

**COMSATS UNIVERSITY ISLAMABAD**

**DEPARTMENT OF COMPUTER SCIENCE**

COMSATS University Islamabad, Vehari Campus



Project Proposal

Semi-Autonomous Entity Carrier Using External Sensors

Version 1.0

***By***

**CIIT/SP18-BCS-093/VHR M. Ammar Tariq**

**CIIT/SP18-BCS-106/VHR Salman Shafiq**

***Supervisor***

**Dr. Ali Shahid**

***Bachelor of Science in Computer Science (2018 – 2022)***

## Table of Contents

[Table of Contents 3](#_Toc66134397)

[Category 5](#_Toc66134398)

[Abstract 5](#_Toc66134399)

[Introduction 5](#_Toc66134400)

[Tools and Technology 6](#_Toc66134401)

[Nano: 6](#_Toc66134402)

[Vim: 6](#_Toc66134403)

[Pip: 7](#_Toc66134404)

[Arduino IDE: 7](#_Toc66134405)

[Open SSH 7](#_Toc66134406)

[Programming Language 7](#_Toc66134407)

[Library 7](#_Toc66134408)

[Python 7](#_Toc66134409)

[RPI.GPIO 7](#_Toc66134410)

[C 7](#_Toc66134411)

[Keyboard.h 7](#_Toc66134412)

[Project Stakeholders and Sponsors 8](#_Toc66134413)

[Block Diagram 8](#_Toc66134414)

[Methodology 9](#_Toc66134415)

[Functionality 9](#_Toc66134416)

[System Limitations/Constraints 10](#_Toc66134417)

[Working on different surfaces 10](#_Toc66134418)

[Scope 10](#_Toc66134419)

[Algorithm 11](#_Toc66134420)

[Gantt Chart 12](#_Toc66134421)

[Future scope 13](#_Toc66134422)

[Reference 14](#_Toc66134423)

Category**:** Robotics / Hardware / AI

## Abstract

Self-navigating object carrying hardware utilizes Raspberry PI [1]. It is fundamentally a robot that follows a specific path or direction and chooses its own navigation-plan interacting with the type of obstacle. The path can be a dark line on the white floor. Its applications start from fundamental homegrown uses to mechanical industry utilization, and so on. The utilization of line following mechanical vehicle is transport the materials starting with one spot then onto the next place in the hemisphere. This robot development totally relies upon the track. In our project presentation our robot will be conveying the bundles or materials starting with one spot then onto the next place utilizing the carrying trolly. The robot can do anything you set them to do in our case we will use it for carrying small objects.

## Introduction

The human labour in the large-scale industries around the world is not as much productive. Our purposed hardware will provide solution to this problem as it will replace human labour and will be able to provide a rather more efficient solution. Our robot uses a microprocessor controller taking input from sensors for its acknowledgement information and feeding back to the microprocessor, which is used for guiding our vehicle. For better understanding of line following robot, prior knowledge of Raspberry PI [1], IR sensors and Motor driver module is necessary.

Raspberry PI [1] is a Linux based single board full fledge computer runs on 32-bit single-core RISC processor, GPU and 2 GB RAM. SD card for its OS and data storage. It can control our mechanical hardware using its GPIO pins.

ATtiny85 [2] microcontroller is one of the small and high-performance AVR microcontroller, which is based on RISC CPU. We will be using it to initialize our code headless as it will be injecting keystrokes into Linux CLI to initialize the controls.

IR sensors is an electronic device used to detect properties of surroundings. It detects using infrared radiation. On coloured surface in our case (black) it does not reflect any light at all hence the output is 0. On the other hand, on white surface the output is 1.

Motor Driver is used to drive motors in any direction and acts as a bridge between the controller and the motor driver.

## Tools and Technology

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| Raspberry PI [1] | Vim / Nano |
| ATtiny85 (Digispark) [2] | Open SSH |
| Driver Hat | Pip |
| DC Motors | Arduino IDE |
| Robot Chassis |  |
| IR Sensor Module |  |
| Insulation Tape |  |
| Conductors |  |
| Battery |  |

### Nano:

GNU nano [3] is a text editor for Unix-like computing systems or operating environments using a command line interface

### Vim:

Vim [4] is a highly configurable text editor built to make creating and changing any kind of text very efficient. It is included as "vi" with most UNIX systems and with Apple OS

### Pip:

Pip [5] is the package installer for Python. You can use pip to install packages from the Python Package Index and other indexes.

### Arduino IDE:

The Arduino Integrated Development Environment [6]- or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

### Open SSH

OpenSSH [7] is the premier connectivity tool for remote login with the SSH protocol. It encrypts all traffic to eliminate eavesdropping, connection hijacking, and other attacks. In addition, OpenSSH provides a large suite of secure tunnelling capabilities, several authentication methods, and sophisticated configuration options.

|  |  |
| --- | --- |
| Programming Language | Library |
| Python | RPI.GPIO |
| C | Keyboard.h Micronucleus |

## Project Stakeholders and Sponsors

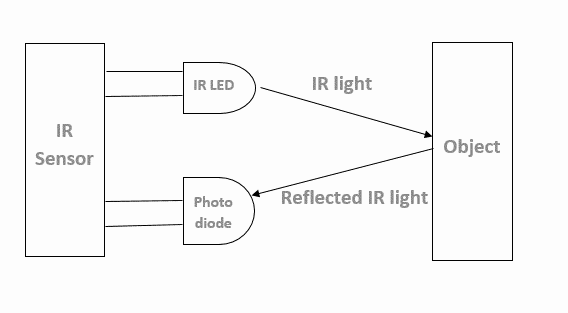
|  |  |
| --- | --- |
| **Project Sponsor** | Self-Funded |
| **Stakeholder** | Ammar Tariq  Salman Shafiq  Dr. Ali Shahid |

## Block Diagram

*Fig 1. Block Diagram*

This robot stays in the idle state after it successfully boots up to change from idle to functional state, we need to insert ATtiny85 [2] chip into one of its UBS port to inject keystrokes and set it to functional state from then on it detects white surface and black strips using its sensors and feed the information to Raspberry PI [1] using the data fed via sensors the motor driver is driven to control motors. This robot functions in a way when both sensors are on the white surface both motors run on full speed this way our robot will move in a straight line. 2nd scenario for when one sensor is on black line either left or right and other one is on white surface our robot will change its direction to move either left or right respectively. For stopping mechanism our robot will continue to move unless both sensors are on a black surface.

## Methodology



*Fig 2*

As described in Figure 1: IR sensor will be used to collect information from outside environment (surface). Figure 2, IR sensors detect light using its photodiode emitted by their IR LED whenever it comes near a reflective surface (in our case white surface or a light-coloured surface). Our robot needs to be programmed in such a way when right or left IR sensor detects input as ‘True’ it spins left or right motor at a full speed whereas if one sensor i.e., left one stops detecting IR light due to the black surface it will reduce its relevant motor speed to 50% hence our robot will move to left and on the other case if the right IR sensor stops detecting light it will reduce speed of right motor to 50% and our car will potentially turn to right. Ultimately when both sensors detect black line the robot will stop as previously told.

## Functionality

At very first we will use ATtiny85 microcontroller to start the code from then on Raspberry PI [1] will identify colour variation from the given input via IR sensors and obstacle distance from ultrasonic sensor sending the signal to motor driver correspondingly. Using PMW the robot will change the speed of its motor to move in potential direction.

## System Limitations/Constraints

The proposed system best works in controlled environment. It may not properly work under low lightning. There should be less distortion as possible as it can be for the proposed system to yield the best output. Read table 1 for further details.

### Working on different surfaces

|  |  |
| --- | --- |
| **Non-Uniform** | **Uniform** |
| On distorted surface our robot cannot move as path is not fixed and is distorted accordingly | On flat surface our robot can move reliably as the path is smooth hence sensor move in a fix path |
| On a non-uniform surface, speed of motors is fluctuating due to imbalance between wheels | Speed is constant with having a uniform surface |
| Sharp turns and hard surface can cause error in turning edges | Robot will turn smoothly with the uniform surface |

*Table 1*

## Scope

In big warehouses these robots play role in transporting items from one spot to another spot here no man power is required there for human labour can focus on other important matters this robot helps with the fast-paced delivery of materials required to fulfil daily routines for example moving packed materials from packing lines to delivery lines and moving items from inventory to packing lines so on and so forth.

Our version of robot costs lesser and provides somewhat similar functionality of those expensive robots. Therefore, our robot is more reliable and affordable this puts it apart from other line following robots.

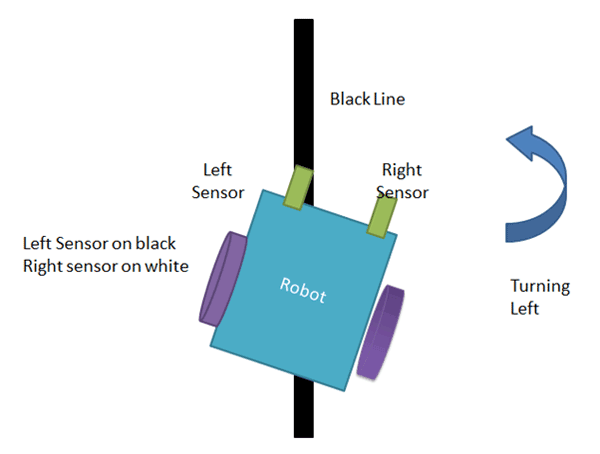
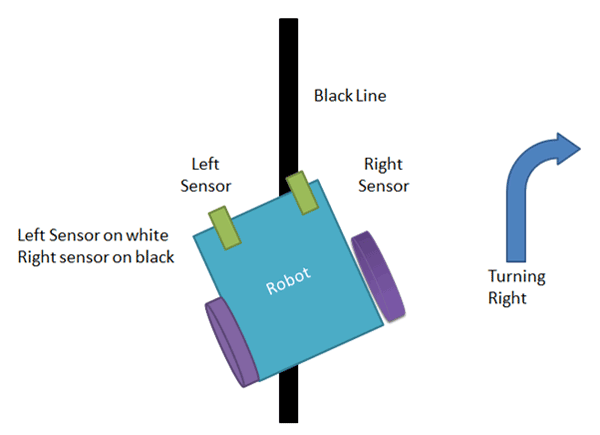
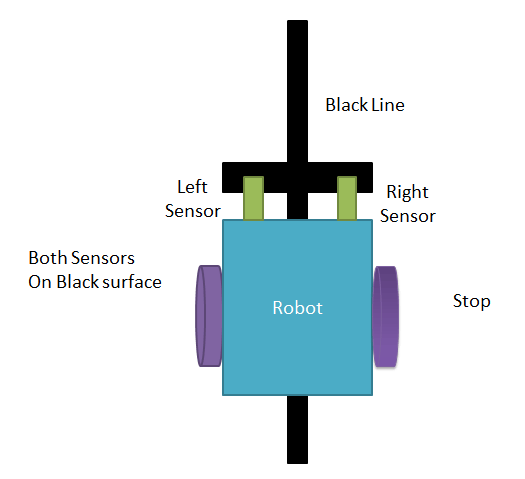
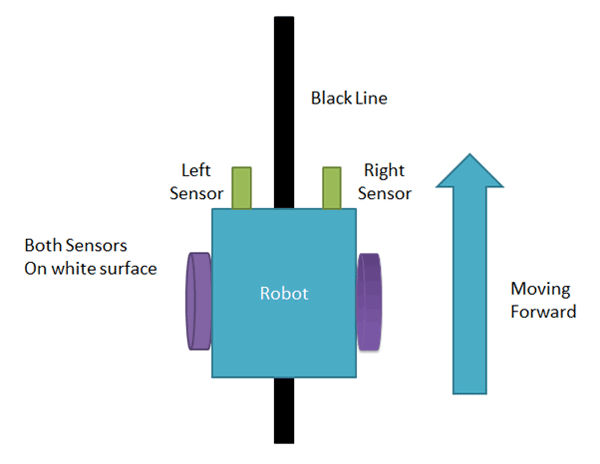
## Algorithm

[START]

1. Input L-IR and R-IR
2. If L-IR and R-IR Output = 1
3. Go Straight
4. **Go to step-1**
5. If L-IR Output = 0
6. Turn to the Left
7. **Go to step 1**
8. If R-IR Output = 0
9. Turn to the Right
10. **Go to step 1**
11. If L-IR and R-IR Output = 0
12. Stop

[STOP]

## Mock-up



## Gantt Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | June-July  2020 | Sept-Oct  2020 | | | Nov-Dec  2020 | | | Jan-Feb  2021 | | | Mar  2021 | | | | | Apr – Jun  2021 | | Oct  2021 | | Nov  2021 | | December  2021 |
| Initial Study |  | June 2020 – July 2020 | | | | | | | | | | | | | | | | | | | | |
| Project Selection |  |  | | | September 2020 – October 2020 | | | | | | | | | | | | | | | | | |
| Literature  Review |  | | |  | | | | | | October 2020 – January 2020 | | | | | | | | | | | | |
| Hardware Testing |  | |  | | | October 2020 – December 2020 | | | | | | | | | | | | | | | | |
| Design  Testing |  | | | | | |  | | | | December 2020 – January 2021 | | | | | | | | | | | |
| Proposal Writing |  | | | | | | | |  | | | | January – March 2021 | | | | | | | | | |
| Defence |  | | | | | | | | | | |  | | | March 2021 | | | | | | | |
| Plan Enforcement |  | | | | | | | | | | | | |  | | March 2021 | | | | | | |
| Mid-Term Report | April 2021 – June 2021 | | | | | | | | | | | | | | |  | | |  | | | |
| Hardware Assembly | July 2021 – November 2021 | | | | | | | | | | | | | | | |  | | | |  | |
| Testing and Evaluation of the system | December 2021 | | | | | | | | | | | | | | | | | | | |  | |

Pending

Finished

## Future scope

We will include following in our future documentation:

1. Time Chart
   1. Case 1 (Distance Covered on straight path in delta time)
   2. Case 2 (Distance Covered on curved path in delta time)
2. Performance Chart
   1. Case 1 (Simple Curve)
   2. Case 2 (Conic Curve)
   3. Bezier curve (S-Curve)

## Reference

1. <https://www.raspberrypi.org/>
2. <https://modernelectronics.com.pk/product/digispark-attiny85/>
3. https://www.nano-editor.org/
4. <https://www.vim.org/>
5. <https://www.arduino.cc/en/software>
6. <https://pypi.org/project/pip/>
7. <https://www.openssh.com/>
8. https://circuitdigest.com/sites/default/files/inlineimages/Working-of-Arduino-Line-Fol.gif
9. <https://circuitdigest.com/sites/default/files/inlineimages/Turning-left-to-line-follow.gif>
10. <https://circuitdigest.com/sites/default/files/inlineimages/Turning-right-to-line-follo.gif>
11. <https://circuitdigest.com/sites/default/files/inlineimages/Stopping-Line-Follower-Robo.gif>