

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

SIMULACIÓN DE DOS SISTEMAS DE CONTROL INDUSTRIAL MEDIANTE CONTROLLINO A TRAVÉS DE UNA INTERFAZ HUMANO MÀQUINA HMI.

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTROMECAÁNICA**

RONY STEVEN SALAZAR QUINAUCHO

rony.salazar@epn.edu.ec

DIRECTOR: ING. ALAN DANIEL CUENCA SÁNCHEZ MSC.

alan.cuenca@epn.edu.ec

CODIRECTOR: ING. CARLOS ORLANDO ROMO HERRERA MSC.

carlos.romo@epn.edu.ec

Octubre, 2020

DECLARACIÓN

Yo, Rony Steven Salazar Quinaucho, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación -COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional. Entregaré toda la información técnica pertinente. En el caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente".



Rony Steven Salazar Quinaucho

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Rony Steven Salazar Quinaucho, bajo nuestra supervisión.

Ing. Alan Daniel Cuenca Sánchez

DIRECTOR DE PROYECTO

Ing. Carlos Orlando Romo Herrera

CODIRECTOR DE PROYECTO

AGRADECIMIENTO

Agradecimiento profundo hacia mis padres, hermana y hermano por el apoyo incondicional durante esta etapa de mi vida.

A la ESCUELA POLITÉCNICA NACIONAL Y ESCUELA DE FORMACIÓN DE TECNÓLOGOS, por permitirme realizar mi formación profesional y a sus docentes por impartir conocimiento teórico y en base a su experiencia aportar un mayor crecimiento profesional.

Un especial agradecimiento a mi tutor del proyecto Ing. Alan Cuenca por el tiempo y conocimiento brindado durante el desarrollo del proyecto.

Rony Salazar

DEDICATORIA

Este proyecto va dedicado a mis padres Cesar Salazar y María Quinaucho por formar un hijo con carácter y personalidad, gratitud por su esfuerzo rotundo y apoyo durante mi formación profesional, a mi hermana Jazmin Salazar y hermano Augusto Salazar quienes me brindaron un consejo en momentos difíciles a lo largo de mi vida personal y profesional. Este proyecto fue desarrollado gracias a ustedes y para ustedes por el esfuerzo generado hacia mi persona.

El conocimiento compartido genera aprendizaje, oportunidades y solidifica el conocimiento adquirido que un conocimiento guardado y sin ser compartido.

Rony Salazar

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	1
1.1 Planteamiento del problema.....	1
1.2 Justificación	1
1.3 Objetivos.....	2
2. METODOLOGÍA	3
2.1 Herramientas utilizadas para el desarrollo del proyecto	3
3. RESULTADOS Y DISCUSIÓN.....	4
3.1 PLC's Controllino	4
• Definición.....	4
• Software de Programación	6
3.2 Esquemas Electrónicos	6
• Entradas Controllino.....	7
• Salidas Controllino.....	8
• Subcircuitos	11
• Unidad Central de Procesamiento (CPU)	13
• Panel Frontal	14
3.3 Algoritmos de Control	16
• Librerías e instrucciones	16
• Control ON-OFF de Temperatura	20
• Control de un motor de dos velocidades	21
3.4 Interfaz Humano Máquina HMI	23
• Selección inicial	23
• Fuentes de texto.....	23
• Botones.....	25
• Indicadores	26
• Ventanas HMI	27
3.5 Pruebas de funcionamiento	29
• Pruebas de entradas y salidas.	29
• Prueba de comunicación Controllino-HMI	32
• Pruebas de los sistemas de control	34
4. CONCLUSIONES Y RECOMENDACIONES.....	38
4.1 Conclusiones.....	38
4.2 Recomendaciones	39
5. BIBLIOGRAFÍA	40
6. ANEXOS	42

Índice de figuras

Figura 1. Controllino Mini.....	4
Figura 2. Controllino Maxi.....	5
Figura 3. Circuito de entrada Controllino.....	7
Figura 4. Circuito de activación de salidas.....	8
Figura 5. Circuito para salidas tipo mosfet (Controllino Mini).....	9
Figura 6. Circuito para salidas tipo mosfet (Controllino Maxi).....	10
Figura 7. Circuito para salidas tipo relay.....	10
Figura 8. Subcircuito Controllino Mini.....	11
Figura 9. Subcircuito Controllino Maxi.....	12
Figura 10. ATMEGA 328P y oscilador XT.....	13
Figura 11. ATMEGA 2560 y oscilador XT.....	14
Figura 12. Panel Controllino Mini.....	15
Figura 13. Panel Controllino Maxi.....	16
Figura 14. Selección de controlador.....	17
Figura 15. Diagrama de Control on-off por histéresis.....	21
Figura 16. Circuito de potencia para velocidad rápida.....	22
Figura 17. Circuito de potencia para velocidad lenta.....	22
Figura 18. Selección de tipo y dirección de pantalla HMI.....	23
Figura 19. Fuentes de texto creadas.....	24
Figura 20. Atributos e identificador de texto.....	25
Figura 21. Atributos e identificador del botón.....	26
Figura 22. Atributos e identificador de los indicadores.....	27
Figura 23. Ventana de Control On-Off de Temperatura.....	28
Figura 24. Ventana de Control de un motor de dos velocidades.....	28
Figura 25. Valores de voltaje en entrada analógica.....	29
Figura 26. Voltaje de salida en pines digitales.....	30
Figura 27. Relay encendido.....	31
Figura 28. Señal PWM.....	32
Figura 29. Envío de datos mediante Controllino.....	32
Figura 30. Recepción de datos por parte de la HMI.....	33
Figura 31. Transmisión de dato HMI.....	33
Figura 32. Recepción de datos en el controlador.....	34
Figura 33. Simulación del sistema de Control On-Off de Temperatura.....	35
Figura 34. Simulación del control de un motor de dos velocidades.....	36

Índice de Tablas

<i>Tabla 1. Velocidades de transmisión de datos</i>	18
<i>Tabla 2. Pines Controllino Mini, ATMEGA 328P TQFP y Arduino Uno</i>	19
<i>Tabla 3. Pines Controllino Maxi, ATMEGA 2560 TQFP y Arduino Mega</i>	19
<i>Tabla 4. Fuentes de texto</i>	24
<i>Tabla 5. Parámetros de texto</i>	25
<i>Tabla 6. Parámetros de los botones</i>	26
<i>Tabla 7. Parámetros de los indicadores</i>	27
<i>Tabla 8. Valores registrados para entradas analógicas</i>	30
<i>Tabla 9. Valores registrados para salidas digitales</i>	30
<i>Tabla 10. Pruebas de los sistemas de control</i>	34

Resumen

El presente proyecto de titulación, “SIMULACIÓN DE DOS SISTEMAS DE CONTROL INDUSTRIAL MEDIANTE CONTROLLINO A TRAVÉS DE UNA INTERFAZ HUMANO MÁQUINA HMI”. Tiene como objetivo presentar el funcionamiento y aplicación industrial de Controladores Lógicos Programables PLC’s de open source o código abierto Controllino. En base al estudio realizado, el presente documento está conformado por cuatro capítulos que se detallan a continuación.

El capítulo uno contiene la introducción que abarca el planteamiento del problema, en el cuál se resaltan las limitaciones en cuanto a funcionalidades de los sistemas de control desarrollados, debido a no disponer de la licencia del software de programación de los PLC’s disponibles en el Laboratorio de Tecnología Industrial de la ESFOT. Adicionalmente, en este capítulo se presenta la justificación e importancia de realizar la simulación de sistemas de control utilizando PLC’s basados en hardware y software libre, para complementar y mejorar las prestaciones de los algoritmos de control de los PLC’s del LTI-ESFOT y finalmente se establecen los objetivos a cumplir con el desarrollo del proyecto.

El capítulo dos describe las técnicas y herramientas utilizadas, además de las actividades fundamentales para la ejecución de los objetivos planteados, empleando los conocimientos previamente adquiridos en la investigación del proyecto.

El capítulo tres describe el análisis de resultados de la simulación de los sistemas de control industrial basados en Controllino mediante una Interfaz Humano Máquina HMI. Además, se puntualizan las herramientas de comunicación entre el respectivo controlador y su interfaz mediante puertos seriales virtuales proporcionados por Virtual Serial Ports Emulator.

Finalmente, el capítulo cuatro describe las conclusiones y recomendaciones obtenidas del desarrollo del proyecto.

Abstract

The present degree project, "SIMULATION OF TWO INDUSTRIAL CONTROL SYSTEMS THROUGH HUMAN INTERFACE HMI MACHINE". It aims to present the operation and industrial application of Programmable Logic Controllers PLC's open source or open source Controllino. Based on the study conducted, this document consists of four chapters that are detailed below.

Chapter one contains the introduction that covers the problem, which highlights the limitations in terms of functionality of the control systems developed, due to not having the programming software license of the PLC's available in the Industrial Technology Laboratory of the ESFOT. Additionally, this chapter presents the justification and importance of simulating control systems using PLCs based on hardware and free software, to complement and improve the performance of the control algorithms of the LTI-ESFOT PLCs and finally sets the objectives to be met with the development of the project.

Chapter two describes the techniques and tools used, as well as the fundamental activities for the execution of the proposed objectives, using the knowledge previously acquired in the project's research.

Chapter three describes the analysis of simulation results of industrial control systems based on Controllino through a Human Machine Interface HMI. In addition, the communication tools between the respective controller and its interface through virtual serial ports provided by Virtual Serial Ports Emulator are pointed out.

Finally, chapter four describes the conclusions and recommendations obtained from the development of the project.

1. INTRODUCCIÓN

1.1 Planteamiento del problema

Los Controladores Lógicos Programables PLC son dispositivos de control que presentan componentes electrónicos robustos para soportar efectos adversos industriales. Además, debido a sus características los PLC's son utilizados para el desarrollo de sistemas de control en procesos industriales. El software de programación de los controladores lógicos programables permite realizar una simulación del algoritmo de control diseñado para su posterior implementación y puesta en marcha dentro del proceso, sin embargo, se debe disponer de una licencia para su respectiva utilización.

Debido a la limitación del costo de las licencias, se han desarrollado PLC's de open source o código abierto como Controllino, los cuales son basados en microcontroladores de la familia Arduino para aplicaciones industriales y se pueden programar en varias plataformas de sistemas embebidos. El Laboratorio de Tecnología Industrial de la ESFOT cuenta con PLC's SIEMENS mediante los cuales los estudiantes de Tecnología en Electromecánica realizan prácticas, sin embargo, no se cuenta con la respectiva licencia del software de programación lo cual limita las funcionalidades de los sistemas de control desarrollados.

Por esta razón, fue necesario desarrollar un simulador virtual para PLC's utilizando Arduino, el cual es una plataforma de código abierto, basada en hardware y software libre, flexible y fácil de utilizar. Por lo cual, en el presente proyecto se realizó una simulación de algoritmos de control para procesos industriales a través de PLC's Controllino.

1.2 Justificación

Los controladores Controllino ofrecen distintas ventajas sobre el tipo de programación, comunicación y manipulación con respecto a los PLC's tradicionales [1]. Además, están basados en hardware y software libre lo cual reduce los respectivos costos de implementación. En el presente proyecto se realizó una simulación utilizando Controllino mediante Proteus V8.9 [2] y el entorno de programación Arduino IDE para el desarrollo de dos aplicaciones de control industrial. A través de la simulación se plantea la posibilidad de utilizar estos PLC's basados en hardware y software libre para complementar y mejorar las prestaciones de los algoritmos de control de los PLC's SIEMENS S7-1200 del LTI-ESFOT. Para la programación se utilizó lenguaje C [3] para sistemas embebidos o microcontroladores, en el cual se procedió a estructurar el código o

algoritmo de control para Controllino con los microcontroladores Atmega328p TQFP y Atmega2560 puntualizando sus respectivas ventajas. Adicionalmente, se realizó el respectivo control local y remoto para comandar actuadores mediante valores de temperatura y se diseñó una Interfaz Humano Máquina HMI [4] mediante Nextion Editor [5] para controlar, monitorear y supervisar los procesos establecidos.

1.3 Objetivos

- **Objetivo general**

Simular dos sistemas de control industrial mediante Controllino a través de una Interfaz Humano Máquina HMI.

- **Objetivos específicos**

- Adquirir información de compatibilidad de dispositivos Controllino.
- Realizar la respectiva simulación mediante Proteus V8.9.
- Desarrollar dos algoritmos de control para Controllino mediante arduino IDE.
- Diseñar la Interfaz Humano Máquina HMI mediante Nextion editor.
- Realizar pruebas de funcionamiento.
- Elaborar un manual de usuario.

2. METODOLOGÍA

2.1 Herramientas utilizadas para el desarrollo del proyecto

Para el desarrollo del proyecto se utilizaron documentos situados en línea con información técnica (Datasheet) sobre Controladores Lógicos Programables PLC's de open source o código abierto Controllino. Dichos documentos brindaron un conocimiento elemental sobre el manejo, manipulación y funcionamiento del controlador Controllino.

La simulación se realizó mediante un diseño electrónico en Proteus v8.9 utilizando microcontroladores AVR ATMEGA 328P para Controllino Mini y ATMEGA 2560 para Controllino Maxi. En base a la información técnica adquirida previamente, se diseñaron entradas análogas/digitales mediante amplificadores operacionales LM741 y diodos Zener, mientras que para las salidas digitales se utilizaron configuraciones tipo mosfet y tipo relé, y finalmente para las salidas analógicas (PWM) se utilizó la configuración tipo mosfet. Adicionalmente, se utilizó comunicación I2C para una pantalla LCD 16x2 en la cual se visualizan las distintas variables de los sistemas de control industrial.

El algoritmo de control fue desarrollado utilizando lenguaje C en el software de programación IDE de Arduino, incluyendo librerías proporcionadas por el fabricante para la correcta compilación del algoritmo de control. Para el diseño de la Interfaz Humano Máquina HMI se utilizó Nextion Editor, el cual proporciona una herramienta de simulación en tiempo real mediante comunicación serial.

Además, para el envío de datos entre Controllino y la respectiva HMI se utilizaron puertos seriales virtuales mediante Virtual Serial Ports Emulator. Se realizaron pruebas de funcionamiento en entradas y salidas, panel de control, Interfaz Humano Máquina HMI y recepción y envío de datos.

Finalmente, se realizó un manual que permita el uso de los controladores Controllino simulados y su posible implementación.

3. RESULTADOS Y DISCUSIÓN

3.1 PLC's Controllino

- **Definición**

Controllino es un Controlador Lógico Programable PLC de open source o código abierto compaginable con Arduino basados en microcontroladores AVR por lo cual son sistemas de control programables modernos. Los cuales cumplen las normas legales europeas de funcionamiento y condiciones para una operación segura y eficiente. Además, dispone de una clasificación según el grado de automatización de Controllino. A continuación, se puntualizarán las características más relevantes de los Controllinos utilizados en el presente proyecto.

a) Mini

Controlador con unidad central de procesamiento basado en el microcontrolador AVR ATMEGA 328P, el cual dispone 8 entradas (analógicas/digitales), 14 salidas digitales (6 tipo relay y 8 tipo mosfet), 3 salidas PWM (utilizadas como analógicas). Además, se dispone de comunicación SPI y UART. En la Figura 1, se muestra el panel frontal de Controllino Mini con sus respectivas partes.

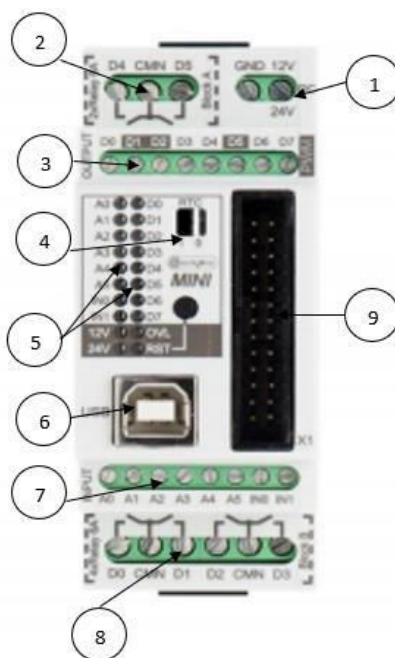


Figura 1. Controllino Mini
Fuente: Datasheet Controllino [1]

- 1.-Alimentación de 24 VDC
- 2.-Salidas tipo relé(D4-CMM-D5)
- 3.-Salidas de 24 VDC (D0-D7)
- 4.-Reloj de tiempo real (RTC)
- 5.-Indicadores de I/O
- 6.-USB conexión serial (carga y comunicación con PC)
- 7.-Entradas e Interrupciones (A0-IN1)
- 8.-Salidas tipo relé(D0-D3)
- 9.-Bus de terminales X1

b) Maxi

Controlador con unidad central de procesamiento basado en el microcontrolador AVR ATMEGA 2560, el cual dispone de 12 entradas (analógicas/digitales), 22 salidas digitales (10 tipo relé y 12 tipo mosfet), las mismas salidas tipo mosfet serán utilizadas para salidas PWM (utilizadas como analógicas). Adicionalmente dispone comunicación SPI, UART, I2C y Ethernet (RJ45-10/100Mbps). En la Figura 2, se presenta el panel frontal con sus respectivas partes.

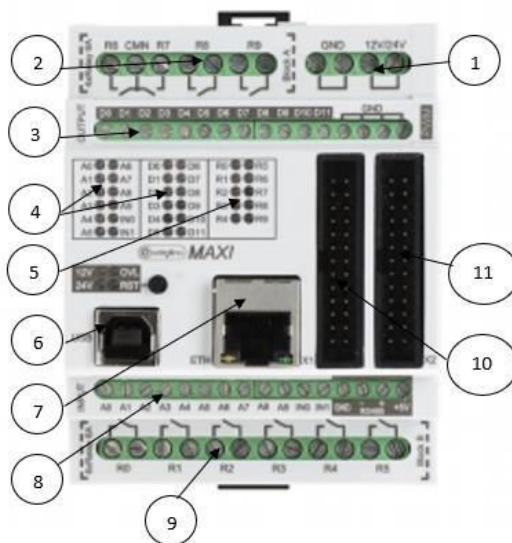


Figura 2. Controllino Maxi
Fuente: Datasheet Controllino [1]

- 1.-Alimentación de 24 VDC
- 2.-Salidas tipo relé(R6-R9)
- 3.-Salidas de 24 VDC (D0-D11)
- 4.-Indicadores de I/O
- 5.-Indicadores de Salida relé.
- 6.-USB conexión serial (carga y comunicación con PC)
- 7.-Conexión ethernet
- 8.-Entradas e Interrupciones (A0-IN1)
- 9.-Salidas tipo relé(R0-R5)
- 10.-Bus de terminales X1
- 11.-Bus de terminales X2

- **Software de Programación**

Controllino dispone de una alta compatibilidad con diferentes software de programación, gracias a su característica open source o código abierto [2]. Algunos de los principales programas mediante los cuales se pueden programar los controladores Controllino se listan a continuación:

- a) Arduino IDE
- b) Atmel Studio
- c) Flowcode
- d) Labview
- e) logi.CAD
- f) Visuino
- g) Matlab
- h) GNU Octave

3.2 Esquemas Electrónicos

Para la simulación de los Controladores Lógicos Programables PLC's Controllino Mini [3] y Controllino Maxi [4] se utilizó el software Proteus v8.9. En dicho software se realizaron esquemas electrónicos de entradas (analógicas/digitales) y salidas (digitales, relé y pwm) tomando en cuenta las características mencionadas en las hojas técnicas de los controladores. Todos los esquemas electrónicos diseñados se pueden observar en el ANEXO B.

- **Entradas Controllino**

Son parte fundamental de los controladores, ya que permiten recibir la información proveniente de los elementos de entrada tanto analógicos como digitales. Las señales ingresadas son de carácter eléctrico en forma de voltaje dependiendo del rango de operación del controlador. Las entradas analógicas permiten la variación de voltaje en el rango establecido por el controlador y las entradas digitales contienen dos estados lógicos de alto y bajo. Tomando en cuenta estas consideraciones, para las entradas digitales se utilizaron valores de voltaje de 24 [Vdc]. Mientras que, para las entradas analógicas se consideraron valores admisibles en un rango de voltaje de operación de 0 [Vdc] a 5 [Vdc]. Tomando en consideración la Ecuación 1, se obtuvo el valor de resistencia para operar en el rango de voltaje descrito anteriormente.

$$R2 = \frac{-V_{out} * R1}{V_{out} - V_{in}} \quad (1)$$

La Ecuación 1, se desarrolló considerando un voltaje de entrada “Vin” de 24 [Vdc], un voltaje de salida “Vout” de 5 [Vdc] y la resistencia “R1” de 2 [KΩ], con lo cual se obtuvo un valor de 526,32 [Ω] para la resistencia “R2”.

Adicionalmente, se implementó un seguidor de tensión mediante un C.I. LM741, el cual permitió obtener un voltaje continuo sin dependencia de la carga, además se añadieron diodos Zener de 4.7 [Vdc] para protección de la entrada de la unidad central de procesamiento. El esquema electrónico con la topología descrita sobre las entradas y su respectivo acondicionamiento se presenta en la Figura 3, donde el pin “A0-ENTRADA” recibe valores que oscilan entre 0 [Vdc] y 24 [Vdc] con relación a la alimentación del controlador y la salida “INT0” proporciona rangos de operación de 0 [Vdc] a 5 [Vdc].

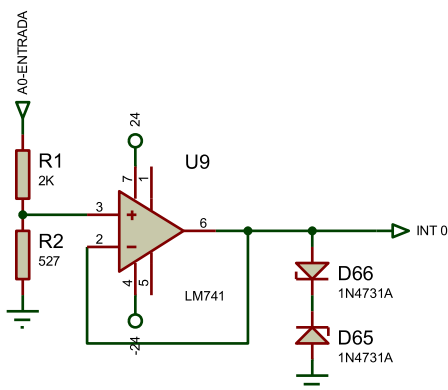


Figura 3. Circuito de entrada Controllino

- **Salidas Controllino.**

Las salidas controlan circuitos externos o dispositivos interconectados de forma independiente según los requerimientos del proceso. La dinamización de las salidas obedece a las acciones realizadas por el algoritmo de control proporcionando un voltaje a la salida, que viene determinado por un esquema electrónico previamente desarrollado. La señal enviada por la unidad central de procesamiento por medio de sus pines al esquema electrónico se aísla mediante optoacopladores PF817 [5] y diodos schokty con el fin de preservar el funcionamiento del controlador sin riesgo a sobrevoltajes. El esquema electrónico que cumple las carecterísticas mencionadas se presenta en la Figura 4.

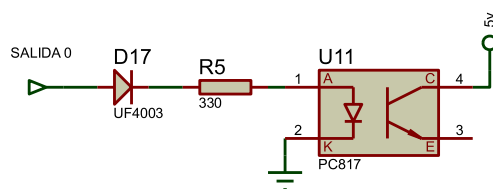


Figura 4. Circuito de activación de salidas

Para determinar el consumo de corriente por parte de los pines de las salidas digitales de los controladores, se utiliza la Ecuación 2:

$$I = \frac{V_{out} - V_{ak} - V_{op}}{R5} \quad (2)$$

Desarrollado con Voltajes “Vout” de 5[Vdc] proporcionado por la unidad central de procesamiento, “Vak” diodo ánodo-cátodo 0,7[Vdc] , “Vop” optoplador de 1,4[Vdc] y Resistencia dependiente del controlador la cual se visualiza en los ANEXO C.1 y ANEXO C.2.

Permitiendo obtener valores de 9.39 [mA] en cada pin y en el caso en el cual todos los pines estén activados consumirán una corriente total de 75.12 [mA] de la unidad central de procesamiento del Controllino Mini. Para la unidad central de procesamiento de Controllino Maxi se obtienen los valores de 6.596 [mA] para salidas tipo mosfet y 0.939 [mA] para salidas tipo relé en cada pin y en el caso en el cual todos los pines estén activados consumirán una corriente total de 113.69 [mA].

A continuación, se describen los tipos de salidas que disponen los controladores simulados:

- **Salidas Tipo Mosfet**

Para las salidas digitales se utilizó la configuración (High-Side-Switch) de un mosfet [6], el cual realiza la acción de conmutación, permitiendo proporcionar un voltaje fijo de salida en base a dos estados lógicos (alto y bajo).

a) Controllino Mini

La conmutación se realiza mediante un C.I. 4066 switch cuádruple bilateral, permitiendo la activación de los pines D0 del controlador. El circuito diseñado permite cumplir la restricción de activación digital y se visualiza en la Figura 5.

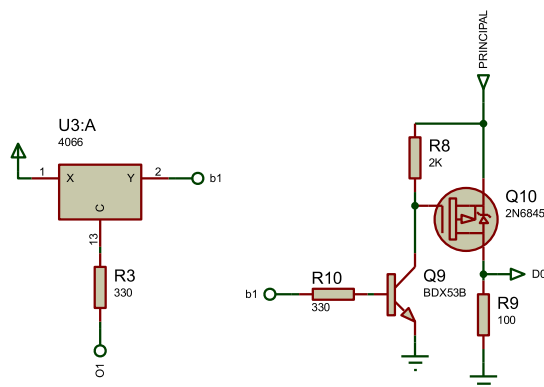


Figura 5. Circuito para salidas tipo mosfet (Controllino Mini)

b) Controllino Maxi

La conmutación se realiza mediante el driver UDN2982A hacia la configuración (High-Side-Switch), proporcionando un voltaje de 24 [Vdc] en el pin D0 del controlador. El circuito diseñado se visualiza en Figura 6.

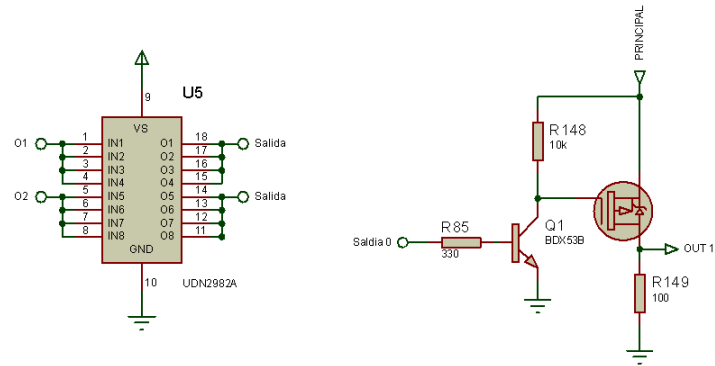


Figura 6. Circuito para salidas tipo mosfet (Controllino Maxi)

○ Salidas Relé

Dicha salida permite la conexión de voltajes en AC y DC de distinto valor y la comutación viene dada por el driver UDN2982A dirigida al transistor npn BDX53B, formando una diferencia de potencial en los extremos de la bobina o selenoide del relé. La configuración diseñada para las salidas tipo relé de los dos controladores Mini y Maxi es la misma. El circuito diseñado se visualiza en la Figura 7.

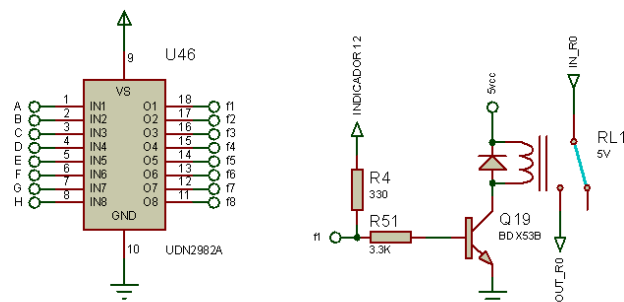


Figura 7. Circuito para salidas tipo relay

○ Salidas PWM

Realiza la respectiva modulación por ancho de pulso, para la obtención de un voltaje de salida promedio. A través, de la respectiva variación del ciclo de trabajo (duty cycle) permite la conexión de motores de 24[Vdc] para control de velocidad y aplicaciones en general que requieran dichos parámetros. La configuración diseñada para las salidas tipo PWM de los dos controladores Mini y Maxi es la misma utilizada en la Figura 6.

- **Subcircuitos**

Se han diseñado subcircuitos, los cuales poseen internamente los esquemas electrónicos de entradas y salidas mencionados en el apartado anterior. Dichos subcircuitos permiten cumplir con el funcionamiento de Controllino en base a las características técnicas proporcionadas por el fabricante.

- **Controllino Mini**

Subcircuito denominado “C.I CONTROLLINO MINI” (ver Figura 8), internamente posee los circuitos electrónicos que permiten el manejo de las entradas digitales (A0-A7), los mismos que son alimentados con 24 [Vdc] y poseen circuitos de conversión analógica/digital para ingresar señales a la unidad central de procesamiento (MICRO 0-7) de un voltaje reducido a 5 [Vdc] con un límite máximo de 5.5 [Vdc]. El mismo subcircuito posee las configuraciones para activación y desactivación (MICRO 8-15) de los diferentes tipos de salida. Adicionalmente, para la obtención de voltajes de la salida tipo mosfet (D0-D7) y salida tipo relé (RELAY-OUT 0-5) con conexión directa al panel frontal, se realizaron indicadores de activación y desactivación de las salidas (INDICADOR 0-7). Además, se disponen pines de conexión común del panel frontal para salida tipo relé (CMM0-5) y la alimentación principal del controlador de 24 [Vdc] para obtener voltajes de salida de 5 [Vdc] y 3.3 [Vdc] para distintas aplicaciones.

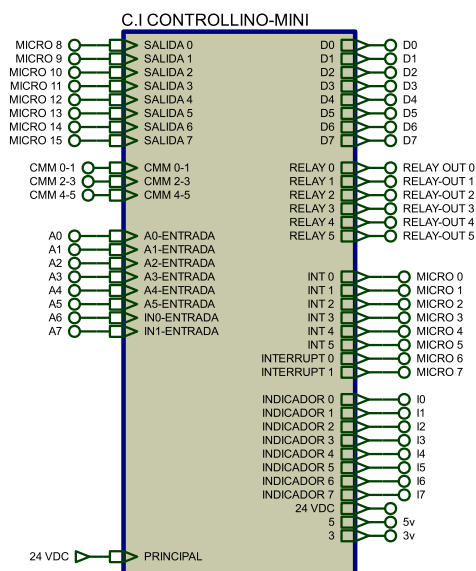


Figura 8. Subcircuito Controllino Mini

- **Controllino Maxi**

Subcircuito denominado “C.I CONTROLLINO MAXI” (ver Figura 9), internamente posee los circuitos electrónicos que permiten el manejo de las entradas digitales (I0-I15), los mismos que son alimentados con 24 [Vdc] y poseen circuitos de conversión analógica/digital para ingresar señales a la unidad central de procesamiento (MICRO 0-10) de un voltaje reducido a 5 [Vdc] con un límite máximo de 5.5 [Vdc]. El mismo subcircuito posee las configuraciones para activación y desactivación (MICRO 12-33) de los diferentes tipos de salida. Adicionalmente, para la obtención de voltajes de la salida tipo mosfet (OUT1-12) y salida tipo relé (OUT_R 0-9) con conexión directa al panel frontal, se realizaron indicadores de activación y desactivación de las salidas (INDICADOR 0-21). Además, se dispone pines de conexión del panel frontal para las salidas tipo relay (IN_R 0-9) y la alimentación principal del controlador de 24 [Vdc] para obtener voltajes de salida de 5 [Vdc] y 3.3 [Vdc] para distintas aplicaciones.

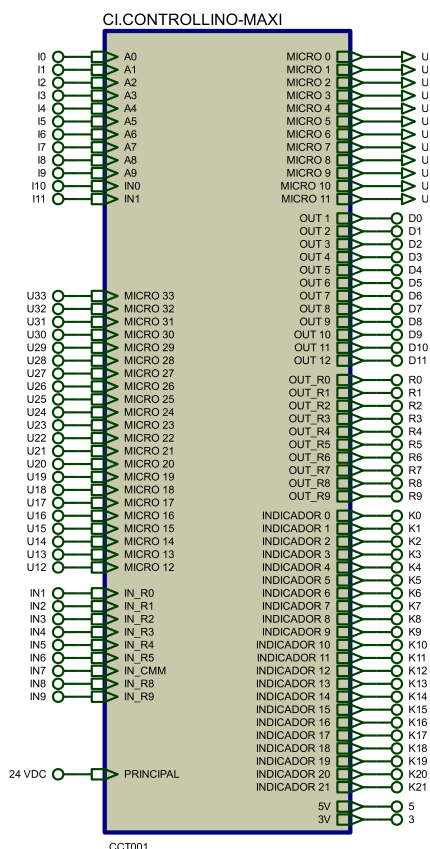


Figura 9. Subcircuito Controllino Maxi

- **Unidad Central de Procesamiento (CPU)**

La CPU de los controladores Controllino está basada en microcontroladores AVR ATMEGA, los cuales permiten realizar operaciones y acciones para el respectivo funcionamiento del controlador. Adicionalmente, reciben señales externas enviadas por periféricos para su respectivo procesamiento. Dichos microcontroladores son requeridos en base al tipo de Controllino.

Para el diseño de Controllino Mini y Controllino Maxi en base a Pinout Table proporcionado por el fabricante, se utilizaron los bloques de los microcontroladores AVR ATMEGA 328P (ver Figura 10) y AVR ATMEGA 2560 (ver Figura 11) respectivamente, juntamente con un oscilador externo XT con cristal de cuarzo de 16MHz.

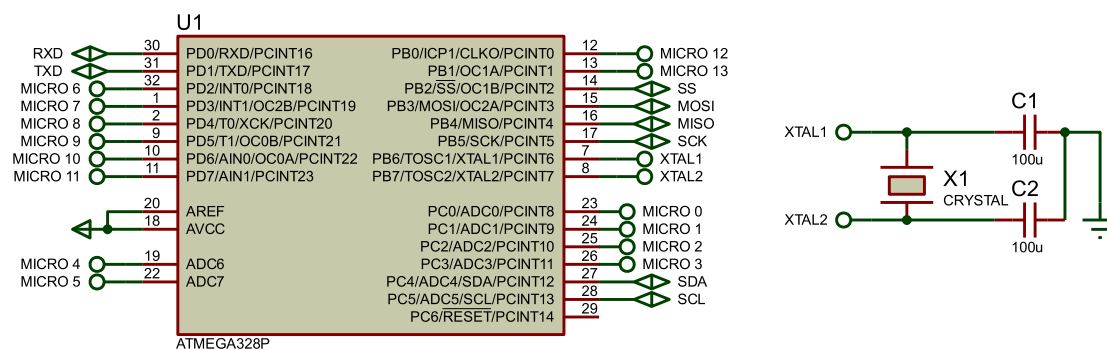


Figura 10. ATMEGA 328P y oscilador XT

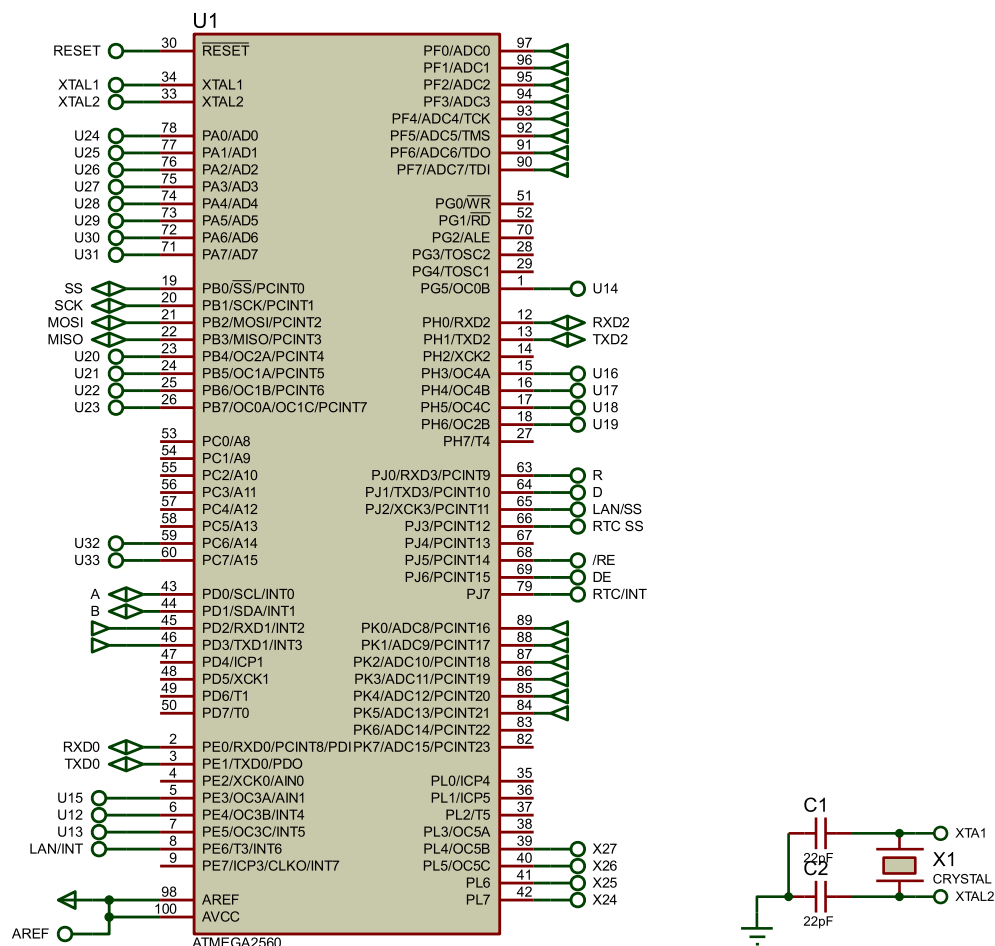


Figura 11. ATMEGA 2560 y oscilador XT

- **Panel Frontal**

Una vez que se disponen los diferentes subcircuitos organizados en bloques y los respectivos microcontroladores tanto para Controllino Mini y Controllino Maxi, se diseñó un panel frontal el cual permite la interacción y conexión de entradas y salidas con diversos dispositivos de medición y control. Estos diseños se realizaron en Proteus y se detallan a continuación.

- **Controllino Mini**

Este panel dispone borneras de conexión para salidas digitales tipo mosfet (D0, D3, D4, D6, D7), salidas PWM (D1, D2, D5), entradas analógicas/digitales (A0-IN1) y salidas digitales tipo relé (R0-R5). Adicionalmente, se disponen terminales para conexión directa a la unidad central de procesamiento. Dicho panel se puede observar en la Figura 12.

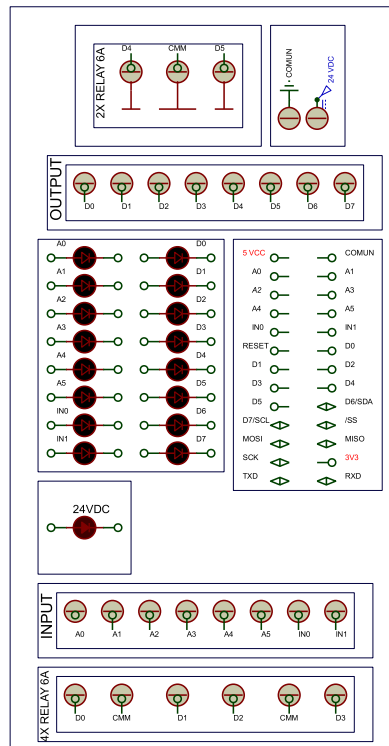


Figura 12. Panel Controllino Mini

- **Controllino Maxi**

Este panel dispone borneras de conexión para salidas digitales tipo mosfet (D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11) las mismas que también permiten disponer señales PWM, salidas digitales tipo relé (R0-R7) y entradas analógicas/digitales (A0-IN1). Adicionalmente, se disponen terminales de conexión directa a la unidad central de procesamiento. Dicho panel se puede observar en la Figura 13.

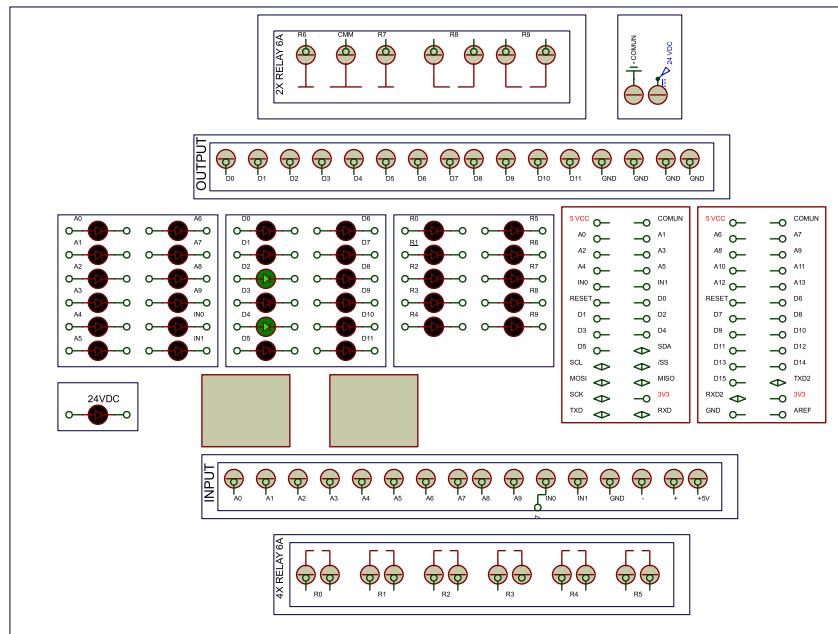


Figura 13. Panel Controllino Maxi

3.3 Algoritmos de Control

Una vez diseñados los circuitos en Proteus, se desarrollaron los respectivos algoritmos de control mediante lenguaje C para sistemas embebidos o microcontroladores, además se utilizaron instrucciones mediante librerías del compilador IDE de Arduino.

- **Librerías e instrucciones**
 - **Configuración inicial**

Antes de empezar con la respectiva programación se seleccionó en la plataforma de programación la placa de desarrollo Controllino tal cual lo muestra la Figura 14.

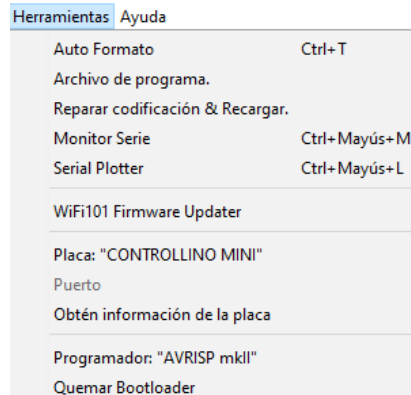


Figura 14. Selección de controlador.

Adicionalmente, para el reconocimiento del controlador y su respectiva comunicación se incluyen las librerías “**Controllino.h**” y “**SPI.h**” [7], como se muestra en el siguiente ejemplo:

```
'Librería de Comunicación
#include <SPI.h>
'Librería de Controllino.
#include <Controllino.h>
```

Para la comunicación I2C [8] entre Controllino Mini [9] y el LCD 16x2, se utilizó la librería “**LiquidCrystal_I2C.h**”, la cual permite el envío de datos del controlador hacia el módulo de comunicación PCF8574P para presentación de los datos en el LCD. Además, mediante la instrucción “**LiquidCrystal_I2C lcd(lcd_dirección, lcd_columna, lcd_fila)**” se realizó la configuración de inicialización de comunicación I2C para el LCD.

En el siguiente ejemplo se muestra la utilización de la librería e instrucción mencionadas anteriormente:

```
'Librería de comunicación I2C
#include<LiquidCrystal_I2C.h>
'Inicialización comunicación I2C en LCD.
LiquidCrystal_I2C lcd(0x27,16,2)
```

Para la creación de puertos seriales adicionales se utilizó la librería “**SoftwareSerial**” [10] ya que el Controllino Mini posee un solo puerto de comunicación serial UART. Mediante la instrucción “**SoftwareSerial función (RXD, TXD)**”, se logró convertir los pines digitales indicados como RXD y TXD en puertos seriales. Además, se debe configurar la velocidad de transmisión de

los datos mediante la instrucción **“función.begin(Baudios)”**. Las distintas velocidades se pueden observar en la Tabla 1.

Tabla 1. Velocidades de transmisión de datos

Baudios (bps)	
2.4k	19.2k
4.8k	28.8k
9.6k	38.4k
14.4k	57.6k

Los puertos seriales adicionales se utilizaron para la comunicación entre la Interfaz Humano Máquina HMI y el respectivo controlador, los cuales serán de recepción y transmisión independientes de datos. En el siguiente ejemplo se muestra la utilización de la librería e instrucción mencionadas anteriormente:

‘Librería de creación de puertos seriales.

#include <SoftwareSerial.h>

‘Instrucción de asignación de pines seriales.

SoftwareSerial sms(11,12)

‘Instrucción para velocidad de transmisión

sms.begin(9600)

- **Salidas y entradas I/O**

Para la declaración de entradas y salidas, se utilizó la instrucción **“pinMode”**, **“DDRx”** y **“PORTx”** tomando en cuenta el Pinout Table de los controladores (**ANEXO A**). La información referente a los pines de entrada y salida de cada controlador se puede observar en la Tabla 2 y Tabla 3.

Tabla 2. Pines Controllino Mini, ATMEGA 328P TQFP y Arduino Uno

OUTPUT			INPUT		
ATMEGA328P TFQP	CONTROLLINO	ARDUINO UNO R3	ATMEGA 328 TFQP	CONTROLLINO	ARDUINO UNO R3
PD4	D0	4	PC0	A0	A0
PD5	D1	5	PC1	A1	A1
PD6	D2	6	PC2	A2	A2
PD7	D3	7	PC3	A3	A3
PB0	D4	8	ADC6	A4	A6
PB1	D5	9	ADC7	A5	A7
PC4	D6	18			
PC5	D7	19			

Tabla 3. Pines Controllino Maxi, ATMEGA 2560 TQFP y Arduino Mega

OUTPUT			INPUT		
ATMEGA328P TFQP	CONTROLLINO	ARDUINO MEGA	ATMEGA 328 TFQP	CONTROLLINO	ARDUINO MEGA
PE4	D0	2	PF0	A0	A0
PE5	D1	3	PF1	A1	A1
PG5	D2	4	PF2	A2	A2
PE3	D3	5	PF3	A3	A3
PH3	D4	6	PF4	A4	A4
PH4	D5	7	PF5	A5	A5
PH5	D6	8	PF6	A6	A6
PH6	D7	9	PF7	A7	A7
PB4	D8	10	PK0	A8	A8
PB5	D9	11	PK1	A9	A9
PB6	D10	12	PD3	IN0	TX1
PB7	D11	13	PD2	IN1	RX1

- **Envío de datos**

Para el envío de datos de las variables de los sistemas de control industrial a desarrollar se utilizó la instrucción “**Serial.print()**”, la cual imprime caracteres ASCII [11] en forma legible por medio de los pines del puerto serial TXD. Adicionalmente, se utilizó la instrucción “**Serial.write()**”, la cual permite imprimir un byte (8 bits) por el puerto TXD. En el siguiente ejemplo se muestra la utilización de las instrucciones mencionadas anteriormente:

Serial.print()

'Permite imprimir caracteres por el puerto serial.

Serial.print("n0.val=")

Serial.print(variable)

Serial.write()

'Permite imprimir datos tipo byte por el puerto serial.

Serial.write(0xff)

- **Recepción de datos**

Para la recepción de los respectivos datos, registrados en el puerto serial, se utilizó la instrucción "**Serial.available**", habilitando la lectura de los caracteres almacenados en el búfer de recepción del puerto serie. En el siguiente ejemplo se muestra la utilización de las instrucciones mencionadas anteriormente:

'Lectura serial y almacenar en la variable

while(Serial.available()){

- **Control ON-OFF de Temperatura**

El controlador todo-nada es uno de los más utilizados en los procesos industriales que no requieren elevada precisión. Se caracteriza por la naturaleza oscilante de los actuadores al momento de corregir el error y por su banda de histéresis.

Para el funcionamiento del controlador se establece una banda de histéresis, dentro de la cual oscila la variable a controlar en el proceso. Los actuadores realizan las acciones pertinentes de activación y desactivación durante la oscilación de la variable con el fin de reducir el error.

En la Figura 15, se muestra la oscilación de la variable controlada con respecto al valor de consigna (set point) y la activación/desactivación de la salida (actuadores) en caso de sobrepasar la variable del proceso los límites de la banda de histéresis.

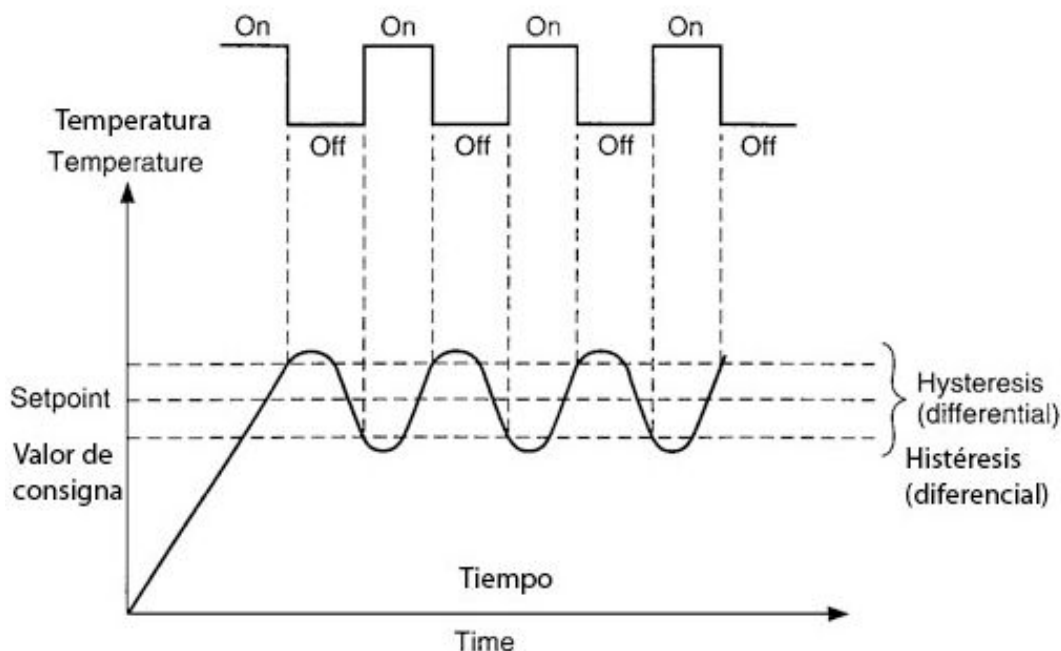


Figura 15. Diagrama de Control on-off por histéresis
Fuente: Modos de control de los controladores industriales

El diagrama de flujo y el algoritmo desarrollado para Controllino Mini se pueden observar en el ANEXO C.1 y en ANEXO D.1 respectivamente.

- **Control de un motor de dos velocidades**

Motor Dhalander posee dos velocidades con una relación 2:1 en el número de polos y utilizado para propósito general en aplicaciones que requieran accionamientos no forzados. Para el sistema de control del motor se utilizaron las relaciones para velocidad rápida y velocidad lenta en función de la temperatura.

Para velocidad rápida se utilizó el circuito de la Figura 16, cortocircuitando los bornes (U2,V2,W2) mediante un contactor KM3. El funcionamiento del circuito de potencia permite el accionamiento consecutivo del contactor KM1, con lo cual se logra evitar fluctuaciones y picos de voltaje en los devanados del motor.

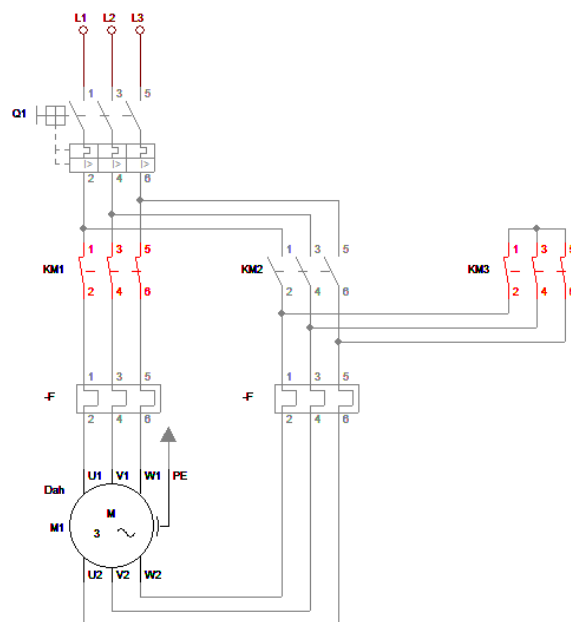


Figura 16. Circuito de potencia para velocidad rápida

Para velocidad lenta se utilizó el diagrama de potencia de la Figura 17, en el cual se realiza el accionamiento del contactor KM2.

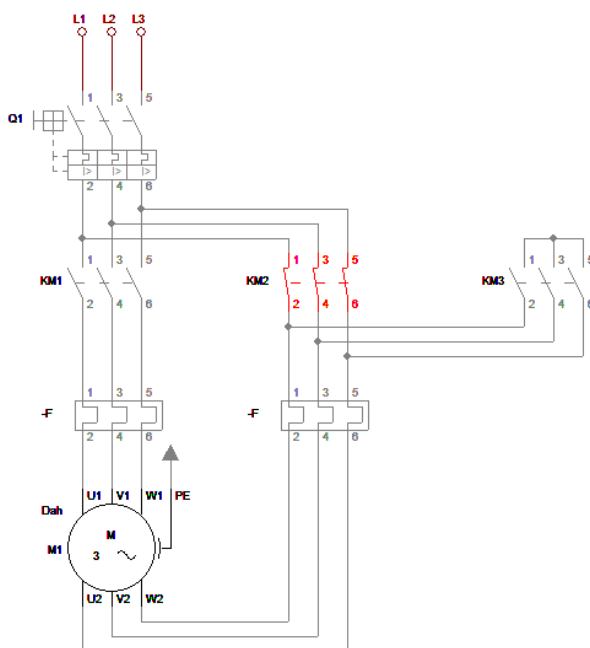


Figura 17. Circuito de potencia para velocidad lenta

El diagrama de flujo y el algoritmo desarrollado para Controllino Maxi se pueden observar en el ANEXO C.2 y en ANEXO D.2 respectivamente.

3.4 Interfaz Humano Máquina HMI

La Interfaz Humano Máquina HMI permite mostrar información en tiempo real para que el operador pueda controlar, monitorear y supervisar el proceso de forma adecuada y eficiente. Por tal motivo, el desarrollo de la respectiva HMI de los sistemas de control establecidos se realizó mediante el software Nextion Editor, el cual es compatible con la plataforma Arduino IDE.

- **Selección inicial**

Para iniciar con el desarrollo de la Interfaz Humano Máquina HMI, se elige el tipo, modelo y la dirección de display. Al tratarse de un trabajo de simulación se puede elegir cualquiera de las opciones que presenta el Nextion Editor, sin embargo, para el presente proyecto se utilizó la pantalla básica NX1060P101_011 con la dirección de display 180 Horizontal. En la Figura 18, se muestra la selección de los parámetros mencionados.

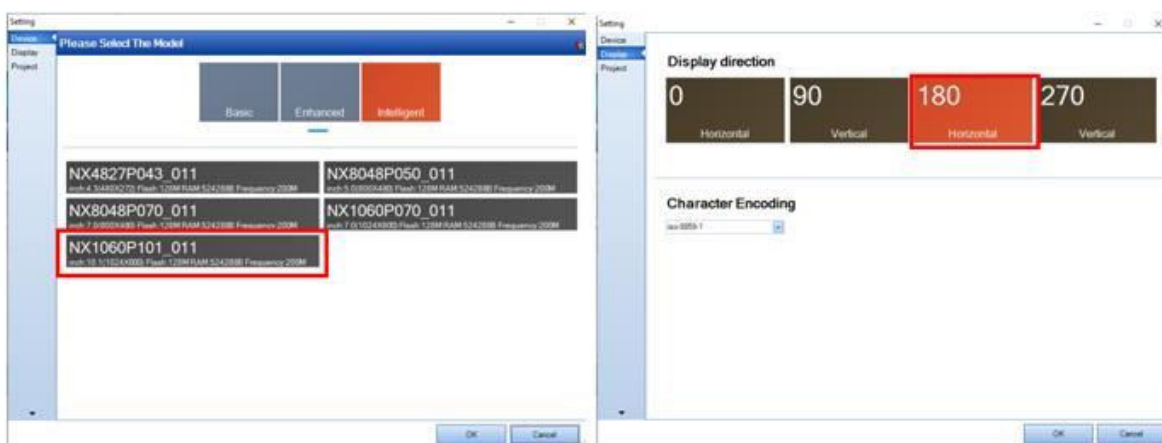


Figura 18. Selección de tipo y dirección de pantalla HMI

- **Fuentes de texto**

Las fuentes permiten mejorar la visualización del texto en base al tipo, tamaño y forma de letra, por lo tanto, se realizó la creación de fuentes de textos en base a la Tabla 4. Se añadió una carpeta dentro del proyecto para almacenar dichas fuentes y poder utilizarlas en la presentación

de las variables del proceso. Posterior a la creación de fuentes se incluyó el respectivo “id” identificador para cada fuente. En la Figura 19 se pueden observar las fuentes respectivas.

Tabla 4. Fuentes de texto

Características de las Fuentes de Texto			
Tamaño	Fuente	Rango	Codificación
16	Microsoft Sans Serif	Todos los caracteres	iso-8859-1
16	Arial Black	Todos los caracteres	iso-8859-1
24	Microsoft Sans Serif	Todos los caracteres	iso-8859-1
24	Arial Black	Todos los caracteres	iso-8859-1

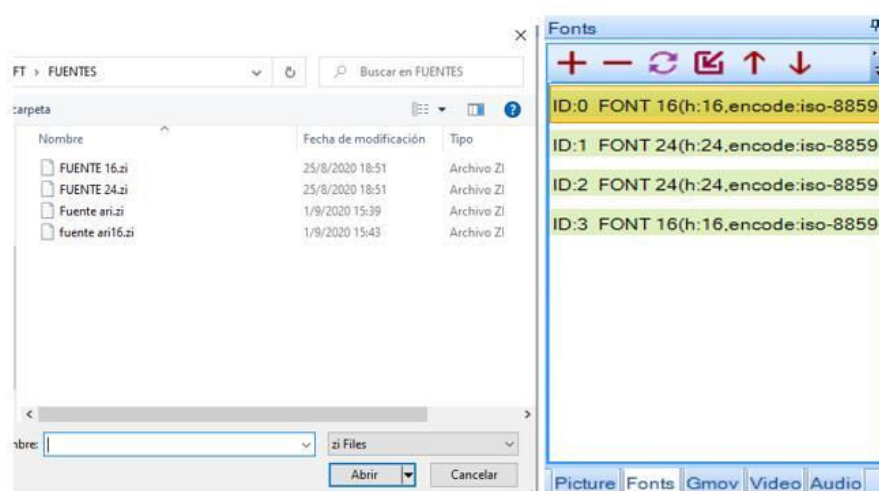


Figura 19. Fuentes de texto creadas

Finalmente, para la colocación de texto previamente se configuraron los parámetros o atributos con un número máximo de caracteres en base a la Tabla 5, adicionalmente, se ingresó el “id” identificador de la fuente creada en los parámetros de texto (ver Figura 20).

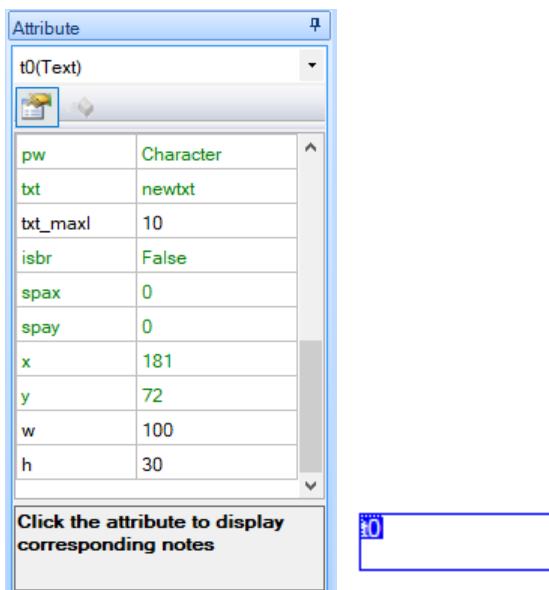


Figura 20. Atributos e identificador de texto

Tabla 5. Parámetros de texto

Texto					
ObjName	Txt_maxl	Fuente	ObjName	Txt_maxl	Fuente
txt0	50	2	txt3	10	3
txt1	60	2	txt4	10	3
txt2	300	2	txt5	10	3

- **Botones**

Al incluir botones, se modificaron los parámetros o atributos de los mismos (ver Tabla 6) y el texto identificador de la acción a realizar por cada botón. Los botones agregados tienen un “id” de identificación (ver Figura 21), al igual que las fuentes de texto. Dichos botones envían caracteres hacia Controllino por el puerto serial, los cuales permiten cumplir el evento del botón presionado gracias a las instrucciones del algoritmo de control realizado previamente.

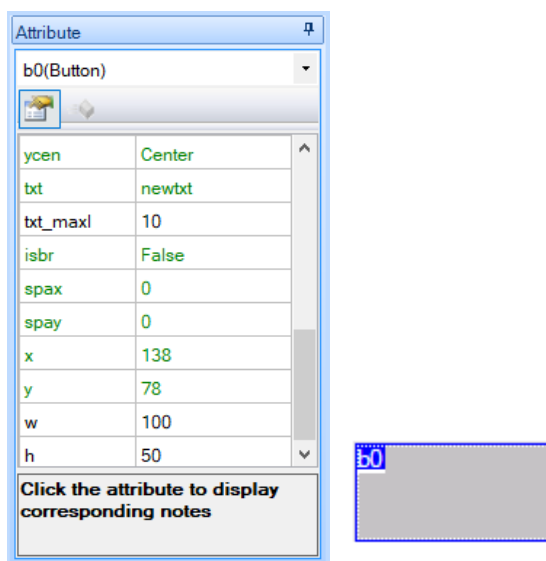


Figura 21. Atributos e identificador del botón

Tabla 6. Parámetros de los botones

Botones							
ObjName	Txt_maxl	Fuente	Nº Color	ObjName	Txt_maxl	Fuente	Nº Color
b0	10	0	50712	b3	10	0	63488
b1	10	0	50712	b4	50	0	50712
b2	10	0	63488	b5	50	0	50712

- **Indicadores**

La utilización de indicadores en la interfaz humano máquina permitió la visualización de las variables del sistema (temperatura) para evaluar su comportamiento a través del tiempo. Para esto fue necesario realizar una configuración de parámetros o atributos en base a la Tabla 7 y la inicialización y configuración de las variables tal cual se muestra en la Figura 22. Cabe resaltar que los indicadores reciben los datos de Controllino mediante comunicación serial para su respectiva visualización en la HMI.

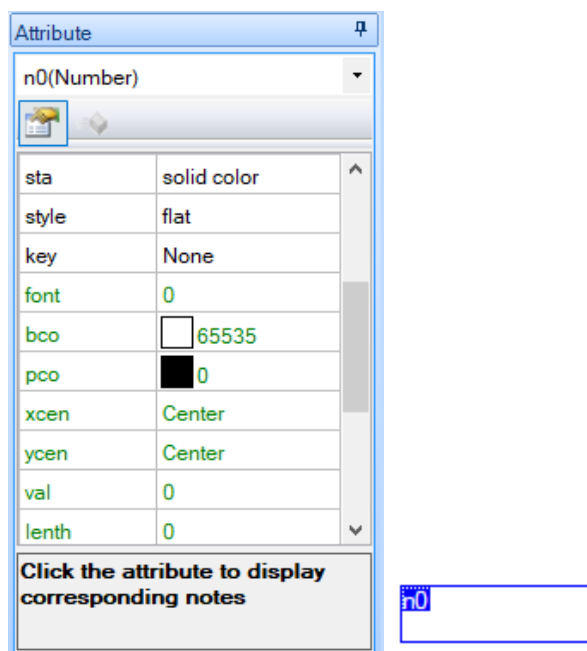


Figura 22. Atributos e identificador de los indicadores

Tabla 7. Parámetros de los indicadores

Indicadores					
ObjName	val	Fuente	ObjName	val	Fuente
n0	0	0	n3	0	0
n1	0	0	n4	0	0
n2	0	0	-	-	-

- **Ventanas HMI**

Una vez configurados los respectivos parámetros de las herramientas de la interfaz humano máquina, se diseñaron las dos ventanas para los sistema de control de Controllino Mini y Controllino Maxi. Dichas ventanas se pueden visualizar en la Figura 23 y Figura 24.



**ESCUELA POLITÈCNICA NACIONAL
ESUELA DE FORMACIÒN DE TECNÒLOGOS**

**SIMULACIÒN CONTROL ON-OFF DE TEMPERATURA
CONTROLLINO-MINI-INTERFAZ HUMANO MÀQUINA HMI**

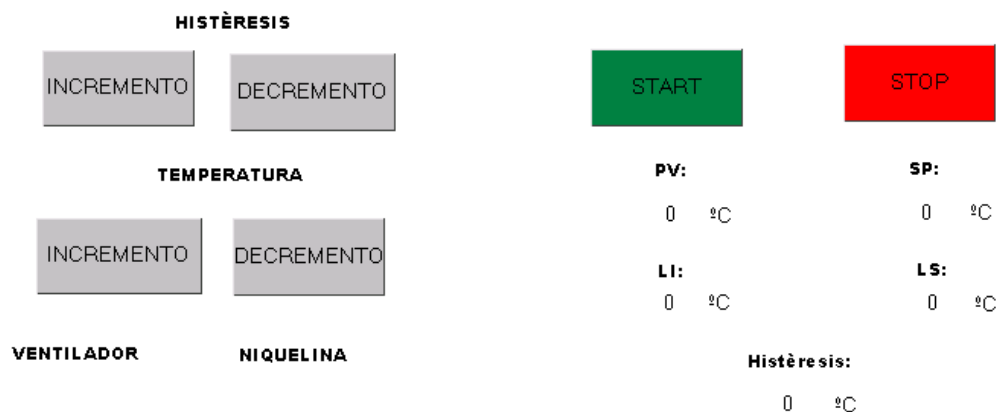


Figura 23. Ventana de Control On-Off de Temperatura



**ESCUELA POLITECNICA NACIONAL
ESCUELA DE FORMACIÒN DE TECNÒLOGOS**

**CONTROL DE UN MOTOR DE DOS VELOCIDADES
SIMULACIÒN CONTROLLINO MAXI-HMI NEXTION**

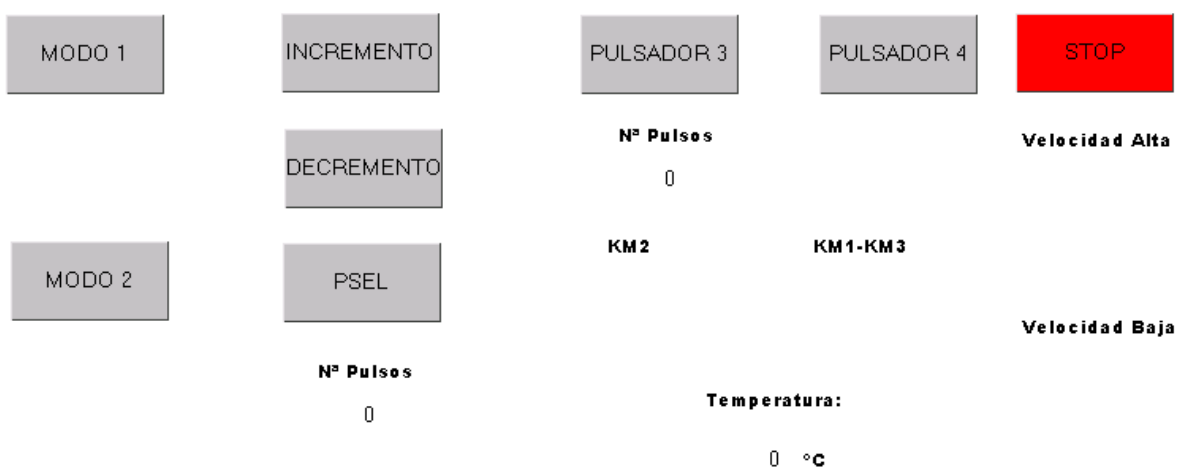


Figura 24. Ventana de Control de un motor de dos velocidades

3.5 Pruebas de funcionamiento

En esta sección se procede a verificar que los sistemas simulados funcionen de forma correcta. Por tal motivo, se realizaron pruebas de funcionamiento de los controladores Controllino y sus respectivos algoritmos de control e interfaces humano máquina. Las pruebas realizadas se listan a continuación:

- a) Pruebas de entrada y salida
- b) Pruebas de comunicación Controllino-HMI
- c) Pruebas de los sistemas de control

- **Pruebas de entradas y salidas.**

- **Entradas analógicas**

Se utilizó un potenciómetro, el cual se encuentra conectado a una fuente de 24[Vdc], la misma que simulará la variación de voltaje de la señal analógica, (ver Figura 25). Debido al subcircuito de acondicionamiento desarrollado, las señales de voltaje proporcionadas en el rango de 0 [Vdc] a 24 [Vdc] se convirtieron en valores de salida admisibles por la unidad central de procesamiento en un rango de 0 [Vdc] a 5 [Vdc]. El registro de valores de entrada y salida dentro del rango establecido se puede observar en la Tabla 8.

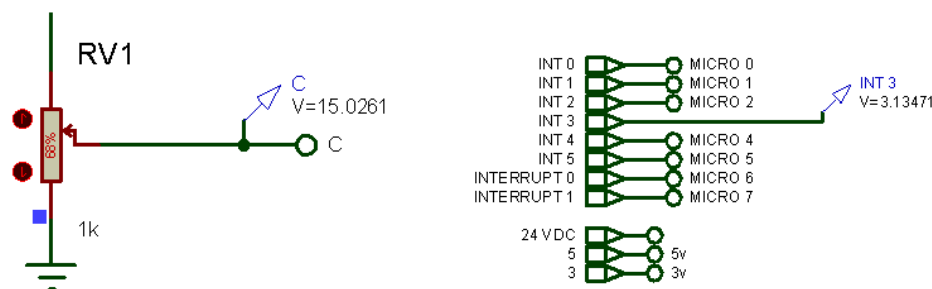


Figura 25. Valores de voltaje en entrada analógica

Tabla 8. Valores registrados para entradas analógicas

Vin (V)	Vout	Vout	Error %
	Ideal (V)	Proteus (V)	
24.0	5.0	5.0	0
15.0	3.1	3.1	0
12.0	2.6	2.6	0
8.0	1.7	1.7	0
5.0	1.0	1.0	0
2.5	0.5	0.5	0
0.0	0.0	0.0	0

o Salidas digitales

Se realizó la medición de voltaje a la salida de los pines digitales (ver Figura 26) del panel frontal de Controllino para comprobar el voltaje máximo de alimentación de 24[Vdc]. Se puede observar en la Tabla 9 los respectivos valores registrados los cuales cumplen con los límites de error establecidos menores al 1%.

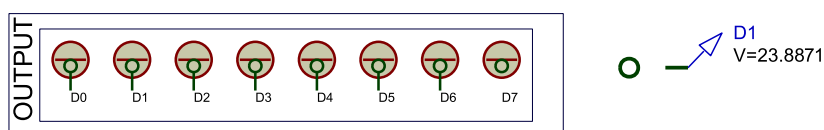


Figura 26. Voltaje de salida en pines digitales

Tabla 9. Valores registrados para salidas digitales

Pin	Vout	Vout	Error %
	Ideal (V)	Proteus (V)	
D0	24.0	23.8	0.8
D1	24.0	23.8	0.8
D2	24.0	23.8	0.8
D3	24.0	23.8	0.8

Pin	Vout	Vout	Error
	Ideal (V)	Proteus (V)	%
D4	24.0	23.8	0.8
D5	24.0	23.8	0.8
D6	24.0	23.8	0.8
D7	24.0	23.8	0.8

○ Salidas tipo Relé.

En la Figura 27, se visualiza la prueba de encendido de la salida tipo relé, realizando conexiones en el panel frontal con una fuente de voltaje de 120 [Vac] y las respectivas cargas. La activación correcta de las cargas mediante las salidas tipo relay (pin D0 y pin D1) se puede verificar en la gráfica presentada en la Figura 29.

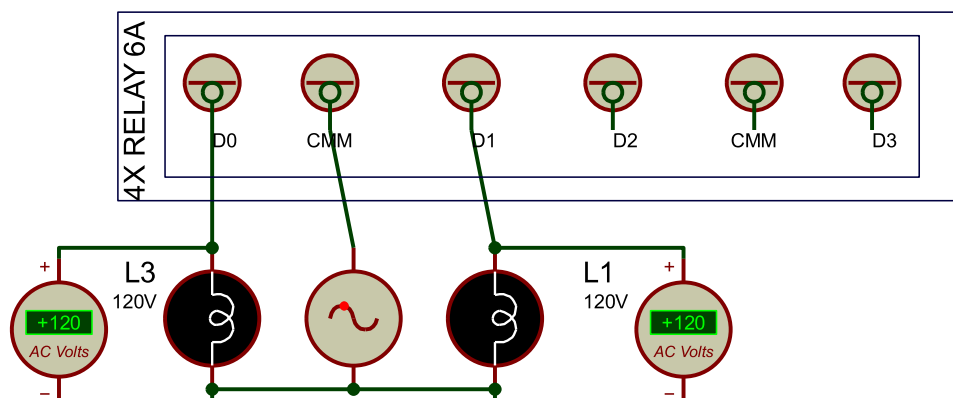


Figura 27. Relé encendido.

○ Salidas PWM

Para verificar el funcionamiento de las salidas PWM se utilizó un osciloscopio de Proteus mediante el cual se pudo visualizar la forma de onda y su respectiva modulación. La señal con su respectivos valores se puede observar en la Figura 28.

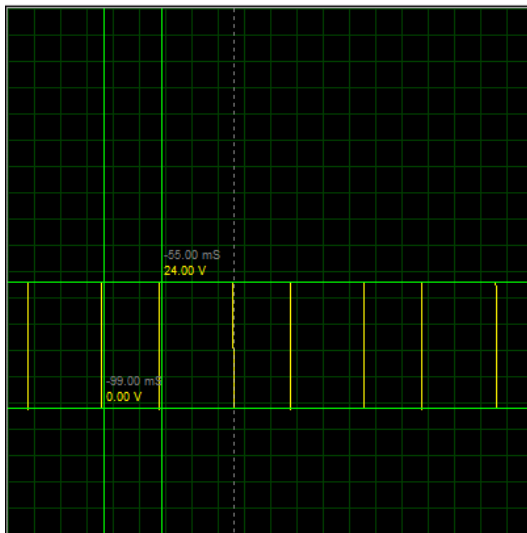


Figura 28. Señal PWM

- **Prueba de comunicación Controllino-HMI**
 - **Envío de datos Controllino-HMI**

Para la prueba de envío de datos mediante comunicación serial [12], se utilizó la herramienta “Virtual Terminal” del software Proteus v8.9. Dichos datos se pueden visualizar en la Figura 29 los cuales son enviados en código ASCII en caracteres tipo char “n4.val=1” para su decodificación por parte de la interfaz humano máquina HMI.

Figura 29. Envío de datos mediante Controllino

Los datos enviados por el Controllino y recibidos por la ventana HMI en código ASCII en caracteres hexadecimal “6E 34 2E 76 61 6C 3D 3D FF FF FF” se pueden observar en la Figura 30.



Figura 30. Recepción de datos por parte de la HMI

○ Transmisión HMI-Controllino

Finalmente, para verificar el envío de datos por medio de comunicación serial desde la ventana HMI hacia el Controllino, se utilizó una de las herramientas de dicha ventana en la cual mediante el accionamiento de un botón se puede observar el dato enviado. En la Figura 31, se visualiza el envío del dato “31” en código ASCII formato hexadecimal al presionar el botón “INCREMENTO”.

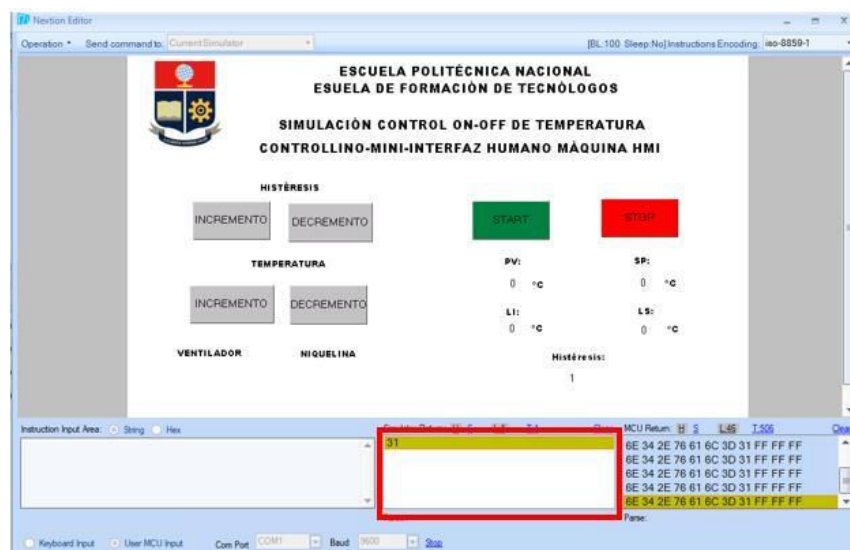


Figura 31. Transmisión de dato HMI

El dato enviado por parte de la HMI es recibido en código ASCII formato char mediante comunicación serial por el controlador tal cual se muestra en la Figura 32, verificando de esta manera la correcta comunicación entre los dispositivos de control y supervisión.

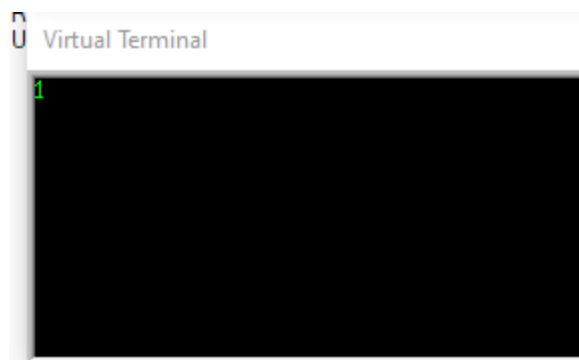


Figura 32. Recepción de datos en el controlador

- **Pruebas de los sistemas de control**

Finalmente, se establecieron las respectivas pruebas presentadas en la Tabla 10 para verificar el correcto funcionamiento de los sistemas de control.

Tabla 10. Pruebas de los sistemas de control

Prueba	Funcionalidad		
	100 %	50%	0%
Entradas y Salidas Controllino	✓		
Indicadores visuales de salidas del Panel frontal	✓		
Comunicación I2C Controllino-LCD	✓		
Visualización de variables en LCD 16x2	✓		
Comunicación Serial Controllino-HMI	✓		
Indicadores visuales de salidas del HMI	✓		

Prueba	Funcionalidad		
	100 %	50%	0%
Botones de HMI	✓		
Indicadores de variables del sistema en el HMI	✓		
Envió de datos desde Controllino a HMI	✓		
Envió de datos desde HMI a Controllino	✓		

A continuación, se presenta el funcionamiento de las simulaciones de los sistemas de control mediante Controllino y su respectiva Interfaz Humano Máquina HMI:

○ Control ON-OFF Temperatura

En la Figura 33, se observa el funcionamiento de la simulación del sistema de control on-off de temperatura, en el cual se muestran las variables que intervienen en el proceso (Histéresis (H), Process Value (PV), Set Point (SP), Límite Superior (LS) y Límite Inferior (LI)) de manera simultánea mediante la Interfaz Humano Máquina HMI desarrollada y el LCD 16x2.

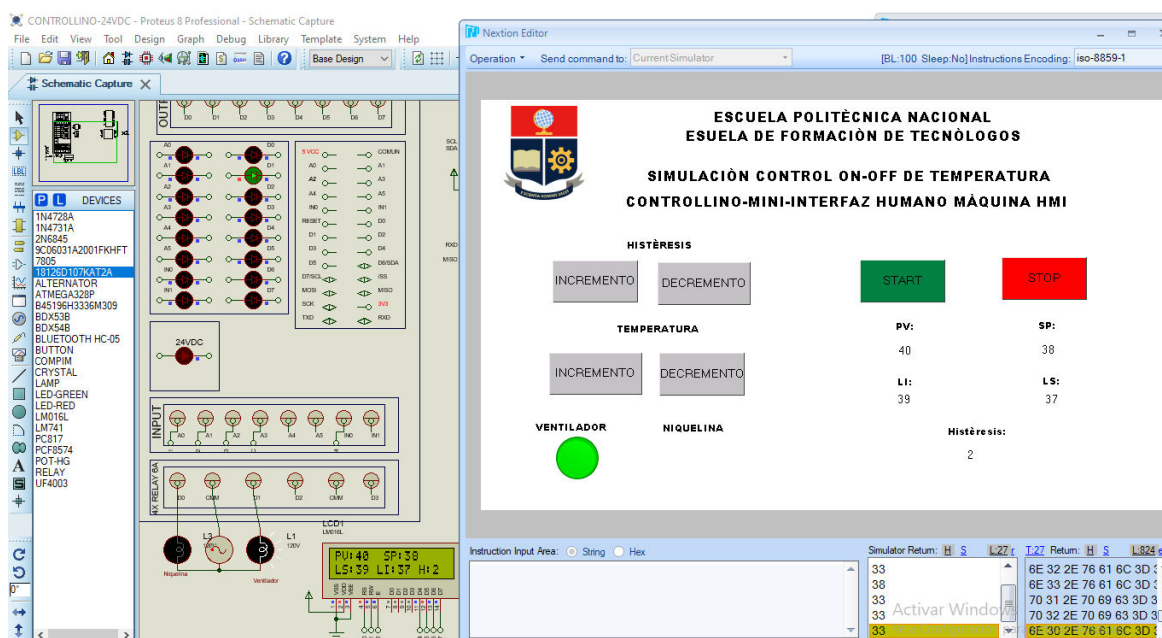


Figura 33. Simulación del sistema de Control On-Off de Temperatura

Se comprobó que el sistema simulado realiza el control de temperatura mediante una banda de histéresis, cuyo rango de seguridad puede ser ingresado mediante pulsadores conectados al panel frontal de Controllino Mini, es decir, de forma local. Además, se verificó que mientras la temperatura esté fuera de la banda de histéresis el sistema activa los respectivos actuadores (Ventilador y Niquelina) para controlar la temperatura.

Finalmente, también se verificó que todas las acciones descritas anteriormente se puedan cumplir desde la Interfaz Humano Máquina HMI.

- **Control de un motor de dos velocidades**

En la Figura 34, se observa el funcionamiento de la simulación del sistema de control de un motor de dos velocidades, en el cual se muestran las variables que intervienen en el proceso (Temperatura, Velocidad, N° Pulsos de P3 para mando alternado y N° Pulsos de PSEL para selección de velocidad), las mismas que son visualizadas de manera simultánea mediante la Interfaz Humano Máquina HMI desarrollada y el LCD 16x2.

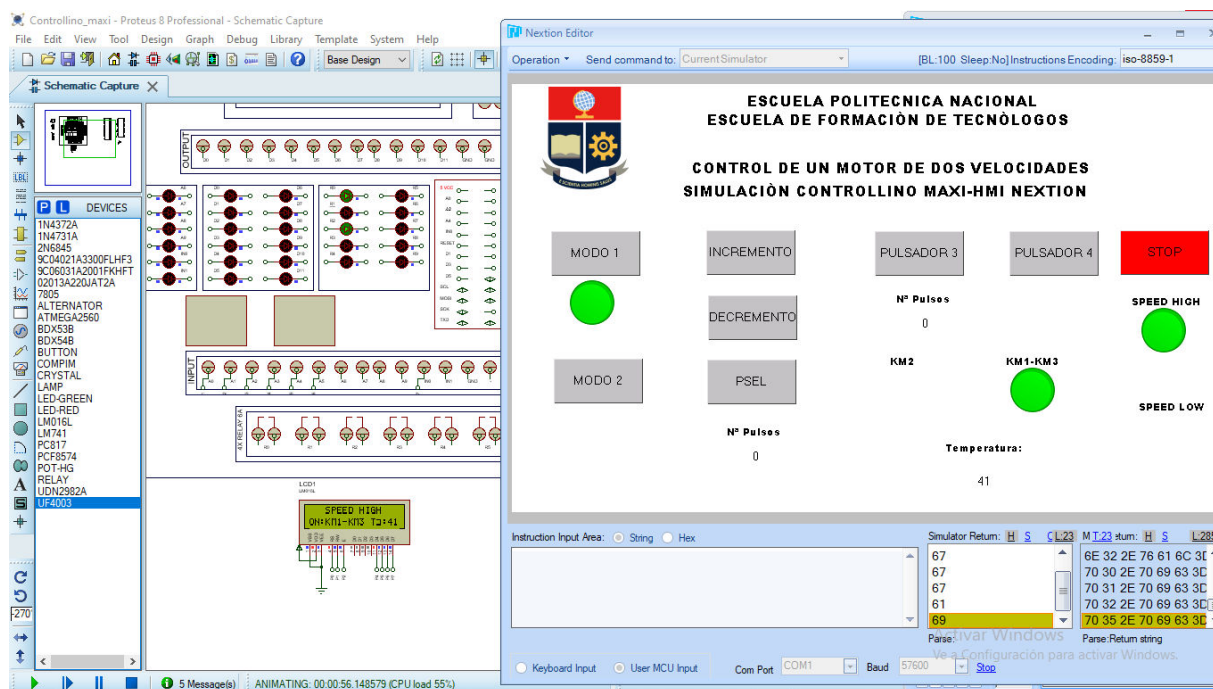


Figura 34. Simulación del control de un motor de dos velocidades

Para este sistema se crearon dos modos de funcionamiento dependientes de la temperatura ingresada por los pulsadores conectados al panel frontal de Controllino Maxi y desde la Interfaz Humano Máquina HMI según el respectivo diagrama de flujo del algoritmo desarrollado (ANEXO

C.1). Además, en el panel frontal de Controllino Maxi se visualizaron las salidas activadas tipo relé.

Para observar el respectivo funcionamiento de la simulación de las aplicaciones mediante Controllino Mini y Maxi se puede acceder al siguiente link:

<https://1drv.ms/u/s!AtlDqOH7YULugbcDj9fOEmWJM1I6vA?e=qZ3zMv>

Finalmente, una vez realizadas las pruebas de funcionamiento, se elaboró el manual de usuario de las respectivas simulaciones, el cual se encuentra detallado en el ANEXO E, el mismo que podría ser utilizado para una posible implementación de los controladores Controllino Mini y Controllino Maxi

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El diseño de los controladores Controllino permite interactuar a través de la simulación desarrollada con toda la gama de posibilidades y beneficios que ofrecen los PLC's de código abierto para aplicaciones industriales.
- El aislamiento de los pines de salida del microcontrolador se realiza para precautelar la vida útil de la unidad central de procesamiento (CPU) del controlador, la cual recibe y procesa las señales para posterior ejecución de las acciones de control.
- El diseño de entradas digitales/analógicas multiplexadas facilita la operación, conexión de dispositivos y evita la utilización de módulos externos en comparación a PLC's tradicionales, lo cual es una gran ventaja de los controladores Controllino basado en código abierto.
- El trabajo desarrollado puede ser utilizado para la futura implementación de módulos didácticos basados en Controllino, lo cual permitiría reducir costos de implementación que se presentan utilizando PLC's tradicionales.
- Las salidas tipo mosfet y relay de Controllino permiten la manipulación de diferentes tipos de voltaje en un solo controlador en comparación con PLC's tradicionales de gama baja los cuales manejan un solo tipo de salida.
- La velocidad de comunicación del puerto serial influye en la adquisición y transmisión del número de datos por segundo, por lo cual en el presente proyecto se utilizaron velocidades de 9600 y 57000 baudios para los sistemas de control simulados, ya que con estos valores se evitó la pérdida de información.
- Controllino utiliza software de programación de código abierto, lo que facilita la interacción con diferentes interfaces humano máquina para aplicaciones de carácter industrial por disponer comunicación serial y ethernet.
- Controllino permite la utilización de protocolos de comunicación industrial tales como Profibus y Profinet sin ningún módulo externo de alto costo, lo cual permite la comunicación con PLC's e Interfaz Humano Máquina HMI multimarca.
- Los PLC's de open source o código abierto Controllino pueden cumplir con todos los requerimientos de los sistemas de control desarrollados con PLC's tradicionales, con la gran ventaja que utilizan dispositivos económicos y disponibles en el mercado.

4.2 Recomendaciones

- Para la posible implementación de los módulos Controllino tomar en cuenta que las salidas PWM (Pulse-Width-Modulation) poseen un rango máximo de 80% de duty cycle debido al diseño de la etapa de potencia del controlador.
- Para la comunicación entre las variables de la interfaz humano máquina es necesario colocar los caracteres especificados en la lista de instrucciones de Nextion editor.
- Revisar la hoja técnica de los PLC's Controllino para la referencia del rango de voltaje máximo admisible y realizar el diseño adyacente al mismo.
- Para la comunicación mediante ethernet es necesario añadir librerías que permitan enviar y recibir datos entre Controllino y cualquier otro tipo de controladores.
- Para configuración de pines y pines Controllino se requiere revisar el pinout table de dicho controlador, tomando en cuenta el tipo del mismo con el fin de cumplir con todos los parámetros técnicos establecidos.
- Para verificar el correcto funcionamiento de la simulación desarrollada para Controllino Mini y Maxi, se debe tener en cuenta que las prestaciones del computador sean adecuadas evitando de esta manera la pérdida de información.
- Realizar respaldo de información del algoritmo de control debido, a que no existe forma de descargar el programa de la memoria del controlador al ser basado en un microcontrolador que no dispone de esta opción.
- Al realizar uso de los terminales del controlador, tener en consideración que dichos pines manejan voltajes de 5 [Vdc] TTL por su conexión directa al microcontrolador.
- El trabajo desarrollado puede ser mejorado utilizando protocolos de comunicación disponibles en el Controllino, como I2C, SPI y UART, lo cual puede complementar las prestaciones de los módulos Controllino y ampliar su campo de aplicación.
- Para la implementación de los módulos Controllino diseñados y simulados en el presente trabajo tomar en cuenta todas las características y parámetros de los mismos, para evitar posibles errores.

5. BIBLIOGRAFÍA

- [1] Conelcom GmbH, «Controllino,» 2016. [En línea]. Available: <https://www.controllino.biz/downloads/#datasheets>. [Último acceso: 13 08 2020].
- [2] S. A. C. C. L. C. A. BYRON MEDINA DELGADO, «Tecnologías de código abierto para la gestión de un proceso industrial,» *Gerencia Tecnológica Informática*, pp. 43-58, 2014.
- [3] Conelcom GmbH, «Controllino Mini,» 19 04 2016. [En línea]. Available: https://www.controllino.biz/wp-content/uploads/2018/10/MINI_DATASHEET-19-04-20161.pdf. [Último acceso: 12 08 2020].
- [4] Conelcom GmbH, «Controllino Maxi,» 19 04 2016. [En línea]. Available: https://www.controllino.biz/wp-content/uploads/2018/10/MAXI_DATASHEET-19-04-20161.pdf. [Último acceso: 12 08 2020].
- [5] SHARP, «Device Specification for photocoupler».
- [6] I. D. S. T. N. S. J. A. R. A. R. Rosnazri Ali, «Design of High-Side MOSFET Driver Using Discrete Components for 24V Operation,» de *The 4th International Power Engineering and Optimization Conf. (PEOCO2010)*, Malaysia, 2010.
- [7] Controllino, «Controllino,» 2016. [En línea]. Available: <https://www.controllino.biz/downloads/#libraries>. [Último acceso: 13 07 2020].
- [8] J. M. R. Gutiérrez, «Academia,» Agosto 2012. [En línea]. Available: https://www.academia.edu/11795143/Bus_I2C_de_Arduino. [Último acceso: 16 08 2020].
- [9] Controllino, «Controllino Mini,» 19 04 2016. [En línea]. Available: https://www.controllino.biz/wp-content/uploads/2018/10/MINI_DATASHEET-19-04-20161.pdf.
- [10] Arduino , «Arduino,» 09 03 2020. [En línea]. Available: <https://www.arduino.cc/en/Reference/SoftwareSerial>. [Último acceso: 15 08 2020].
- [11] J. A. V. Frank Benalcazar, «Comunicación UART de la PC con la Plataforma Interactiva creada, Basada en la tarjeta de desarrollo AVR BUTTERFLY a través del puerto Serie RS232,» Guayaquil, 2012.
- [12] U. Osiogogu, «Seminar Paper on Serial Communication,» 2015.

- [13] Nextion, «Nextion Editor,» 2011. [En línea]. Available: <https://nextion.tech/instruction-set/>.
- [14] V. Rossano, Proteus VSM, Buenos Aires, 2013.
- [15] R. R. Leo, «DISEÑO DE UN PLC INDUSTRIAL USANDO,» Pamplona, 2015.
- [16] O. T. Artero, «Arduino curso practico de formaciòn,» RC Libros, España, 2013.
- [17] «Indonesia Journal of Science & Technology,» 01 09 2017. [En línea]. Available: <https://ejournal.upi.edu/index.php/ijost/article/view/8019/5057>. [Último acceso: 12 07 2020].
- [18] Conelcom GmbH, «Controllino,» 2016. [En línea]. Available: <https://www.controllino.biz/downloads/#libraries>. [Último acceso: 13 07 2020].

6. ANEXOS

ANEXO A: Datasheet

Pinout Table de Controllino Mini

Pinout Table de Controllino Maxi

ANEXO B: Esquemas electrónicos

Controllino Mini 24 VDC

Controllino Maxi 24 VDC

ANEXO C: Diagramas de Flujo

Control ON-OFF de Temperatura

Control de un motor de dos velocidades

ANEXO D: Algoritmos de Control

Control ON-OFF de Temperatura

Control de un motor de dos velocidades

ANEXO E: Manual de Usuario

ANEXO A

DATASHEET

ANEXO A.1

PINOUT TABLE CONTROLLINO MINI

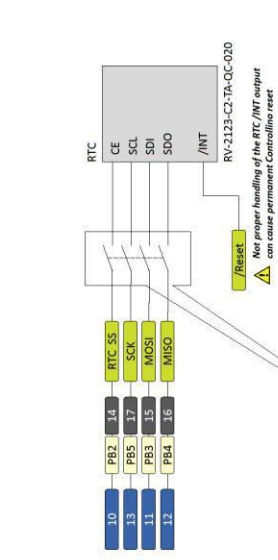
CONTRILINO FUNCTION PORT PIN

ATTACHED I/O BOARD

General Information

Pay attention

CHIP used: **Atmega 328P-AU**

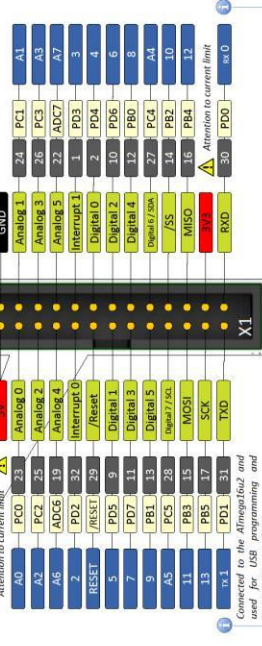


Contrilino MINI

PINOUT V1.1

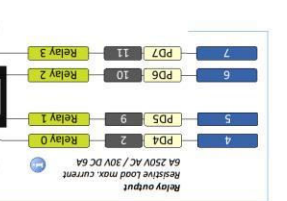
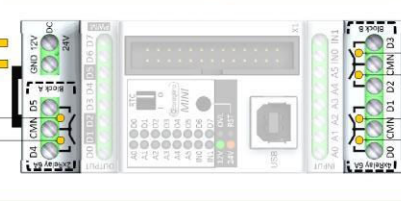
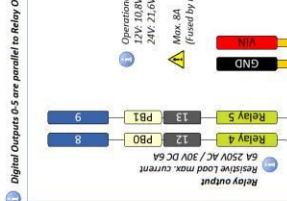
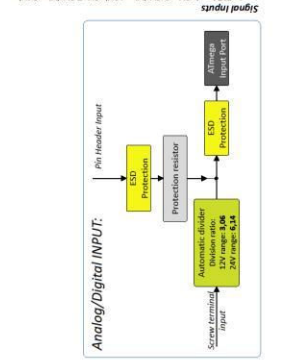
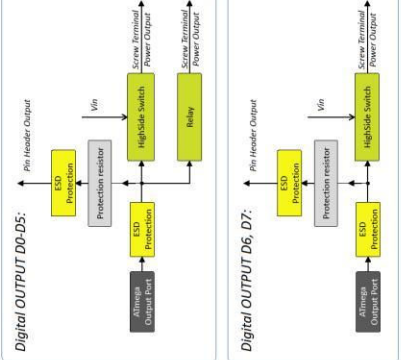
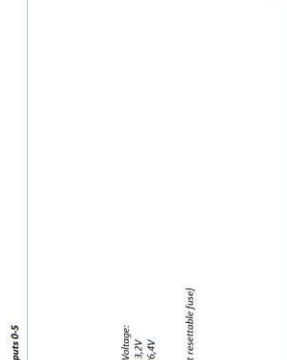
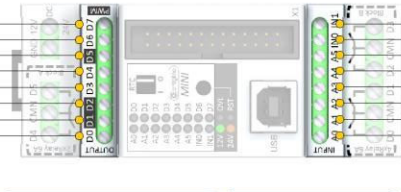


Pin Header current limit:
 Absolute max per pin 40mA
 recommended 20mA
 Absolute max 200mA for entire Pin Header
 Current limit @5V + 3V3 max 200mA (fused by resettable fuse)
 Current limit @3V3 only 150mA
 All signals are protected with serial resistance



Attention to current limit

Caution for the Atmega328P and the I2C communication with PC



Digital Outputs 0-5 are parallel to Relay Outputs 0-5

Relay output max. current: 6A 250V AC / 30V DC 6A

Resistive load max. current: 6A 250V AC / 30V DC 6A

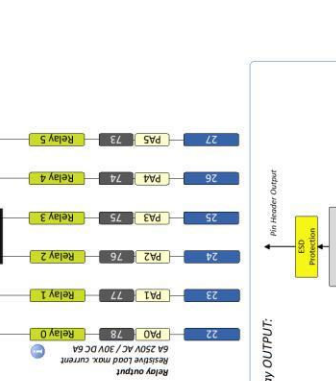
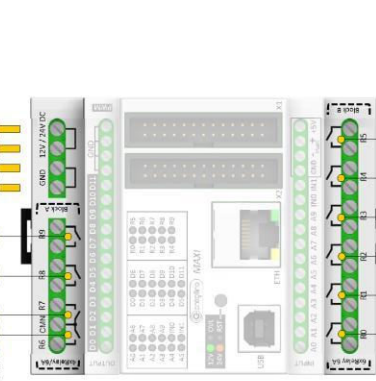
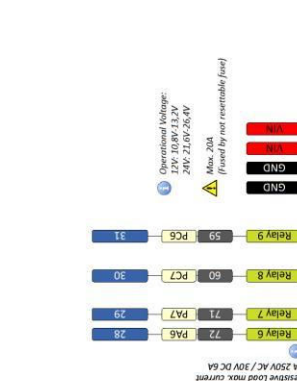
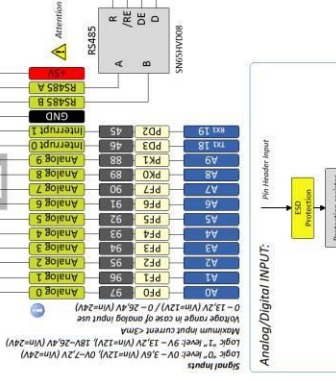
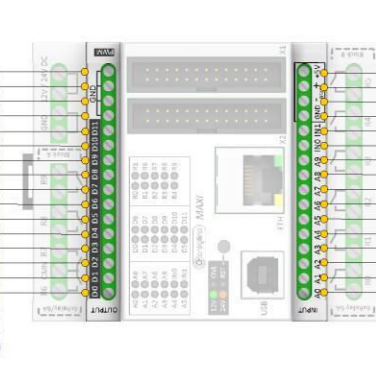
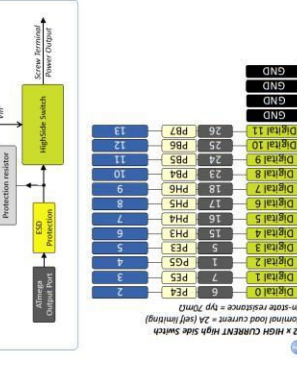
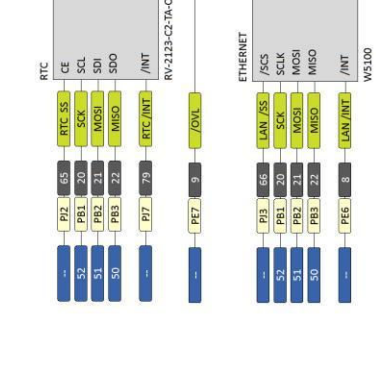
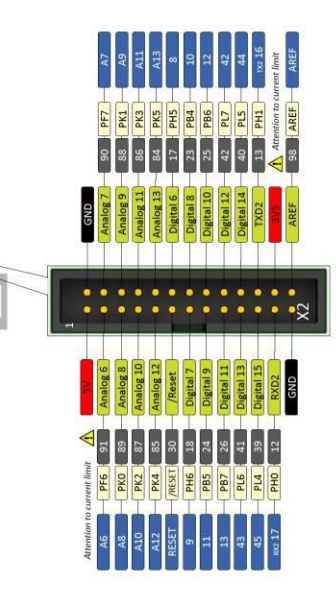
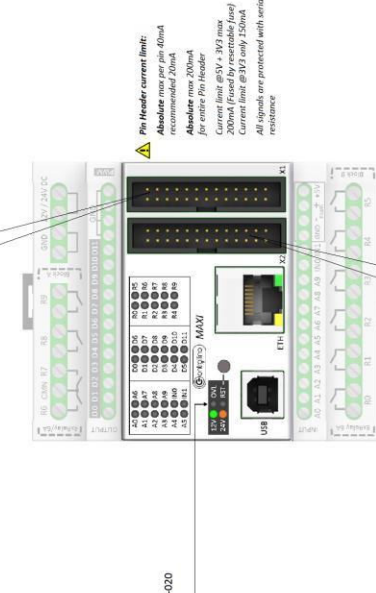
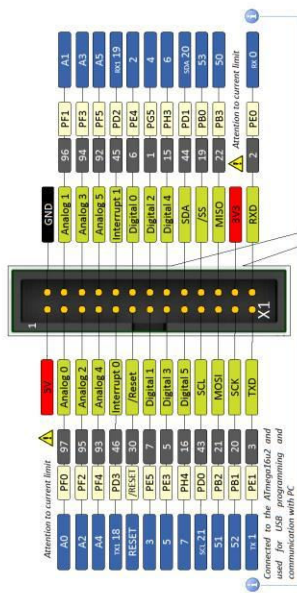
A4 - A5 only Analog inputs

ANEXO A.2

PINOUT TABLE CONTROLLINO MAXI



MAXI PINOUT V1.1

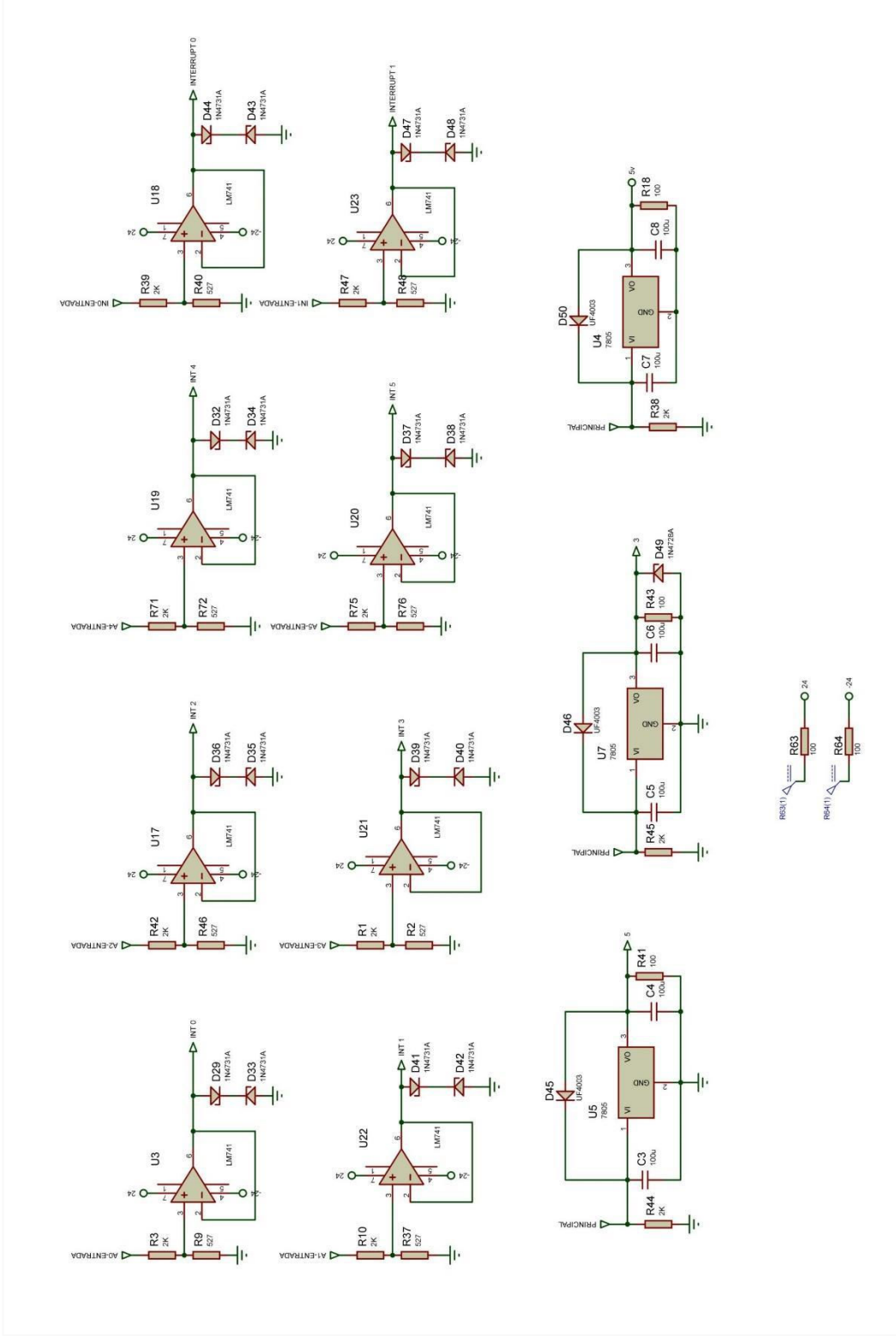


ANEXO B

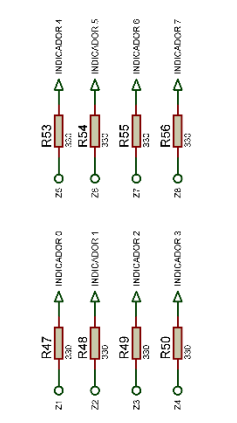
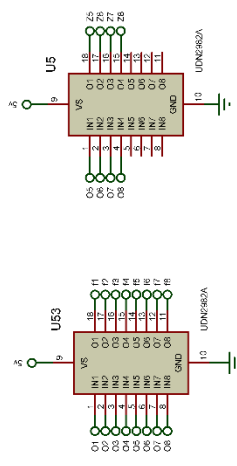
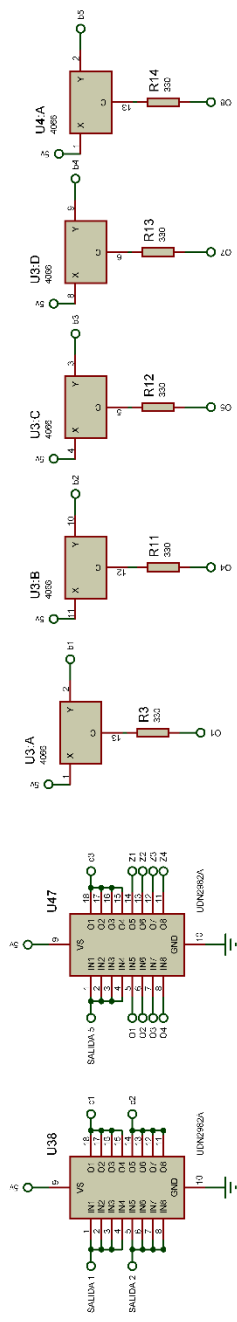
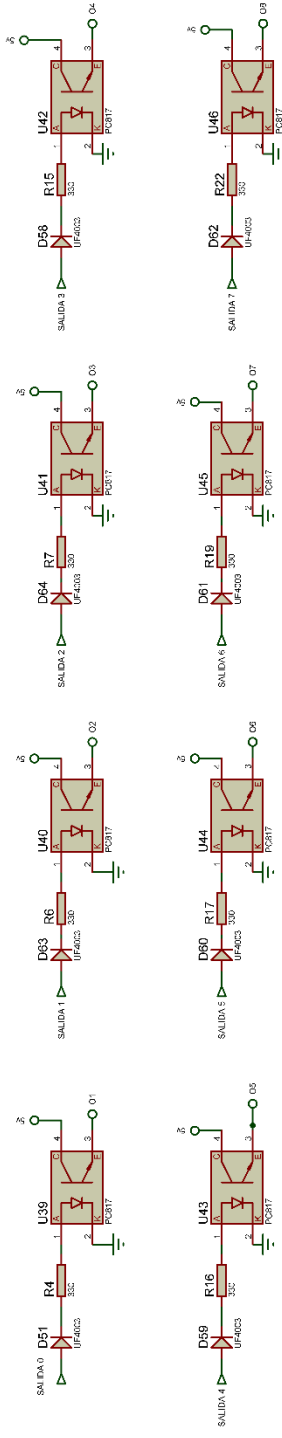
ESQUEMAS ELECTRÓNICOS

ANEXO B.1

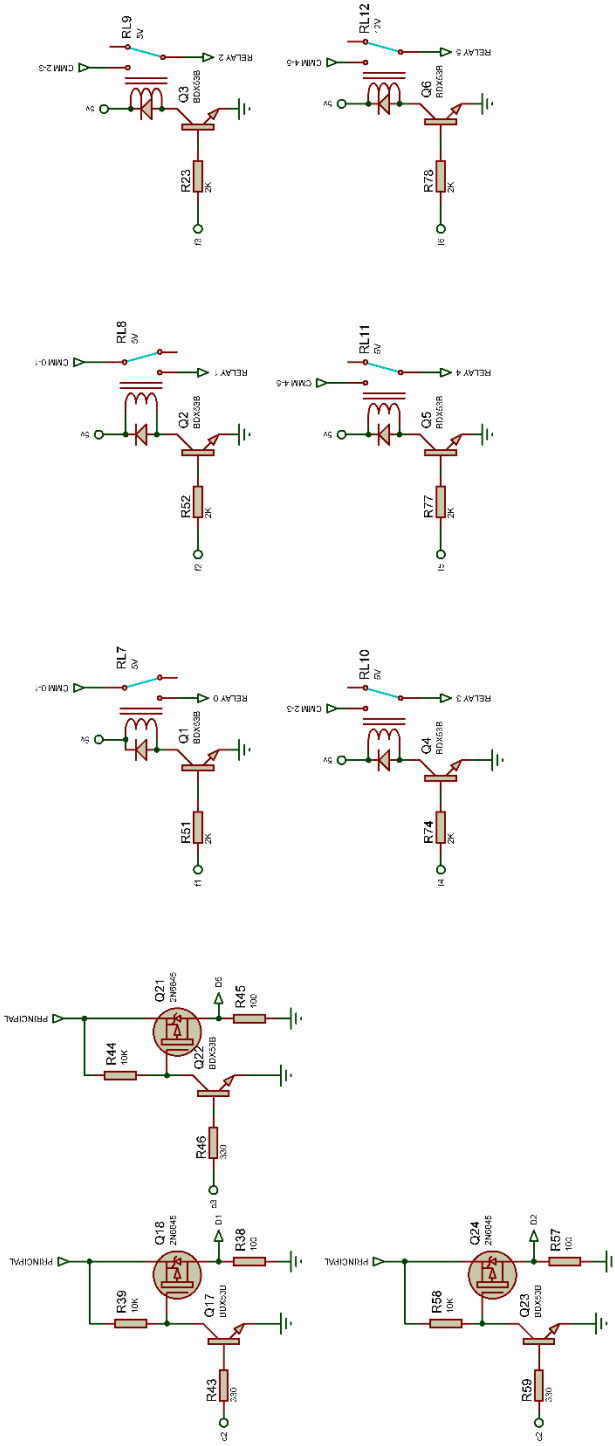
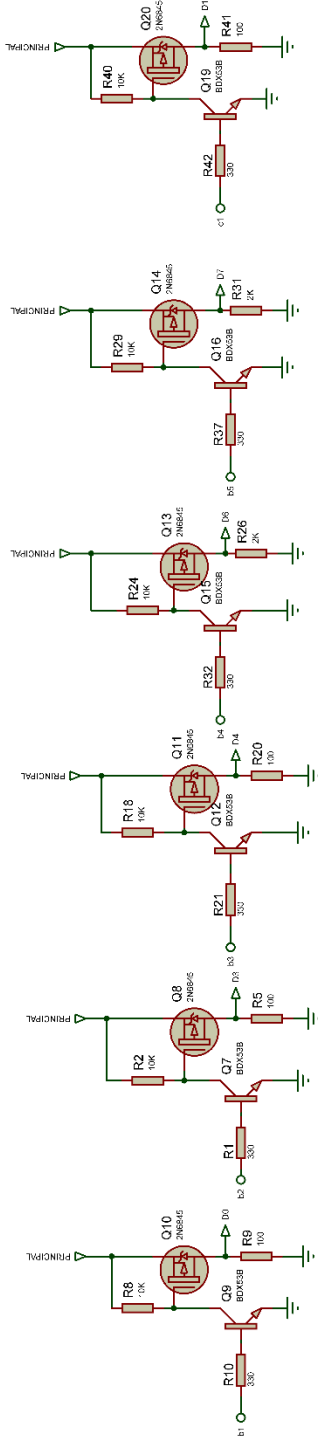
CONTROLLINO MINI 24 VDC




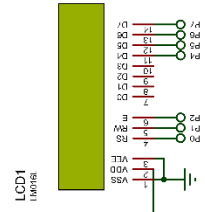
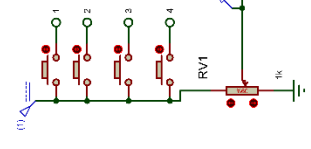
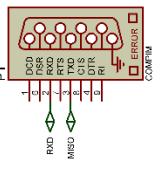
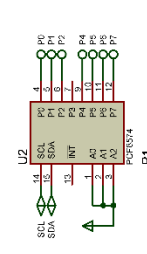
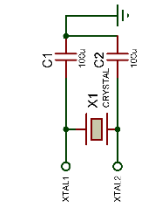
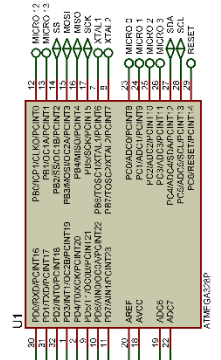
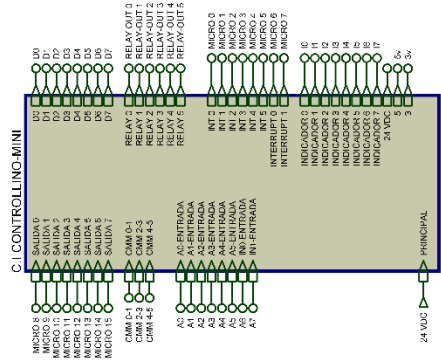
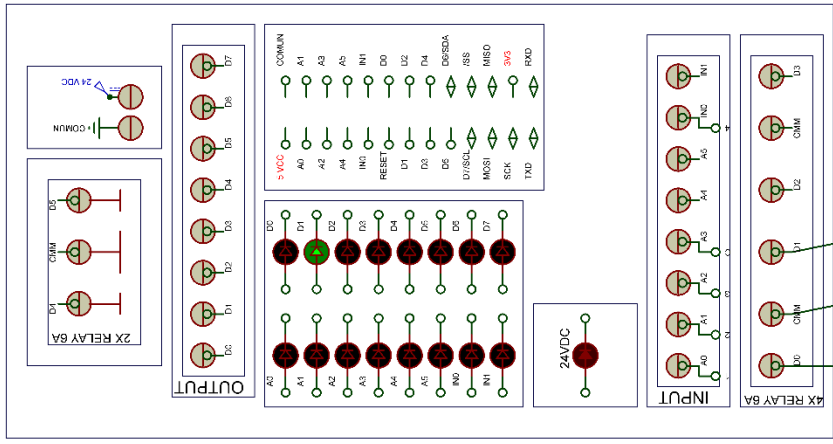
ESCUELA POLITÉCNICA NACIONAL ESCUELA DE FORMACIÓN DE TECNÓLOGOS	ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	
	REVISADO: ING. ALAN CUENCA	
ESQUEMA ELECTRÓNICO DE ENTRADAS CONTROLLINO MINI	PROYECTO INTEGRADOR	
	FECHA: 22/10/2020	LÁMINA: N° B.1.1




ESCUELA POLITÉCNICA NACIONAL	ELABORADO POR:	SALAZAR QUINAUCHO
	ESCUELA DE FORMACIÓN DE TECNÓLOGOS	RONY STEVEN
PROYECTO INTEGRADOR	REVISADO:	ING. ALAN CUENCA
	FECHA:	LÁMINA: Nº B 1.2



	ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	
	REVISADO: ING. ALAN CUENCA	
ESCUELA POLITÉCNICA NACIONAL ESCUELA DE FORMACIÓN DE TECNÓLOGOS		FECHA: 22/10/2020
SALIDAS MOSFET Y RELAY CONTROLLINO MINI		LÁMINA: N° B.1.3
PROYECTO INTEGRADOR		

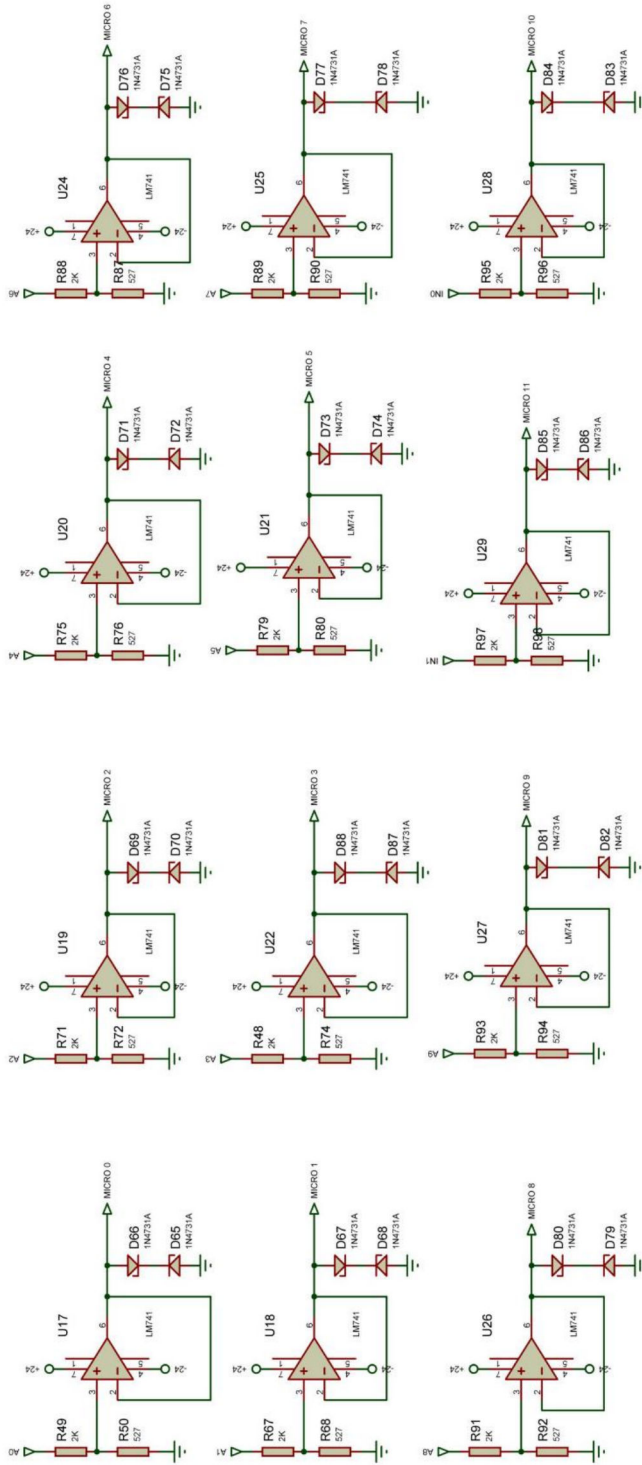


	ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	
	REVISADO: ING. ALAN CUENCA	
ESQUEMA DE CONEXIONES CONTROLINO MINI		FECHA: 22/10/2020
PROYECTO INTEGRADOR		LÁMINA: Nº B.1.4

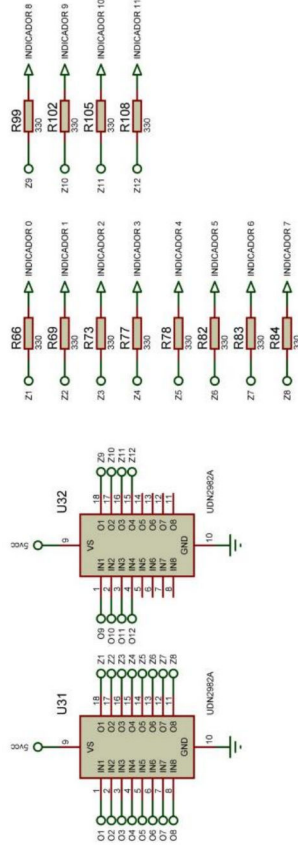
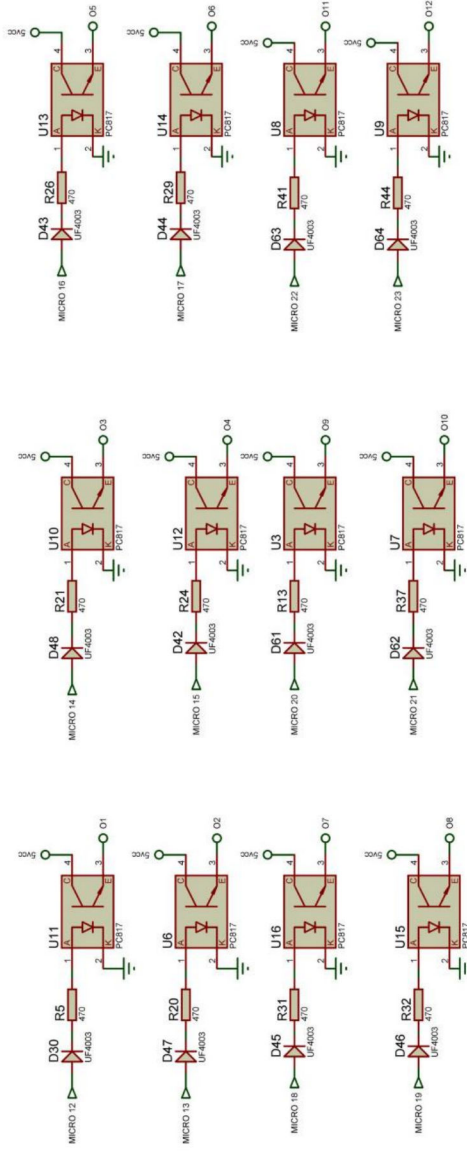
ESCUOLA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
ESQUEMA DE CONEXIONES
CONTROLINO MINI
PROYECTO INTEGRADOR


ANEXO B.2

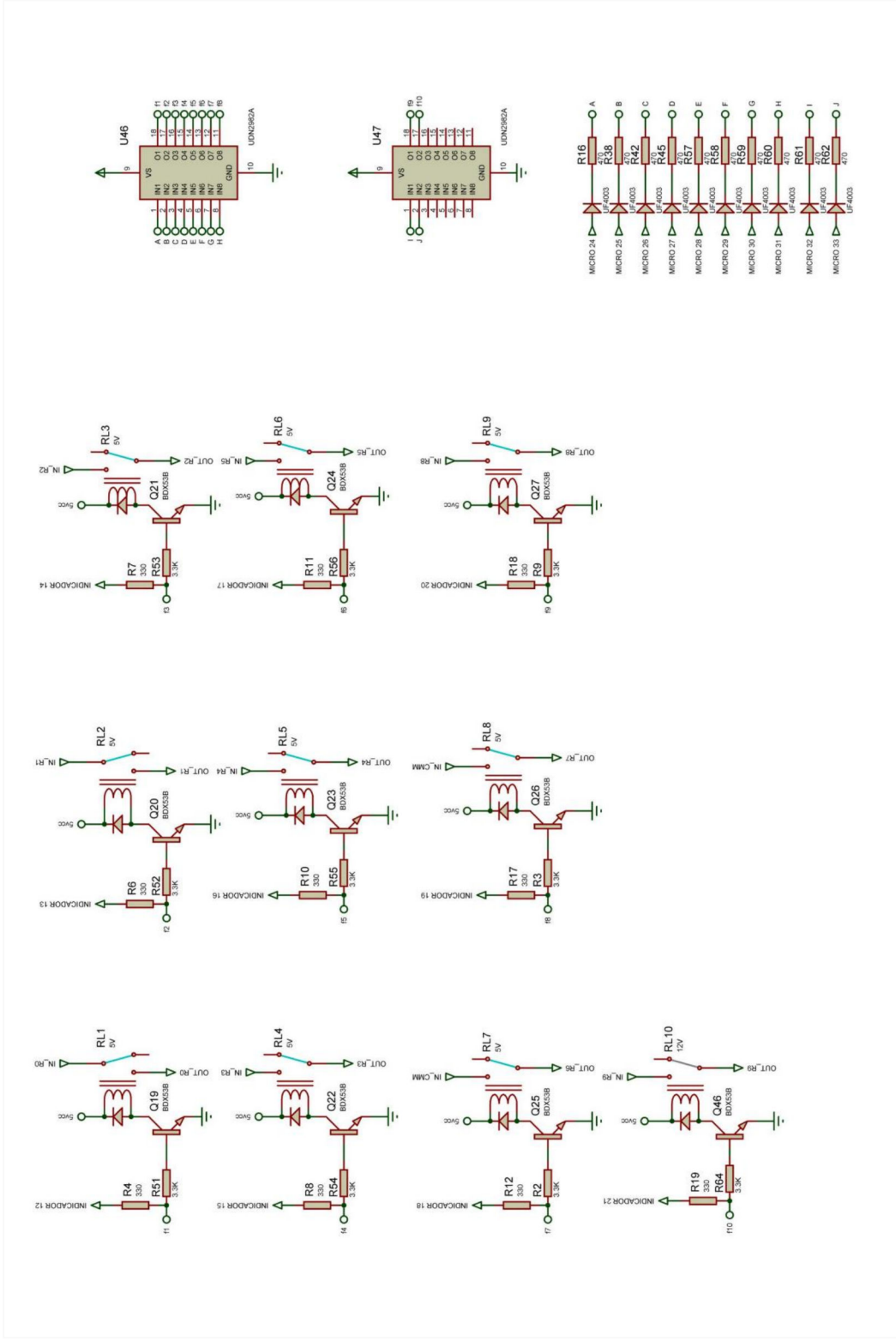
CONTROLLINO MAXI 24 VDC




ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	ESCUELA POLITÉCNICA NACIONAL
	ESCUELA DE FORMACIÓN DE TECNÓLOGOS
REVISADO: ING. ALAN CUENCA	ESQUEMA ELECTRÓNICO DE ENTRADAS CONTROLLINO MAXI
FECHA: 22/10/2020	PROYECTO INTEGRADOR
LÁMINA: N° B.2.1	

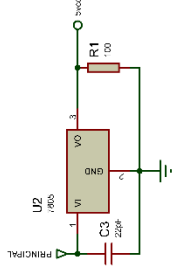
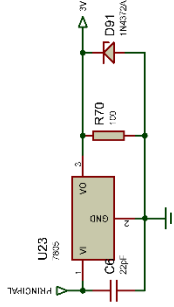
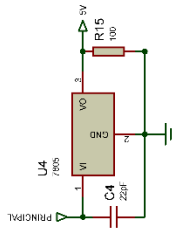
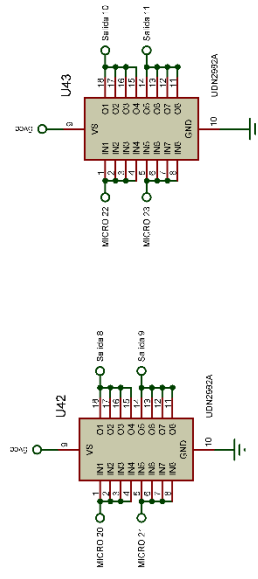
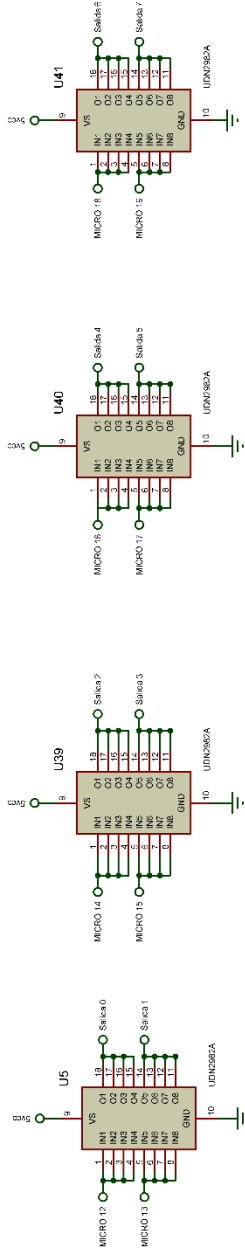


	ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	
	REVISADO: ING. ALAN CUENCA	
ESCUELA POLITÉCNICA NACIONAL ESCUELA DE FORMACIÓN DE TECNÓLOGOS		FECHA: 22/10/2020
ESQUEMA ELECTRÓNICO DE SALIDAS CONTROLLINO MAXI		LÁMINA: Nº B.2.2
PROYECTO INTEGRADOR		

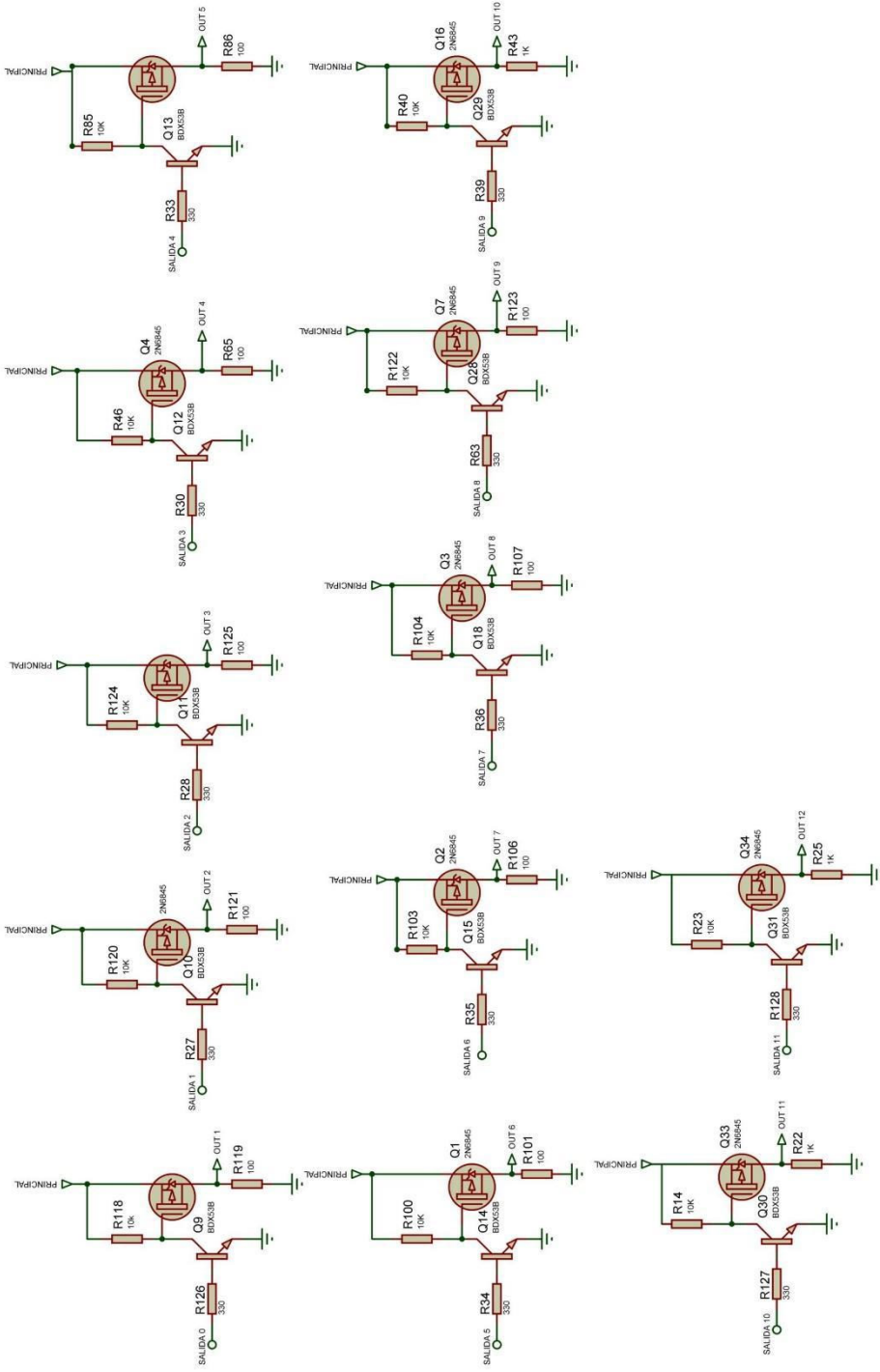


	ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	
	REVISADO: ING. ALAN CUENCA	
ESQUEMA ELECTRÓNICO DE SALIDAS RELAY CONTROLINO MAXI		FECHA: 22/10/2020
PROYECTO INTEGRADOR		LÁMINA: N° B.2.3

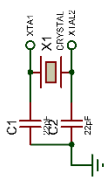
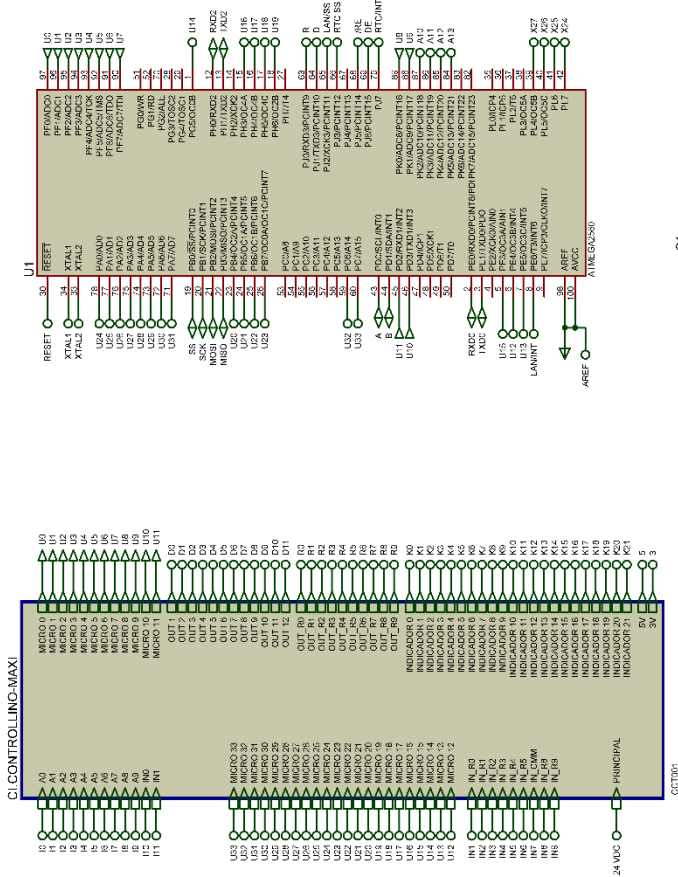
ESCUELA POLITÉCNICA NACIONAL
 ESCUELA DE FORMACIÓN DE TECNÓLOGOS
 PROYECTO INTEGRADOR




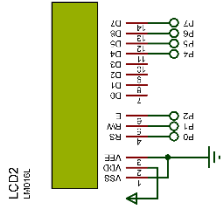
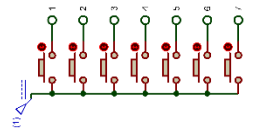
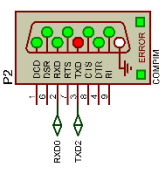
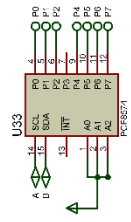
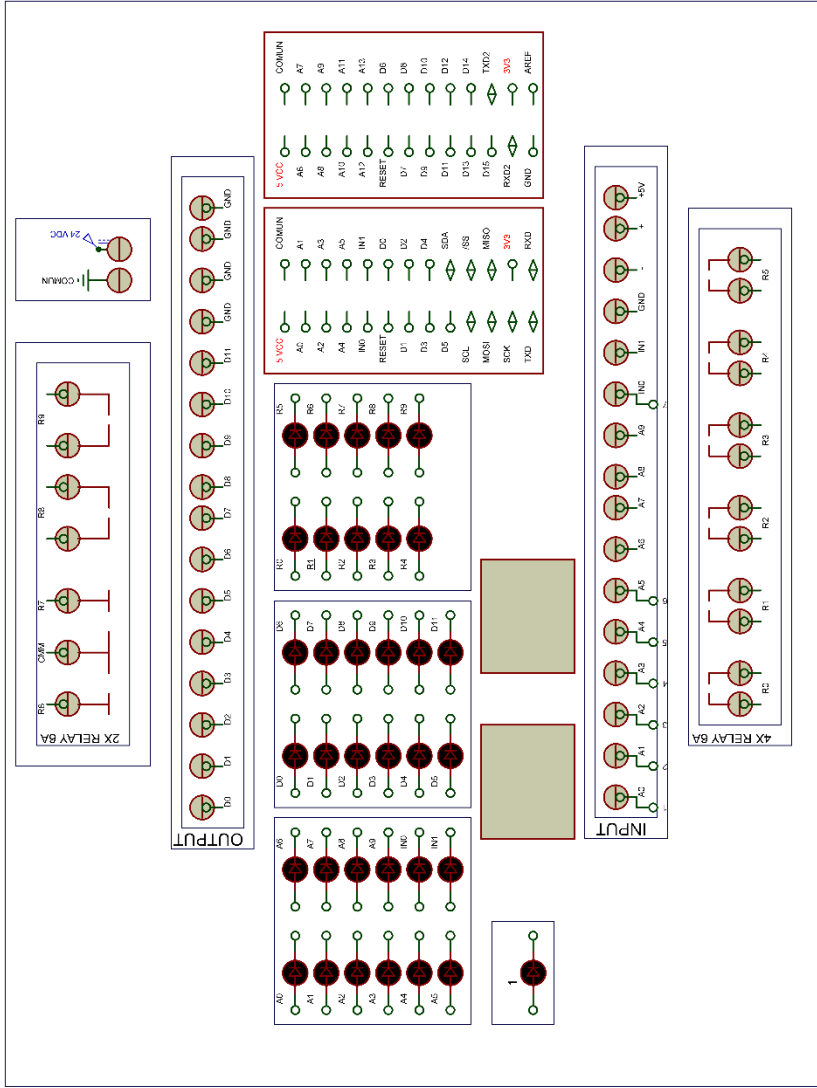
ESCUELA POLITÉCNICA NACIONAL	ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	
	ESCUELA DE FORMACIÓN DE TECNÓLOGOS	
PROYECTO INTEGRADOR	REVISADO: ING. ALAN CUENCA	
	FECHA: 22/10/2020	LÁMINA: N° B.2.4




ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	ESCUELA POLITÉCNICA NACIONAL
	ESCUELA DE FORMACIÓN DE TECNÓLOGOS
REVISADO: ING. ALAN CUENCA	ESQUEMA ELECTRÓNICO DE SALIDAS MOSFET CONTROL LINO MAXI
FECHA: 22/10/2020	PROYECTO INTEGRADOR
LÁMINA: N° B.2.5	



	ESCUOLA POLITÉCNICA NACIONAL	ELABORADO POR:
	ESCUELA DE FORMACIÓN DE TECNÓLOGOS	SALAZAR QUINAUCHO RONY STEVEN
	SUBCIRCUITO CONTROLINO MAXI	REVISADO: ING. ALAN CUENCA
PROYECTO INTEGRADOR	FECHA: 22/10/2020	LÁMINA: Nº B.2.6



	ELABORADO POR: SALAZAR QUINAUCHO RONY STEVEN	
	ESCUELA POLITÉCNICA NACIONAL ESCUELA DE FORMACIÓN DE TECNÓLOGOS	
ESQUEMA DE CONEXIONES CONTROLLINO MAXI		REVISADO: ING. ALAN CUENCA
PROYECTO INTEGRADOR		FECHA: 22/10/2020
		LÁMINA: Nº B.2.7

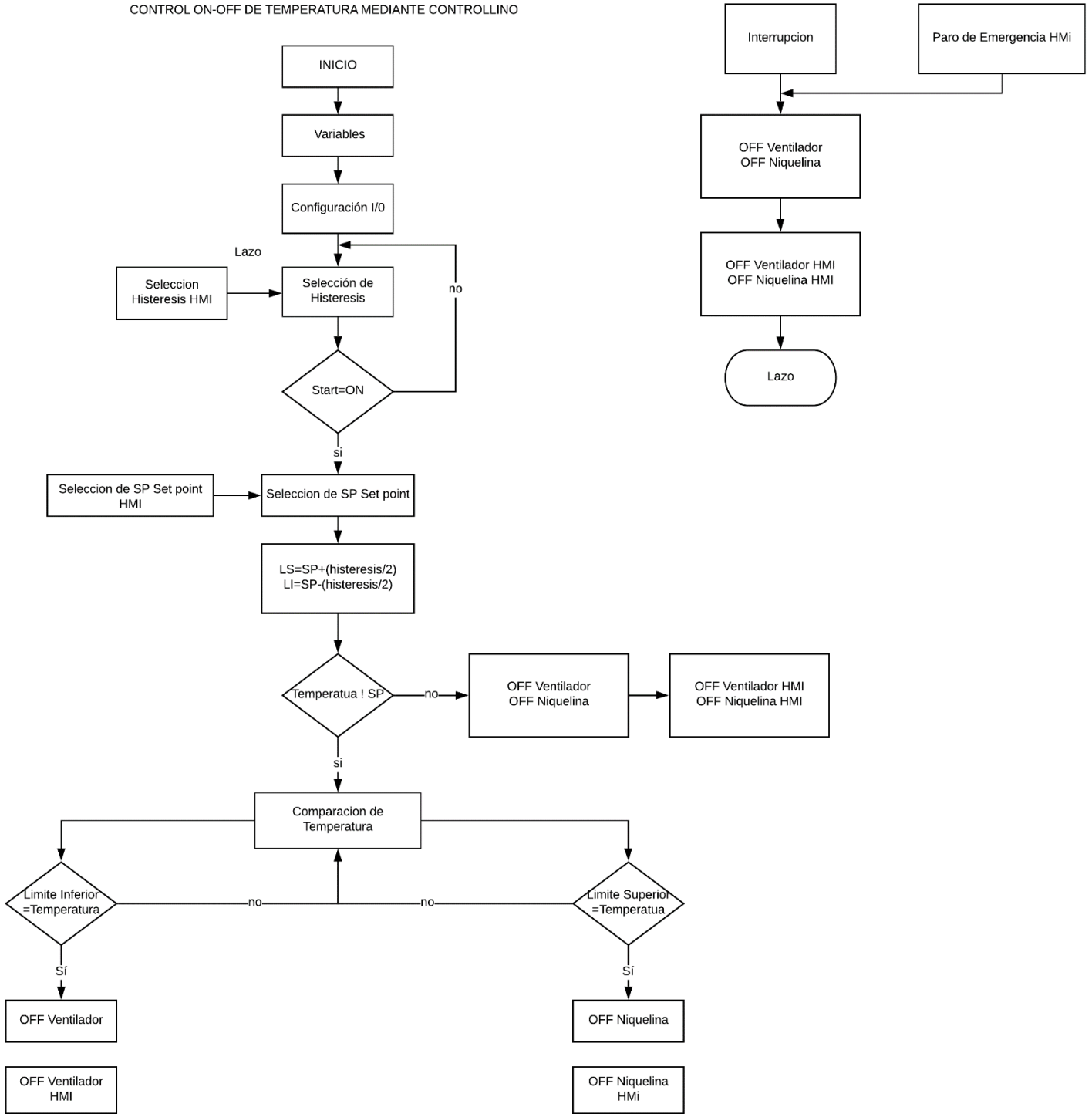
ANEXO C

DIAGRAMAS DE FLUJO

ANEXO C.1

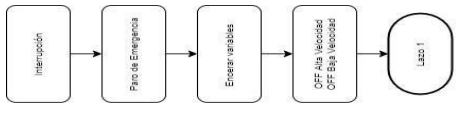
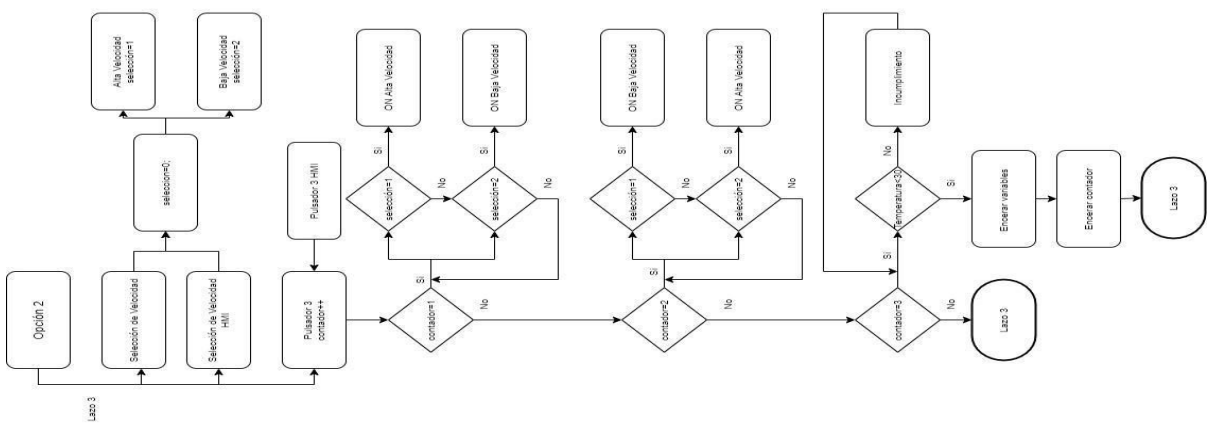
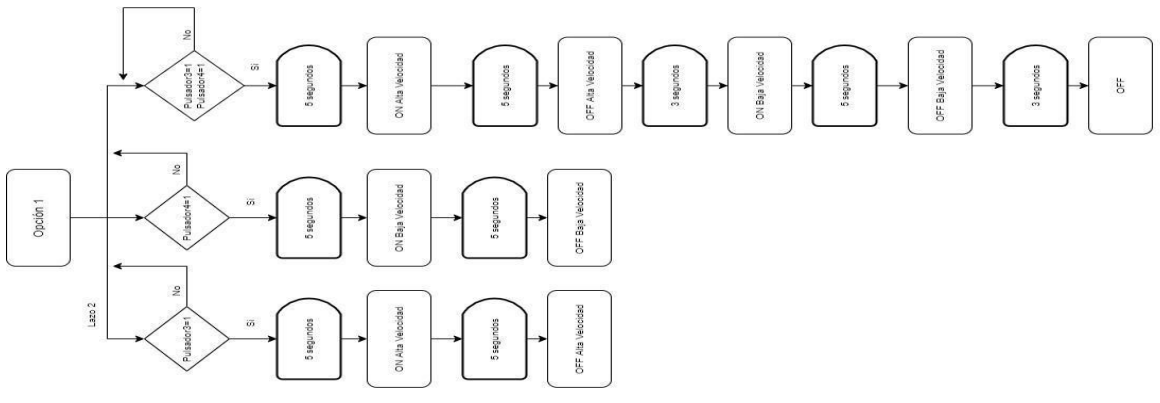
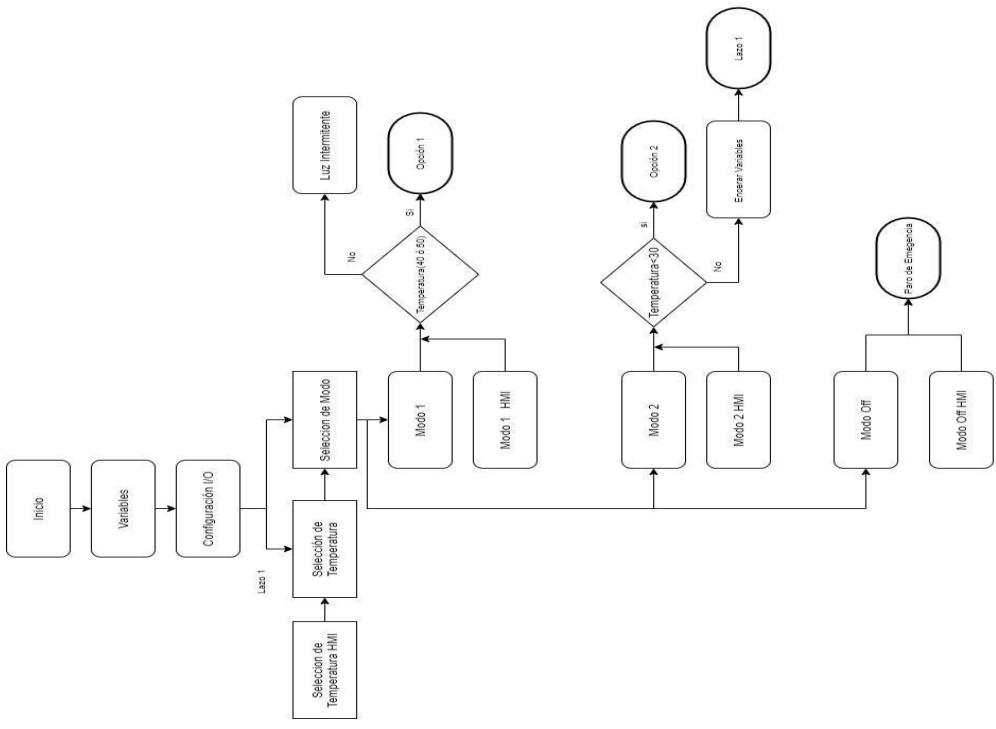
CONTROL ON-OFF DE TEMPERATURA

CONTROL ON-OFF DE TEMPERATURA MEDIANTE CONTROLLINO



ANEXO C.2

CONTROL DE UN MOTOR DE DOS VELOCIDADES



ANEXO D

ALGORITMOS DE CONTROL

ANEXO D.1

CONTROL ON-OFF DE TEMPERATURA

```

// ESCUELA POLITÉCNICA NACIONAL
// ESCUELA DE FORMACIÓN DE TECNÓLOGOS
// SIMULACIÓN DE UN CONTROLLINO-MINI
// SISTEMA DE CONTROL ON-OFF DE TEMPERATURA
// LIBRERIA DE CONTROLLINO
#include <SPI.h>
#include <Controllino.h>
// LIBRERIA DE COMUNICACIÓN I2C CON PANTALLA LCD 16*2
#include <LiquidCrystal_I2C.h>
//LIBRERIA PARA CONEXIÓN SERIAL Y TRANSMICIÓN DE DATOS SERIALES
#include <SoftwareSerial.h>
// DECLARACIÓN DE VARIABLES
#define M_welcome 0
#define M_control 1
#define emergencia 10
SoftwareSerial sms(11,12); //DECLARACIÓN DE PINES PARA COUNICACIÓN SERIAL
//DECLARACIÓN DE PIN DE INTERRUPCIÓN DE CONTROLLINO
const uint8_t paro_emergencia=CONTROLLINO_IN0;
// DECLARACIÓN DE VARIABLES
uint8_t start;
uint8_t incremento;
uint16_t decremento;
uint16_t incre_estado;
uint16_t in_anterior;
uint16_t decre_estado;
uint16_t de_anterior;
uint16_t cas;
uint16_t pas;
uint16_t cas1;
uint16_t cas2;
volatile uint8_t estado;
volatile uint8_t boton_estado;
volatile uint16_t histeresis;
volatile uint16_t temperatura;
uint32_t medicion;
uint8_t particion;

//INICIALIZACIÓN DE LCD 16*2
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
//INICIALIZACIÓN DE BAUDIOS PARA LA COMUNICACIÓN SERIAL
Serial.begin(9600);
sms.begin(9600);
//DECLARACIÓN DE PINES DE ENTRADA EN CONTROLLINO
pinMode(CONTROLLINO_A0,INPUT);
pinMode(CONTROLLINO_A1,INPUT);
pinMode(CONTROLLINO_A2,INPUT);

```

```

pinMode(CONTROLLINO_A3,INPUT);
pinMode(paro_emergencia,INPUT);
DDRD = 0b00110000 ; //DECLARACIÓN DE PUERTOS DE SALIDA
//DECLARACIÓN DE INTERRUPTACION
attachInterrupt(digitalPinToInterrupt(paro_emergencia), estado_emergencia, RISING);
//LLAMADA AL BUCLE DEL LCD
bienvenida();
delay(100);
titulo();
delay(1000);
//ESTABLECIENDO VALORES A LAS VARIABLES
estado=M_welcome;
temperatura=20;
}
void loop() {
//COLOCANDO LA VARIABLES A LOS PINES DE ENTRADA DEL CONTROLLINO
start=digitalRead(CONTROLLINO_A0);
decremento=digitalRead(CONTROLLINO_A1);
incremento=digitalRead(CONTROLLINO_A2);
particion=Serial.read();
//CONDICIÓN DE PULSADORES DE FLANCO DE BAJADA
//ALMACENAR DATOS EN VARIBALES
if(start==HIGH ){
    boton_estado=true;
}
if(incremento==HIGH ){
    incre_estado=true;
}
if(decremento==HIGH ){
    decre_estado=true;
}
//CONDICIÓN DE AL SOLTAR EL BOTON REALIZA LA EJECUCION DE CADA BOTÓN
switch (estado){
    case M_welcome:
        seleccion();
//CONDICIÓN PARA VERIFICAR SI SE PRESIONÓ BOTON DE START
//COLOCACIÓN DE ESTADO A 1
    if(boton_estado==HIGH && start==LOW){
        estado=M_control;
    }
    break;
//CONDICIÓN DE PRESIONAR EL BOTÓN DE START
//SALTO AL BUCLE PROCESO DONDE SE EJCUTA EL PROGRAMA
    case M_control:
        proceso();
        break;
//PARO DE EMERGENCIA INGRESADA POR HMI
}
}

```



```

    case emergencia:
        estado_emergencia();
        break;
}
}
// FUNCIÓN DE PARO DE EMERGENCIA CON LA INTERRUPCIÓN Y ENCERRAR LAS SALIDAS Y VARIABLES GLOBALES
void estado_emergencia(){
    estado=M_welcome;
//ENCERRAR VARIABLE DE ESTADOS
    boton_estado=false;
    decre_estado=false;
    incre_estado=false;
//ENCERRAR SALIDAS DE CONTROLLINO MINI
    PORTD=0X00;
// APAGADO DE INDICADORES DE VENTILADOR EN HMI
    sms.print("p1.pic=");
    sms.print("4");
    sms.write(0xff);
    sms.write(0xff);
    sms.write(0xff);
//APAGADO DE INDICADORES DE NIQUELINA EN HMI
    sms.print("p2.pic=");
    sms.print("4");
    sms.write(0xff);
    sms.write(0xff);
    sms.write(0xff);
    return loop; //REGRESO A LA EJECUCIÓN DEL PROGRAMA
}
//FUNCIÓN SELECCIÓN EN LA CUAL REALIZA EL INCREMENTO Y DECREMENTO DE LA HISTÉRESIS
void seleccion(){
//INCREMENTO DE HISTÉRESIS DESDE PANEL FRONTLA DE CONTROLLINO MINI
if(incre_estado==HIGH && incremento==LOW ){
    if(incremento != in_anterior ){
        if(incremento==LOW ){
            pas++;
        }
    }
}
in_anterior = incremento;
//DECREMENTO DE HISTÉRESIS DESDE PANEL FRONTAL DE CONTROLLINO MINI
if(decre_estado==HIGH && decremento==LOW ){
    if(decremento != de_anterior ){
        if(decremento==LOW){
            if(pas>0 && pas<=9){
                pas--;}
        }
    }
}
}
}

```

```

}

de_anterior = decremento;
pantalla()); // REALIZA SALTO A LA FUNCION PARA LA IMPRESION DE VALORES EN LCD Y HMI
}
void proceso(){
//LECTURA ANALOGICA Y CONVERSION ADC MEDIANTE ECUACIÓN
medicion=(analogRead(A3)* 0.058594+20);
//INCREMENTO DE SET POINT DESDE PANEL FRONTAL CONTROLLINO MINI
if(incre_estado==HIGH && incremento==LOW ){
    if(incremento != in_anterior ){
        if(incremento==LOW){
            cas2++;
        }
    }
}
in_anterior = incremento;
//DECERMENTO DE SET POINT DESDE PANEL FRONTAL DE CONTROLLINO MINI
if(decre_estado==HIGH && decremento==LOW ){
    if(decremento != de_anterior ){
        if(decremento==LOW ){
            cas2--;
        }
    }
}
de_anterior = decremento;
//CONDICIÓN DE TEMPERATURA INGRESADA POR EL PANEL FRONTAL DE CONTROLLINO MINI
//TEMPERATURA MAYOR A 80°C
if(cas2>80){
    cas2=80;
}
//TEMPERATURA MENOR A 20°C
if(cas2<20){
    cas2=20;
}
//CONDICIÓN DE ACTIVACION DE SALIDA Y ENVIO DE DATOS A HMI
//PROCESS VALUE MAYOR A SET POINT
if((medicion)>temperatura){
//ENCENDIDO DE SALIDA D0 DE CONTROLLINO MINI
PORTD=0b0100000;
//ENVIÓ DE ACTIVACIÓN DE INDICADOR DE VENTILADOR
sms.print("p1.pic=");
sms.print("2");
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
//ENVIÓ DE DESACTIVACIÓN DE INDICADOR DE NIQUELINA

```

```

sms.print("p2.pic=");
sms.print("4");
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
}
//CONDICIÓN DE
if((medicion)<temperatura){
PORTD=0b00010000;
//ENVIÓ DE VARIABLE PROCESS VALUE DEL SISTEMA A INTERFAZ HUMANO MÁQUINA HMI
sms.print("n0.val=");
sms.print(medicion);
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
//ENVIÓ DE ACTIVACIÓN DE INDICADOR DE NIQUELINA
sms.print("p2.pic=");
sms.print("2");
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
//ENVIÓ DE DESACTIVACIÓN DE INDICADOR VENTILADOR
sms.print("p1.pic=");
sms.print("4");
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
}
//CONDICIÓN DE APAGADO DE SALIDAS CUANDO CUMPLE LA PROCESS VALUE Y SET POINT IGUALES
if(medicion==temperatura ){
//APAGADO DE SALIDAS DE CONTROLLINO MINI
PORTD=0b00000000;
//ENVIÓ DE APAGADO DE INDICADORES EN INTERFAZ HUMANO MÁQUINA HMI
//APAGADO DE NIQUELINA EN INTERFAZ HUMANO MÁQUINA HMI
sms.print("p1.pic=");
sms.print("4");
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
//APAGADO DE VENTILADOR EN INTERFAZ HUMANO MÁQUINA HMI
sms.print("p2.pic=");
sms.print("4");
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
}
pantalla2();

```

```

}
//HABILITACIÓN DE LECTURA CUANDO EXISTA UN DATO ENVIADO POR LA INTERFAZ HUMANO MÁQUINA HMI
void serialEvent() {
while(Serial.available()){
particion=Serial.read();
switch (particion){
// CASO PARA INCREMENTO DE HISTÉRESIS MEDIANTE HMI
case ('1'):
cas=cas+1;
pantalla();
//CONDICIÓN DE HISTERESIS INGRESADA POR HMI NO MAYOR A 9
if(cas>9){
cas=9;
}
break;
//CASO DE DECREMENTO DE HISTÉRESIS MEDIANTE HMI
case ('2'):
//CONDICIÓN DE FUNCINAMIENTO DE DECREMENTO DE HISTÉRESIS NO MENOR A 0
if(cas>0 && cas<=9){
cas=cas-1;
}
//SALTO AL BUCLE PANTALLA PARA IMPRESION LCD DE DATOS Y ENVIO HMI
pantalla();
break;
//CASO DE INCREMENTO DE SET POINT MEDIANTE HMI
case('3'):
cas1=cas1+1;
//CONDICIÓN DE SET POINT NO MAYOR A 80°C
if(cas1>80){
cas1=80;
}
break;
//CASO DE DECREMENTO DE SET POINT MEDIANTE HMI
case('4'):
cas1=cas1-1;
//CONDICIÓN DE SET POINT NO MENOR A 20°C
if(cas1<0){
cas1=0;
}
break;
//CASO DE PREISONAR START AL SISTEMA
case ('s'):
estado=1;
proceso();
break;
//ESTADO DE EMERGENCIA
case('p'):

```

```

estado=10;
estado_emergencia();
break;
}
}
}
//IMPRESIÓN EN LCD 16*2 DE CARACTERES AL INICIALIZAR CONTROLLINO MINI
void bienvenida(){
  lcd.init();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("SIMULACION");
  lcd.setCursor(2,1);
  lcd.print("CONTROLLINO");
  delay(1000);
  lcd.clear();
}
//IMPRESIÓN DEL SISTEMA DE CONTROL EN LCD 16*2
void titulo(){
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print(" ");
  lcd.print("CONTROL ON-OFF ");
  lcd.setCursor(0,1);
  lcd.print(" ");
  lcd.print(" TEMPERATURA");
  delay(1);
}
void pantalla(){
//ADICIÓN DE HISTERESIS INGRESO POR EL CONTROLADOR Y LA INTERFAZ HUMANO MÁQUINA HMI
(histeresis=cas+pas);
//CONDICIÓN DE HISTERESIS NO MAYOR A 9 °c
if(histeresis>9){
  histeresis=9;
}
//CONDICIÓN DE HISTÉRESIS NO MENOR A 0°C
if(histeresis<=0){
  histeresis=histeresis*0;
}
}
//IMPRESIÓN DE VARIABLE DEL SISTEMA EN LCD 16*2
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print(" HISTERESIS ");
  lcd.setCursor(0,1);

```

```

lcd.print("  ");
lcd.print("H:");
lcd.print(histeresis);
lcd.print("  ");
delay(1);
//ENVIÓ DE VARIBALE DE SISTEMA A INTERFAZ HUMANO MÁQUINA HMI
sms.print("n4.val=");
sms.print(histeresis);
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
return;
}
void pantalla2(){
//ADICCIÓN DE VALORES INGRESADOS POR EL CONTROLADOR Y LA INTERFAZ HUMANO MÁQUINA HMI
temperatura=cas1+cas2;
// CONDICIÓN DE TEMPEMPERATURA MAYOR A 80°C
if(temperatura>80){
temperatura=80;
}
//CONDICIÓN DE TEMPERATURA MENOR A 20°C
if(temperatura<20){
temperatura=20;
}
//IMPRESIÓN DE VARIABLES DEL SISTEMA EN LCD 16*2
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("PV:");
lcd.print(medicion);
lcd.print(" ");
lcd.print("SP:");
lcd.print(temperatura);
lcd.print("  ");
lcd.setCursor(0,1);
lcd.print("LS:");
lcd.print(temperatura+(histeresis/2));
lcd.print(" ");
lcd.print("LI:");
lcd.print(temperatura-(histeresis/2));
lcd.print(" ");
lcd.print("H:");
lcd.print(histeresis);
//ENVIÓ DE DATOS A LA INTERFAZ HUMANO MÁQUINA HMI
//ENVIÓ DE VARIBALE PROCESS VALUE A INTERFAZ HUMANO MÁQUINA
sms.print("n0.val=");
sms.print(medicion);
sms.write(0xff);

```

```
sms.write(0xff);
sms.write(0xff);
delay(1);
//ENVIÓ DE VARIABLE TEMPERATURA A INTERFAZ HUMANO MÁQUINA HMI
sms.print("n1.val=");
sms.print(temperatura);
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
delay(1);
//ENVIÓ DE VARIABLE LIMITE SUPERIOR A INTERFAZ HUMANO MÁQUINA HMI
sms.print("n2.val=");
sms.print(temperatura+(histeresis/2));
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
delay(1);
//ENVIÓ DE VARIABLE LIMITE INFERIOR A INTERFAZ HUMANO MÁQUINA HMI
sms.print("n3.val=");
sms.print(temperatura-(histeresis/2));
sms.write(0xff);
sms.write(0xff);
sms.write(0xff);
return;
}
```

ANEXO D.2

CONTROL DE UN MOTOR DE DOS
VELOCIDADES


```

//ESCUELA POLITÉCNICA NACIONAL
//ESCUELA DE FORMACIÓN DE TECNÓLOGOS
//SIMULACIÓN DE CONTROL DE UN MOTOR DE DOS VELOCIDADES
//LIBRERIA DE COMUNICACIÓN I2C PARA LCD
#include <LiquidCrystal_I2C.h>
//LIBRERIA DE CONTROLLINO
#include <SPI.h>
#include <Controllino.h>
//INICIALIZACIÓN DE LCD 16*2
LiquidCrystal_I2C lcd(0x27,16,2);
//DECLARACIÓN DE VARIABLES
#define Seleccion_modo1 0
#define Seleccion_modo2 1
#define Seleccion_off 2
const uint16_t pulsador1=CONTROLLINO_A0;
const uint16_t pulsador2=CONTROLLINO_A1;
volatile uint16_t temperatura=20;
const uint8_t paro_emergencia=CONTROLLINO_IN0;
const uint16_t off=CONTROLLINO_IN1;
volatile uint16_t ingreso ;
volatile uint8_t estado=0;
volatile uint8_t estado1;
volatile uint16_t estado2;
volatile uint8_t estado3;
volatile uint8_t estado4;
volatile uint8_t estado5;
volatile uint8_t estado6;
volatile uint8_t estado7;
volatile uint8_t estado8;
volatile uint8_t estado9;
volatile uint8_t estado10;
volatile uint16_t tiempo;
uint8_t comando;
uint16_t pulso;
uint16_t pulso2;
uint16_t pulso3=0;
uint16_t pulso31=0;
uint16_t pulso32=0;
uint16_t pulso33=0;
uint16_t in_anterior;
uint16_t incre_estado;
uint16_t in;
uint16_t in_pasada;
uint16_t sel=0;
uint16_t sel1=0;
uint16_t plus=0;
uint8_t incremento;

```

```

uint8_t      incre_estado2;
uint8_t      decremento;
uint8_t      decre_estado;
uint8_t      in_anterior2;
uint8_t      in_anterior3;
uint8_t      tem1;
uint8_t      tem2;

void setup() {
//INICIALIZACIÓN DE VELOCIDAD DE COMUNICACIÓN SERIAL
Serial.begin(57600);
Serial2.begin(57600);
//DECLARACION DE PÓRTICOS DE SALIDA Y ENTRADA
DDRA|=0XFF;
DDRF|=0X00;
//ENCERAR ENTRADAS Y SALIDAS DEL CONTROLADOR
PORTA=0X00;
PORTF|=0X00;
//DECLARACION DE PIN DE INTERRUPCIÓN PARA PARO DE EMERGENCIA POR CAMBIÓ DE ESTADO
attachInterrupt(digitalPinToInterrupt(CONTROLLINO_IN0),caso,RISING);
//INICIALIZACIÓN DE VARIABLES CON ESTADO DESIGNADOS
estado=4;
temperatura=20;
//SALTO DE FUNCIÓN DE IMPRESION EN LCD 16*2
bienvenida();
//TIEMPO DE RETARDO PARA VISUALIZACIÓN EN LCD 16*2
delay(1000);
//SALTO DE FUNCIÓN DE IMPRESION EN LCD16*2
titulo();
//TIEMPO DE RETARDO PARA VISUALIZACIÓN EN LCD 16*2
delay(1000);
}
void loop() {
//ENVIÓ DE VARIABLE DEL SISTEMA A INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("n0.val=");
Serial2.print(temperatura);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
//SALTO A FUNCIÓN PARA INDICADORES EN INTERFAZ HUMANO MÁQUINA HMI
indicadores();
//ASIGNACIÓN DE VARIABLES A PINES DE ENTRADA
incremento=(digitalRead(pulsador1));
decremento=(digitalRead(pulsador2));
//CONDICIÓN DE PULSO A ENTRADAS DIGITALES PARA
//REALIZAR BOTONES POR FLANCO DE BAJADA
if(incremento==1){
    incre_estado2=true;

```

```

}
if(decremento==1){
  decre_estado=true;
}
//CONDICIÓN PARA REALIZAR EL INCREMENTO DE TEMPERATURA POR FLANCO DE BAJADA
if(incre_estado2==HIGH && incremento==LOW ){
  if(incremento != in_anterior2 ){
    if(incremento==LOW ){
      tem1++;
    }
  }
}
in_anterior2 = incremento;
//CONDICIÓN PARA REALIZAR EL DECREMENTO DE TEMPERATURA POR FLANCO DE BAJADA
if(decre_estado==HIGH && decremento==LOW ){
  if(incremento != in_anterior3 ){
    if(decremento==LOW ){
      tem1--;
    }
  }
}
in_anterior3 = decremento;
//ADICCIÓN DE TEMPERATURA INGRESADA POR PANEL FRONTAL DE CONTROLLINO MAXI Y HMI
temperatura=tem1+tem2+20;
//CONDICIÓN DE FUNCIONAMIENTO EN EL RANGO DE TEMPERATURA ENTRE 40 Y 50
if(temperatura>=40 or temperatura<=50){
  //CONDICIÓN DE PULSACIÓN DE BOTÓN P3 INGRESADO POR PANEL FRONTAL DE CONTROLLINO MAXI Y
  // LA INTERFAZ HUMANO MÁQUINA HMI
  if(((digitalRead(CONTROLLINO_A4))==HIGH && (digitalRead(CONTROLLINO_A5))==LOW) or estado5==4){
    //CONDICIÓN SI SE PRESIONO BOTON DE MODO 1
    //SEA POR PANEL FRONTAL CONTROLLINO MAXI O INTERFAZ HUMANO MÁQUINA HMI
    if(estado==Seleccion_mod01){
      //ASIGNACION DE VALORES A VARIABLES GLOBALES
      estado1=5;
      estado9=4;
    }

    if(estado9==2){
      estado10=3;
    }
  }
}
// CONDICIÓN DE PULSACIÓN DE BOTÓN P4 INGRESADO POR PANEL FRONTAL DE CONTROLLINO MAXI Y
// LA INTERFAZ HUMANO MÁQUINA HMI
if(((digitalRead(CONTROLLINO_A5))==true && (digitalRead(CONTROLLINO_A4))==false) or estado5==2){
  //CONDICIÓN SI SE PRESIONO BOTON DE MODO 1
  //SEA POR PANEL FRONTAL CONTROLLINO MAXI O INTERFAZ HUMANO MÁQUINA HMI
  if(estado==Seleccion_mod01){

```

```

//ASIGNACION DE VALORES A VARIABLES GLOBALES
estado1=6;
if(estado9==4){
    estado10=estado9+5;
}
}
estado9=2;
}
if(estado10==9){
    if(estado==Seleccion_mod01){
        estado1=7;
    }
}
}

//ASIGNACION DE CASOS MEDIANTE LA VARIABLE ESTADO
switch (estado){
// MODO 1
case Seleccion_mod01:
    if(temperatura<=39 or temperatura>=51 ){
        estado1=20;
    }
    visualizacion();
    proceso();
    break;
//MODO 2
case Seleccion_mod02:
    visualizacion1();
    proceso2();
    break;
//MODO OFF
case Seleccion_off:
    limpieza();
    break;
case 4:
//CASO PARA IMPRESION SEGUN ELECCION DE MODO EN LCD 16*2
    seleccion();
//CONDICIÓN AL PRESIONAR MODO 1
    if((digitalRead(CONTROLLINO_A2))==HIGH ){
//ASIGNAR A VARIABLE GLOBAL ESTADO EL VALOR DE 0
        estado=Seleccion_mod01;
//FUNCIÓN DE LIMPIEZA DE LCD 16*2
        limpieza();
//FUNCIÓN DE IMPRESION EN LCD 16*2 LA ELECCION DEL OPERADOR
        modone();

}
//CONDICIÓN AL PRESIONAR MODO 2

```

```

if((digitalRead(CONTROLLINO_A3))==true){
//ASIGNAR A VARIABLE GLOBAL ESTADO EL VALOR DE 1
    estado=Seleccion_modos2;
//FUNCIÓN DE LIMPIEZA DE LCD 16*2
    limpieza();
//FUNCIÓN DE IMPRESION EN LCD 16*2 LA ELECCIÓN DEL OPERADOR
    modotwo();
}
    break;
}
}
//FUNCIÓN PARA MODO 1
void proceso(){
    switch (estado1){
        case 5:
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
            limpieza();
            alta();
            delay(5000);
            alta1();
//ENCENDIDO DE SALIDA TIPO RELAY MEDIANTE PINES DEL PÓRTICO A
            PORTA|=(1<<0);
            PORTA|=(1<<2);
            delay(5000);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
            limpieza();
            alta2();
//APAGADO DE SALIDA TIPO RELAY MEDIANTE PIN DEL PÓRTICO A
            PORTA&=~(1<<0);
            PORTA&=~(1<<2);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
            limpieza();
            alta();
            apagado();
//ENCERRAR VARIABLES GLOBALES DEL SISTEMA DE CONTROL
            estado1=0;
            estado5=0;
            break;
        case 6:
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
            limpieza();
            baja();
            delay(5000);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
            baja1();
//ENCENDIDO DE SALIDA TIPO RELAY MEDIANTE PINES DEL PÓRTICO A
            PORTA|=(1<<1);

```

```

        delay(5000);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
        limpieza();
        baja2();
        apagado1();
//APAGADO DE SALIDA TIPO RELAY MEDIANTE PIN DEL PÓRTICO A
        PORTA&=~(1<<1);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
        apagado1();
        limpieza();
//ENCERRAR VARIABLES GLOBALES DEL SISTEMA DE CONTROL
        estado1=0;
        estado5=0;
        break;
        case 7:
        delay(5000);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
        limpieza();
        alta1();
//ENCENDIDO DE SALIDA TIPO RELAY MEDIANTE PINES DEL PÓRTICO A
        PORTA|= (1<<0);
        PORTA|= (1<<2);
        delay(5000);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
        limpieza();
        alta2();
        apagado();
//APAGADO DE SALIDA TIPO RELAY MEDIANTE PIN DEL PÓRTICO A
        PORTA&=~(1<<0);
        PORTA&=~(1<<2);
        delay(3000);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
        limpieza();
//APAGADO DE SALIDA TIPO RELAY MEDIANTE PIN DEL PÓRTICO A
        PORTA&=~(1<<0);
        PORTA&=~(1<<2);
        PORTA&=~(1<<1);
        delay(5000);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
        limpieza();
        baja1();
//ENCENDIDO DE SALIDA TIPO RELAY MEDIANTE PINES DEL PÓRTICO A
        PORTA|= (1<<1);
        delay(3000);
//LLAMADO DE FUNCIONES DE LCD 16*2 Y LA INTERFA HUMANO MÁQUINA HMI
        limpieza();
        apagado1();

```

```

//APAGADO DE SALIDA TIPO RELAY MEDIANTE PIN DEL PÓRTICO A
    PORTA&=~(1<<0);
    PORTA&=~(1<<1);
    PORTA&=~(1<<2);
//ENCERRAR VARIABLES GLOBALES DEL SISTEMA DE CONTROL
    estado1=0;
    estado5=0;
    estado9=0;
    estado10=0;
    break;
    case 20:
//ENCENDIDO DE SALIDA TIPO RELAY MEDIANTE PINES DEL PÓRTICO A
    PORTA|=(1<<3);
    delay(100);
//APAGADO DE SALIDA TIPO RELAY MEDIANTE PIN DEL PÓRTICO A
    PORTA&=~(1<<3);
    delay(100);
    break;

}
} //FUNCIÓN PARA MODO 2
void proceso2(){
//CONDICIÓN DE RANGO DE TEMPERATURA Y MODO 2 SELECCIONADO
    if(temperatura<=30 &&estado==Seleccion_modo2){
//CONTADOR DE PULSO DEL PULSADOR PSEL POR FLANCO DE BAJADA
//ESCOGE LA VELOCIDAD DE ALTA O BAJA
        if((digitalRead(CONTROLLINO_A4)) != in_anterior){
            if((digitalRead(CONTROLLINO_A4))==LOW ){
                sel++;
            }
        }
        in_anterior = (digitalRead(CONTROLLINO_A4));
//CONDICIÓN DE SELECCION DE VELOCIDAD NO SER MAYOR A 3 EL CONTADOR DE PULSOS
//INGRESADO POR EL PANEL FRONTAL DE OCNTROLLINO MAXI
        if(sel==3){
            sel=0;
        }
//ADICIÓN DE VARIABLES INGRESADAS POR LOS PULSO DESDE EL PANEL FRONTAL DE CONTROLLINO MAXI Y
//LA INTERFAZ HUMANO MÁQUINA HMI
        sel1=sel+plus;
//CONDICIÓN DE PRIMER PULSO DE SELECCION DE VELOCIDAD MEDIANTE MODO 2
        if(sel1==1){
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
            estado7=14;
        }
//CONDICIÓN DE SEGUNDO PULSO DE SELECCION DE VELOCIDAD MEDIANTE MODO 2
        if(sel1==2){

```

```

//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
    estado7=15;
}
//CONDICIÓN DE TERCER PULSO DE SELECCION DE VELOCIDAD MEDIANTE MODO 2
    if(sel1==3){
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
        estado7=16;
    }
//CONDICIÓN MÚLTIPLE DE EVENTOS MEDIANTE LA SELECCIÓN DE VELOCIDAD
    switch (estado7){
//CASO PARA CUANDO SE SELECCIONA VELOCIDAD ALTA
        case 14:
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
            estado3=1; //alta
            estado4=0;
            break;
//CASO PARA CUANDO SE SELECCIONA VELOCIDAD BAJA
        case 15:
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
            estado3=0;
            estado4=1;
            break;
//CASO CUANDO SE PRESIONA TRES VECES EL PULSADOR DE SELECCIÓN DE VELOCIDAD
        case 16:
            if(sel==3){
//ENCERRAR VARIABLE DE CONTADOR DE PULSOS DE SELECCION DE VELOCIDAD
                sel=0;
            }
            break;
        }
//CONDICIÓN CUANDO SE PRESIONA PULSADOR P4 EN MODO 2
//CONTADOR DE PULSOS DE P4 CON FLANCO DE BAJADA
        if((digitalRead(CONTROLLINO_A5)) != in_pasada){
            if((digitalRead(CONTROLLINO_A5))==HIGH ){
                pulso2++;
            }
        }
        in_pasada = (digitalRead(CONTROLLINO_A5));
//ADICIÓN DE PULSOS PRESIONADO POR MEDIO DE PANEL FRONTAL DE CONTROLLINO MAXI Y
//INTERFAZ HUMANO MÁQUINA HMI
        pulso=pulso2+pulso3;
//CONDICIÓN DE PULSOS REALIZADOS POR P4
        if(pulso2==4){
//ENCERRAR VARIABLE DE CONTADOR DE PULSOS DE P4
            pulso2=0;
            in_pasada=0;
        }
    }

```



```

//CONDICIÓN DE PRESIONAR P4 EN MODO 2
if((digitalRead(CONTROLLINO_A5))==HIGH or pulso3==1 ){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ UNA PULSACIÓN DE P4
  if(pulso==1){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ LA SELECCIÓN DE VELOCIDAD ALTA
    if(estado7==14){
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
      estado2=8;
    }
  }
}
//CONDICIÓN DE PRESIONAR P4 EN MODO 2
if((digitalRead(CONTROLLINO_A5))==HIGH or pulso3==1 ){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ UNA PULSACIÓN DE P4
  if(pulso==1){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ LA SELECCIÓN DE VELOCIDAD BAJA
    if(estado7==15){
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
      estado2=9;
    }
  }
}
//CONDICIÓN DE PRESIONAR P4 EN MODO 2
if((digitalRead(CONTROLLINO_A5))==HIGH or pulso3==2){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ DOS PULSACIONES DE P4
  if(pulso==2){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ LA SELECCIÓN DE VELOCIDAD ALTA
    if(estado7==14){
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
      estado2=9;
    }
  }
}
//CONDICIÓN DE PRESIONAR P4 EN MODO 2
if((digitalRead(CONTROLLINO_A5))==HIGH or pulso3==2){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ DOS PULSACIONES DE P4
  if(pulso==2){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ LA SELECCIÓN DE VELOCIDAD BAJA
    if(estado7==15){
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
      estado2=8;
    }
  }
}
//CONDICIÓN DE PRESIONAR P4 EN MODO 2
if((digitalRead(CONTROLLINO_A5))==HIGH or pulso3==3){
//CONDICIÓN DE VERIFICACION CUANDO SE REALIZÓ TRES PULSACIONES DE P4

```

```

if(pulso==3){
//ALMACENAR ESTADO EN VARIABLE GLOBAL DEL SISTEMA DE CONTROL
    estado2=10;
}
}

switch (estado2){
//CASO DE FUNCIONAMIENTO DE SE PRESIONA P4 POR PRIMERA VEZ Y SELECCION DE VELOCIDAD ALTA
    case 8:
//LLAMADO A FUNCIONES DE LCD Y HMI PARA ENVIAR EL EVENTO SUCEDIDO
        alta1();
        Serial2.print("p5.pic=");
        Serial2.print("1");
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.print("p2.pic=");
        Serial2.print("1");
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.print("p3.pic=");
        Serial2.print("2");
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.print("p4.pic=");
        Serial2.print("2");
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.write(0xff);

//ENCENDIDO DE SALIDA TIPO RELAY MEDIANTE PINES DEL PÓRTICO A
        PORTA|=(1<<0);
        PORTA|=(1<<2);
//APAGADO DE SALIDA TIPO RELAY MEDIANTE PIN DEL PÓRTICO A
        PORTA&=~(1<<1);
        estado2=0;

        break;
    case 9:
//LLAMADO A FUNCIONES DE LCD Y HMI PARA ENVIAR EL EVENTO SUCEDIDO
        baja1();
        estado7=15;
        Serial2.print("p4.pic=");
        Serial2.print("1");
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.write(0xff);
        Serial2.print("p2.pic=");

```

```

Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p3.pic=");
Serial2.print("1");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p5.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
//APAGADO DE SALIDA TIPO RELAY MEDIANTE PIN DEL PÓRTICO A
PORTA&=~(1<<0);
PORTA&=~(1<<2);
//ENCENDIDO DE SALIDA TIPO RELAY MEDIANTE PINES DEL PÓRTICO A
PORTA|=(1<<1);
estado2=0;
break;
case 10:
//APAGADO DE VARIABLES GLOBALES DEL MODO 2
//LLAMADO A FUNCIONES DE LCD Y HMI PARA ENVIAR EL EVENTO SUCEDIDO
apagado();
apagado1();
limpieza();
estado2=0;
estado7=0;
sel,sel1,plus=0;
pulso=0;
pulso2=0;
pulso3=0;
//APAGADO DE SALIDA TIPO RELAY MEDIANTE DEL PÓRTICO A
PORTA=0x00;
break;
}
}
}
void caso(){
limpieza();
//APAGADO DE SALIDA TIPO RELAY MEDIANTE DEL PÓRTICO A
PORTA=0X00;
//ASIGNAR ESTADO A VARIABLE GLOBAL
estado=4;
//ENCERAR VARIABLES GLOBALES DEL SISTEMA DE CONTROL

```

```

estado1=0;
estado2=0;
sel1,sel,plus=0;
pulso,pulso2,pulso3=0;
estado6=0;
estado7=0;
estado9=0;
estado10=0;
//APAGADO DE INDICADORES DE LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p0.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p1.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p2.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p3.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p4.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p5.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
return loop;
}
void serialEvent() {
//HABILITAR LECTURA DE DATOS
while(Serial.available()){
comando=Serial.read();
//CONDICIONAL SEGÚN EL CARACTER INGRESO POR EL PUERTO SERIE
switch (comando){

```

```

//SELECCIÓN DE MODO 1 POR INTERFAZ HUMANO MÁQUINA HMI
    case ('a'):
//ASIGNAR ESTADO A VARIABLE GLOBAL
    estado=Seleccion_modo1;
//FUNCIONES DE IMPRESION EN LCD 16*2 E INTERFAZ HUMANO MÁQUINA HMI
    limpieza();
    modone();
    proceso();
    break;
//SELECCIÓN DE MODO 2 POR INTERFAZ HUMANO MÁQUINA HMI
    case ('b'):
//ASIGNAR ESTADO A VARIABLE GLOBAL
    estado=estado=Seleccion_modo2;
//ASIGNAR ESTADO A VARIABLE GLOBAL
    estado6=true;
//FUNCIONES DE IMPRESION EN LCD 16*2 E INTERFAZ HUMANO MÁQUINA HMI
    limpieza();
    modotwo();
    proceso2();
    break;
//SELECCION DE MODO OFF POR INTERFAZ HUMANO MÁQUINA HMI
    case ('c'):
C
    estado=Seleccion_off;
    estado5=false;
    estado6=false;
//FUNCIONES DE IMPRESIÓN EN LCD 16*2 E INTERFAZ HUMANO MÁQUINA HMI
    caso();
    return loop();
    break;
//BOTON DE INCREMENTO POR INTERFAZ HUMANO MÁQUINA HMI
    case ('d'):
//CONDICIÓN DE SELECCION INICIAL DE MODO 2
    if(estado6==true){
//CONTADOR DE PULSOS
    plus=plus+1;
    if(plus==3){
    plus=0;
    }
//FUNCION DE IMPRESIÓN EN LCD 16*2 E INTERFAZ HUMANO MÁQUINA HMI
    proceso2();
    }
    break;
//BOTÓN DE DECREMENTO POR INTERFAZ HUMANO MÁQUINA HMI
    case ('i'):
//FUNCIONES DE IMPRESION EN LCD 16*2 E INTERFAZ HUMANO MÁQUINA HMI
    estado5=4;

```

```

//FUNCIÓN VOID LOOP
    loop ();
//CONDICIÓN DE SELECCION INICIAL DE MODO 2
    if(estado6==true){
//CONTADOR DE PULSOS
    pulso3=pulso3+1;
//CONDICIÓN AL PULSAR 4 VECES EL P4
    if(pulso3==4){
//ASIGNACIÓN DE ESTADO A VARIABLES GLOBALES
    pulso3=0;
    }
    proceso2();}
    return;
    break;
//BOTÓN DE
    case ('f'):
//ASIGNACIÓN DE ESTADO A VARIABLES GLOBALES
    estado5=2;
    return;
    break;
//BOTÓN DE INCREMENTO DE TEMPERATURA
    case ('g'):
//CONTADOR CRECIENTE DE LA VARIBALE TEMPERATURA DE INTERFAZ HUMANO MÁQUINA HMI
    tem2=tem2+1;
    return;
    break;
//BOTÓN DE DECREMENTO DE TEMPERATURA
    case('h'):
//CONTADOR DECRECIENTE DE LA VARIBALE TEMPERATURA DE INTERFAZ HUMANO MÁQUINA HMI
    tem2=tem2-1;
    return;
    break;
}
}
}
void bienvenida(){
//IMPRESIÓN DE TEXTO EN LCD 16*2
    lcd.init();
    lcd.init();
    lcd.backlight();
    lcd.setCursor(3,0);
    lcd.print("SIMULACION");
    lcd.setCursor(2,1);
    lcd.print("CONTROLLINO");
//TIEMPO DE VISUALIZACIÓN EN LCD 16*2
    delay(1000);
//INSTRUCCION DE LIMPIEZA DE LCD 16*2

```

```

    lcd.clear();
}
void titulo(){
//IMPRESIÓN DE TEXTO EN LCD 16*2
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print(" ");
    lcd.print("CONTROL DE");
    lcd.setCursor(2,1);
    lcd.print(" ");
    lcd.print(" MOTOR  ");
//TIEMPO DE VISUALIZACIÓN EN LCD 16*2
    delay(1);
}
void seleccion(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print(" ");
    lcd.print(" SELECCIONAR  ");
    lcd.setCursor(0,1);
    lcd.print(" MODO ");
    lcd.setCursor(10,1);
    lcd.print(" T°:");
    lcd.print(temperatura);
//TIEMPO DE VISUALIZACIÓN EN LCD 16*2
    delay(1);
}
void modone(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print(" MODO 1  ");
    lcd.setCursor(0,1);
    lcd.print(" ");
    lcd.setCursor(10,1);
    lcd.print(" T°:");
    lcd.print(temperatura);
//TIEMPO DE VISUALIZACIÓN EN LCD 16*2
    delay(1);
}
void modotwo(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print(" ");
    lcd.print(" MODO 2  ");

```

```

lcd.setCursor(0,1);
lcd.print(" ");
lcd.setCursor(10,1);
lcd.print(" T°:");
lcd.print(temperatura);
//TIEMPO DE VISUALIZACIÓN EN LCD 16*2
delay(1);
}
void alta(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
lcd.backlight();
lcd.setCursor(0,0);
lcd.print(" ");
lcd.print(" SPEED HIGH ");
lcd.setCursor(0,1);
lcd.print(" ");
lcd.setCursor(10,1);
lcd.print(" T°:");
lcd.print(temperatura);
//ENCENDIDO DE INDICADOR SPEED HIGH EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p2.pic=");
Serial2.print("1");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
void alta1(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
lcd.backlight();
lcd.setCursor(0,0);
lcd.print(" ");
lcd.print(" SPEED HIGH ");
lcd.setCursor(0,1);
lcd.print("ON:KM1-KM3");
lcd.setCursor(10,1);
lcd.print(" T°:");
lcd.print(temperatura);
//ENCENDIDO DE INDICADOR KM1-KM3 EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p5.pic=");
Serial2.print("1");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
void alta2(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
lcd.backlight();

```



```

lcd.setCursor(0,0);
lcd.print(" ");
lcd.print("SPEED HIGH  ");
lcd.setCursor(0,1);
lcd.print("OFF:KM1-KM3");
lcd.setCursor(10,1);
lcd.print(" T°:");
lcd.print(temperatura);
//APAGADO DE INDICADOR SPEED HIGH EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p2.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
//APAGADO DE INDICADOR KM1-KM3 EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p5.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
//IMPRESIÓN DE TEXTO EN LCD 16*2
void alta3(){
lcd.backlight();
lcd.setCursor(0,0);
lcd.print(" SPEED HIGH  ");
lcd.setCursor(0,1);
lcd.print(" ");
delay(1);

}
void baja(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
lcd.backlight();
lcd.setCursor(0,0);
lcd.print(" ");
lcd.print("SPEED LOW  ");
lcd.setCursor(0,1);
lcd.print(" ");
lcd.setCursor(10,1);
lcd.print(" T°:");
lcd.print(temperatura);
//ENCENDIDO DE INDICADOR SPEED LOW EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p3.pic=");
Serial2.print("1");
Serial2.write(0xff);
Serial2.write(0xff);

```

```

Serial2.write(0xff);
}
void baja1(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
lcd.backlight();
lcd.setCursor(0,0);
lcd.print(" ");
lcd.print("SPEED LOW ");
lcd.setCursor(0,1);
lcd.print("ON:KM2 ");
lcd.setCursor(10,1);
lcd.print(" T°:");
lcd.print(temperatura);
//ENCENDIDO DE INDICADOR KM2 EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p4.pic=");
Serial2.print("1");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
void baja2(){
//IMPRESIÓN DE TEXTO Y VARIABLE DEL SISTEMA DE CONTROL EN LCD 16*2
lcd.backlight();
lcd.setCursor(0,0);
lcd.print(" ");
lcd.print("SPEED LOW ");
lcd.setCursor(0,1);
lcd.print("OFF:KM2");
lcd.setCursor(10,1);
lcd.print(" T°:");
lcd.print(temperatura);
//APAGADO DE INDICADORES EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p3.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p4.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
void baja3(){
//IMPRESIÓN DE TEXTO EN LCD 16*2
lcd.backlight();

```

```

lcd.setCursor(0,0);
lcd.print(" ");
lcd.print(" SPEED LOW ");
lcd.setCursor(0,1);
lcd.print(" ");
}
//FUNCIÓN DE LIMPIEZA DE LCD 16*2
void limpieza(){
lcd.backlight();
lcd.setCursor(0,0);
lcd.print(" ");
lcd.print(" ");
lcd.setCursor(0,1);
lcd.print(" ");
lcd.print(" ");
return;
}
void visualizacion(){
//ENCENDIDO DE INDICADORES EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p0.pic=");
Serial2.print("1");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p1.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
void visualizacion1(){
//ENCENDIDO DE INDICADORES EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p0.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p1.pic=");
Serial2.print("1");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
//ENCENDIDO DE INDICADORES EN LA INTERFAZ HUMANO MÁQUINA HMI
void indicadores(){
Serial2.print("n1.val=");

```

```
Serial2.print(pulso);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("n2.val=");
Serial2.print(sel1);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
void apagado(){
//APAGADO DE INDICADORES EN LA INTERFAZ HUMANO MÁQUINA HMI
Serial2.print("p2.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p5.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
//APAGADO DE INDICADORES EN LA INTERFAZ HUMANO MÁQUINA HMI
void apagado1(){
Serial2.print("p3.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.print("p4.pic=");
Serial2.print("2");
Serial2.write(0xff);
Serial2.write(0xff);
Serial2.write(0xff);
}
```

ANEXO E

MANUAL DE USUARIO

1. Configuración de Software Proteus v8.9

Para una la simulación de Controllino mediante el software Protesuv8.9, se debe cambiar las configuraciones de simulación SPICE del software por “Setting for better Convergence” en base a la Figura E.1, esto permitirá realizar una simulación correcta. Los valores predeterminados mediante la configuración no es necesario modificarlos. A continuación, se describen los pasos a seguir para el cambio de simulación SPICE.

Abrir Proteusv8.9

Dirigirse a la herramienta “**System**”

Seleccionar la opción “**Set Simulation Options**”

Posterior, elegir la opción “**Setting for better Convergence**”

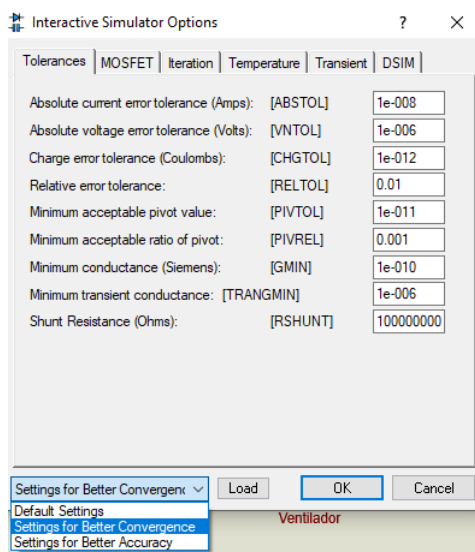


Figura E.1 Configuración de Simulación

2. Configuración y carga del algoritmo de control en microcontrolador Controllino

Para la carga del algoritmo de control a Controllino, es recomendable realizar la configuración respectiva al microcontrolador de Controllino (ver Figura E.2) y disponer del archivo o dirección con formato hex. A continuación, se detallan los pasos para realizar la configuración y carga del archivo al microcontrolador.

- 1) Seleccionar el microcontrolador de Controllino
- 2) Elegir la opción “**Edit properties**” del microcontrolador
- 3) Modificar los parámetros de microcontrolador:
 - CLKDIV8 (Divide clock by 8) por Unprogrammed

- CKSEL Fuses por (000) Ext. Clock
 - Clock Frequency por (16MHz)
- 4) Carga de algoritmo de control:
- Compilar algoritmo de control mediante el compilador Arduino Ide sketch
 - Seleccionar la opción “**Programa**” del compilador Arduino Ide sketch
 - Elegir la opción “**Exportar a binarios compilados**” del compilador Arduino Ide sketch
 - Elegir la opción “**Edit properties**” del microcontrolador
 - Ingresar la dirección del archivo hex en “**Program Files**” del microcontrolador
- 5) Finalmente seleccionar la “**Ok**” para confirmar cambios realizados

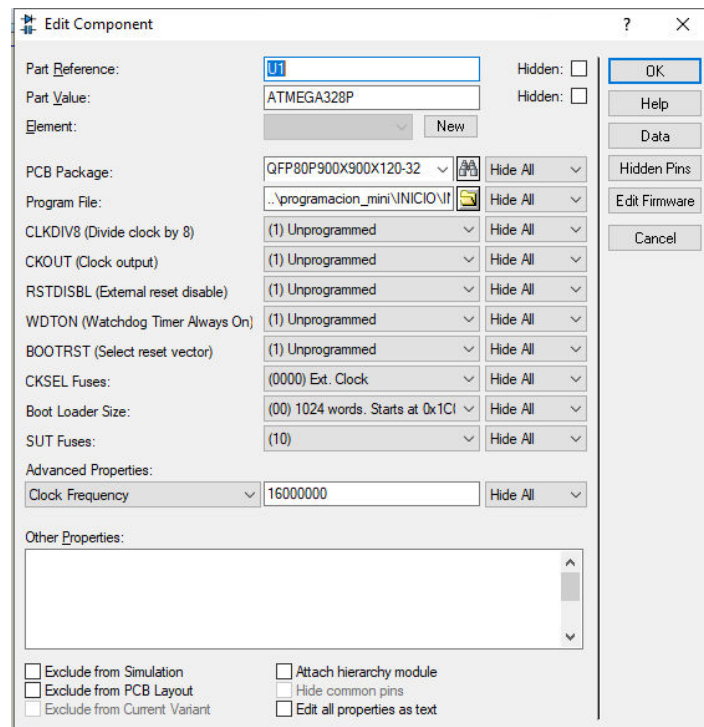


Figura E.2 Configuración del microcontrolador Controllino

3. Configuración y carga de la Interfaz Humano Máquina HMI mediante Nextion Editor

Para la utilizar la HMI mediante Nextion Editor se requiere compilar la interfaz desarrollada, además se debe seleccionar el puerto de comunicación con Controllino. La velocidad de

transmisión depende de la inicialización del puerto serial a ser utilizado por el algoritmo desarrollado mediante el compilador Arduino Ide Steck.

La velocidad de transmisión de datos entre Nextion Editor y Proteus v8.9 son iguales, para evitar inconvenientes o pérdidas de información. A continuación, se detallan los pasos para la simulación de la Interfaz Humano Máquina HMI:

- 1) Seleccionar “**Debug**”.
- 2) Elegir la opción “**User MCU Input**”.
- 3) Colocar el Puerto Serial en “**Com Port**”.
- 4) Ingresar la velocidad de comunicación en “**Baud**”.
- 5) Finalmente, Dar en “**Start**”.

En la Figura E.3 se presenta la ventana en la cual se visualizan los datos de recepción y envío de la Interfaz Humano Máquina HMI. En la parte A, se visualiza los valores, en caso de estar en línea con el Controlador Lógico Programable Controllino y en la parte B, se visualizan los datos en caso de conexión virtual para simulación.

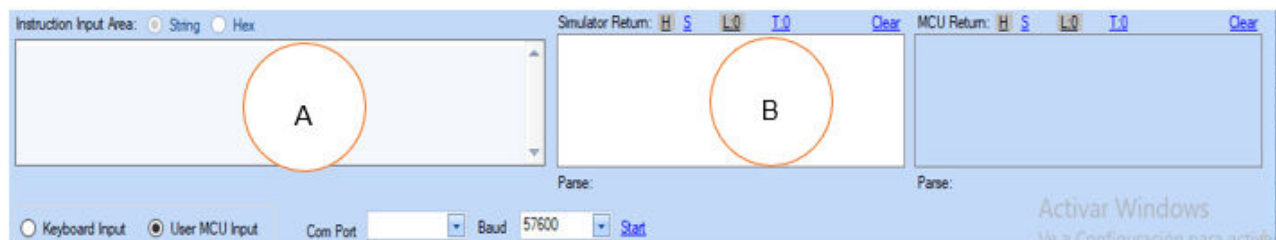


Figura E.3 Ventana de simulación Interfaz Humano Máquina HMI.

Para comunicación Controllino hacia Interfaz Humano Máquina Multimarca (Siemens, Delta, Allen Bradley, etc) se requiere de comunicación Modbus o servidor opc.

Para complementar el respectivo Manual de Usuario y mejorar la comprensión de cada uno de los pasos a cumplir para simular los sistemas de control desarrollados en Controllino se ha realizado un video explicativo el cual se puede visualizar en el siguiente link:

<https://1drv.ms/u/s!AtlDqOH7YULugbcCNwltpk1y-AnGiA?e=zsiK8l>