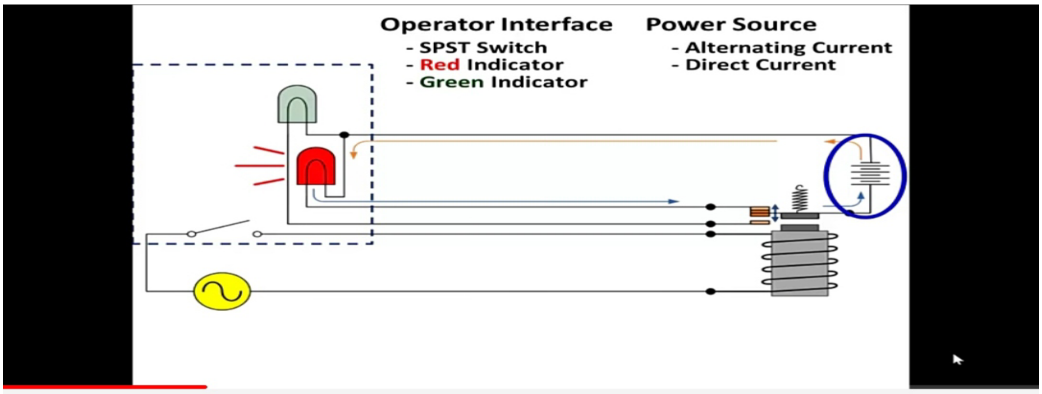
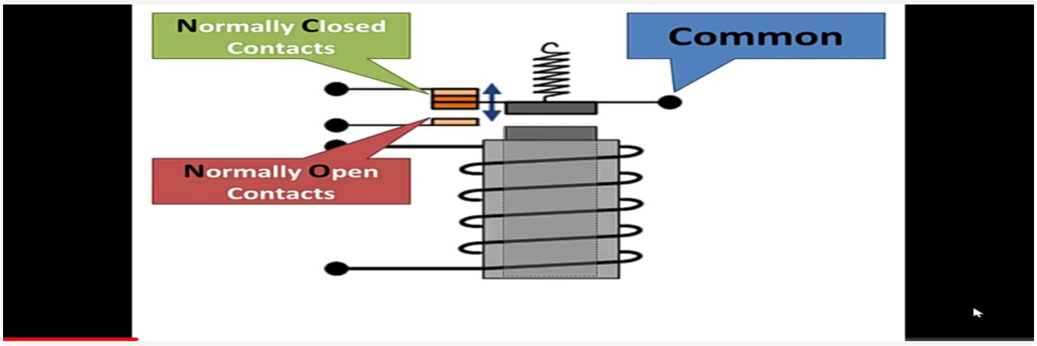
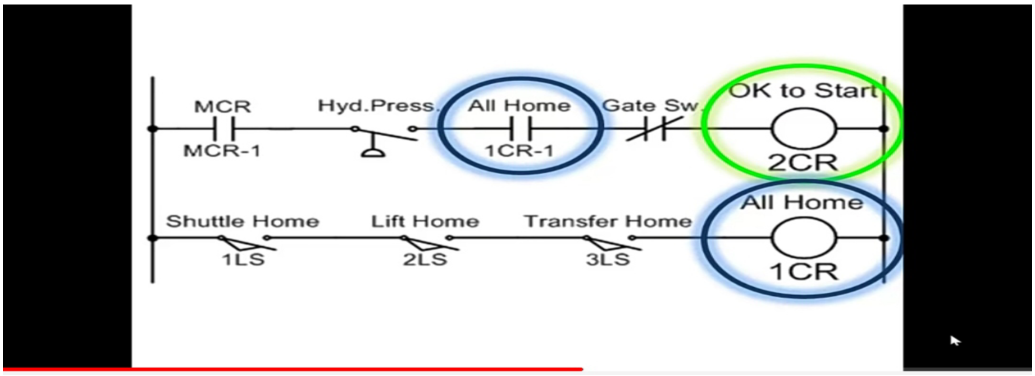
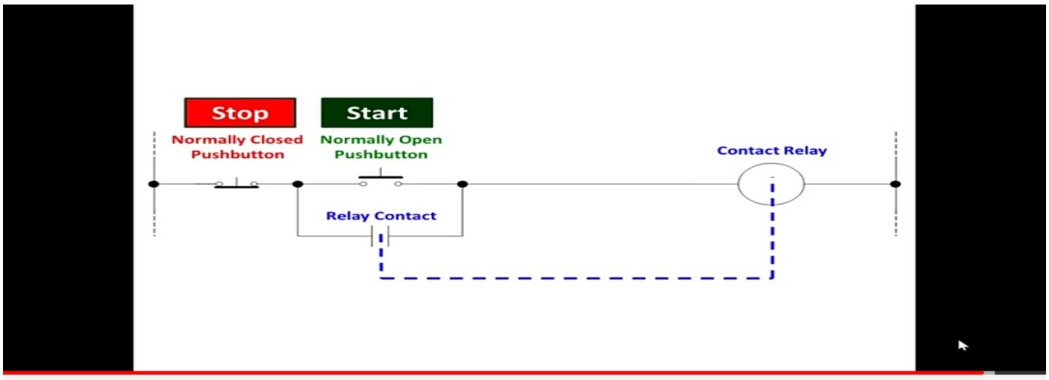
Date:	30-05-2020	Name:	Rajeshwari Gadagi
Course:	Logic design	USN:	4AL17EC076
Topic:	Allpication of programmable logic controllers	Semester and section:	6th sem and B sec



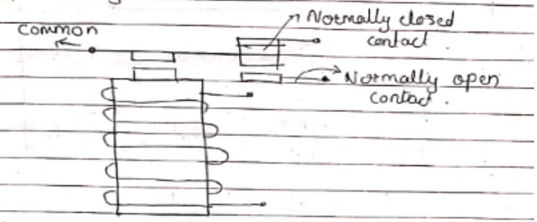






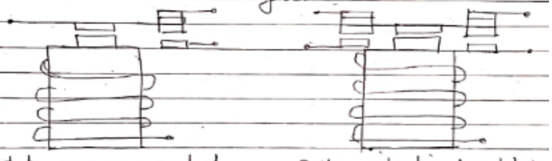
Applications of Programmable logic Controllers.

* Control Relay



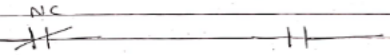
In this we can use more contact. i.e., set of coil.
In this we use only one terminal

How the contacts are designed



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

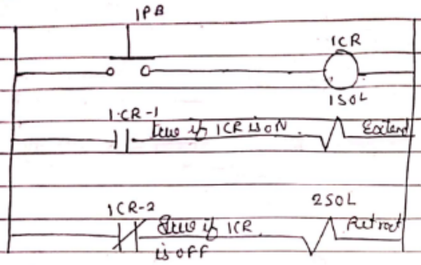
Pulls a shorting bar b/w the normally closed & the normally open contact.



Normally close

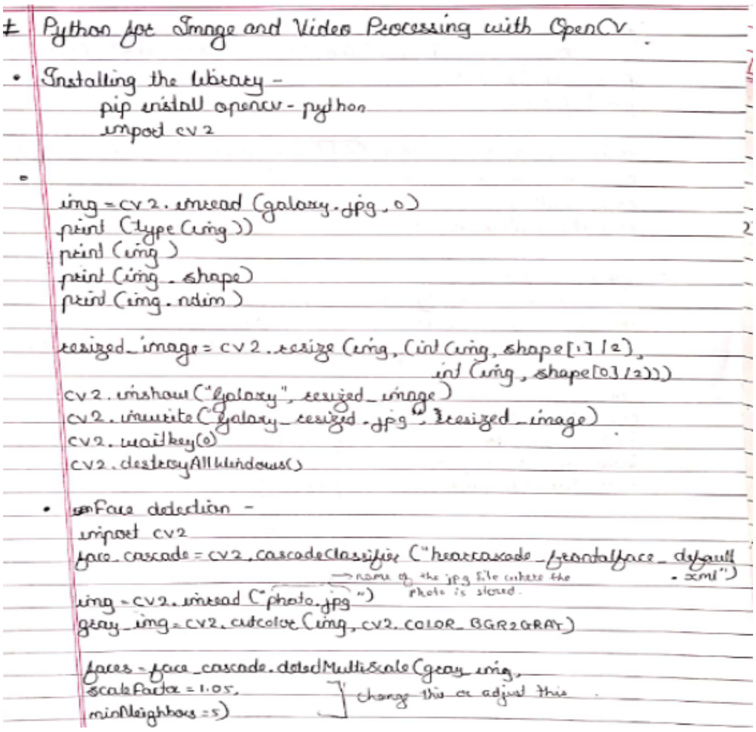
Normally open

* Numerical simulation value:



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

Date:	30-05-2020	Name:	Rajeshwari Gadagi
Course:	Python programming	USN:	4AL17EC076
Topic:	Python for image and video processing using open cv	Semester and section:	6th sem and B sec



Python for Image and Video Processing with OpenCV.

- Installing the library –

```
pip install opencv-python
import cv2
```

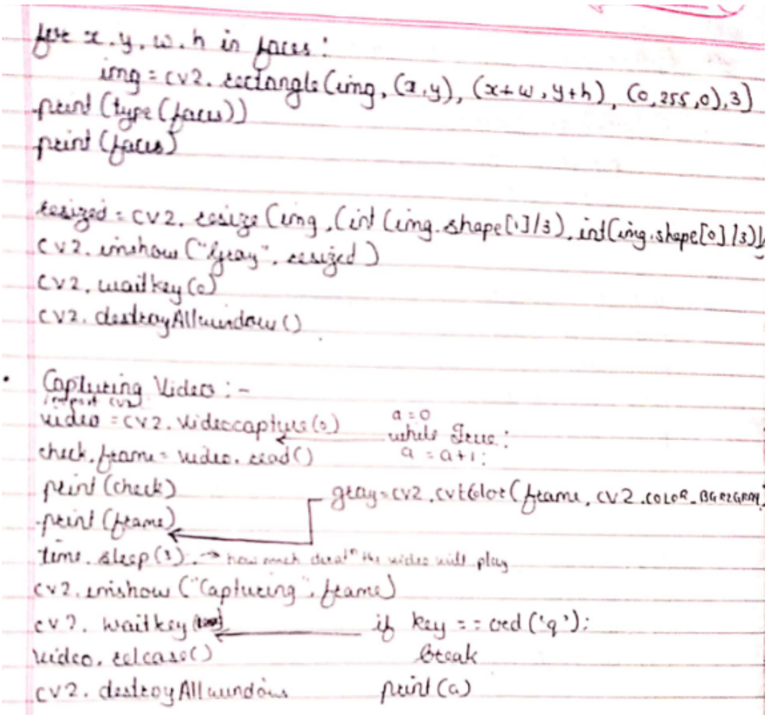
```
img=cv2.imread(galaxy.jpg,0)
print (type(img))
print (img )
print (img.shape)
print (img.ndim )

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
img=cv2.imread ("photo.jpg")     photo is stored.
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.destroyAllwindow()
```

- Capturing Video :–

```
import cv2
video=cv2.Videocapture(0)            a=0
                                     while True:
check,frame=video.read()              a=a+1;
print (check)
print (frame)           gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
time.sleep(3) → how much durat.n the video will play
cv2.imshow("Capturing", frame)
cv2.waitkey(1)                       if key==ord('q'):
video.release()                        break
cv2.destroyAllwindows                 print(a)
```