Mechatronic Systems Design – Exercise 1

1. Blink LED connected to pin 13
2. Blink 3 LEDs (11,12,13) alternatively
3. Read the switch connected to pin 7 and use LED in pin 13 to indicate the status of the switch
4. Use 3 switches (7,8,9) to turn on/off the 3 LEDs
5. Vary the intensity of R, G and B individually in RGB LED connected in pins 3, 5 and 6 (PWM pins)
6. Combine various intensities of R, G and B and get different colours
7. Print a text in serial monitor
8. Connect and read a potentiometer in pin. A2 and display the read value in serial monitor (use bread board to connect potentiometer to induino)
9. Vary the intensity of R in RGB LED in pin. 3 using a potentiometer in pin A2
10. Display a text in LCD (use bread board to connect LCD to Induino according to the connection table shown in class). Connect pin 3 of LCD to GND directly (No contrast adjustment)
11. Read the LDR connected to pin A3 and display it in LCD and serial monitor
12. Set a threshold on the value read from LDR and accordingly turn on/off an LED. Also display the value of LDR and status of LED in different rows of LCD and serial monitor.
13. Connect a servo to pin 5 and sweep between its min and max positions
14. Identify the working range of the servo given to you
15. Control the position of servo using a potentiometer
16. Write a code for 3 bit binary counter using 3 LEDs
17. Write a code to count the number of times a switch is pressed and display it in LCD
18. Write a code to count the number of times your finger crosses just above the LDR, within a known time (5s) and display the count in LCD
19. Write a code to continuously receive a string of format "a***b***c" (* represents numbers) through serial port, extract the numbers from the string, save them in separate variables and display the numbers in LCD
20. Communicate between two Induinos in duplex mode (Two teams combined). Connect Rx and Tx of two Induinos using wires, make one the master and another the slave. Both master and slave should have a LDR and a POT connected with suitable circuits.
    - Master should read LDR and POT values and start sending data in the format "aLDRbPOTc" (LDR – value read from LDR, POT – value read from potentiometer). Slave should wait until master sends data.
    - Once slave receives data, extract LDR and POT values that the master has sent, display them in LCD, read LDR and POT values, make it to a string of same format and send to master. Master should wait until slave sends data.
    - This whole process should continue indefinitely

Note: Create a separate folder for this course and save all the programs separately for evaluation.

Scanned by CamScanner

MECHATRONIC SYSTEMS DESIGN LAB

Exercise 2

1. Use the switch 7 to rotate the servo in clockwise direction and switch 8 for counter-clockwise direction from initial position of servo (90 deg). Every single switch press and release event should correspond to 1 degree rotation.
2. Test the proximity sensor provided to you.
3. Write a code to Turn-ON a LED when the proximity sensor senses an obstacle and vice-versa.
4. Write a code to stop a servo motor (which is already sweeping between its maximum and minimum range) when the proximity sensor senses an obstacle and continue sweeping when the obstacle is removed.
5. Write a code to start/stop servo sweeping by receiving commands (Ex: 1- ON, 2-OFF) serially and wait till the user enters a valid command.

   Note: Do nothing when an invalid input is entered. Brief instruction may be provided every time before the user enters the input.

Scanned by CamScanner

MECHATRONIC SYSTEMS DESIGN LAB

Exercise 3

1. Test the electromagnet provided to you.
2. Write a code to Turn-ON the electromagnet when the proximity sensor senses an obstacle and vice-versa.
3. Calibrate the colour sensor manually. Write a function *ManualCalibrate()* which when called should do the following.
    a. Position the colour sensor over the white patch stuck to the base platform (by rotating base motor).
    b. Run the following algorithm in loop until the user calibrates and enters a stop command (Display the valid stop command. Ex. Press 1 once the calibration is done).
        i. Shine red, green and blue colours with maximum intensity (keep all three potentiometers at its minimum position) individually using tri-colour LED, read the corresponding reflected intensities using LDR and store them in three different variables. (Important - Give a small delay 50-100 ms every time before reading the LDR value)
        ii. Print the read values in the following format in LCD
        R:_____ G:_____ B:_____
        iii. Find the max value (minimum intensity) of the three and adjust the potentiometers to get equal values in all R, G and B (Follow manual calibration procedure explained in class)
        iv. Give a small delay
    c. Once user enters the stop command, exit the above said loop
4. Write a code to auto calibrate the colour sensor. Convert this code to a function *AutoCalibrate()* which when called should do the following.
    a. Set all the potentiometers of RGB LED to its minimal resistance position. **(Note: To be done manually)**
    b. Position the colour sensor over the white patch stuck to the base platform (by rotating base motor).
    c. Read the LDR values by shining R, G and B LEDs individually and store the corresponding reflected intensity values in three different variables.
    d. Find the max value (min intensity) of these three.
    e. Now, vary the intensities of the other two LEDs (analogWrite from 0-255) so that the intensities received from all three are equal.

Scanned by CamScanner

Note: You may print the status of calibration and the calibrated values in LCD/Serial monitor to understand what is happening at each level

5. Write a program to prompt the user to choose manual or auto calibration and accordingly use the functions to do colour calibration. Once the calibration is done, sweep the base servo within its operating range and display the colour sensed at each step. Display the R, G and B values in LCD.

Expected project demonstration flow

1. Start the program
2. Prompt the user for auto or manual calibration or no-calibration (if calibrated already)
3. Finish calibration and again prompt user for pick up and drop colour location
4. Search for the pick-up colour location using colour sensor (using base motor) and stop once reached.
5. Start the arm motor from its initial position until it senses the object
6. Energize the electromagnet and wait for a pre-defined time (5sec)
7. Retract the arm back to its initial position and search for the destination colour location.
8. Once destination location is reached, bring the arm down, drop the object, retract the arm and go to step 2.

Instructions:

- Save all the codes for evaluation purposes.
- Additional sophistications can be brought in to the project during the demonstration other than what is expected which may have a weightage during evaluation.
- Any similarity in codes between teams will be considered as malpractice.
- All the team members should be able to explain every part of the project during demonstration.

Scanned by CamScanner