

# Daily Assessment 2020

Date: 30/05/2020

Course: Logic Design

Topic: Application of programmable logic controllers

Name: Jyoti S. Dhanu

Vsn: 4AK17E1039

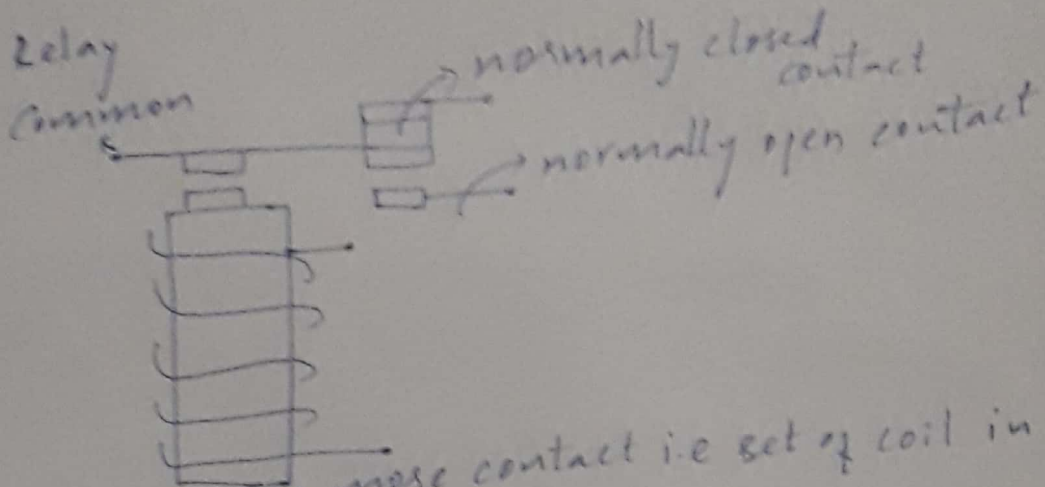
GitHub repository: jyoti-conceal

## forenoon session details

### Report

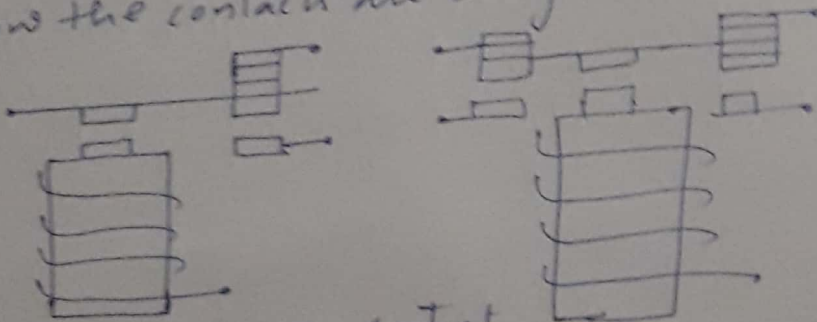
control relay

Common



so this we can use more contact i.e. set of coil in this we use only one terminal.

How the contacts are designed.

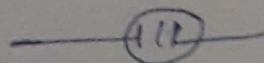


Rotates a common contact b/w the normally closed & normally open

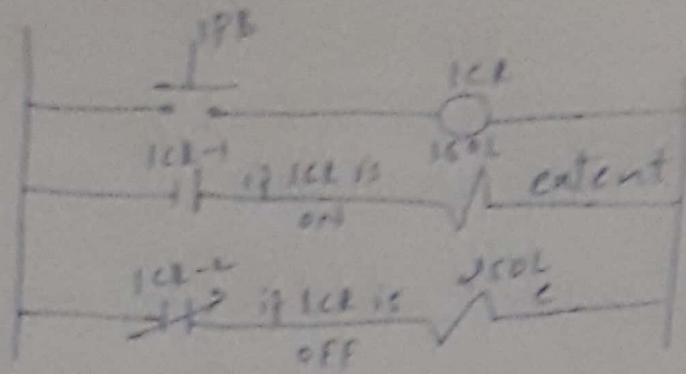
pulls a shorting bar b/w the normally closed & the normally open contact.

NC  
normally close

normally open



## Numerical simulation value



- 25OL is currently energized
- currently 1CR-2 is true & IPB & 1CR-1 is false
- with IPB closed 1CR & 15OL gets energized & IPB & 1CR-1 is ON & 1CR-2 is OFF.

Date: 30/05/2020  
Course: python

Topic: python for image & video  
processing using open cv

### Afternoon session details

#### # python for image & video processing with open cv

- 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")`

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

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

`faces = face_cascade.detectMultiScale(gray_img, scaleFactor=  
1.05, minNeighbors=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)))
cv2.imshow('gray', resized)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

• Capturing videos.

```
import cv2
video = cv2.VideoCapture(0)
check_frame = video.read()
print(check)
print(frame)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
time.sleep(3)
cv2.imshow("capturing", frame)
cv2.waitKey(1)
if key == ord('q'):
    break
print(a)
video.release()
cv2.destroyAllWindows()
```

← a=0  
while true:  
a=a+1: