

Name:- Date 30/5/2020

Course:- Logic design

Title:- Application of  
Programmable logic control

Name:- Poojary Sushant

USN:- 4AL18EC400

Sem:- 6<sup>th</sup> sem B sec

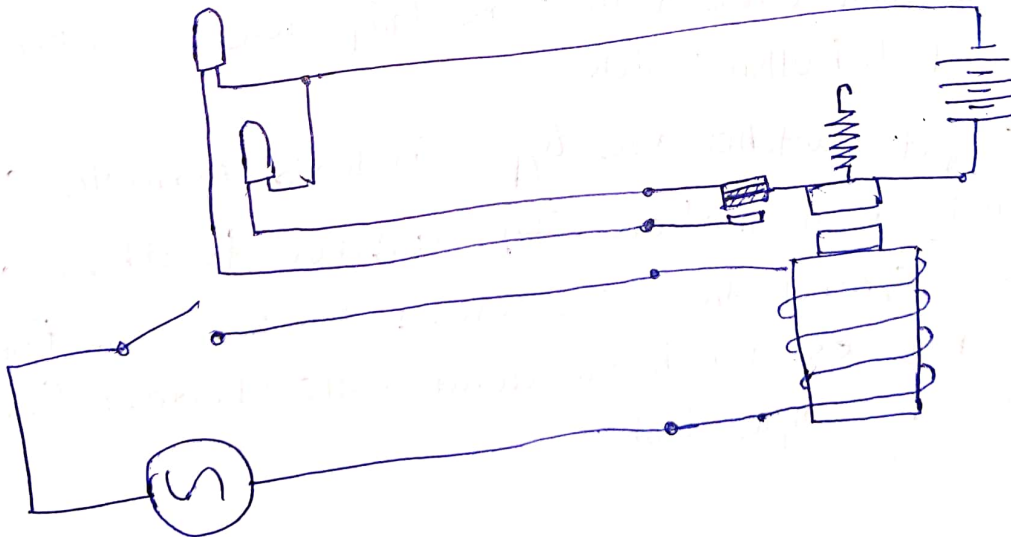
✱ An introduction to Programmable logic controller's

### Relays

Relays are switches that open & close circuits electromechanically or electronically. Relays control one electrical circuit by opening & closing contacts in another circuit. When a relay contact is Normally closed (NC) there is a closed contact when the relay is not energized.

→ Normally closed (NC) :- NC contact allow current through when the relay or contactor is not energised

→ Normally open (NO) :- NO contact allows current when the relay is energized



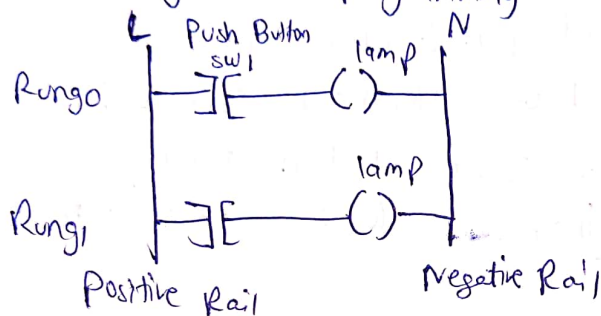
## ★ PLC Programming Basics:-

A CPU of the PLC executes two different program

→ The operating system

→ The user program

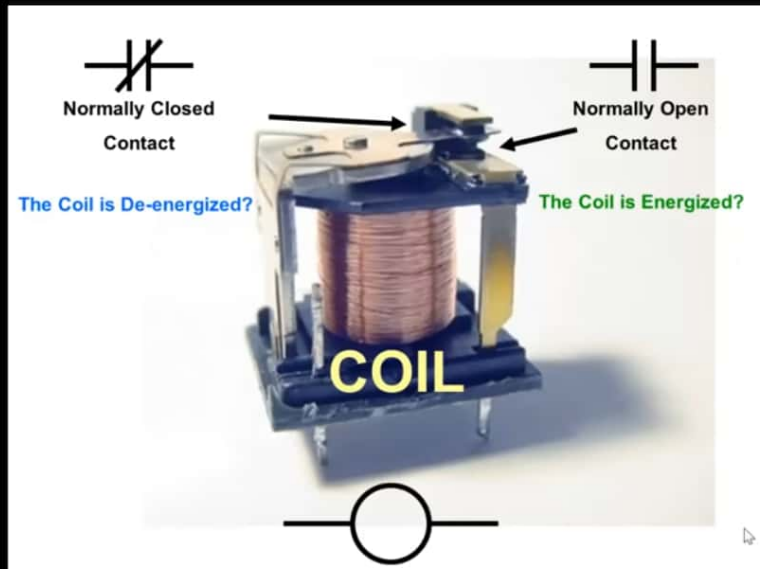
### Ladder Logic PLC Programming



Among several programming languages, ladder logic diagram is the most basic & simplest form of programming the PLC. Before to program the PLC with this language, one should know some basic information about it. The below figure shows the hardwired ladder diagram where in the same lamp load is controlled by two push button switches.

Input switches are types included normally closed & normally as shown above. In addition to above given function symbols, there are several functions like timer, counter, PID etc. which are stored in the standard library to program complex tasks.

triangle (img, (x, y), (x+w, y+h), (0, 255, 0), 3)



All

PLC

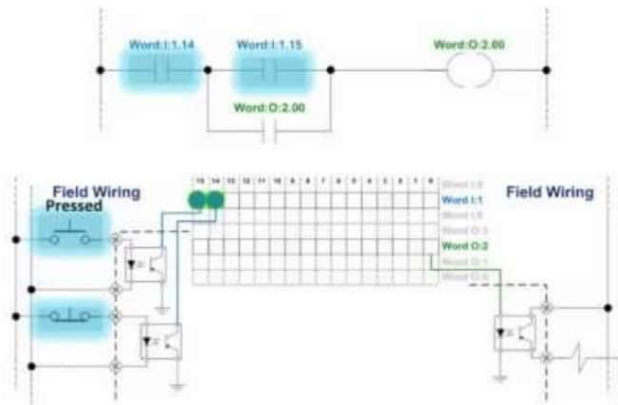
Related

From plcprofessor

Ad-Free Music. Free with Prime Ad-free music with free offline downloads. Included with Prime.

Ad 4.4 ★★★★★ FREE

How could this be done with a normally closed PB?



1:34:34



What is a PLC? PLC Basics Pt2

plcprofessor · 537K views · 7 years ago

Stories and short videos

0:29

Me too 

0:15

Diy earrings

0:26



(2)

Date: 30/5/2020

Name: Poojary sushant

Title:

Course: Python

USN: 4ALIBEC400

Title Python for video &amp;

Sem: 6<sup>th</sup> sem 'B'Image processing using  
Open CV

Loading, Displaying, Resizing &amp; writing images;

```
import cv2
```

```
img = cv2.imread("galaxy.jpg", 0)
```

```
print(type(img))
```

```
print(img)
```

```
print(img.shape)
```

```
print(img.ndim)
```

```
resized_image = cv2.resize(img, (int(img.shape[0]/2), int(img.shape[1]/2)))
```

```
cv2.imshow("Galaxy", resized_image)
```

```
cv2.imwrite("Galaxy_resized.jpg", resized_image)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

Face detection

```
import cv2
```

```
face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
```

```
img = cv2.imread("photo.jpg")
```

```
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
faces = face_cascade.detectMultiScale(gray_img,
```

```
scale_factor = 1.05
```

```
min_neighbors = 5)
```

```
for x, y, w, h in faces:
```

```
img = cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 3)
```

```
print (type (faces))
```

```
print (faces)
```

```
resized = cv2.resize (img, (int(img.shape[1]/3), int(img.shape[0]/3)))
```

```
Video capturing
```

```
import cv2, time
```

```
video = cv2.VideoCapture(0)
```

```
a = 0
```

```
while True:
```

```
    a = a + 1
```

```
    check, frame = video.read()
```

```
    print (check)
```

```
    print (frame)
```

```
    gray = cv2.cvtColor (frame, cv2.COLOR_BGR2GRAY)
```

```
    # time.sleep (3)
```

```
    cv2.imshow ("capturing", gray)
```

```
    key = cv2.waitKey (1)
```

```
    if key == ord ('q'):
```

```
        break
```

```
print (a)
```

```
video.release()
```

```
cv2.destroyAllWindows
```

