Course No: EEE 402

CONTROL SYSTEM I LABORATORY

Name of the Project

MAZE SOLVER ROBOT WITH PID CONTROL & COLOR SENSING

Submitted to —

Dr. Celia Shahnaz
Associate Professor, Department of Electrical & Electronic Engineering
Bangladesh University of Engineering & Technology

Mrs. Fariah Hayee
Lecturer, Department of Electrical & Electronic Engineering
Bangladesh University of Engineering & Technology

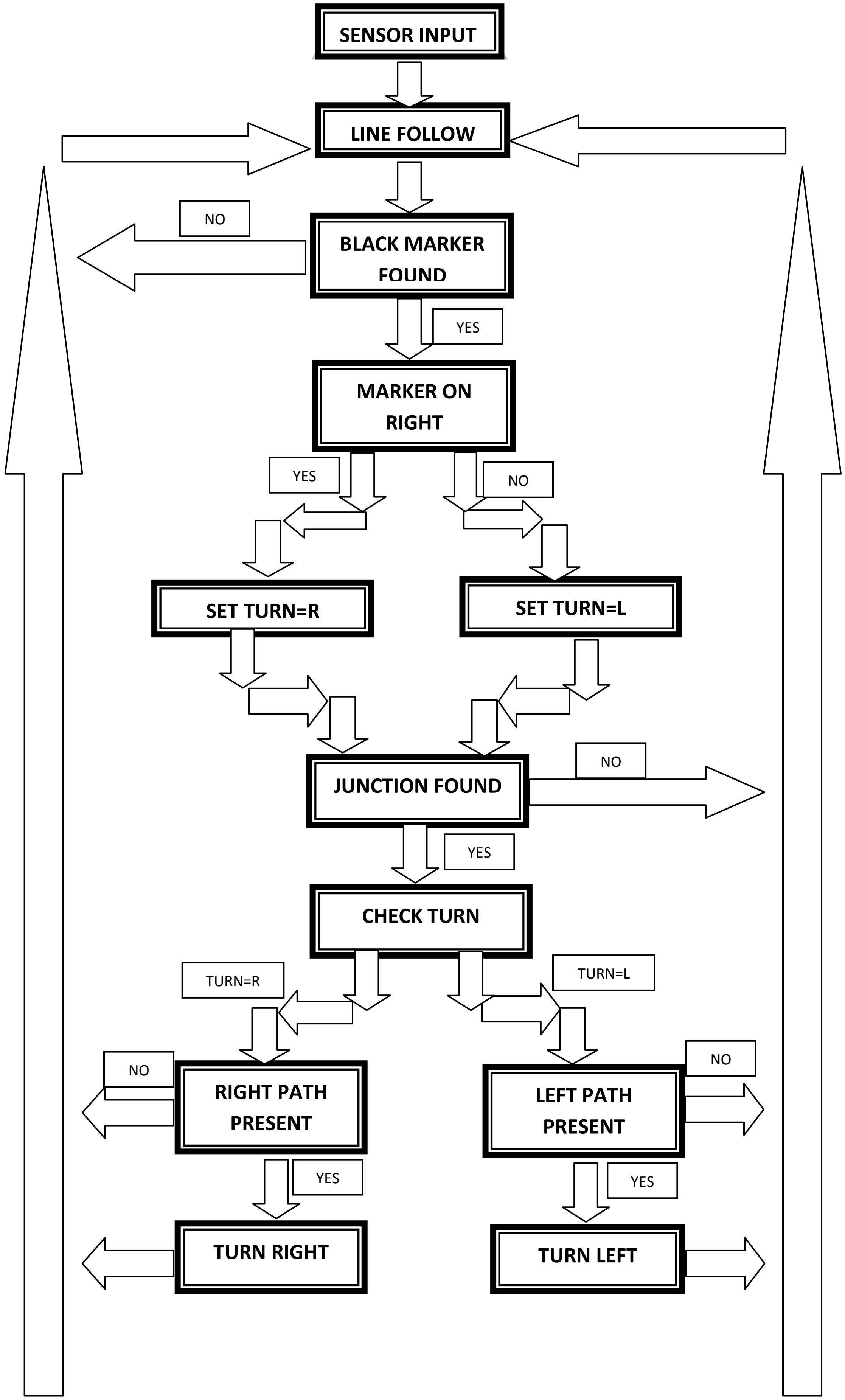
Submitted By

Raziur Rahman	***Std. No: 0806010***
Md. Rashed Hassan Bipu	***Std. No: 0806012***
Saugato Rahman Dhruba	***Std. No: 0806018***
Abul Hasan Fahad	***Std. No: 0806021***
A. S. M. Khalid Hasan	***Std. No: 0806022***
Monjurul Feeroz Meem	***Std. No: 0806028***

ABSTRACT

In this project, we have implemented the Maze-solving Robot with PID control & Color Sensing. Here, we have used an LDR Sensor array to sense the track to follow. Also, the LDR is used to differentiate among colors — Green, Black & White. Before the robot is set to solve the maze, the threshold values for the three colors are stored by calibration. After that, the Robot solves a circular maze while following the white line as path and turning at the junctions following a black marker. The black marker's purpose is that- If the sensor detects a black marker on either side, then it stores a value in a variable to indicate the direction of the next turn. After the robot reaches a junction, it uses this value to turn in that direction. For steady line following & turning we have used the PID Control algorithm. The working values of Kp, Ki & Kd are set by using the trial & error method. For controlling the speed of the robot, we have used the fast PWM method of ATmega32. The fast PWM differs from the other PWM options by its single-slope operation. At the end of the maze, the robot enters a region of the maze where all sensors detect black color. This is the final destination and the robot stops.

PROCESS FLOW CHART:



COMPONENTS

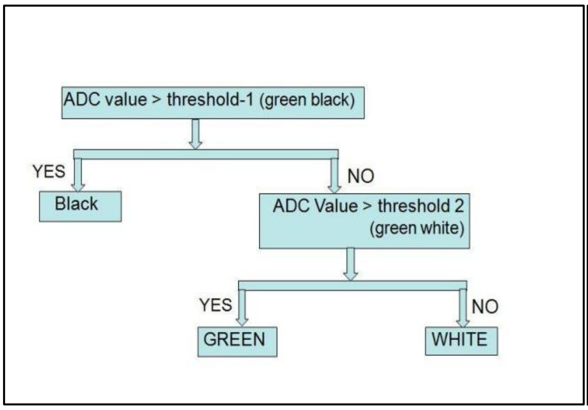
- LED
- LDR
- 7805
- L293D
- ATMEGA32
- GEAR MOTOR
- WHEEL
- PCB
- CASTER WHEEL
- BATTERY
- SWITCH
- RESISTANCE
- CAPACITOR
- PORTS, WIRES, HEAT-SINKS

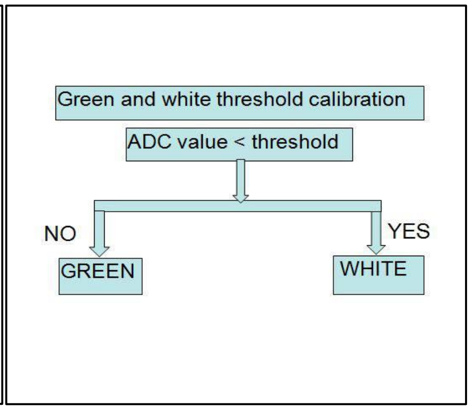
HOW THE SENSOR WORKS

- In our project we have used LED-LDR arrangement for sensor. The purpose of the sensor is to differentiate between Green (background color), White (line), Black(marker) colors. The main idea here is that, the light reflection from different color is different, maximum for white, minimum for black and somewhere in the middle for green. So, depending on the reflecting light the drop across the LRD is different for different color. This voltage is feed to the ATMEGA32 for further calculation.
- For ambient light protection & noise immunity, we have encircled the LDRs with black tape.
- The position diagram of the sensor is given below-
- 

- Here the sensor ‘1’ and ‘7’ can differentiate between white,black and green and check for markers and rest of the sensors can only detect green/white and aid in line following.

- Here, Sensor ‘1’ and ‘7’ we have set two threshold voltages, ‘’threshold-1’’ for differentiating green and black, and ‘’threshold-2’’ for differentiating green and white. Others have only one threshold. Here we have used the ADC feature of ATMEGA32 for calculating threshold.





- The ATmega32 features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows 8 single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND).
- At first calibration when switch 2 is pressed the the sensor is placed in a way that ‘sensor-7’ is on a black marker.ADC takes the reading of the sensors and set threshold for green/white for the 5 sensors and threshold green/white for the corner sensors.Then for second calibration the corner sensors are set above white and threshold-2 is calculated.

The block diagram of the ADC in ATMEGA32 is given below —

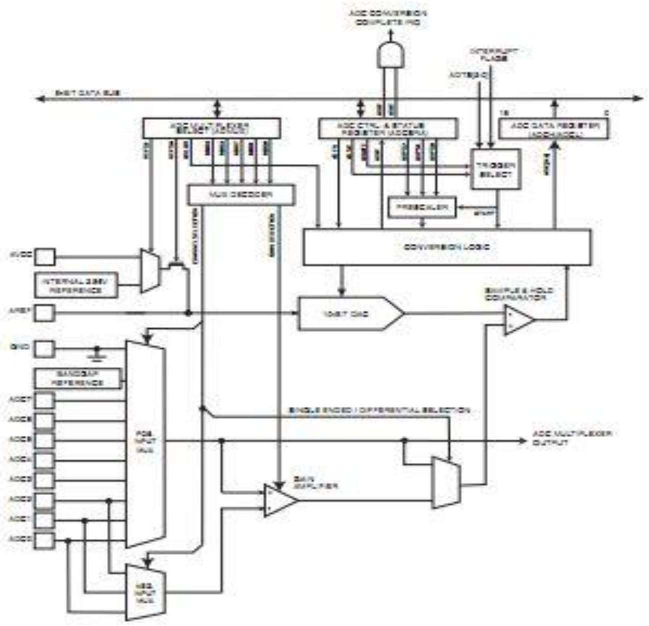


Figure: ADC Block Diagram

Here we have only used 7 ADC for 7 Sensor.

PID LINE FOLLOW & MAZE SOLVE

PID Control stands for the three terms it associated with — Proportional, Integral & Derivative Control. The brief description is given below —

Target – It is the position the line follower where it always meant to be, that is, the center of the robot.

Current Position – It is the current position of the robot with respect to the line.

Error – It is the difference between the current position and the target. It can be negative, positive or zero.

Proportional – It tells how far the robot is from the line like – to the right, to the extreme right, to the left or a little to the left. Proportional is the fundamental term used to calculate the other two. A term Kp is associated with it.

Integral – It gives the accumulated error over time. It tells us if the robot has been on the line in the last few moments or not. A term Ki is associated with it.

Derivative – It is the rate at which the robot oscillates to the left and right about the line. A term Kd is associated with it.

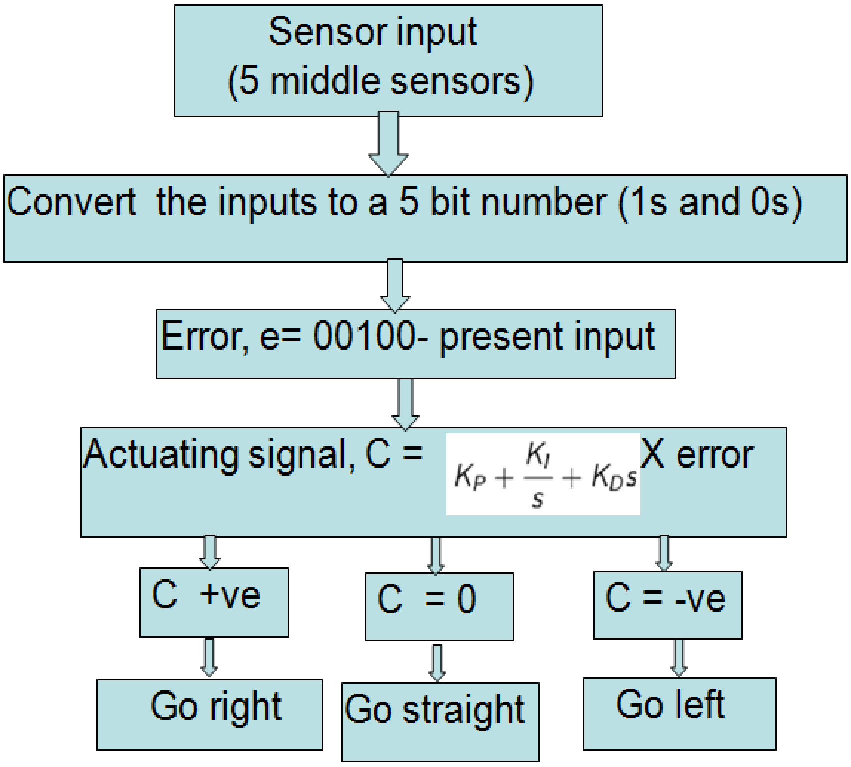
The values of Kp, Ki & Kd are determined using trial & error method repeatedly.

The first requirement in implementing PID for a line follower is calculating the error of the robot. To calculate the error the current position of the robot with respect to the line is needed to know. There are a number of ways of knowing this. We have used the sensor in the center on line as a measure of error. Comparing the value generated when the other sensors on the line with this, we can calculate the error.

We have used the traditional sensor array here. Using an array of sensors, we calculated the error by knowing which sensor is on the line. We have used an array of 7 sensors each placed equally apart from each other. Here, the 2nd to 6th sensor from the left detect the line. Using a sensor array the processor can calculate the error faster as there is no ADC required to be done.

We have used black markers on the track to indicate the next turn. The two sensors on both ends of the sensor array are used to sense the black markers. While following line, if the sensor finds a black marker, it immediately checks on which side the marker is — left or right. For right marker a variable turn is set as — turn = ‘R’ & for left marker, turn = ‘L’. After this, when the next junction is found, it checks for available path in the direction set in turn. If the available path is found, then the robot turns in that direction.

The PID algorithm flowchart is given below —

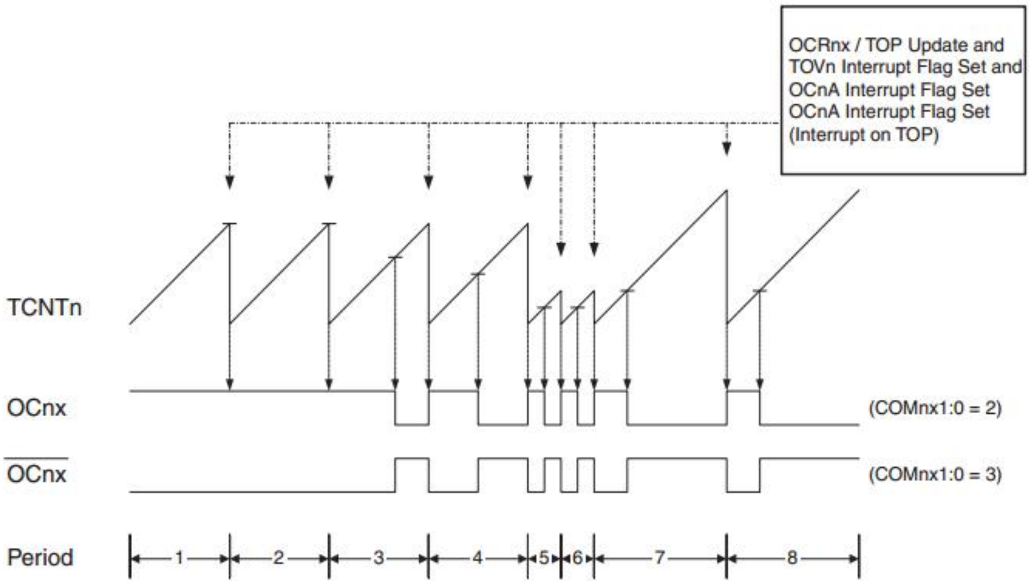


SPEED CONTROL USING PWM

Here, we have used the Fast PWM Mode of ATMega32. The fast Pulse Width Modulation or fast PWM mode (WGM13:0 = 5,6,7,14, or 15) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM options by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. In this inverting Compare Output mode output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the other PWM modes using dual-slope operation.

The PWM resolution for fast PWM can be fixed to 8-bit, 9-bit, or 10-bit, or defined by either ICR1 or OCR1A. We have used the 8-bit resolution here for each of the two motors driving wheels. We have used the non-inverting mode of Compare Output where both COM1A1 & COM1B1 = 1 (PRESCALER=64 MODE 5).

The timing diagram of Fast PWM Mode is shown below —



COST ANALYSIS

Components	Quantity	Price per unit	Price(BTD)
LED	7	2	14
LDR	7	10	70
7805	1	10	10
L293D	1	100	100
ATMEGA32	1	350	350
GEAR MOTOR	2	150	300
WHEEL	2	100	200
PCB	1	250	250
CASTER BALL	1	40	40
BATTERY	3	300	900
SWITCH	4	5	20
RESISTANCE	14	-	10
CAPACITOR	2	2	4
PORTS,WIRES,HEAT SINKS	-	-	100
MISCELENIOUS	-	-	200
TOTAL COST	-	-	2568

DISCUSSION

In the course of building the desired robot, we faced a number of problems —

- Unequal speed of two motors used to drive the wheels because of construction related problems. This problem was solved using two different PWM values for each motor.
- The quality of available PCB is quite poor. So, we had to manually correct some connections on the board even after it was printed.
- Mechanical designing problems were faced because of our inexperience in this field.
- Optimum Weight-to-Power ratio was very difficult to determine. This fact needs more attention.
- We had to go through a huge number of trial & error methods for selecting suitable values of Kp, Ki & Kd.

CONCLUSION

Our project has instilled a great deal of enthusiasm in us for practical Control System Design & Analysis. Our stride in engineering has just begun. There is a lot to explore. This project will go a long way in our future success in this field. We have our heart-filled gratitude to our instructors who are really inspiring. Their valuable suggestions have made significant contribution in the completion of the project.