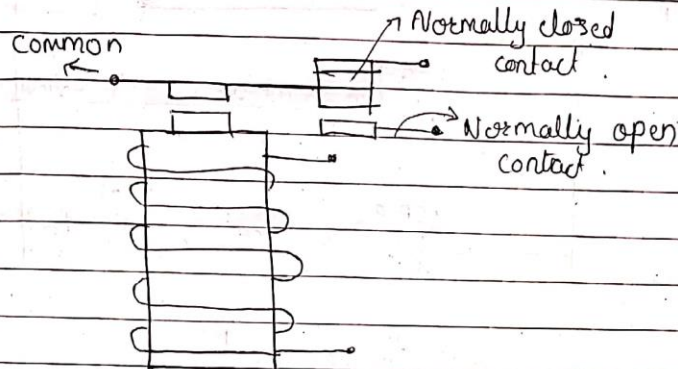


Date:	30-05-2020	Name:	poorvi
Course:	Logic design	USN:	4AL17EC071
Topic:	Allpication of programmable logic controllers	Semester and section	6 <sup>th</sup> sem And B sec

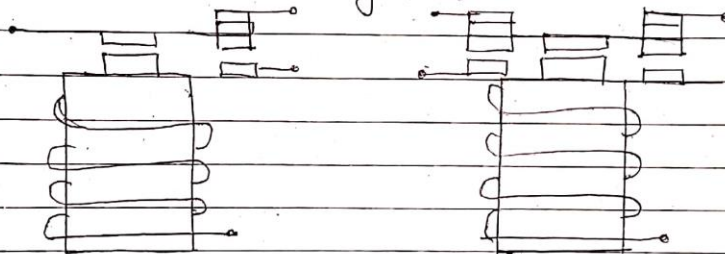
## # Applications of Programmable Logic Controllers

### \* Control Relay



In this we can use more contact, i.e., set of coil.  
In this we use only one terminal.

How the contacts are designed



Rotates a common contact  
b/w the normally closed  
& normally open.

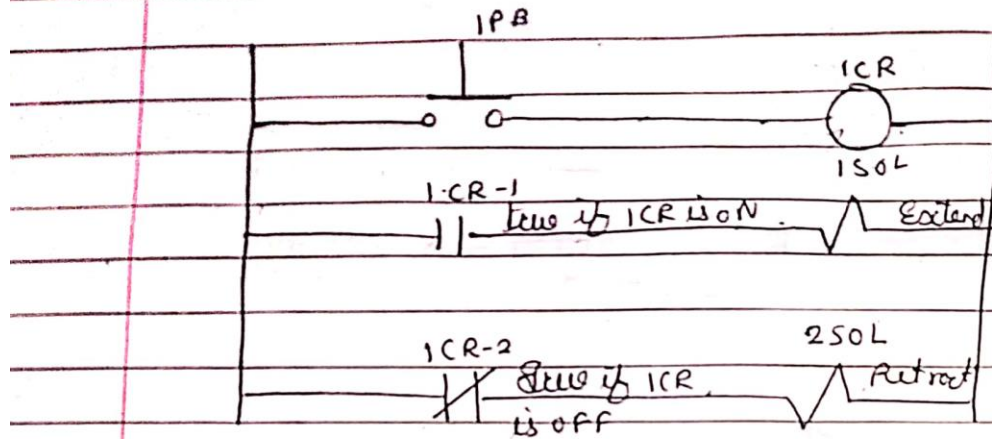
Pulls a shunting bar b/w  
the normally closed & the  
normally open contact.

NC  
Normally close

Normally open

ICR

## \* Numerical simulation value :



\* 2SOL is currently energized.

- Currently 1CR-2 is true & IPB & 1CR-1 is false
- with IPB closed 1CR & 1SOL gets energized & IPB & 1CR-1 is ON & 1CR-2 is OFF.

Date:	30-05-2020	Name:	poorvi
Course:	Python programming	USN:	4AL17EC071
Topic:	Python for image and video processing using open cv	Semester and section	6 <sup>th</sup> sem And B sec

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

## # Python for Image and Video Processing with OpenCV

- Installing the library -

```
pip install opencv-python
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[1]/2),
                                int(img.shape[0]/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")
# name of the jpg file where the photo is stored.
img = cv2.imread("photo.jpg")
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray_img,
scaleFactor=1.05,
minNeighbors=5) ] change this or adjust this
```



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)))  
cv2.imshow("gray", resized)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

#### • Capturing Video :-

```
import cv2
video = cv2.VideoCapture(0)
a = 0
while True:
    check, frame = video.read()
    a = a + 1
    print(check)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    print(frame)
    time.sleep(3)  # how much delay the video will play
    cv2.imshow("Capturing", frame)
    cv2.waitKey(0)
    video.release()
    if key == ord('q'):
        break
    print(a)
cv2.destroyAllWindows()
```