

ACTIVIDAD 1

MARCOS MANZO TORRES

JONATHAN FONSECA CAMARENA

CONTROLADORES LÓGICOS PROGRAMABLES

MORÁN GARABITO CARLOS ENRIQUE

PRESENTACIÓN:

En esta actividad y tras haber configurado durante aproximadamente dos semanas nuestra RASPBERRY PI3, comenzamos a trabajar en la programación de nuestra primera actividad para lo cual también necesitamos elaborar nuestra caja de operaciones para asimilar la ejecución de una bomba de presión.

Es muy importante el hecho de resaltar, que a la primera prueba comprobamos que el diagrama presentaba algunos fallos por lo que lo tuvimos que editar y por consecuente volver a realizar en LOGICLAB para después volverlo a cargar en nuestra RASPBERRY PI 3.

A continuación se presenta arte de la elaboración que llegamos para llegar a nuestra objetivo.

MARCO TEORICO:

Para programar un autómata con Ladder, además de estar familiarizado con las reglas de los [circuitos de conmutación](https://es.wikipedia.org/wiki/Circuito_de_conmutaci%C3%B3n), (también denominada Lógica de Contactos), es necesario conocer cada uno de los elementos de que consta este lenguaje. A continuación, se describen de modo general los más comunes.

|  |  |  |
| --- | --- | --- |
| **Elementos básicos en Ladder** | | |
| **Símbolo** | **Nombre** | **Descripción** |
| [Capacitor symbol.svg](https://commons.wikimedia.org/wiki/File:Capacitor_symbol.svg) | Contacto NA | Se activa cuando hay un uno lógico en el elemento que representa; esto es, una entrada (para captar información del proceso a controlar), una variable interna o un bit de sistema. |
| [BobinaNC.svg](https://commons.wikimedia.org/wiki/File:BobinaNC.svg) | Bobina NC | Se activa cuando la combinación que hay a su entrada (izquierda) da un cero lógico. Su activación equivale a decir que tiene un cero lógico. Su comportamiento es complementario al de la bobina NA. |
| [BobinaSet.svg](https://commons.wikimedia.org/wiki/File:BobinaSet.svg) | Bobina SET | Una vez activa (puesta a 1) no se puede desactivar (puesta a 0) si no es por su correspondiente bobina en RESET. Sirve para memorizar bits y, usada junto con la bobina RESET, dan una enorme potencia en la programación. |
| [BobinaJump.svg](https://commons.wikimedia.org/wiki/File:BobinaJump.svg) | Bobina JUMP | Permite saltarse instrucciones del programa e ir directamente a la etiqueta que se desee. Sirve para realizar subprogramas. |

Se suele indicar mediante los caracteres B ó M y tienen tanto bobinas como contactos asociados a las mismas de los tipos vistos en el punto anterior. Su número de identificación suele oscilar, en general, entre 0 y 255. Su utilidad fundamental es la de almacenar información intermedia para simplificar esquemas y programación.

Los bits de sistema son contactos que el propio autómata activa cuando conviene o cuando se dan unas circunstancias determinadas. Existe una gran variedad, siendo los más importantes los de arranque y los de reloj, que permiten que empiece la ejecución desde un sitio en concreto y formar una base de tiempos respectivamente. Su nomenclatura es muy diversa, dependiendo siempre del tipo de autómata y fabricante.

En forma práctica, los contactos pueden representar, de acuerdo a la programación que se realice, la entrada de información, por ejemplo el activamiento de un sensor, o el activamiento de una memoria interna, comúnmente denominada M.

Las Bobinas, representan a esas memorias M así como las salidas al exterior, activan o desactivan elementos como puede ser una luz, un motor eléctrico, un solenoide de una válvula neumática, etc.

En algunas versiones de Ladder, como el rele inteligente ZELIO, las bobinas representan también funciones, como temporizadores, contadores y comparadores.

DESCRIPCIÓN DE ACTIVIDAD:

*Se tiene un cilindro en posición de home (retraído) y con el sensor A activo. Cuando el operador oprime el botón P el sensor sale hasta el sensor C, y si el botón se deja de presionar, se regresa a HOME pero si presiona el botón Q, el cilindro sale hasta B y cuando deje de presionar regresa a HOME.*

CONDICIONES

* **HOME, el cilindro hasta atrás, sensor A activo**
* **P y A activados= mover hasta C……P Y C activos= regresa a HOME**
* **Q y A activados= mover hasta B……Q y B activos= Regreso a HOME**

Teniendo las operaciones ye indicaciones, pasamos a desarrollar el programa el LOGICLAB el cual nos quedó de la siguiente manera:

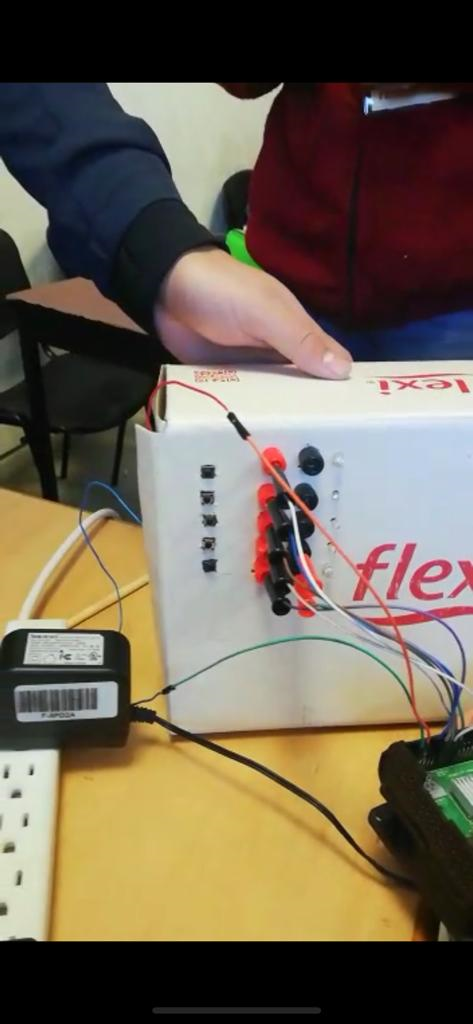


Programación en logiclab.

Barra de herramientas.

Diagrama de escalera.

ACOMODO DE NUESTRA CAJA DE OPERACIONES:



INDICADORES LED

BOTON NUMERO 1

TIERRA DE NUESTRA CAJA DE OPERACIONES (CABLE NARANJA))

BOTON NUMERO 2

SALIDAS DE NUESTRO PROGRAMA

BOTON NUMERO 3

ENTRADAS DE NUESTRO PROGRAMA

BOTON NUMERO 5

BOTON NUMERO 6

ENTRADA DE VOLTAJE(CABLE AZUL)

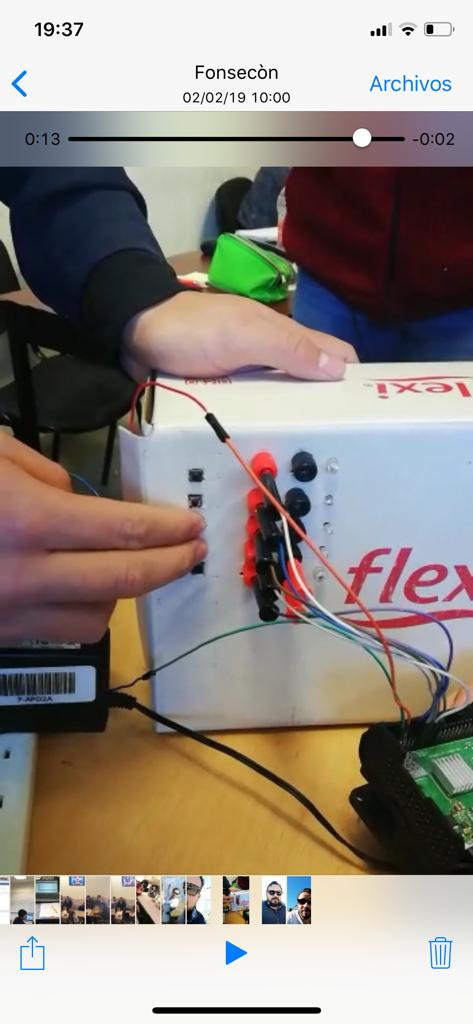
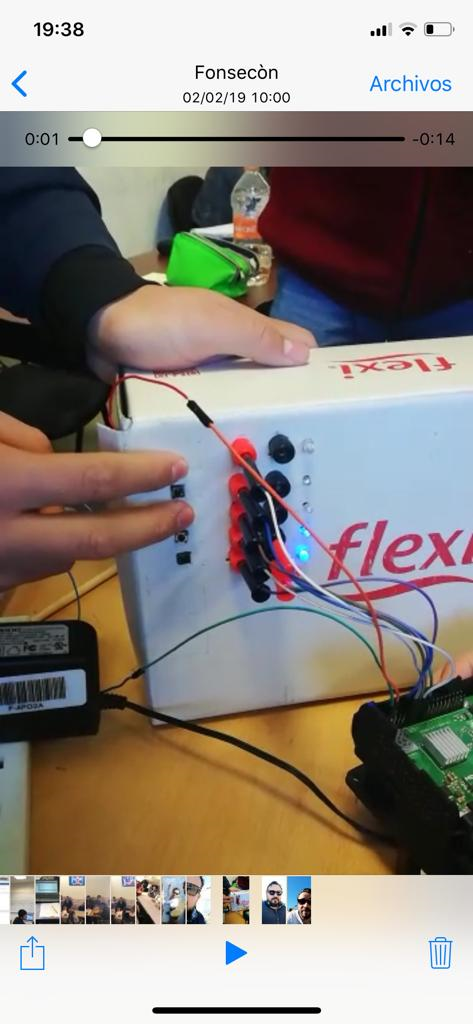
REALIZACIÓN DE LA ACTIVIDAD:

A continuación, tenemos la evidencia de nuestra actividad, de acuerdo al problema que intentamos solucionar con nuestra programación tenemos las siguientes combinaciones que podemos observar en las imágenes:

Primera combinación: para que se encienda el primer led, debemos presionar el botón numero 1 con el botón número 3.

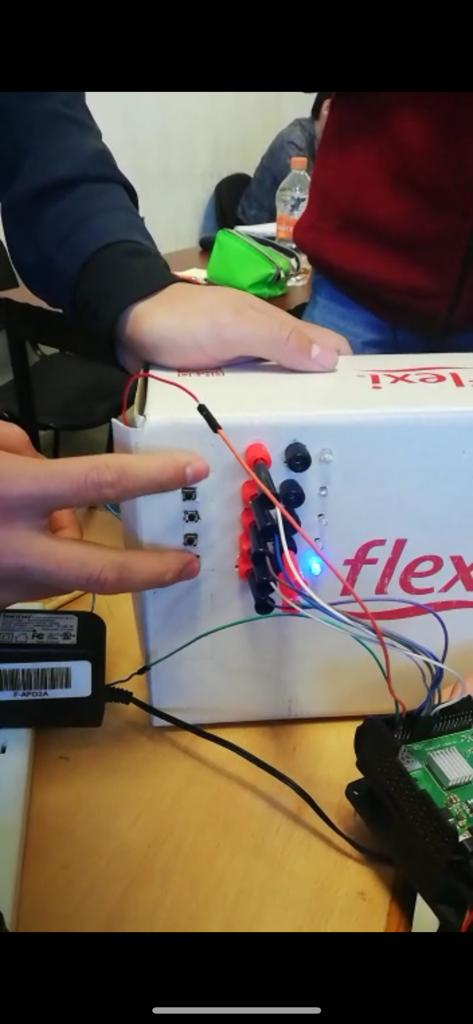
Segunda combinación: después de la primera combinación, presionamos el botón numero 1 y el botón numero 5 y pasamos a encender el segundo led.

Tercera combinación: presionamos el botón numero numero 3 y el botón numero 4 y se apagan por completo todos los led.



TERCERA COMBINACIÓN

PRIMERA COMBINACIÓN



SEGUNDA COMBINACIÓN

SEGUNDA COMBINACIÓN

CONCLUSIÓN: Esta actividad, lo mas complicado que resaltaría es el hacer que todo funcionara por primera vez, ya que aunque en teoría la misma fue fácil de elaborar, lo mas difícil fue asociar todos los elementos tales como: hacer que nuestra configuración de RASBERYY estuviera completa y la detectara nuestro LOGICLAB para después proceder a hacer la conexión entre los dos elementos y que esta apareciera de manera correcta en los dos indicadores que nos presenta nuestro LOGICLAB.

Posterior a ello otro paso era el que nuestro programa fuera compilado correctamente al haber declarado las variables y al hacer nuestro diagrama de escalera.

Después de todo lo mencionado anteriormente y tras comprobar que estuviera correcto, pasamos a descargar el programa a nuestra RASPBERRY para ahora comenzar a trabajar conectada con nuestra caja de operaciones de manera de entradas y salidas en la RASPBERRY.

Cabe resaltar que un equipo fue el más avanzado de todos y quien nos fue guiando o mas bien, del que fuimos aprendiendo de los errores para posterior a ello realizar ciertas correcciones y realizar todo de una maneras más fácil y efectiva.

Presentamos dificultades ya que la programación de nuestra actividad no estaba de la manera correcta, la cuarta combinación que era con el botón 2 y con el botón numero 4, no prendía el primer led que nos regresaba a home, con lo cual tuvimos que acudir al profesor para que nos orientara varias veces.

Al final, el aprendizaje que nos llevamos como equipo es el haber realizado nuestra primera programación de PLC y con lo que esperamos que sea la primera de muchas que vamos a realizar a lo largo de nuestra carrera profesional.

