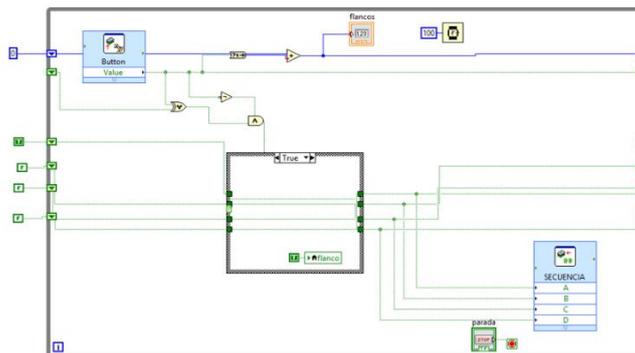
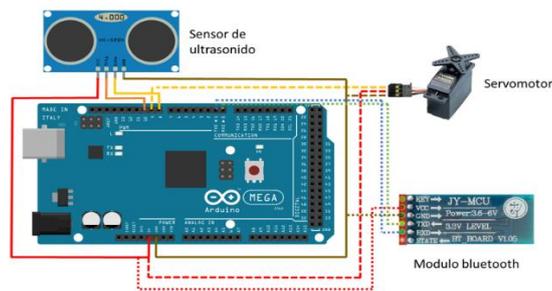
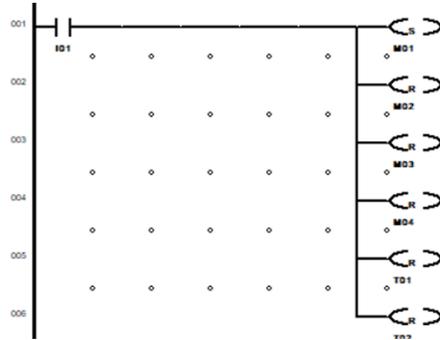


# GUÍA DE LABORATORIO DE MECATRÓNICA I

## LMEC 432



Febo Flores

## Contenido

PRÁCTICA 1.....	4
Inversión de giro de motor trifásico con automatismos.....	4
PRÁCTICA 2.....	8
Inversión de giro motor trifásico con autómata programable .....	8
PRÁCTICA 3.....	12
Arranque secuencial de motores con autómata programable .....	12
PRÁCTICA 4.....	15
Arranque estrella triángulo con autómata programable.....	15
PRÁCTICA 5.....	23
Funciones de programación especial, sensores de proximidad tres hilos.....	23
PRÁCTICA 6.....	28
Control motor paso a paso.....	28
PRÁCTICA 7.....	35
Inversión de giro de un motor CC con autómata programable .....	35
PRÁCTICA 8.....	38
Aplicaciones con señales analógicas .....	38
PRÁCTICA 9.....	49
Comunicación con un HMI .....	49
PRÁCTICA 10.....	55
Programas en Arduino .....	55
PRÁCTICA 11.....	61
Introducción Arduino (UNO) entradas-salidas digitales y conexión interfaz.....	61
PRÁCTICA 12.....	64
Adquisición datos LabVIEW con usb 6001 (NI) .....	64
PRÁCTICA 13.....	69
Control con LabVIEW (MyRIO NI), secuencias .....	69
PRÁCTICA 14.....	73
Integración Arduino (UNO) y LabVIEW, almacenamiento de datos .....	73
(ANEXO).....	76
PRÁCTICA 15-A .....	77
Aplicación de Arduino y shields en proyecto TRONIC.....	77
PRÁCTICA 15-B .....	96
Comunicación en protocolo MODBUS entre Arduino y HMI .....	96
GUÍA BÁSICA DE AUTÓMATA PROGRAMABLE PC WORX .....	102
GUÍA BÁSICA DE AUTÓMATA PROGRAMABLE MICROLOGIX 1100B .....	111



## PRÁCTICA 1

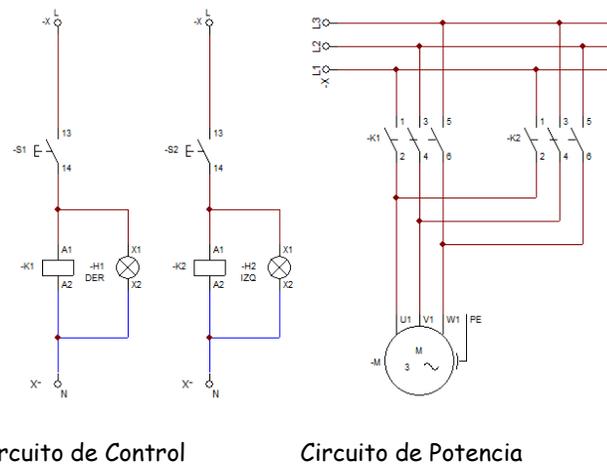
### Inversión de giro de motor trifásico con automatismos

**OBJETIVO:** Control combinacional ejecutado con automatismos aplicado a inversión de giro de un motor trifásico jaula de ardilla.

**COMPETENCIA:** Aplica control combinacional en inversión de giro de un motor trifásico jaula de ardilla, ensamblando el circuito con automatismos en base a contactores.

#### FUNDAMENTO TEÓRICO

Los automatismos están siempre compuestos de circuitos de control o mando (lado izquierdo de la figura) y de potencia o fuerza (lado derecho de la figura).



Circuito de Control

Circuito de Potencia

Figura 1: Diagrama de conexión inversión de sentido de rotación motor 3 $\phi$

En el circuito de control se aprecia que dependiendo si se pulse S1 o S2, se accionará la bobina K1 o K2 que tienen accionamiento en sus contactos auxiliares normalmente abiertos mostrados en el circuito de potencia.

Mientras se mantiene activado K1;

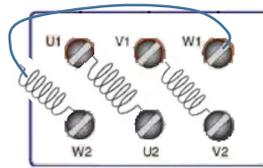
- L1 conecta U1,
- L2 conecta V1,
- L3 conecta W1.

Mientras que si se activa K2;

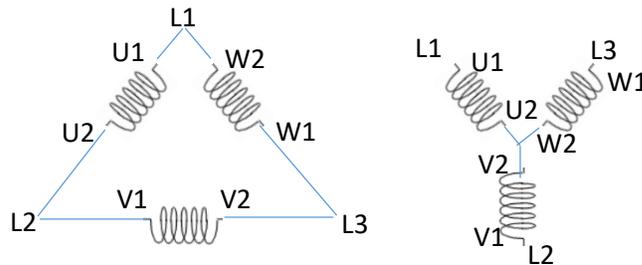
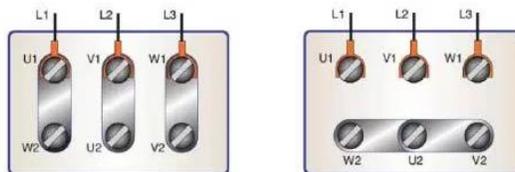
- L1 conecta U1,
- L2 conecta W1,
- L3 conecta V1.

Es como si con las manos estuviera cambiando de:

L2 – V1 y L3 – W1 a: L2 – W1 y L3 – V1



Distribución de las bobinas



Conexión triángulo

Conexión estrella

Figura 2: Bornes del motor trifásico de inducción o jaula de ardilla

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Medir con el voltímetro el voltaje de línea a línea de la red trifásica.

Medir con voltímetro el voltaje de línea a neutro de la red trifásica.

Medir con voltímetro el voltaje de línea a tierra de la red trifásica.

Armar el circuito de control como se muestra en la figura 1.

Pulsar S1 y verificar que mientras se pulse el contactor K1 se acciona así como la lámpara o señalizador conectado en paralelo.

Del mismo modo, mientras se pulse S2, se accionará K2 con su señalizador.

Armar el circuito de potencia como se muestra en la figura 1, nótese que el motor tiene seis bornes, que vienen a ser las terminales de las tres bobinas internas del motor trifásico.

Si conecta en modo triángulo, el voltaje que llega a cada bobina del motor es la correspondiente de línea a línea, si en cambio conecta en estrella, el voltaje que llega a los extremos de cada bobina es de  $V/\sqrt{3}$  Verifique con un voltímetro.

Observar si al pulsar S1 el sentido de rotación mirando hacia el motor es horario o antihorario, debe cambiar el sentido con S2.

Notar que el motor es energizado mientras se presiona S1 o S2, de lo contrario éste deja de ser accionado eléctricamente y la rotación es solamente por inercia.

NO PRESIONE SIMULTANEAMENTE S1 CON S2, de lo contrario se producirá un corto circuito con el voltaje de la red trifásica.

Precisamente para evitar riesgos de hacer cortos circuitos, anexar a los contactores, contactos auxiliares NC (Normally Closed) o normalmente cerrados y hacer la siguiente modificación:

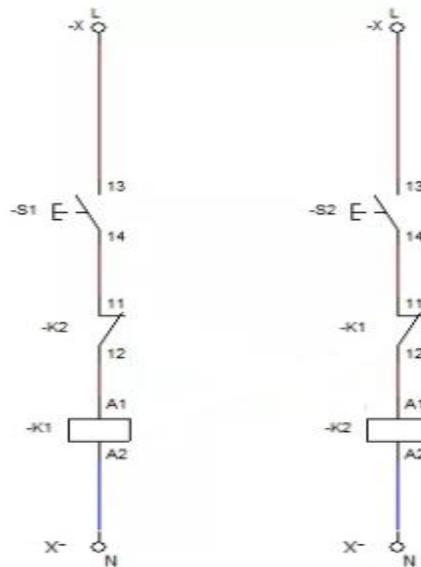


Figura 3: Circuito de control con enclavamiento eléctrico

Los contactos auxiliares normalmente cerrados NC (11 – 12) cortan la alimentación a la bobina opuesta, de manera de evitar que ambas se energicen simultáneamente. Existen también relés para enclavamiento mecánico que pueden adicionarse para brindar mayor seguridad al presente circuito. Para protección del motor contra sobrecargas débiles y prolongadas suele incluirse en la instalación, un relé térmico que contiene contactos auxiliares a adicionarse al circuito de control. Al adicionarse un relé térmico deben emplearse los terminales (NC) 95 y 96 en el circuito de control, de manera que al incrementarse la temperatura, dicho contacto se abre y desenergiza a las bobinas de los contactores. Contactos 97 y 98 pueden usarse para encender una lámpara de alarma.

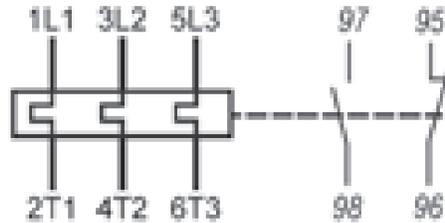


Figura 4: Relé Térmico, Extraído del catálogo de WEG Automatización-Contactores y relés de sobrecarga

#### MATERIAL:

- Red trifásica de CA
- Disyuntor electromagnético
- Contactor termomagnético
- Pulsadores de tablero eléctrico
- Señalizadores de tablero eléctrico
- Cables Nº 10
- Herramientas: Multímetro, desarmadores y alicates para cables

#### CUESTIONARIO

1. ¿Cuánto es la diferencia de voltaje entre líneas, línea a neutro, línea a tierra y entre neutro a tierra en la red trifásica del laboratorio?
2. Indique las diferencias entre el circuito mostrado en la figura del fundamento teórico y el circuito armado en laboratorio.
3. ¿Cómo identifica las terminales de la bobina del contactor?
4. ¿Cuál es el principio de funcionamiento del enclavamiento mecánico?
5. ¿Cuáles son las diferencias entre disyuntor termomagnético y relé térmico?
6. Dibuje los esquemas eléctricos de los circuitos como realmente fueron realizados en la práctica

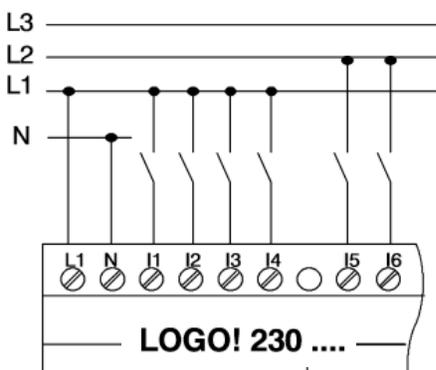
## PRÁCTICA 2

### Inversión de giro motor trifásico con autómatas programables

**OBJETIVO:** Control combinacional ejecutado con micro autómatas aplicado a inversión de giro de un motor trifásico de inducción o jaula de ardilla.

**COMPETENCIA:** Opera control combinacional de inversión de giro de un motor trifásico jaula de ardilla, ensamblando y programando contactores con micro autómatas programables.

#### FUNDAMENTO TEÓRICO

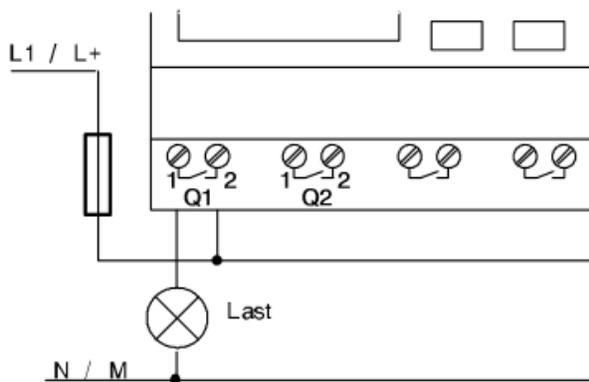


Los modelos LOGO! 230... cuentan con entradas divididas en dos grupos de cuatro entradas.

Estado operacional 0: < 40 V CA o < 30 V CC

Estado operacional 1: > 79 V CA o > 79 V CC

Figura 1: Entradas en modelo LOGO, extraído de manual LOGO!



Los modelos con salida de relé para operación en CA o CC, se recomienda instalación de un fusible de hasta 16 A en la línea L1/L+.

La ventaja de salidas de relé es que pueden manejarse actuadores que con CA o CC y cualquier voltaje, siempre y cuando la corriente de operación no exceda la permitida por los contactos del autómatas

Figura 2: Salidas de relé, extraído de manual LOGO!

Revise el video realizado por un antiguo estudiante de la asignatura:

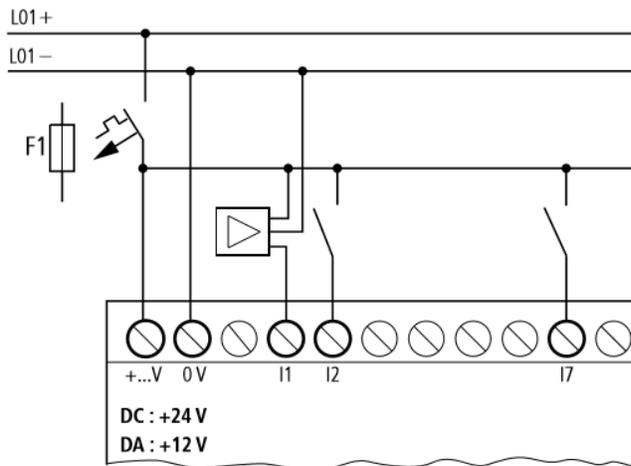
<https://youtu.be/9abWngomMpk>



El modelo EASY Messenger con salida de relé para operación en CA o CC, se recomienda instalación de un fusible de hasta 16 A.

La novedad de este modelo es la posibilidad de enviar y recibir mensajes SMS. Es decir, realizar control mediante teléfono móvil con un alcance y cobertura en función del chip de la empresa de telefonía móvil.

Figura 3: Aspecto físico de micro autómatas EASY-Messenger!



Existen dos tipos de entrada de autómatas programables PNP o NPN o del inglés “sourcing” o “sinking” respectivamente.

Cuando la entrada es del tipo PNP (sourcing), las entradas son alimentadas de la línea viva o positiva, como se ve en la figura con entradas I2 e I7, mientras que en I1 está conectado un sensor de proximidad de tres hilos también configurado PNP.

Figura 4: Conexiones físicas en un PLC de entradas PNP

Aplicación con finales de carrera: El autómatas programables tiene cuatro entradas; I01, I02, L01 y L02, que son respectivamente; pulsador para rotación sentido horario, pulsador para rotación sentido anti horario, final de carrera horario y final de carrera anti horario.

Se emplearán dos salidas del autómatas programables; Q01 y Q02 para mando sentido horario y sentido antihorario respectivamente.

Para el mando combinacional y a fin de asegurar operación segura se realiza una tabla de verdad para verificar todas las combinaciones posibles de las entradas y evaluar en que condiciones se obtiene operación segura.

I01	I02	L01	L02	Q01	Q02
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Fig. 5: Tabla de verdad operación de motor

$$Q01 = I01 \times \overline{I02} \times \overline{L01} \times \overline{L02} + I01 \times \overline{I02} \times L01 \times \overline{L02} = I01 \times \overline{I02} \times \overline{L02}$$

$$Q02 = \overline{I01} \times I02 \times \overline{L01} \times \overline{L02} + \overline{I01} \times I02 \times \overline{L01} \times L02 = \overline{I01} \times I02 \times \overline{L01}$$

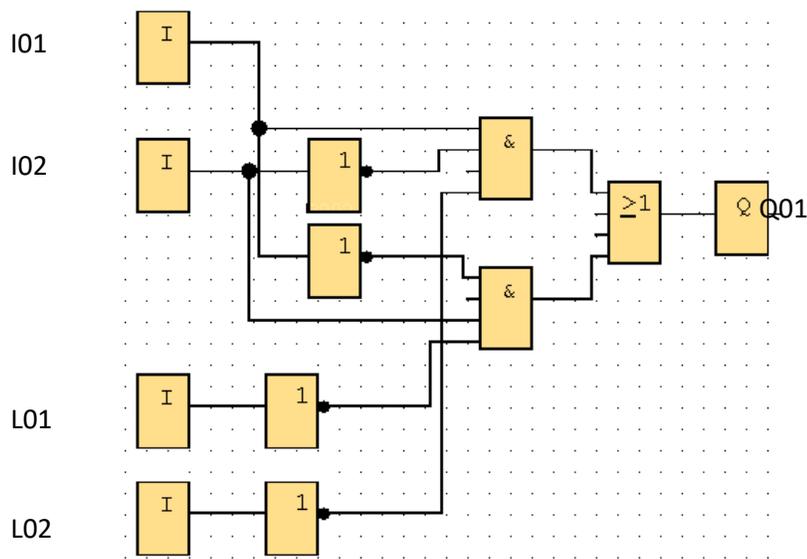


Fig. 6: Diagrama de bloques correspondiente a salida PLC Q01

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Armar el circuito de potencia como el mostrado en la figura de la práctica anterior.

Conectar los pulsadores y finales de carrera a entradas I1, I2, I3 e I4 y salidas Q1 y Q2 que alimentan K1 y K2 respectivamente del autómata programables.

Programar el circuito según las ecuaciones obtenidas de la tabla de verdad de la figura 5. La figura 6 muestra la lógica de Q1.

Completar el diagrama de la figura 6 con salida Q2 para cambio del sentido de rotación.

Repetir la experiencia pero en diagrama escalera o diagrama de contactos.

Una vez simulados los programas, verifique la comunicación entre la computadora y el autómatas, muchos autómatas cuentan con terminales RS232 que las computadoras ya no llevan, en ese caso debe usarse un cable convertidor RS232 a puerto USB, éste debe incluir un software o “driver” que proporciona el fabricante del cable convertidor (figura 7).



Figura 7: USB a RS232

#### MATERIAL:

- Red trifásica de CA
- Disyuntor electromagnético
- Contactor termomagnético
- Pulsadores de tablero eléctrico
- Señalizadores de tablero eléctrico
- Cables Nº 10
- Micro autómatas programable de salida de relé (lenguaje escalera)
- Micro autómatas programable de salida de relé (lenguaje de bloques lógicos)
- Herramientas: Multímetro, desarmadores y alicates para cables

#### CUESTIONARIO:

1. Realice las mismas aplicaciones empleando “Boolean function” del diagrama de bloques disponible en eSms Config de Messenger
2. Dibuje cómo conectaría señales todo o nada en autómatas con entrada NPN (sinking).
3. Dibujar el programa en lenguaje escalera (de contactos).
4. Dé un ejemplo de aplicación de control combinacional.
5. Dibuje los esquemas eléctricos de los circuitos como realmente fueron realizados en la práctica.

## PRÁCTICA 3

### Arranque secuencial de motores con autómatas programables

**OBJETIVO:** Control secuencial en función a eventos aplicado en arranque de dos motores comandados por pulsadores.

**COMPETENCIA:** Arranca dos motores de inducción trifásicos en secuencia y comandados por pulsadores, realizando el cableado y programación en un micro autómatas programables con diagrama de bloques y de contactos.

#### FUNDAMENTO TEÓRICO

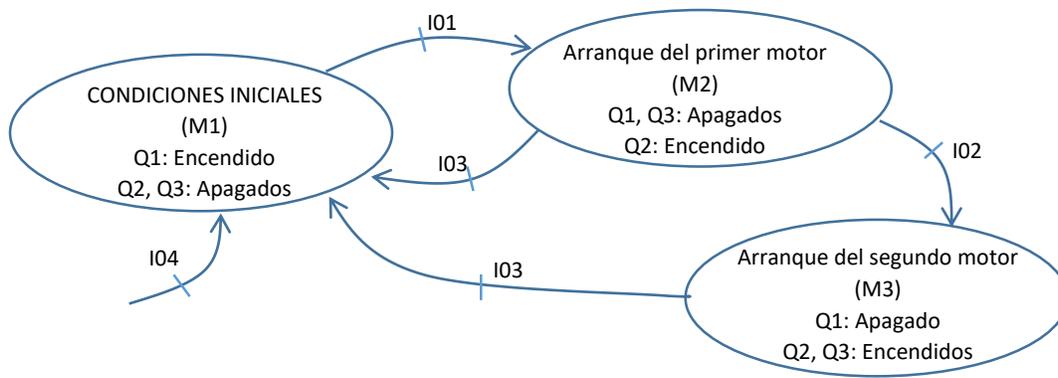


Figura 1: Parte secuencial del programa

M1	M2	M3	Q1	Q2	Q3
1	0	0	1	0	0
0	1	0	0	1	0
0	0	1	0	1	1

Tabla 1: Tabla de verdad suponiendo que sólo una marca se activa por vez

Las ecuaciones para Q1, Q2 y Q3 suponiendo que sólo se activa una marca  $M_i$  por vez:

$$Q1 = M1$$

$$Q2 = M2 + M3$$

$$Q3 = M3$$

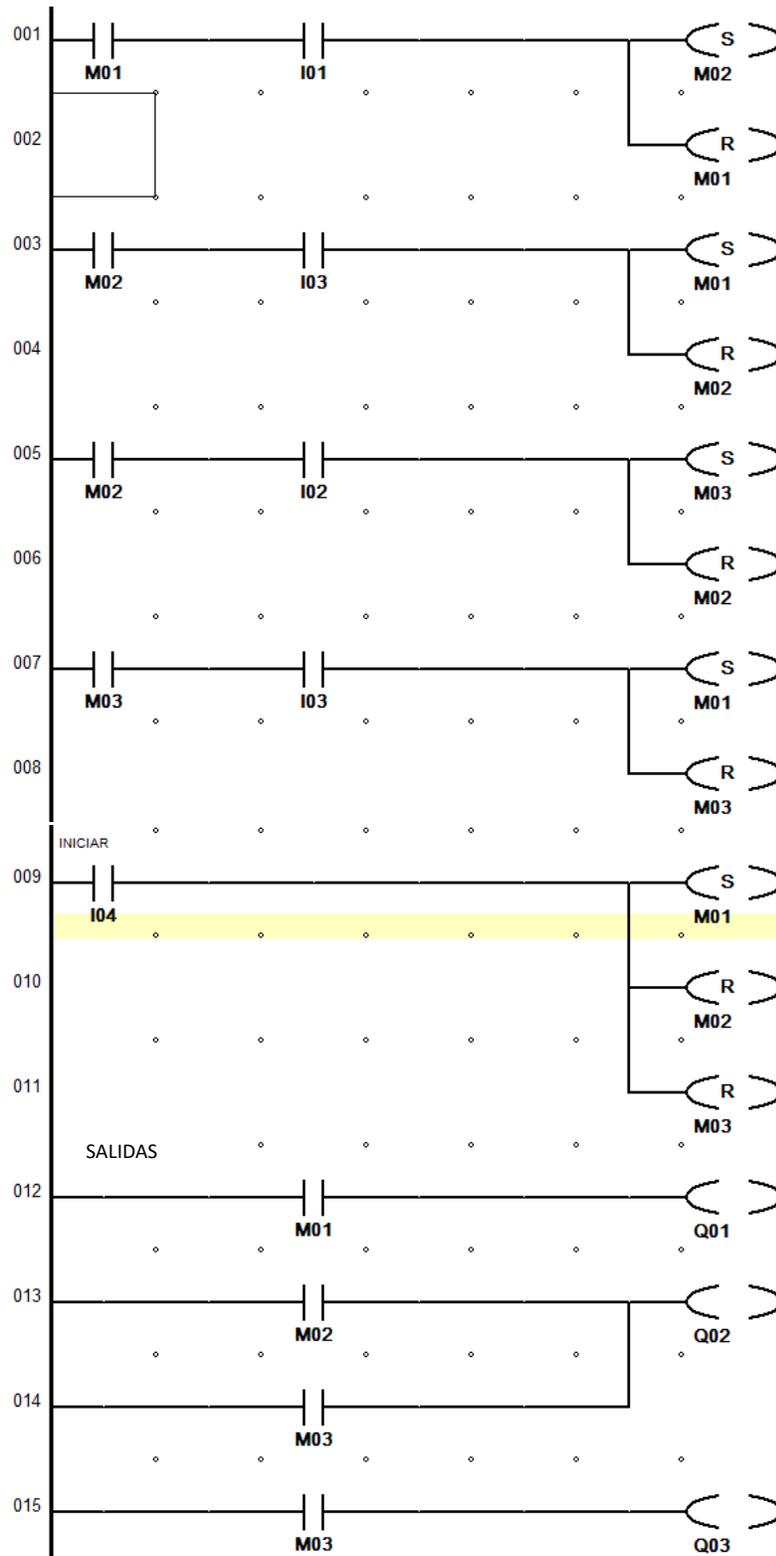


Figura 2: Programa en diagrama de contactos (escalera, "ladder" o "KOP")

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Una vez editado el programa de la figura 2 en el autómata programable, simularlo para verificar si se cumplen todas las transiciones mostradas en la figura 1, comprobado el correcto funcionamiento, baje el programa al autómata y arme el circuito correspondiente, reemplazando Q1, Q2 y Q3 por lámparas o contactores; I1, I2, I3 e I4 con pulsadores e instale al autómata programable para comprobar funcionamiento.

Realice el mismo programa en diagrama de bloques y simule la operación en los correspondientes simuladores, luego baje el programa al micro autómata Messenger mediante el bus de comunicación Modbus RTU. Puede hacer monitoreo del funcionamiento con la computadora haciendo uso de “Online test” 

**MATERIAL:**

- Red trifásica de CA
- Disyuntor electromagnético
- Contactor termomagnético
- Pulsadores de tablero eléctrico
- Señalizadores de tablero eléctrico
- Cables Nº 10
- Micro autómata programable de salida de relé (lenguaje escalera)
- Micro autómata programable de salida de relé (lenguaje de bloques lógicos)
- Herramientas: Multímetro, desarmadores y alicates pela cables

**CUESTIONARIO:**

1. Dibuje los esquemas eléctricos de los circuitos como realmente fueron realizados en la práctica.
2. Dibuje el diagrama GRAFCET correspondiente al grafo de la figura 1.
3. Indique las modificaciones que tendría que hacer en el programa para el grafo mostrado en la siguiente figura 3 con respecto al de la figura 1.
4. Explique cuál es la diferencia de salidas físicas con marcas o banderas.

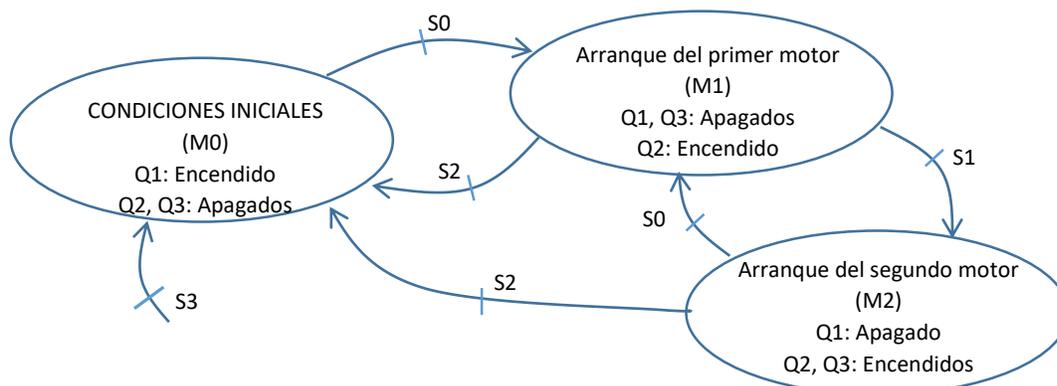


Figura 3: Grafo para representar en diagrama de contactos

## PRÁCTICA 4

### Arranque estrella triángulo con autómatas programables

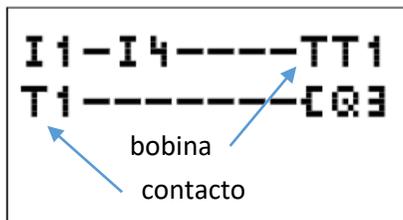
**OBJETIVO:** Aplicación del método de grafos para programar arranque de un motor en modalidad estrella triángulo haciendo uso de un micro autómatas programables.

**COMPETENCIA:** Arranca un motor de inducción trifásico, realizando el cableado y programación en un micro autómatas programables con diagrama de bloques y de contactos.

**FUNDAMENTO TEÓRICO:** El arranque estrella triángulo, a diferencia de la práctica anterior (secuencia en base a eventos), es una secuencia en base a tiempo transcurrido por lo que se hace necesario el uso de funciones de temporización.

Existen varios tipos de funciones de temporización, pero lo recomendable es recurrir a los manuales que presentan los fabricantes aunque todos ellos traten del mismo tipo de funcionamiento, es posible encontrar ligeras diferencias entre uno y otro, por lo que es recomendable hacer interpretación de las curvas temporales que presentan los mismos.

La marca EASY por ejemplo, ofrece 16 relés temporizadores T1 a T16 a escoger. Con relés temporizadores puede cambiarse el tiempo de conexión y la hora de conexión y desconexión de un contacto de manobra.



Los tiempos de retardo regulables se encuentran entre 2 ms y 99 h 59 min. Como valores de consigna se pueden utilizar valores positivos, valores de entradas analógicas, valores reales de relés contadores y relés temporizadores.

Nótese la diferencia en representar bobina con contacto.

Fig. 1: Contacto T1 y bobina TT1 de un relé temporizador

Función de las bobinas

Representación del esquema de contactos	Indicador easy	Función de las bobinas	Ejemplo
	<b>C</b>	Función de contactor	C01, C02, C54, C:1, CM1
	<b>J</b>	Función de contactor con resultado denegado	J01, J02, JS4
	<b>lr</b>	Impulso de ciclo con flanco negativo	lr03, lrM4, lrD0, lrS1
	<b>lr</b>	Impulso de ciclo con flanco positivo	lr04, lrM5, lrD1, lrS3
	<b>J</b>	Función de impulso de corriente	J03, JM4, JD0, JS1
	<b>S</b>	Activar (enclavar)	S00, SM2, SD3, SS4
	<b>R</b>	Desactivar (desenclavar)	R04, RM5, RD1, RS3

Los relés auxiliares M y N se utilizan como "marcas". El relé S puede utilizarse como salida de una ampliación o como marca, en caso de que no exista ampliación. Sólo se diferencian del relé de salida Q por no tener ningún borne de salida.

Fig. 2: Funciones de bobina de autómatas EASY - Moller

**Valoración de flancos negativos (impulso de ciclo)  $lr$**



Si la bobina sólo debe conectarse en un flanco negativo, se utilizará esta función. En caso de caída del estado de la bobina de "1" a "0", la bobina conecta durante un tiempo de ciclo sus contactos de cierre en el estado "1".

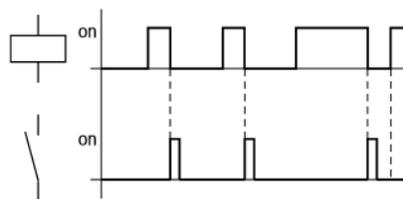


Fig. 42: Diagrama de comportamiento Impulso de ciclo con flanco negativo

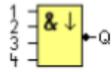
Representación en easy:

- Marcas M, N:  $lrM1$  a  $lrM16$ ,  $lrN1$  a  $lrN16$
- Saltos:  $lr:1$  a  $lr:8$



Puesto que se generará un impulso de ciclo, no deberán utilizarse salidas físicas.

Fig. 3: Operaciones de flanco en autómatas programables diagrama escalera, Fuente EASY x-Messenger



The output of a NAND with edge evaluation is only 1 **at least one** input is 0 and **all** inputs were 1 during the last cycle.

The output is set to 1 for the duration of one cycle and must be reset to 0 at least for the duration of the next cycle before it can be set to 1 again.

A block input that is not used (x) is assigned:  $x = 1$ .

Timing diagram of a NAND with edge evaluation

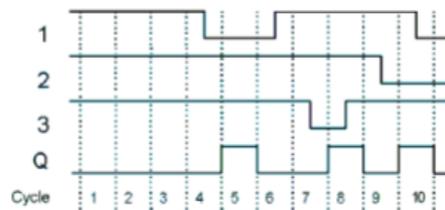


Fig. 4: Operaciones de flanco en autómatas programables con esquema de funciones, Fuente EASY x-Messenger

Puede utilizarse el método de grafos, como grafacet, en caso de escogerse el primero, la aplicación de arranque estrella triángulo sería el siguiente:

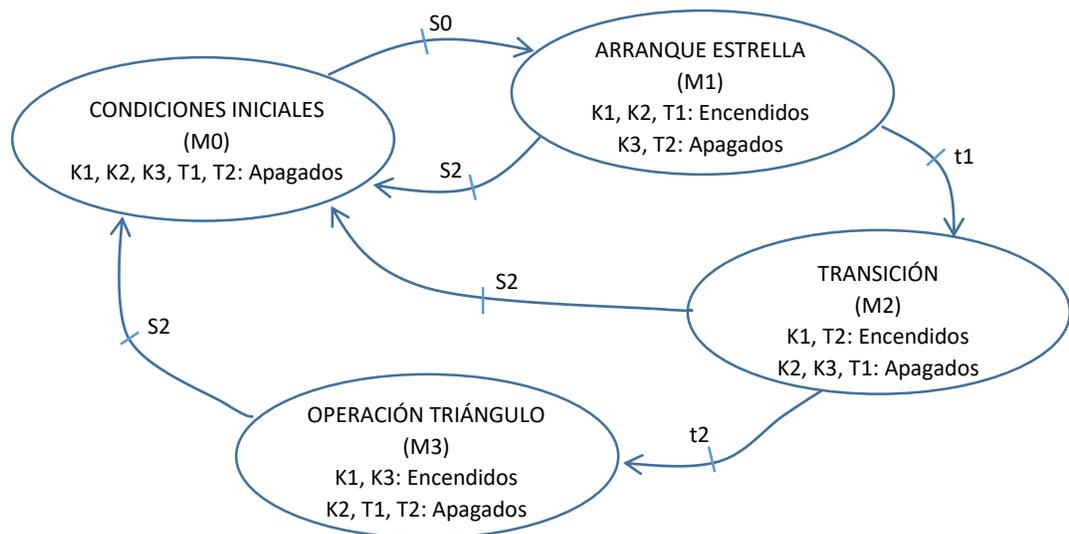


Fig. 5: Grafo para secuencia de arranque seguro estrella triángulo

Donde:

- K1 es contactor principal de línea
- K2 es contactor de operación estrella

- K3 es contactor de operación triángulo
- T1 es la bobina del temporizador de retardo de activación, por ejemplo 50 ms
- T2 es la bobina del temporizador de retardo de activación, contabiliza por ejemplo 10 s
- S0 es pulsador de encendido
- S2 es pulsador de parada
- t1 es contacto del temporizador T1
- t2 es contacto del temporizador T2

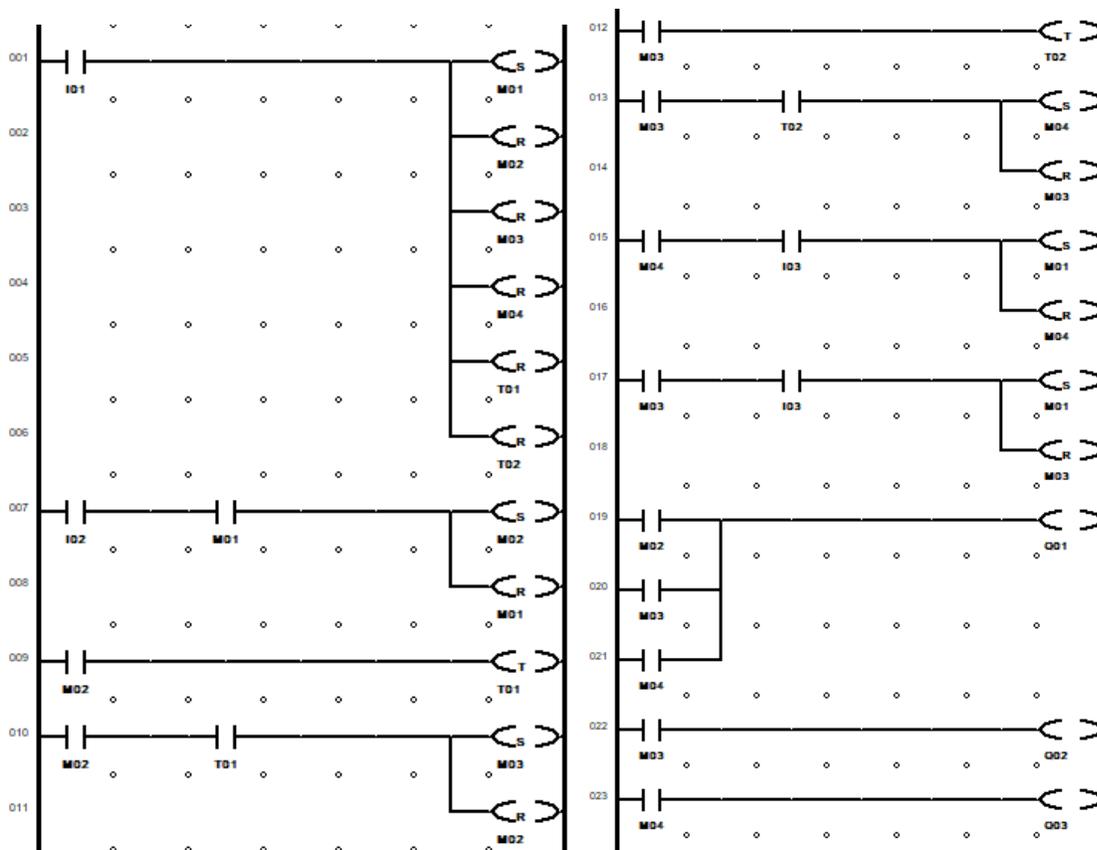


Fig. 6: Diagrama escadera para arranque estrella triángulo

Es también posible realizar un control combinando cableado con programa según se muestra en la siguiente figura:

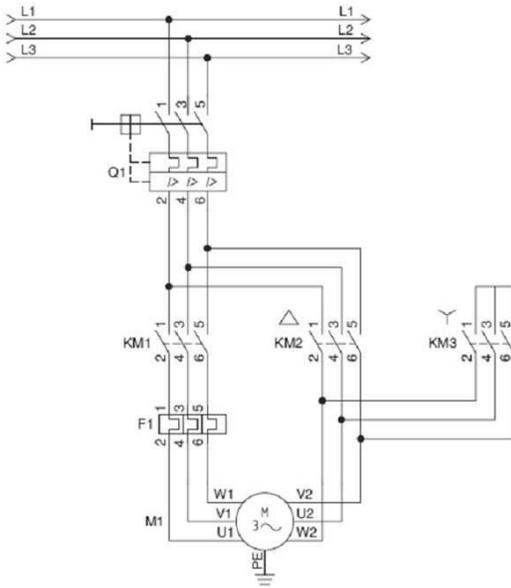


Fig. 7: Circuito potencia arranque estrella-triángulo

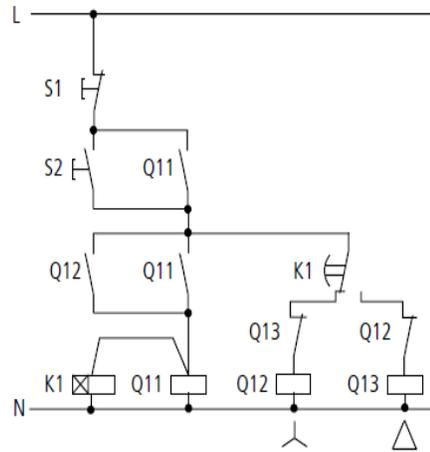


Fig. 8: Circuito de control con automatismo

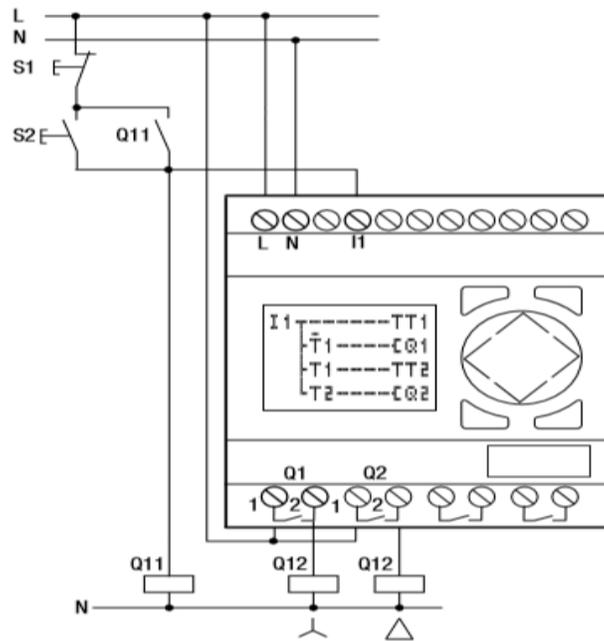


Fig. 9: Control con autómatas EASY-Moller para arranque estrella triángulo (ejemplo obtenido de Manual de usuario 01/05 AWB2528-1508E)

Especificaciones de temporizadores proporcionados por fabricantes de micro autómatas programables disponibles en laboratorio.

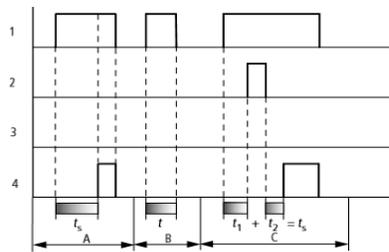


Diagrama de comportamiento Relé temporizador con temporización de trabajo (con/sin conexiones aleatorias)

- 1: bobina de disparo TTx
- 2: bobina de detención HTx
- 3: bobina reset RTx
- 4: contacto de maniobra (contacto de cierre) Tx

- $t_s$ : tiempo nominal
- Margen A: el tiempo nominal ajustado se para de forma normal.
- Margen B: el tiempo nominal ajustado no se para porque la bobina de disparo se desexcita demasiado pronto.
- Margen C: la bobina de detención detiene el transcurso del tiempo.

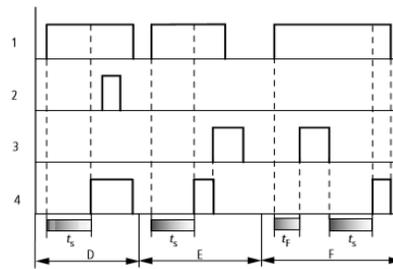


Diagrama de comportamiento Relé temporizador con temporización de trabajo (con/sin conexiones aleatorias)

- Margen D: la bobina de detención deja de tener efecto una vez transcurrido el tiempo
- Margen E: la bobina reset desactiva el relé y el contacto
- Margen F: la bobina reset desactiva el tiempo durante el transcurso. Una vez se ha desexcitado la bobina reset, el tiempo se para de forma normal.

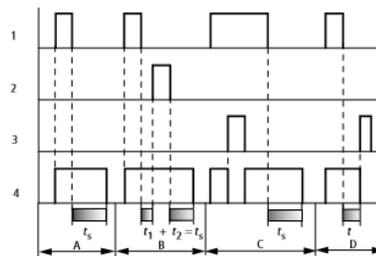


Diagrama de comportamiento Relé temporizador con temporización de reposo (con/sin conexiones aleatorias)

- 1: bobina de disparo TTx
  - 2: bobina de detención HTx
  - 3: bobina reset RTx
  - 4: contacto de maniobra (contacto de cierre) Tx
- $t_s$ : tiempo nominal

- Margen A: tras la desconexión de la bobina de disparo el tiempo se para.
- Margen B: la bobina de detención detiene el transcurso del tiempo.
- Margen C: la bobina reset desactiva el relé y el contacto. Una vez se desexcita la bobina reset, el relé sigue trabajando de forma normal.
- Margen D: la bobina reset desactiva el relé y el contacto durante el transcurso del tiempo.

Fig. 10: Diagramas de operación de temporizador; Fuente EASY-Moller

View in xLogic	Name of the special function	Rem
Timer		
	On-delay	REM
	Off-delay	REM
	On-/Off-delay	REM
	Retentive on-delay	REM

Fig. 11: Operadores de temporización en diagrama de esquema de funciones o de bloques; Fuente LOGO

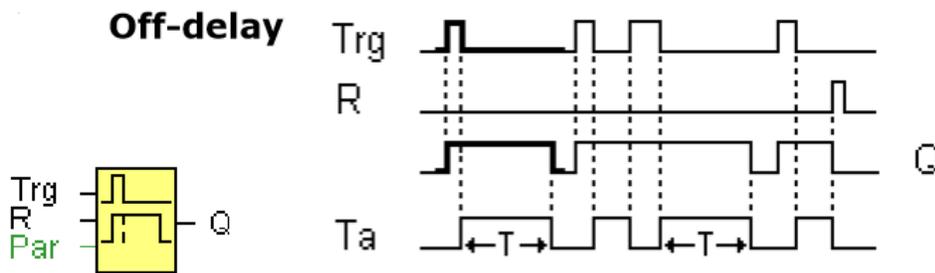


Fig. 12: Diagramas de operación de temporizador; Fuente LOGO

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Verifique si el programa de la figura 6 corresponde al grafo de la figura 5.

Edite el programa de la figura 6, simúlelo y una vez comprobado bájelo al autómeta.

Arme el circuito de potencia, debe ser similar al de la figura 7.

Compruebe el funcionamiento del arranque.

**MATERIAL:**

- Red trifásica de CA
- Disyuntor electromagnético
- Contactor termomagnético
- Pulsadores de tablero eléctrico
- Señalizadores de tablero eléctrico
- Cables Nº 10
- Micro autómeta programable de salida de relé (lenguaje escalera)
- Micro autómeta programable de salida de relé (lenguaje de bloques lógicos)

Herramientas: Multímetro, desarmadores y alicates para cables

CUESTIONARIO:

1. Dibuje los esquemas eléctricos de los circuitos como realmente fueron realizados en la práctica.
2. Transforme el grafo de la figura 5 a su equivalente en grafnet.
3. Realice el programa de la figura 6 en esquema de funciones o bloques.
4. Comente las diferencias en los controles de arranque estrella triángulo de las figuras 6, 8 y 9.
5. Compruebe si es posible obtener un retardo de activación memorizado combinando, un retardo de activación con un S-R.

## PRÁCTICA 5

# Funciones de programación especial, sensores de proximidad tres hilos

**OBJETIVO:** Conexiones de sensores de proximidad tres hilos, salidas de tipo transistor de micro autómatas programables y función Messenger SMS.

**CAPACIDADES:** Conecta a micro autómatas programables sensores de tres hilos, reconociendo tipo de conexión NPN o PNP.

Conecta aplicaciones a micro autómatas programables con salidas de transistor, reconociendo tipo de conexión NPN o PNP.

Hace comandos de control con teléfono celular del tipo todo o nada con la función SMS Messenger.

**FUNDAMENTO TEÓRICO:**

Características del autómata “Messenger” Función que permite enviar y recibir mensajes SMS entre el autómata y un teléfono celular

**x-Messenger (SMS Micro PLC) Model Selection chart (excluding accessories)**

Model	EXM-8AC-R	EXM-12DC-D-R	EXM-12DC-D-A-R	EXM-12DC-DAI-R	EXM-12DC-D-TN	EXM-12DC-DA-TN	EXM-12DC-DAI-TN
Supply Voltage	110~240VAC	12 ~ 24 VDC					
Inputs	6 digital	8 digital	4 digital/analog+4 digital	2digital/analog+2analog +4 digital	8 digital	4 digital/analog+4 digital	
Analog Input signal	No	No	4 DC (0..10V)	2 (DC 0...10V) + 2 (0...20mA)	No	4 DC (0..10V)	2 (DC 0...10V) + 2 (0...20mA)
Outputs	2 relay(10A)		4 relay(10A)		4 transistor(0.3A)		
PWM	No				2 channel (Max.333Hz)		
High Speed Count(I7,I8)	No				(I7,I8)60kHz		
SMS	Yes (64 different short message configuration)						
Voice alarming	No						

Fig.1: Modelo x-Messenger a usar en laboratorio

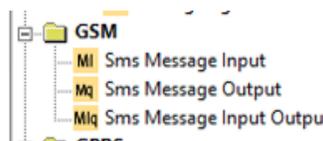


Fig. 2: Bloques SMS del x-Messenger

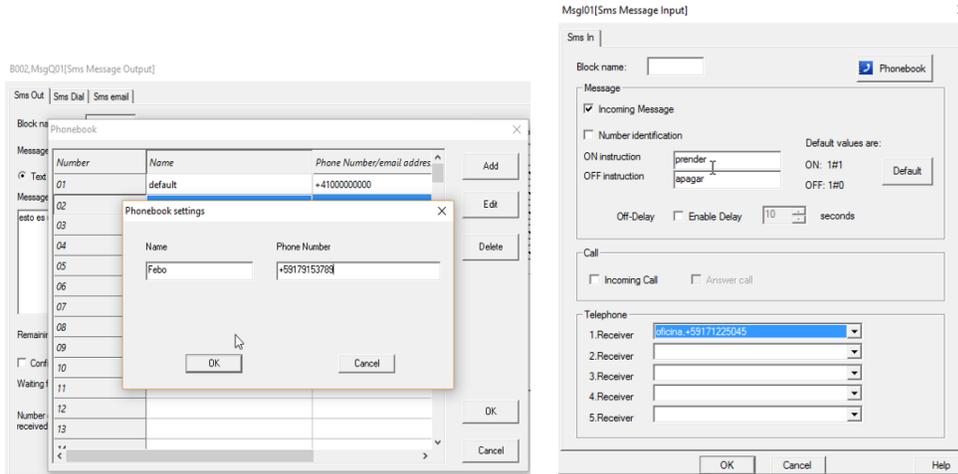


Fig. 3: Parametrización del x-Messenger

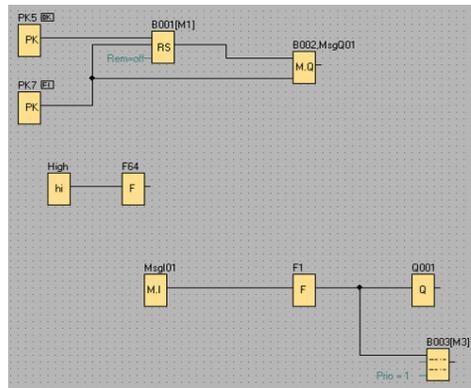


Fig. 4: Ejemplo de envío y recepción de mensajes con Messenger

Uso de sensores de tres hilos:

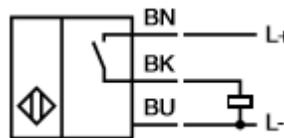


Fig. 5: Sensor de proximidad tres hilos PNP

Los PLC vienen con dos tipos de entradas, Entradas PNP (Sourcing) o Entradas NPN (Sinking). Si se conecta un sensor a una entrada de otro tipo éste no va a funcionar. Analizar el código de colores que trae el sensor. Por lo general éstos vienen con tres hilos, el de color café es (+), el azul (-) y el color negro es el cable de señal, que es el que va en la entrada del PLC.

Como se observa en la siguiente gráfica, mientras que el tipo de salida del sensor sea compatible con el tipo de entrada del PLC se conectan igual. Sin embargo, si el sensor es PNP el PLC debe ser del mismo tipo PNP o, si el sensor es NPN, el PLC debe tener entrada tipo NPN.

Antes de hacer la conexión debe saberse el tipo de entradas del PLC, NPN o PNP, así se podrá determinar si una de las terminales del botón va a la terminal positiva o negativa y la otra a la entrada del PLC.

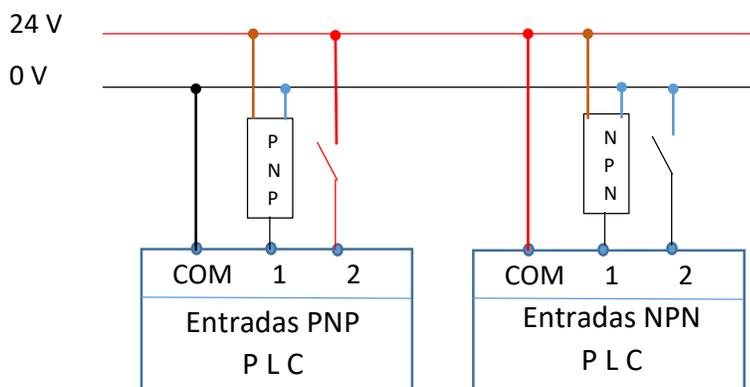


Fig. 6: Conexión de sensores tres hilos a entradas de autómatas programables

Probando los sensores de proximidad de tres hilos:

Verificación de si es PNP o NPN

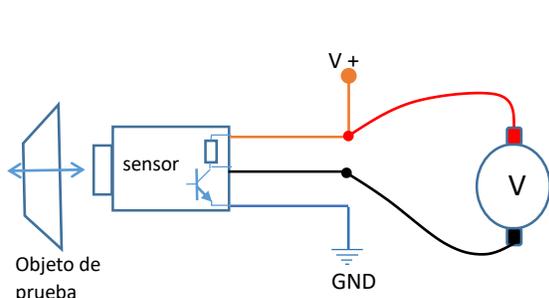


Fig. 7: Sensor de proximidad de tres hilos NPN

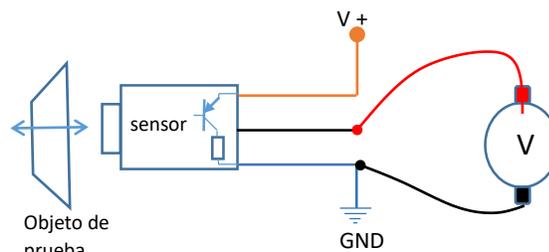


Fig. 8: Sensor de proximidad de tres hilos PNP

Sensor de proximidad tres hilos tipo NPN: Si el sensor de tres hilos es del tipo NPN y es normalmente abierto “NO”, el voltímetro dará lectura si se aproxima un objeto de prueba que detecte el sensor (como se muestra en la figura), en cambio, si el sensor es NPN normalmente cerrado “NC”, el voltímetro dará lectura al alejarse el objeto de prueba del área de detección del sensor.

Sensor de proximidad tres hilos tipo PNP: Si el sensor de tres hilos es del tipo PNP y es normalmente abierto “NO”, el voltímetro dará lectura si se aproxima un objeto de prueba que detecte el sensor (como se muestra en la figura 8), en cambio, si el sensor es PNP normalmente cerrado “NC”, el voltímetro dará lectura al alejarse el objeto de prueba de la zona de detección del sensor.

Salidas tipo transistor de autómatas programables. La principal ventaja de salidas de transistor de autómatas programables, es que éstas pueden operar a elevadas frecuencias de conmutación, sin embargo solo pueden trabajar con voltaje DC, en el modelo de laboratorio “ELC-18DC-DA-TN-U(PNP)” voltajes no mayores a 60 V y niveles de corriente que no pueden exceder los 0,3 A.

En la figura 9, se muestra la conexión externa cuando la salida es de tipo PNP.

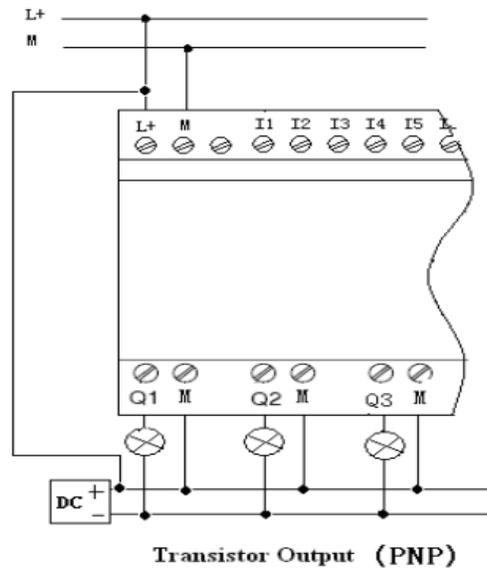


Fig. 9: Conexión a salidas transistor tipo PNP; Fuente EASY xLogic

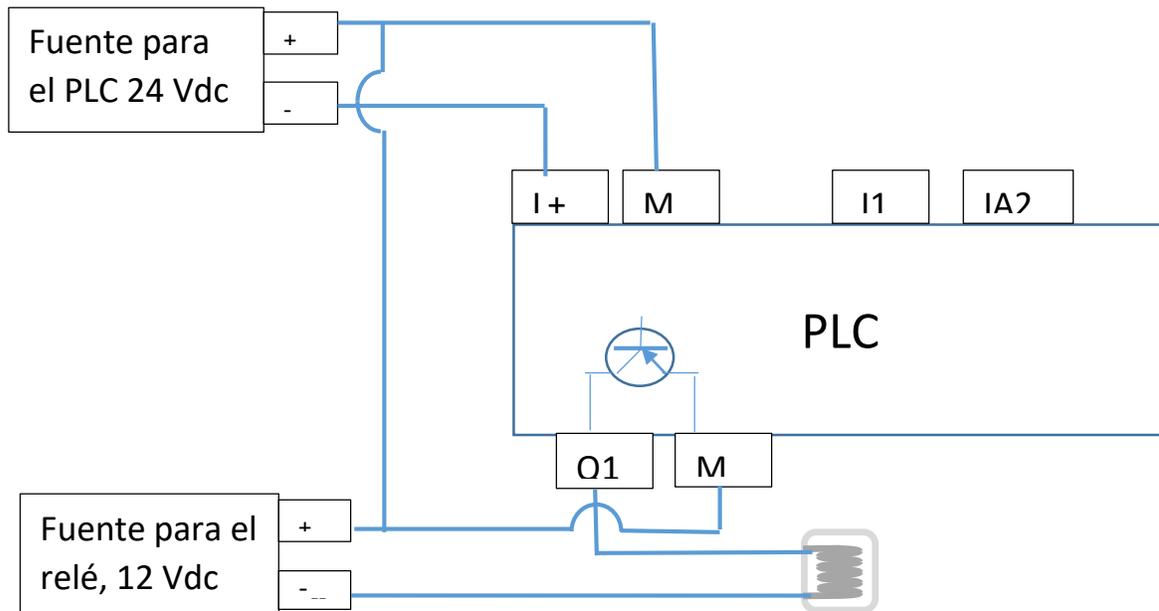


Fig. 10: Aplicación de encendido de un relé de 12 V con salida tipo transistor PNP

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Se realizarán tres actividades en la presente práctica:

1. Envío y recepción de mensajes SMS (autómata – celular) con la función Messenger.
2. Conexión de sensores de tres hilos.
3. Manejo de señales en autómatas con salida de transistor.

Descripción de actividades a realizar:

1. Realizar pruebas de envío y recepción de mensajes SMS entre el autómata x-messenger y un teléfono celular, siguiendo instrucciones de las figuras 2, 3 y 4. Debe verificar que el autómata tenga insertada un chip de alguna empresa telefónica y el mismo cuente con crédito para llamadas. La comunicación puede realizarse desde cualquier lugar a donde llega señal de telefonía celular.
2. Inspeccione y haga pruebas de funcionamiento con sensores de proximidad, ya sea inductivos, capacitivos u ópticos, que sean de tres hilos, éstos son normalmente usados en autómatas con entradas  $V_{DD}$ , mientras que los de dos hilos se emplean en autómatas de  $V_{CA}$ . Verifique si se trata del tipo PNP o NPN haciendo pruebas según se muestra en figuras 7 y 8, luego conecte al autómata como se ve en figura 6.
3. Autómatas con salida de transistor son mayormente empleados para conmutación de elevada frecuencia y voltaje de corriente continua y debe conocerse si éstos son del tipo PNP o NPN. En figuras 9 y 10 se muestran las conexiones a realizar, mismas que requieren dos fuentes de alimentación, una para la alimentación y entradas del autómata y otra para las conexiones de salida.

**MATERIAL:**

- Micro autómata programable de salida de relé (lenguaje escalera)
- Micro autómata programable de salida de relé (lenguaje de bloques lógicos), con función Messenger
- Sensores de proximidad inductivos, capacitivos y ópticos de tres hilos
- Herramientas: Multímetro, desarmadores y alicates y cables

**CUESTIONARIO:**

1. Comente aplicaciones que podría hacer con el autómata x-messenger.
2. Dibuje programas realizados en la práctica usando la función messenger.
3. Indique cuándo usar sensores inductivos, capacitivos u ópticos.
4. Dibuje la conexión que hizo en laboratorio empleando los sensores de tres hilos.
5. Indique cómo funcionan sensores de proximidad con más de tres hilos.
6. Dibuje la conexión que hizo en laboratorio usando el micro PLC con salidas de transistor.

## PRÁCTICA 6

### Control motor paso a paso

**OBJETIVO:** Control básico de motores paso a paso.

**COMPETENCIAS:**

Controla posición de un motor paso a paso monopolar, conectando y programando el mismo a través de un micro PLC.

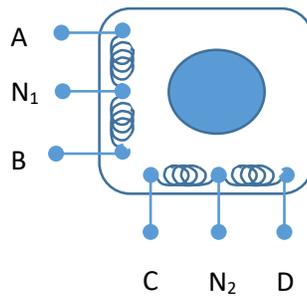


Figura 1: Esquema de un motor paso a paso monopolar con seis hilos

#### FUNDAMENTO TEÓRICO

Secuencias para manejar motores paso a paso (unipolar)

**Paso simple:** Esta secuencia de pasos es la más simple de todas y consiste en activar cada bobina una a una y por separado, con esta secuencia de encendido de bobinas no se obtiene mucha fuerza ya que solo es una bobina cada vez la que arrastra y sujeta el rotor del eje del motor

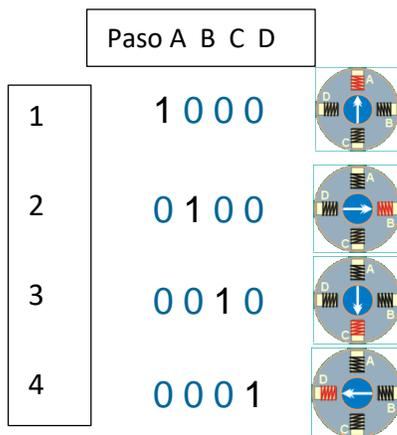


Fig. 2: Paso simple

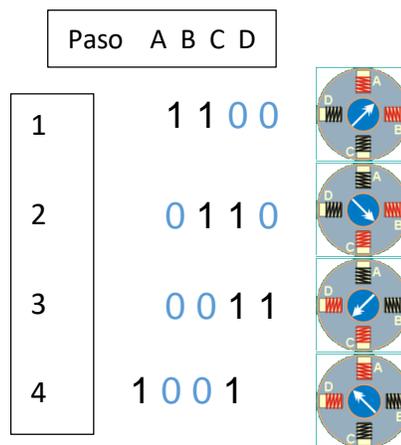


Fig. 3: Paso doble

**Paso doble:** Con el paso doble activamos las bobinas de dos en dos con lo que hacemos un campo magnético más potente que atraerá con más fuerza y retendrá el rotor del motor en el sitio. Los pasos también serán algo más bruscos debidos a que la acción del campo magnético es más poderosa que en la secuencia anterior.

**Medio Paso:**

Combinando los dos tipos de secuencias anteriores podemos hacer moverse al motor en pasos más pequeños y precisos y así pues tenemos el doble de pasos de movimiento para el recorrido total de 360° del motor.

Paso	A	B	C	D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1



Fig.4: Medio paso

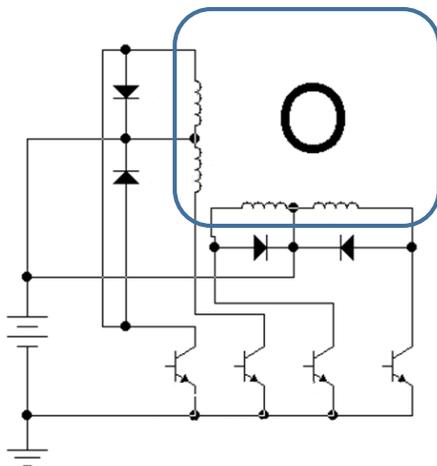


Figura 5: Conexión de motor monopolar

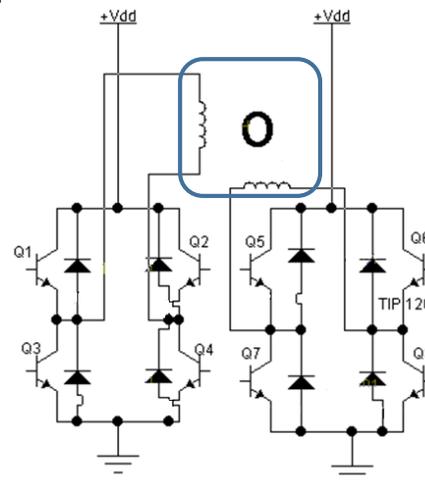


Figura 6: Conexión de motor bipolar

Aplicación de función de contador para limitar el número de pulsos en un motor paso a paso mediante un micro autómatas programable haciendo uso de salidas tipo transistor y tipo relé.

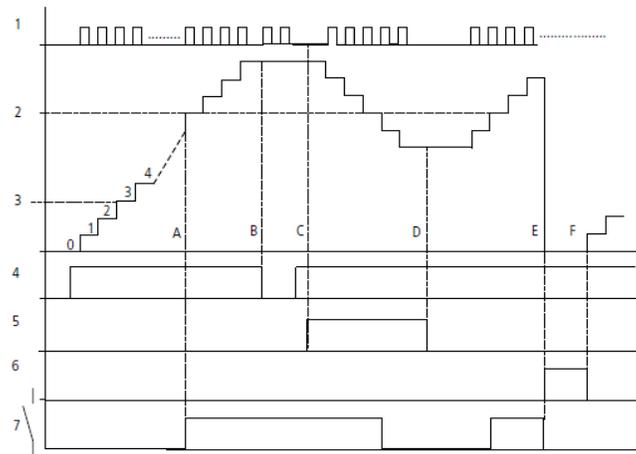


Figura 57: Diagrama de comportamiento Contador rápido  
 1: impulsos de conteo en la entrada de contador I1(I2)  
 2: valor de consigna del contador  
 3: valor real del contador  
 4: autorización del contador, CC13 (CC14)  
 5: dirección de conteo, bobina de dirección DC13 (DC14)  
 6: bobina de desactivación del contador RC13 (RC14)  
 7: contacto del contador, CC13 (CC14)

Figura 7: Diagrama de contador extraído de manual EASY 700 Módulo de Control

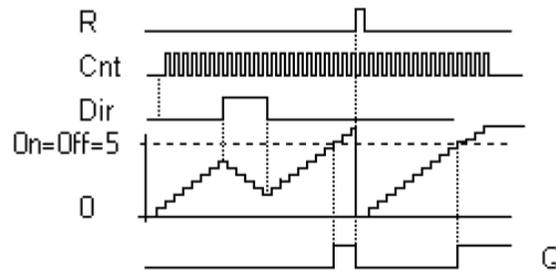


Figura 8: Diagrama de contador extraído de manual EASY xLogic Micro PLC

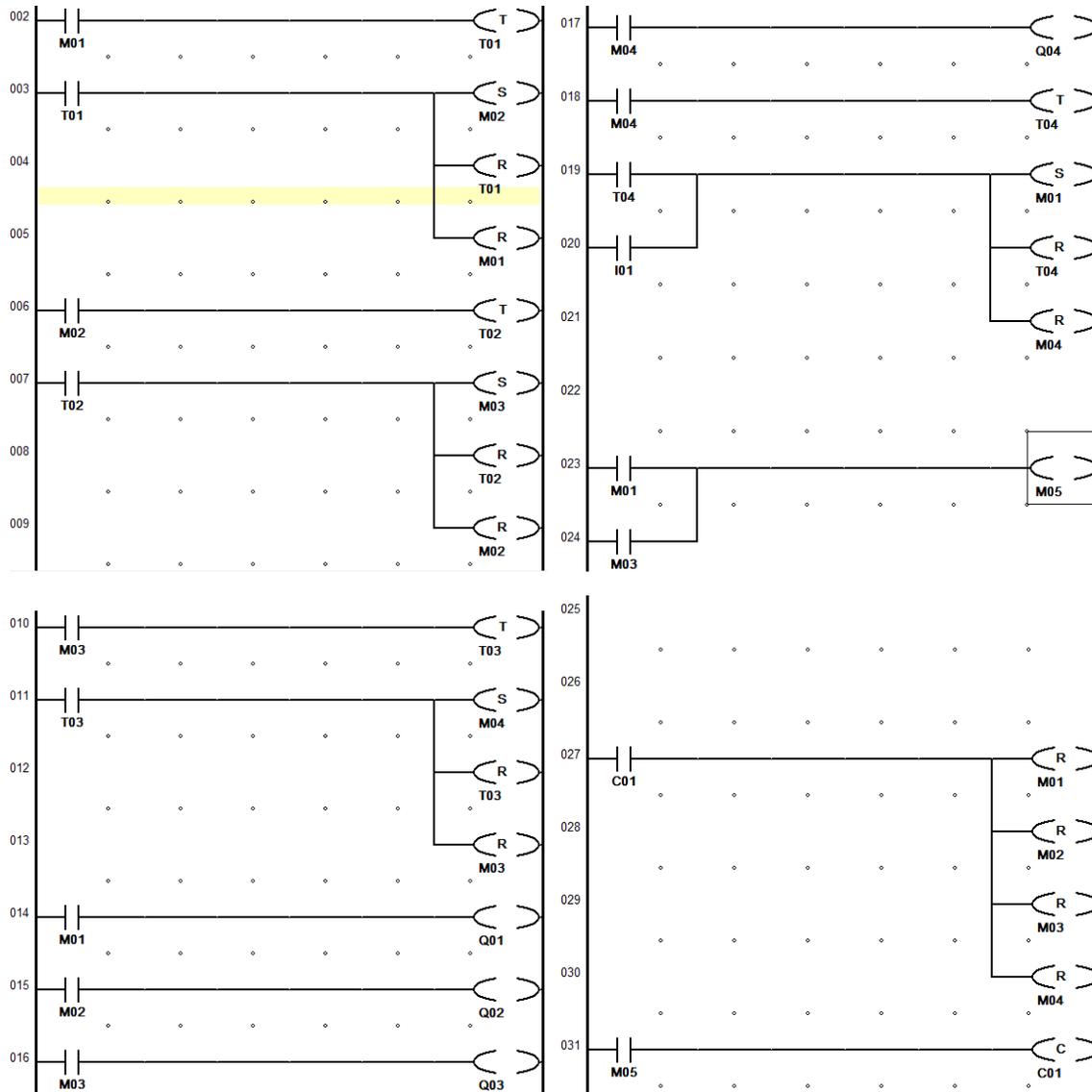


Figura 9: Programa en lenguaje escalera para controlar un motor paso en paso simple A,B,C,D,A,B,...

El programa incluye un contador, mismo que en este caso cuenta las veces que se activan las marcas M01 y M03; una vez llegado al valor de consigna, el contacto auxiliar “resetea” a todas las marcas parando de este modo al motor.

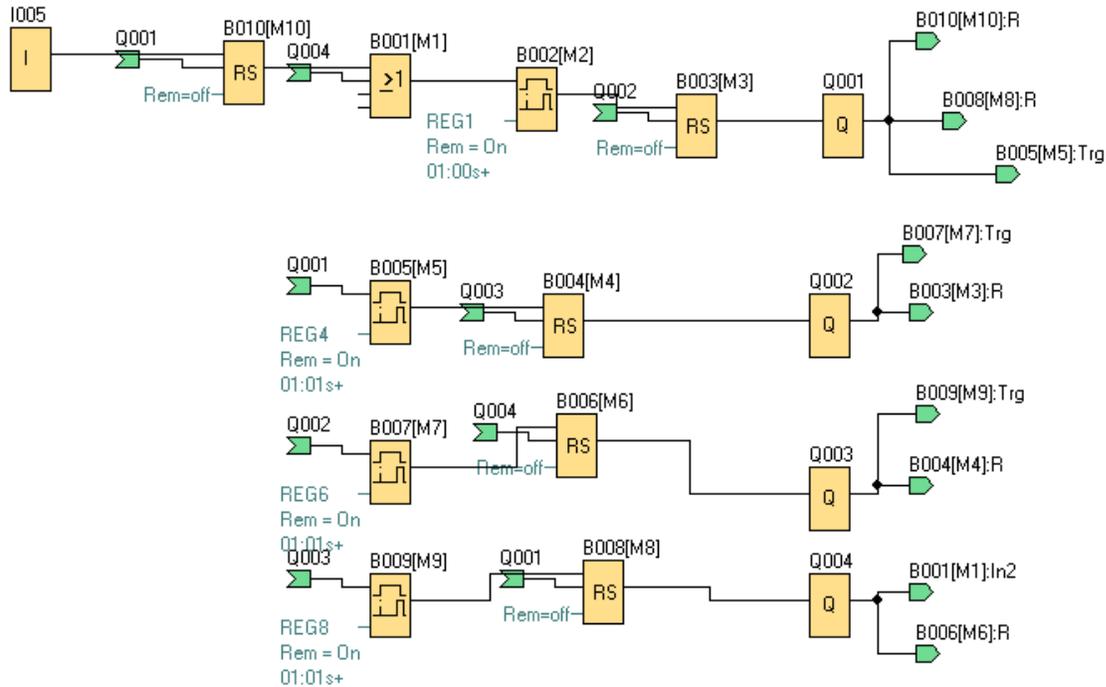


Figura 10: Programa en diagrama de bloques para controlar un motor paso en paso simple A,B,C,D,A,B,...

El programa de arriba es el más simple para accionar un motor paso a paso simple, no incluye pulsador de parada o contador para delimitar el número de pulsos del motor. Es posible simplificar dicho programa si en vez de usar un retardo de activación con un RS, se usara un retardo de activación memorizado.

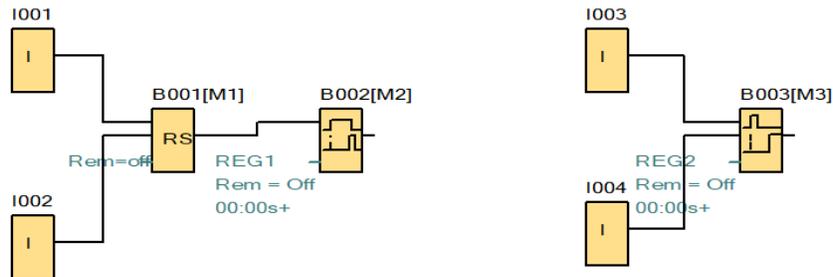


Figura 11: Equivalencia de programación RS + On Delay = Retentive On Delay

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Identifique las terminales del motor paso a paso:

Con un ohmímetro mida la resistencia entre las diferentes terminales del motor, y debe obtener:

- $R_{A-N1} = R_{B-N1} = R_{C-N2} = R_{D-N2} = R$
- $R_{A-B} = R_{C-D} = 2 \cdot R$
- $R_{(A, N1, B) - (C, N2, D)} = \infty$

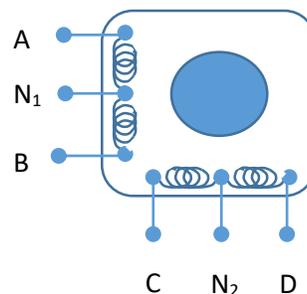


Fig. 12: Esquema motor paso a paso

Una los cables  $N_1$  con  $N_2$  y conéctelos a la fuente positiva de la fuente con el voltaje recomendado por el fabricante del motor, luego conecte el borne negativo o de tierra de la fuente con las terminales A, B, C y D del motor (uno por vez), si sigue correctamente la secuencia, el eje del motor girará paso a paso, la cantidad de grados por paso especificado por el fabricante. Si no estuviera bien la secuencia, dígase A, B, D, C, podrá apreciar que el eje del motor no gira de modo continuo.

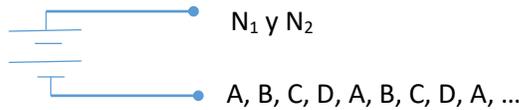


Fig. 13: Alimentación de un motor paso a paso

Verifique el torque que genera el eje del motor agarrándolo suavemente con los dedos. Si incrementa el valor de voltaje de la fuente verificará que el torque se hace mayor, por otra parte si cambia la secuencia a A, D, C, B, A, D, C, B, A, ..., el sentido de rotación se invierte.

Mida la cantidad de grados que gira el motor por cada pulso o paso.

Realice uno de los programas mostrados en figuras 9 y 10 según qué tipo de programa dispone el autómata a utilizar y simúlelo.

Haga la conexión al autómata, la figura 14 muestra la conexión al autómata de salida de transistores y haga la prueba de funcionamiento. Los diodos se instalan para disipar corrientes inducidas en las desconexiones.

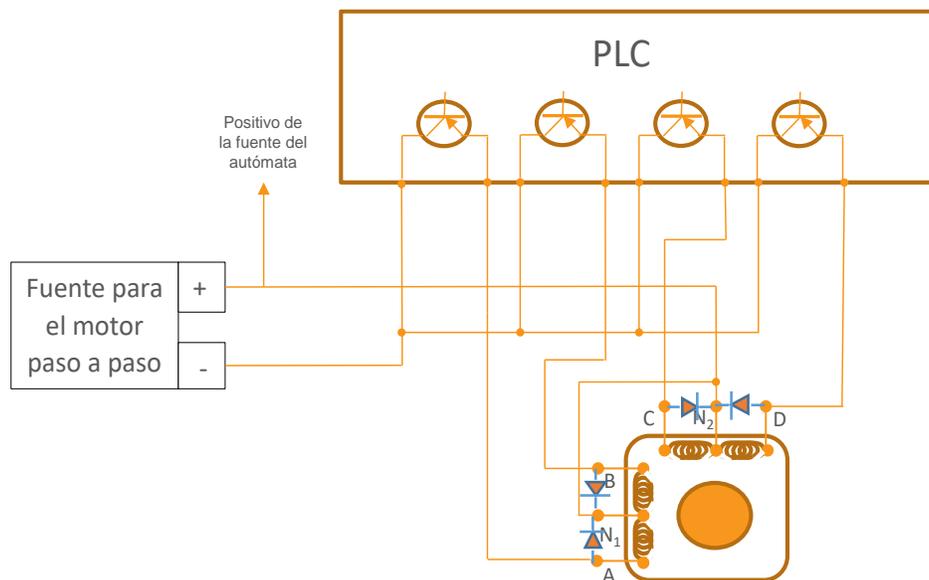


Figura 14: Conexión del motor paso a paso monopolar a un autómata con salida de transistores

Haga la conexión para autómatas con salida de relé y haga la prueba de funcionamiento.

Haga variaciones respecto al programa básico de paso simple.

Haga aplicación de funciones de conteo de los autómatas para que el motor ejecute una determinada cantidad de pulsos o pasos y pare.

**MATERIAL:**

- Micro autómata programable de salida de relé.
- Micro autómata programable de salida de relé.
- Micro autómata programable de salida de transistor.
- Motores paso a paso monopolares.
- Fuentes de alimentación
- Cables de conexión

Herramientas: Multímetro, desarmadores y alicates para cables

**CUESTIONARIO:**

1. Programe la secuencia “doble paso”,.... ¿qué beneficio obtiene de esta secuencia?
2. Averigüe la secuencia de “medio paso” y prográmela. ¿qué beneficio obtiene de esta secuencia?
3. Dibuje el programa en esquema de funciones para que el motor ejecute cierta cantidad de pulsos y pare.
4. Dibuje la conexión de un autómata con salidas de relé a un motor paso a paso monopolar.
5. Indique las modificaciones que debe realizar tanto en conexiones como en programación para operar un motor paso a paso bipolar.
6. Existen en el mercado variedad de “drivers” que simplifican el manejo de los motores paso a paso, investigue sobre ellos e indique específicamente que beneficios se obtienen de su uso.

## PRÁCTICA 7

### Inversión de giro de un motor CC con autómatas programables

**OBJETIVO:** Inversión de giro de un motor de corriente continua.

**COMPETENCIAS:**

Aplica el circuito puente H para inversión en el sentido de giro de un motor de corriente continua.

**FUNDAMENTO TEÓRICO**

Inversión de sentido de rotación de un motor de corriente continua

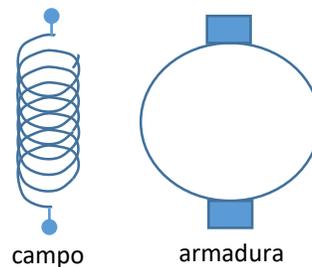


Figura 1: Diagrama esquemático de un motor de CC

El motor de corriente continua con escobillas está normalmente compuesto de dos partes: el campo una bobina o un imán en motores pequeños en la carcasa y una bobina en el rotor llamada armadura y conectada al exterior mediante escobillas. Existen distintas configuraciones de conexión de ambos bobinados, pero como en laboratorio se trabajará con un pequeño motor, en el que el campo es producido por un imán, de aquí en adelante al motor se lo representará solamente con la armadura.

Dependiendo de la polaridad de conexión a la armadura se obtendrá el sentido de rotación, por ejemplo:

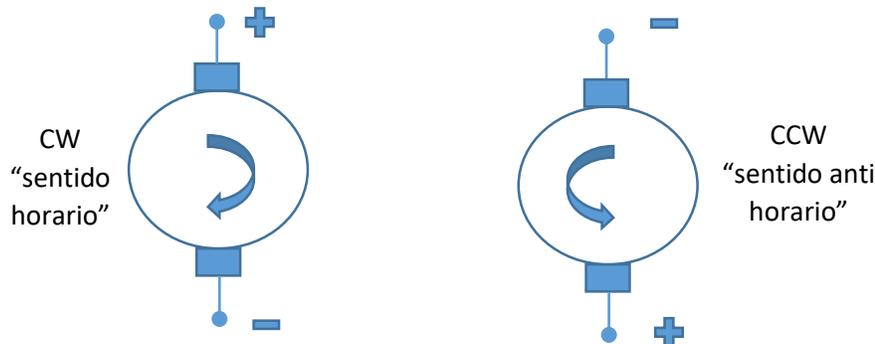


Fig. 2: Polarización de la armadura de un motor de CC

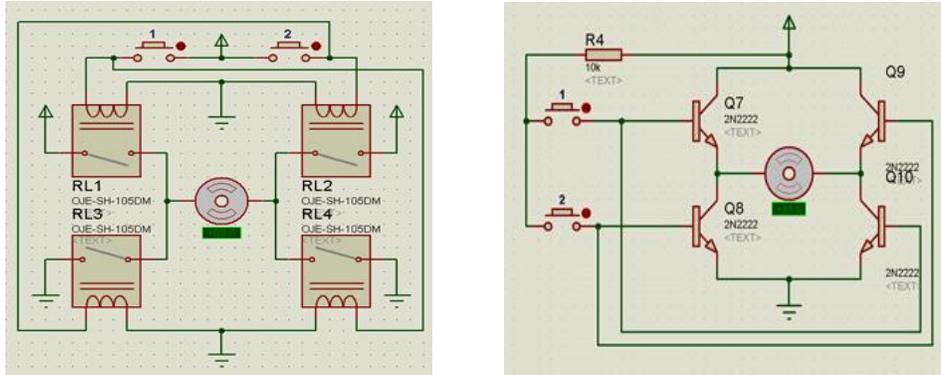


Fig. 3: Diagrama de puente H para inversión de giro de un motor paso a paso

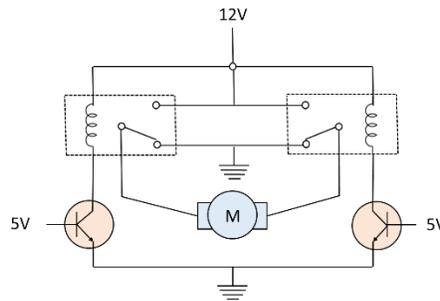


Fig. 4: Puente H empleando una combinación de transistor con relé

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Haga funcionar el motor de corriente continua aplicando entre sus terminales de armadura un voltaje según la especificación del fabricante, incremente o disminuya ligeramente el valor del mismo y compruebe con un tacómetro la velocidad de su eje.

Ejercite manualmente la inversión de giro cambiando la polaridad de los cables de alimentación al motor.

Si se dispone de amperímetro, medir la corriente en el motor y apreciar la dependencia de ésta en función de carga que podría ejercitar suavemente con los dedos de la mano sobre el eje del motor.

De acuerdo a disponibilidad de autómatas, escoger alguno de los circuitos puente H de la figura 3 o figura 4, note que con los circuitos de la figura 3 requiere emplear cuatro salidas del autómata, mientras que para el circuito de la figura 4 solo dos.

Los programas a realizar serán similares a los empleados en la práctica 2 de la presente guía, luego de verificar la simulación bajo el programa al autómata y opere el motor con inversión de giro.

**MATERIAL:**

- Micro autómata programable de salida de relé.

- Micro autómatas programables de salida de relé.
- Micro autómatas programables de salida de transistor.
- Motores de corriente continua.
- Tacómetro.
- Fuentes de alimentación.
- Cables de conexión

#### CUESTIONARIO:

1. Realice un gráfico que relacione voltaje de alimentación vs velocidad de rotación del eje del motor.
2. ¿Cómo es la relación entre carga o freno aplicado al eje del motor e intensidad de corriente?
3. Dibuje la conexión eléctrica a realizar para control de inversión de giro para autómatas con salida de relé y con salida de transistor correspondientes a la figura 3.
4. Dibuje la conexión eléctrica a realizar para control de inversión de giro para autómatas con salida de transistor y relé correspondiente a la figura 4.

## PRÁCTICA 8

### Aplicaciones con señales analógicas

**OBJETIVO:** Manejo de sensores y módulos de entrada/salida de tipo analógicos.

**COMPETENCIA:** Controla señales analógicas, conectando sensores analógicos, haciendo uso de tratamiento de señal y módulos de entrada/salida analógicos de autómatas programables y HMI (interfaz hombre máquina).

**FUNDAMENTO TEÓRICO:**

#### PT 100

Un Pt100 es un sensor de temperatura, consiste en un alambre de platino que a 0 °C tiene 100  $\Omega$  y que al aumentar la temperatura aumenta su resistencia eléctrica.

El incremento de la resistencia es aproximadamente lineal y creciente, característico del platino de tal forma que mediante tablas es posible encontrar la temperatura exacta a la que corresponde.

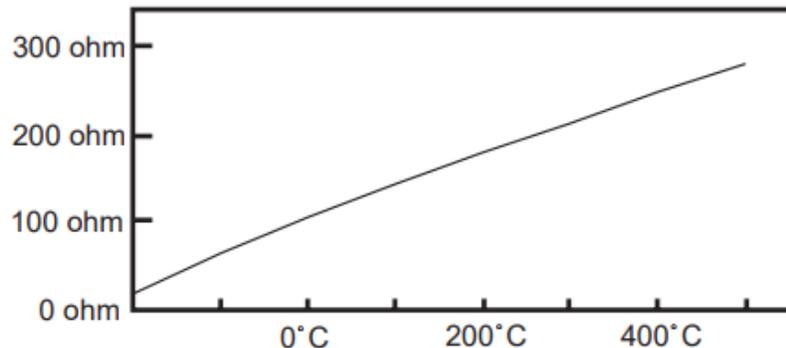


Fig. 1: Un Pt100 es un tipo particular de RTD (Dispositivo Termo Resistivo)

Normalmente las Pt100 industriales se consiguen encapsuladas en la misma forma que las termocuplas, es decir dentro de un tubo de acero inoxidable u otro material (vaina), en un extremo está el elemento sensible (alambre de platino) y en el otro está el terminal eléctrico de los cables protegido dentro de una caja redonda de aluminio (cabezal). Por otra parte los Pt100 siendo levemente más costosos y mecánicamente no tan rígidos como las termocuplas, las superan especialmente en aplicaciones de bajas temperaturas. (-100 a 200 °). Una ventaja del sensor PT100 es que al contrario que otros sensores que se degradan con el tiempo y dan lecturas erróneas, el PT100 abre el circuito y se puede saber cuándo es necesario cambiarlo.

Otro beneficio de usar una termo-resistencia (PT100) es que la relación resistencia temperatura es prácticamente lineal, como se muestra en figuras 1 y 2.

Pt 100 ohms	°C	0	1	2	3	4	5	6	7	8	9
-50		80.25	79.85	79.45	79.06	78.66	78.26	77.86	77.46	77.06	76.66
-40		84.22	83.83	83.43	83.03	82.64	82.24	81.84	81.44	81.05	80.65
-30		88.18	87.79	87.39	87.00	86.60	86.21	85.81	85.41	85.02	84.62
-20		92.13	91.74	91.35	90.95	90.56	90.16	89.77	89.37	88.98	88.58
-10		96.07	95.68	95.29	94.89	94.50	94.11	93.71	93.32	92.92	92.53
0		100.00	99.61	99.22	98.82	98.43	98.04	97.65	97.25	96.86	96.47
0		100.00	100.39	100.78	101.17	101.56	101.95	102.34	102.73	103.12	103.51
10		103.90	104.29	104.68	105.07	105.46	105.85	106.24	106.63	107.02	107.41
20		107.79	108.18	108.57	108.96	109.35	109.74	110.12	110.51	110.90	111.29
30		111.67	112.06	112.45	112.84	113.22	113.61	114.00	114.38	114.77	115.16
40		115.54	115.93	116.32	116.70	117.09	117.47	117.86	118.24	118.63	119.01
50		119.40	119.78	120.17	120.55	120.94	121.32	121.71	122.09	122.48	122.86
60		123.24	123.63	124.01	124.39	124.78	125.16	125.54	125.93	126.31	126.69
70		127.07	127.46	127.84	128.22	128.60	128.99	129.37	129.75	130.13	130.51
80		130.89	131.28	131.66	132.04	132.42	132.80	133.18	133.56	133.94	134.32
90		134.70	135.08	135.46	135.84	136.22	136.60	136.98	137.36	137.74	138.12
100		138.50	138.88	139.26	139.64	140.02	140.40	140.77	141.15	141.53	141.91

Fig. 2: Tabla de resistencia de un PT 100

**Sensor de temperatura, easy-DA, easy-DC**

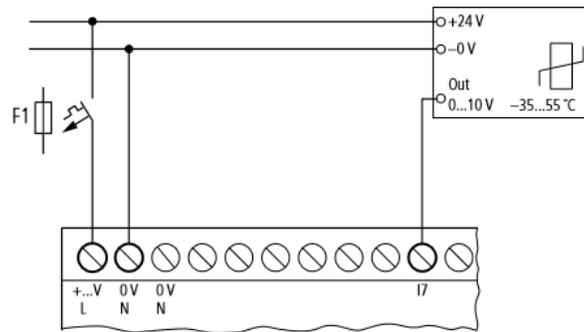


Figura 3: Conexión con transmisor de temperatura, Fuente EASY Moller

Potenciómetro

El potenciómetro es una resistencia variable regulable con una manija o eje movimiento en arco de circunferencia, se constituye prácticamente en un sensor de desplazamiento angular, cuyo valor de resistencia se define por el ángulo desplazado por la manija.

Este sensor tiene amplia utilización en la ingeniería mecánica ya que permite censar posicionamiento, por ejemplo cuánto presiona un conductor el pedal acelerador.

### Potenciómetro de señal analógico easy-DC

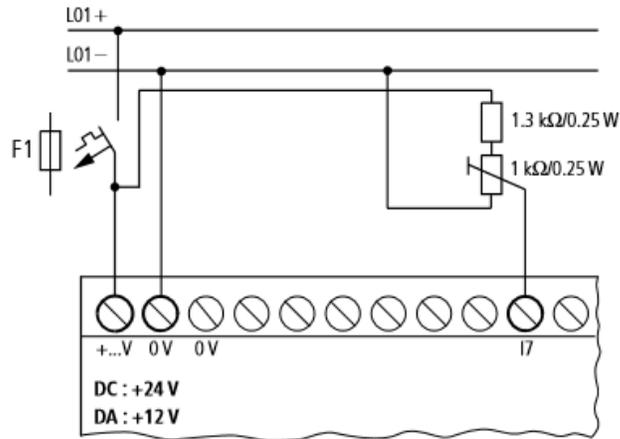


Figura 4: Potenciómetro de señal analógica con alimentación de 24 VDC, Fuente EASY Moller

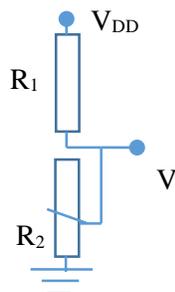
La configuración de la conexión del reóstato mostrada en la figura 4, corresponde al circuito divisor de voltaje también mostrado en figura 5, con la ecuación que describe cómo obtener el voltaje de salida. Aplicando esta ecuación a los datos de la figura 4 se tiene:

Mínimo voltaje: 
$$V = 24 \cdot \left( \frac{0}{0+1300} \right) = 0 V$$

Máximo voltaje: 
$$V = 24 \cdot \left( \frac{1000}{1000+1300} \right) = 10,43 V$$

Es decir que I7 de la figura 4 recibirá entre 0 y 10 V (redondeado), rango característico de señales de entrada de autómatas en el rango de voltajes.

Divisor de voltaje:



$$V = V_{DD} \left( \frac{R_2}{R_1 + R_2} \right)$$

Figura 5: Divisor de voltaje

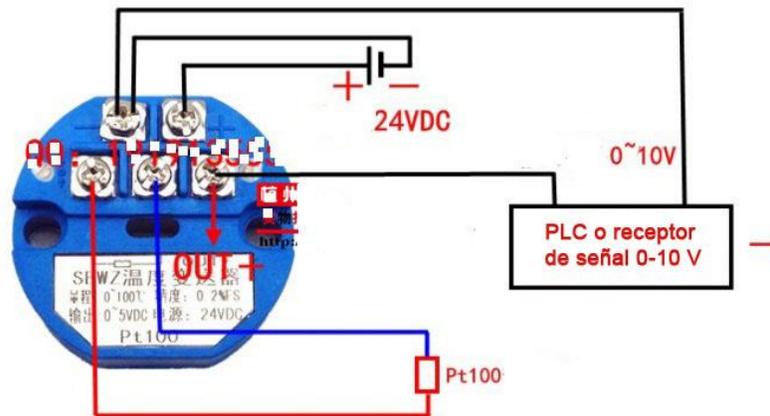


Figura 6: Transmisor de temperatura a utilizarse en clases

### CELDA O CÉLULAS DE CARGA

Una célula de carga (o celda de carga) es un transductor que convierte la fuerza aplicada sobre ella en una señal eléctrica medible. A pesar de existir varios tipos de sensores, las células de carga son el sensor de fuerza más común del mercado.

Los diseños de células de carga se pueden distinguir de acuerdo con el tipo de señal de salida generada (neumático, hidráulico, eléctrico) o de acuerdo con la forma en que detectan el peso (flexión, cizalladura, compresión, tensión, etc).

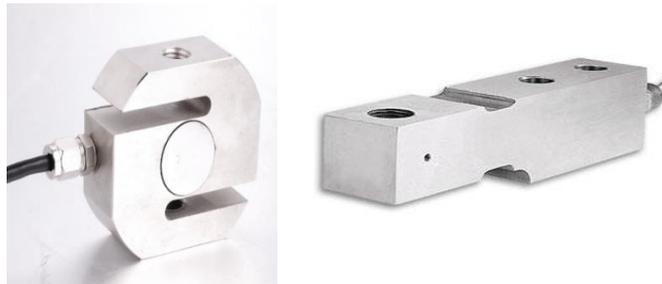


Fig. 7: Celdas o células de carga en forma de S y forma de viga (fuente [www.omega.com](http://www.omega.com))

La celda de carga es uno de los elementos más importantes de una báscula electrónica, ya que se encarga de traducir la fuerza en una señal de voltaje (celda de carga analógica) o en un valor digital (celda de carga digital). La celda de carga analógica con galgas extensométricas es la que se utiliza más comúnmente y es la que se empleará en la práctica.

Las celdas de carga consisten en un metal que sufre una deformación conforme se le aplica una fuerza. Este metal se calcula para soportar un rango de fuerza (que va desde cero fuerza hasta la capacidad máxima) ya sea a tensión, compresión o ambos. La deformación se realiza en la "parte elástica", esto es lo que limita la capacidad de una celda de carga. Al momento de

sobrepasar la parte elástica del metal, sufre una deformación permanente, así como un resorte que se estira de más y ya no regresa a su punto inicial (cuando detecta cero fuerza).

Al metal, se le adhieren galgas extensométricas. Las galgas extensométricas consisten en un metal que al flexionarse varía su resistencia. Las galgas se conectan en un arreglo de puente de Wheatstone, de tal forma que al alimentarse con un voltaje entregan una señal de voltaje proporcional a la fuerza aplicada. La señal de voltaje entregada es en el orden de milivolts. Este voltaje se representa comúnmente proporcional al voltaje de alimentación y a máxima carga (capacidad de la celda), por ejemplo 2mV/V nominal. Quiere decir que si se alimentan con 10Vdc la señal que se va a tener a la capacidad máxima es de 20mV. En el caso ideal, la señal es lineal, esto es, para el ejemplo anterior si se aplica el 50% de la capacidad se tendrían 10mV y si se aplica cero fuerza se obtendrían 0mV.

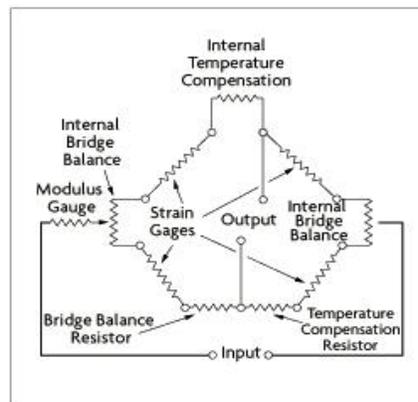


Fig.8: Puente de Wheatstone c/compensación de temperatura

### Entradas analógicas de un autómata programable

Los módulos de entrada y salida analógica de los autómatas programables operan tanto en señales de voltaje como de corriente eléctrica:

- El rango de diferencia de voltaje es de 0 a 10 V.
- El rango de intensidad de corriente es de 4 a 20 mA.

Las entradas analógicas de EASY 700 son I7, I8, I11 e I12; son posibles las siguientes comparaciones

Valor en la entrada de valor del relé de función I1	Funciones de comparación		Valor en la entrada de valor del relé de función I2
		Selección del modo operativo en el relé de función	
Entrada analógica I7, I8, I11, I12			Entrada analógica I7, I8, I11, I12
Valor de consigna 0000 a 9999			Valor de consigna 0000 a 9999
Valor real del relé contador C1 a C16			Valor real del relé contador C1 a C16
Valor real del relé contador T1 a T16	menor que	LT	
	menor o igual que	LE	
	igual	EQ	
	mayor o igual que	GE	
	mayor que	GT	

Figura 9: Funciones de comparación de EASY Moller

Relé de función A1 entrada de valor I1		Relé de función A1 entrada de valor I2
17	GE (mayor o igual)	I8
17	LE (menor o igual)	I8
17	GE (mayor o igual)	Valor de consigna
17	LE (menor o igual)	Valor de consigna
18	GE (mayor o igual)	Valor de consigna
18	LE (menor o igual)	Valor de consigna

Fig. 10: Ejemplos de comparación, Fuente EASY-Moller

Resolución de entradas analógicas I7, I8, I11 e I12 en EASY Moller 700; poseen la siguiente resolución: La señal analógica de 0 a 10 Vdc se transforma en un valor digital de 10 bits de 0 a 1023. Un valor de 100 digital significa un valor de 1,0 V analógico (exactamente 0,98 V).

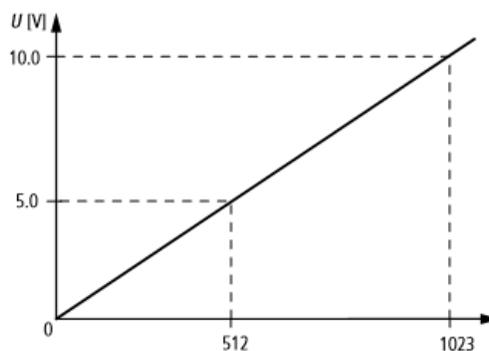


Fig. 11: Resolución de entradas analógicas EASY Moller 700

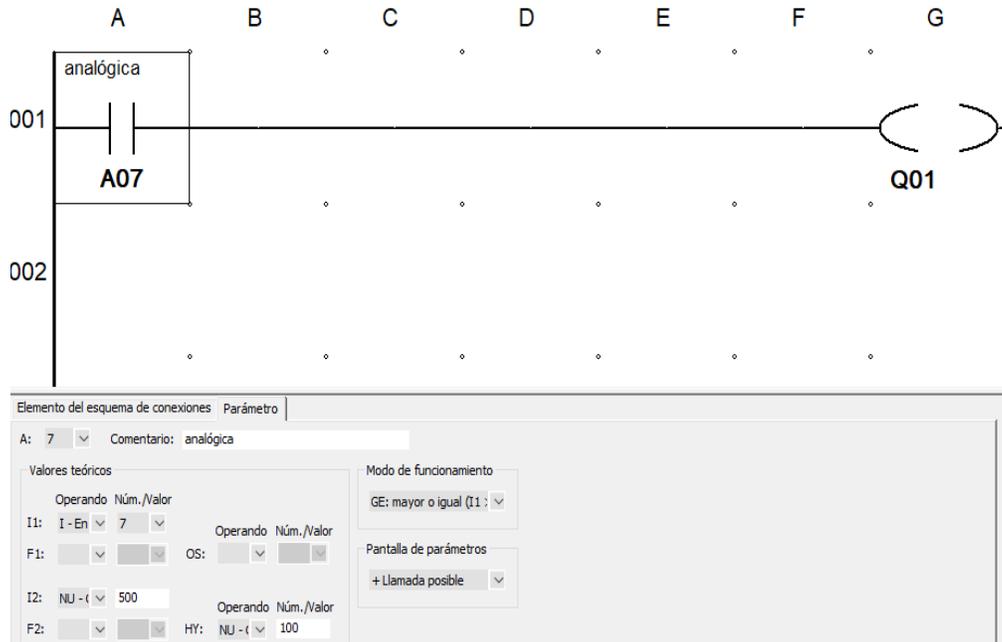


Fig. 12: Ejemplo de programación en EASY-Moller

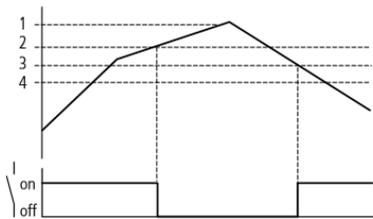


Diagrama de comportamiento Comparador de valores analógicos en el modo operativo "menor que"

- 1: valor real en I7
- 2: valor de consigna más valor de la histéresis
- 3: valor de consigna
- 4: valor de consigna menos histéresis

Fig. 13: Modo "menor que"

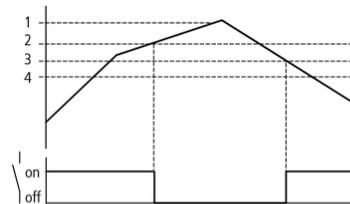


Diagrama de comportamiento Comparador de valores analógicos en el modo operativo "menor o igual que"

- 1: valor real en I7
- 2: valor de consigna más valor de la histéresis
- 3: valor de consigna
- 4: valor de consigna menos histéresis

Fig. 14: Modo "menor o igual que"

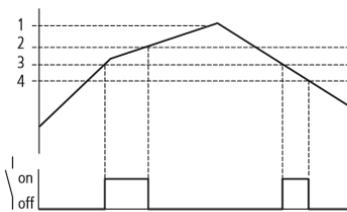


Diagrama de comportamiento Comparador de valores analógicos en el modo operativo "igual que"

- 1: valor real en I8, multiplicado con el factor de multiplicación F2
- 2: valor de consigna más valor de la histéresis
- 3: valor de consigna
- 4: valor de consigna menos histéresis

Fig. 15: Modo "igual que"

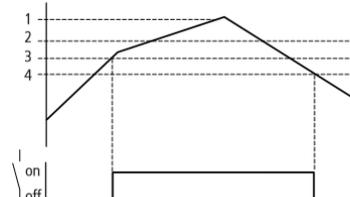


Diagrama de comportamiento Comparador de valores analógicos en el modo operativo "mayor o igual que"

- 1: valor real en I7
- 2: valor de consigna más valor de la histéresis
- 3: valor de consigna
- 4: valor de consigna menos histéresis

Fig. 16: Modo "mayor o igual que"

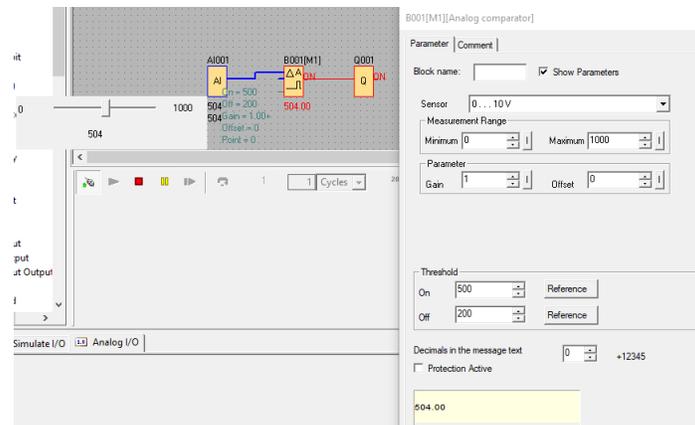


Fig. 17: Simulación de entradas analógicas con diagramas de bloque

### Salidas analógicas de un autómata programable

Así como en las entradas de los autómatas, las salidas normalmente están disponibles en voltaje de 0 a 10 V y, en intensidad de corriente de 4 a 20 mA. Para la práctica, está previsto el uso de un módulo de expansión de EASY, específicamente el modelo ELC-E-AQ (V) que permite tener dos salidas 0 a 10 VDC.

El módulo de expansión debe conectarse al microautómata mediante el Puerto de expansión RS485 y alimentarse independientemente con 24 VDC y seleccionar en su programación (xLogic Soft) la siguiente instrucción mostrada en la figura

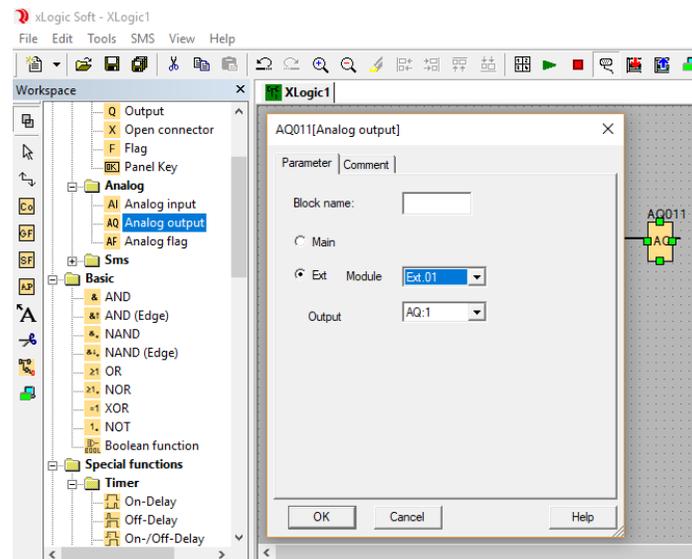


Fig. 18: Selección de módulo externo en xLogic Soft

Las salidas analógicas permiten modular el accionar de los actuadores, por ejemplo conjuntamente a las entradas analógicas, se pueden realizar controles en lazo cerrado PID.

Para observar el efecto de las señales analógicas, en la práctica se empleará como actuador una válvula de bola motorizada (figura 19), es decir que la apertura de la misma está en función del voltaje aplicado.



Fig. 19: Vista de la válvula motorizada para modulación de flujo controlado por voltaje o corriente, fuente Winner Control Ball Valves Co., Ltd.

La válvula debe alimentarse con 24VDC (café + negra -) y conectar su entrada (amarilla) de 0~10 V a la salida analógica del módulo de expansión del microautómata, con tierra común (terminal negativa de alimentación "24 V con alimentación negativa de la válvula "negra" y del módulo de expansión analógica "M" del microautómata).

#### OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:

##### PT 100

Apreciar la variación de la resistencia eléctrica del sensor PT 100 cuando éste es sometido a distintas temperaturas y en la configuración de la figura 2.

Conectar el sensor PT 100 al transmisor de temperatura y verificar el voltaje en la salida del transmisor, alternativamente, armar un circuito divisor de voltaje y medir también el voltaje en la salida del circuito divisor.

Verificar funcionamiento con el PT 100 empleando transmisor de temperatura o circuito divisor de voltaje.

Programar en el autómata, ya sea con diagrama de contactos o con diagrama de bloques, la función de entradas analógicas para entradas de voltaje en alguna aplicación de comparación de señales y simular operación, luego bajar el programa al autómata para verificación. No olvidar verificar cuáles son los bornes de señales analógicas en el autómata a utilizar.

##### Celda de carga

Como la celda de carga genera señales de voltaje muy pequeñas, se requiere de un transmisor o interfaz de tratamiento de señal que amplifica el voltaje en un rango de 0 a 10 V. El fabricante entrega información acerca de las conexiones a realizar, en el caso de la celda de carga a emplearse en laboratorio:

Celda de carga marca DAYANG con rango de medida de 0 a 10 kg, los cables de la celda de carga son de los colores: negro, rojo, verde y blanco; mismos que deben conectarse al transmisor de celda de carga de la forma que se muestra en la tabla 1.

Si la celda de carga es de compresión-tracción, aplicar una fuerza que no sea mayor al rango de medida para el cual está diseñada la celda.

Utilizar el mismo programa empleado para la aplicación del PT 100.

CELDA	TRANSMISOR DE CELDA		Voltaje externo
	color cable	Input	
negro	GND	GND	Fuente -
blanco	S- (S+ tracción)	V0	Salida V
verde	S+ (S- tracción)	I0	
rojo	E+	24V	Fuente +

Tabla 1: Conexiones en la celda de carga DAYANG

### Reóstato

Arme el circuito divisor de voltaje tal como se muestra en las figuras 4 y 5; luego de ello verifique que el voltaje de salida fluctúe entre 0 y 10 V.

Conecte dicho circuito a otro borne de entrada analógica del autómatas, para hacer programas de comparación entre dos señales analógicas, dígame por ejemplo entre la correspondiente a la celda de carga y la del reóstato. De ese modo el reóstato estará funcionando como un regulador de valor de consigna o “set point”.

### Modulación de apertura con válvula motorizada

Programe en EASY mediante xLogic (figura 18) el módulo de expansión para salidas analógicas de voltaje de manera que el valor de éste se relacione, dígame por ejemplo con la mitad de voltaje recibido en la entrada analógica del microautómata con un potenciómetro. De ese modo regular la apertura de la válvula de bola mediante un potenciómetro.

### MATERIAL:

- Sensor PT 100
- Transmisor de temperatura
- Celda de carga – tracción compresión marca DAYANG
- Transmisor de carga marca DAYANG

- Reóstato o potenciómetro
- Cables Nº 10
- Fuentes de 24 VDC
- Micro autómatas programables con entrada analógica (lenguaje escalera)
- Micro autómatas programables con entrada analógica (lenguaje de bloques lógicos)
- Módulo de expansión de salidas analógicas
- Válvula motorizada de bola
- Fuente de calor (encendedor, hornilla, etc.)
- Medidor de temperatura
- Herramientas: Multímetro, desarmadores y alicates para cables

#### CUESTIONARIO:

1. Grafique la relación entre los niveles de voltaje generados por los sensores en función de la magnitud física que registran.
2. Dibuje los programas realizados para aplicación con el reóstato, PT 100 y la celda de carga.
3. Dibuje las conexiones tal como se hicieron en laboratorio para las aplicaciones del reóstato, PT 100 y la celda de carga.
4. Dibuje el programa de aplicación realizado para que la apertura de la válvula responda en función de una señal analógica de entrada proveniente de un potenciómetro.
5. Dibuje las conexiones tal como hicieron en laboratorio para el uso de la válvula motorizada controlada por el módulo de expansión del microautómata.

## PRÁCTICA 9

### Comunicación con un HMI

**OBJETIVO:** Introducción al empleo de Human Machine Interface “HMI” en comunicación con un autómatas.

**COMPETENCIA:** Conecta y configura comunicación MODBUS entre un autómatas y un HMI, realizando programas básicos de aplicación de control.

**FUNDAMENTO TEÓRICO:**

El HMI (Human Machine Interfaz) o “Interfaz hombre-máquina), consiste en una pantalla táctil que facilita la interacción entre el PLC y el usuario, por lo que dicha pantalla permite interacción visual con botonera, palancas, lámparas y otros tipos de indicadores virtuales. En la práctica se hará uso del HMI Samkoon SK – 070AS, mismo que opera con el software SK Workshop V5.0.2.

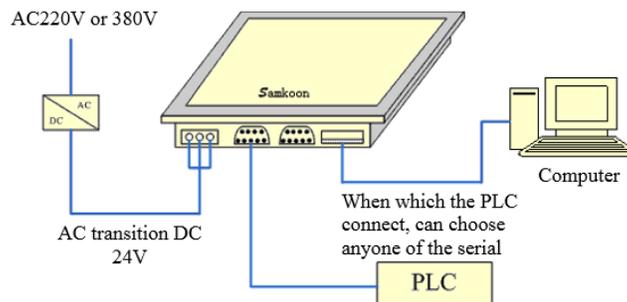


Fig. 1: Diagrama de conexión de las pantallas táctil HMI Samkoon serie SK

Protocolo de comunicación entre xLogic y el HMI

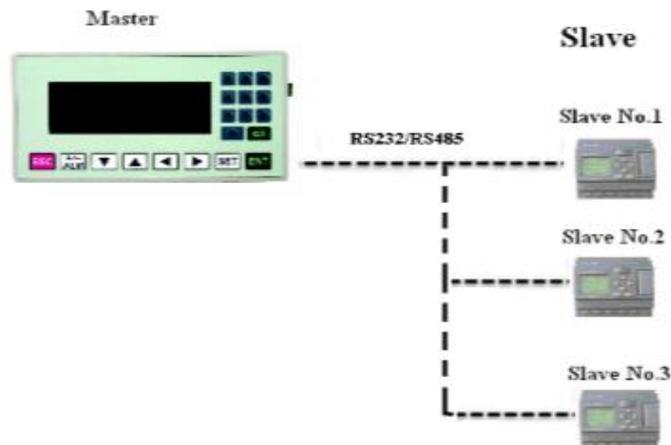


Fig. 2: Protocolo de comunicación entre el HMI y los xLogic

En la actualidad, xLogic se comunica con 9600 [baud rate], es la velocidad de comunicación por defecto, así como el modo de verificación (non parity).

El protocolo de comunicación por defecto de xLogic es MODBUS RTU

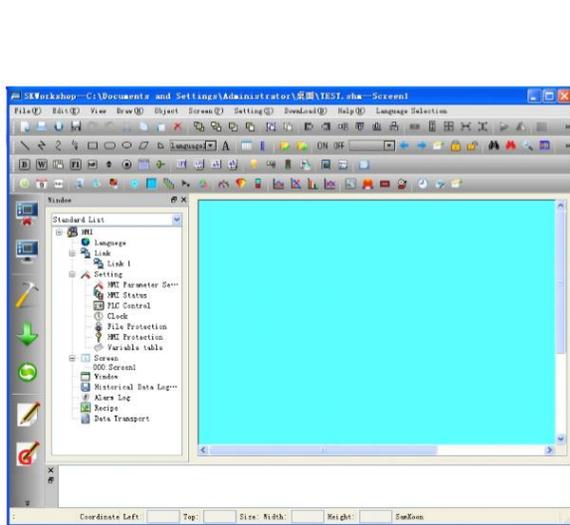


Fig. 3: Página de presentación del software

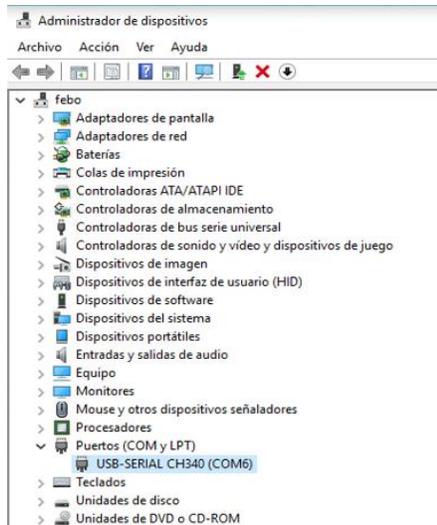
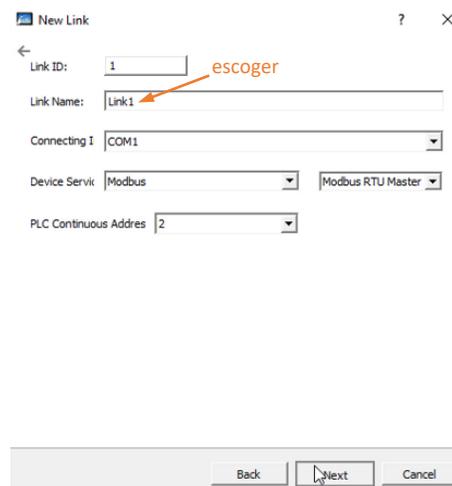
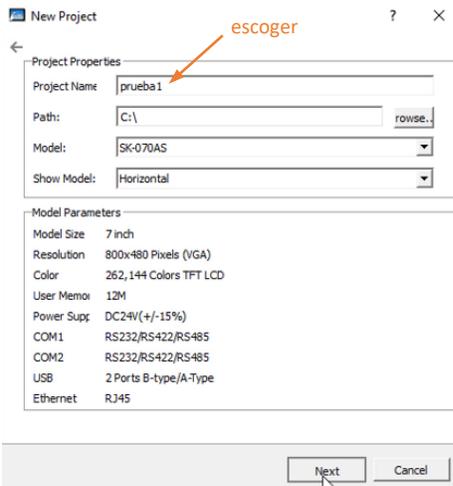


Fig. 4: Reconocimiento del HMI por la PC



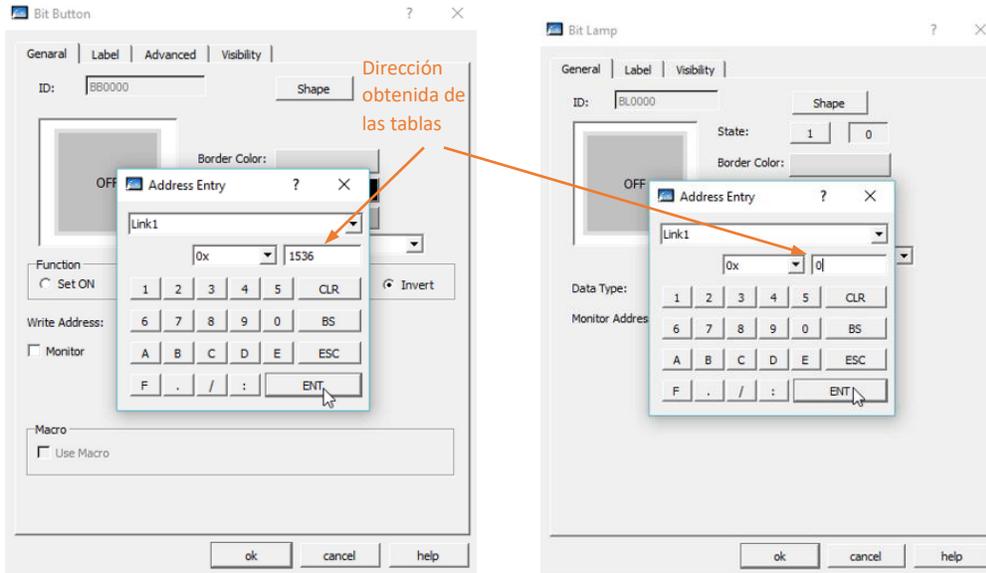


Fig. 5: Parametrización para comunicación Modbus

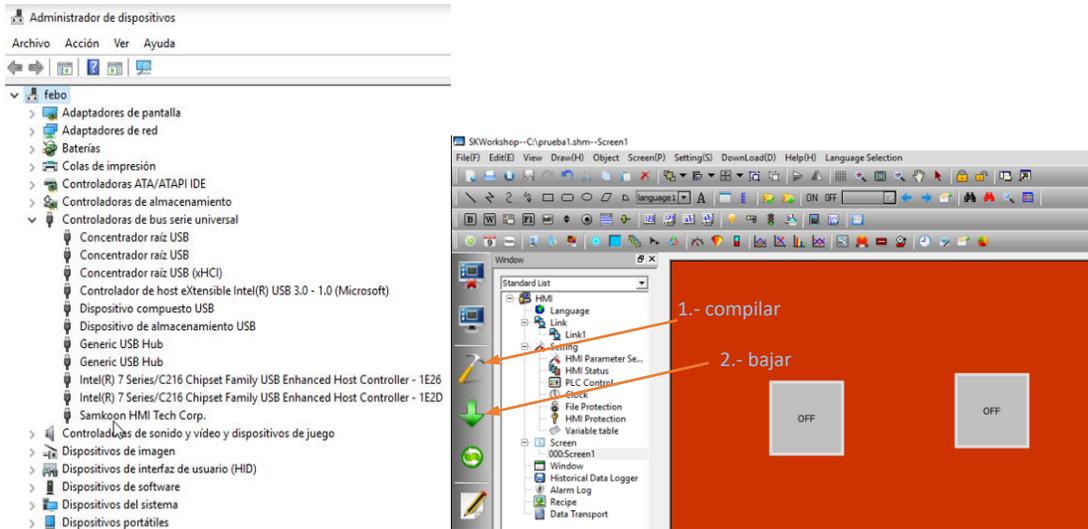


Fig. 6: Bajar el programa a la computadora personal

Digital outputs		BIT	R/W
<p>1 </p> <p>(0x)</p> <p>MODBUS code: 01(read) 05 (single Write)</p> <p>15 (Multiple Write)</p>	<p><b>ELC-6(CPU):</b>0~1</p> <p><b>ELC-18(CPU):</b>0~5 ELC-E-16(EXT1):8~15 ELC-E-16(EXT2):16~23 ELC-E-16(EXT3):24~31 . . ELC-E-16(EXT9):72~79 ELC-E-16(EXT10):80~87</p> <p><b>EXM-12/ELC-12(CPU):</b>0~7 ELC12-E-8(EXT1):8~15 ELC12-E-8(EXT2):16~23 ELC12-E-8(EXT3):24~31 . . ELC12-E-8(EXT7):56~63 ELC12-E-8(EXT8):64~71</p> <p><b>ELC-22/26(CPU):</b>0~9 ELC-E-16(EXT1):10~17 ELC-E-16(EXT2):18~25 ELC-E-16(EXT3):26~33 . . ELC-E-16(EXT9):74~81 ELC-E-16(EXT10):82~89</p> <p>SMS Output:512~515 SMS Message Output:516~525</p> <p>SSR-12(CPU):0~3</p>		

Fig. 7: Direcciones de comunicación Modbus con el EXM-12

Digital Flag		BIT	R/W
<p>F1 </p> <p>(0x)</p> <p>MODBUS code: 01(read)  05 (single Write)</p> <p>15 (Multiple Write)</p>	<p>ELC-6&amp; Economic ELC-12 Series: 1536~1567</p> <p><b>EXM-12/Standard ELC-12:</b> 1536~1663</p> <p>ELC-18 Series: 768~799 Upgraded ELC-18 Series: 1536~1663</p> <p>ELC-22/26 Series: 1536~1663</p> <p>SSR-12 series: 1536~1663</p>		

<p><b>Analog input</b> AI001  (4x)</p> <p>MODBUS code: 03(read)</p>	<p>EXM-12/ELC-12 Series: (1024~1279)</p> <p>CPU:1024~1031 EXT1:1032~1039 EXT2:1040~1047 .....</p> <p>EXT8:1088~1095</p> <p>Standard ELC-18 Series:(256~511)</p> <p>CPU:256~263 EXT1:264~271 EXT2:272~279 .....</p> <p>EXT9:328~335</p> <p>Upgraded ELC-18 Series: CPU:1024~1031 EXT1:1032~1039 EXT2:1040~1047 .....</p> <p>EXT9:1096~1103</p> <p>ELC-22/26 Series CPU :1024~1031 EXT1:1032~1039 EXT2:1040~1047 .....</p> <p>EXT9:1096~1103</p> <p>SSR-12 Series: 1024~1027</p>	<p>Signed short</p>	<p>R</p>
<p><b>Analog output</b> AQ001  (4x)</p> <p>MODBUS code: 03(read) 06(Single Write) 16(Multiple Write)</p>	<p>EXM-12/ELC-12 Series: (1280~1535)</p> <p>CPU:1280~1281 EXT1:1282~1283 EXT2:1284~1285 .....</p> <p>EXT8:1296~1297</p> <p>ELC-18 Series: (512~531) CPU:512~513 EXT1:514~515 EXT2:516~517 .....</p> <p>EXT9:530~531</p> <p>ELC-22/26/Upgraded ELC-18 Series: CPU:1280~1281 EXT1:1282~1283 EXT2:1284~1285 .....</p> <p>EXT9:1298~1299</p> <p>SSR-12 Series:1280~1281</p>	<p>Signed short</p>	<p>R/W</p>

Fig. 7: Direcciones de comunicación Modbus con el EXM-12

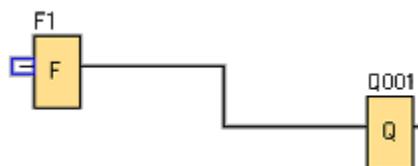


Fig. 8: Programa básico en el autómata para enlazar con el HMI

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Realizar un programa básico de entradas y salidas todo o nada en un autómata con bus de comunicación MODBUS (figura 8). La bandera “F” ejecutará como esclavo las instrucciones de la botonera programada en el HMI, este último ejecutará como maestro.

Programar en el HMI entradas y salidas figuras 3 a 6, haciendo coincidir el número de entradas y salidas con las programadas en el autómata.

Una vez bajado el programa en el HMI, se desconecta la computadora del HMI, en cambio se conecta la HMI al autómata y se verifica si este último recibe las instrucciones a través de la pantalla táctil del HMI.

Repetir el mismo ejercicio para aplicaciones con más entradas y salidas.

Repetir el mismo ejercicio haciendo uso de entradas y salidas analógicas.

**MATERIAL:**

- HMI Samkoon SK – 070AS
- Autómata x-logic
- Cables de conexión

**CUESTIONARIO:**

1. Enumere los materiales usados en la presente práctica.
2. Dibujar todos los programas realizados en el autómata.
3. Dibujar todos los programas realizados en el HMI.
4. Dibujar paso a paso las conexiones eléctricas a realizadas.

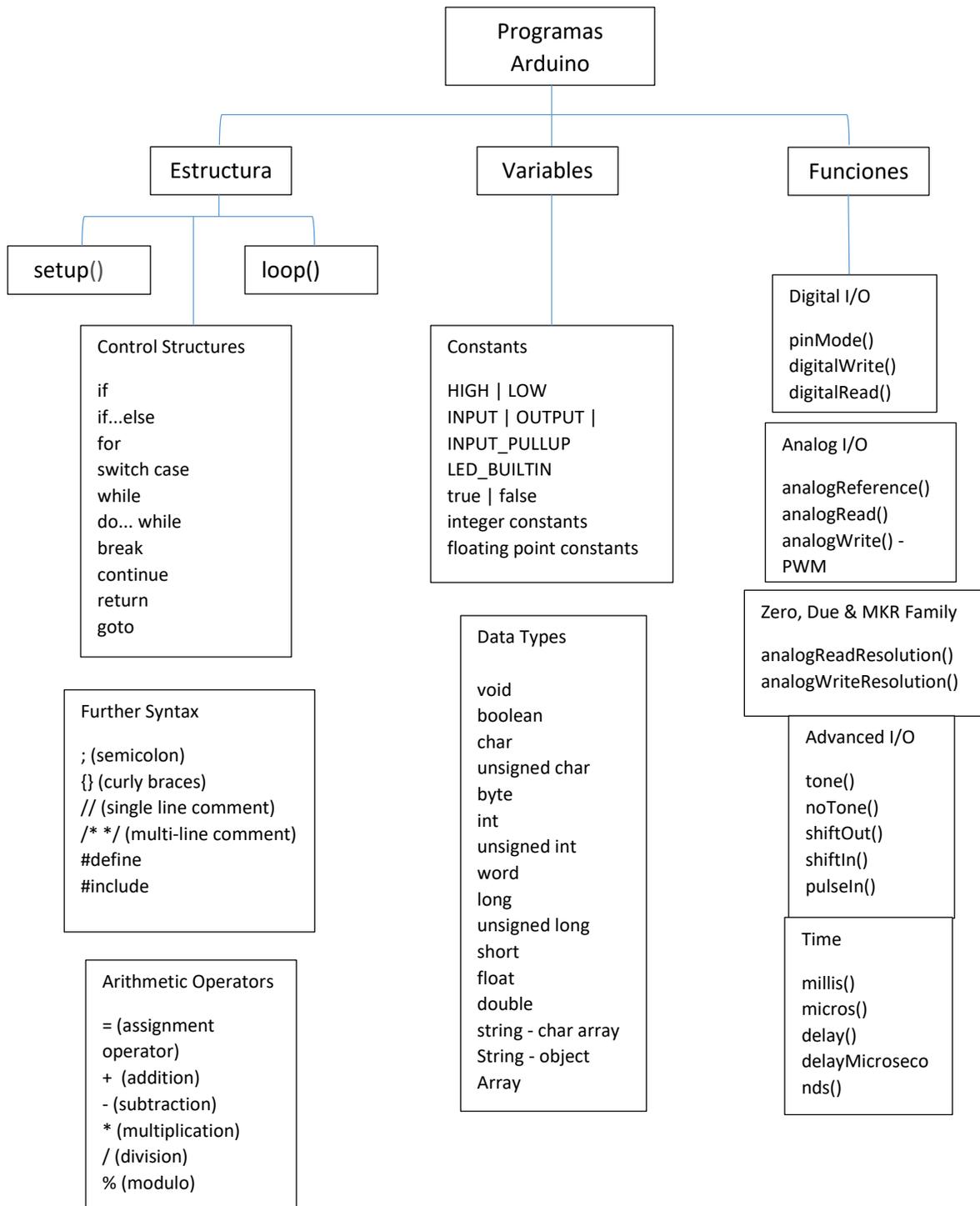
# PRÁCTICA 10

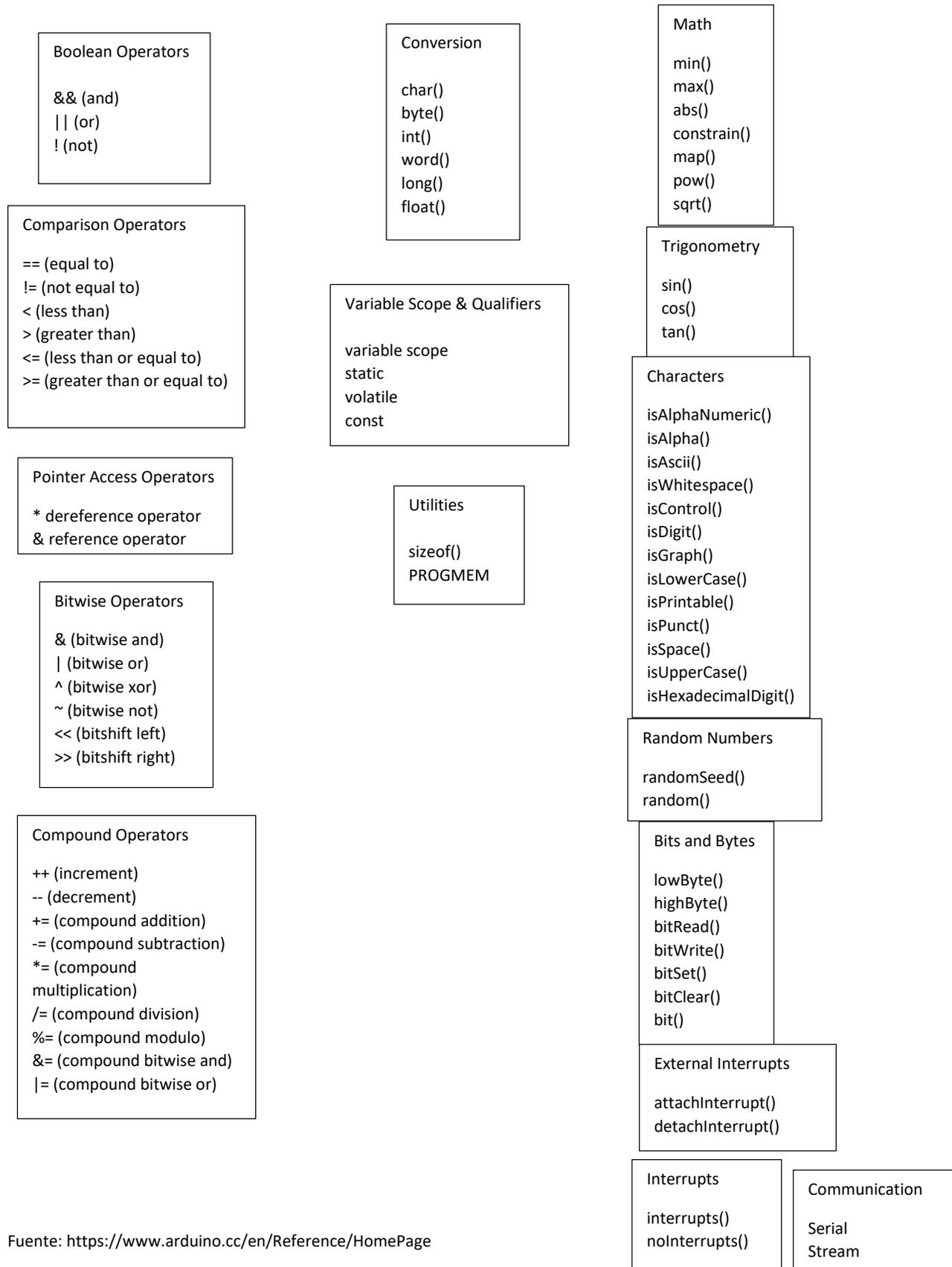
## Programas en Arduino

**OBJETIVO:** Introducción a la programación de microcontroladores Arduino.

**COMPETENCIA:** Conecta, programa y ejecuta aplicaciones básicas con Arduino.

**FUNDAMENTO TEÓRICO:**





Fuente: <https://www.arduino.cc/en/Reference/HomePage>

## Estructuras de programación

### Estructura condicional simple: IF

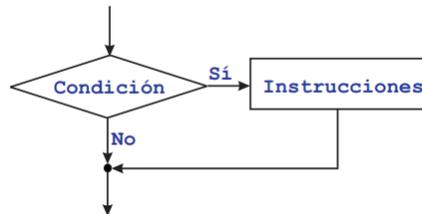
Este es el tipo más sencillo de estructura condicional. Sirve para implementar acciones condicionales del tipo siguiente:

Si se verifica una determinada condición, ejecutar una serie de instrucciones y luego seguir adelante.

Si la condición NO se cumple, NO se ejecutan dichas instrucciones y se sigue adelante.

```

if (Condición)
{
    // Instrucciones
}
    
```



### Estructura condicional doble: IF - ELSE

Este tipo de estructura permite implementar condicionales en los que hay dos acciones alternativas:

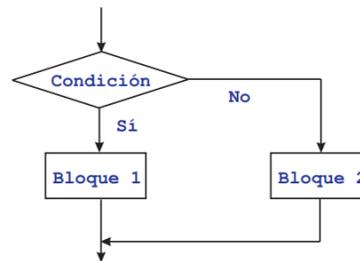
Si se verifica una determinada condición, ejecutar una serie de instrucciones (bloque 1).

Si no, esto es, si la condición NO se verifica, ejecutar otra serie de instrucciones (bloque 2).

En otras palabras, en este tipo de estructuras hay una alternativa: se hace una cosa o se hace la otra. En ambos casos, se sigue por la instrucción siguiente a la estructura IF - ELSE.

```

if (Condición)
{
    // Bloque 1
}
else
{
    // Bloque 2
}
    
```



### Estructura condicional múltiple: IF - ELSEIF - ELSE

En su forma más general, la estructura IF - ELSEIF - ELSE permite implementar condicionales más complicados, en los que se “encadenan” condiciones en la forma siguiente:

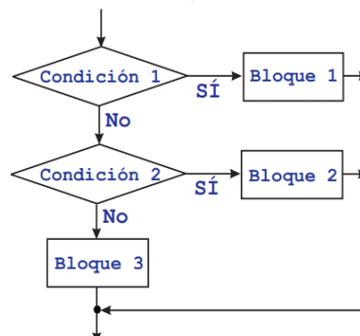
Si se verifica la condición 1, ejecutar las instrucciones del bloque 1.

Si no se verifica la condición 1, pero Sí se verifica la condición 2, ejecutar las instrucciones del bloque 2.

Si no, esto es, si no se ha verificado ninguna de las condiciones anteriores, ejecutar las instrucciones del bloque 3.

```

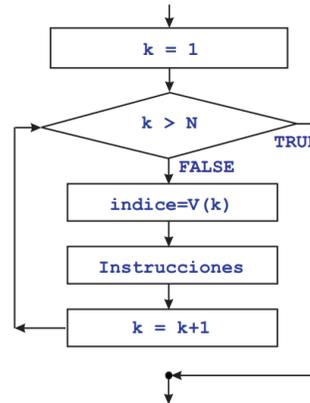
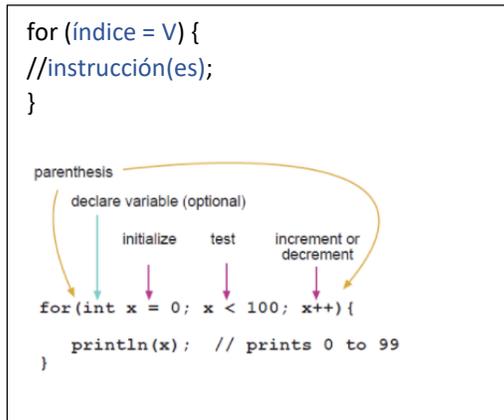
if (Condición 1)
{
    // Bloque 1
}
else if (Condición 2)
{
    // Bloque 2
}
else
{
    // Bloque 3
}
    
```



### Estructura de repetición indexada: FOR

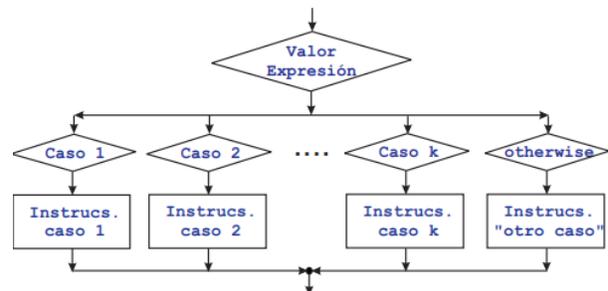
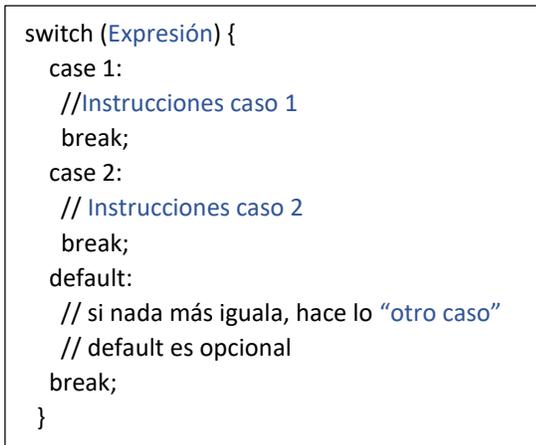
Este tipo de estructura permite implementar la repetición de un cierto conjunto de instrucciones un número pre-determinado de veces.

Para ello se utiliza una variable de control del bucle, llamada también índice, que va recorriendo un conjunto pre-fijado de valores en un orden determinado. Para cada valor del índice en dicho conjunto, se ejecuta una vez el mismo conjunto de instrucciones.



### Estructura de elección entre varios casos: SWITCH

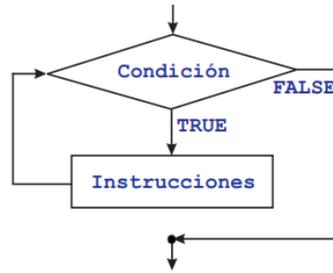
Este tipo de estructura permite decidir entre varios caminos posibles, en función del valor que tome una determinada instrucción.



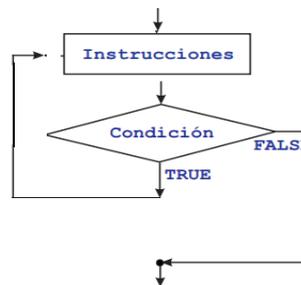
### Estructura repetitiva condicional: WHILE

Permite implementar la repetición de un mismo conjunto de instrucciones mientras que se verifique una determinada condición: el número de veces que se repetirá el ciclo no está definido a priori.

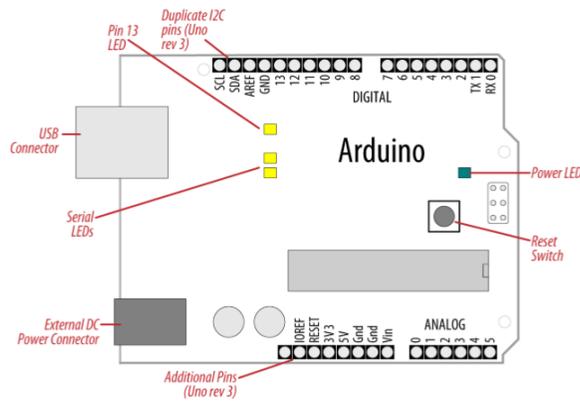
```
while(Condición){
  // Instrucción(es)
}
```



```
do
{
  // Instrucciones
} while (condición);
```



### ARDUINO UNO



Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Límite)	6 – 20V
Pines para entrada- salida digital.	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin 3.3V	50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Fig. 1: Características de Arduino UNO

### Análisis de entradas y salidas de Arduino UNO

Cada uno de los 20 pines digitales de Arduino puede ser usado como entrada o salida, usando las funciones adecuadas. Todos los pines funcionan a 5 V, por lo que si se conecta a una tensión mayor puede que dejen de funcionar permanentemente. Cada pin tiene una salida de intensidad de 40 mA máximo y una resistencia de pull-up integrada de unos 20-50 kOhms.

Además algunos pines tienen alguna función adicional:

Serie: 0 (RX) y 1 (TX). Se usan para recibir y transmitir datos serializados con niveles TTL. Estos pines están conectados al conversor USB-Serie para comunicarse con el ordenador. Si en algún proyecto se utiliza esta comunicación, sus pines digitales no estarán disponibles.

Interrupciones externas: 2 y 3, pueden ser configurados para interrumpir la ejecución de código y realizar otra tarea. Pueden ser configurados para interrumpir por nivel, por flanco o por cambio de valor.

PWM: Seis pines marcados en la placa con el símbolo ~, pueden proveer una salida de PWM con una resolución de 8 bits (0 a 255) y con una frecuencia en el orden de kHz.

Entradas Analógicas: Pines A0 a A5, con resolución de  $2^{10}$  (0 a 1023).

SPI: Protocolo de comunicación serie mediante 4 hilos muy utilizado para comunicación de corto alcance entre microcontroladores. Suele ser utilizado también para programar el microcontrolador Atmel mediante un programador compatible.

I2C: Protocolo de comunicación serie mediante 2 hilos muy utilizado para la comunicación de corto alcance con sensores inteligentes.

OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:

<https://www.arduino.cc/en/Tutorial/BuiltInExamples>

El docente le planteará un programa para generar algún tipo de señal a visualizar con salida número 13 (led incluido en Arduino).

El docente le planteará un programa de transmisión de datos mediante comunicación UART.

MATERIAL:

- Computadora personal con Software Arduino (IDE), mismo que puede bajarse libremente de <https://www.arduino.cc/en/Main/Software>
- Arduino UNO con conector de comunicación UART.

CUESTIONARIO:

1. El docente le planteará un programa para generar algún tipo de señal a visualizar con salida número 13 (led incluido en Arduino).
2. El docente le planteará un programa de transmisión de datos mediante comunicación UART.

# PRÁCTICA 11

## Introducción Arduino (UNO) entradas-salidas digitales y conexión interfaz

**OBJETIVO:** Introducción a la conexión y programación de entradas/salidas digitales del microcontrolador Arduino.

**COMPETENCIA:** Conecta, programa y ejecuta aplicaciones básicas con entradas y salidas de microcontrolador Arduino.

**FUNDAMENTO TEÓRICO:**

Señales de entrada discreta todo o nada:

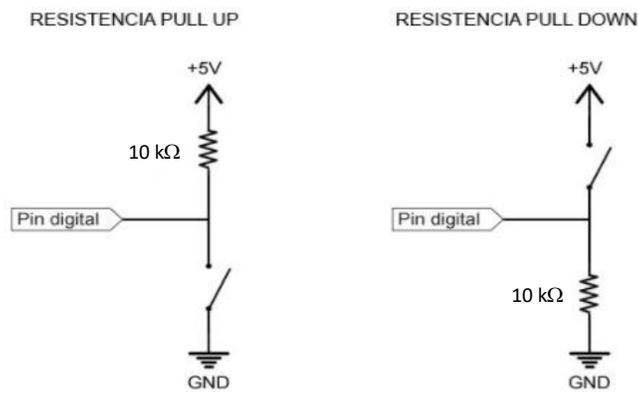


Fig.1: Modos de conexión entradas discretas a Arduino

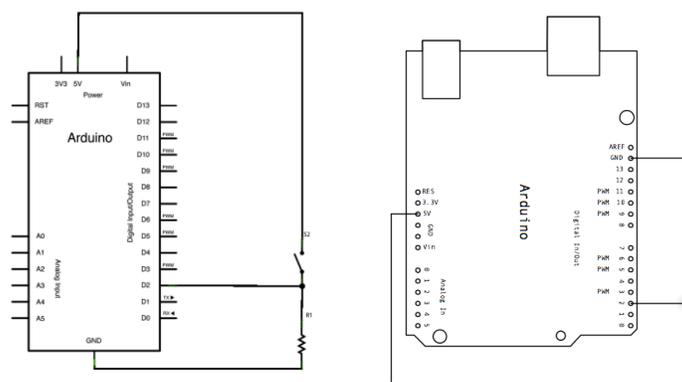


Fig. 2: Entradas "pull down" en Arduino

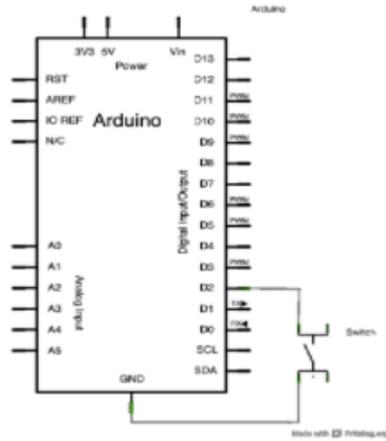


Fig. 3: Conexión Pull Up (resistencia interna Arduino)

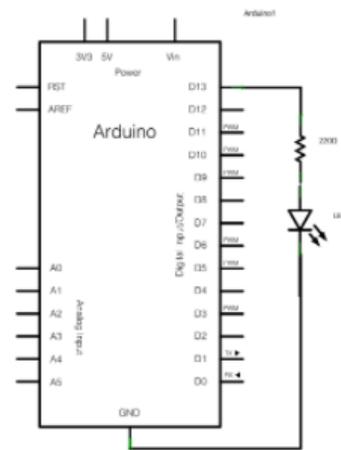
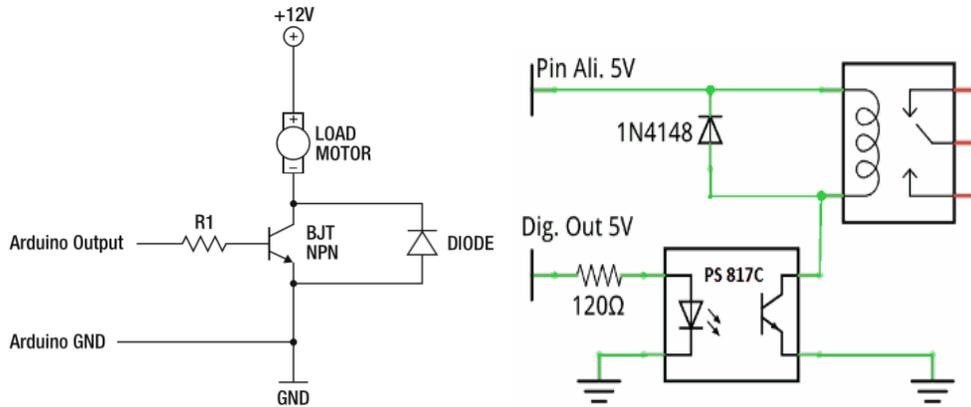


Fig. 4: Conexión de salidas discretas

Interfaz para circuitos de potencia:



(a) Con transistor operando en saturación

(b) Con optoacoplador y relé para potencia

Fig. 2: Circuitos interfaz para potencia

La corriente que puede generar el Arduino es de aproximadamente 20 a 30 mA, insuficientes para aplicaciones que requieran mayores niveles de corriente, sin embargo con un transistor bipolar NPN, como se muestra en la figura 2(a), funcionando en SATURACIÓN, éste se comporta como un conmutador que permite el funcionamiento de por ejemplo, un motor de CC 12V. En vez del motor se puede energizar la bobina de un relé como se ve figura 2 (b), donde además se emplea un optoacoplador para dar aislamiento galvánico.

Con un relé se puede formar parte de un circuito de 220 CA alimentando el contactor de un motor trifásico.

En la práctica debe medirse con un voltímetro la caída de voltaje entre colector y emisor del transistor u optoacoplador para cuando se alimenta su base, éste estará saturado cuando la lectura del voltímetro se hace próximo a cero.

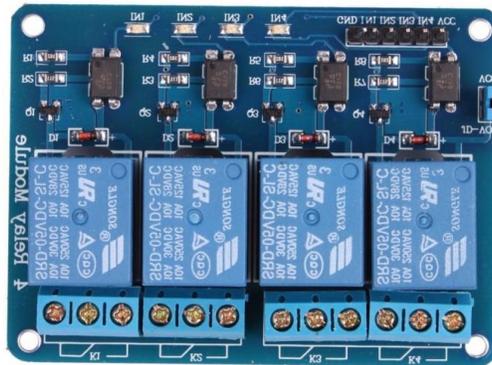


Figura 3: Módulo de relés para Arduino

También están disponibles en el mercado módulos para Arduino con 1, 2, 4 (como figura 3) y 8 relés para fácil conexión a Arduino.

#### OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:

El docente le planteará un programa para generar algún tipo de relación digital entrada/salida con Arduino.

Aplique alguno de los circuitos interfaz, por ejemplo los de las figuras 2 para energizar con una salida de Arduino, algún elemento de potencia como un contactor.

Mida caída de voltaje entre colector y emisor del transistor para verificar si el mismo está saturado.

#### CUESTIONARIO:

1. Dibuje el circuito armado en laboratorio.
2. Investigue otros diagramas de circuitos interfaz y dibújelos.
3. Indique las diferencias de conexión “pull up” y “pull down” en Arduino.
4. Explique en pocas palabras qué entiende por saturación de un transistor.
5. Indique cómo escoge el valor de R1 en el circuito de la figura 2 (a).
6. Enumere los materiales usados en la presente práctica.

## PRÁCTICA 12

### Adquisición datos LabVIEW con usb 6001 (NI)

**OBJETIVO:** Introducción a la adquisición de datos mediante data logger y programación de LabVIEW.

**COMPETENCIA:** Conecta, programa en LabVIEW y adquiere datos en una computadora con un data logger.

**FUNDAMENTO TEÓRICO:**

Feature	NI USB-6001	NI USB-6002	NI USB-6003
<b>Analog Input</b>			
Analog-to-digital converter (ADC) Resolution	14-bit	16-bit	16-bit
Maximum Sample Rate (aggregate)	20 kS/s	50 kS/s	100 kS/s
<b>Analog Output</b>			
DAC Resolution	14-bit	16-bit	16-bit
Absolute Accuracy, Typical, at full scale	9.1 mV	8.6 mV	8.6 mV

Fig. 1: Características de módulos adquisición de datos NI

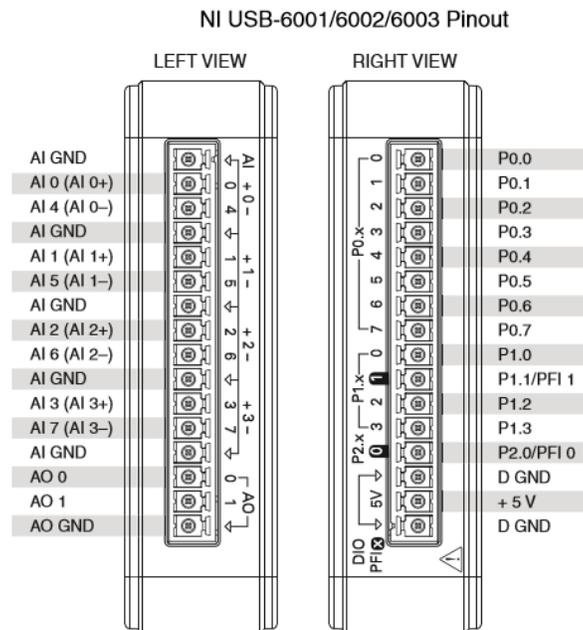


Fig. 2: Módulo 6001

Signal Name	Reference	Direction	Description
AI GND	—	—	<b>Analog Input Ground</b> —The reference point for single-ended analog input measurements.
AI <0..7>	AI GND	Input	<b>Analog Input Channels 0 to 7</b> —For single-ended measurements, each signal corresponds to one analog input voltage channel. For differential measurements, AI 0 and AI 4 are the positive and negative inputs of differential analog input channel 0. The following signal pairs also form differential input channels: AI <1,5>, AI <2, 6>, and AI <3, 7>. Refer to the <a href="#">Analog Input</a> section for more information.

Fig. 3: Descripción de señales

Signal Name	Reference	Direction	Description
AO GND	—	—	<b>Analog Output Ground</b> —The reference point for analog output.
AO <0, 1>	AO GND	Output	<b>Analog Output Channels 0 and 1</b> —Supplies the voltage output of the AO channels. Refer to the <a href="#">Analog Output</a> section for more information.
P0.<0..7>	D GND	Input or Output	<b>Port 0 Digital I/O Channels 0 to 7</b> —You can configure each signal individually as an input or output. Refer to the <a href="#">Digital I/O</a> section for more information.
P1.<0..3>	D GND	Input or Output	<b>Port 1 Digital I/O Channels 0 to 3</b> —You can configure each signal individually as an input or output. Refer to the <a href="#">Digital I/O</a> section for more information.
P2.0	D GND	Input or Output	<b>Port 2 Digital I/O Channel 0</b> —You can configure each signal individually as an input or output. Refer to the <a href="#">Digital I/O</a> section for more information.
PFI 0, 1	D GND	Input	<b>Programmable Function Interface or Digital I/O Channels</b> —An edge counter input or digital trigger input. Refer to the <a href="#">PFI 0 and PFI 1</a> section for more information.
D GND	—	—	<b>Digital Ground</b> —The reference point for digital signals.
+5 V	D GND	Output	<b>+5 V Power Source</b> —Provides +5 V power up to 150 mA. Refer to the <a href="#">+5 V Power Source</a> section for more information.

Fig. 4: Descripción de señales

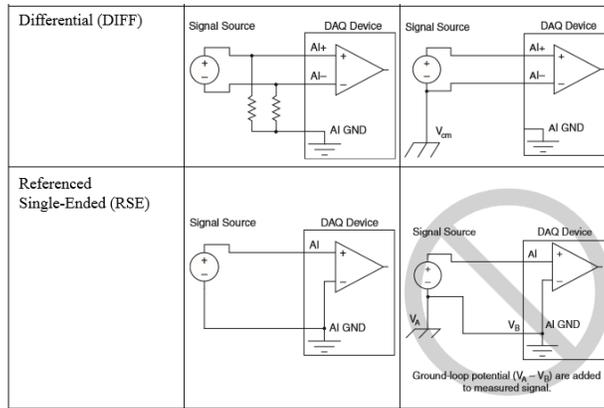


Fig. 5: Modos de conexión de señales de entrada

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Adquisición de temperatura con termistor:

El termistor a usarse en laboratorio es marca EPCOS modelo/serie: N° B57164K103J, que corresponde a un tipo NTC, 10kΩ a 25°C y tolerancia de ± 5%.

Información complementaria: [youtu.be/xi0VlpGpf4w](https://youtu.be/xi0VlpGpf4w)

$$T(R) = \left( A_1 + B_1 \ln \left( \frac{R}{R_{25}} \right) + C_1 \ln^2 \left( \frac{R}{R_{25}} \right) + D_1 \ln^3 \left( \frac{R}{R_{25}} \right) \right)^{-1} \quad (1) \text{ Relación resistencia temperatura}$$

Donde:  $A_1=3,354016 \times 10^{-3}$ ,  $B_1=2,569850 \times 10^{-4}$ ,  $C_1=2,620131 \times 10^{-6}$ ,  $D_1=6,383091 \times 10^{-8}$

Del divisor de voltaje,

Se tiene:  $V_{salida} = \frac{R_{termistor}}{R_{termistor} + R_2} \cdot V_{CC}$ ,

despejando  $R_{termistor}$ , se tiene:

$$R_{termistor} = \left( \frac{V_{salida}}{V_{CC} - V_{salida}} \right) \cdot R_2 = \frac{V}{(5-V)} \cdot 10000 \quad (2) \text{ Resistencia del termistor}$$

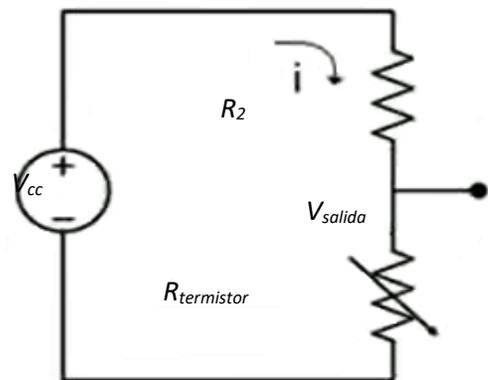


Fig. 6: Circuito divisor de voltaje

Información Complementaria: [youtu.be/9KUVD7RkxNI](https://youtu.be/9KUVD7RkxNI)

Opcionalmente se puede calcular la incertidumbre de medición de la resistencia, aplicando la fórmula:

$$U_R = S_R \cdot \frac{z}{\sqrt{n}} \quad \text{pero:} \quad S_R = \frac{\partial R}{\partial V_{salida}} \cdot S_{V_{salida}} \quad \text{ya que:} \quad \Delta R = \frac{\partial R}{\partial V_{salida}} \cdot \Delta V_{salida}$$

Por tanto puede también desplegarse el resultado de incertidumbre de medida la resistencia:

$$U_R = S_R \cdot \frac{z}{\sqrt{n}} = \left( \frac{V_{CC} \cdot R_2}{(V_{CC} - V_{salida})^2} \right) \cdot \frac{z}{\sqrt{n}} \quad (3) \quad \text{Incertidumbre de medición de la resistencia}$$

Información complementaria: [youtube.com/watch?v=7znYlkk-mw](https://www.youtube.com/watch?v=7znYlkk-mw)

Donde:  $n$  podría ser 100 para 100 mediciones  $z = 1,96$  para 95% Nivel de Confianza

Las ecuaciones 1,2, y 3 se emplean en el programa de la figura 7 con bloques denominados “Fórmula”, “Fórmula 2” y “Fórmula 4”:

Fórmula  $V/(5-V) \cdot 10000$

Fórmula 2  $50000/(5-V) \cdot 2 \cdot S_v \cdot 1,96/\text{sqrt}(100)$

Fórmula 3  $(3.354016E-03 + 2.569850E-04 \cdot \ln(R/10000) + 2.620131E-06 \cdot (\ln(R/10000))^2 + 6.383091E-08 \cdot (\ln(R/10000))^3) \cdot -1$

(las separaciones decimales con punto en vez de coma si es versión en inglés)

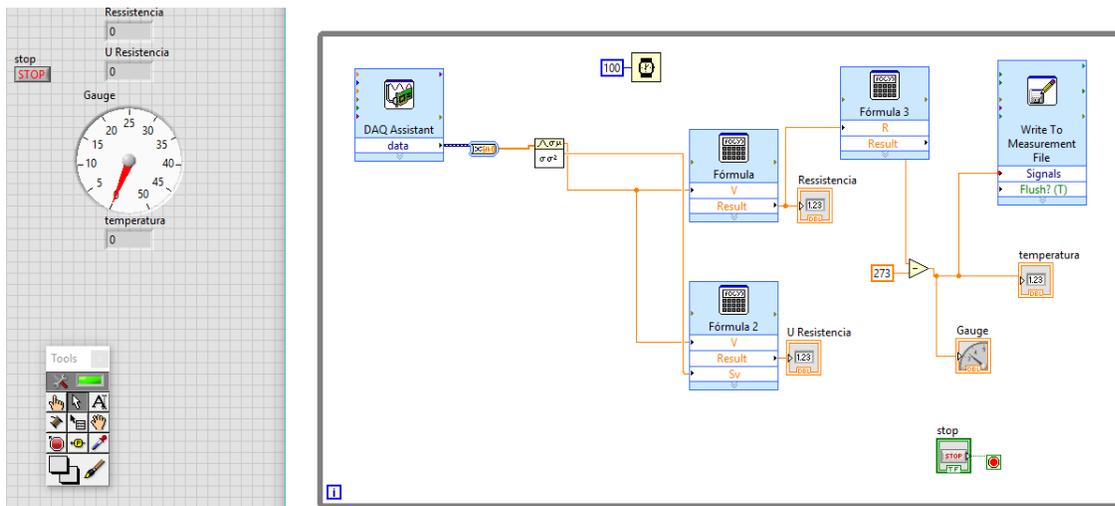


Fig. 7: Programa para medición de temperatura con un termistor

El bloque “DAQ Assistant” debe configurar con el módulo 6001 conectado a la computadora.

Los bloques de fórmulas ahorran el uso de muchos operadores matemáticos.

El bloque “Write to Measurement” puede configurar por ejemplo como se muestra en la figura 8, de ese modo podrá registrar los datos en una planilla Excel, el nombre del archivo se muestra en el lado superior izquierdo de la figura.

Figura 8: Configuración del bloque Write to Measurement

### Adquisición de señales generadas por Arduino:

Programa en Arduino la generación de una señal senoidal mediante una de sus salidas pseudo analógicas y mida la misma con el data logger para visualizarla con un instrumento de medida virtual de LabVIEW.

#### MATERIALES:

- NI USB-6001
- Arduino UNO
- Computadora personal con IDE (Arduino), NI-LabView y NI-DAQmx
- Termistor modelo/serie: N° B57164K103J
- Resistencia de 10 K $\Omega$
- Fuentes de voltaje

#### CUESTIONARIO:

1. Imprima el formulario EXCEL con los valores de temperatura obtenidos con el termistor y genere una curva en el tiempo.
2. Dibuje los circuitos armados en laboratorio.
3. Escriba el programa que usó para generar una señal senoidal con Arduino.
4. Dibuje la señal registrada con los instrumentos virtuales de LabVIEW.

## PRÁCTICA 13

### Control con LabVIEW (MyRIO NI), secuencias

**OBJETIVO:** Introducción al manejo de myRIO de NI programando secuencias temporales y de eventos con LabVIEW.

**COMPETENCIA:** Conecta, programa y hace aplicaciones de secuencias en función a tiempo y eventos con myRIO y LabVIEW.

**FUNDAMENTO TEÓRICO:**

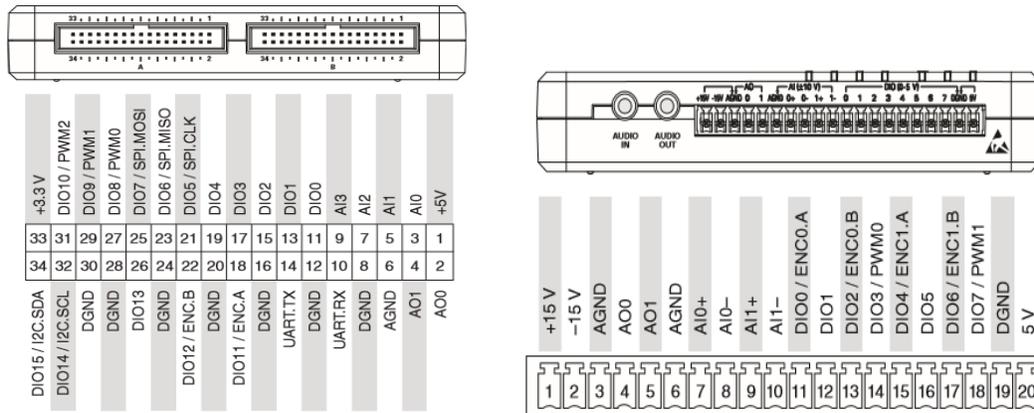


Fig. 1: Detalle de entradas y salidas de MyRIO- NI

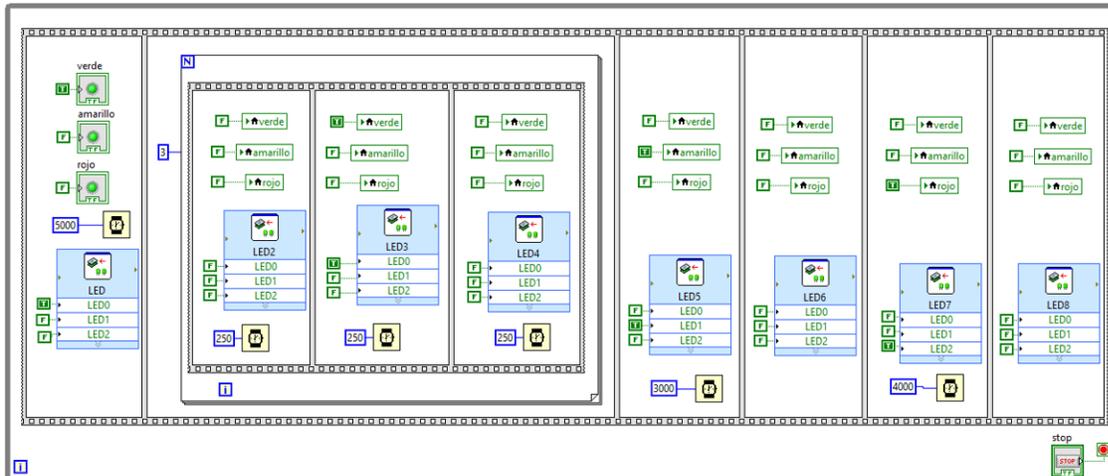


Fig. 2: Secuencia en base a tiempo con "flat sequence"

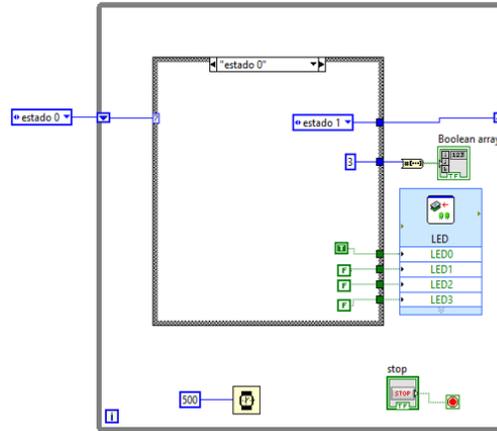


Fig. 3: Secuencia en base a tiempo con "case structure"

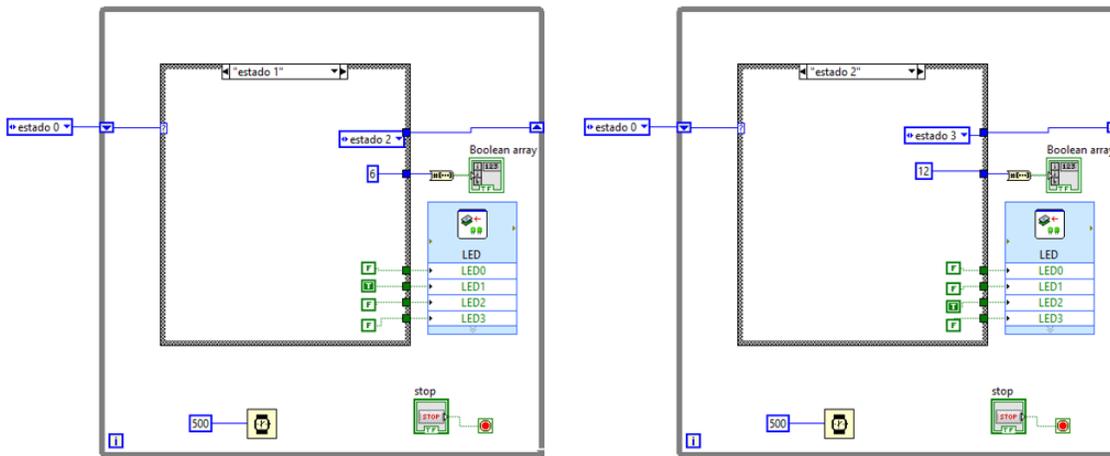


Fig. 4: Secuencia en base a tiempo con "case structure"

Secuencia por eventos

$$ETAPA 1 = \overline{DIO2} \times DIO3 \times DIO4$$

$$ETAPA 2 = DIO2 \times \overline{DIO3} \times DIO4$$

$$ETAPA 3 = DIO2 \times DIO3 \times \overline{DIO4}$$

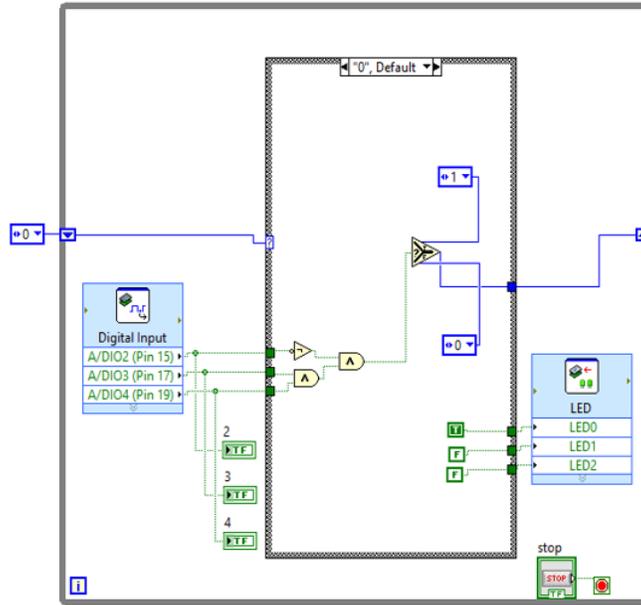


Fig. 5: Secuencia de eventos con "case structure"

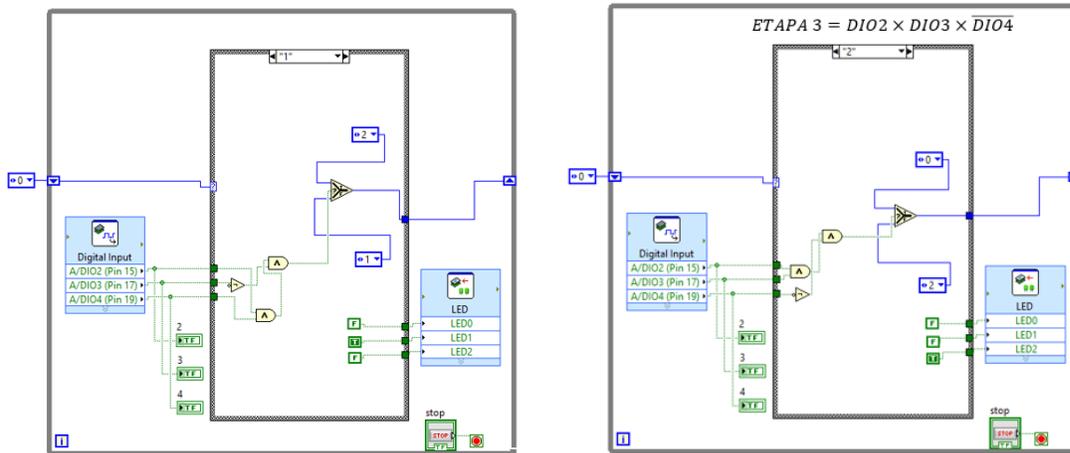


Fig. 6: Secuencia de eventos con "case structure"

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

En base a figuras 2, 3 y 4; realice programas de secuencia en función a lapsos de tiempo, pero además incluyendo o combinando salidas físicas de MyRIO, antes de hacer las conexiones físicas consulte con el docente o auxiliar.

En base a figuras 5 y 6, realice programas de secuencia en función a eventos en las señales de entrada, pero además incluyendo o combinando con entradas y salidas físicas de MyRIO, antes de hacer las conexiones físicas consulte con el docente o auxiliar.

Para las conexiones considere si las conexiones son "pull up" o "pull down"

## MATERIALES:

- Computadora con LabVIEW y módulo MyRIO
- NI – MyRIO
- Pulsadores, y señalizadores

## CUESTIONARIO:

1. Dibuje los circuitos tanto de entradas como de salidas físicas realizadas en la práctica.
2. Dibuje los programas de secuencias hechos en laboratorio, tanto el de secuencia por tiempo como el de secuencia por eventos.
3. Dibuje un programa que realice detección de flancos de subida de una señal de entrada y que además cuente los mismos.

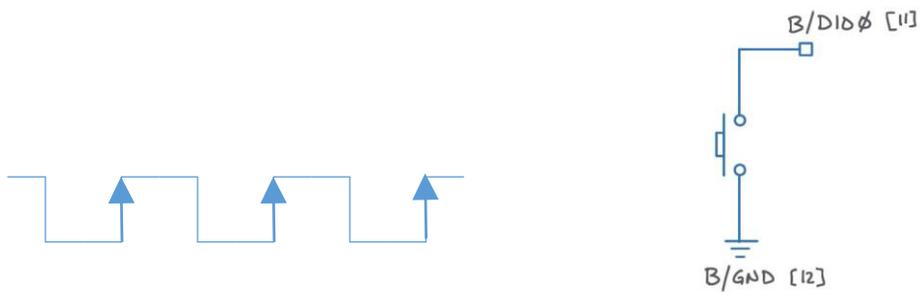


Fig. 7: Detección de flancos de subida

## PRÁCTICA 14

## Integración Arduino (UNO) y LabVIEW, almacenamiento de datos

**OBJETIVO:** Introducción al manejo de entrada analógica y salida pseudo analógica combinando tarjeta Arduino con programas LabVIEW.

**COMPETENCIA:** Conecta, programa y hace aplicaciones con entrada analógica y salida pseudo analógica de una tarjeta Arduino UNO e instrumentos virtuales de LabVIEW.

**FUNDAMENTO TEÓRICO:**

LabVIEW no cuenta por defecto con módulos o bloques para conectar Arduino, por ello primero debe bajarse de LabVIEW el “NI-VISA” y el “VI Package Manager”, luego de ello se instalan los bloques Arduino en el menú de LabVIEW, existen muchos tutoriales virtuales que muestran cómo hacerlo, como en los siguientes vínculos:

<https://www.youtube.com/watch?v=p5qlwufUeEM>

[http://www.naylampmechatronics.com/blog/23\\_TUTORIAL-ARDUINO-Y-LABVIEW.html](http://www.naylampmechatronics.com/blog/23_TUTORIAL-ARDUINO-Y-LABVIEW.html)

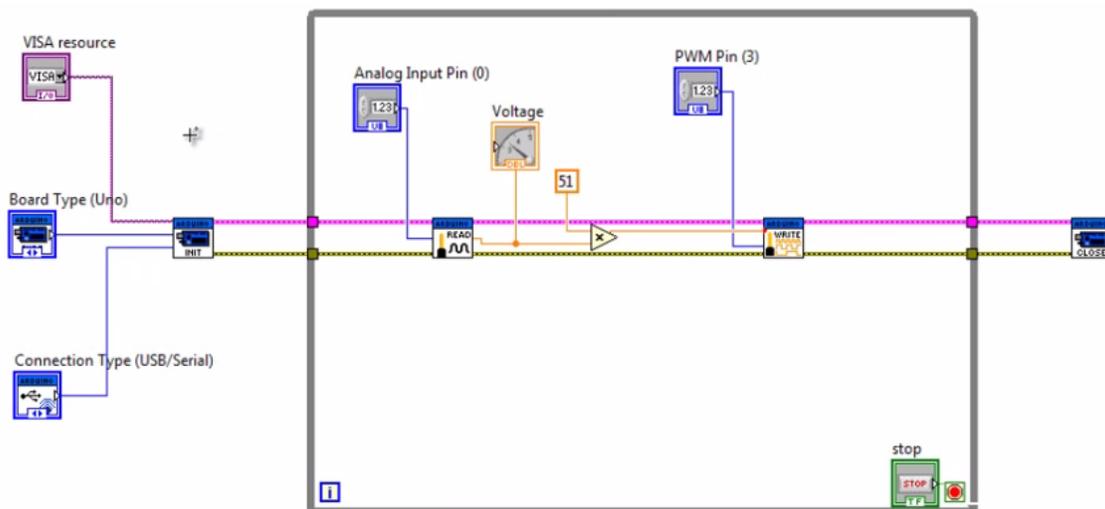


Fig. 1: Programa Arduino Read Analog Voltage con LabVIEW

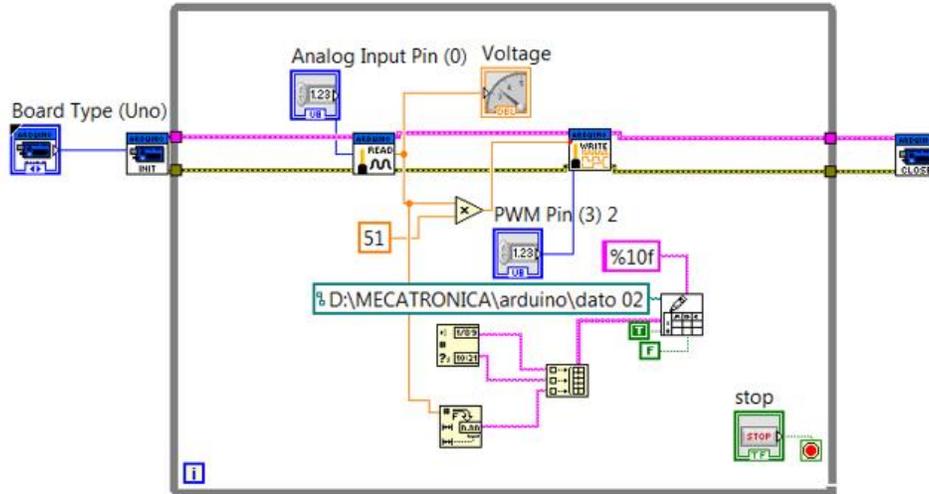


Fig. 2: Programa Arduino Read Analog Voltage y adquisición de datos con LabVIEW



Fig. 3: Funciones "File I/O" Write To Spreadsheet File; Write To Measurement File

En la figura 3 se muestran dos funciones de "File I/O" que pueden emplearse para almacenar los registros en un archivo Excel, para la figura 2 se utilizó "Write To Spreadsheet File", intente utilizar "Write To Measurement File".

Debe abrir el LIFA\_Base que al bajarlo usualmente se almacena por defecto en un directorio parecido al que se detalla a continuación:

Disco local (C:)/

Archivos de programa(x86)/

National Instrument/

LabVIEW 2012/

vi.lb/

LabVIEW Interface for Arduino/

Firmware/

LIFA\_Base

**OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:**

Programa en LabVIEW de acuerdo a ejemplos de las figuras 1 y 2, y haga pruebas en Arduino UNO.

Realice otras aplicaciones acorde a su criterio o del docente.

**MATERIAL:**

- Computadora con LabVIEW – Arduino
- Arduino UNO
- Potenciómetros leds y cables

**CUESTIONARIO:**

1. Haga un flujograma donde muestre paso a paso la forma de cargar módulo Arduino en LabVIEW, además de la forma cómo baja el programa a la tarjeta Arduino.
2. Dibuje los circuitos y programas realizados en clases.
3. Proponga ejemplo a realizar con uso de Arduino y LabVIEW.

## (ANEXO)

Práctica 15-A “VEHÍCULO DE RASTREO (TRONIC)”: Proyecto de la asignatura LMEC 432 gestión 1/15, documentado por Nelson Condorena auxiliar de la asignatura en las gestiones 2016 y 2017.

Práctica 15-B Comunicación en protocolo MODBUS entre Arduino y HMI: Proyecto realizado en la asignatura LMEC 432 en Gestión1/18 por el estudiante Ariel Anza.

GUÍA BÁSICA DE AUTÓMATA PROGRAMABLE PC WORX: Manual de introducción al uso del autómata programable PC WORX instalado en el tablero didáctico del laboratorio del IIME.

GUÍA BÁSICA DE AUTÓMATA PROGRAMABLE MICROLOGIX 1100 B (Allen Bradley): Manual de introducción al uso del autómata programable MicroLogix 1100 B del laboratorio del IIME.

## PRÁCTICA 15-A

# Aplicación de Arduino y shields en proyecto TRONIC

### OBJETIVOS

#### OBJETIVO GENERAL

- Construir un vehículo de rastreo para obtener datos de un espacio físico de forma automática.

#### OBJETIVOS ESPECIFICOS

- Diseñar la estructura mecánica del vehículo de rastreo.
- Realizar el circuito de control para la obtención de datos de un espacio físico.
- Realizar el circuito de potencia para el funcionamiento del motor del vehículo de rastreo.
- Realizar la aplicación interactiva para la maniobra del funcionamiento del vehículo.

#### FUNDAMENTO TEORICO

##### Arduino

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

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa. Se programa en el ordenador para que la placa controle los componentes electrónicos.

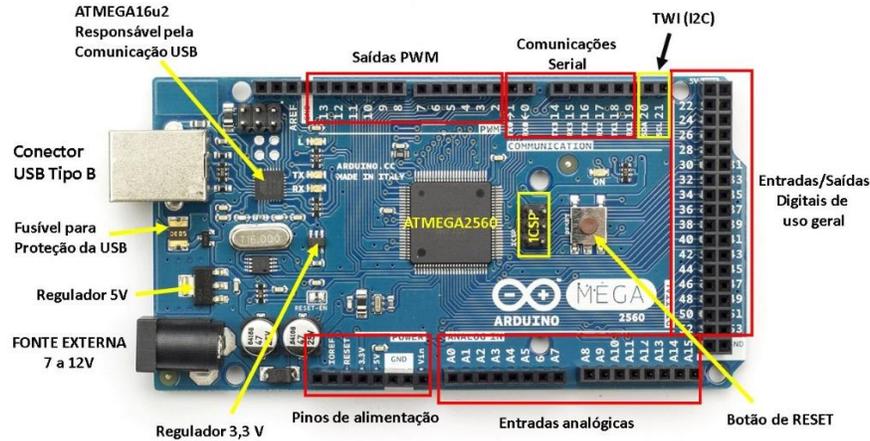
Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado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

##### Arduino Mega

El Arduino Mega posee 16 entradas analógicas, 30 entradas y salidas digitales, 11 salidas PWM, conexión usb y puerto de alimentación entre 7 y 12 V.

Esta placa ofrece mayor rendimiento que el Arduino Leonardo, uno, nano, entre otros, además que tiene mayor cantidad de entradas y salidas.

FIGURA 1: Arduino Mega 2560



FUENTE: [www.arduino.cc](http://www.arduino.cc)

Las características principales de la placa arduino mega 2560 se pueden detallar en la siguiente tabla:

TABLA 1: Características Arduino Mega 2560

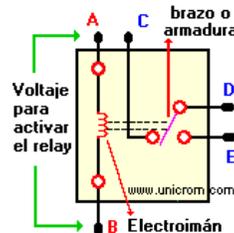
Característica	Descripción
Microcontrolador	ATmega 2560
Voltaje de Operación	5V
Tensión de Entrada (Recomendada)	7-12V
Tensión de Entrada (límite)	6-20V
Pines Digitales de E/S	54 (15 para PWM)
Pines de Entrada Analógicos	16
Corriente DC por pin E/S	40mA
Corriente DC para pin 3.3V	50mA
Memoria Flash	256 KB
SRAM	8 KB
EEPROM	4 KB
Frecuencia de Reloj	16 MHz

FUENTE: [www.monografias.com](http://www.monografias.com)

### Relé – Relay - Relevador

El Relé es un interruptor operado magnéticamente. El relé se activa o desactiva (dependiendo de la conexión) cuando el electroimán (que forma parte del relé) es energizado (le ponemos un voltaje entre sus terminales para que funcione). Esta operación causa que exista conexión o no, entre dos o más terminales del dispositivo (el relé). Esta conexión se logra con la atracción o repulsión de un pequeño brazo, llamado armadura, por el electroimán. Este pequeño brazo conecta o desconecta los terminales antes mencionados.

FIGURA 2: Funcionamiento del relé



FUENTE: [www.unicrom.com](http://www.unicrom.com)

Si el electroimán está activo jala el brazo (armadura) y conecta los puntos C y D. Si el electroimán se desactiva, conecta los puntos D y E. De esta manera se puede conectar algo, cuando el electroimán está

activo, y otra cosa conectada, cuando está inactivo. Es importante saber cuál es la resistencia del bobinado del electroimán (lo que está entre los terminales A y B) que activa el relé y con cuanto voltaje este se activa. Este voltaje y esta resistencia nos informan que magnitud debe de tener la señal que activará el relé y cuanta corriente se debe suministrar a éste.

La corriente se obtiene con ayuda de la Ley de Ohm:  $I = V / R$ . donde:

- $I$  es la corriente necesaria para activar el relé
- $V$  es el voltaje para activar el relé
- $R$  es la resistencia del bobinado del relé

Ventajas tiene un relé

- El Relé permite el control de un dispositivo a distancia. No se necesita estar junto al dispositivo para hacerlo funcionar.
- El Relé es activado con poca corriente, sin embargo puede activar grandes máquinas que consumen gran cantidad de corriente.
- Con una señal de control, puedo controlar varios relés a la vez.

FIGURA 2: Relé



FUENTE: [www.taringa.net](http://www.taringa.net)

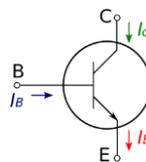
## Transistor

El transistor es un dispositivo electrónico semiconductor utilizado para entregar una señal de salida en respuesta a una señal de entrada. Cumple funciones de amplificador, oscilador, conmutador o rectificador. El término «transistor» es la contracción en inglés de transfer resistor («resistor de transferencia»). Actualmente se encuentran prácticamente en todos los aparatos electrónicos de uso diario: radios, televisores, reproductores de audio y video, relojes de cuarzo, computadoras, lámparas fluorescentes, tomógrafos, teléfonos celulares, entre otros.

## Transistor de unión bipolar

El transistor de unión bipolar (o BJT, por sus siglas del inglés bipolar junction transistor) se fabrica sobre un monocristal de material semiconductor como el germanio, el silicio o el arseniuro de galio, cuyas cualidades son intermedias entre las de un conductor eléctrico y las de un aislante. Sobre el sustrato de cristal se contaminan en forma muy controlada tres zonas sucesivas, N-P-N o P-N-P, dando lugar a dos uniones PN.

FIGURA 3: Diagrama de Transistor NPN



FUENTE: [es.wikipedia.org](http://es.wikipedia.org)

Las zonas N (en las que abundan portadores de carga Negativa) se obtienen contaminando el sustrato con átomos de elementos donantes de electrones, como el arsénico o el fósforo; mientras que las zonas P (se generan portadores de carga Positiva o «huecos») se logran contaminando con átomos aceptadores de electrones, como el indio, el aluminio o el galio.

Las tres zonas contaminadas, dan como resultado transistores PNP o NPN, donde la letra intermedia siempre corresponde a la región de la base, y las otras dos al emisor y al colector que, si bien son del mismo tipo y de signo contrario a la base, tienen diferente contaminación entre ellas (por lo general, el emisor está mucho más contaminado que el colector). El mecanismo que representa el comportamiento semiconductor dependerá de dichas contaminaciones, de la geometría asociada y del tipo de tecnología de contaminación (difusión gaseosa, epitaxial, etc.) y del comportamiento cuántico de la unión.

### Transistor de efecto de campo

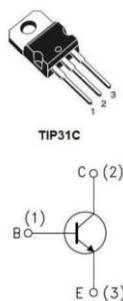
El transistor de efecto de campo de unión (JFET), fue el primer transistor de efecto de campo en la práctica. Lo forma una barra de material semiconductor de silicio de tipo N o P. En los terminales de la barra se establece un contacto óhmico, tenemos así un transistor de efecto de campo tipo N de la forma más básica. Si se difunden dos regiones P en una barra de material N y se conectan externamente entre sí, se producirá una puerta. A uno de estos contactos le llamaremos surtidor y al otro drenador. Aplicando tensión positiva entre el drenador y el surtidor y conectando la puerta al surtidor, estableceremos una corriente, a la que llamaremos corriente de drenador con polarización cero. Con un potencial negativo de puerta al que llamamos tensión de estrangulamiento, cesa la conducción en el canal.

El transistor de efecto de campo, o FET por sus siglas en inglés, que controla la corriente en función de una tensión; tienen alta impedancia de entrada.

- Transistor de efecto de campo de unión, JFET, construido mediante una unión PN.
- Transistor de efecto de campo de compuerta aislada, IGFET, en el que la compuerta se aísla del canal mediante un dieléctrico.
- Transistor de efecto de campo MOS, MOSFET, donde MOS significa Metal-Óxido-Semiconductor, en este caso la compuerta es metálica y está separada del canal semiconductor por una capa de óxido.

### Características y aplicaciones del transistor TIP 31

FIGURA 4: Transistor TIP31C



FUENTE: [www.electronicoscaldas.com](http://www.electronicoscaldas.com)

### Características

- Transistor NPN de media potencia

- $I_C$  max: 3 A
- $I_C$  pico max: 5 A
- $I_B$  max: 1 A
- $P_{TOT}$ : 40 W
- $V_{CEO}$ : 100 V,  $V_{CBO}$ : 100 V,  $V_{EBO}$ : 5 V
- $h_{FE}$ : 10 a 50 (@  $I_C=3$  A,  $V_{CE}=4$  V)
- Alta velocidad de suicheo
- Complementario: [TIP32](#)
- Encapsulado: TO-220

#### Aplicaciones:

- Conmutación y amplificación lineal
- Amplificadores de audio
- Circuitos de propósito general

#### Sensor ultrasonidos HC-SR04

El sensor de ultrasonidos se enmarca dentro de los sensores para medir distancias o superar obstáculos, entre otras posibles funciones.

En este caso vamos a utilizarlo para la medición de distancias. Esto lo consigue enviando un ultrasonido (inaudible para el oído humano por su alta frecuencia) a través de uno de la pareja de cilindros que compone el sensor (un transductor) y espera a que dicho sonido rebote sobre un objeto y vuelva, retorno captado por el otro cilindro.

Este sensor en concreto tiene un rango de distancias sensible entre 3cm y 3m con una precisión de 3mm.

FIGURA 5: Sensor de ultrasonido



FUENTE: [www.electronicoscaldas.com](http://www.electronicoscaldas.com)

#### ¿Qué recibimos en el sensor?

El tiempo que transcurre entre el envío y la recepción del ultrasonido.

¿Cómo vamos a traducir dicho tiempo en distancia?

Aprovechando que la velocidad de dicho ultrasonido en el aire es de valor 340 m/s, o 0,034 cm/microseg (ya que trabajaremos con centímetros y microsegundos). Para calcular la distancia, recordaremos que  $v=d/t$  (definición de velocidad: distancia recorrida en un determinado tiempo).

De la fórmula anterior despejamos  $d$ , obteniendo  $d=v \cdot t$ , siendo  $v$  la constante anteriormente citada y  $t$  el valor devuelto por el sensor a la placa Arduino.

También habrá que dividir el resultado entre 2 dado que el tiempo recibido es el tiempo de ida y vuelta.

#### Módulo bluetooth HC-05

El módulo de bluetooth HC-05 es el que ofrece una mejor relación de precio y característica, ya que es un módulo Maestro-Eslavo, quiere decir que además de recibir conexiones desde una PC o tablet, también

es capaz de generar conexiones hacia otros dispositivos bluetooth. Esto nos permite por ejemplo, conectar dos módulos de bluetooth y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos. En otro artículo posterior veremos cómo configurar dos módulos HC-05 para que se enlacen entre ellos y podamos transmitir información de un punto a otro.

FIGURA 6: Módulo bluetooth HC-05



FUENTE: [www.articulomercadolibre.com](http://www.articulomercadolibre.com)

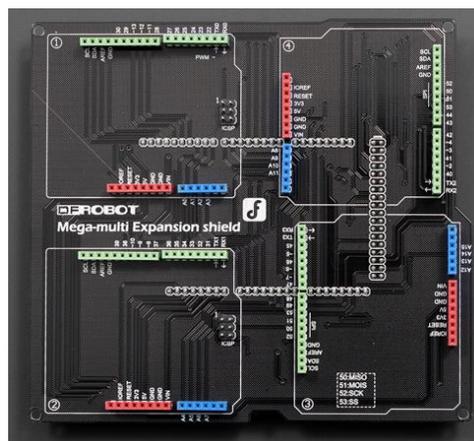
Módulo HC-05 montado en una tarjeta adaptadora compatible con zócalos Xbee

El HC-05 tiene un modo de comandos AT que debe activarse mediante un estado alto en el PIN34 mientras se enciende (o se resetea) el módulo. En las versiones para protoboard este pin viene marcado como “Key”. Una vez que estamos en el modo de comandos AT, podemos configurar el módulo bluetooth y cambiar parámetros como el nombre del dispositivo, password, modo maestro/esclavo, etc.

Para comunicarnos con el módulo y configurarlo, es necesario tener acceso al módulo mediante una interfaz serial. Podemos usar un arduino con un par de cables (aprovechando el puente USB-Serial del Arduino), un kit para XBee o un simple MAX3232 en el puerto serie de la PC. Para este artículo estaremos usando un módulo Bluetooth Bee Pro que comercializamos en nuestra tienda web, así como una interfaz USB-Serial con socket Xbee “Foca” que también vendemos. Este es el hardware que utilizamos en nuestro taller, pero también es posible hacer los ensayos con el módulo HC-05 suelto o en formato para insertar en el protoboard sin mayor problema.

Tarjeta multi shield

FIGURA 7: Mega multi shield



FUENTE: [www.articulomercadolibre.com](http://www.articulomercadolibre.com)

Si bien nuestra controladora Mega 2560/ADK es muy útil con sus múltiples pines IO, es frecuente encontrar problemas al querer montar diversos shields. Por ejemplo el Shield LCD montado sobre un shield de expansión con servomotores puede ser un verdadero dolor de cabeza. Si eres de los que quiere realmente aprovechar su Arduino Mega/ADK prueba este shield y verás cómo te facilita la vida.

Usar este shield significa que podrás montar 4 diferentes shields al mismo tiempo. Todos los pines del Mega se expanden entre sus 4 cuadrantes, eliminando conflictos y problemas de compatibilidad en los shields. Además, hace que tu proyecto se vea más profesional, ordenado y limpio.

#### Servomotor

Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

Un servomotor es un motor eléctrico que puede ser controlado tanto en velocidad como en posición.

Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos.

FIGURA 8: Servomotor



FUENTE: [www.electrosome.com](http://www.electrosome.com)

#### Diodo Led

Un led (del acrónimo inglés LED, light-emitting diode: 'diodo emisor de luz'; el plural aceptado por la RAE es leds) es un componente optoelectrónico pasivo y, más concretamente, un diodo que emite luz.

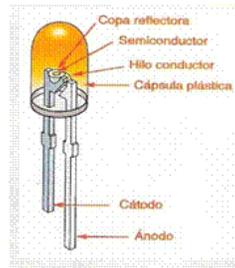
Los led se usan como indicadores en muchos dispositivos y en iluminación. Los primeros leds emitían luz roja de baja intensidad, pero los dispositivos actuales emiten luz de alto brillo en el espectro infrarrojo, visible y ultravioleta.

Debido a su capacidad de operación a altas frecuencias, son también útiles en tecnologías avanzadas de comunicaciones y control. Los leds infrarrojos también se usan en unidades de control remoto de muchos productos comerciales incluyendo equipos de audio y video.

Existen tres formas principales de conocer la polaridad de un led:

- La pata más larga siempre va a ser el ánodo.
- En el lado del cátodo, la base del led tiene un borde plano.
- Dentro del led, la plaqueta indica el ánodo. Se puede reconocer porque es más pequeña que el yunque, que indica el cátodo.

FIGURA 9: LED



FUENTE: [www.mografias.com](http://www.mografias.com)

### Resistencia

Se le denomina resistencia eléctrica a la igualdad de oposición que tienen los electrones al moverse a través de un conductor. La unidad de resistencia en el Sistema Internacional es el ohmio, que se representa con la letra griega omega ( $\Omega$ ), en honor al físico alemán Georg Ohm, quien descubrió el principio que ahora lleva su nombre.

FIGURA 10: Tabla de colores de Resistencias

TABLA DE COLORES DE LAS RESISTENCIAS				
	1°	2°	3°	4°
	---	0	1	10% 10% PLATA
MARRON	1	1	x10	5% ORO
ROJO	2	2	x100	
NARANJA	3	3	x1.000	1er DIGITO
AMARILLO	4	4	x10.000	2do DIGITO
VERDE	5	5	x100.000	3er DIGITO (Multiplicador)
AZUL	6	6	x1.000.000	4to DIGITO (Resistencia)
VIOLETA	7	7	10	
GRIS	8	8	ORO	
BLANCO	9	9		

FUENTE: [www.electronicas.com](http://www.electronicas.com)

### MATERIALES Y EQUIPO

- 1 barra angular de aluminio 30x30
- 4 ruedas de goma rígida de 15cm
- 2 ejes de diámetro 20mm y largo 30cm
- 1 motor de corriente continua de 12V (de un limpiaparabrisas)
- 1 batería de 12V (de moto)
- 1 Placa Arduino Mega
- 1 protoboard
- 5 pilas de 1,5V
- 1 módulo bluetooth
- 1 sensor ultrasonido
- 1 servo motor
- 1 sensor de gas y humo
- 2 transistores tip31
- 1 servo motor
- 1 tarjeta multi shield
- 2 relés
- 1 juego de cables de conexión y leds
- 10 resistencias de 100k $\Omega$

## PROCEDIMIENTO

Para poder describir el procedimiento en la construcción del vehículo de rastreo, se separara el mismo en sistemas, los cuales son:

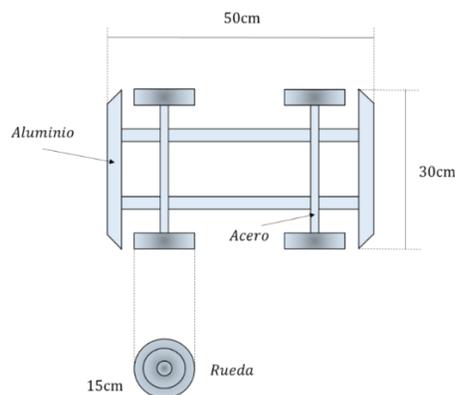
- Estructura y elementos mecánicos
- Circuito de Control
- Circuito Interfaz
- Circuito de Potencia
- Maniobra desde Android

### Estructura y elementos mecánicos

La estructura debe ser tal que soporte todos los elementos tanto electrónicos como mecánicos que requiere el vehículo.

La estructura es armada con aluminio, ejes de acero, ruedas de goma rígida y le acoplamos un motor de corriente continua de un limpiaparabrisas.

FIGURA 11: Estructura del vehículo



FUENTE: Elaboración Propia

### Circuito de Control

El circuito de control se hace con la tarjeta multi shield para incrementar las entradas y salidas en la placa de Arduino Mega. También se utiliza el módulo bluetooth para comunicación inalámbrica con el vehículo. Asimismo se usan sensores de ultrasonido, gas y humo, servomotor para rotación de llantas delanteras y cables de conexión.

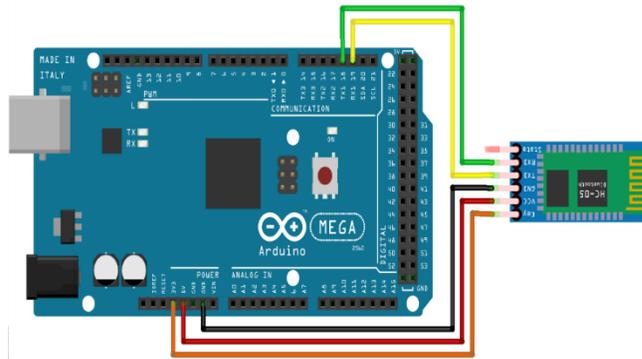
### Conexión del módulo bluetooth

Para la conexión del módulo bluetooth se necesita fundamentalmente 1 terminal de transferencia (TX), 1 terminal de recepción (RX), 1 alimentación de 5V (VCC) y 1 terminal a tierra (GND).

### Programación para el módulo bluetooth con Arduino

Una vez conectado el módulo bluetooth, ya existe una comunicación entre por vía bluetooth. Por ejemplo se puede tener la siguiente programación en Android:

FIGURA 12: Conexión modulo Bluetooth



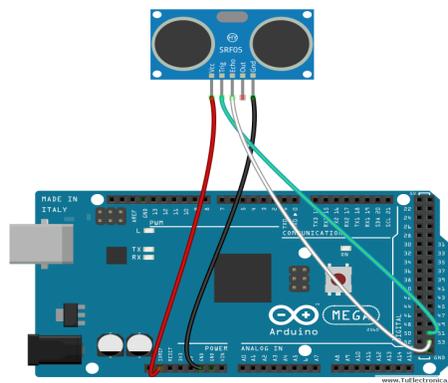
FUENTE: [www.blascarr.com/modulobluetooth](http://www.blascarr.com/modulobluetooth)

```
void setup()
Serial.begin(9600);// Conexión vía Bluetooth es la velocidad
void loop()
{
  char dato = Serial.read(); //lee datos de android
  if(dato=='i'){
    myservo.write(30); //Si la señal de android es b entonces cero grados 0º (izquierda)
  }
}
```

Conexión para el sensor de ultrasonido

Para el sensor de ultrasonido se utilizan 4 terminales, 1 terminal para alimentar con 5V (VCC), 1 terminal para tierra (GND), 1 terminal para enviar el pulso digital (trig) y 1 terminal para recibir el pulso de eco (echo).

FIGURA 13: Conexión modulo Bluetooth



FUENTE: [www.tuelectronica.es](http://www.tuelectronica.es)

Programación para el sensor de ultrasonido en Arduino

```
digitalWrite(10, LOW); // el ping trigger nivel bajo
delayMicroseconds(2); // Retardo para estabilizar la señal
digitalWrite(10, HIGH); // genera el pulso de trigger
delayMicroseconds(10); // duración del pulso de disparo
digitalWrite(10, LOW);
duracion = pulseIn(8, HIGH); // lee la longitud de un pulso de entrada
distancia = (duracion/2)/29; // calcula la distancia efectiva en centímetros
delay (10);
```

```

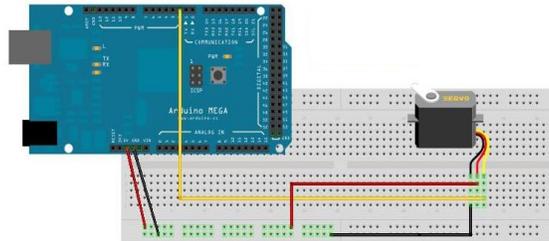
if (distancia <= 30 && distancia >=2){ // si la distancia es menor a 15cm
digitalWrite(5,HIGH);
digitalWrite(6, HIGH);// enciende el led
delay(10);
if(distancia >30){
digitalWrite(5,LOW);
digitalWrite(6, LOW);// enciende el led

```

### Conexión para el servomotor

El servomotor tiene tres terminales para su conexión, 1 terminal para 5V (VCC), 1 terminal para tierra (GND) y 1 pin PWM para regular su posición.

FIGURA 14: Conexión para el servomotor



FUENTE: [www.BicoGeek.com](http://www.BicoGeek.com)

### Programación para el servomotor en Arduino

```

include <Servo.h> //llamando función servo
Servo myservo; // función del servomotor que nos permite controlar el servo
void setup()
{
  myservo.attach(9); // Pin de comunicación del servomotor
}
void loop()
{
  myservo.write(90); // Si la señal de android es c entonces 90° (centro)

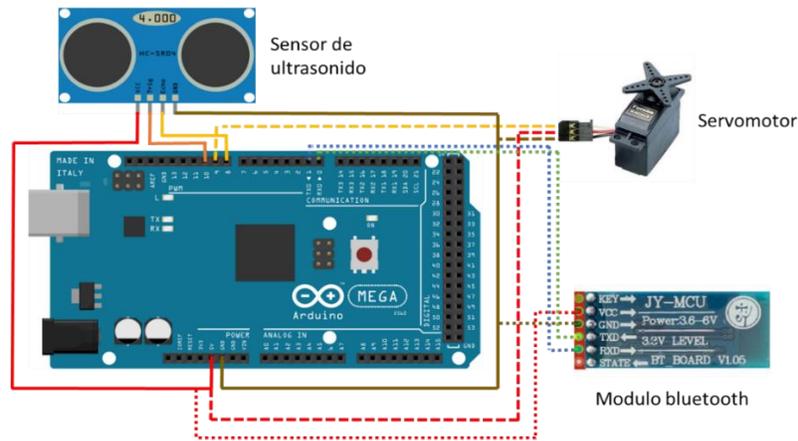
  myservo.write(150); //Si la señal de android es d entonces 180° (derecha)
  myservo.write(30); //Si la señal de android es b entonces cero grados 0° (izquierda)
}

```

### Conexión general para el circuito de control

La conexión general para el circuito de control se describe en la siguiente figura:

FIGURA 15: Conexión general para el circuito de control



FUENTE: Elaboración propia

### Programación general en Arduino

A continuación se muestra el código de Arduino para la programación que engloba todos los elementos que componen al circuito de control del vehículo de rastreo.

```
//PROGRAMACION DE TRONIC
//para controlar la dirección
#include <Servo.h> //llaman a la función servo
Servo myservo; // función del servomotor
//variables para el sensor de ultrasonido
int duracion;
int distancia;
void setup()
{
  myservo.attach(9); // Pin de comunicación del servomotor
  Serial.begin(9600); // comunicación inalámbrica vía Bluetooth
  pinMode(13, OUTPUT); //pin de avance
  pinMode(12, OUTPUT); //pin de retroceso
  pinMode(11, OUTPUT); //bocina
  pinMode(9, OUTPUT); //servomotor
  pinMode(10, OUTPUT); //pin de salida trig (ultrasonido)
  pinMode(8, INPUT); //pin de entrada echo (ultrasonido)
  pinMode(6, OUTPUT); //luz de señalización
  pinMode(5, OUTPUT); //luz de señalización
}
void loop()
{
  //lectura de datos enviados por android
  char dato = Serial.read(); //lee datos de android, asigna el valor a la variable dato
  //control de dirección
  if(dato=='i'){
    myservo.write(30); //Si la señal de android es i entonces 30º(izquierda)
  }
  if(dato=='d'){
    myservo.write(150); //Si la señal de android es d entonces 150º (derecha)
  }
}
```

```

}
if(dato=='c'){
myservo.write(90);// Si la señal de android es c entonces 90° (centro)
}
//avance y retroceso
if(dato=='a'){
digitalWrite(13, HIGH); //coloca el pin 13 en alto
digitalWrite(12,LOW); //coloca el pin 12 en bajo
}
if(dato=='r'){
digitalWrite(12, HIGH); //coloca el pin 12 en alto
digitalWrite(13,LOW); //coloca el pin 13 en bajo
}
}
//para todas las acciones
if(dato=='s'){
digitalWrite(13, LOW);
digitalWrite(12, LOW);
digitalWrite(11, LOW);
digitalWrite(10, LOW);
digitalWrite(8, LOW);
digitalWrite(6, LOW);
digitalWrite(5, LOW);
}
}
//sensor de ultrasonido
digitalWrite(8, HIGH); //puede recibir el pulso en el pin echo
digitalWrite(10, HIGH); // genera el pulso de trigg
delayMicroseconds(10); // duración del pulso de disparo
digitalWrite(10, LOW); // apaga el envío del pulso de trigg
duracion = pulseIn(8, HIGH); // lee el tiempo del echo
distancia = (duracion/2)/29; // calcula la distancia en centímetros
delay (10);
if (distancia <= 30 && distancia >=2){ // si la distancia esta entre 2 y 30 cm
digitalWrite(5,HIGH); //enciende el led
digitalWrite(6,HIGH);// enciende el led
delay(10);
if(distancia >30){
digitalWrite(5,LOW);
digitalWrite(6,LOW);// apaga el led
}
}
else
{
digitalWrite(5,LOW);
digitalWrite(6,LOW);
}
}
}
//activacion de bocina
if(dato=='b'){

```

```

digitalWrite(11, HIGH);
delay(2000);
digitalWrite(11, LOW);
}
//luces de señalizacion
if(dato=='z'){
for (int k = 0; k < 5; k++){
digitalWrite(6, HIGH);
digitalWrite(5, HIGH);
delay(500);
digitalWrite(6, LOW);
digitalWrite(5, LOW);
delay(500);
}
}}

```

La tarjeta Arduino Mega estará conectada mediante la placa Expansión SHIELD, donde dará más conexiones al Arduino Mega.

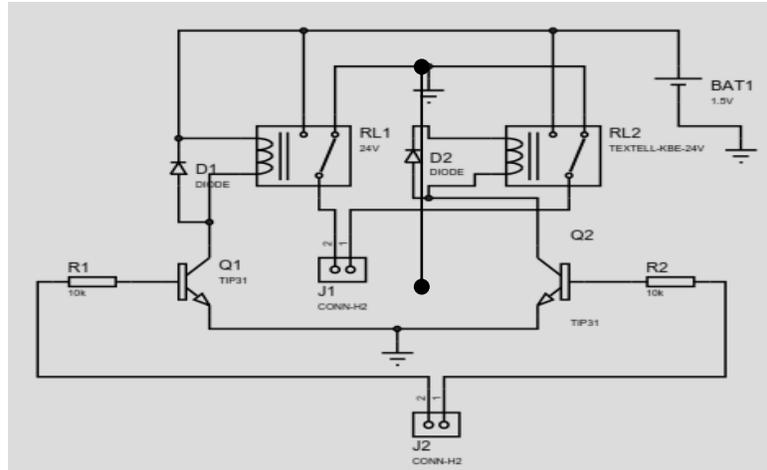
El detalle de los pines se muestra a continuación:

- 8: Conexión al sensor de ultrasonido
- 9: Conexión al servo motor
- 10: Conexión al sensor de ultrasonido
- GND: Conexión tierra
- VCC : Conexión fuente de 7.5 [volt]
- TX3: Conexión al módulo bluetooth
- RX3: Conexión al módulo bluetooth
- 11: Conectado a la placa de interface del motor de tracción.
- 12: Conectado a la placa de interface del motor de tracción.

#### Circuito Interfaz

El circuito interfaz logra pasar del circuito de control al circuito de potencia. Está compuesto por 2 transistores (TIP31) y 2 relés, y es equivalente en funcionamiento a un puente H que se arma con 4 relés. Un pulso (5V) va por el pin 11 para la base del transistor, sale del colector y se conecta a la bobina del relé. Otra conexión idéntica se hace para el pin 12, se unen los mismos como se muestra en la figura y se lograra con ello que el motor gire en un sentido o en el otro.

FIGURA 16: Conexión general para el circuito interfaz



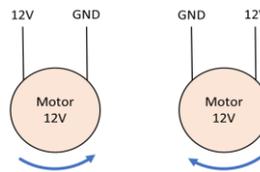
FUENTE: Elaboración propia

Circuito de Potencia

El circuito de potencia básicamente está dado por el funcionamiento del motor de corriente continua de 12V que es de un limpiaparabrisas.

El motor estará conectado en los relés, 1 terminal en cada relé, con lo que se logra poner en marcha el motor en un sentido o en el otro.

FIGURA 17: Inversión de giro para el motor

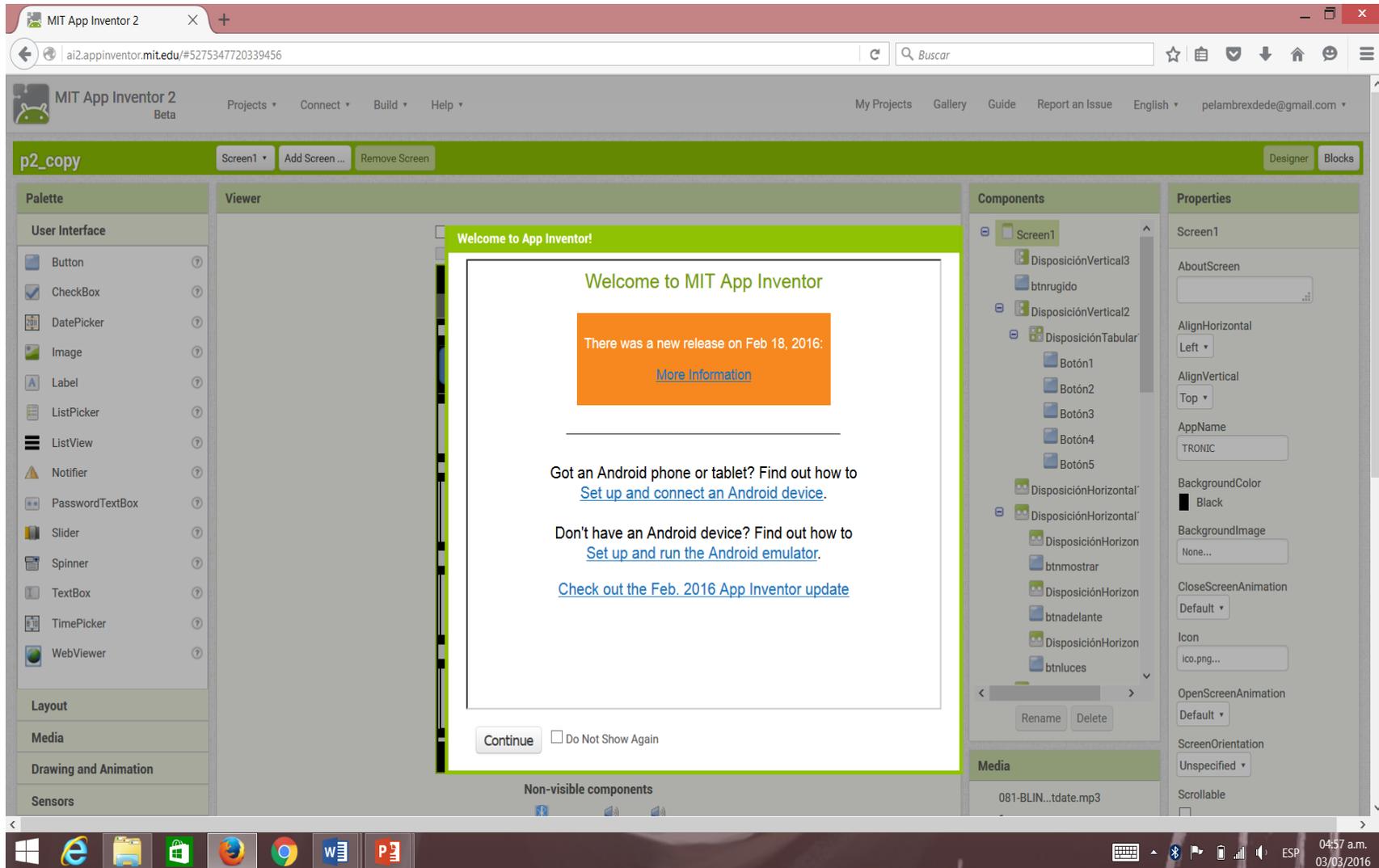


FUENTE: Elaboración propia

Maniobra desde Android

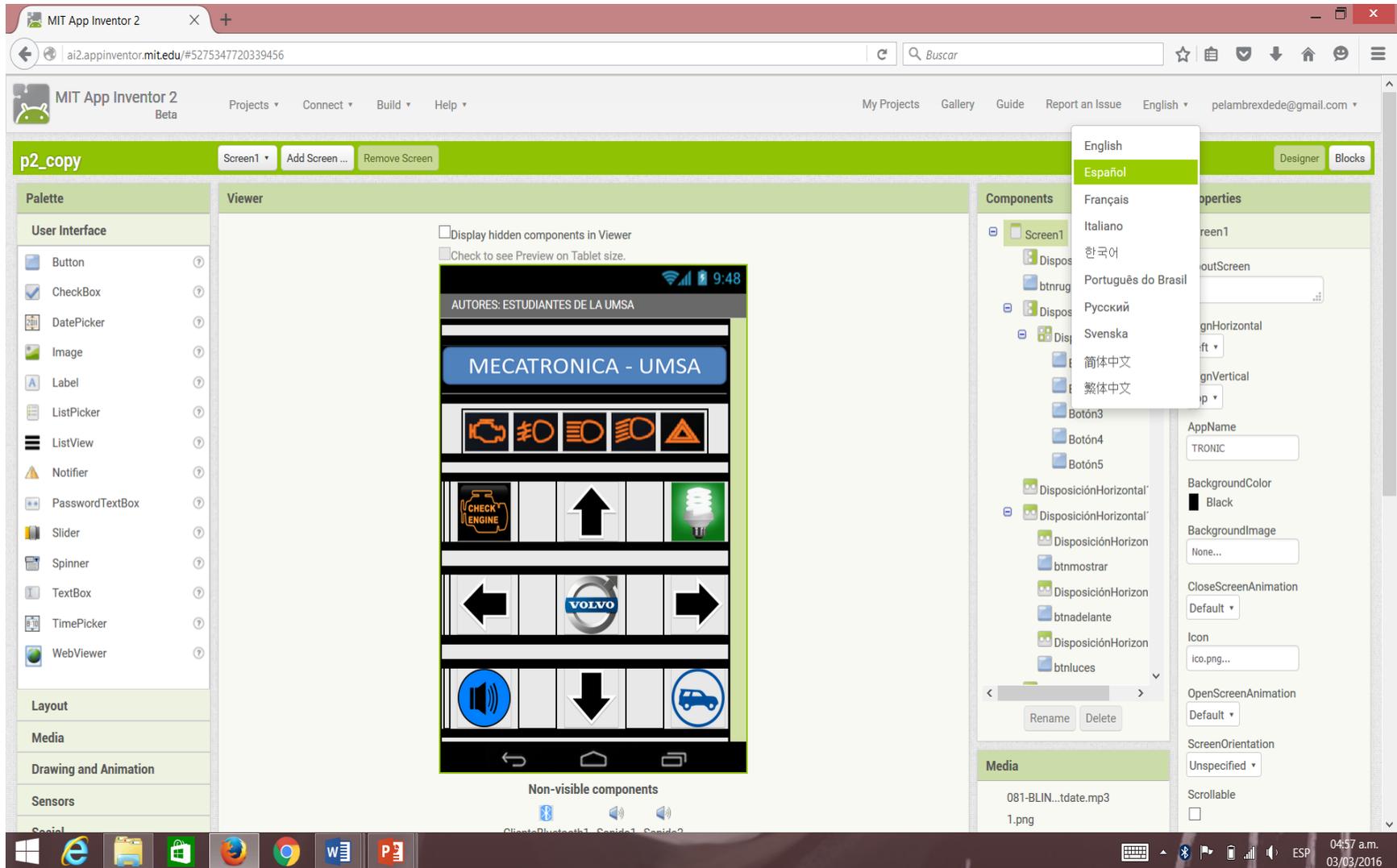
Para la maniobra de funcionamiento, utilizamos la tecnología Android. Creamos una aplicación para celulares, Tablet, etc. por medio de la plataforma de la MIT que es: <http://ai2.appinventor.mit.edu>.

FIGURA 18: App Inventor 2



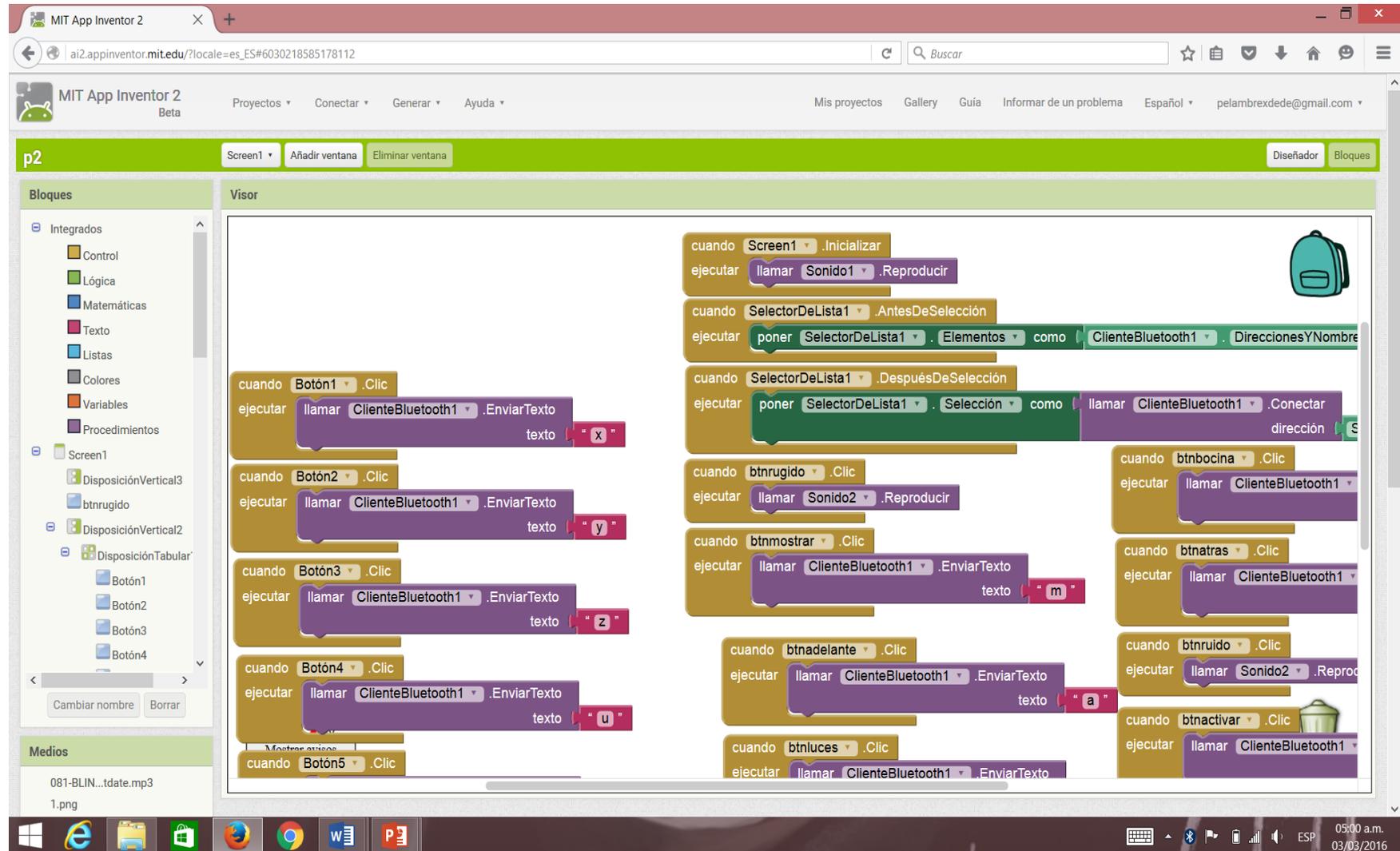
FUENTE: Elaboración propia

FIGURA 19: Diseño en App Inventor 2



FUENTE: Elaboración propia

FIGURA 20: Programación en bloques para App Inventor 2



FUENTE: Elaboración propia

La aplicación que se rescata de la plataforma de la MIT se guarda en un archivo .apk el cual se instala en tecnología Android, posteriormente se conecta por línea bluetooth con el módulo en el Arduino para el control sea inalámbrico.

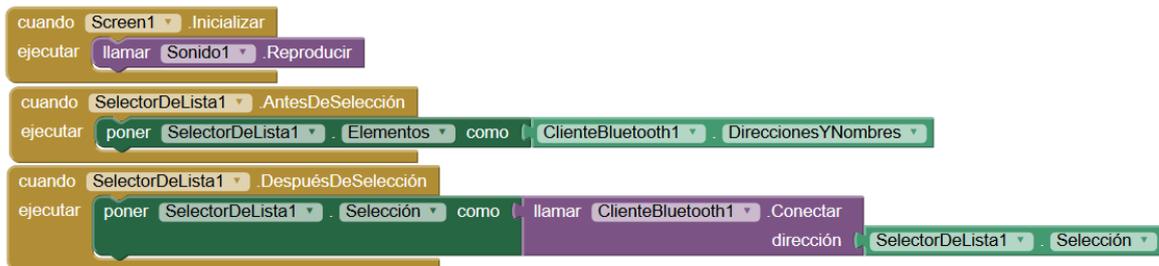
FIGURA 21: Interfaz de android



FUENTE: Elaboración propia

En la siguiente figura se muestran los bloques de programación que corresponden al emparejamiento mediante bluetooth entre el dispositivo Android y el Arduino.

FIGURA 22: Bloques para conectarse mediante bluetooth



FUENTE: Elaboración propia

**OBSERVACIONES**

Las observaciones más relevantes encontradas en el proyecto son:

Si bien existe un módulo de expansión para Arduino Mega, y éste permite la incorporación de por ejemplo un LCD (display) u otros elementos como sensores, se debe tomar en cuenta que existe una caída de tensión en las salidas del Arduino lo que hace que éste sea menos eficiente. Debido a las conexiones que se tienen en el sistema de control puede existir riesgo de que las terminales se desconecten puesto que las mismas están conectadas con conectores macho – macho, que son susceptibles a desconexión.

**CONCLUSIONES**

- Se ha diseñado la estructura mecánica del vehículo de rastreo.
- Se ha realizado el circuito de control para la obtención de datos de un espacio físico.
- Se ha realizado el circuito de potencia para el funcionamiento del motor del vehículo de rastreo.
- Se ha realizado la aplicación interactiva para la maniobra del funcionamiento del vehículo.

**BIBLIOGRAFÍA**

[www.arduino.cc](http://www.arduino.cc)  
[www.electronicas.com](http://www.electronicas.com)  
[www.BicoGeek.com](http://www.BicoGeek.com)  
[www.electronicoscaldas.com](http://www.electronicoscaldas.com)

## PRÁCTICA 15-B

### Comunicación en protocolo MODBUS entre Arduino y HMI

**OBJETIVO:** Realizar la conexión y comunicación del HMI marca SANKORE (maestro) y Arduino (esclavo) mediante el protocolo MODBUS.

**COMPETENCIA:** Aplica conocimiento de comunicación industrial MODBUS RTU a nivel hardware y software.

#### FUNDAMENTO TEÓRICO:

##### Tramas Modbus

Puede que en algún momento se tenga que ir un paso más allá que simplemente leer datos desde un SCADA, ya sea para diagnosticar un problema o para verificar la red, y analizar los datos crudos que vienen de un equipo Modbus. Por eso, en esta entrada vamos a analizar cómo se conforman las tramas Modbus.

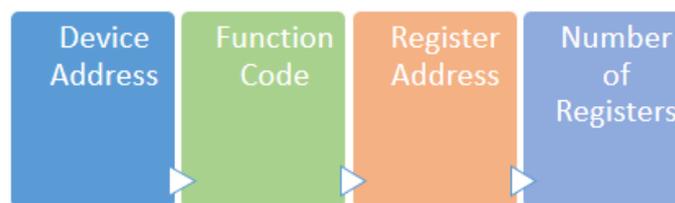
¿Qué es una trama?

En protocolos de comunicación, las tramas son unidades de datos que envía un dispositivo hacia otro/s, y que, en conjunto, conforman un mensaje. Por lo tanto, y a modo de simplificación, podría decirse que en definitiva la trama es el conjunto de bits que conforman un mensaje.

El formato de los mensajes depende, además del protocolo en sí, de la capa física con la que trabaje el protocolo. Por lo tanto, no se puede esperar que la trama de mensajes de Modbus RTU y Modbus TCP sean exactamente igual, aunque en ambos casos conservan similitudes.

¿Qué campos conforman las tramas Modbus?

A continuación, se debe identificar qué campos identifican las tramas Modbus:



- Device Address: Dirección del dispositivo hacia donde se dirige el emisor.
- Function Code: Función que necesita realizar el emisor.
- Register Address: Dirección a la que queremos acceder.
- Number of Registers: Número de registros sobre los que queremos realizar la función.

Por ejemplo, para leer dos registros del Holding Register 40212 de un Slave cuya dirección es 15, el comando que el Master debería enviar sería 15 -03-0212-02.

Luego, puede haber más campos, por ejemplo, normalmente al final del mensaje nos encontramos con el Error Check.

**Error Check:** El CRC es una verificación del mensaje, para comprobar que éste no tiene malformaciones. Esta verificación se corresponde con el checksum de la trama.

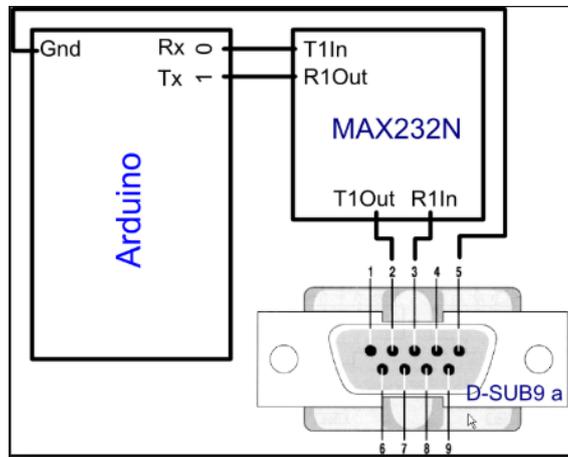
Por supuesto, este análisis no completa la trama completa del protocolo completo – RTU, TCP, etc – pero sí son los campos de cualquier protocolo Modbus, sea cuál sea el medio.

**HARDWARE**

Para la conexión se requiere una nivelación de los niveles de voltaje que maneja el puerto COM1 del HMI (12V) y Arduino (5V), por lo que se añade el módulo MAX232 para tal fin.



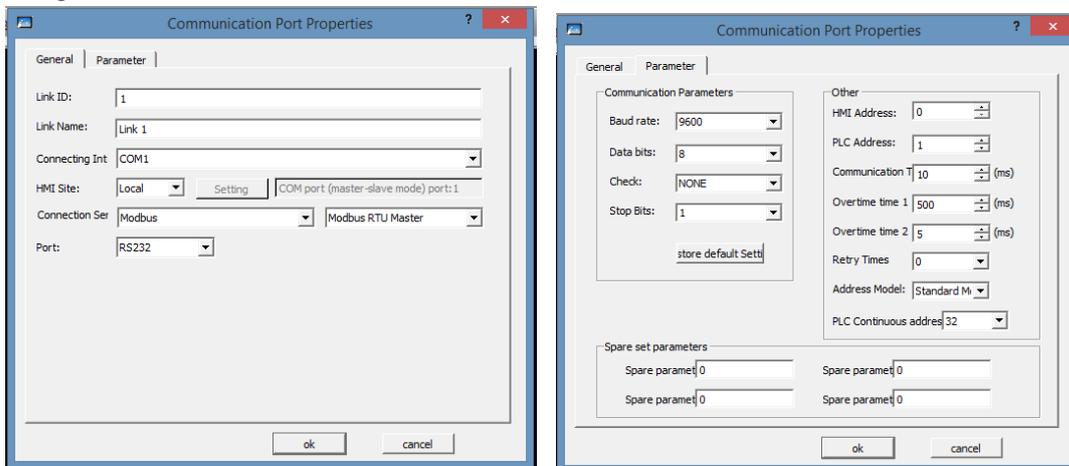
La conexión de Arduino y el módulo MAX232 se muestra en la siguiente figura.



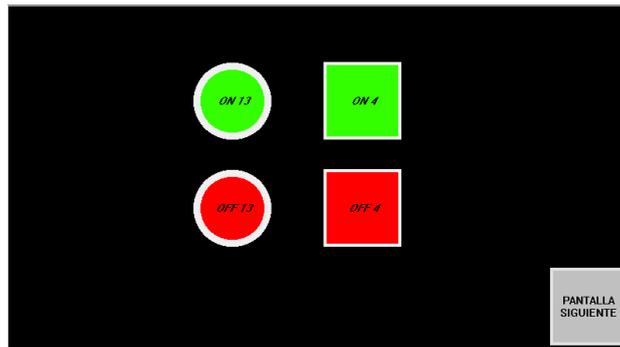
**SOFTWARE**

Para implementar la comunicación se deben configurar tanto el HMI como Arduino realizando los siguientes pasos.

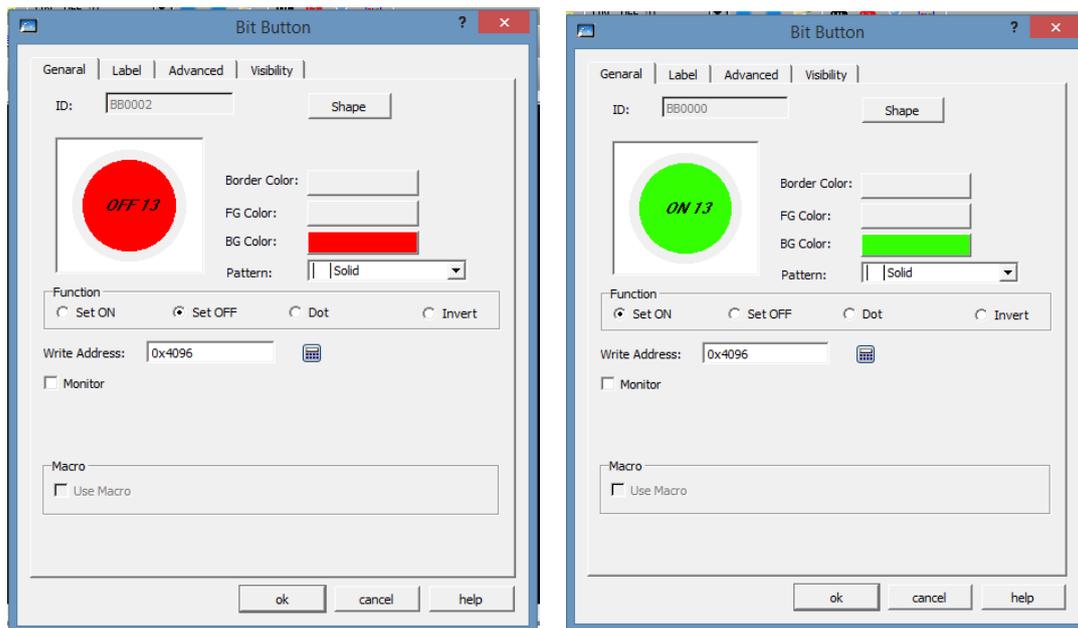
Configuración inicial del enlace



La pantalla uno realiza la conexión a nivel bit



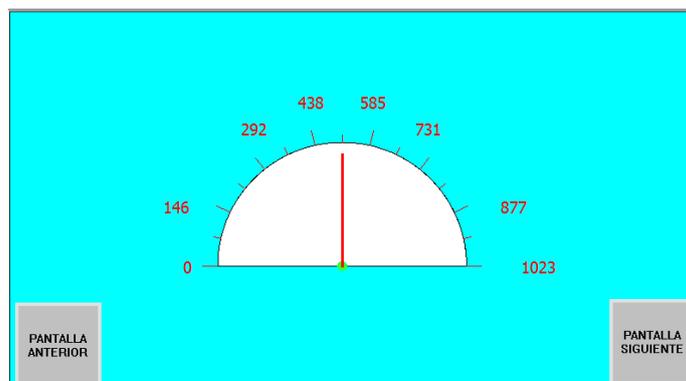
Dentro la configuración (dicha configuración es similar para “ON4” y “OFF4”) de “ON13” y “OFF13” vemos lo siguiente.



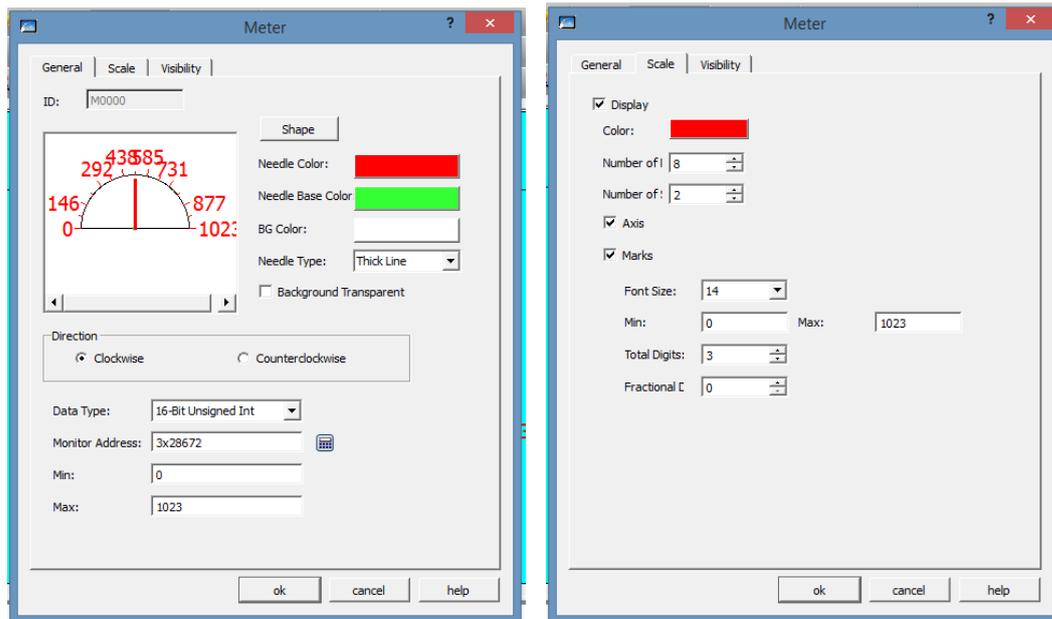
Que corresponden a las siguientes líneas de código en Arduino.

```
digitalWrite(13, getBit(_M, 0));
digitalWrite(4, getBit(_M, 1));
```

La pantalla 2 que corresponde a una lectura analógica es



La configuración realizada para esta pantalla es



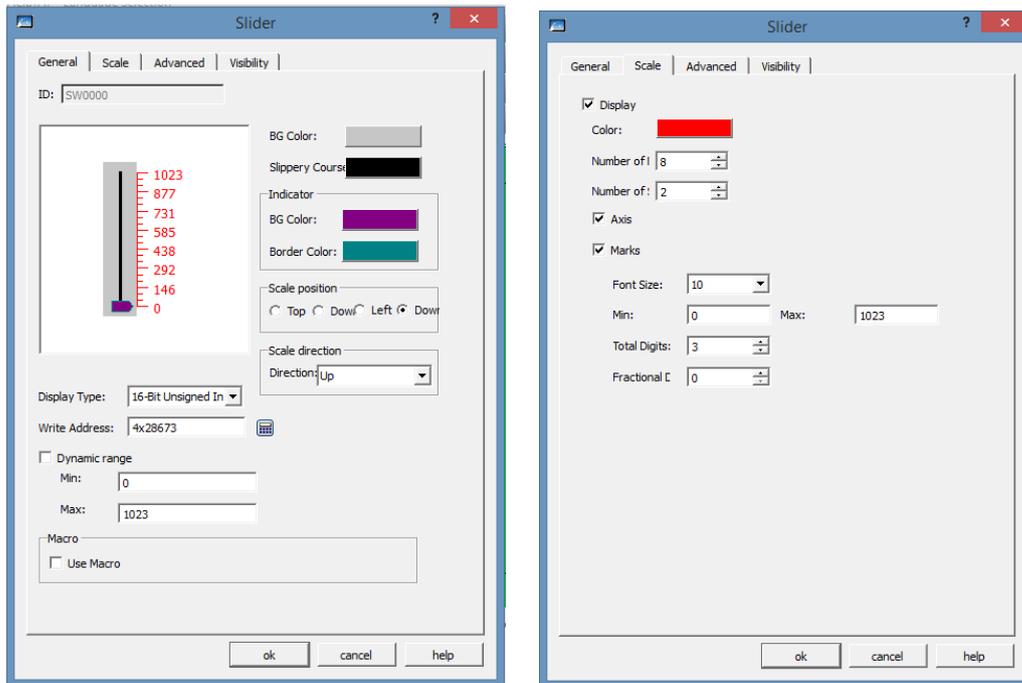
Correspondiente a la siguiente línea de código

```
_D[0]=analogRead(A0);
```

Finalmente para establecer la escritura analógica (pseudoanalógica en Arduino por medio del PWM) se establece una tercera pantalla



Cuya configuración es



Correspondiente a la siguiente línea de código

```
analogWrite(9, _D[1]/4);
```

Se debe hacer hincapié en la importación de la librería ModbusRTUSlave.h dentro del IDE de Arduino para la correcta compilación del código mostrado a continuación.

```
#include <ModbusRTUSlave.h>
u16 _D[100];
u8 _M[50];
ModbusRTUSlave rtu(1, &Serial);
void setup() {
  pinMode(13, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(9, OUTPUT);
  //Banco de memoria para RTU a nivel WORD
  rtu.addWordArea(0x7000, _D, 100);
  //Banco de memoria para RTU a nivel BIT
  rtu.addBitArea(0x1000, _M, 50);
  //configuración inicial del puerto RTU
  rtu.begin(9600);
}
```

```
void loop() {  
  //Actualización del proceso RTU  
  rtu.process();  
  //Escritura a nivel BIT por parte del HMI sobre Arduino  
  digitalWrite(13, getBit(_M,0));  
  digitalWrite(4, getBit(_M,1));  
  //Lectura analógica (lectura analógica por parte del HMI)  
  _D[0]=analogRead(A0);  
  //Escritura PWM por parte del HMI  
  analogWrite(9, _D[1]/4);  
}
```

## CONCLUSIONES

La librería Modbus funciona muy bien respetando las velocidades, recomendado utilizarla a 9600 bauds, no significa que a 115200 no funcione, sino más seguridad en las transmisiones.

La librería fue modificada para funcionar con el Serial1, en este caso con el Arduino UNO, de requerir que funcione con el puerto por defecto de Arduino editar la librería y reemplazar en todos los casos Serial1 con Serial1 y debería funcionar normalmente.

## RECOMENDACIONES

El uso de la librería de Arduino como esclavo Modbus en el caso de aplicaciones básicas dé automáticamente, como lectura de periféricos I/O y posterior envío de datos a un maestro ya sea PLC, PAC HMI, SCADA opc o dispositivos IoT, a medida de avanzar en pruebas realizando integraciones con PLC, HMI, SCADA y IoT (Node-RED).

## GUÍA BÁSICA DE AUTÓMATA PROGRAMABLE PC WORX

**OBJETIVO:** Introducción a la parametrización y programación del autómata programable PC WORX del tablero didáctico del IIME.

**COMPETENCIA:** Establece comunicación entre la PC y el procesador del autómata y módulos de expansión, realiza un programa básico de variables booleanas.

**FUNDAMENTO TEÓRICO:**

Procesador: ILC 130 ETH 01/3.90 – Controlador Inline para redes Ethernet con 8 entradas digitales y 4 salidas digitales, de la familia de controladores altamente modular de los controladores Inline de Phoenix Contact. A través de su interfaz Ethernet integrada, puede parametrizarse y programarse mediante el software de automatización PC Worx según IEC 61131, e intercambia paralelamente datos con servidores OPC y se comunica con participantes aptos para TCP/IP.

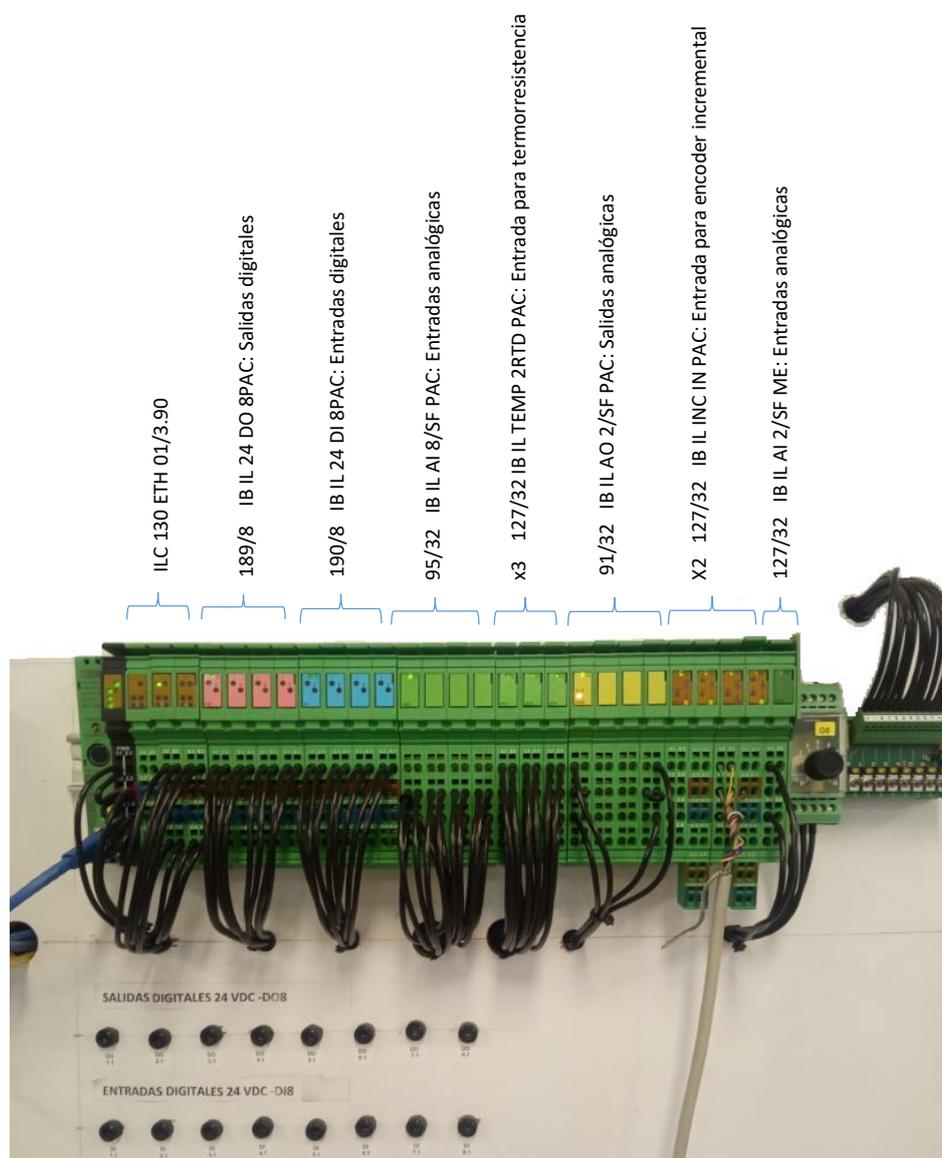


Figura 1: Foto del procesador principal y módulos de expansión del tablero didáctico del IIME

Programación

1er PASO: Verificar que el conector RJ45 de Ethernet esté adecuadamente conectado al “switch” y verificar que exista buena comunicación mediante el “Símbolo del sistema”. En Windows 8 y Windows 10, el símbolo del sistema se abre siguiendo las instrucciones:

Pulsamos la tecla Windows. Se nos abrirá la pestaña de Inicio.

Sin entrar a ningún programa ni hacer clic en ningún punto de la pantalla, tecleamos una de dos opciones:

«Cmd»

«Símbolo del sistema»

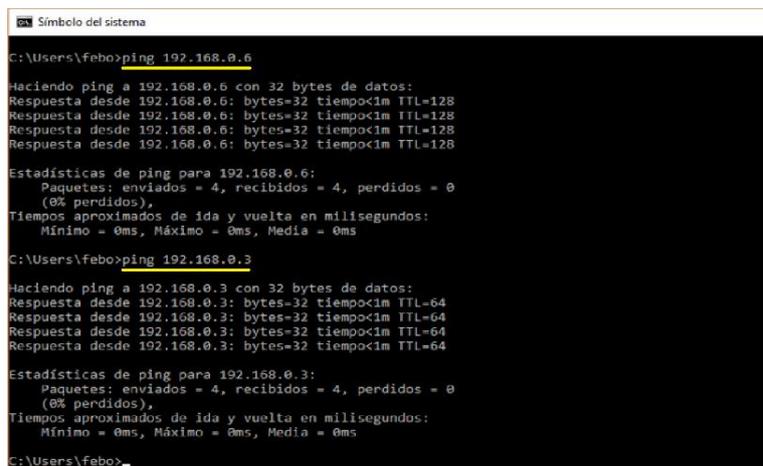


Figura 2: Símbolo del sistema

En la figura 2, se muestra la respuesta de “cmd” cuando hay buena comunicación al escribir a la derecha del prompt “>” subrayado con amarillo la dirección IP de la subred local. En este caso particular, con las IP correspondientes a la PC y al procesador de PC WORX. Los registros de IP pueden variar de acuerdo a cómo se hayan definido éstas en la PC y el procesador.

La dirección IP puede definirse en la PC, mediante el “panel de control”.

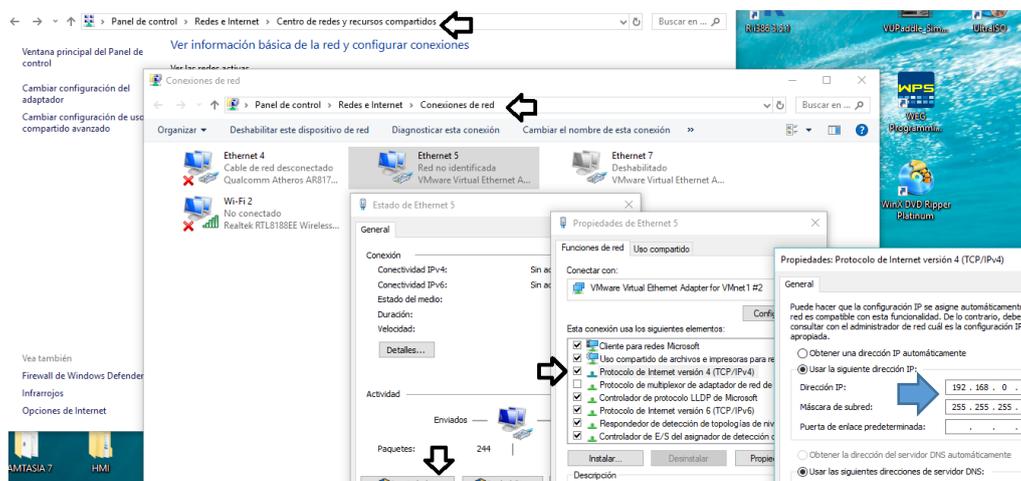


Figura 3: Secuencia de pasos para parametrizar comunicación Ethernet

En la figura 3, se establece una dirección IP de sub red local 192.168.0.4

INICIANDO UN NUEVO PROYECTO

Abrir el programa PC WORX 6.30.601 

Nuevo proyecto

Seleccionar: ILC 130 ETH Rev. > 01/3.90. Correspondiente al modelo del laboratorio.

Establecimiento de la conexión Ethernet verificando el IP del procesador ILC 130.

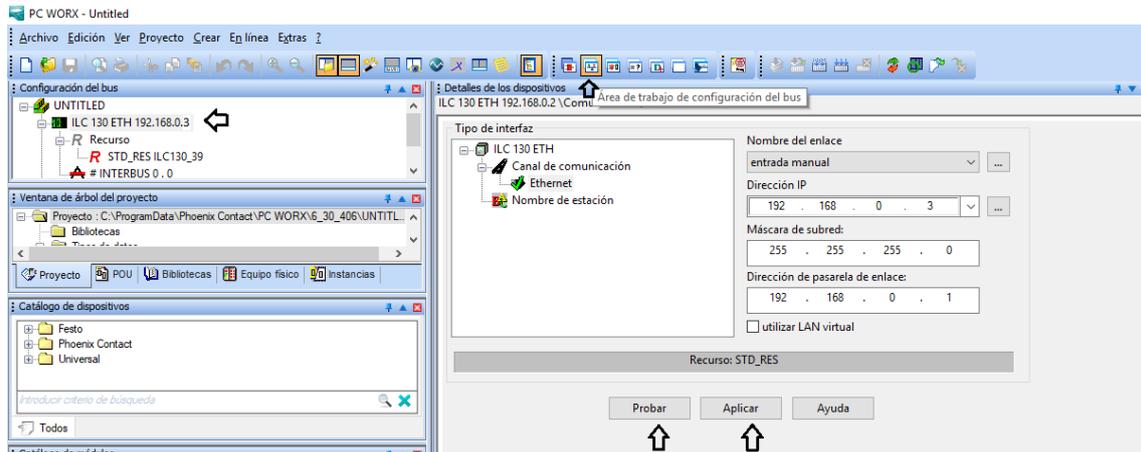


Figura 4: Reconocimiento del IP del procesador

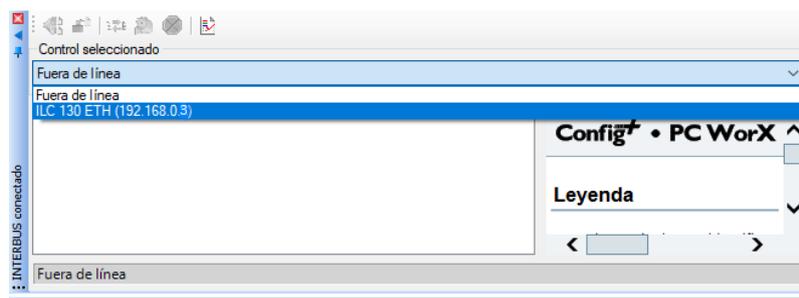


Figura 5: Conexión en línea PC - procesador, mediante ventana INTERBUS conectado

Una vez establecida la conexión PC – procesador ILC 130, mediante la misma ventana conectar todos los módulos conectados mediante INTERBUS, en el caso del equipamiento del laboratorio del IIME, son 10 módulos, use el lado derecho del ratón.

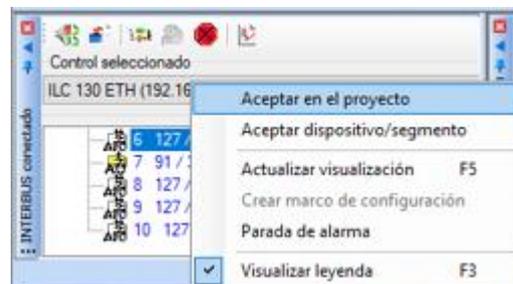


Figura 6: Seleccionar uno a uno, los 10 módulos instalados en el tablero del laboratorio

- 1 189/8 IB IL 24 DO 8PAC
- 2 190/8 IB IL 24 DI 8PAC
- 3 95/32 IB IL AI 8/SF PAC

- 4 127/32 IB IL TEMP 2RTD PAC
- 5 127/32 IB IL TEMP 2RTD PAC
- 6 127/32 IB IL TEMP 2RTD PAC
- 7 91/32 IB IL AO 2/SF PAC
- 8 127/32 IB IL INC IN PAC
- 9 127/32 IB IL INC IN PAC
- 10 127/32 IB IL AI 2/SF ME

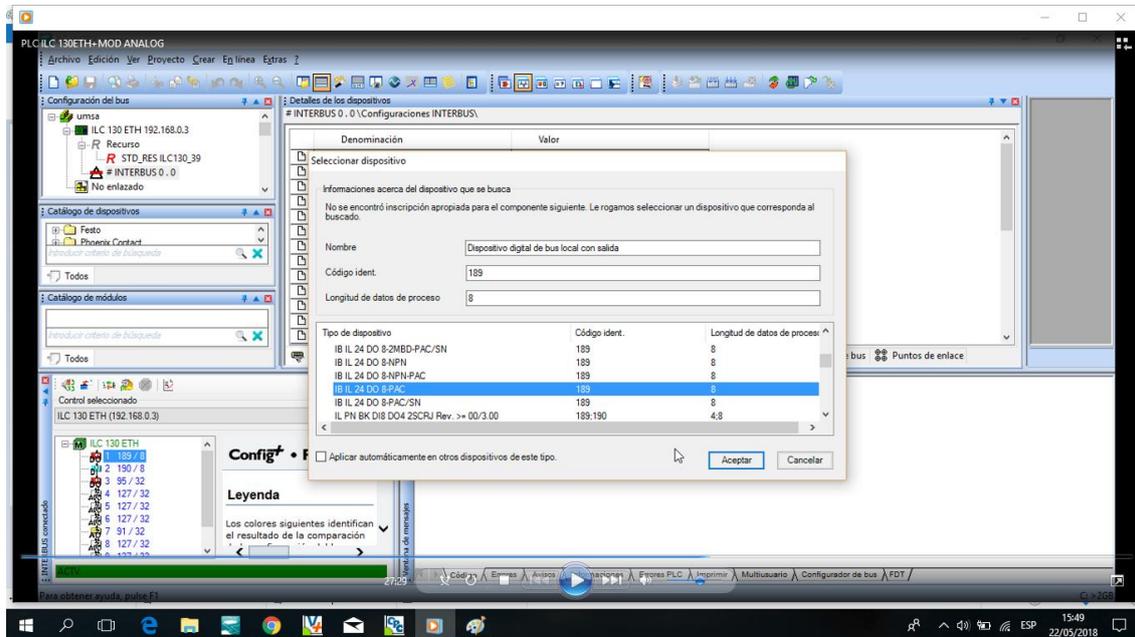


Figura 7: Selección de los módulos de expansión

### INICIANDO A PROGRAMAR

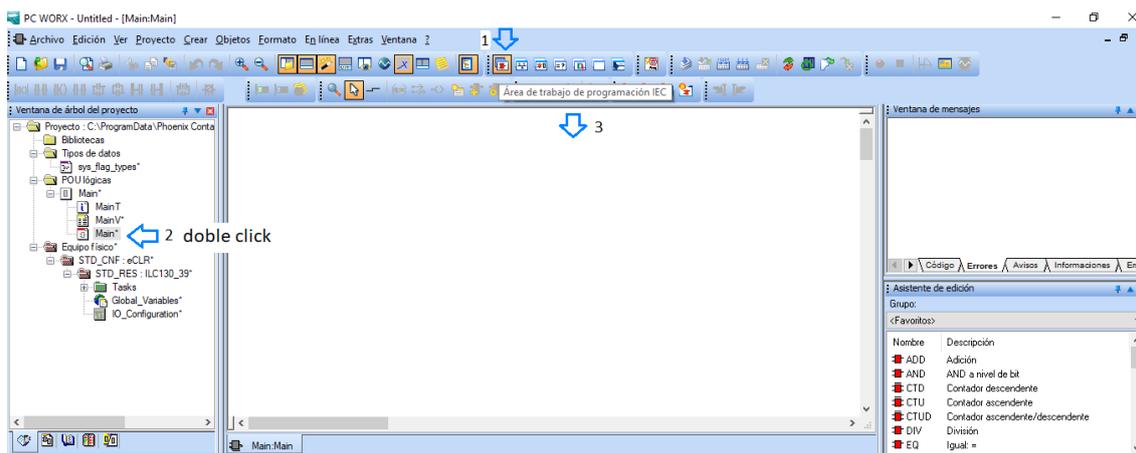


Figura 8: Ingresando al área de trabajo de programación

Click en el menú de la barra superior “Área de trabajo de programación IEC”, luego doble click en “Main\*” de la “Ventana de árbol del proyecto” y, para empezar a programar, Click con el lado derecho del ratón sobre el “área de trabajo de programación” para escoger Variable y se despliegue la siguiente figura:

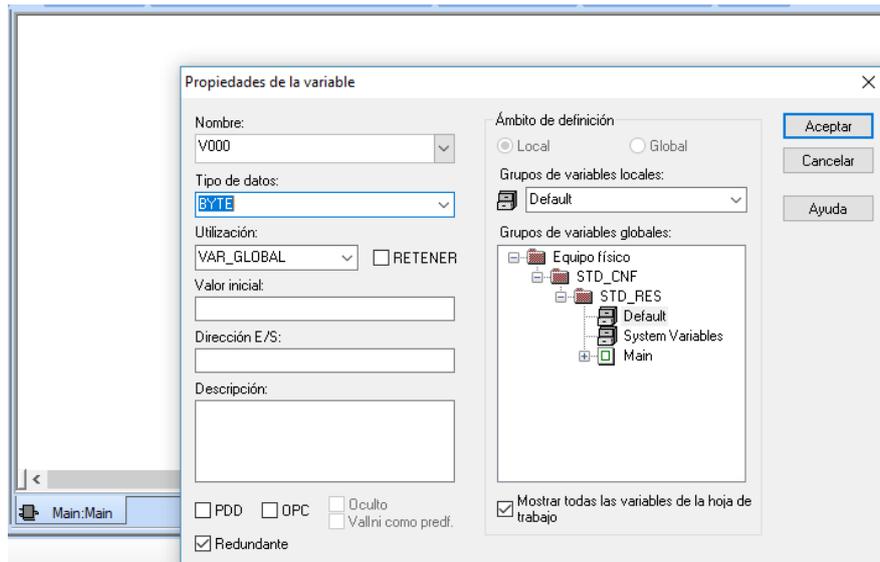


Figura 9: Despliegue de propiedades de la variable

Deben especificarse propiedades de la variable V000 por ejemplo, en “Tipo de dato” cambiar BYTE por BOOL si se va a hacer un programa sencillo de entradas booleanas. Demás propiedades se ajustarán de acuerdo el programa a realizar.

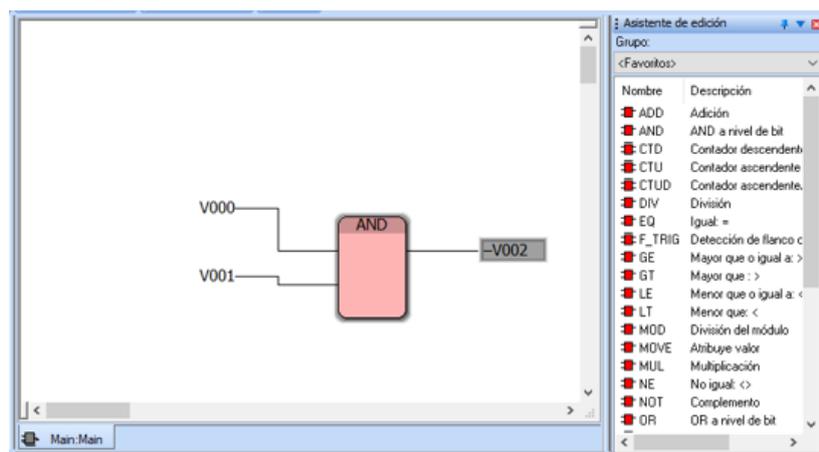


Figura 10: Programa básico de prueba

En la Figura 10, se muestra un programa básico de entradas y salida tipo BOOL y empleando el bloque AND extraído de la ventana “Asistente de edición” que muestra diagramas de funciones de bloque.

El siguiente paso consiste en asociar las variables V000, V001 y V002 con entradas y salidas de los módulos de expansión que se utilizarán.

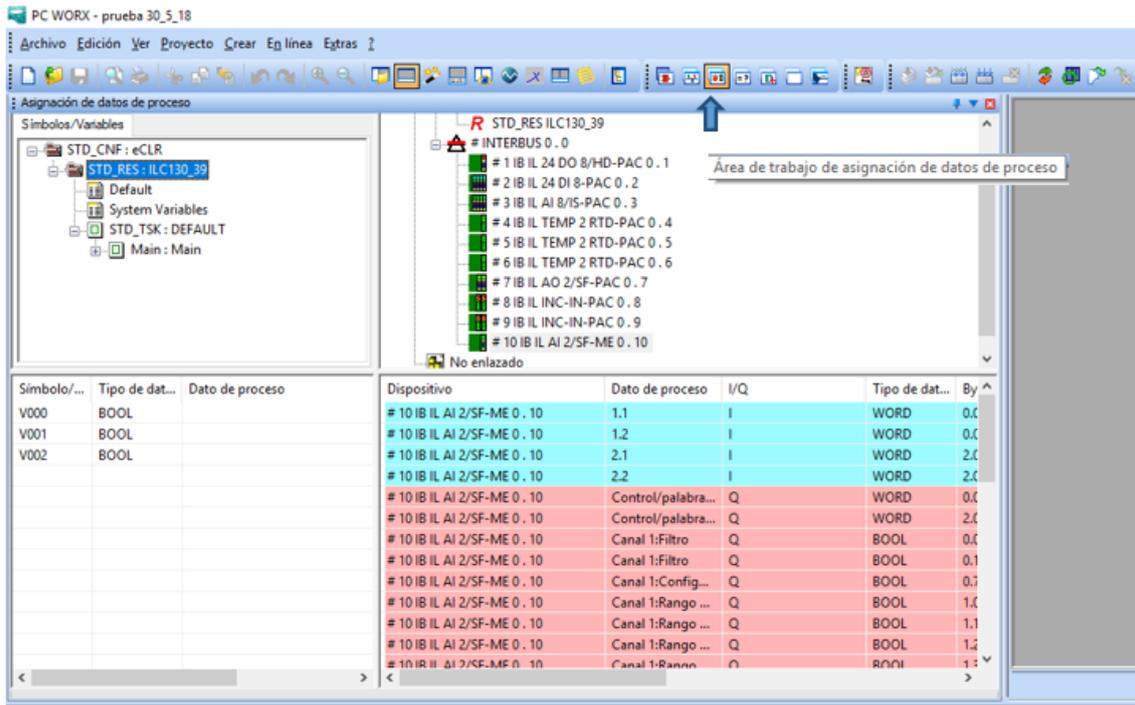


Figura 11: Ingreso al Área de trabajo de asignación de datos de proceso

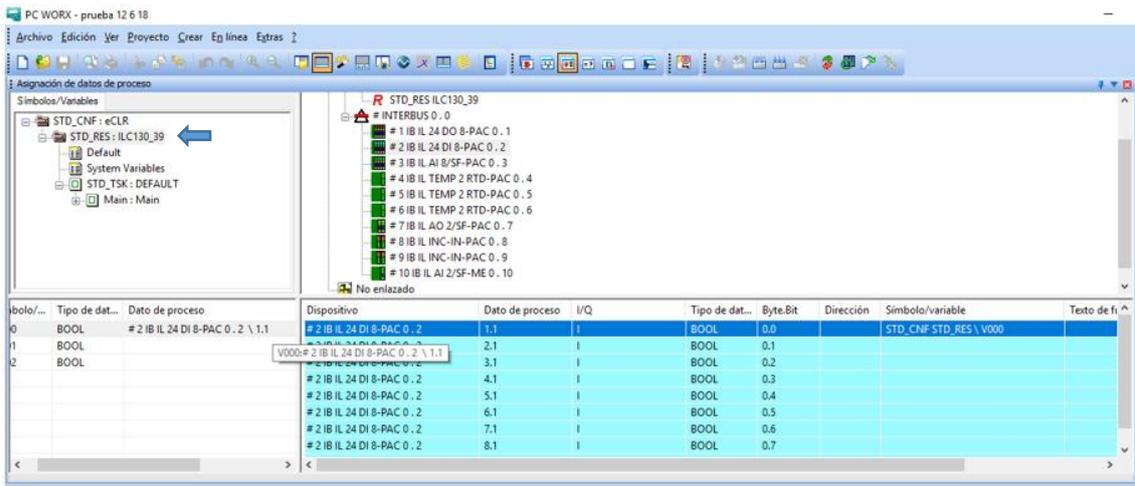


Figura 12: Selección del módulo de expansión y correspondiente bit de entrada

En la Figura 12 se muestra la selección del módulo de expansión, en este caso #2 IB IL 24 DI 8-PAC, con la entrada 1.1, misma que debe arrastrarse con el ratón para asociar a V000 en la ventana de la izquierda de “Asignación de datos de proceso”. Ese proceso debe repetirse para todas las variables que deben asociarse a módulos de expansión.

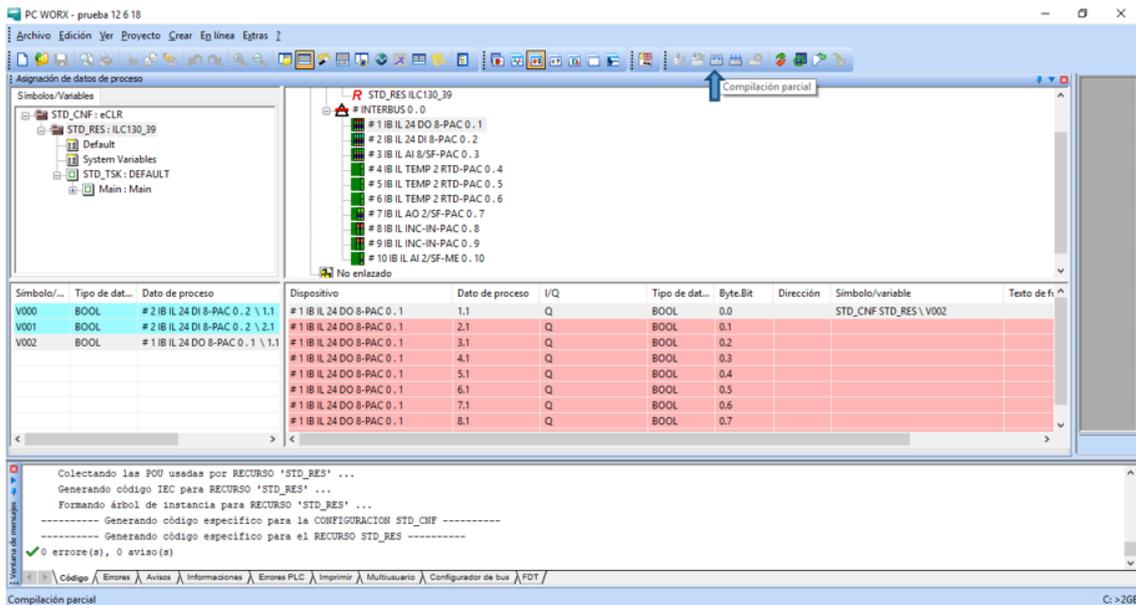


Figura 13: Compilación del programa.

Luego de compilar, se abre la ventana de “Diálogo de control de proyecto”, para controlar acciones del proyecto, tal como se muestra en la figura 14.

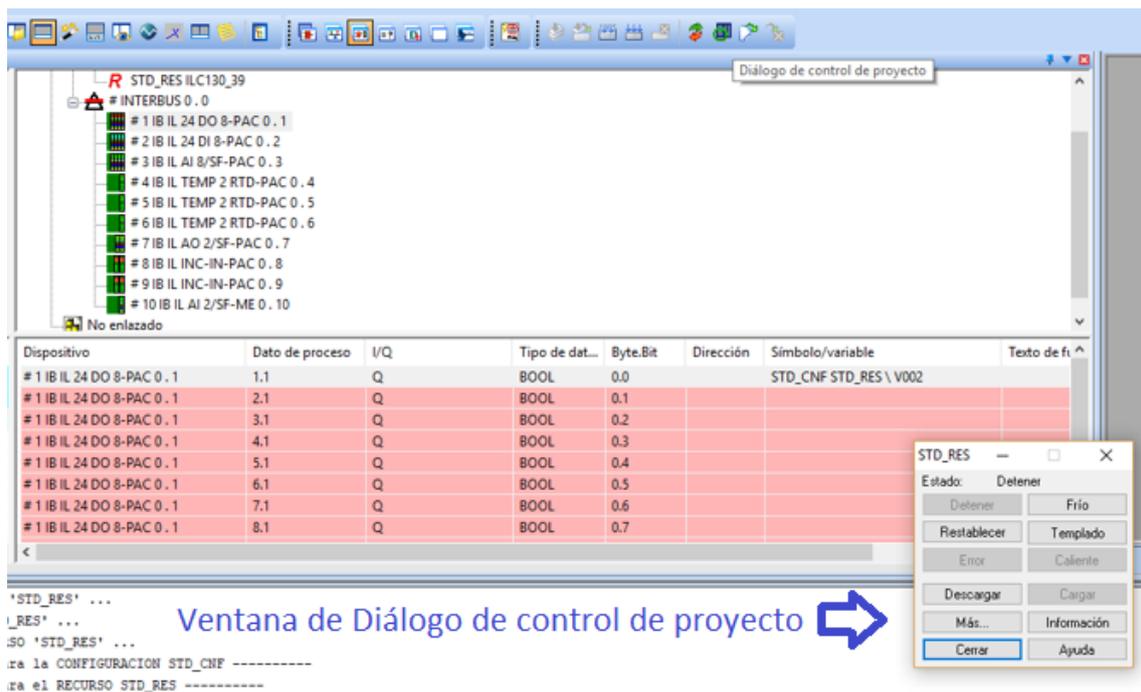


Figura 14: Abriendo la ventana de “Diálogo de control de proyecto”

En el flujograma de la figura 15, se muestran las acciones que pueden realizarse con la ventana de “Diálogo de control de proyecto”.

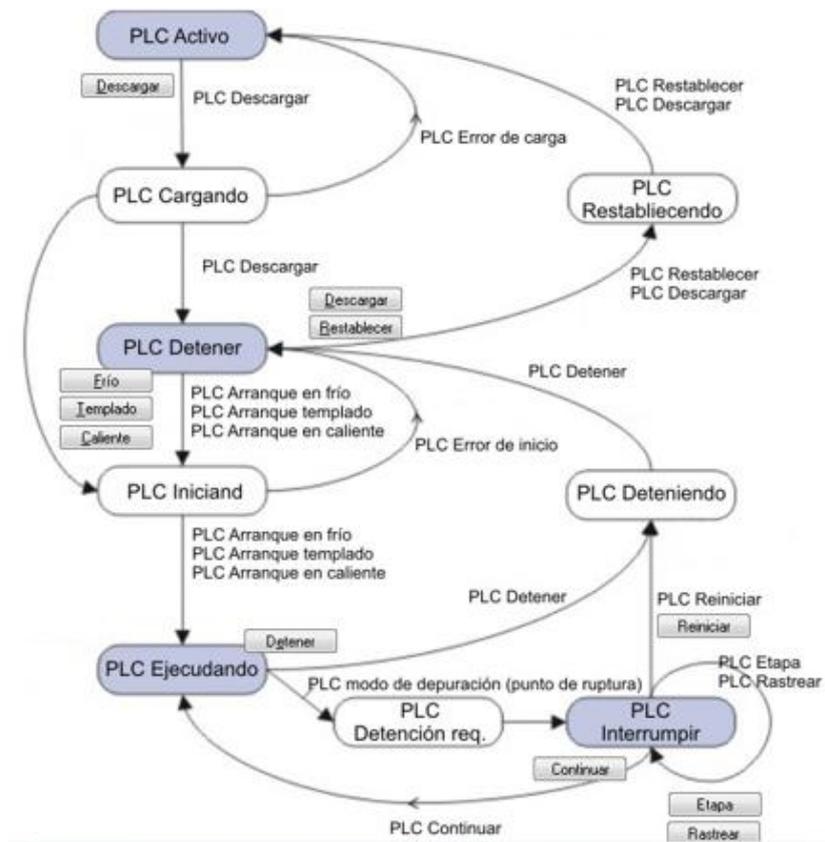


Figura 15: Flujo de acciones que pueden realizarse con la ventana de “Diálogo de control de proyecto”

Para bajar el proyecto al procesador, debe verificar que el selector en el mismo esté en “ON”.

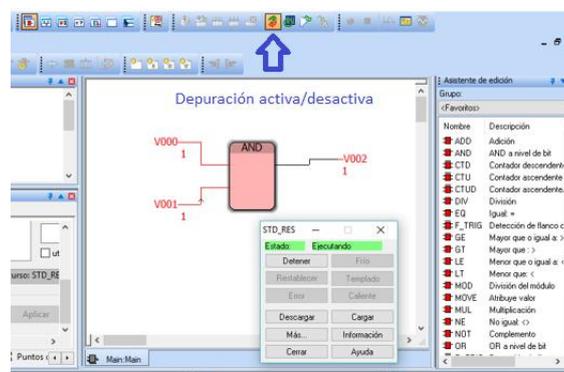


Figura 16: Monitoreo de operación

Puede realizarse monitoreo de operación con “Depuración activa/desactiva”, al activarse cambian a color rojo, para el ejemplo de la figura debe aplicarse 24V a las entradas a los pines 0.0 y 0.1 del módulo de expansión IB IL 24 DI 8-PAC y se obtiene salida en el pin 0.0 del módulo de expansión IB IL DO 8-PAC por la condición lógica AND.

Los módulos de expansión cuentan con leds que permiten saber si sus conexiones han sido activadas.

El tablero del IIME cuenta con entradas de interruptor conectadas directamente al procesador ILC 130 ETH, más fácil de usar y de programar pues ya no necesita realizarse el paso de asignación de datos de proceso, pues ya aparecen como “ONBOARD\_INPUT\_BIT0 y así sucesivamente, como se muestra en figura 17.

El programa quedaría así:

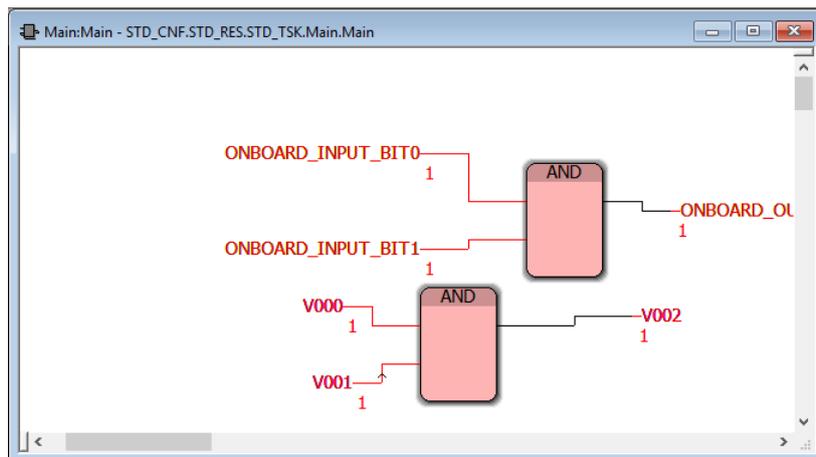


Figura 17: Programa empleando I/O del procesador principal como del módulo de expansión

Al monitorear, las variables energizadas se manifiestan con color rojo, tal como se ve en la figura 17.

## GUÍA BÁSICA DE AUTÓMATA PROGRAMABLE MICROLOGIX 1100B

**OBJETIVO:** Introducción a la parametrización y programación del autómata programable PC MicroLogix 1100 B instalado en un tablero del IIME.

**COMPETENCIAS:** Establece comunicación entre la PC y el procesador del autómata y módulos de expansión, realiza un programa básico de variables booleanas.

**FUNDAMENTO TEÓRICO:**

El controlador lógico programable MicroLogix™ 1100 Boletín 1763 añaden EtherNet/IP™ incorporada, edición en línea y un panel LCD a la familia MicroLogix. El panel LCD incorporado muestra el estado del controlador, el estado de E/S y mensajes del operador simples. Tiene diez entradas digitales y seis salidas digitales.

**Características**

- Incluye un puerto EtherNet/IP™ de 10/100 MBps incorporado para mensajería entre dispositivos similares.
- Proporciona una memoria de 8 KB (4 KB de programas de usuario con 4 KB de datos de usuario).
- Permite el acceso, el monitoreo y la programación desde cualquier conexión Ethernet
- Admite la edición en línea.
- Proporciona un servidor web incorporado que permite configurar los datos del controlador para que aparezcan como una página web.
- Contiene un puerto combinado RS-232/RS-485 aislado para comunicación en serie y conectada en red.
- Permite monitorear y modificar los datos del controlador a través de una pantalla LCD incorporada.
- Compatible con módulos de expansión de E/S MicroLogix 1762 (hasta cuatro módulos por controlador).
- Admite un máximo de 144 puntos de E/S digitales.

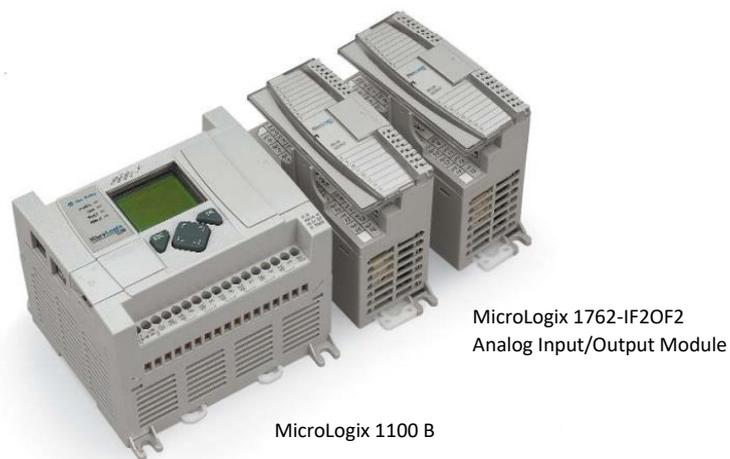


Figura 1: Autómata programable Micrologix 1100 B y módulos de expansión

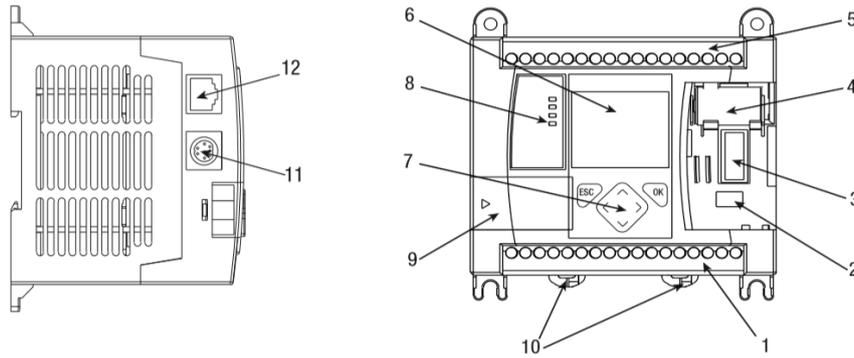


Figura 2: Vistas lateral y frontal de MicroLogix 1100 B

Donde:

1: Bloque de salida (Output)	4: Batería	7: Teclado acceso a LCD; OK, ESC, UP, DOWN, LEFT, RIGHT	10: Ganchos rieles DIN
2: Conector de batería	5: Bloque de entradas (Input)	8: Leds indicadores de estado	11: Puerto de comunicación RS 232/485 (canal 0 aislado)
3: Bus conector de interfaz para expansiones I/O	6: LCD	9: Módulos de memoria, el incluido y el opcional	12: Puerto ETHERNET (canal 1)

PROGRAMACIÓN

Softwares:

- RSLogix 500, Micro Starter Lite versión 8.30.00: Edición de programas.
- RSLinx Classic, versión 4.00.01 CPR 9 SR 10.0: Comunicación Ethernet entre la PC y el autómatas.

1er Paso: Abrir el software RSLogix Micro Starter Lite

Para programación del PLC, RSLogix y para iniciar un nuevo programa, mediante el menú de la barra superior “File/New” seleccionar un nombre al procesador, el modelo del autómatas de laboratorio así como el driver de comunicación AB\_ETH1 como se muestra en la figura 3. Si solo se desea simular escoger EMU500-1.

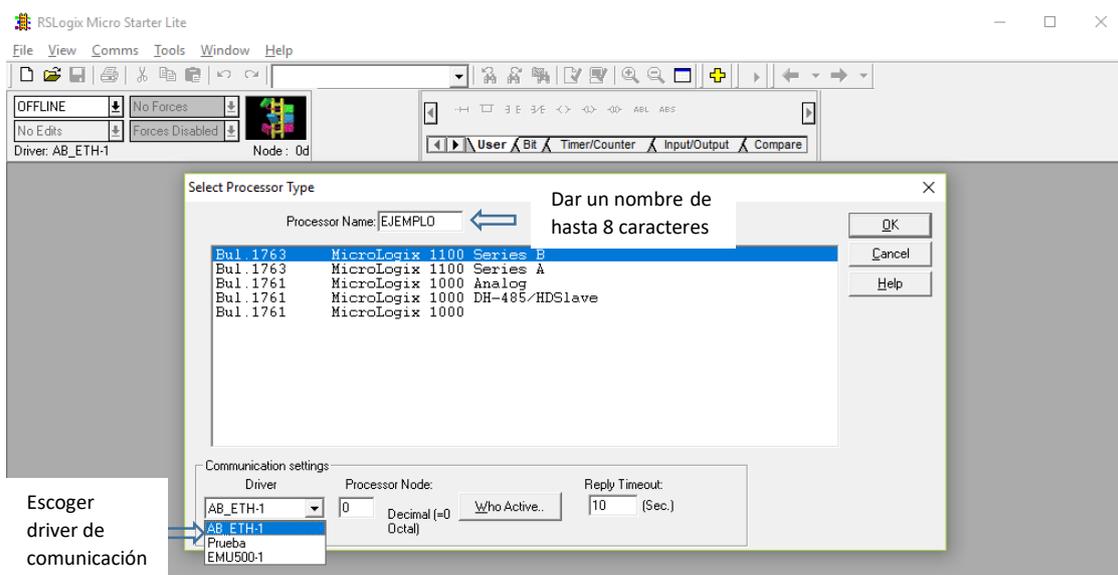


Figura 3: Iniciando un Nuevo proyecto

2do Paso: Abrir el software:  RSLinx Classic Lite

Para establecer comunicación Ethernet entre la computadora y el MicroLogix. Desplegar de la “ventana de árbol del proyecto” “comunicación AB\_ETH-1, Ethernet la IP del MicroLogix, misma que aparecerá con una “x” roja si no se ha establecido comunicación, como se muestra en la figura 4.

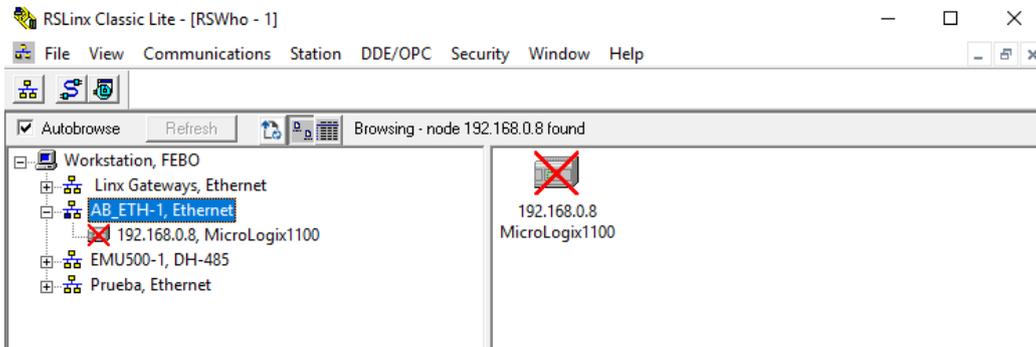


Figura 4: RSLinx, software de comunicación Ethernet

Debe conectarse mediante el conector RJ45 la computadora y el MicroLogix, de ese modo si la dirección de IP es correcta, pudiéndose comprobar también mediante “Símbolo del sistema” como se muestra en la figura 5.

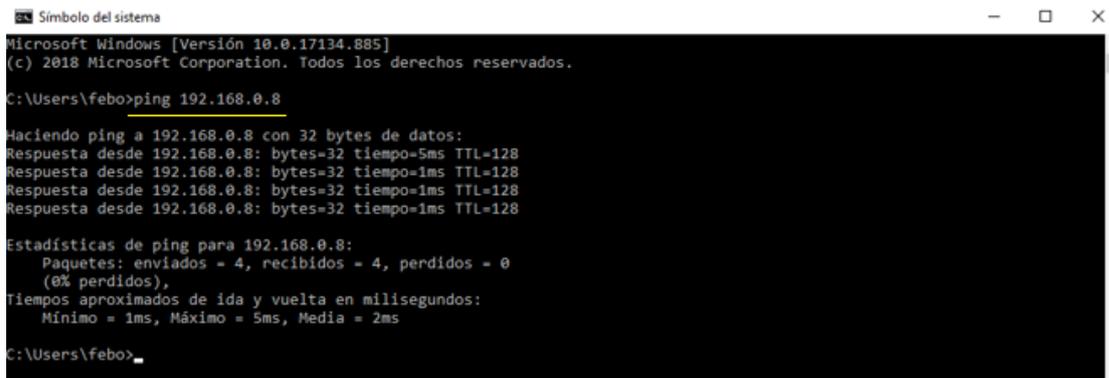


Figura 5: Verificación de conexión Ethernet mediante “Símbolo del sistema” de Windows

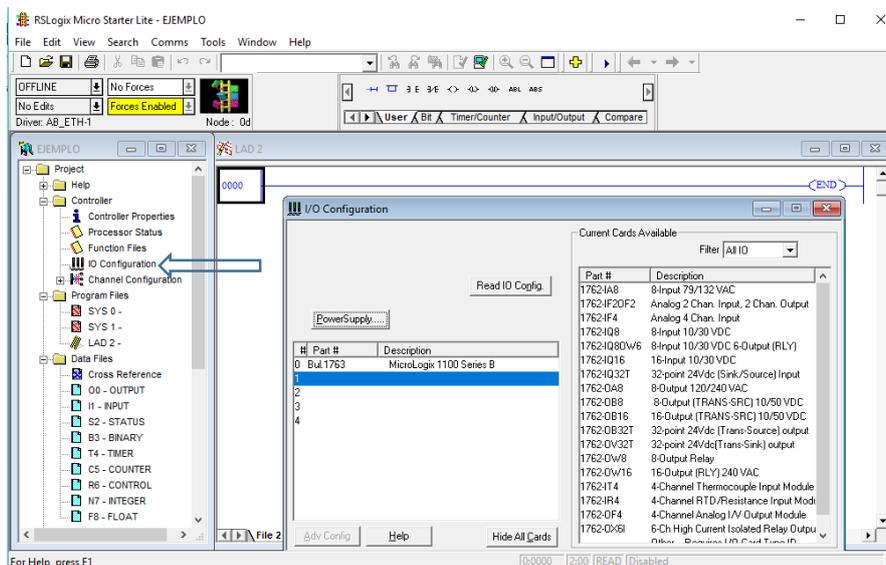


Figura 6: Configuración de módulos de expansión

Al establecerse conexión, la “x” de color rojo desaparece. Posteriormente deben configurarse los módulos de expansión entrando en la “ventana del árbol del proyecto” como se muestra en la figura 6.

Los módulos de expansión del tablero didáctico del IIME son IF20F2, mismos que son seleccionados como se muestra en la figura 7.

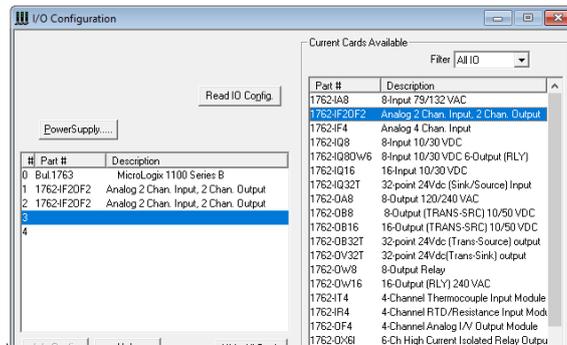


Figura 7: Selección de los dos módulos de entradas y salidas analógicas del IIME

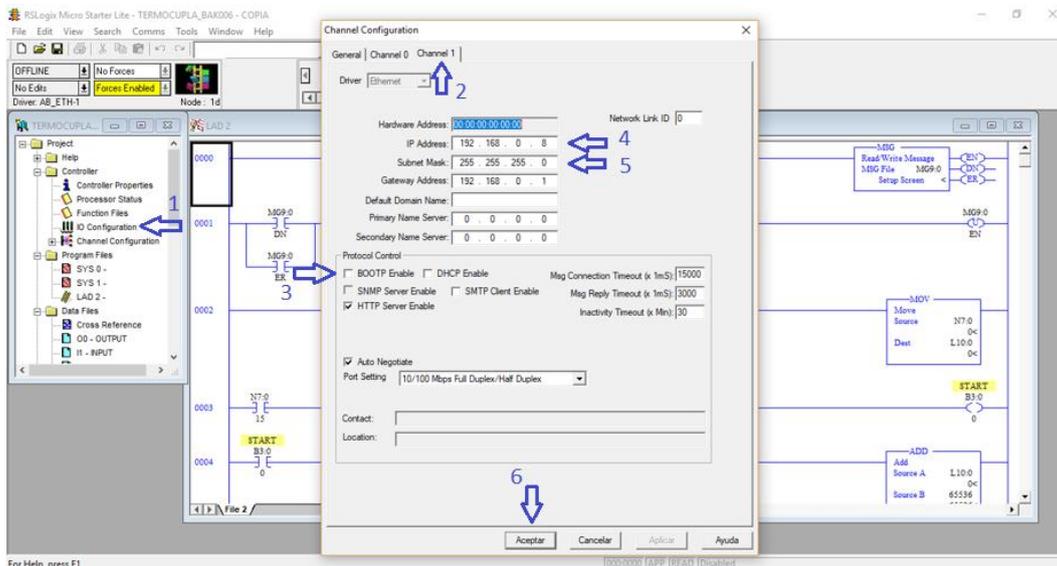


Figura 8: Establecimiento de conexión Ethernet mediante Canal 1 de MicroLogix

En la Figura 8 se muestra la secuencia de pasos para conectar mediante Ethernet MicroLogix con la computadora personal. El IP debe corresponder con el que se indique en la pantalla LCD del MicroLogix, al cual se accede mediante:

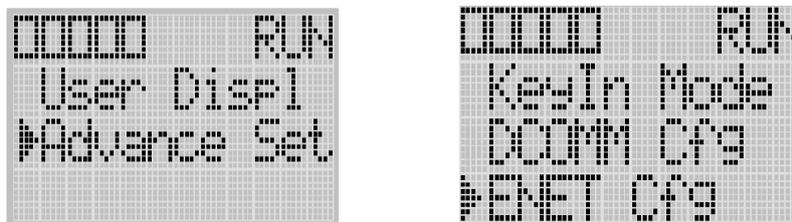


Figura 9: Despliegue en la pantalla LCD de MicroLogix 1100 B

Deberá desplegarse el número MAC/IP; en este caso el IP está con **192.168.0.8**.

Si no estuviera configurado un IP en MicroLogix, debe hacerse a través de “Chanel 0” o mediante el programa BOOT que puede descargarse de <https://bootp-dhcp-server.software.informer.com/download/?lang=es> o [www.ab.com/networks/bootp/index.html](http://www.ab.com/networks/bootp/index.html)

También debe configurarse la comunicación en “RSLinx Classic” con el IP correspondiente.

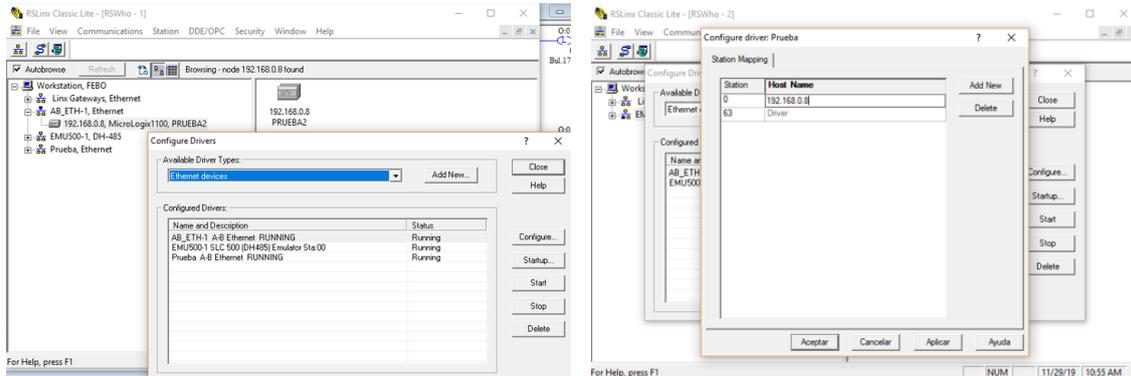


Figura 10: Configuración de “Ethernet devices” e “IP” en RSLinx

### 3er Paso: Programar

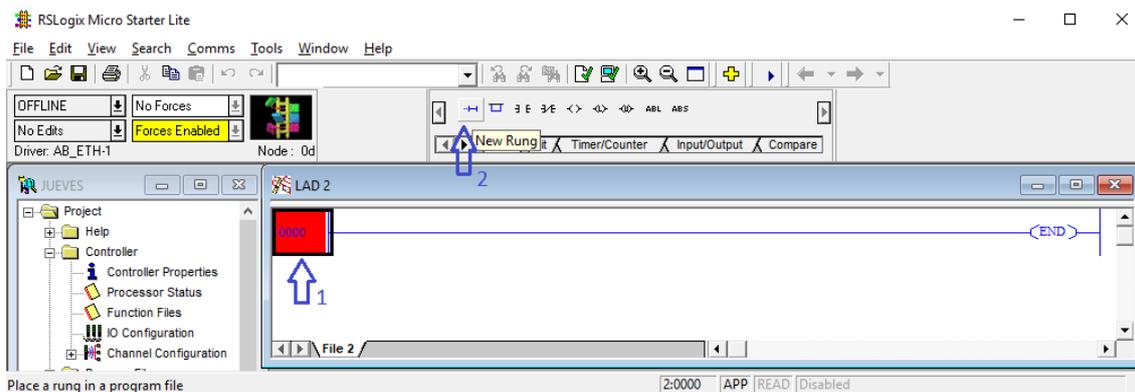


Figura 11: Inicio de programación en RSLogix

Marcando con el ratón en el encabezado del primer escalón, éste se pondrá de color rojo, posteriormente del menú escogemos “new Rung” nuevo escalón en español. De ese modo se adicionan más escalones, contactos y bobinas de acuerdo al programa a realizar.

Al introducir un contacto en el primer escalón (000), podemos escoger el adecuado del “archivo de datos” de la ventana de árbol del proyecto, por ejemplo “I1 INPUT” que corresponde a las entradas físicas del MicroLogix o, por ejemplo el registr “B3 BINARY” que permite generar señales de entrada al MicroLogix a través del teclado de la computadora, tal como se muestra en la figura xx, se puede escoger por ejemplo B3:0/0 el primer bit del registro B3, que debe escribirse en el margen superior del marco que rodea al contacto.

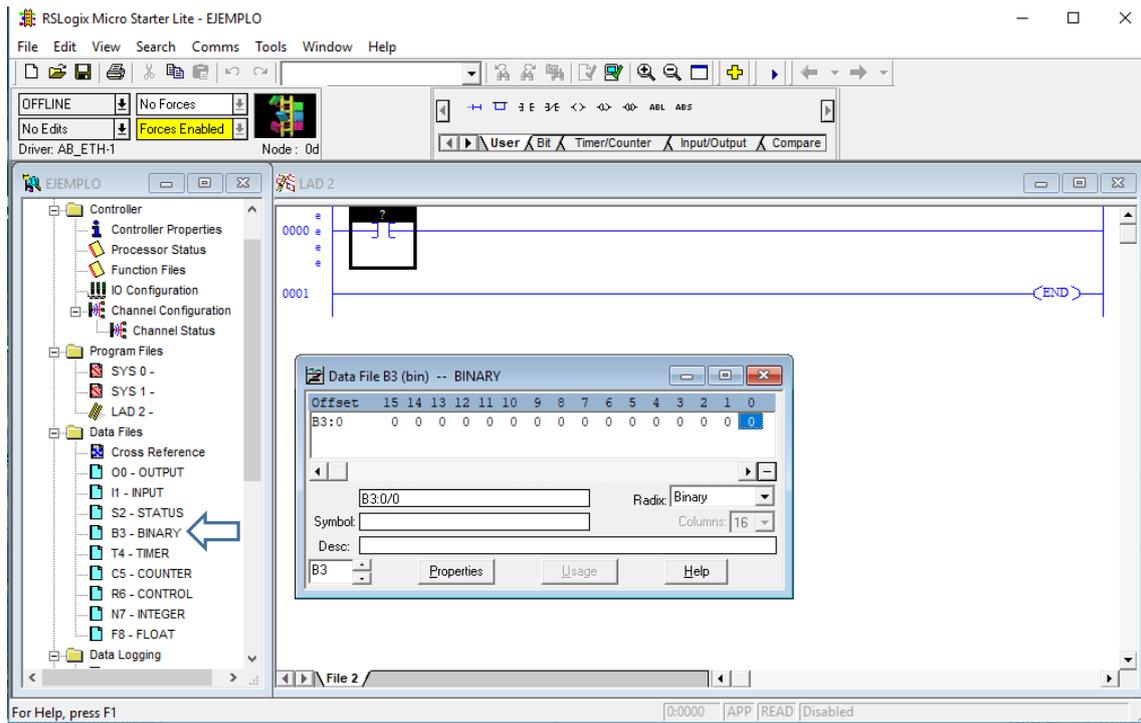


Figura 12: Selección de la variable para el contacto del escalón 000.

Para obtener finalmente, por ejemplo el programa básico de variables booleanas mostrado en la figura 12.

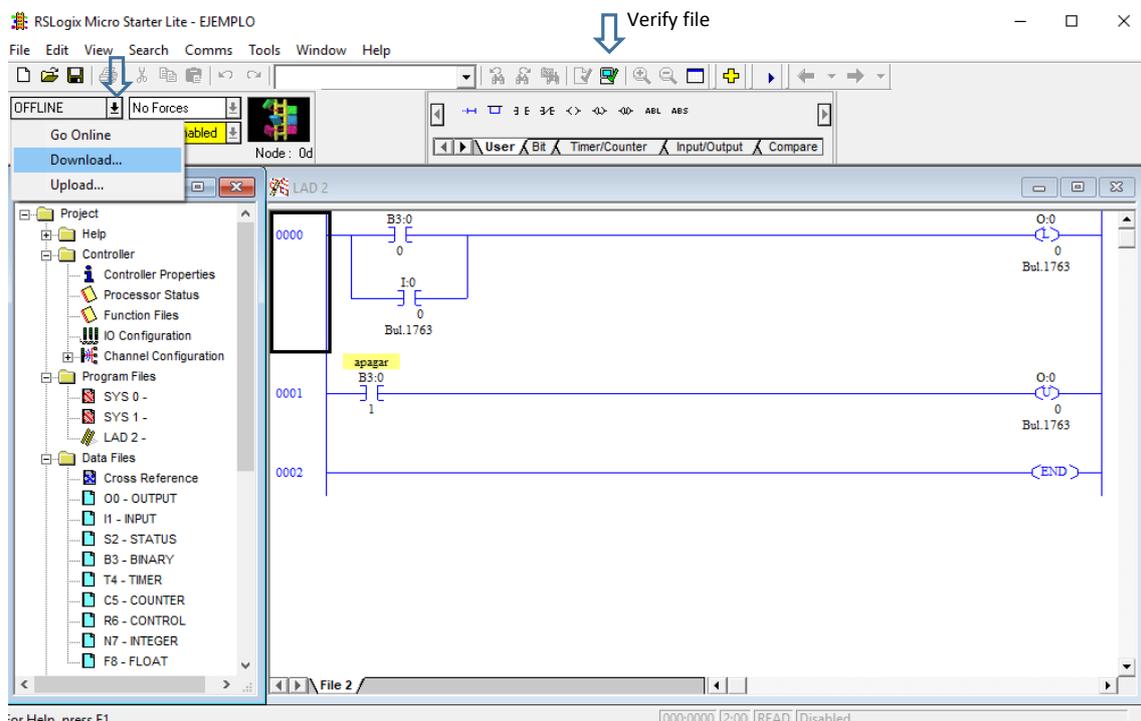


Figura 13: Programa básico de ejemplo

En el programa básico de ejemplo mostrado en la figura 13, ya se ejecutó la verificación del programa mediante “Verify file” de la barra de menú superior. En los contactos y bobinas aparecen textos, cuyo significado se muestra en la figura 14.



Figura 14: Información acompañada a contactos y bobinas

Posteriormente se va a la ventana “Processor Status” para descargar con “download” el programa al procesador del MicroLogix.

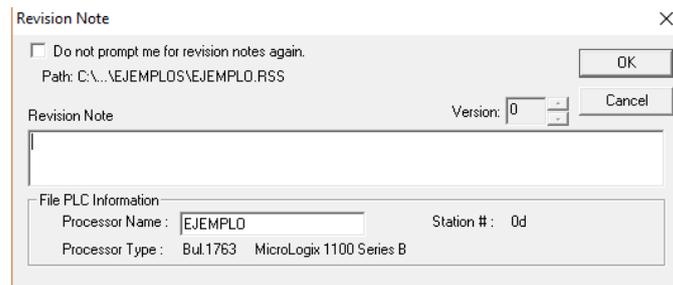


Figura 15: Cuadro de diálogo para ejecutar EJEMPLO

En la Figura 15, colocamos “OK”, que prodría generar otro cuadro de diálogo solicitando que en MicroLogix, mediante su teclado y pantalla LCD se cambie del modo RUN al modo PROG como se muestra en la figura 16

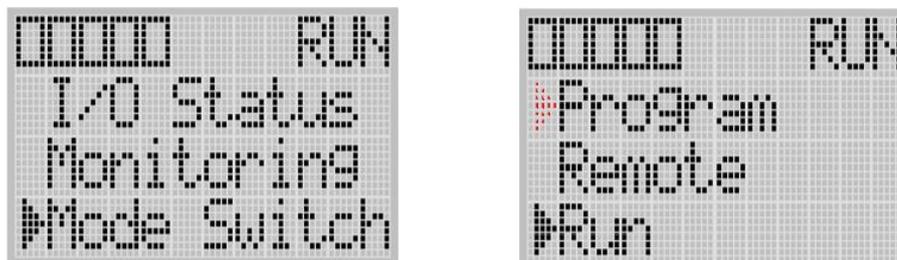


Figura 16: Cambio del modo Program a Run y viceversa

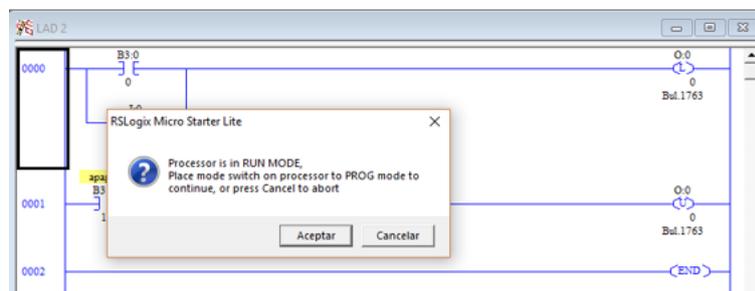


Figura 17: Cuadro de diálogo para cambiar modo de funcionamiento con teclado de MicroLogix

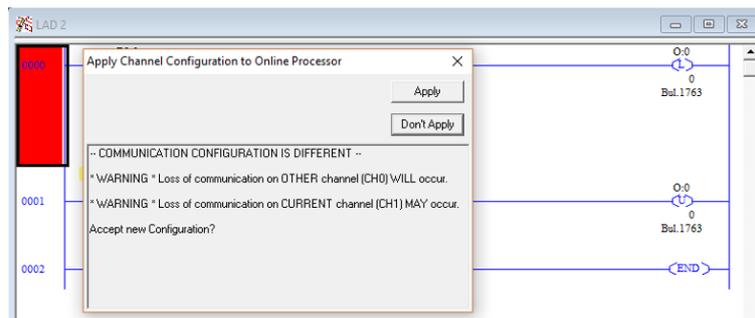


Figura 18: Mensaje de advertencia, aplicar “Don’t Apply”

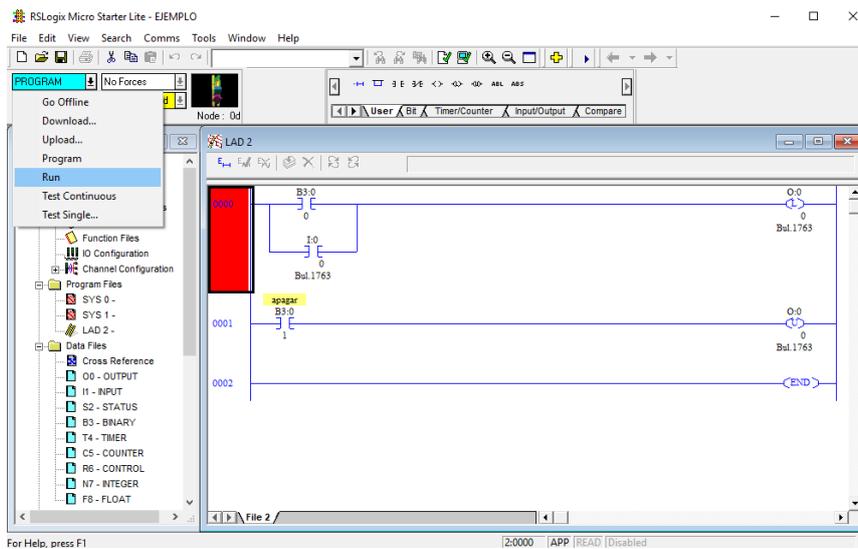


Figura 19: De ventana “Processor Status” ir a modo RUN

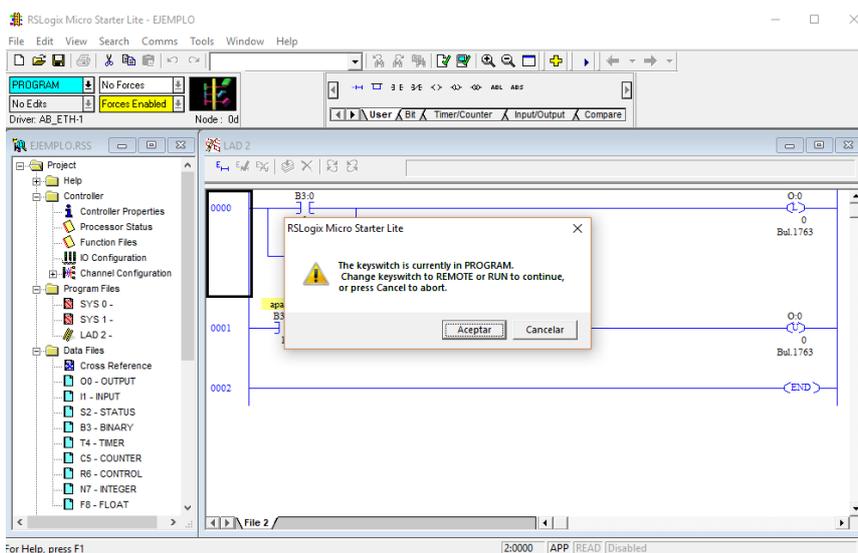


Figura 20: Mensaje para cambiar de modo PROG a RUN (teclado - LCD MicroLogix)

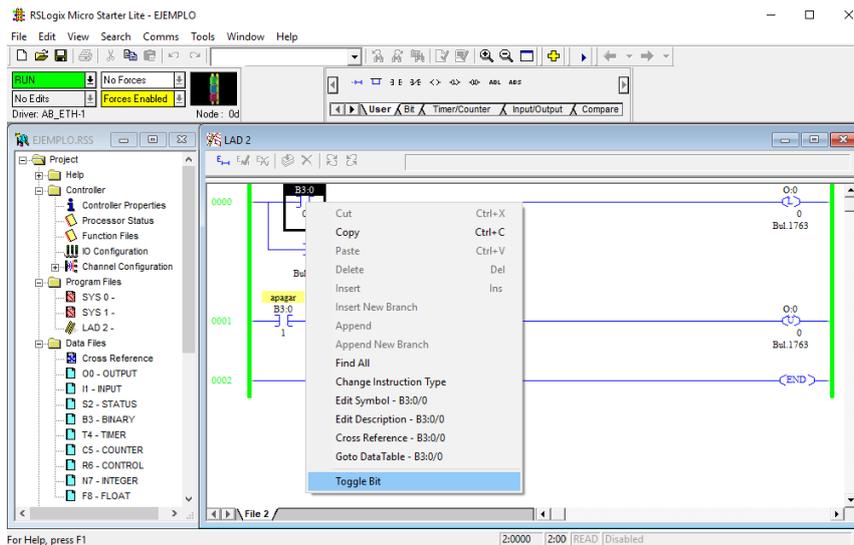


Figura 21: Conectando mediante la computadora al contacto B3:0/0

En la Figura 21, se muestra monitoreo de MicroLogix ya corriendo, las líneas verdes representan que están energizadas. B3:0/0 se activa pulsando el lado derecho del ratón sobre dicho contacto. La Bobina -(L)- representa Latch (Set) y -(U)- es Unlatch (Reset), que activan y desactivan a la salida física O:0/0 respectivamente.

Referencias:

[https://literature.rockwellautomation.com/idc/groups/literature/documents/um/1763-um001\\_-en](https://literature.rockwellautomation.com/idc/groups/literature/documents/um/1763-um001_-en)

<https://ab.rockwellautomation.com/es/Programmable-Controllers/MicroLogix-1100>