

DAILY ASSESSMENT REPORT

Date:	30/05/2020	Name:	Abhishek M Shastry K
Subject:	Logic Design	USN:	4AL17EC002
Topic:	1] Applications of Programmable logic controllers	Semester & Section:	6 th 'A'
Github Repository:	AbhishekShastry-Courses		

FORENOON SESSION DETAILS

Image of session

The image displays two screenshots of a YouTube video titled "What is a PLC? PLC Basics Pt1" by UG & PG Courses. The video is 11:44 long and has 1,658,253 views as of November 5, 2012. The first screenshot shows a complex PLC ladder logic diagram with various components like a power supply, system lights, and motor starters. The second screenshot shows a diagram of pushbutton symbols, including a red "Stop" button (Normally Closed Pushbutton) and a green "Start" button (Normally Open Pushbutton), along with a relay contact symbol.

Report

Programmable logic controllers

- **PLC** originated in the late 1960s in the automotive industry in the US and were designed to replace relay logic systems.
- **Relays** are switch that open and close circuits electromechanically or electronically. Relays control one electrical circuit by opening and closing contacts in another circuit. As relay diagrams show, when a relay contact is normally open (NO), there is an open contact when the relay is not energized. When a relay contact is Normally Closed (NC), there is a closed contact when the relay is not energized. In either case, applying electrical current to the contacts will change their state.



Fig: Electro-mechanical Relay

- A programmable logic controller (PLC) or programmable controller is an industrial digital computer which has been ruggedized and adapted for the control of manufacturing processes, such as assembly lines, or robotic devices, or any activity that requires high reliability, ease of programming and process fault diagnosis.
- PLCs can range from small modular devices with tens of inputs and outputs (I/O), in a housing integral with the processor, to large rack-mounted modular devices with a count of thousands of I/O, and which are often networked to other PLC and SCADA systems.

- PLCs were first developed in the automobile manufacturing industry to provide flexible, rugged and easily programmable controllers to replace hard-wired relay logic systems. Since then, they have been widely adopted as high-reliability automation controllers suitable for harsh environments.
- A PLC is an example of a "hard" real-time system since output results must be produced in response to input conditions within a limited time, otherwise it will result in unintended operation.
- Regular programmable logic controller consists of:
 - ✓ A processor unit (CPU) which interprets inputs, executes the control program stored in memory and sends output signals.
 - ✓ A power supply unit which converts AC voltage to DC.
 - ✓ A memory unit storing data from inputs and program to be executed by the processor.
 - ✓ An input and output interfaces, where the controller receives and sends data from/to external devices.
 - ✓ A communications interface to receive and transmit data on communication networks from/to remote PLCs.
- Some of the applications of PLC are:
 - ✓ Glass Industry - The production of glass is an elaborate and sophisticated process so the companies involved often use PLCs with the bus technology in its control mode. Overall, the PLC is applied in both analogue data recording in the glass production, and in digital quality and position control.
 - ✓ Paper industry - In the paper industry, PLCs are used in various processes. These include controlling the machines that produce paper products at high speeds. For instance, a PLC controls and monitors the production of book pages or newspapers in offset web printing.
 - ✓ Cement manufacturing - A distributed control system comprised of PLC in its user mode and a configuration software are used in the industry's production and management processes. The PLC in particular, controls ball milling, coal kiln and shaft kiln.

Date:	30/05/2020	Name:	Abhishek M Shastry K
Course:	The Python Mega Course: Build 10 Real World Applications	USN:	4AL17EC002
Topic:	1] Python for Image and Video Processing with OpenCV	Semester & Section:	6 th 'A'
Github Repository:	AbhishekShastry-Courses		

AFTERNOON SESSION DETAILS

Image of session

The screenshot displays a Udemy video player interface. The video title is "222. Capturing Video". The video content shows a man wearing a hat, with a green bounding box around his face. The video player interface includes a progress bar, course content, overview, Q&A, bookmarks, and announcements. Below the video player is a Windows taskbar with various application icons and system tray information.

The video player shows the following code in the background:

```

1 while True:
2     check, frame = video.read()
3     print(check)
4     print(frame)
5     gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
6     cv2.imshow("Capturing",gray)
7     key=cv2.waitKey(1)
8     if key==ord('q'):
9         break
10    print(a)
11    video.release()
12    cv2.destroyAllWindows()

```

The Windows taskbar shows the following information:

- Search bar: Type here to search
- Taskbar icons: File Explorer, Edge, Chrome, VS Code, Word, etc.
- System tray: U: 0.00 MB/s, D: 0.00 MB/s, ENG, 16:11, 30-05-2020

Below the taskbar is a Visual Studio Code editor window showing the file explorer, explorer, and terminal. The file explorer shows the following files:

- face_detector.py
- face_detector
- EX1 (Batch Processing)
- galaxy.jpg
- im_100x100.py
- kangaroos-rain-australia_71370_99...
- Lighthouse.jpg
- Moon sinking, sun rising.jpg
- resized_galaxy.jpg
- resized_kangaroos-rain-australia_7...
- resized_Lighthouse.jpg
- resized_Moon sinking, sun rising.jpg
- Face_detector
- face_detector.py
- haarcascade_frontalface_default.xml
- news.jpg
- photo.jpg
- Im_processing
- Galaxy_resized.jpg
- galaxy.jpg
- image_processing.py
- video_capture
- video_capture.py

The explorer shows the following code in the background:

```

1 import cv2
2
3 face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
4
5 img = cv2.imread("photo.jpg") #Default as color image
6 gray_img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
7
8 faces = face_cascade.detectMultiScale(gray_img,
9     scaleFactor=1.1,
10    minNeighbors=5)
11
12 for x,y,w,h in faces:
13     img = cv2.rectangle(img, (x,y),(x+w,y+h),(0,255,0),4)
14
15 print(type(faces))
16 print(faces)
17
18 resized_img = cv2.resize(img, (int(img.shape[1]/3),int(img.shape[0]/3)))
19
20 cv2.imshow("Gray",resized_img)
21 cv2.waitKey(0)
22 cv2.destroyAllWindows()

```

The terminal shows the following output:

```

PS D:\udemy_pycodes\OpenCV\Im_processing> cd ..
PS D:\udemy_pycodes\OpenCV> cd Face_detector
PS D:\udemy_pycodes\OpenCV\Face_detector> python face_detector.py
<class 'numpy.ndarray'>
[[152  84 384 384]]

```

Report

Python for Image and Video Processing with OpenCV

- **OpenCV** (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel. The library is cross-platform and free for use under the open-source BSD license. In simple language it is library used for Image Processing. It is mainly used to do all the operation related to Images.
- The images are stored in the form of **n-dimensional numpy array**.
- Some of the functions used for image processing are:
 - ✓ **cv2.imread ()** method loads an image from the specified file.
 - ✓ The shape of an image is accessed by **img.shape**. It returns a tuple of the number of rows, columns, and channels (if the image is color).
 - ✓ Resizing an image means changing the dimensions of it, be it width alone, height alone or both. Also, the aspect ratio of the original image could be preserved in the resized image. To resize an image, OpenCV provides **cv2.resize ()** function.
 - ✓ **cv2.imshow ()** method is used to display an image in a window. The window automatically fits to the image size.
 - ✓ **cv2.imwrite ()** method is used to save an image to any storage device. This will save the image according to the specified format in current working directory.
 - ✓ **waitKey()** is a keyboard binding function. Its argument is the time in milliseconds. The function waits for specified milliseconds for any keyboard event. If you press any key in that time, the program continues.
 - ✓ **destroyAllWindows ()** function will destroy all of the HighGUI windows.
- Some of the functions used for video capturing are:
 - ✓ **cv2.VideoCapture ()** is used to get a video capture object for the camera.
 - ✓ **Video.release ()** function Closes video file or capturing device.
 - ✓ Videos are captured in terms of frames, the number of frames to be saved can be adjusted using **waitKey ()** function.
- For batch processing of images **glob** library is used which finds all the pathnames matching a specified pattern, although results are returned in arbitrary order in the form of list.