Tivainoc Modovour Line Follower

Filza Shahid
Dept. of Electrical Engineering
University of Engg. and Tech.
Lahore, Pakistan
filzashahid2001@gmail.com

Misbah Ishaq
Dept. of Electrical Engineering
University of Engg. and Tech.
Lahore, Pakistan
misbahbatol6@gmail.com

Haseeb Faraz
Dept. of Electrical Engineering
University of Engg. and Tech.
Lahore, Pakistan
haseebfaraz77@gmail.com

***Abstract*—A line Following robot as it name suggests an electronic machine that can easily detect the path to be followed . The path could be visible such as black or white and invisible such as magnetic field. For such configuration IR(infra-red) sensor is installed to differentiate between the paths but we would be focusing our concern to visible path and in this case it would be black. Signals would then be delivered to the processor which would compute the signals and delivers the path to motor drivers which in turn would move the robot according to the path specified by user. The paper contains the technical, mechanical issues occurred like excessive divergence from the provided path and the best possible solution to these problems like Bang Bang and PID along with their limitations. It also contain the investigations to the hindrances and the methods to avoid them while developing and the controlling the robot.**

***Index Terms*—Line follower,Circuit,Programming,Controller**

I. Introduction

The modern logistics industry is becoming a hub for exciting robotic applications. It was slow to adopt automation at first, but the last few years have seen a rise in robotic solutions to logistics problems. According to Machine Handling and Logistics, ”Some of the largest [logistics] operators in the US have plans to automate almost every physical move in their facilities within the next two to three years.” These applications range from palletizing tasks all the way up to robotic ports, warehousing robots, autonomous picking last mile delivery drones and self-driving vehicles.

With the rapid advances made in the past few years, autonomous and Automatic Parking vehicles are the most obvious example of self-driving technology. A line following robot is self monitoring gadget that uses sensor to detect path which in our case is visible. Therefore it follows a black line drawn on the surface and changes its trajectory according to the path . The main operations of the robot contains :

- Infra red sensor captures the path on the surface. Sensor contains five photo reflectors which in terms have high resolution and great robotics.
- The path is followed without any steering mechanism however the stability and control of the system is developed by algorithms such as pid controller.
- Speed of the robot is also controlled according to the path detection and variation in speed are subjected due to change in path.

In this paper, we intend to share our experiences. Therefore, the line follower robot structure and architecture issues and challenges along with their technical issues and problems will be discussed in section II, III, IV, V and the algorithm of procedure along with programming aspects will be explained in section VII. After that results will be discussed in section VIII. At last, the conclusion will be inked in section IX.

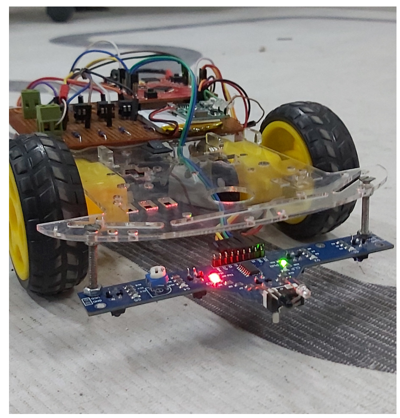

Fig. 1: Line Follower Robot

II. Design of The System

The system is designed on a line following robot chassis which is controlled by a micro controller TIVA that obtains its computational data from infrared sensor. Then TIVA supplies the computed information to motor driver which then directs the path to be followed by robot depending upon the information fed by IR sensor.

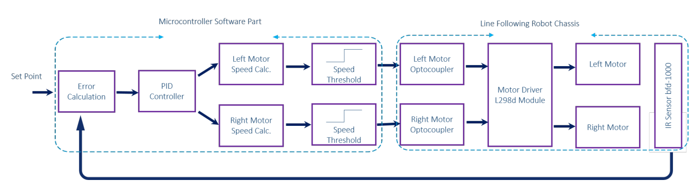

Fig. 2: Block Diagram of Designed System

III. Circuitry of the System

In order to have two separate grounds for the system, an electrical circuit is designed on Veroboard. With the help of this circuit, the ground for TIVA is separated from the ground of motor driver. This is done to secure TIVA as it is current sensitive. The following components are used to design the circuit:

A. Optocoupler

This component is unique in its nature as it couples isolated circuits while using light emitting and light sensitive components.The optocoupler used for the robot is 4n35 manufactured by Vishay Semiconductor Opto Division. Its works on the principle of optical link between two completely isolated devices. The optical link is contained within a chip. A Light Emitting Diode inside the chip shines on a photo-diode, photo-transistor or other photo device. When the photo device sees illumination, the resistance between its terminals reduces.

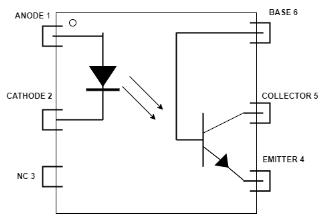

Fig. 3: Circuit Diagram of an Optocoupler

B. Resistors

Since resistors are passive devices i.e absorbing voltages therefore used to avoid excess voltage in the circuit . Moreover pull-down resistors are used since it is an ensuring method that the inputs of digital logic gates and circuits can not self-bias and float about is to either connect the unused pins directly to ground (0V) for a constant low “0” input, (OR and NOR gates) or directly to Vcc (+5V) for a constant high “1” input (AND and NAND gates).

The resistors for pull-down operation are 1k ohms.Furthermore a 220 ohms resistor is also used between the input from TIVA and optocoupler.

C. Block Connectors

These terminal block are a type of insulated electrical connector that connects or secures two or more wires together. They enable the safe connection of wires between circuits.Block connectors used for the robot are green in colour and have two terminals i.e for ground and Vcc.

D. Jumpers

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Since there are three types if jumpers i.e male to male,female to female and male and male to female. The robot uses all of them in its circuitry.

E. Connections

The whole circuit is patched on Veroboard.Optocouplers contain six pins as shown in the figure . The input from TIVA is provided to pin 1 while the second pin contains 210 ohms resistor connected to ground . Pin 3 is left empty. The next pins provides with the output to the actuator which drives the wheels however a pull down resistor is installed at this particular pin. Therefore we collect our signals from the port above the 1k ohms resistor i.e pull down . Pin 5 is supplied by a 5V battery for the functionality of the system and alternatively pin 6 is also left empty. The complete circuit for 1 Optocoupler is shown in figure 4.

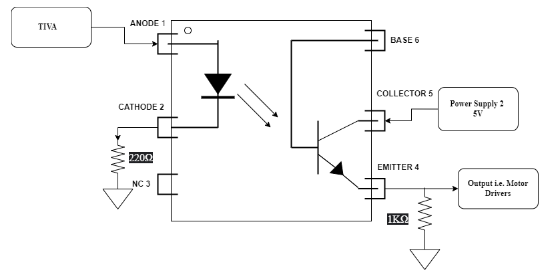

Fig. 4: Circuit Diagram Using a single Optocoupler

4 similar circuits as shown in figure 4 are soldered on Veroboard. Their grounds on both sides are made common. The input on pin 1 is provided from each of 4 pins of TIVA. Pin 5 of four of them is made common and then connected to the 5V input from motor driver. The output from pin 4 is fed to the each of 4 pins on motor driver.

IV. Actuators

An actuator is a device that produces a motion by converting energy and signals going into the system. The motion it produces can be either rotary or linear. An actuator is a device that produces a motion by converting energy and signals going into the system. The actuator could be electrical, pneumatic or hydraulic.

A. Specifications

Actuator installed in robot, also known as Motor Driver, is L298N Dual H Bridge Motor Driver with the following specs.

- Double H Bridge Drive Chip: L298N.
- Logical Voltage: 5V.
- Drive Voltage: 5V-35V.
- Logical Current: 0-36mA.
- Drive current: 2A (MAX single bridge)
- Max Power: 25W.
- Dimensions: 43 x 43 x 26mm.
- Weight: 26g.

B. Working Principle

As optocoupler has 6 pins. The pin4 from 4 optocouplers each provides with output to the motor driver which in turn send the signals to motor A and B and move the wheels. This input provided from optocoupler circuit is in form of 1 and 0 which suggests the functionality of the motion.

The input is provided by TIVA to Optocouplers which in turns supplies these signals to the motor driver hence TIVA and motor drivers are indirectly connected with is the prime purpose of the optocouplers.

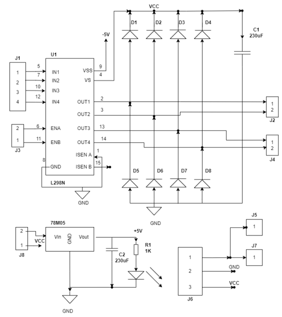

Fig. 5: L298N Motor Driver

V. IR Sensor

An infrared sensor (IR sensor) is a radiation-sensitive opto-electronic component with a spectral sensitivity in the infrared wavelength range 780 nm - 50 µm.The IR sensor used for configuration of the path is BFD-1000 Channel Line Tracking Sensor Module (BFD-1000) is a sensor board designed for use with line follower robots. This module has been sufficient to meet the day-to-day task of tracking, but also with the infrared distance sensor and touch detection sensor, the board makes your robot design able to adapt to the situation easily.

BFD – 1000 specifically designed as a black line (white) especially suitable for the complex of black and white line, cross black and white line detection, it has 6 roads high sensitivity infrared sensor (5 escapes patrol, 1 road), to the recognition of black and white line accurately.

A. Specifications

- input voltage 3.0-5.5V
- Output Type Digital output high and low
- Length (mm) 13
- Width (mm) 5
- Height (mm) 2
- Weight (gm) 12
- Weight 0.016 kg

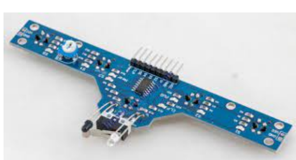

Fig. 6: BFD-1000 IR Sensor

B. Working Principle

The IR sensor senses the path and differentiate between the white and black surface. The ideal state for the sensor is that its centered three sensors are on black line . If the one of these sensor are deviated from the path i.e if the left or right sensor among these threes observed a white line . The sensor then would set an interrupt to the micro controller mentioning the exactness of the interrupt . therefore the controller would provide the driver with such signals that the sensor moves towards its ideal state .

VI. TM4C1233H6PM Micro controller

The Tiva^{TM} C Series ARM Cortex-M4 micro-controllers provide top performance and advanced integration. It offers a 80 MHz Cortex-M with FPU, a variety of integrated memories and multiple programmable GPIO.

A. Features

TM4C1233H6PM Micro-controller has following main features:-

- 32-bit ARM Cortex-M4F Processor core with 80-MHz operation, 100 DMIPS performance and Thumb-2 mixed 16-/32-bit instruction set
- RTOS System Tick Timer
- Nested Vectored Interrupt Controller(NVIC)
- System Control Block (SCB)
- Memory Protection Unit (MPU)
- Floating-Point Unit (FPU) (see page 130)
- On-Chip Memory with 32 KB single-cycle SRAM, 256 KB Flash memory and 2KB EEPROM
- Serial Communications Peripherals
- System Integration
- Advanced Motion Control like PWM, fault inputs, QEI
- ADC, JTAG and ARM Serial Wire Debug

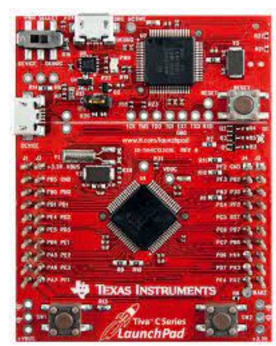

Fig. 7: TM4C1233H6PM Microcontroller

B. Applications

The TM4C1233H6PM microcontroller is targeted for industrial applications, including remote monitoring, electronic point-of-sale machines, test and measurement equipment, network appliances and switches, factory automation, HVAC and building control, gaming equipment, transportation, and fire and security.

VII. Programming

The instructions fed to the motor drivers are provided by the processor itself provided the data was retrieved from the sensor.Therefore, processing of the data takes place and the set of instructions are written for the implementation of the task . The language used for the instructions is C .

A. Environment

Kiel UVsison4 manufactured by subsidiary of Arm Holdings provides project management, run-time environment, build facilities, source code editing, and program debugging in a single powerful environment. Moreover it provides both C/C++ languages to code instructions.

B. Algorithm

For the position control, we are first setting the reference value as '10001' i.e; when the black surface encounters, the sensor's respective bit will be '0' while for the white surface, it shows up with '1'. There are 5 sensors regarded as S1, S2, S3, S4, S5. So, for sensors' output value '10001' indicates that it's on track (as sensors S2, S3, S4 gives 0 for black). If the bits indicated by sensors are '11000' or '11100' or '11110' i.e black is detected on left side, so car is needed to turn left. For this, we need to decrease speed of left motor and increase for right motor. On other hand, if '00011' or '00111' or '01111' is detected by sensor, then black is sensed on right side. So, by decreasing speed of right motor and increasing it for left, car is turned right. This behaviour has been summarized in Figure 10.

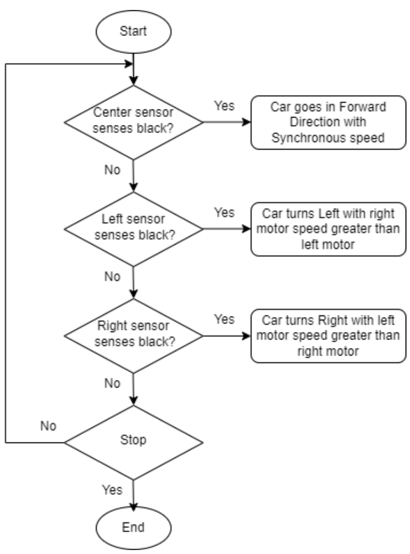

Fig. 8: Flow chart of the system

C. Implementation of PWM

Following steps are followed for the configuration of PWM:

1) **Timer** In order to control the motion of motors, we make use of Timer module. The CLK for this module is enabled. In this case we are using 32-bit TIMER1 which is divided into 2 parts i.e. 16 bit TIMER1A for Motor A and 16 bit TIMER1B for Motor B. 2 output pins of TIVA corresponding to each timer are set. Pin PF2 and PB4 are configured with TIMER1A and pin PF3 and PB5 are configured with TIMER1B. All of these things are enabled in respective registers built for TIMERs configuration on TM4C1233H6PM.
   After that, at first we enable the Periodic mode and then move to PWM Capture mode. We also enable the fine tuner configuration in which we decide to enable which timer at which time.
   Then comes the Interval Load registers for Timer A and B. In them we will load the certain frequency value which will decide the motion of the motor by comparing its value with the value that is stored in Match Registers of both TIMER1A and TIMER1B separately. This determines the duty cycle of the output PWM signal. If the value in interval load matches the value that is present in match register, output interrupt is generated.
2) **Interrupts** In order to configure the output interrupt, we mask the corresponding required pins in Interrupt Mask Register. We are going to configure the interrupt as Edge Triggered and at Falling Edge. For the sake of avoiding any kind of risk, we first clear all kind of interrupts (if there are any which are indicated in Raw Interrupt and Masked Interrupt Status Registers) in Interrupt Clear Register so that no interrupt in repeated over and over again. Also before enabling any kind of Interrupt, we First make sure to disable it to avoid errors in program.
3) **GPIOs** First of all, enable CLK for the ports B and F. Then we enable alternate function for port F pin PF2 and PF3 and port B pins PB4 and PB5 because they are being used by TIMER to generate PWM. Also set Control Port for them as TIMER. Then make these pins as digital enable in DEN and set them in direction register DIR as writable.
   Now, we are taking the 5 inputs from sensor on port A. For that, we first enable the CLK for port A. Then mask the required bits in corresponding DATA Register which contains the Data input obtained from sensor. We are using PA2, PA3, PA4, PA5 and PA6 pins to get input. Make these pins digital enable in DEN register and set their direction in DIR as readable.

D. Implementation of Bang-Bang Technique

Bang bang techniques allows to switch between two states abruptly. It uses binary input to control the system. This technique also tests the presupposed simultaneous checking of the components.Since it is quite a simple mode, therefore its provides flexibility to the user. In our case we used this approach to continuously monitor the system's feedback while the sensor provides with the input to the processor. More accurately, we used if then block to determine the states of the block. The condition states that if the sensor provides with the particular value to the controller(in our case it's Tiva) the controller configure the values and compare with the IT block (if then)and executes the instructions to the motor driver . Hence, IR sensor provides the value to the Tiva and then ,Tiva postulated the value and provide the instruction to the motor driver which directed the robot to move in either straight or left or right direction.

E. Implementation of PID Controller

The robot operates on PID controller due to the limitations in the Bang-Bang Method. Upon using the Bang-bang technique it was observed that the robot followed the path but the divergences was quite high.Moreover,it was also observed that the robot wasn't easily determining the path once it was deviated. A PID controller is a mechanism to regulate the controller in feedback loop. It comprises of a Proportional, Integrator and Derivative parameters. So, to implement this technique we used hit and trial rule to observe the perfect the values of the parametes which are illustrated in the following table. The following equations express how the actual PID functions:

$$\Delta error = error - prev.error \tag{1}$$

This quantity is responsible to slow down the rate of change of the physical quantity when it comes close to the initial point. The derivative term is used to improve the overshoot or how much the system should "over correct".

$$\sum error = i_e rror + error \tag{2}$$

It is responsible for the quickness of response of the system to the change from the initial point. The integral term eliminates the steady state error required by the proportional term.

$$pid = Kp*(error) + Kd*(\Delta error) + Ki*(\sum error) \tag{3}$$

The system calculates the 'error', or 'deviation' of the physical quantity from the initial point, by measuring the current value of that physical quantity using a sensor(s). To get back to the initial point, this 'error' should be minimized, and should ideally be made equal to zero. Also, this process should happen as quickly as possible. Ideally, there should be zero lag in the response of the system to the change in its set point.

Now , setting the values of Kp=0 while other parameters to zero , upon implementing this we could tune the value of Kp .If the robot react slowly or quickly then increase or decrease the value of Kp respectively. After the robot is tuned and works smoothly then add the values of Kd and Ki using hit and trial rule until the robot works perfectly.

VIII. Discussion and Results

We were assigned 4 weeks to complete this project. The weekly progress report for this project is shown below:-

WEEKLY PROGRESS SCHEDULE

Date	WEEKS	MON	TUESDAY-THURSDAY	FRIDAY-SATURADY
22/11/2021	Week1	Purchasing of Equipment along with Components	Testing and Implementation	Assemblying of Components
29/11/2021	Week2	Programming	Bang-Bang Algorithm	Debugging
6/12/2021	Week3	Developing PID Controller	Hit and Trial rule to adjust the values of Parameters	Final Testing and Configuration
13/12/2021	Week4	Template for Paper	Covering The Contents	Submission

Fig. 9: Weekly Progress Schedule

A. Testing of Components and Hardware Implementation

After the purchasing of the components, all components were tested separately. The behavior of motor driver, IR sensor , both motors and optocouplers were observed using different indicating devices. Then robot and all components were properly assembled using jumpers, vero-board, tape etc. The basic micro-controller's programming based operations have been performed on the robot to test the hardware. After eliminating hardware faults (if any) and with all satisfaction related to connections and safety measures, control part of the system has been started.

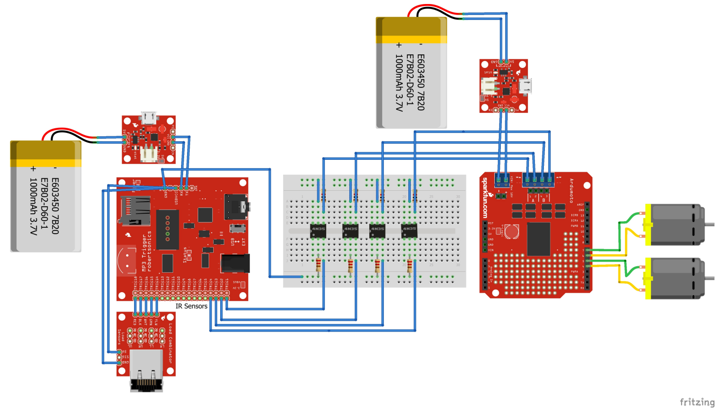

Fig. 10: Complete Circuit Diagram of the System

B. Controlling the Robot

As already has been discussed, the response of line following robot is controlled by two different techniques. Using bang-bang approach, one can simply observed that while tracking the line, if there's any turn or interrupt encounters, the angle of deviation increases. In other words if the system gets unbounded, the response goes too and difficult to achieve the steady state position again. Also, error and settling time increase making the system less efficient.

To overcome the effect of slow rise time and steady state error, we add the proportional constant (K_p) with the value of 200. With K_p the system has now fast rise time and less steady state error(but not eliminated). And it has less effect on the settling time but the system's overshoot increases (when line tracking robot detects the line; and gets sudden response; the required output has been achieved but with some additional value. That's make the robot a bit of out of track for a while and then again in the track and goes on...).

To get rid of this overshoot response of the robot we add another controller parameter called Derivative Constant K_d. Through this the system's overshoot response has been minimized along with the reduction in settling time and no change in steady state error.

To eliminate the steady state error, Integral Constant K_i can be added with very appropriate value. Thus using the PID Controller, the closed-loop system of line tracking robot has fast rise time, no overshoot and no steady state error i.e; the desired response has been achieved.

IX. CONCLUSION

The automated robotics has a wide range of applications now and incoming era of emerging technology. In this paper, a black line/track following robot ”Tivainoc Modovour Line Follower” has been presented from scratch. From IR Sensor, one signal (representing black/white surface) is processed at a time with the help of micro-controller TM4C1233H6PM on TIVA launchpad. The position of robot is controlled through the speed of rotatory wheels by updating the value in the match registers. In other words, both control signals have been generated through duty cycles of PWM(Pulse Width Modulation) for both motors to operate separately. The response of robot has been achieved by PID Controller with no overshoot, fast rise time and no steady state error by selecting the appropriate values for K_p, K_d, K_i manually. Moreover, The response of system using different techniques has been also discussed. The future of line tracking robots/vehicles could be result in an automated parking system (vehicles' automated tracking of the parking path) or following of any desired track.

ACKNOWLEDGMENT

Thereby, upon concluding the article , we would like to thank the faculty of Electrical Department of University of Engineering and Technology for providing us the great environment and atmosphere required for this great initiative. Moreover we would like to express our gratitude to Dr. M. Tahir , Dr. A.R Kashif,Ms. Sheezen Malik and Sir Ali Shafique for guiding us through thick and thin times while developing the robot. Furthermore, we would also like extend our appreciation towards the Lab In-charge Instructors for providing us the work space.

REFERENCES

[1] Tahir, Muhammad and Javed, Kashif, ”ARM microprocessor systems: cortex-M architecture, programming, and interfacing”, tahir2017arm, CRC Press, 2017
[2] Line-Follower-Robot-PID-Control
[3] Reference Designs
[4] l298n-motor-driver-module
[5] research-gate-publications
[6] fritzing.org
[7] app.diagrams.net