
- Sistemas escalables.
- Fácil mantenimiento.

DESVENTAJAS

- Para modificar el HMI, se requiere de personal calificado.
- De acuerdo al programa a utilizarse el costo de la respectiva licencia del software.
- Conocimientos básicos de sistemas HMI.

CAPITULO III

DISEÑO E IMPLEMENTACIÓN DEL HARDWARE

3.1.DISEÑO DEL TABLERO DE CONTROL DE UNA MAQUINA AGITADORA MEDIANTE SIEMENS LOGO!

- **Fuente de alimentación**

El circuito a montarse no necesita de una fuente de alimentación específica, ya que el dispositivo de control se puede alimentar a 110V de corriente alterna, eso implica que se lo puede conectar a cualquier toma de la industria.

- **Elemento de protección**

Se emplea un relé termomagnético, que permite proteger al circuito de cualquier falla por sobre voltaje o corriente. Y de manera adicional protege a los usuarios.

Las características más relevantes de este dispositivo son:

Tabla III. I Datos Relé Termomagnético Para LOGO!

Tipo de Montaje	Riel DIN
Número de Polos	2
Tensión Nominal	110-220V
Corriente Máxima	10 A



Fig. III. 1Relé Termomagnético LOGO! 230 RC

Fuente: Autor

➤ **Siemens LOGO! 230 RC**

Uso de las entradas:

Tabla III. II Detalle de Entradas LOGO! 230RC

ENTRADA DEL LOGO!	DESCRIPCIÓN EN EL PANEL
I 1	Automático
I 2	Encendido Motor 1
I 3	Apagado Motor 1
I 4	Encendido Motor 2
I 5	Apagado Motor 2
I 6	Encendido Motor 3
I 7	Apagado Motor 3

Uso de las salidas:

Tabla III. III Detalle de Salidas LOGO! 230RC

SALIDA DEL LOGO!	DESCRIPCIÓN EN EL PANEL
Q 1	Activa motor 1
Q 2	Activa motor 2
Q 3	Activa motor 3

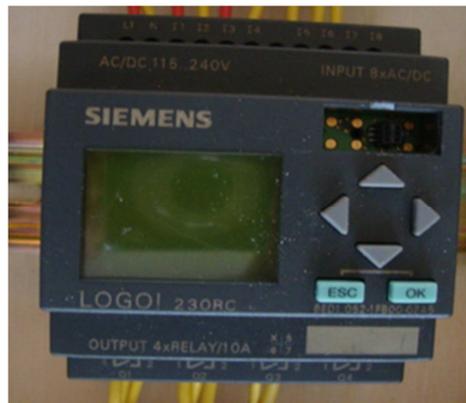


Fig. III. 2 SIEMENS LOGO! 230 RC

Fuente: Autor

- **Accionadores**

Se usan 3 switch de 3 posiciones y 1 switch de dos posiciones.



Fig. III. 3 Instalación Switch

Fuente: Autor

- **Actuadores**

Se usa tres focos que se encienden a 110V, los mismos que simulan la activación de la bobina que poseen los guardamotores, a los cuales se encuentran conectados los motores.



Fig. III. 4 Actuadores Del LOGO!

Fuente: Autor

- **Cable.**

El cable que se empleó para realizar todas las conexiones necesarias es: cable flexible N18, multifilar.



Fig. III. 5 Cable Multifilar

Fuente: Autor

- **Borneras de conexión por tornillo.**

Se las emplea para realizar las conexiones que necesitan un punto en común, estas borneras se las usa sobre riel DIN, las caras se encuentran cubiertas por material aislante como el thermoplast.



Fig. III. 6 Borneras De Conexión LOGO! 230RC

Fuente: Autor

- **Canaleta**

La canaleta permite colocar los cables de manera correcta y que sean llevados de manera ordenada y adecuada, para evitar que existan cables sobrepuestos o que se crucen de manera indebida.



Fig. III. 7 Canaleta Montaje LOGO! 230 RC

Fuente: Autor

- **Riel DIN**

El riel DIN sirve para dar soporte a todos los elementos que se colocan en el tablero, sobre este objeto se encuentra colocado el relé termomagnético, borneras y el Siemens LOGO 230RC!



Fig. III. 8 Riel DIN LOGO! 230 RC

Fuente: Autor

El siguiente esquema muestra la conexión que se realizó, indicando los accionamientos y las salidas que posee el LOGO!

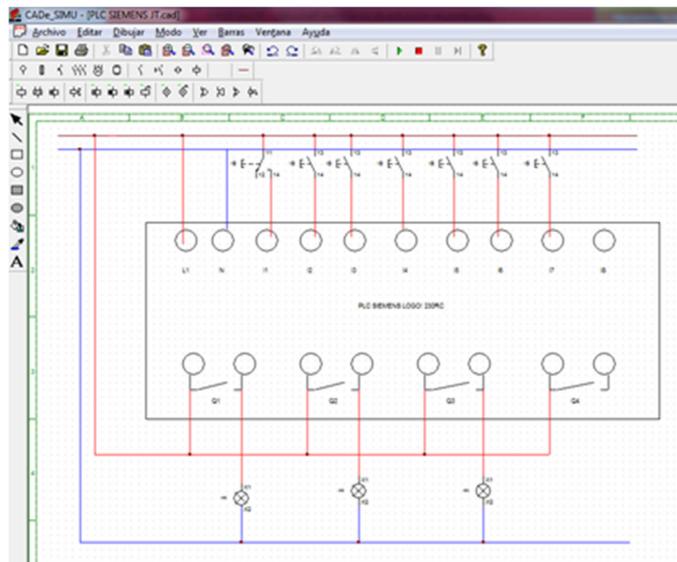


Fig. III. 9 Esquema De Conexión LOGO! 230 RC

Fuente: Autor

A continuación se muestra el tablero terminado.



Fig. III. 10 Tablero LOGO! 230 RC

Fuente: Autor

3.2.DISEÑO DEL TABLERO DE CONTROL DE UNA MAQUINA AGITADORA MEDIANTE ARDUINO MEGA 2560

- **Fuente de alimentación**

El diseño para este circuito requiere de dos alimentaciones diferentes.

- La fuente de alimentación para Arduino.

Arduino al estar conformado por un microcontrolador el voltaje de alimentación muy bajo en niveles de voltaje lógico. Arduino puede ser alimentado de dos maneras distintas en las cuales el voltaje que se puede suministrar al dispositivo se encuentra en el rango de 6-20V.

Para suministrar este voltaje se emplea un cargador para teléfono, las características de este se detallan a continuación:

Tabla III. IV Datos de Placa de la Fuente Arduino

DETALLE	VALOR
Voltaje de entrada	AC 120V – 60Hz
Consumo	2.4W
Voltaje de salida	DC 7.5V
Corriente de salida	150 mA



Fig. III. 11 Fuente De Poder Arduino

Fuente: Autor

- La fuente de alimentación para la etapa de potencia.
El voltaje para la etapa de potencia es de 110V. El mismo que se conecta o se obtiene de la red eléctrica común.

- **Elemento de protección**

Se emplea un relé termomagnético, que permite proteger al circuito de cualquier falla por sobre voltaje o corriente. Y de manera adicional protege a los usuarios.

Las características más relevantes de este dispositivo son:

Tabla III. V Datos Relé Termomagnético Para Arduino

Tipo de Montaje	Riel DIN
Número de Polos	1
Tensión Nominal	110-220V
Corriente Máxima	10 A.



Fig. III. 12 Relé Termomagnético Arduino

Fuente: Autor

➤ **Arduino Mega 2560**

Uso de las entradas:

Tabla III. VI Detalle de Entradas Arduino Mega 2560

ENTRADAS DEL ARDUINO	DESCRIPCIÓN EN EL PANEL
Pin digital 3	Automático
Pin digital 2	Manual
Pin digital 5	Encendido Motor 1
Pin digital 6	Apagado Motor 1
Pin digital 8	Encendido Motor 2
Pin digital 9	Apagado Motor 2
Pin digital 11	Encendido Motor 3
Pin digital 12	Apagado Motor 3

Uso de las salidas:

Tabla III. VII Detalle de Salidas Arduino Mega 2560

SALIDAS DEL ARDUINO	DESCRIPCIÓN EN EL PANEL
Pin digital 38	Activa relé 1
Pin digital 42	Activa relé 2
Pin digital 46	Activa relé 3
Pin digital 50	Activa relé 4

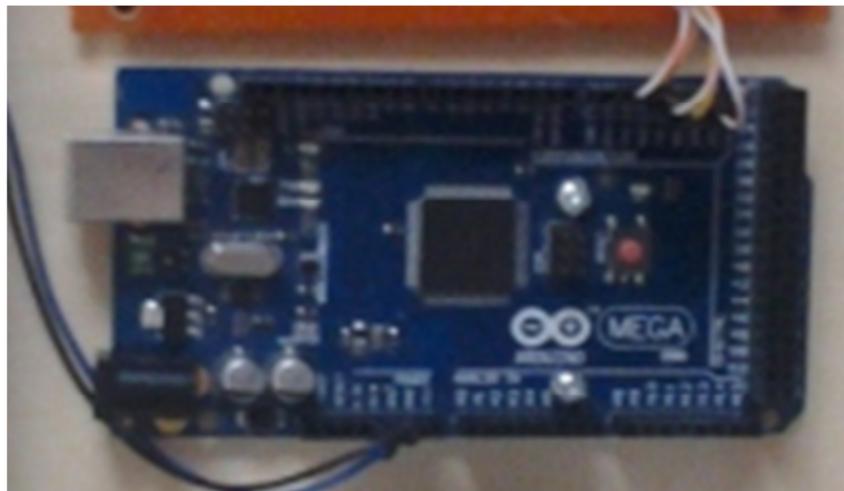


Fig. III. 13 Arduino Mega Para El Agitador

Fuente: Autor

- **Etapa de Potencia**

Para poder utilizar la etapa de potencia se hace necesario hacer uso de relés, los mismos que permiten manejar voltajes elevados. Para la etapa de potencia se utiliza:

- Resistencias de 4,7K Ω .
- Transistores 3904 (NPN).
- Relés de 10 A.

La simulación se la realizo en Proteus, y se la muestra a continuación:

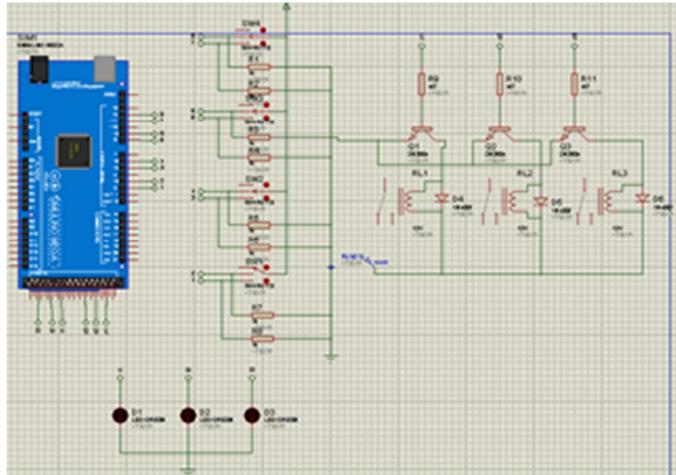


Fig. III. 14 Simulación Tablero De Control Arduino
Fuente: Autor

Y el diseño del circuito se presenta a continuación:

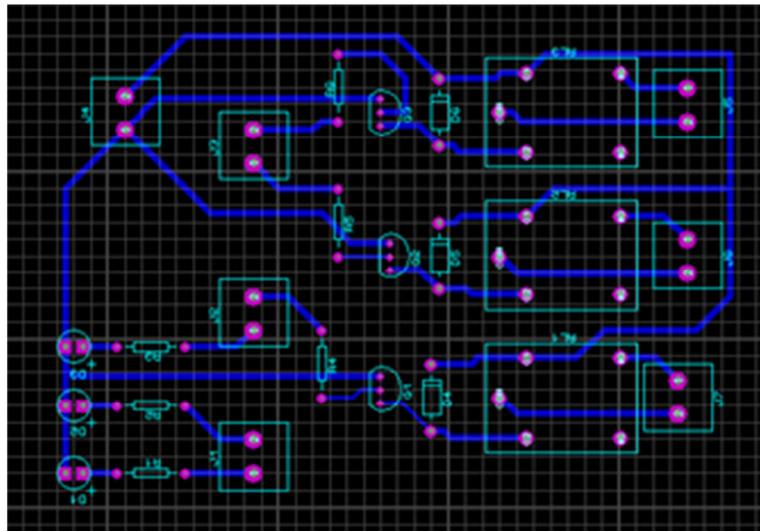


Fig. III. 15 Diseño De Placa En Ares
Fuente: Autor

Diseño implementado de la placa que se emplea para la etapa de potencia:

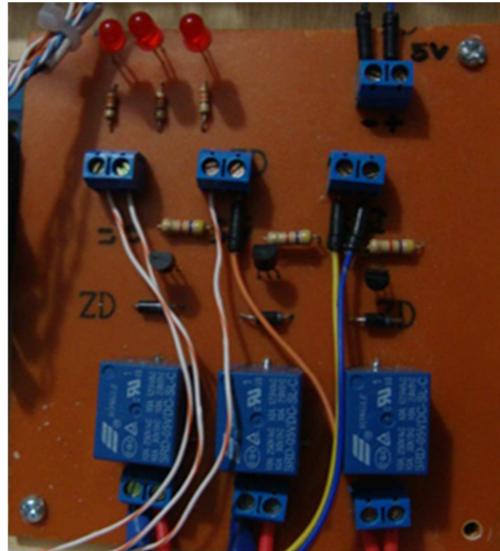


Fig. III. 16 Placa De Potencia Terminada
Fuente: Autor

- **Accionadores**

Se usan 4 switch de 3 posiciones.

Se adiciona un switch de tres posiciones ya que por motivos de programación es necesario hacer uso de un contacto más para realizar el paro de los actuadores, por cualquier posible falla.

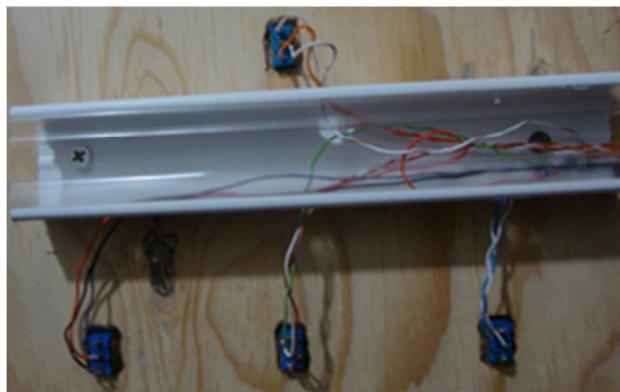


Fig. III. 17 Instalación Accionadores Tablero Arduino
Fuente: Autor

- **Actuadores**

Se usa tres focos que se encienden a 110V, los mismos que simulan la activación de la bobina que poseen los guardamotores, a los cuales se encuentran conectados los motores.



Fig. III. 18 Actuadores Tablero Arduino

Fuente: Autor

- **Cable.**

El cable que se empleó para realizar todas las conexiones necesarias es: UTP multifilar.



Fig. III. 19 Cable UTP Multifilar

Fuente: Autor

- **Borneras de conexión por tornillo.**

Se las emplea para realizar las conexiones que necesitan un punto en común, estas borneras se las usa sobre riel DIN, las caras se encuentran cubiertas por material aislante como el thermoplast.



Fig. III. 20 Bornera Tablero Arduino

Fuente: Autor

- **Canaleta**

La canaleta permite colocar los cables de manera correcta y que sean llevados de manera ordenada y adecuada, para evitar que existan cables sobrepuestos o que se crucen de manera indebida.



Fig. III. 21 Canaleta Tablero Arduino

Fuente: Autor

- **Riel DIN**

El riel DIN sirve para dar soporte a todos los elementos que se colocan en el tablero, sobre este objeto se encuentra colocado el relé termomagnético, y las borneras.



Fig. III. 22 Riel DIN Tablero Arduino

Fuente: Autor

- **Caja metálica**

Se crea una caja metálica para evitar posibles fallas en el funcionamiento de Arduino, la misma evitara que se vea afectado por campos electromagnéticos que se pueden producir alrededor de este dispositivo.

Dimensiones del elemento:

Tabla III. VIII Dimensiones Caja Protectora de Arduino

<i>Descripción</i>	<i>Dimensión</i>
Largo	11.0 cm
Ancho	9.0 cm
Alto	5.5ccm

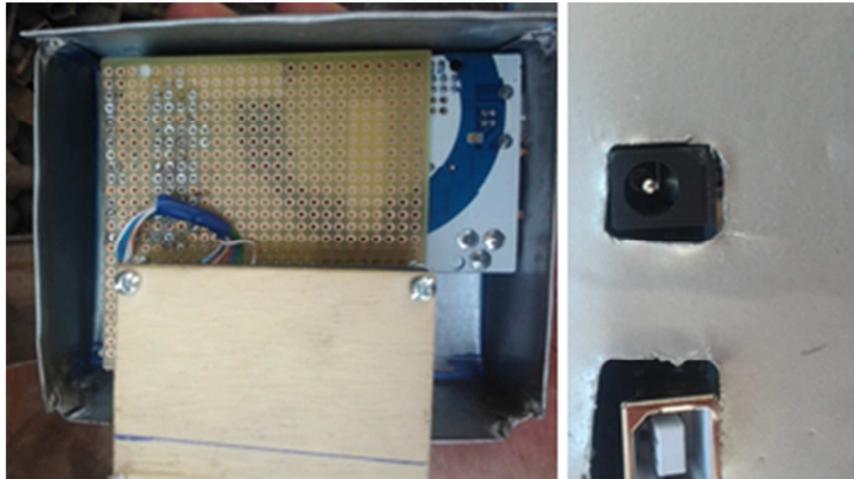


Fig. III. 23 Caja Protectora Arduino

Fuente: Autor

El diseño culminado del control mediante Arduino es el siguiente:



Fig. III. 24 Tablero Arduino Terminado

Fuente: Autor

3.3. DISEÑO DE UN CONTROL DE TEMPERATURA PARA LA COCCION DE BALDOSAS MEDIANTE ARDUINO Y MONITOREADO POR MEDIO DE UN HMI.

- **AD595**

Este dispositivo es un instrumento de amplificación completo y un compensador de unión fría para termocuplas.

Este dispositivo permite linealizar la salida de la termocupla tipo K, de este modo se puede adquirir los datos de manera lineal, para ser almacenados y procesados en el HMI.

El pin N°12 del AD595 se encuentra conectado a la alarma que posee dicho dispositivo en caso de que falle el termopar.



Fig. III. 25 AD595 Termocupla K

Fuente: Autor

- **RELES 5VDC**

Realizan la conexión al voltaje de trabajo para los elementos de conexión, el voltaje de activación es de 5 VDC lo que facilita su activación por los voltajes manejados por Arduino, estos elementos pueden trabajar en un amplio rango de trabajo, el mismo que se detalla a continuación:

Tabla III. IX Rango de Trabajo Relés

VOLTAJE DE ALIMENTACION	AMPERIOS
250 VAC	10 A
125 VAC	10 A
30 VDC	10 A
28 VDC	10 A



Fig. III. 26 Relé Etapa de Potencia

Fuente: Autor

- **TERMOCUPLA TIPO K**

El rango de operación se encuentra entre los -200°C a $+1372^{\circ}\text{C}$ con una variación respecto a la sensibilidad de $41 \mu\text{V}/^{\circ}\text{C}$.

Debido a su característica constructiva son muy resistentes a altas temperaturas y a la oxidación.



Fig. III. 27 Termopar Tipo K

Fuente: Autor

- **INDICADORES LUMÍNICOS (LED's).**

El circuito se encuentra diseñado con cinco indicadores luminosos, los mismos que indican el grado de temperatura que se encuentra sensando el termopar ubicado en este sistema, trabajan de manera simultánea con los indicadores ubicados en el panel frontal del HMI.



Fig. III. 28 LEDs Sistema de Control De Temperatura

Fuente: Autor

Se encuentran de la siguiente manera:

Tabla III. X Descripción Pines De Salida Arduino

NUMERO DE PIN	DESCRIPCION DEL PIN
Pin digital 22	T. < a 0° C
Pin digital 23	T. BAJA
Pin digital 24	T. IDEAL
Pin digital 25	T. ALTA
Pin digital 26	T.EXCESIVA

- **BORNERAS DE CONEXIÓN**

Están ubicadas 4 borneras las mismas que permiten realizar la conexión a 110V de manera sencilla y externa, lo que facilita la manipulación del sistema.



Fig. III. 29 Borneras Sistema de Control De Temperatura

Fuente: Autor

- **ARDUINO**

Se usaron varios pines, entre los cuales se debe destacar: 8 pines digitales, 1 pin analógico. El número de pin usado y la descripción se detalla a continuación:

Tabla III. XI Pines Empleados En Arduino

NUMERO DE PIN	DESCRIPCION DEL PIN
Pin digital 22	T. < a 0° C
Pin digital 23	T. BAJA
Pin digital 24	T. IDEAL
Pin digital 25	T. ALTA
Pin digital 26	T.EXCESIVA
Pin digital 49	Activa relé del Chispero
Pin digital 50	Activa relé del Ventilador
Pin digital 51	Activa relé de la Sirena
Pin analógico 2	Lectura de la señal de termopar

En la fig.III.30 se muestra La conexión física de Arduino para el control de temperatura.

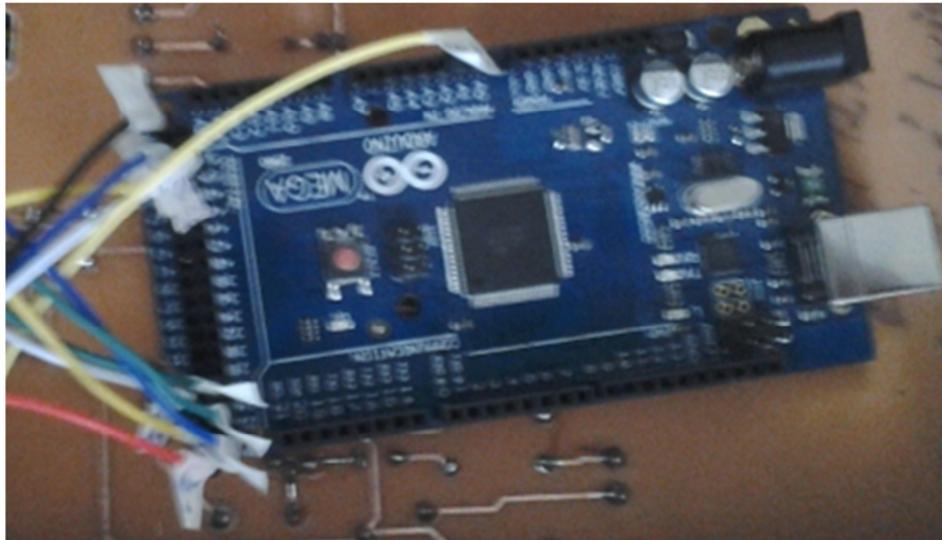


Fig. III. 30 Arduino Mega Control de Temperatura

Fuente: Autor

- Simulación del circuito en Proteus

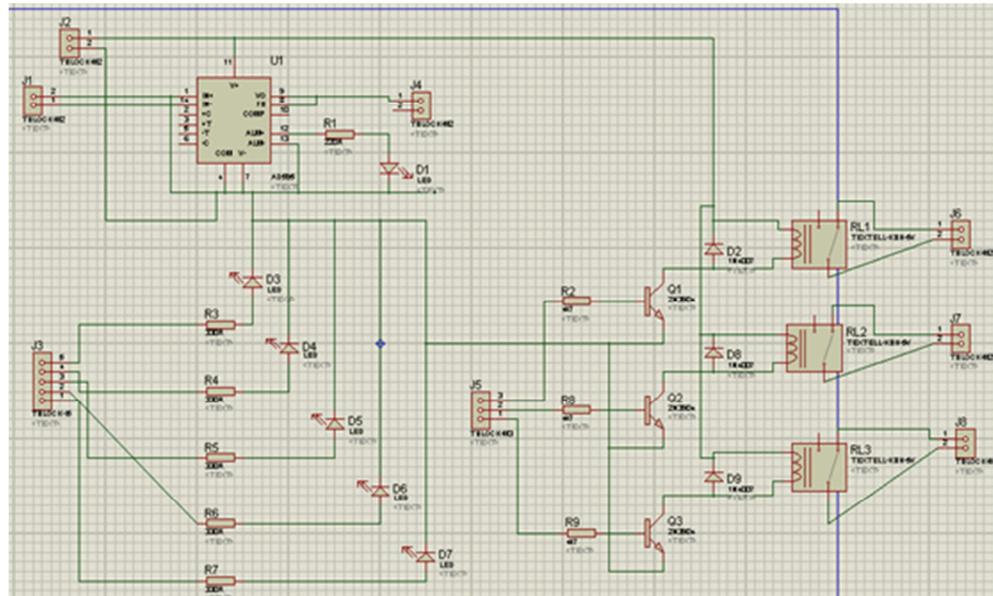


Fig. III. 31 Simulación Control de Temperatura

Fuente: Autor

- Diseño del circuito en Ares

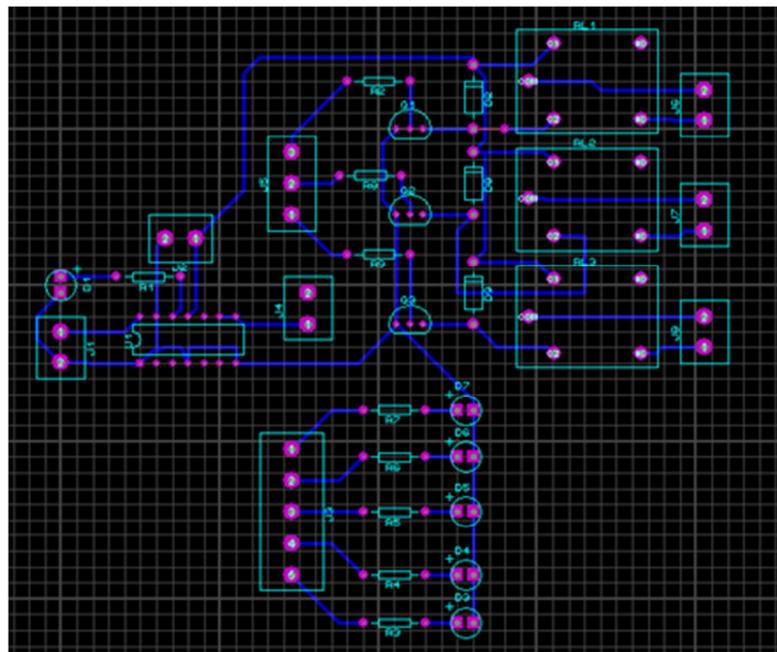


Fig. III. 32 Diseño de Placa en Ares

Fuente: Autor

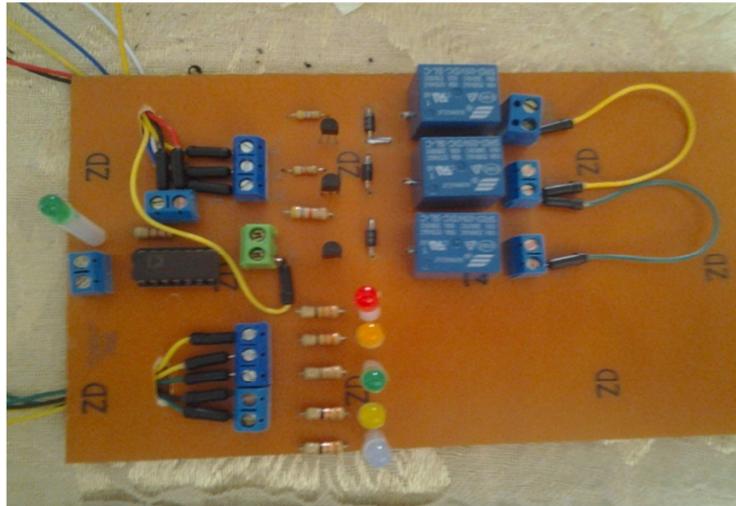


Fig. III. 33 Etapa de Potencia Control de Temperatura

Fuente: Autor

- **CAJA**

El sistema fue instalado en una caja de plástico, para proveer de protección a todos los elementos antes mencionados.



Fig. III. 34 Controlador Arduino Control de Temperatura

Fuente: Autor

CAPITULO IV

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

4.1. LOGO! SOFT COMFORT

Para realizar un programa en LOGO!SoftComfort se los puede crear en tres formatos, cada uno contiene formas y características específicas.

Diagrama de funciones (FUP): es aquel donde se debe destacar el uso de símbolos gráficos del algebra booleana para realizar la lógica de programación. Se recomienda este tipo de programación para personas que poseen mucho conocimiento acerca de algebra booleana.

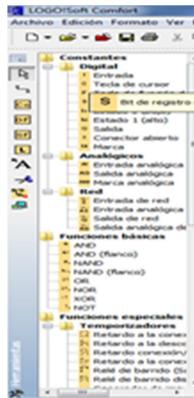


Fig. IV. 1 Panel de Programación FUP

Fuente: Autor

Diagrama UDF: son programas previamente configurados por el programador, estos bloques se los puede adicionar o agregar a los programas existentes.



Fig. IV. 2 Panel de Programación UDF

Fuente: Autor

Esquema de contactos (KOP): se lo emplea por ser similar a los esquemas de circuitos, se emplean contactos normalmente abiertos, cerrados y elementos que generalmente se emplean en un circuito eléctrico, se basa en lenguaje ladder.

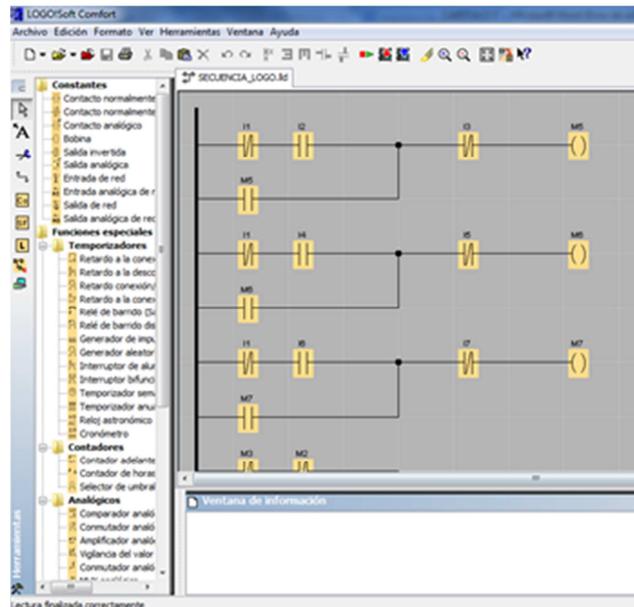


Fig. IV. 3 Panel de Programación KOP

Fuente: Autor

4.1.1. ELEMENTOS EMPLEADOS PARA LA PROGRAMACION KOP

- **CONTACTO NORMALMENTE ABIERTO (NA)**

Los contactos normalmente abiertos representan los pulsadores o switch que dan las señales al LOGO!. Estos permanecen abiertos mientras no exista una señal que haga cambiar su estado.

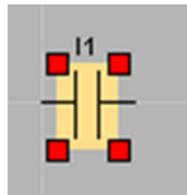
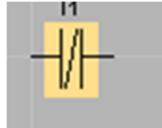


Fig. IV. 4Contacto NA. LOGO! SOFT COMFORT

Fuente: logo! Soft comfort

- **CONTACTO NORMALMENTE CERRADO (NC)**

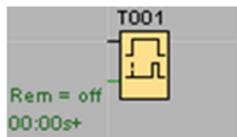
Los contactos normalmente cerrados funcionan a la inversa de uno abierto, estos se emplean dentro de la programación para realizar un mejor control de las condiciones de trabajo del programa, con el uso de estos contactos se logra ejecutar solo una de las rutinas.



*Fig. IV. 5 Contacto NC LOGO! SOFT COMFORT
FUENTE: LOGO! SOFT COMFORT*

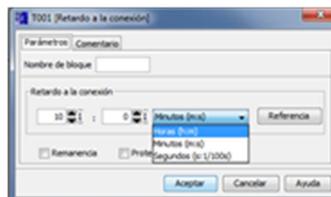
- **TEMPORIZADORES**

Los temporizadores son empleados para ejecutar las sentencias por periodos de tiempo, definidos por el programador, estos pueden variar de acuerdo a las necesidades que se tengan en el programa, se puede configurar en segundos, minutos y horas.



*Fig. IV. 6 Temporizador LOGO! SOFT COMFORT
Fuente: logo! Soft comfort*

Para configurar el tiempo se debe dar doble click sobre este elemento, y se desplegara una nueva ventana como la siguiente:



*Fig. IV. 7 Pantalla Configuración Temporizador
Fuente: logo! Soft comfort*

- **MEMORIAS**

Las memorias se las usa para realizar enclavamientos, los mismos que permiten mantener un determinado estado hasta que exista una señal que cambie el mismo.

Ayudan a mejorar el esquema de programación y se puede activar una sola salida utilizando varias memorias a la vez.



Fig. IV. 8 Memorias LOGO! SOFT COMFORT

Fuente: logo! Soft comfort

- **BOBINAS**

Representan las salidas del LOGO!, las mismas que se activan de acuerdo a las necesidades que posea el sistema.

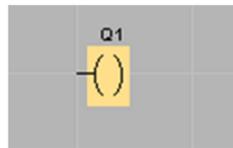


Fig. IV. 9 Bobina LOGO! SOFT COMFORT

Fuente: logo! Soft comfort

4.2. ARDUINO

Para la programación en Arduino se debe tener en cuenta las principales características:

- **Palabras reservadas.**

Es muy importante tener en cuenta las palabras reservadas con las que trabaja el IDE de Arduino, las palabras deben ser escritas de forma correcta respetando las mayúsculas de las minúsculas, ya que si no se las escribe de esa manera se presentaran problemas al momento de la compilación.

Las palabras empleadas en el programa son las siguientes:

- *Int*: sirve para declarar una variable de tipo entera, y también para asignar un nombre a los pines digitales, esto facilita la escritura del programa.
- *attachInterrupt*: se usa para hacer referencia a la interrupción que se generará en el programa. Se debe escribir de la siguiente forma:
 - `attachInterrupt(0,paro,CHANGE);`
donde: 0 es el pin digital de la interrupción, paro es el nombre de la función a la cual se dirigirá el programa cuando exista una interrupción, y CHANGE que es el tipo de interrupción.
- *pinMode*: esta palabra reservada se usa para declarar a un pin digital como entrada o salida. Estas declaraciones se las realiza dentro de la función void setup.
 - `pinMode(pin, INPUT/OUTPUT)`
- *INPUT*: se la emplea para declarar a un pin digital como entrada, este pin mantendrá esta característica durante toda la ejecución del programa.
- *OUTPUT*: se la emplea cuando se desea usar a un pin digital como salida, este pin será considerado como una salida durante toda la ejecución del programa.
- *digitalRead*: este comando obtiene el valor de que existe en el pin, este puede ser un 1 lógico o un 0 lógico. La forma en la que debe ser escrito es la siguiente:
 - `valor=digitalRead(pin);`
donde: el número del pin digital debe ser escrito en el lugar de pin.
- *digitalWrite*: permite escribir un valor en dicho pin, este puede ser escrito como HIGH o LOW dependiendo del valor con el que se desea trabajar dicha salida. La sintaxis debe ser la siguiente:
 - `digitalWrite(pin, HIGH/LOW);`

- **Voidloop**

Dentro de esta función se encuentra el programa en sí, donde constan todas las condiciones necesarias para realizar la función específica para la cual se emplea Arduino. Se debe tener muy en cuenta todas las llaves que se abren, ya que no cerrar una llave puede generar errores al compilar el programa.



Fig. IV. 11 Void Loop IDE Arduino

Fuente: Autor

- **Interrupción**

La interrupción detendrá la ejecución del programa, las interrupciones no deben contener tiempo de retardo ya que generan problemas de compilación.

El gráfico representa la interrupción existente en el programa:

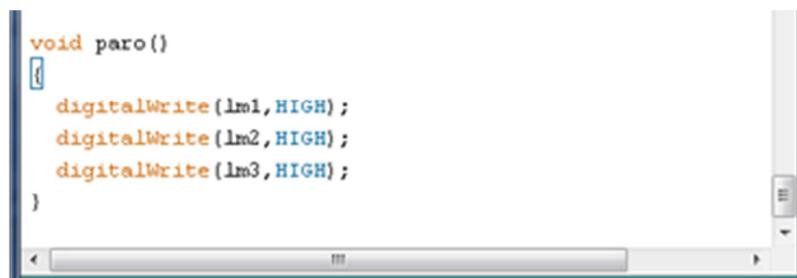


Fig. IV. 12 Interrupción IDE Arduino

Fuente: Autor

- **Estructura de control empleada**

Se empleó la estructura IF-ELSE ya que permite un control adecuado del programa sin conllevar mucha dificultad. Se emplearon varios IF anidados, para cumplir todas las posibles condiciones de trabajo.

- **Cálculo de tiempo**

Es necesario transformar el tiempo en el que se desea trabajar a milisegundos, ya que estas son las unidades de tiempo en las que trabaja Arduino.

Fue necesario emplear las siguientes fórmulas para el cálculo del tiempo en milisegundos.

Para horas:

$$mseg = HORAx60x60x1000$$

Para minutos:

$$mseg = MINx60x1000$$

Para la respectiva implementación del código fue necesaria la creación de un diagrama de flujo, el mismo que facilita la comprensión y escritura del programa.

4.3. LABVIEW-ARDUINO

La interfaz Hombre-Máquina o también conocida como HMI (Human Machine Interface), es un elemento indispensable para realizar un sistema de control complejo, ya que permite realizar tareas muy utilizadas en la industria tales como: monitoreo, supervisión, adquisición, almacenamiento y procesamiento de datos.

Nos permite llevar un control más oportuno y preciso de todos los elementos que se encuentran conectados en el sistema.

El HMI de este proyecto de investigación tiene como finalidad el control de temperatura dentro de un horno de cocción industrial.

4.3.1. PANEL FRONTAL

El panel frontal es la interface de comunicación entre el usuario y el sistema a ser supervisado, monitoreado o manipulado.

En el panel frontal se puede ubicar indicadores y controladores, los cuales representan las entradas y salidas del Instrumento Virtual. Los controladores representan todos los dispositivos de entrada que están conectados al sistema y son los encargados de transmitir los datos al diagrama de bloques, en tanto los indicadores simulan a todos los dispositivos de salida.



Fig. IV. 13 Panel Frontal Arduino-LabView

Fuente: Autor

4.3.2. DIAGRAMA DE BLOQUES

En el diagrama de bloques constan el programa en sí, en esta ventana se coloca todos los elementos necesarios para realizar la programación de acuerdo a las necesidades y características que posee el sistema.

En esta ventana permite controlar todos los procesos que se define en el panel frontal, considerando los indicadores y controladores, para realizar un correcto procesamiento de la información para que el trabajo en tiempo real sea exacto y correcto.

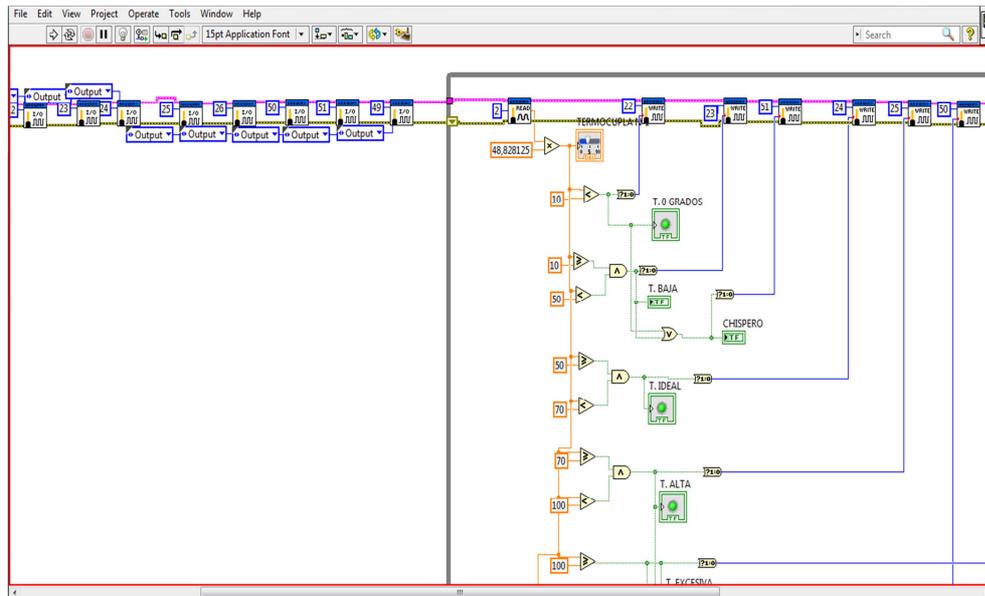


Fig. IV. 14 Diagrama de Bloques Arduino-LabView

Fuente: Autor

4.4. CONEXIÓN ARDUINO- LABVIEW

Arduino es una plataforma de software y hardware libre, la cual se encuentra basada en un microcontrolador ATmega 2560. La placa posee 54 pines digitales entrada/salida, de los cuales 14 pines pueden ser utilizados como salidas PWM y 16 entradas analógicas.

Las entradas digitales pueden el voltaje de operación es de 5 voltios, con una corriente máxima de 40Ma.

LabView es un software que posee la capacidad de automatizar procesos desde un ordenador, por lo cual este software es capaz de manejar instrumentos que tienen características diferentes, entre las cuales se puede destacar: GPIB, serial, módulos PXI, USB, Ethernet, puerto paralelo, etc.

Para realizar la comunicación entre LabView y Arduino se usa un protocolo de comunicación serial, para usar este tipo de comunicación se debe tener en cuenta los siguientes parámetros:

- **Velocidad en Baudios:** indica la velocidad con la que se transmite los datos entre el instrumento (Arduino Mega) que se encuentra conectado al puerto serial.
- **El número de Bits de datos:** los mismos que son transmitidos en valores lógicos como 0 y 1 para leer los valores positivos y negativos respetivamente. Considerando que los datos se transmiten de manera inversa y hacia atrás.
- **Bit de Paridad:** que permite verificar algún error que se pueda producir durante la transmisión de la trama.
- **Bit de parada:** indican cuando se ha terminado de transmitir una trama, y cuando empieza otra.

Los puertos USB o Ethernet son transformados a puertos seriales asincrónicos que sirven para realizar la respectiva comunicación con los instrumentos seriales.

4.4.1. INTERFAZ PARA ARDUINO. (INTERFACE FOR ARDUINO TOOLKIT)

Es necesario realizar una la instalación del software de National Instruments (NI) que permite realizar o establecer una interfaz de manera sencilla con el microcontrolador instalado en la placa del Arduino.

Interface for Arduino Toolkit, permite controlar y adquirir todas las señales de los elementos que se encuentren conectados a la placa de Arduino. Para el procesamiento de todos los datos obtenidos desde el microcontrolador se deberá hacer uso de las librerías existentes en LabView para que sean datos lo suficientemente correctos.

Con esta librería la placa de Arduino se la utiliza como una tarjeta de adquisición de datos, teniendo en cuenta que la placa interactúa con el VI por medio de la una comunicación serial.

Para lograr con este objetivo se debe seguir los siguientes pasos:

- **Instalar el software LabView 2009 o superior.** Para esto existen dos formas obtener el CD de instalación o instalar una versión de prueba que se puede descargar desde la página principal de LabView o desde este link: <http://www.ni.com/download-labview/esa/>
- **Instalar los drivers de NI-VISA.** Para ello es necesario actualizar el NI-VISA a la última versión o descargarse los drivers desde el siguiente link: <https://decibel.ni.com/content/docs/DOC-15971>.

Para descargar dichos drivers se deberá crear una cuenta en NI.

- **Instalar JKI VI Package Manager.** Para instalar este software se lo puede hacer con una simple búsqueda en un navegador de preferencia y se escoge descargar la versión gratuita, una vez finalizada la descarga se ejecuta el programa.
Para evitar posibles errores al descargar este programa se recomienda que se la versión más actual del mismo.
- **Instalar la Interface de Arduino para LabView.**
Se debe seguir los siguientes pasos:

- Ejecutar el VI PACKAGE MANAGER

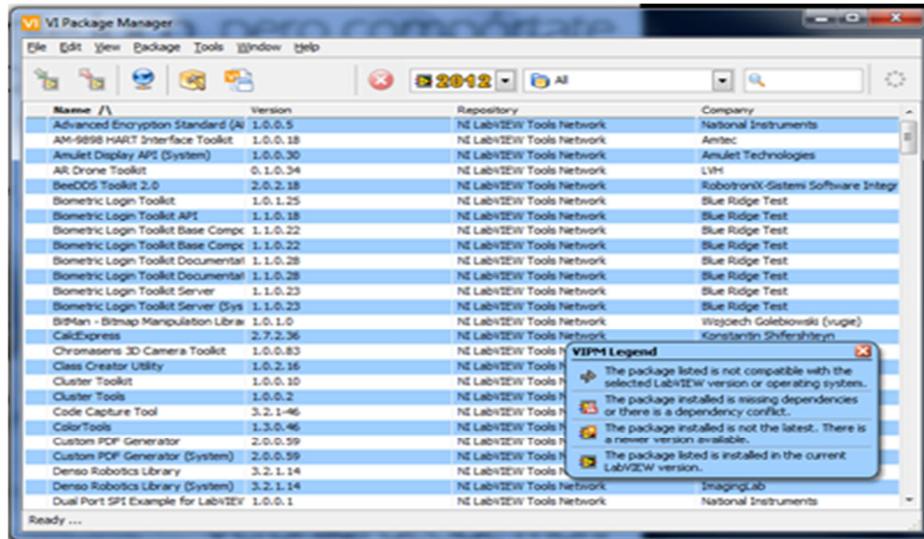


Fig. IV. 15 Pantalla Principal Package Manager

Fuente: Autor

- En el buscador escribimos la palabra Arduino o Interface for Arduino.

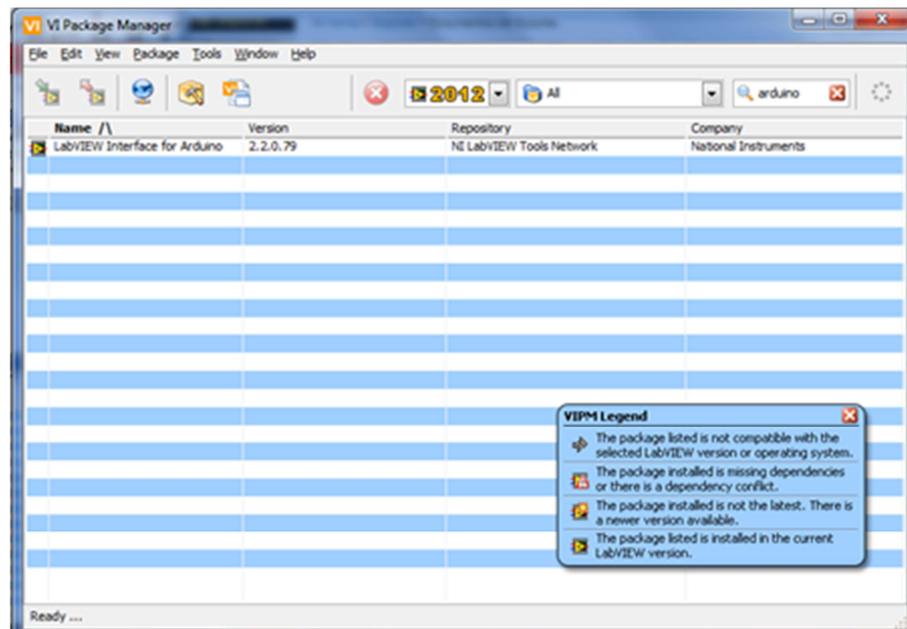


Fig. IV. 16 ToolKit Para LabView

Fuente: Autor

- Se selecciona el único elemento que coincide con la búsqueda y se da click en InstallPackage, que se encuentra ubicado en la parte superior izquierda de la ventana.

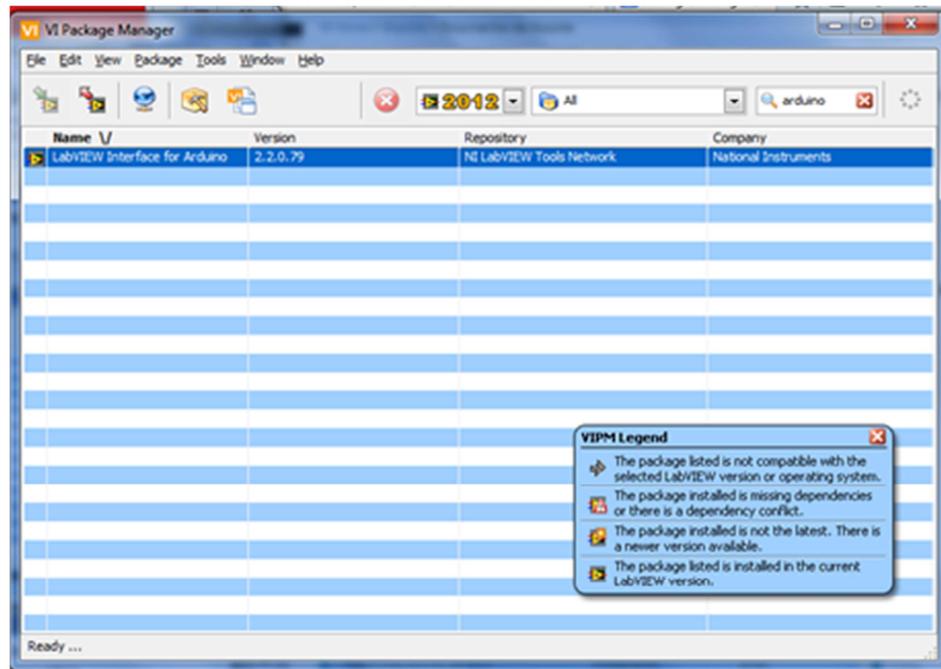


Fig. IV. 17 Instalación ToolKit Para LabView

Fuente: Autor

- Cargar el sketch correspondiente en la placa Arduino. Al instalar la interface de Arduino para LabView se crea una carpeta con un Sketch que se debe cargar a la placa de Arduino, para establecer la conexión con el VI.

C:\Program Files (x86)\National Instruments\LabVIEW 2012\vi.lib\LabVIEW Interface for Arduino\Firmware\LIFA Base

La dirección anteriormente descrita contiene el programa que se debe cargar a la tarjeta Arduino.

4.4.2. BLOQUES PRINCIPALES DE CONEXIÓN PALETA DE CONEXIÓN LABVIEW

ARDUINO INIT

Este bloque inicializa la conexión entre un Arduino y LabView, Arduino debe estar ejecutando la interface de LabView, previamente cargada en el elemento.

En este bloque se deben definir los siguientes parámetros:

- ✓ **VISA Resource:** se debe colocar el puerto al que se encuentra conectado el Arduino. Por ejemplo COM 5.
- ✓ **BaudRate:** se debe definir la velocidad de transmisión de datos. Por ejemplo 115200 bits/seg.
- ✓ **BoardType:** se debe especificar el tipo de tarjeta Arduino con la que se realizará el sistema. LabView trabaja con tres modelos, los mismos que son: Uno, Dimuelanove, Mega 2560. Por ejemplo Mega 2560.
- ✓ **ConnectionType:** se debe indicar el tipo de comunicación con el cual se va a trabajar para la transmisión de datos, existen tres formas para lograr este objetivo las mismas que son: USB/serial, XBee, Bluetooth. Por ejemplo USB/ serial.

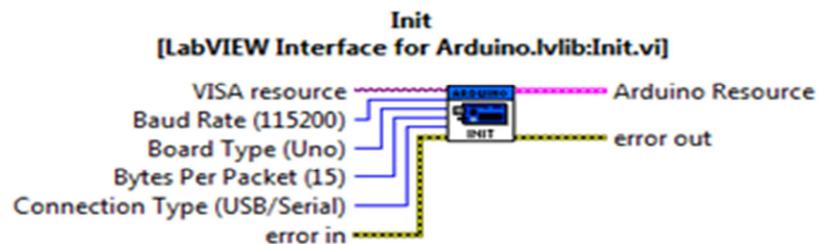


Fig. IV. 20 Bloque INIT

Fuente: LabView 2012

✚ ARDUINO CLOSE

Este bloque cierra todas las conexiones que se encuentren activas con Arduino. En este no hay que añadir ningún tipo de control, pero si es necesario conectar el recurso de Arduino.

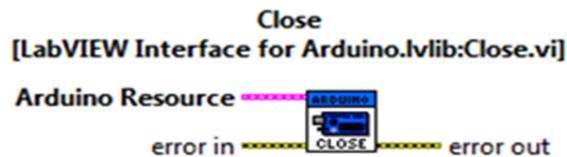


Fig. IV. 21 Bloque CLOSE

Fuente: LabView 2012

✚ ARDUINO LOW LEVEL

Dentro de esta paleta existen varios bloques, para el presente trabajo se utilizaron los siguientes:

➤ SET DIGITAL PIN MODE

Permite configurar cualquiera de los pines digitales como entrada o salida.

- **Digital I/O pin:** se debe colocar el número del pin, el cual será usado en el sistema.
- **Pin Mode:** permite configurar al pin seleccionado para que este cumpla una función de entrada o salida del sistema.

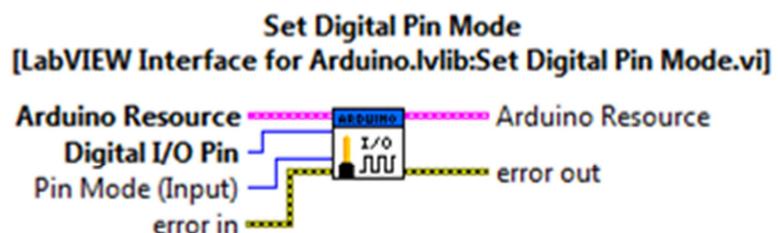


Fig. IV. 22 Bloque SET DIGITAL PIN MODE

Fuente: LabView 2012

➤ **ANALOG READ PIN**

Este bloque lee el voltaje analógico que se encuentra conectado a una de las entradas de la tarjeta, dependiendo del modelo de tarjeta variará el número de entradas analógicas.

- **Analog Input Pin:** se debe definir el número del pin al cual se encuentra conectado el valor a sensar.

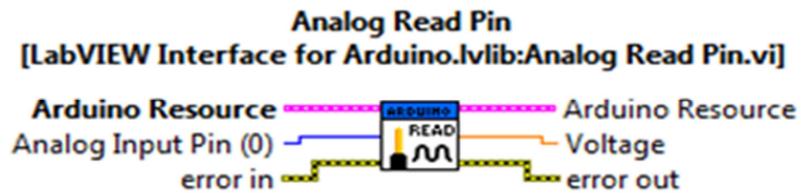


Fig. IV. 23 Bloque ANALOG READ PIN

Fuente: LabView 2012

➤ **DIGITAL WRITE PIN**

Escribe un valor específico sobre una salida específica que previamente debe ser declarada, se debe tener en cuenta los siguientes parámetros:

- **Digital I/O Pin:** se debe colocar el número del pin al cual se pretende escribir un valor.
- **Value:** se coloca el valor que se desea escribir en el pin si es 1 es un voltaje alto o si es 0 un voltaje bajo.

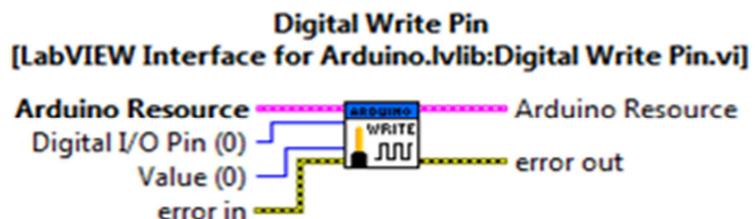


Fig. IV. 24 Bloque DIGITAL WRITE PIN

Fuente: LabView 2012

4.4.3. CONFIGURACIÓN NI MAX LABVIEW

Se debe seguir los siguientes pasos:

- ✚ Conectar la tarjeta Arduino y verificar cual es el puerto al que se encuentra conectada. Para lo cual es necesario ingresar al Administrador de equipos.

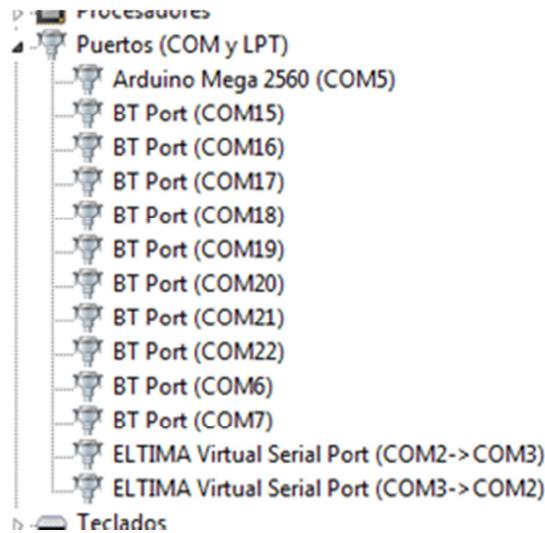


Fig. IV. 25 Administrador de Dispositivos Arduino

Fuente: Autor

- ✚ Abrir el NI MAX. Click sobre Dispositivos e Interfaces.

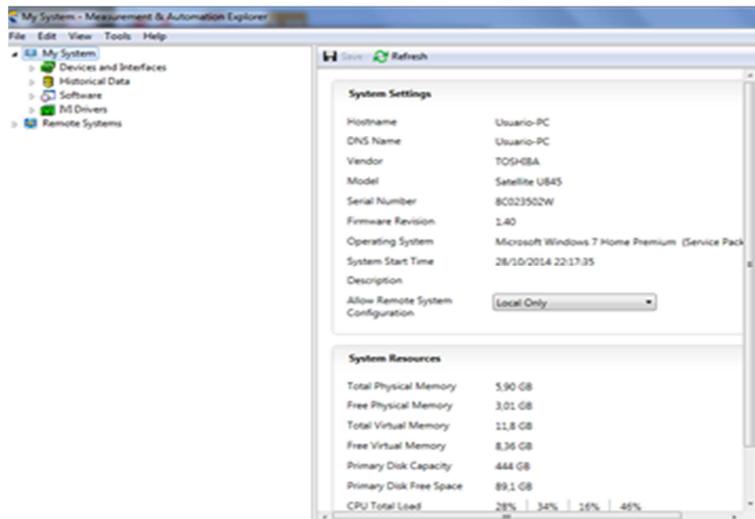


Fig. IV. 26D Dispositivos E Interfaces NI MAX

Fuente: Autor

- Se debe seleccionar el puerto en el cual está conectada la tarjeta de Arduino.

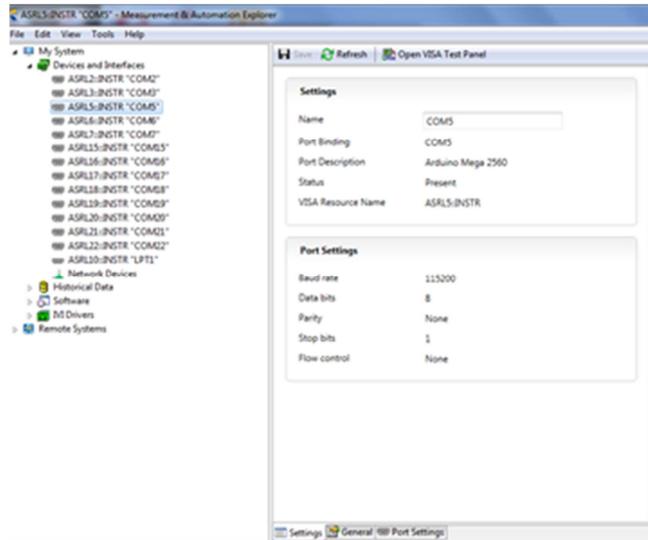


Fig. IV. 27 Configuración Arduino NI MAX

Fuente: Autor

- Para cambiar los parámetros de configuración se debe dar click sobre la pestaña que se encuentra en la parte inferior izquierda etiquetada como “Configuraciones del puerto”, y realizar todos los cambios necesarios y de acuerdo a las necesidades del usuario y del sistema.

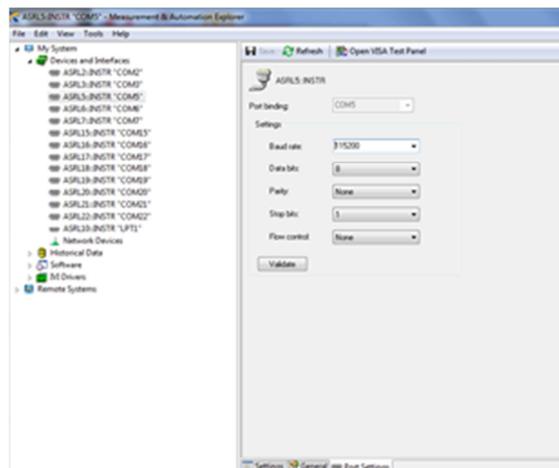


Fig. IV. 28 Configuración de Puerto Arduino NI MAX

Fuente: Autor

4.4.4. CREACION BASE DE DATOS

Una parte muy importante de un sistema de control es el almacenamiento de los datos adquiridos, para que estos sean analizados o comparados.

De manera adicional cabe destacar que la base de datos se almacenan todo cuanto sea necesario, como por ejemplo fecha, hora, valores de temperatura, presión, etc.

Se pueden crear diferentes tipos de bases de datos, de acuerdo a las necesidades y recursos del sistema.

Para la creación de esta base de datos se utiliza un recurso de las librerías de LabView. Se emplea un temporizador para adquirir los datos por intervalos de tiempo.

Se emplearon los siguientes elementos para crear la base:

- **Get Date/Time String:** se usa para convertir valores numéricos o valores de marca en una cadena de fecha y hora, de acuerdo a la zona horaria en la cual se encuentra configurado el ordenador.

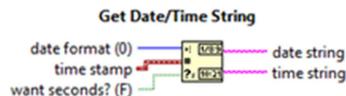


Fig. IV. 29 Datos de fecha Y Hora LabView

Fuente: LabView 2012

- **NumeroToFractionalString:** convierte el número a un formato fraccionario de punto flotante, y la salida de este bloque corresponde a una salida en formato cadena.

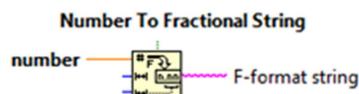


Fig. IV. 30 Transformar Dato a Cadena LabView

Fuente: LabView 2012

- **BuildArray**: se emplea para unir varias matrices o simplemente añade elementos a una matriz de n dimensiones.

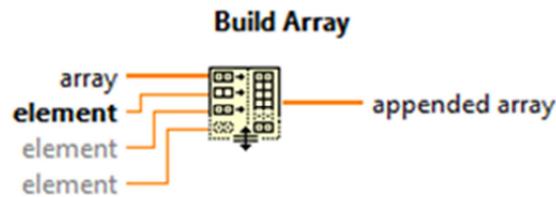


Fig. IV. 31 Construir Matriz en LabView

Fuente: LabView 2012

- **WriteToSpreadsheet File.vi**: permite guardar los archivos en una hoja con extensión .xls o .txt, en este elemento se deben configurar varios ítems para visualizar los datos de manera legible y correcta.
 - **Format(%s)**: la se debe colocar una cantidad considerable para que los datos se puedan visualizar de manera clara, mientras menor sea el numero menos dígitos se desplegaran en el archivo. Por ejemplo: %20f
 - **File Path**: se debe especificar el lugar donde se desea guardar el archivo y cuándo se crea un nuevo archivo. Por ejemplo: F:\base datos\prueba2.xls
 - **1D data**: se selecciona de acuerdo a la necesidad de la base de datos, para seleccionar esta opción se debe tener en cuenta la cantidad de datos a ser almacenados.
 - **Appendto file**: trabaja con valores lógicos, cuando el control de esta línea se encuentra en Verdadero cada vez que se ejecute el VI se sobrescribirá en el archivo, caso contrario se crea un nuevo archivo.

- **Transpose:** se emplean valores lógicos, como verdadero o falso, y permite configurar el modo de almacenamiento en la hoja de texto o cálculo que se crea, esto implica que se puede almacenar de manera horizontal o vertical.

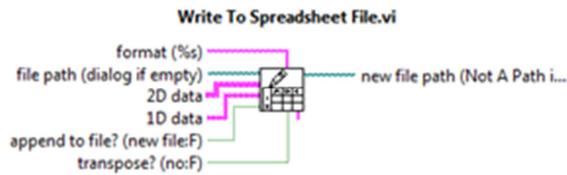


Fig. IV. 32 Hoja de Datos en LabView

Fuente: LabView 2012

4.4.5. ESTRUCTURAS EMPLEADAS.

Para realizar un control de datos de manera adecuada es necesario escoger una o varias estructuras de control que permiten un control de datos de manera deseada por el programador.

Dentro de la paleta de programación se encuentran todas las estructuras que se pueden emplear en LabView.

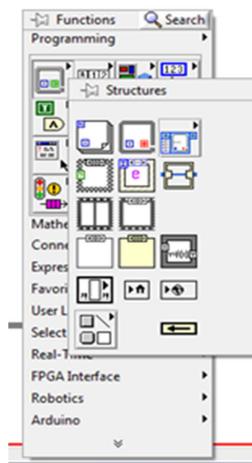


Fig. IV. 33 Paleta de Estructuras de Programación en LabView

Fuente: LabView 2012

- **WhileLoop**

Es la estructura que se emplea para realizar este sistema HMI, el mismo que se detendrá si y solo si se pulsa el botón de STOP que se encuentra conectado en la parte inferior de la estructura de control.



Fig. IV. 34 Control de la Estructura While

Fuente: Autor

En la otra estructura de control no se tiene ningún botón de STOP por lo cual esa estructura se mantiene activa siempre.



Fig. IV. 35 Control Booleano de la Estructura While

Fuente: Autor

4.4.6. PASOS QUE SE SIGUIERON PARA CREAR EL HMI.

- **Primer estructura WhileLoop**

La estructura Whileloop empleada se usa para realizar la base de datos, la misma que sensorará cada 15 minutos, enviando los datos adquiridos a la base de datos, que se crea empleando los elementos anteriormente descritos.

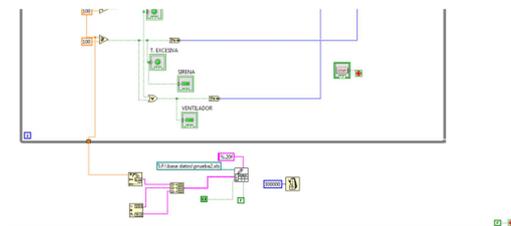


Fig. IV. 36 Diagrama de Bloques Estructura While

Fuente: Autor

Abrir el puerto de comunicación

Es necesario usar el bloque Init.VI que abre una conexión con Arduino, permitiendo de esa manera la transmisión y adquisición de datos desde y hacia la placa.

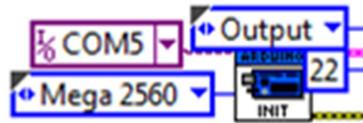


Fig. IV. 37 Inicio de Comunicación Arduino-LabView

Fuente: Autor

Declaración de los pines digitales

Se debe especificar todos los pines que se usaran en la tarjeta Arduino conectada, detallando claramente cuáles serán empleados como entradas y salidas. Para que se tenga un correcto uso y empleo de los pines de la tarjeta.

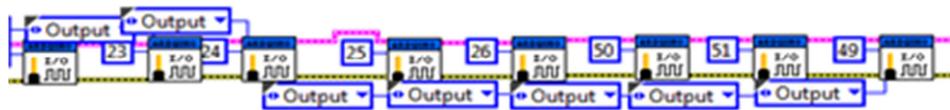


Fig. IV. 38 Declaración de Pines Arduino

Fuente: Autor

Segunda estructura WhileLoop

Se encarga de recibir datos de manera constante desde la termocupla que se encuentra conectada a la entrada lógica de la tarjeta.

Esta estructura se ejecuta cada segundo por lo que se puede visualizar de manera oportuna cualquier cambio considerable que sufra el sistema.

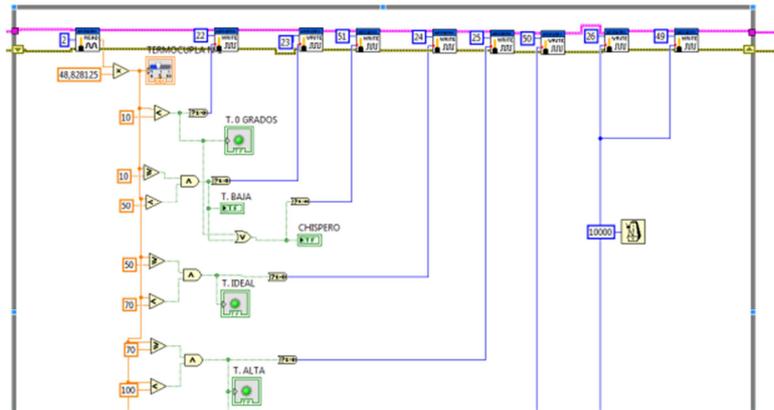


Fig. IV. 39 Estructura del Programa de Arduino

Fuente: Autor

Lectura PIN analógico

Se especifica el pin de la tarjeta Arduino, al cual se encuentra conectado el termopar.



Fig. IV. 40 Lectura Pin Analógico

Fuente: Autor

Representación gráfica del valor obtenido.

Se emplea un controlador numérico para visualizar el valor obtenido, este se encuentra configurado en una escala igual a la de trabajo del termopar, este control variará de acuerdo al valor que se obtenga en el sensor.

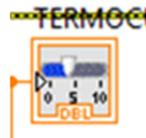


Fig. IV. 41 Representación del valor Análogo Gráficamente

Fuente: Autor

Activación de los pines digitales

Los pines digitales que fueron declarados como salidas serán activados de acuerdo al diseño que se encuentra plasmado en el sistema de control, los mismos activaran diodos led que indican el rango de temperatura en el que se encuentra el termopar, de manera adicional tres etapas de potencia que permiten el control de la temperatura del horno, se ubica un ventilador, una sirena y un encendedor los mismos que puede operar en un rango de Voltaje de 110- 220V.



Fig. IV. 42 Control de Salidas Digitales

Fuente: Autor

Cerrar el puerto de comunicación.

Se debe colocar el bloque Close.VI cierra en puerto de comunicación, este bloque al igual que el que inicializa son indispensables y no pueden ser omitidos.



Fig. IV. 43 Cierre de Comunicación Arduino-LabView

Fuente: Autor

CAPITULO V

PRUEBAS Y RESULTADOS

5.1. TABLEROS DE CONTROL DE LA MAQUINA AGITADORA

PRUEBAS DE FUNCIONAMIENTO

PRUEBA 1

- **TAREA:** ENCENDIDO DE 2 MOTORES A LA VEZ
- **TIEMPO DE PROGRAMACIÓN:** 1 hora
- **TIEMPO DE TRABAJO:** 10 horas
- **SECUENCIA:** Automática
- **FECHA PRUEBA:** lunes, 03 de Noviembre del 2014

PRUEBA 2

- **TAREA:** ENCENDIDO DE 2 MOTORES A LA VEZ
- **TIEMPO DE PROGRAMACIÓN:** 30 minutos
- **TIEMPO DE TRABAJO:** 8 horas
- **SECUENCIA:** Automática
- **FECHA PRUEBA:** martes, 04 de Noviembre del 2014

PRUEBA 3

- **TAREA:** ENCENDIDO DE 2 MOTORES A LA VEZ
- **TIEMPO DE PROGRAMACIÓN:** 15 minutos
- **TIEMPO DE TRABAJO:** 8 horas
- **SECUENCIA:** Automática
- **FECHA PRUEBA:** miércoles, 05 de Noviembre del 2014

PRUEBA 4

- **TAREA:** ENCENDIDO INDIVIDUAL DE LAS SALIDAS
- **TIEMPO DE INDIVIDUAL:** 6 horas c/salida
- **TIEMPO DE TRABAJO:** 6 horas
- **SECUENCIA:** Manual
- **FECHA PRUEBA:** jueves, 06 de Noviembre del 2014

PRUEBA 5

- **TAREA:** ENCENDIDO INDIVIDUAL DE LAS SALIDAS
- **TIEMPO DE INDIVIDUAL:** 1 horas c/salida
- **TIEMPO DE TRABAJO:** 6 horas
- **SECUENCIA:** Manual
- **FECHA PRUEBA:** viernes, 07 de Noviembre del 2014

TOTAL HORAS DE PRUEBA: 38 HORAS

PRUEBAS INDUSTRIALES

PRUEBA 1

- **TAREA:** ENCENDIDO DE 2 MOTORES A LA VEZ
- **TIEMPO DE PROGRAMACIÓN:** 1 hora
- **TIEMPO DE TRABAJO:** 5 horas – 3horas
- **SECUENCIA:** Automática
- **FECHA PRUEBA:** lunes, 10 de Noviembre del 2014 de 8:00 a 13:00
- **FECHA PRUEBA:** miércoles, 12 de Noviembre del 2014 de 15:00 a 18:00

TOTAL HORAS DE PRUEBA1: 8HORAS

PRUEBA 2

- **TAREA:** ENCENDIDO DE 2 MOTORES A LA VEZ
- **TIEMPO DE PROGRAMACIÓN:** 30 minutos
- **TIEMPO DE TRABAJO:** 3 horas – 5horas
- **SECUENCIA:** Automática
- **FECHA PRUEBA:** lunes, 10 de Noviembre del 2014 de 15:00 a 18:00
- **FECHA PRUEBA:** jueves, 13 de Noviembre del 2014 de 8:00 a 13:00

TOTAL HORAS DE PRUEBA: 8HORAS

PRUEBA 3

- **TAREA:** ENCENDIDO DE 2 MOTORES A LA VEZ
- **TIEMPO DE PROGRAMACIÓN:** 15 minutos
- **TIEMPO DE TRABAJO:** 5 horas – 3 horas
- **SECUENCIA:** Automática
- **FECHA PRUEBA:** martes, 11 de Noviembre del 2014 de 8:00 a 13:00
- **FECHA PRUEBA:** jueves, 13 de Noviembre del 2014 de 15:00 a 18:00

TOTAL HORAS DE PRUEBA: 8HORAS

PRUEBA 4

- **TAREA:** ENCENDIDO INDIVIDUAL DE LAS SALIDAS
- **TIEMPO DE INDIVIDUAL:** 3 horas c/salida
- **TIEMPO DE TRABAJO:** 3 horas – 5horas
- **SECUENCIA:** Manual
- **FECHA PRUEBA:** martes, 11 de Noviembre del 2014 de 15:00 a 18:00
- **FECHA PRUEBA:** viernes, 14 de Noviembre del 2014 de 8:00 a 13:00

TOTAL HORAS DE PRUEBA: 8HORAS

PRUEBA 5

- **TAREA:** ENCENDIDO INDIVIDUAL DE LAS SALIDAS
- **TIEMPO DE INDIVIDUAL:** 1 horas c/salida
- **TIEMPO DE TRABAJO:** 5 horas – 3 horas
- **SECUENCIA:** Manual
- **FECHA PRUEBA:** miércoles, 12 de Noviembre del 2014 de 8:00 a 13:00
- **FECHA PRUEBA:** viernes, 14 de Noviembre del 2014 de 15:00 a 18:00

TOTAL HORAS DE PRUEBA: 8HORAS

Tabla V. I Cuadro de Resultados Tablero Arduino

5.1.1. Cuadro de resultados Tablero de control Arduino

Prueba 1	Ruido	Vibraciones	Tiempo trabajo	Funcionamiento	# cambios	Observaciones
Día 1	93 db	Despreciable	5 h	OK	5	Se desfasa arduino con 1 o 2 seg
Día 2	93 db	Despreciable	3 h	OK	3	

Prueba 2	Ruido	Vibraciones	Tiempo trabajo	Funcionamiento	# cambios	Observaciones
Día 1	93 db	Despreciable	3 h	OK	6	Arduino al energizarse ejecuta las sentencias de manera instantanea.
Día 2	92 db	Despreciable	5 h	OK	10	

Prueba 3	Ruido	Vibraciones	Tiempo trabajo	Funcionamiento	# cambios	Observaciones
Día 1	93 db	Despreciable	5 h	OK	20	LOGO! Al energizarse se demora hasta iniciar la comprobacion de entradas.
Día 2	93 db	Despreciable	3 h	OK	12	

Prueba 4	Ruido	Vibraciones	Tiempo trabajo	Funcionamiento	# cambios	Observaciones
Día 1	93 db	Despreciable	3 h	OK	N/A	
Día 2	93 db	Despreciable	5 h	OK	N/A	

Prueba 5	Ruido	Vibraciones	Tiempo trabajo	Funcionamiento	# cambios	Observaciones
Día 1	93 db	Despreciable	5 h	OK	5	
Día 2	93 db	Despreciable	3 h	OK	3	

FUENTE: AUTOR

5.1.2. Análisis de resultados Tablero de Control Arduino

Para este grupo de pruebas se empleó el método de observación Científica obteniendo como resultado que: Arduino mega es capaz de realizar la misma función que el PLC Siemens LOGO! 230RC. Es capaz de trabajar sin ningún inconveniente en un lugar donde se encuentra expuesto a ruido, polvo en si en un ambiente hostil.

El tarjeta Arduino Mega al tener tiempos de programación extensas se desfasa 1 0 segundos con respecto al PLC Siemens LOGO!. La modificación de los tiempos de programación en LOGO! 230RC se los realiza de manera sencilla por medio de la pantalla LCD y teclas de navegación que tiene incorporadas, a diferencia de la tarjeta Arduino que se debe modificar el tiempo de programación en el Sketch y cargar el programa modificado a la tarjeta.

La tarjeta de Arduino Mega es capaz de ejecutar las sentencias de manera inmediata una vez que se energiza el dispositivo, LOGO 230RC no es capaz de reaccionar de la misma manera ya que este elemento tarda varios segundos hasta comprobar su estado.

Las interrupciones de Arduino funcionan de manera correcta para corta el suministro de energía a los actuadores, pero hay que reiniciar el dispositivo para que inicie correctamente la secuencia del programa.

5.2. CONTROL DE TEMPERATURA PARA LA COCCION DE BALDOSAS MEDIANTE ARDUINO

5.2.1. Cuadro de Resultados Tablero de Temperatura Arduino

Tabla V. II *Valores Temperatura Controlador Arduino*

LOGO	ARDUINO	792	792,763508	871	925,579975
744	820,3041	792	794,767967	873	896,00523
742	736,752483	788	797,774655	868	878,479456
742	796,75724	786	796,772425	867	767,848003
745	745,323381	784	798,776884	863	873,002651
746	755,610153	782	798,776884	878	869,716568
744	750,131081	781	798,776884	880	869,716568
743	783,664985	781	799,779114	872	924,484614
739	738,777253	781	797,774655	874	896,00523
738	744,38822	787	798,776884	882	879,574816
735	734,101448	796	797,774655	868	868,621207
739	739,712414	795	797,774655	866	855,476876
740	736,58742	794	798,776884	862	847,80935
764	668,640172	796	797,774655	863	860,953681
760	791,146274	810	797,774655	860	876,288734
756	736,906931	800	797,774655	940	866,430485
756	728,490481	783	796,772425	927	860,953681
762	750,934347	783	796,772425	908	853,286154
758	797,692401	784	795,770196	942	919,007809
753	727,55532	785	796,772425	884	863,144403
751	745,323381	785	795,770196	877	867,525846
765	786,470468	800	795,770196	868	837,951102
763	742,517898	839	795,770196	880	878,479456
762	730,360803	848	779,896973	876	860,953681
760	728,490481	846	878,479456	884	875,193373
764	746,258542	886	870,811929	772	772,978242
765	727,55532	885	808,376357	798	777,87051
762	716,333387	890	877,384095	781	780,80587
755	749,999186	895	906,958839	788	780,80587
752	752,804669	899	862,049042	800	780,80587
753	793,951757	900	890,528426	884	781,784324
751	749,064025	888	935,438223	880	779,827417
751	758,415636	900	892,719147	876	778,848963
748	735,97177	882	844,523267	877	870,811929
753	764,026602	898	848,904711		
762	801,433046	887	867,525846		
759	727,55532	882	924,484614		
752	720,074032	877	867,525846		
742	700,435649	873	896,00523		
742	739,712414	878	880,670177		
742	733,166287	865	915,721727		
762	787,988032	870	913,531005		

Fuente: Autor

5.2.2. Análisis de resultados Tablero de Temperatura Arduino

Para obtener un resultado preciso y que pueda afirmar o negar la hipótesis planteada se realizaron las siguientes pruebas, con el fin de que la investigación cuente con los fundamentos suficientes.

➤ **Test Student para 2 muestras**

```
> attach(temp)
> t.test(LOGO, ARDUINO)

Welch Two Sample t-test

data: LOGO and ARDUINO
t = 0.7241, df = 231.944, p-value = 0.4697
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -9.830004 21.253324
sample estimates:
mean of x mean of y
 814.2222  808.5106
```

Fig. V. 1 Test Student

Fuente: Software R

La cantidad de datos que se obtuvieron nos permiten aplicar este tipo de prueba. En esta prueba se aplica para encontrar la diferencia de entre las medias muestrales de dos poblaciones de datos, nos permite comprobar si las 2 medias son iguales.

El resultado de esta prueba arroja el siguiente resultado:

$$p=0.4697$$

este valor obtenido es mayor que $p=0.05$ por lo cual no se puede rechazar la hipótesis Nula y se acepta que no existe diferencia entre las dos medias.

➤ **Test de Inferencia no paramétrica Test Wilcoxon o Mann-Whitney**

```
> wilcox.test(LOGO, ARDUINO)

      Wilcoxon rank sum test with continuity correction

data:  LOGO and ARDUINO
W = 7183, p-value = 0.5139
alternative hypothesis: true location shift is not equal to 0
```

Fig. V. 2 Test Wilcoxon

Fuente: Software R

Es una prueba no paramétrica, la misma que permite comparar la media de dos grupos de muestras para poder verificar si existe una diferencia significativa entre ellas.

$$p=0.5139$$

al obtener este valor no se puede rechazar la hipótesis.

➤ **Índice de Spearman o Índice de correlación**

```
> cor.test(LOGO, ARDUINO, method="spearman")

      Spearman's rank correlation rho

data:  LOGO and ARDUINO
S = 53783.27, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.7985011

Warning message:
In cor.test.default(LOGO, ARDUINO, method = "spearman") :
  Cannot compute exact p-value with ties
```

Fig. V. 3 Índice de Spearman

Fuente: Software R

También conocido como índice de correlación de rangos, esta prueba relacionará los dos grupos de datos.

El valor obtenido como resultado de esta prueba es:

$$V=0.7985$$

Se obtiene un valor bastante elevado por lo cual se acepta la similitud entre ambas medias, porque mientras más parecido el valor encontrado es a 1, es mejor la relación existente.

$V_1=2 \times 10^{-16}$, valor de correlación, este valor nos indica que ese rechaza la hipótesis nula de no correlación del test de Spearman, lo que implica que ambos conjuntos de datos están correlacionados.

➤ **Test de 2 Varianzas.**

```
> var.test(LOGO, ARDUINO)

      F test to compare two variances

data:  LOGO and ARDUINO
F = 0.9695, num df = 116, denom df = 116, p-value = 0.8677
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.672500 1.397589
sample estimates:
ratio of variances
 0.9694734
```

Fig. V. 4 Test De Varianzas

Fuente: Software R

Se determina que el valor de las Varianzas es igual, ya que el ratio de las varianzas es muy próximo a la unidad y además no se puede rechazar la hipótesis nula de que las varianzas son iguales.

p-value= 0,8677

Ratio Varianzas=0,96947

Lo que nos indica que la media y la varianza son iguales.

CONCLUSIONES

- La tarjeta Arduino Mega es capaz de trabajar de la misma manera que un PLC Siemens LOGO 230RC, en un ambiente industrial donde el dispositivo se encuentra expuesto a ruido, interferencias electromagnéticas, polvo y golpes.
- La tarjeta Arduino Mega se la puede monitorear de manera sencilla y directa, caso contrario, el controlador Siemens LOGO! 230RC no se puede monitorear de manera directa, es necesario emplear una tarjeta de adquisición de datos de LabView lo cual generará más costos.
- La tarjeta Arduino emplea un solo cable USB para la transmisión de programas hacia la tarjeta, y con el mismo elemento se puede monitorear y usar como fuente de alimentación para dicho dispositivo, algo que no se puede emplear en el PLC Siemens LOGO 230RC, ya que este controlador emplea una fuente de alimentación independiente.
- La programación en Arduino puede llegar a ser muy compleja y extensa, además que se generan errores cuando las palabras empleadas no se encuentran escritas de manera correcta, la programación del controlador en LOGO!SoftComfort es sencilla y rápida, de manera adicional el programa puede ser simulado en el mismo Software, sin emplear software diferente.
- El empleo de varias fuentes de poder para la tarjeta Arduino puede representar un problema, ya que en la industria no se dispone de muchos espacios donde conectar las fuentes, Siemens LOGO! 230RC, hace uso de una única fuente de alimentación.
- La relación costo-beneficiosa es muy importante, y juega un papel importante a nivel industrial, un mayor empleo de recursos genera desconformidad con la economía de la empresa, por lo cual Arduino es una excelente opción para reemplazar a este dicho controlador.

RECOMENDACIONES

- El empleo de cable UTP, para evitar posibles errores por interferencias de campos magnéticos, además que debe soportar valores de voltaje no muy altos, para las entradas y salidas de la placa de Arduino que generalmente se encuentran en el orden de 5V.
- Es conveniente proteger a la placa Arduino contra golpes, polvo, ruido e interferencias electromagnéticas, para asegurar el correcto funcionamiento del dispositivo, por lo cual se emplea una caja hecha de metal.
- Construir o colocar borneras que permitan emplear cables de uso industrial, aunque esto puede encarecer los costos.
- El uso de interrupciones para generar los resultados deseados, y ubicar un botón de reseteo para poder reiniciar la placa a las condiciones iniciales.

RESUMEN.

Se implementó módulos de pruebas para medir la eficiencia de los controladores Arduino Mega y Siemens LOGO! 230RC en procesos industriales.

Para el desarrollo de la investigación se obtuvo la información correspondiente acerca de las tecnologías y de los elementos a emplear de acuerdo a su uso y función dentro de cada tablero; se valoraron varios aspectos que pueden afectar al correcto funcionamiento de la tarjeta Arduino entre los cuales se consideraron el ruido, polvo, vibraciones y golpes, que son elementos que se encuentran fácilmente en el sector industrial. Se realizó varias pruebas de las cuales se obtuvieron datos de los tableros que se crearon para la verificación de la hipótesis, se empleó varios software de acuerdo a la necesidad que se generaba durante la ejecución de la investigación entre los cuales se puede destacar: LOGO Soft Comfort, IDE Arduino, Isis Proteus, LabView, y Excel para la creación de la base de datos. Para la parte de hardware se empleó Relés Termomagnéticos, Tarjeta de Arduino, Siemens LOGO! 230RC, los actuadores fueron representados por focos, AD595, diodes led, switch de 2 y 3 posiciones, una sirena y una computadora personal; la protección de la placa de Arduino debe ser metálica para que el dispositivo se encuentre protegido contra cualquier posible y eventual complicación.

El análisis de los resultados que se obtuvieron de las diferentes pruebas muestra que la tarjeta Arduino es capaz de trabajar de la misma manera eficiente que el controlador Siemens LOGO! 230RC, para obtener estos resultados se realizaron varias pruebas estadísticas las mismas que demostraron que tanto la media como la varianza de los datos almacenados son muy similares y en algunas de estas pruebas se podría asumir que estos valores son iguales. Al encontrar estos valores estadísticos se asume que la hipótesis plantada es aceptada.

Se concluye que Arduino Mega es una gran opción para reemplazar al controlador Siemens LOGO! 230RC, siendo un dispositivo que cumple con las mismas funciones y características de dicho controlador, teniendo un mayor número de entradas digitales y analógicas por lo cual no es necesario adquirir módulos adicionales, a un costo económico menor.

Se recomienda que Arduino Mega sea empleado en automatización de procesos industriales porque es una opción más económica con iguales y mejores condiciones que un Siemens LOGO 230RC.

ABSTRACT

Test modules to measure the efficiency of Arduino Mega, Siemens LOGO! 230RC controllers for industrial processes have been implemented.

For the development of the research, it was necessary to get information about technologies and elements to be used according to its use in each panel; some aspects affecting the right Arduino card performance such as: noise, dust, vibrations and shocks were also evaluated; these elements are easily found in the industrial area. The hypothesis verification was done through different trials to get data from the panel. For this, different software such as: LOGO Soft Comfort, IDE Arduino, Isis Proteus, LabView and Excel were used according to the needs the research required during this implementation. Thermomagnetic relays, Arduino Card, Siemens LOGO! 230rc were used for implementing the hardware, the actuators were represented by light bulbs, AD595, led diodes, a 2 and 3 position switch, a siren and a laptop. The protection for the Arduino Card must be metallic for protecting the device against any possible complication. The analysis of the results obtained from different trials show that Arduino Card performance is as efficient as Siemens LOGO! 230RC controller. To demonstrate the similarity existing between the mean and the variance some statistical tests were carried out, some of these tests showed that the values are the same. From these statistical values the hypothesis was accepted.

It is concluded that Arduino Mega is a big choice to substitute Siemens LOGO! 230RC controller, this device has the same functions and characteristics of the controller, since it has more digital and analog inputs, that is why it is unnecessary to get cheaper additional modules.

It is recommended to use Arduino Mega in the industrial automation processes since it is more economic and better than Siemens LOGO! 230RC.

BIBLIOGRAFÍA

BIBLIOGRAFÍA GENERAL

- DORF Richard, B. R. (2005). *Sistemas de Control Moderno*. Madrid: PEARSON EDUCACION.
- National Instruments. (2006). *LabVIEW básico I Introducción*. Texas.
- PALLAS, A. (2003). *Sensores Y Acondicionadores de Señal*. Barcelona-España: Marcombo.
- RUIZ, J. (s.f.). *EDUCACION SECUNDARIA Y UNIVERSITARIOA: ROBÓTICA CONTROL Y SIMULACIÓN*. Recuperado el 12 de 05 de 2014, de <http://josemanuelruizgutierrez.blogspot.com/>

WEBGRAFÍA RELACIONADA AL TEMA

- ARDUINO. (s.f.). *Arduino-Home*. Obtenido de <http://www.arduino.cc/>
- *Automatizacion y control Industrial*. (s.f.). Obtenido de <http://controlyautomatizacionindustrial.blogspot.com/p/arduino-projects.html>
- Benitez, R. (s.f.). *BotScience-imagina, diseña, construye..* Obtenido de <https://botscience.wordpress.com/2012/06/05/historia-de-arduino-y-su-nacimiento/>
- *El documental Arduino narra la historia del open source en el mundo físico*. (s.f.). Obtenido de <http://www.euskadinnova.net/es/innovacion-social/noticias/documental-arduino-narra-historia-open-source-mundo-fisico/7285.aspx>
- Paniagua, S. (s.f.). *Arduino, la revoluvion Silenciosa*. Obtenido de <http://www.sorayapaniagua.com/2011/03/14/arduino-la-revolucion-silenciosa-del-hardware-libre/>
- RUIZ, J. (s.f.). *EDUCACION SECUNDARIA Y UNIVERSITARIOA: ROBÓTICA CONTROL Y SIMULACIÓN*. Recuperado el 12 de 05 de 2014, de <http://josemanuelruizgutierrez.blogspot.com/>

<https://mall.industry.siemens.commall/es/ec/Catalog/Product/6ED1052-1FB00-0BA6>

ANEXOS

ANEXO 1

```
int aut=3; // senal del selector en modo automatico
```

```
int mm1=5; // senal marcha motor 1
```

```
int pm1=6; // senal paro motor 1
```

```
int mm2=8; // senal marcha motor 2
```

```
int pm2=9; // senal paro motor 2
```

```
int mm3=11; // senal marcha motor 3
```

```
int pm3=12; // senal paro motor 3
```

```
int lm1= 38; // pin de activacion led 1
```

```
int lm2= 42; // pin de activacion led 2
```

```
int lm3= 46; // pin de activacion led 3
```

```
int lm4= 50; // pin de activacion led 3
```

```
int est_aut=0;
```

```
int est_mm1=0;
```

```
int est_pm1=0;
```

```
int est_mm2=0;
```

```
int est_pm2=0;
```

```
int est_mm3=0;
```

```
int est_pm3=0;
```

```
int v_inter=0;
```

```
void setup()
```

```
{
```

```
  attachInterrupt(0,paro,CHANGE);
```

```
  pinMode(aut,INPUT);
```

```
  pinMode(mm1,INPUT);
```

```
  pinMode(mm2,INPUT);
```

```
  pinMode(mm3,INPUT);
```

```
  pinMode(pm1,INPUT);
```

```
  pinMode(pm2,INPUT);
```

```
  pinMode(pm3,INPUT);
```

```
  pinMode(lm1,OUTPUT);
```

```
  pinMode(lm2,OUTPUT);
```

```
  pinMode(lm3,OUTPUT);
```

```
}
```

```
void loop()
```

```

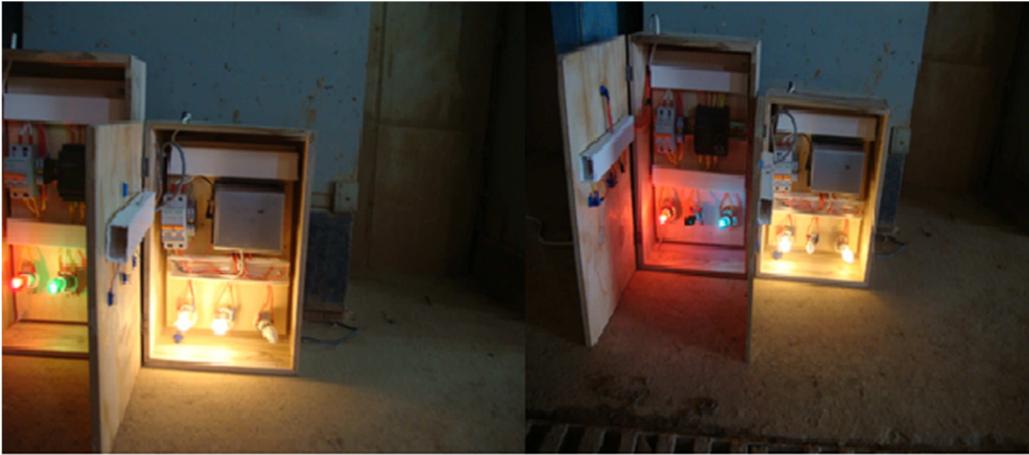
{
  est_aut=digitalRead(aut);
  if(est_aut==HIGH)
  {
    digitalWrite(lm1,LOW);
    digitalWrite(lm2,LOW);
    digitalWrite(lm3,HIGH);
    delay(3600000);
    digitalWrite(lm1,HIGH);
    digitalWrite(lm2,LOW);
    digitalWrite(lm3,LOW);
    delay(3600000);
    digitalWrite(lm1,LOW);
    digitalWrite(lm2,HIGH);
    digitalWrite(lm3,LOW);
    delay(3600000);
  }
  else
  {
    est_pm1=digitalRead(pm1);
    est_pm2=digitalRead(pm2);
    est_pm3=digitalRead(pm3);
    est_mm1=digitalRead(mm1);
    est_mm2=digitalRead(mm2);
    est_mm3=digitalRead(mm3);
    if(est_pm1==HIGH && est_pm2==HIGH && est_pm3==HIGH)
    {
      digitalWrite(lm1,HIGH);
      digitalWrite(lm2,HIGH);
      digitalWrite(lm3,HIGH);
    }
    else
    {
      if(est_mm1==HIGH && est_pm2==HIGH && est_pm3==HIGH)
      {
        digitalWrite(lm1,LOW);

```

```
digitalWrite(lm2,HIGH);
digitalWrite(lm3,HIGH);
}
else
{
if(est_pm1==HIGH && est_mm2==HIGH && est_pm3==HIGH)
{
digitalWrite(lm1,HIGH);
digitalWrite(lm2,LOW);
digitalWrite(lm3,HIGH);
}
else
{
if(est_pm1==HIGH && est_pm2==HIGH && est_mm3==HIGH)
{
digitalWrite(lm1,HIGH);
digitalWrite(lm2,HIGH);
digitalWrite(lm3,LOW);
}
else
{
if(est_mm1==HIGH && est_mm2==HIGH && est_pm3==HIGH)
{
digitalWrite(lm1,LOW);
digitalWrite(lm2,LOW);
digitalWrite(lm3,HIGH);
}
else
{
if(est_mm1==HIGH && est_pm2==HIGH && est_mm3==HIGH)
{
digitalWrite(lm1,LOW);
digitalWrite(lm2,HIGH);
digitalWrite(lm3,LOW);
}
else
```


ANEXO 2

PRUEBA 1

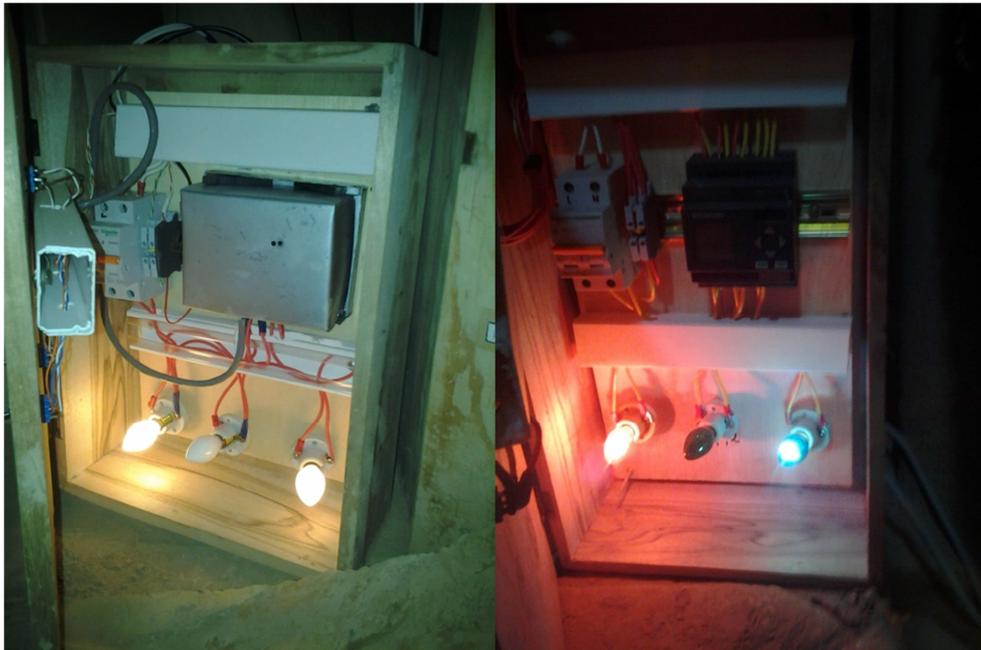


PRUEBA 2

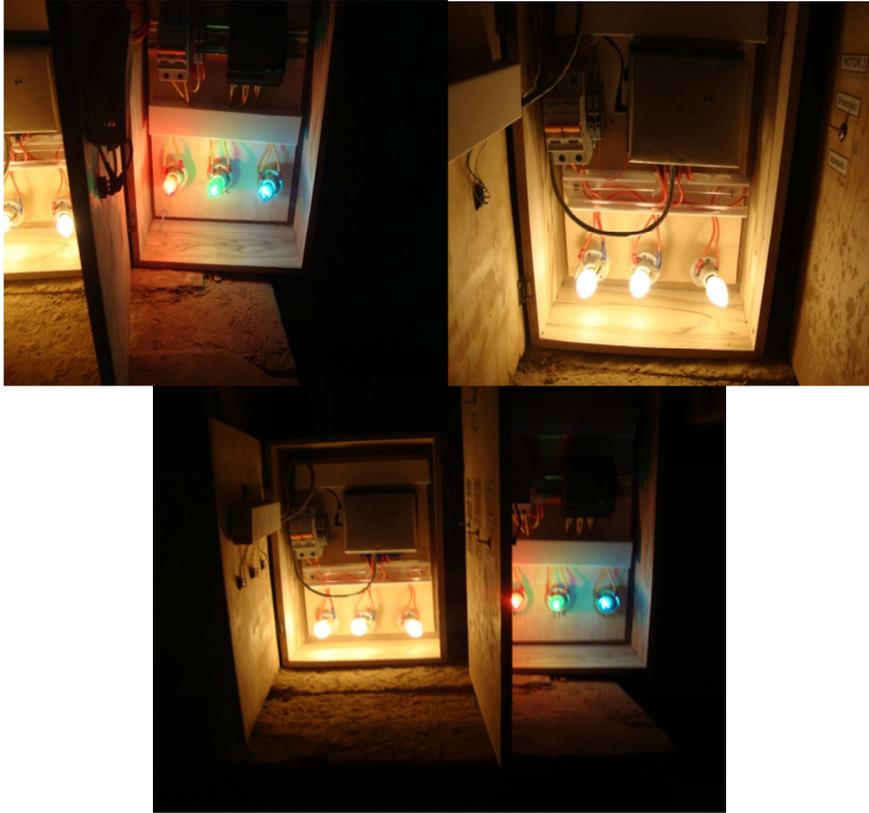




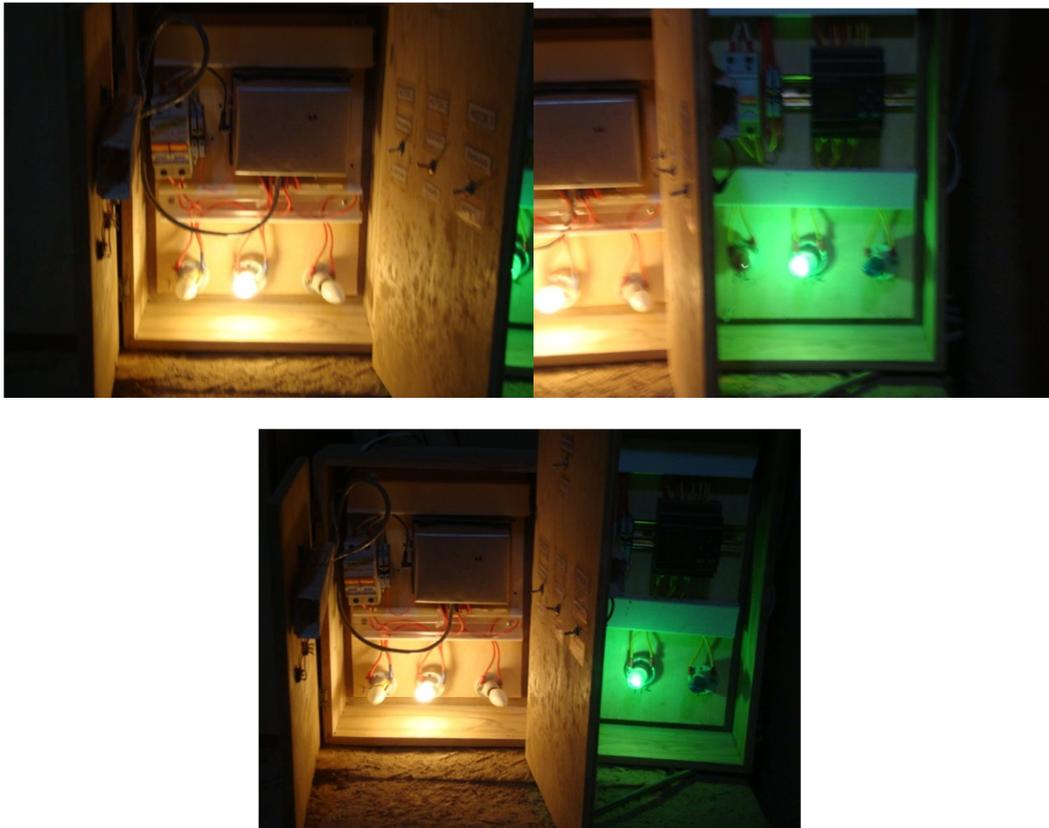
PRUEBA 3



PRUEBA 4



PRUEBA 5



PRUEBA TABLERO DE TEMPERATURA

