**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **30th May 2020** | **Name:** | **Varshini MN** |
| **Course:** | **Logic design** | **USN:** | **4AL16EC089** |
| **Topic:** | **Applications of Programmable logic controllers** | **Semester & Section:** | **8th B** |
| **Github Repository:** | **varshinimn-test** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report:**  **Programmable Logic Controller (PLC):**  Programmable Logic Controller (PLC) is a special computer device used in industrial control systems. Due to its robust construction, exceptional functional features like sequential control, counters and timers, ease of programming, reliable controlling capabilities and ease of hardware usage – this PLC is used as more than a special-purpose digital computer in industries as well as in other control-system areas. Most of the industries abbreviate these devices as “PC” but it is also used for personal computers; due to this, many manufacturers named these devices as PLCs. **Applications of PLC in Glass Industry:** With the development of PLC and increasing demand in the real world, the control mode of the programmable-logic controller with an intelligent device is applied in the glass industry. In making of a float glass, PLC itself cannot finish some controlling tasks because of the complexity of the control system and processing of huge data. For the production of glass, we make use of bus technology to construct the control mode of a PLC with a distributed-control system. This control system deals with analog controlling and data recording; the PLC is also used for digital quality control and position control.  From the year 1980 the Programmable-logic controllers are in use in the glass industry, and they are assembled bit by bit. PLCs are used mainly in every procedure and workshop for controlling the material ratio, processing of flat glasses, etc.  This type of control mode is a big advantage for PLC and DCS for improving reliability and flexibility of the control system. **Applications of PLC in Cement Industry:** Along with the best-quality raw materials, the accurate data regarding process variables, especially during mixing processes within the kiln, ensures that the output provided should be of the best possible quality. Nowadays a DCS with bus technology is used in the production and management industry. By using this existing DCS control system, the PLC is in user mode of SCADA. This mode comprises PLC and configuration software. This SCADA mode comprises the PLC and host computer. The host computer consists of slave and master station. The PLC is used for controlling the ball milling, shaft kiln and Kiln of coal.  To know more about these programmable-logic controllers’ function, you can go through the following project, which is given as a practical example used mostly for industrial automation.  Thus, this article has covered the principle of operation of programmable logic devices or controller and its applications in various industries like glass industry, steel industry and cement industry. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **30th May 2020** | **Name:** | **Varshini MN** | |
| **Course:** | **Udemy** | **USN:** | **4AL16EC089** | |
| **Topic:** | **Python** | **Semester & Section:** | **8th B** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Report:** Adaptive Threshold By using this technique we can apply thresholding on small regions of the frame. So the collective value will be different for the whole frame.  # importing the necessary libraries  import cv2  import numpy as np  # Creating a VideoCapture object to read the video  cap = cv2.VideoCapture('sample.mp4')  # Loop untill the end of the video  while (cap.isOpened()):  # Capture frame-by-frame  ret, frame = cap.read()  frame = cv2.resize(frame, (540, 380), fx = 0, fy = 0, interpolation = cv2.INTER\_CUBIC)  # Display the resulting frame  cv2.imshow('Frame', frame)  # conversion of BGR to grayscale is necessary to apply this operation  gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  # adaptive thresholding to use different threshold  # values on different regions of the frame.  Thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C,  cv2.THRESH\_BINARY\_INV, 11, 2)  cv2.imshow('Thresh', Thresh)  # define q as the exit button  if cv2.waitKey(25) & 0xFF == ord('q'):  break  # release the video capture object  cap.release()  # Closes all the windows currently opened.  cv2.destroyAllWindows()   Smoothing: Smoothing a video means removing the sharpness of the video and providing blurriness to the video. There are various methods for smoothing such as cv2.Gaussianblur(), cv2.medianBlur(), cv2.bilateralFilter().  For our purpose, we are going to use cv2.Gaussianblur().  # importing the necessary libraries  import cv2  import numpy as np  # Creating a VideoCapture object to read the video  cap = cv2.VideoCapture('sample.mp4')  # Loop untill the end of the video  while (cap.isOpened()):  # Capture frame-by-frame  ret, frame = cap.read()  frame = cv2.resize(frame, (540, 380), fx = 0, fy = 0,  interpolation = cv2.INTER\_CUBIC)  # Display the resulting frame  cv2.imshow('Frame', frame)  # using cv2.Gaussianblur() method to blur the video  # (5, 5) is the kernel size for blurring.  gaussianblur = cv2.GaussianBlur(frame, (5, 5), 0)  cv2.imshow('gblur', gaussianblur)  # define q as the exit button  if cv2.waitKey(25) & 0xFF == ord('q'):  break  # release the video capture object  cap.release()  # Closes all the windows currently opened.  cv2.destroyAllWindows() | | | |