The Great Guys

Final Design Report (FDR)

Line Follower Robot

Authors

Name	*University ID*
Rishab Jain	U6721789
Zehao Xu	U6538351
Chaohuo Wu	U6516811
Yanjun Li	U6729572
Xiaochen Liu	U6312197
Chaoyun Gong	U6329142
Hanwen Bi	U6292748
Masahito Takeuchi	U6483489

Executive Summary

The aim of the project is to design and develop a robust line follower robot that can navigate on a black line over a white background autonomously. This report provides deep understanding and detailed discussion of the systematic procedure implemented to successfully complete the project. In this project, a system engineering approach is executed. Systems engineering consists of a collaborative approach towards the development of an interdisciplinary and complex system. Often customer satisfaction is neglected in designing solutions, however the concept of systems engineering is to derive a customer driven product through continuous evaluation and feedback. The project begins with defining a problem statement which is derived from the stakeholder needs. Throughout the project, various phases are completed and reviewed to ensure the customer requirements are being satisfied. The four main stages present in a system engineering project are: Identification of need, preliminary design, detailed design, and manufacturing. After the completion of each stage, the outputs are compared with the stakeholder requirements and needs to ensure maximum customer satisfaction. Finally, the different needs and functions of the line follower robot are identified by applying systems engineering and accordingly prioritized so as to obtain the best result and output from the system.

Table of Contents

Acronyms and Abbreviations

AGV	Automated Guided Vehicle
ANU	Australian National University
CR	Customer Requirement
DC	Direct Current
DoI	Direction of Improvement
DR	Design Requirement
FBD	Functional Block Diagram
FFBD	Functional Flow Block Diagram
IDE	Integrated Development Environment
I/O	Input and Output
IR	Infrared Radiation
IoT	Internet of Things
PWM	Pulse Width Modulation
PID	Project Initiation Document
TPM	Technical Performance Measurement
SR	System Requirement
SRS	System Requirements Specification

Units of Measurement

The International System of Units (SI) is used in this project unless otherwise specified.

Table A: Units of Measurement

Measurement	Symbol	Description
Currency	$AUD	Australian dollar
Degree	°	Degree
Length	cm	Centimetre
Temperature	℃	Degree Celsius
Velocity	cm/s	Centimetre per second
Voltage	V	Electrical tension

1. Introduction

1.1 Background and Context

With the development of robotics techniques, more and more robots are implemented for industry and civil. Line follower robot is a special type robot, which involves control theory, artificial intelligence, and sensor technique, among others. Line follower robot can also be called automatic guided vehicle (AGV). According to [1], AGV is defined as a type of self-driving vehicle that uses battery as power source.

There are several characters for line following robot:

1) Self-driving. The vehicle is equipped with a guiding system that enables it to drive following the planned path, and deliver materials between endpoints.
2) Flexibility. The planned path can be adjusted according to the layout of the working place, and in this way the cost is significantly lower than traditional method such as conveyer belt. With the help of driving system, the robot can navigate flexibly in different environment.
3) Clean, noiseless. Most of line following robots use motor and battery as driver, power source. As compared to using engine as driver, it is more environmental-friendly.

In order to ensure that the line follower robot works normally, its attributes such as position, orientation and speed should be monitored. To acquire this information, sensor and guiding are equipped on the robot. The selection of signals sensors is dependent on the functionality of the robot and the environment that the robot works in. Different signals and sensors are shown below:

1) Electromagnetic signal. In this situation, power wires are buried underground and generate magnitude field. The magnitude sensors are installed on the robot to detect the magnitude. This method is a relatively mature solution and is wildly implemented.
2) Visual signal. In this scenario, coloured lines are attached or printed on the ground. The sensor will sense the difference of colours between the line and the background. Commonly, cameras and infrared radiation (IR) are implemented on the robot to detect the visual signal.

When comparing the sensors, electromagnetic sensor is obviously more reliable than visual sensors. However, it is less flexible since changing the planned path cost more budget and time. Moreover, compared to infrared radiation sensor, camera shows better performance on detecting different colour lines and perform multiple tasks. However, its cost is significantly higher. Considering the budget and working situation, this project opted to use IR sensor to detect the line.

Due to the flexibility of line follower robot, it is broadly implemented in a great number of fields. For example, line follower robot is commonly used in warehouses to deliver materials and goods. In manufacturing industry, with advanced dispatch algorithm, line following robot can deliver semi-finished products to the next beltline, which dramatically saves time and labour. In working place such as post office and library, the delivery work is boring and the work load is massive, thus robot is more suitable for this kind of tasks.

In this project, the objective is to provide a solution and implementation for designing a line following robot that meets the requirements of the stakeholders, e.g., navigating automatically and detect coloured lines against the black or white background. Taking into consideration the budget and schedule, system engineering is implemented within the project. System engineering starts from the concept of system and optimizes the developing loop to achieve the best outcoming. It provides strong tools to analyse the requirements and maximize the performance. The block diagram of system engineering cycle is shown in Figure 1.1.

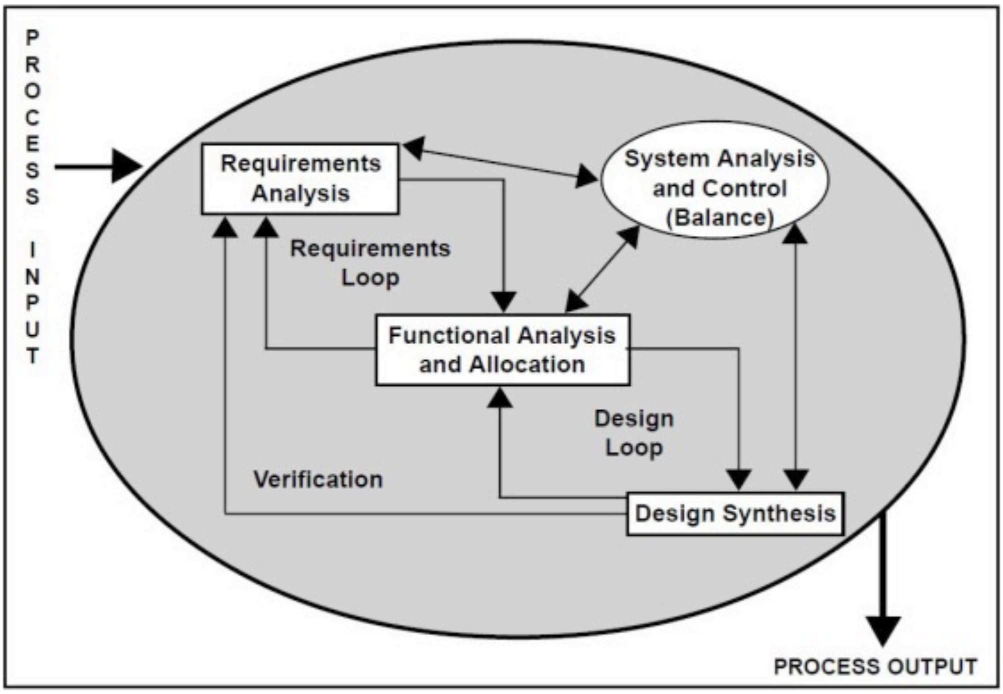


Figure 1.1: Systems Engineering cycle [2]

1.2 Scope

In this report, Chapter 1 provides a brief introduction of the background of this project, including the applications of line following robot and the types of sensors. After that, Chapter 2 specifies the customer requirements (CRs), followed by system requirements (DRs) and functional requirements that are derived from customer requirements.

Chapter 3, Chapter 4 and Chapter 5 illustrate the design overview, functional architecture, physical architecture and design synthesis. Firstly, the sub-systems that fulfil different function are specified according to the functional requirement. Then, the devices and components development are discussed. Lastly, the software design is shown in the end of this part.

The verification of the performance and its discussion is described in Chapter 6 and Chapter 7, while the project cost is shown in Chapter 8. Finally, conclusion and future work is shown in Chapter 9 and Chapter 10.

2. Requirements

Requirements of a system describe the approach of system designing and the functions of system performing. In this section, customer requirements, system requirements and design requirements have been introduced in detail, and the ranking of requirements has also been presented.

2.1 Customer requirements

Customer requirements indicates the relationship between organizations and the customers, which reflect the stakeholder's expectation on the whole system. Customer requirements in this project can be found in Table 2.1.

Table 2.1: Customer Requirements

CR I.D	Customer Requirement
CR-01	The robot shall follow visual black line on the white background.
CR-02	The robot shall be powered by batteries.
CR-03	The robot shall consist of simple and affordable sensors, batteries, microcontroller and motors.
CR-04	The robot shall be easy to maintain and assemble.
CR-05	The robot shall drive autonomously.
CR-06	The robot shall be larger than 10cm x 5cm x 5cm.
CR-07	The robot shall have reliable components in order to ensure a long working life.

2.2 System requirements

System requirements is responsible for illustrating the targets and functions of the entire system, which is derived from customer requirements [3]. Customer requirements and its corresponding priority can be seen in Table 2.2.

Table 2.2: System Requirements

CR I.D.	SR I.D.	Requirement	Justification	Priority
CR-01	SR-01	The robot shall detect a black line against a white background using IR sensors.	The principal objective for this mission is to detect a black line against a white background.	2
	SR-02	The control algorithm shall keep the line in the centre of the robot.	This is to allow the robot to keep tracking the central position on the line.	5
	SR-03	The sensors detection range shall be at least three times wider than the line	This it to avoid missing the line, which can lead to pause or wrong behaviours of the robot.	20
	SR-04	The robot shall follow the line including straight lines and curve lines.	This is to allow the robot to keep moving depending on the black line.	2
CR-02	SR-05	The robot shall independently work by dry cell batteries.	This is to allow the robot to move without an external cable such as USB power supply.	10
	SR-06	The robot will be driven with commonly used power supply.	This is to allow the power supply for the robot is accessible.	10
CR-03	SR-07	The robot shall consist of simple and fundamental components.	This is to allow the robot to satisfy the function of line following with simple components.	1

CR I.D.	SR I.D.	Requirement	Justification	Priority
	SR-08	The robot shall be created within a certain amount of budget, $400.	This is to allow the robot to satisfy the function of line following with low cost and good availability.	10
CR-04	SR-09	Each component shall be easy to attach/detach and maintain.	This is to allow the robot to be repaired when the parts need to be exchanged or related trouble happens.	16
	SR-10	Each component shall be interchangeable and easily available.	This is to allow each component of robot to be composable after trouble without the shortage of the component.	16
	SR-11	The number of components shall be less than 100.	This can limit the assembly time in reasonable range.	5
	SR-12	At least one extra component shall be purchased to be maintained quickly.	This is to allow shorter shopping time, which can result in the save of time and the quick repair.	21
CR-05	SR-13	The robot shall be controlled by pre-programmed code using IR sensors.	This is to allow the robot to follow the black line without external operations once it starts moving.	5
	SR-14	The robot shall move at a minimum speed of 1cm/s on the target line.	This is to allow the robot to move continuously in control, while following the black line without going to other directions.	5
	SR-15	The left and right wheels shall be driven by individual motors.	This is to allow the robot to turn right/left and move forward.	10
	SR-16	The robot shall continuously move on a flat even surface without breaking down.	This is to allow the robot to keep driving without halting in a certain ground condition.	10
CR-06	SR-17	The components of robot shall be assembled together on the chassis with bolts and nuts.	This is to allow the components of the robot to be stable and fixed and prevent them from falling apart.	5
	SR-18	The appearance of the line follower robot shall be normal car-like appearance.	This allows that the appearance is presentable and the robot's body shell is easy to manufacture.	16
	SR-19	The robot shall be dimensionally acceptable to house all components.	This is to ensure that the size of the line follower robot is not too small or too large to present.	16
CR-07	SR-20	The lifecycle of the system shall be longer than 10 years.	This is to ensure that the line follower robot can work for a long period of time.	10
	SR-21	All components and systems shall meet relevant standards.	This is to ensure that all components have high quality and can be operated for a long period time.	2

2.3 Design requirements

Design requirements (DR), derived from customer requirements, indicate the assumptions and expected performance of the whole project. Table 2.3 presents the relationship between the requirements to ensure that the final system design satisfies the customer requirements and expectations. Each design requirement has an associated technical performance metric to ensure that each requirement is measurable which also enables the proposed design to be verified.

Table 2.3 Design requirements and Technical attributes

DR I.D.	Requirement	Metric	DoI	TPM
DR-01	The robot shall detect a black line against a white background using IR sensors.	Voltage	+	volt (V)
DR-02	The control algorithm shall keep the line in the centre of the robot.	Software	o	-
DR-03	The sensors detection range shall be at least three times wider than the line	Width	+	centimetre (cm)
DR-04	The robot shall follow the line including straight lines and curve lines.	Quality	o	-
DR-05	The robot shall independently work by dry cell batteries.	Voltage	o	volt (V)
DR-06	The robot will be driven with commonly used power supply.	Voltage	o	volt (V)
DR-07	The robot shall consist of simple and fundamental components.	Quality	o	-
DR-08	The robot shall be created within a certain amount of budget, $400.	Money	-	Dollar (AUD$)
DR-09	Each component shall be easy to attach/detach and maintain.	Quality	o	-
DR-10	Each component shall be interchangeable and easily available.	Availability	o	-
DR-11	The number of components shall be less than 100.	Number	-	numbers
DR-12	At least one extra component shall be purchased to be maintained quickly.	Software	o	-
DR-13	The robot shall be controlled by pre-programmed code using IR sensors.	Software	o	-
DR-14	The robot shall move at a minimum speed of 1cm/s on the target line.	Speed	-	centimetre /second (cm/s)
DR-15	The left and right wheels shall be driven by individual motors.	Voltage	o	volt (V)
DR-16	The robot shall continuously move on a flat even surface without breaking down.	Quality	o	-
DR-17	The components of robot shall be assembled together on the chassis with bolts and nuts.	Quality	o	-

DR-18	The appearance of the line follower robot shall be normal car-like appearance.	Quality	o	-
DR-19	The robot shall be dimensionally acceptable to house all components.	Length	o	centimetre (cm)
DR-20	The lifecycle of the system shall be longer than 10 years.	Time	+	years
DR-21	All components and systems shall meet relevant standards.	Standards	o	-

2.4 Regulations and Standards

In order to optimize the system, constraints involving parameters, regulations and standards related to the system shall be considered. Regulations and standards can be the guidelines of the project which can help us adjust the critical parameters and promote the performance of the system. The core regulations and standards can be seen in Table 2.4 as follows:

Table 2.4 Standards and Regulations

Standards/ Regulations I.D.	Description	Source
AS:4024.3303:2017 (safety requirements for industrial robots - Robots) [4]	This standard defines the required working environment for the safety of industrial robot systems and supplements the requirements for industrial robots to work together. **[Standard Australia, 2019].** These requirements are designed to protect individuals and prevent risk of property damage caused by electric shock, among others.	Standards Australia
ISO/TC10/SC 6 [5]	This is an international standard for mechanical engineering documentation. **[https://www.iso.org/committee/46064.html].**	International Standard
IEC61326-1 [6]	This standard establishes electrical equipment for control, measurement and laboratory use. **[https://webstore.iec.ch/publication/5275]**	International Standard
AS 293901987 (Industrial robot systems – Safe design and usage) [7]	This standard specifies the industrial construction, protection, design requirements, and installation of robot systems. Furthermore, training for individuals is provided. **[Standard Australia, 2019]**	International Standard

3. Design Overview

Based on the design requirements, we assemble functions of the line follower robot into 6 subsystems, namely Signal Processing Subsystem, Microcontroller Subsystem, Sensor Subsystem, Power Subsystem, Navigation Subsystem, and User Interface Subsystem.

3.1 Functional architecture

After System Requirements Specification (SRS), we obtain a set of concept-independent system level requirements. With the help of functional analysis, a functional architecture is generated, which describes the 'what' of the system rather than the 'how'. The functional architecture is shown in Figure 3.1, which clearly describes the basic functions that the line follower robot must perform. By integrating the SRS and functional architecture, a functional baseline is determined and utilized to design various concepts based on customer needs.

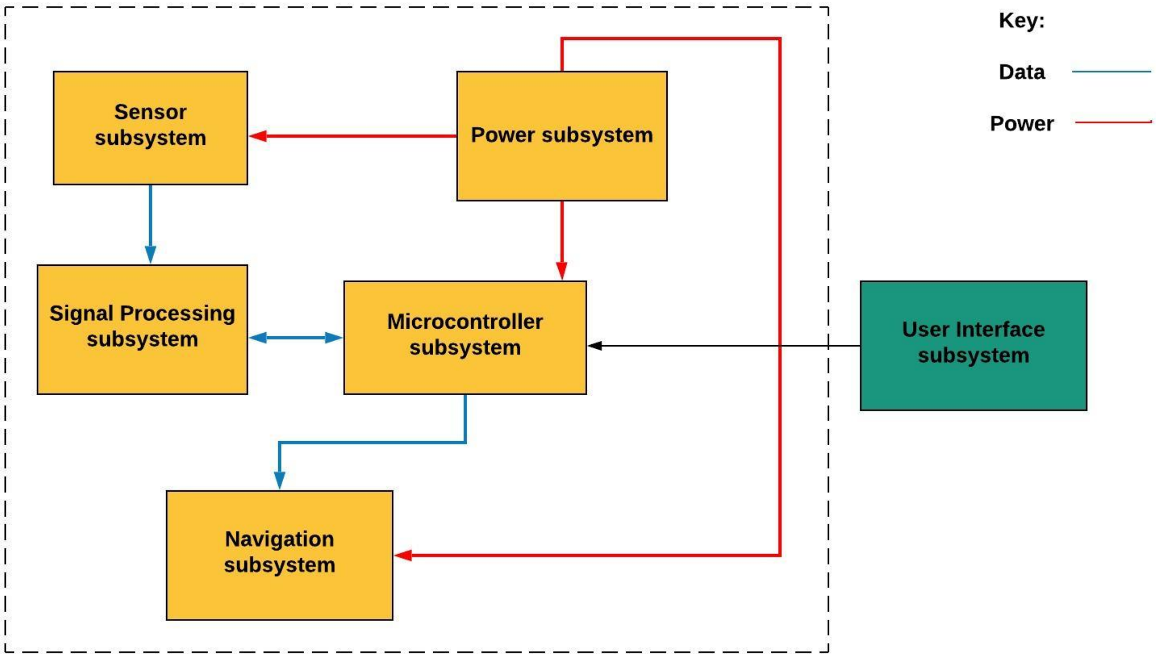


Figure 3.1: Functional architecture

Another critical part of functional analysis is the Functional Flow Block Diagrams (FFBD) that decomposes the functions into sub-functions. These FFBDs create well-defined functions that later transforms them into a physical architecture.

3.2.1 Sensor Subsystem

The Sensor subsystem mainly perform the sense function. It will send and receive Infra-Red light to detect the line in front of the line follower robot. After receiving the reflected Infra-Red signal, the Sensor subsystem will convert it to electronic signal and transmit it to the Signal Processing subsystem.

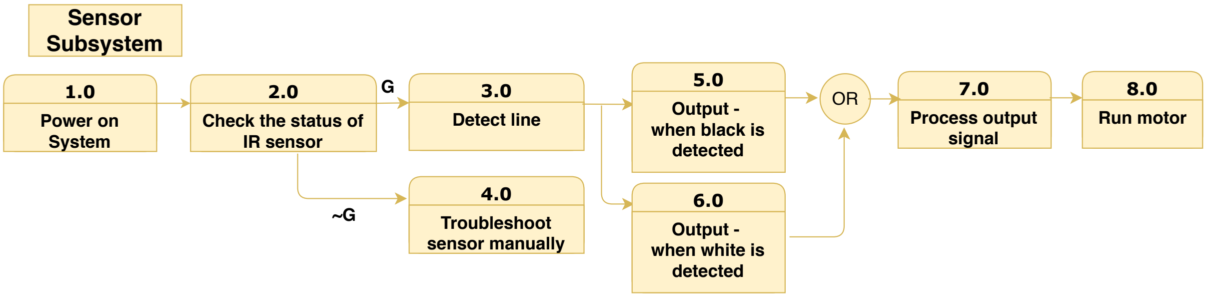


Figure 3.2: Sensor Subsystem

3.2.2 Signal Processing subsystem

The main duty of the Signal Processing subsystem is to receive the electronic signal from the Sensor subsystem and do filtering and de-noising, before transmitting the processed signal to Microcontroller subsystem.

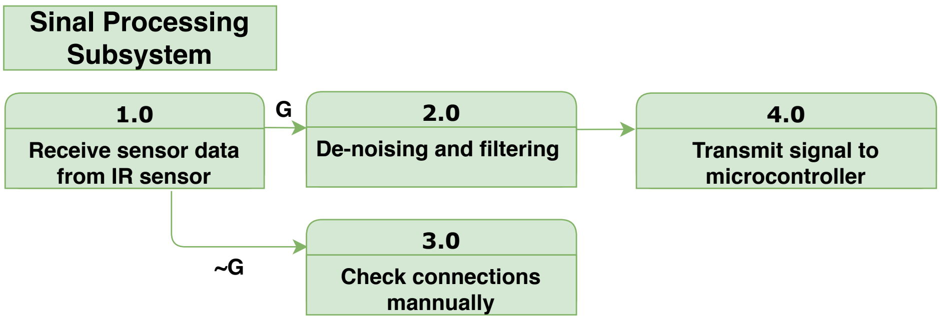


Figure 3.3: Signal Processing Subsystem

3.2.3 Microcontroller subsystem

Microcontroller subsystem controls the movement of the line follower robot by the pre-saved coding. Once the Microcontroller subsystem receives the electronic signal from Signal Processing subsystem, it will control the movement of the line follower robot based on pre-saved control algorithms and send electronic signals to the Navigating subsystem to adjust the movement.

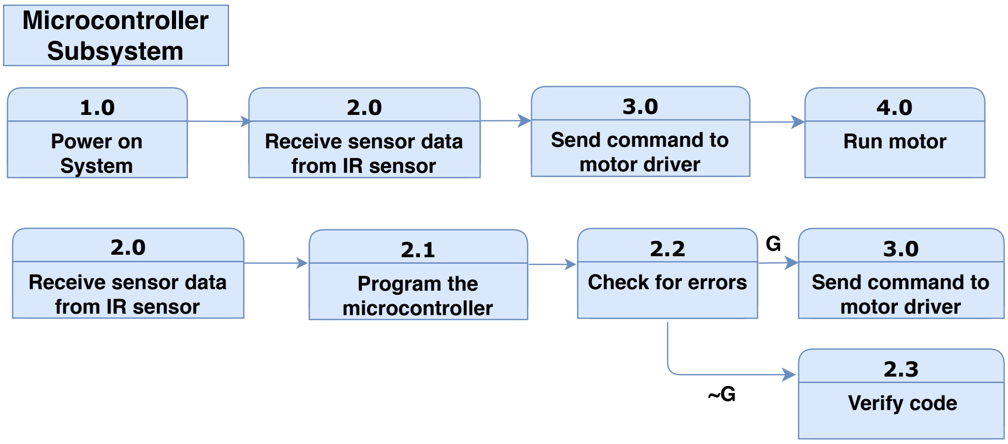


Figure 3.4: Microcontroller Subsystem

3.2.4 Navigating subsystem

The Navigating subsystem convert the electrical energy from the Power subsystem to the output power. It receives the electronic signal from Microcontroller subsystem and adjust the movement of motors based on the electronic signal.

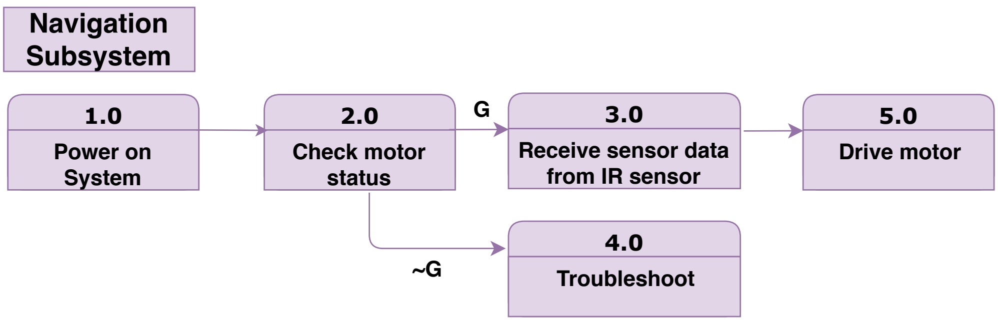


Figure 3.5: Navigation Subsystem

3.2.5 Power subsystem
The Power subsystem provides electrical power to Signal Processing subsystem, Microcontroller subsystem, Sensor subsystem, and Navigation subsystem.

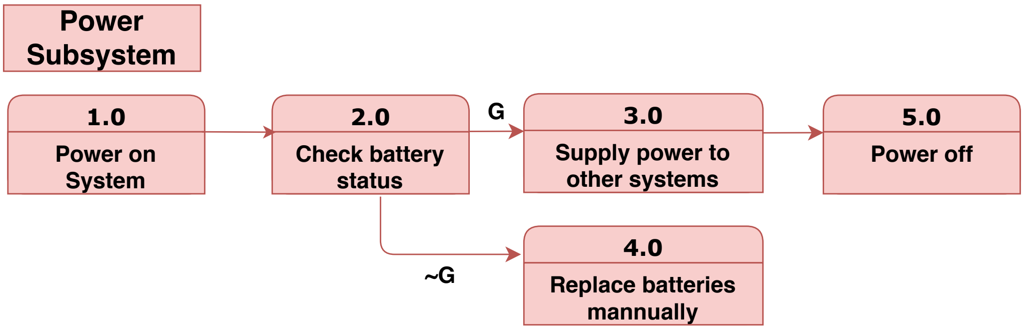


Figure 3.6: Power Subsystem

3.2.6 User Interface subsystem
User Interface subsystem include software, such as the Arduino integrated development environment which allows to program the microcontroller to create and implement the line following algorithm for our robot.

4. Physical Architecture

4.1 Main components

Line follower robot consists of six main components as Figure 4.1 shows. Each component is assembled on a chassis and connected each other.

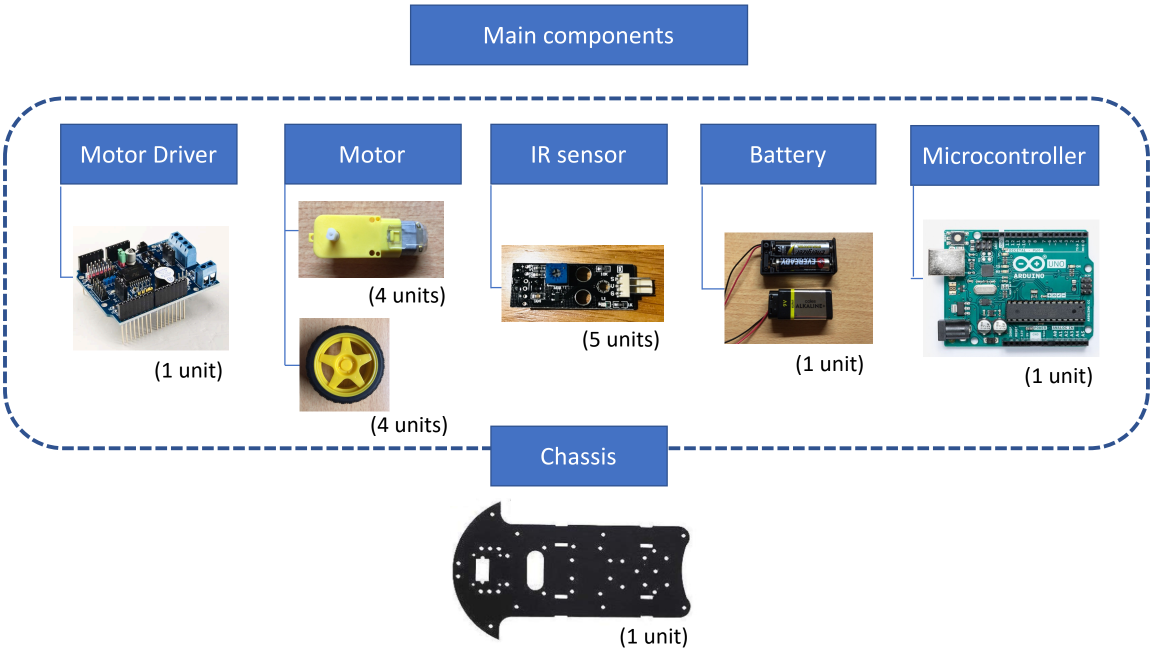


Figure 4.1: Main components

Based on Chapter 3.1, the final design of functional architecture is described with physical components as shown in Figure 4.2. This architecture is traceable corresponding to each function and component to realise design requirements.

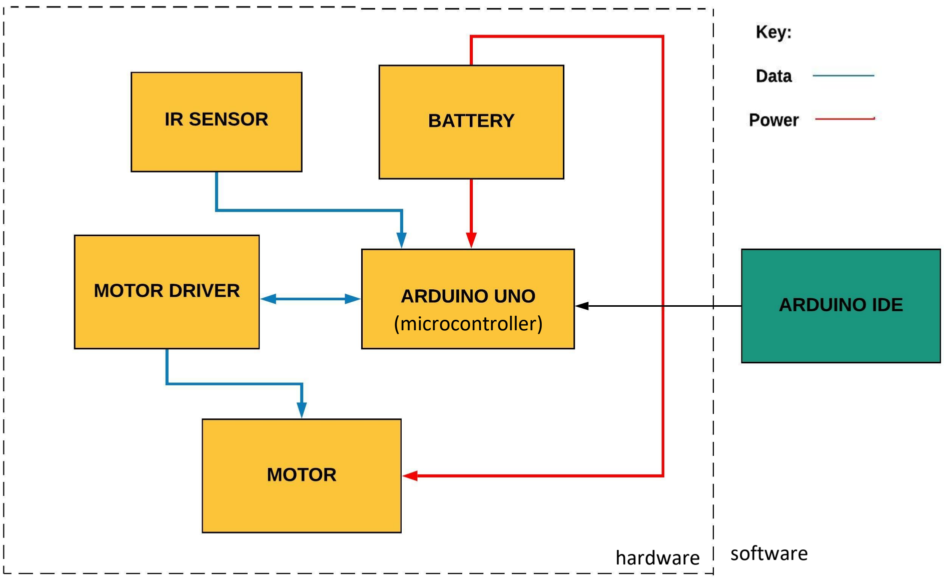
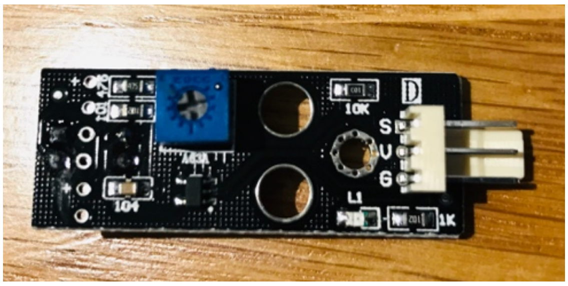


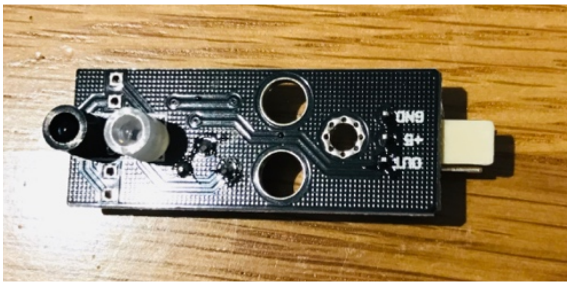
Figure 4.2: Final design of functional architecture

4.1.1 IR sensor

IR sensor can detect the light which has the wavelength between 760 ~ 1000 nm. In this project to distinguish color, IR sensor is quite effective, providing analog or digital output that indicates the presence of an object or color. The output signal can be monitored on the serial monitor on microcontroller system.



(a): Front side



(b): Back side

Figure 4.3: IR sensor

IR sensors are highly susceptible to surrounding environment so each transmission and reflection sensor is covered to avoid the effect from ambient lights. For fine tuning of the sensitivity of IR sensor, potentiometer on the board is used. Figure 4.4 shows the mechanism of how IR sensor works. Once a transmitter emits infrared rays to an object, the corresponding receiver collects the reflected rays from the object. Then, the analog output signal is transferred to a microcontroller, interpreted as digital signal to control the rotation of motors and the direction of robot.

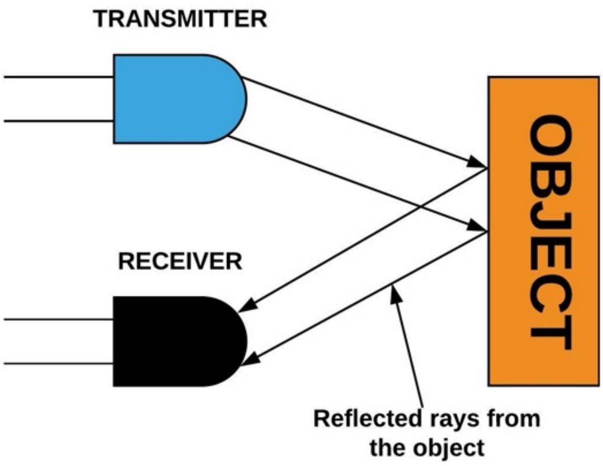


Figure 4.4: Mechanism of how IR sensor works

4.1.2 Motor

Bidirectional DC motors are connected to rubber wheels to make the robot move. Sizes of wheel and motor are chosen based on design requirements. Each specification is described in Table 4.1.

Figure 4.5: Wheel

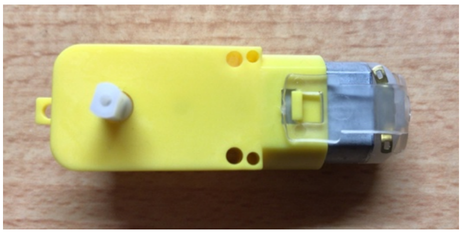


Figure 4.6: Motor

Table 4.1: Specification of wheel and motor [8]

Wheel	**Diameter**	**66 mm**
	Width	**28 mm**
Motor	**Motor Nominal Voltage**	**10V**
	Full load current	**250mA**
	Max. efficiency current	**70mA**

4.1.3 Motor driver shield

Motor driver shield is applied to this project, which equips L293D chip which enables four bidirectional DC motors driven. This shield can be directly plugged into microcontroller, especially Arduino UNO. Thus, this shield offers compact design with better mobility.

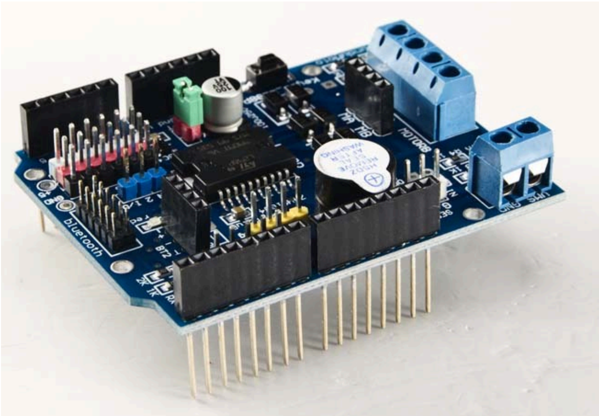


Figure 4.7: Motor driver shield [9]

Pulse Width Modulation (PWM) is used for motor drive. Basically, it is a method to send the power required by turning on and off the pulse. The shield has one enable pin and four input pins per two motors. To control each motor such as clockwise rotation, anti-clockwise rotation or stop, each pin state is changed either HIGH or LOW. Table 4.2 shows the specification of motor driver shield.

Table 4.2: Specification of motor driver shield [9][10]

Supply Voltage to Vcc1(vss)	**4.5V to 7V**
Motor voltage Vcc2 (Vs)	**4.5V to 36V**
Maximum Peak motor current	**1.2A**
Maximum Continuous Motor Current	**600mA**
Transition time	**300ns (at 5Vand 24V)**
Size	**69mm (Width)* 53mm (Length)**

4.1.4 Microcontroller

In this project, Arduino UNO [11] is used as a microcontroller to synthesize signals between IR sensors and motor driver shield. Detail code is shown in Appendix (A).

Figure 4.8: Arduino UNO [11]

Arduino Uno is an open-source microcontroller board based on microchip ATmega328P microcontroller with sets of input and output pins, which enable users operate it as interactive devices with sensors. Arduino's integrated development environment (IDE) is a cross-platform Java application that includes an editor, compiler, and firmware transfer to the board. Table 4.3 shows the specification of Arduino UNO.

Table 4.3: Specification of Arduino UNO [11]

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz
Size	69mm* 53mm*14.3mm
Weight	25g

4.1.5 Battery

In this project, to provide electricity for motor shield and microcontroller, 12V-battery is used as the combination one 9V and 2 AA batteries.

Figure 4.9: Battery

4.2 Circuit Design

Five IR sensors and four motors are connected to motor shield as shown in Figure 4.10. 12V batteries are connected to power supply pin on the motor shield.

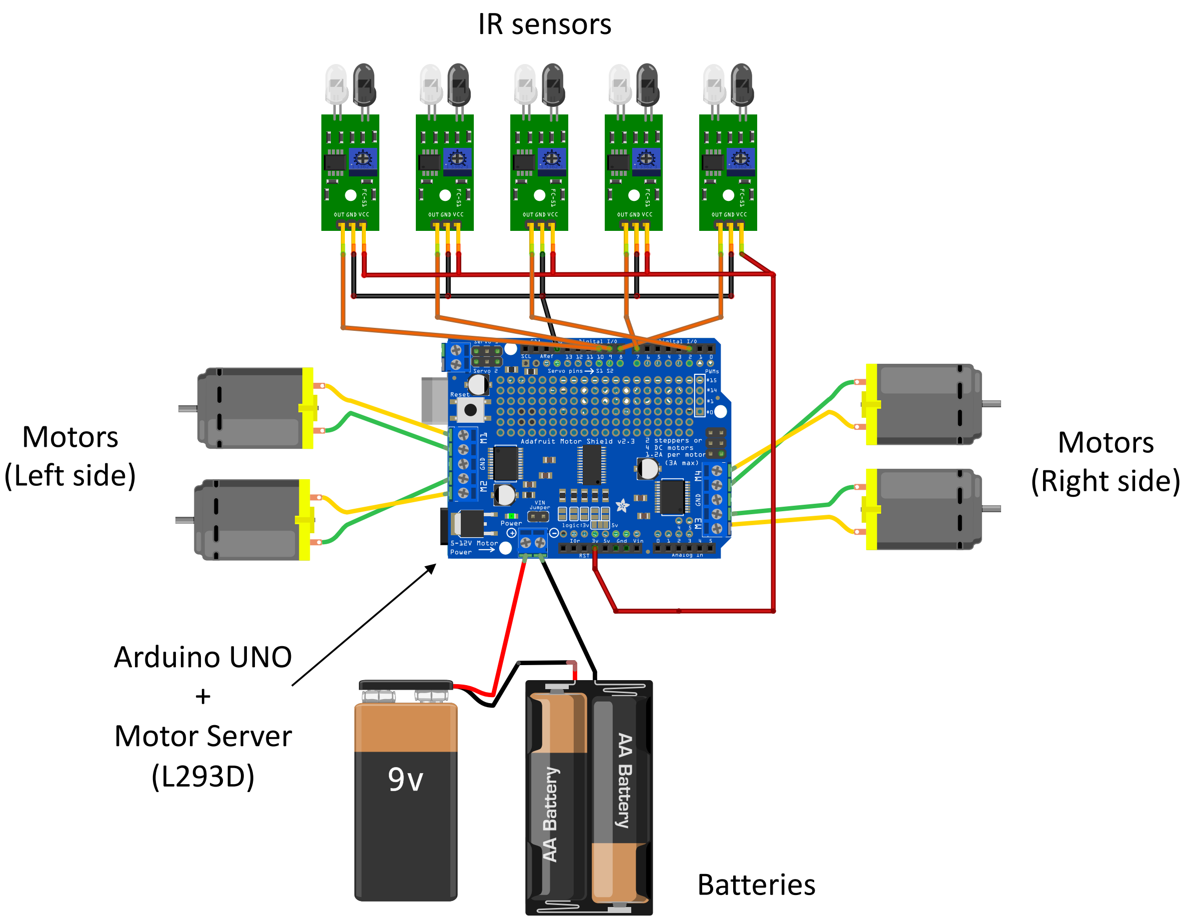
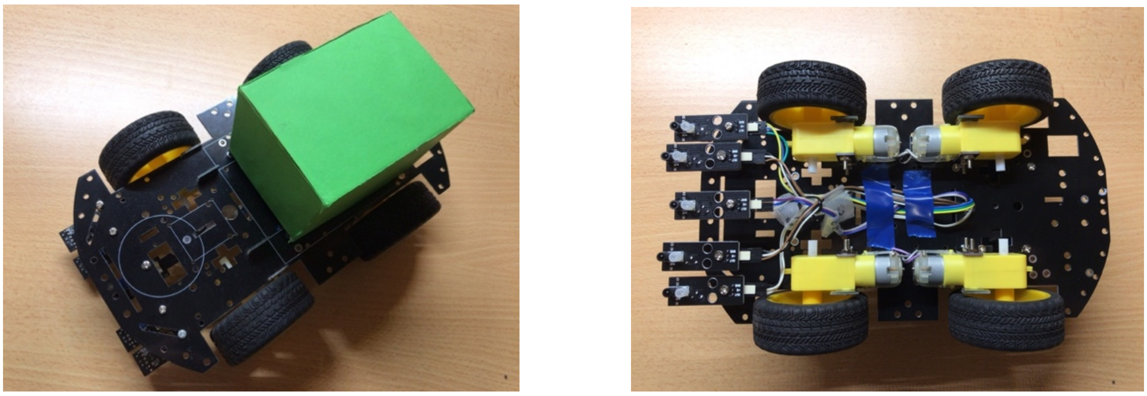


Figure 4.10: Circuit design

4.3 Assembly

Each component is tightly assembled on the car chassis using bolt, knots and tape as Figure 4.11 shows. Four motors are installed to the back of chassis and cables are properly bound, avoiding touching motor shafts. Microcontroller and cables are contained in a box to make the car presentable and to avoid disassembling cables when the car runs. Five IR sensors are facing to the ground to detect black line and are arrayed in a row to cover wide range of areas simultaneously as shown in Figure 4.12.



(a) Front side **(b) Back side**

Figure 4.11: Assembly of line follower robot

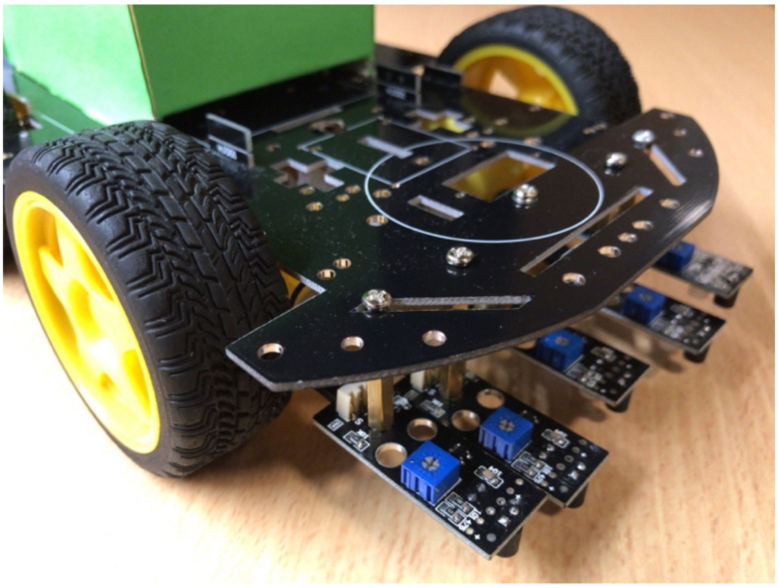


Figure 4.12: Assembly of five IR sensors

5. Design synthesis

To satisfy the design requirements and corresponding the functions, this project has the following design. The flowchart of the design is shown in Figure 5.1 below.

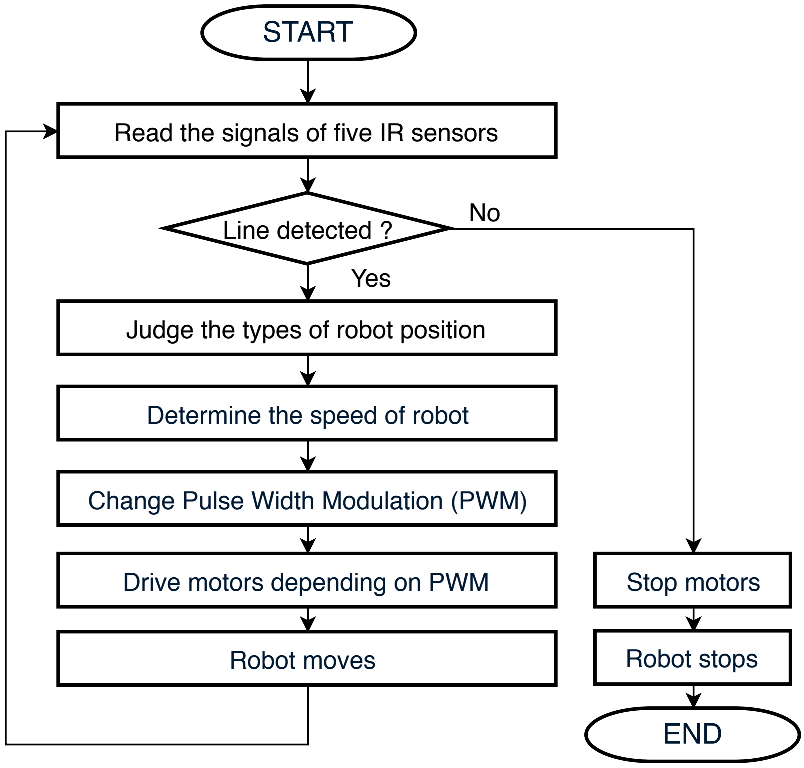


Figure 5.1: The flowchart of the design

In the flowchart, it shows the process algorithm of the robot. The five IR sensors in robot are detecting the black line at the same time in five different position. If the black line has not been detected by any sensor, the four motors in robot will stop, and the robot will stop. If one or more sensors have detected the black line, the signal will send back to control centre (Arduino board). It will judge the types of robot position in the black line. Then it will determine the speed of right and left motors in robot and calculate the PWM values for right and left motors separately. Depending on the motor driving board, the control centre will transmit the PWM values to corresponding motors and drive them. Thus, the robot will move and follow the black line. After that, the process will go back to the beginning and construct a loop system. Those actions execute sequentially, but almost in the same time. Thus, we do not need to care too much about the delay problem. However, if the speed of robot is too high, even the robot has changed its speed and direction, the inertia force will still push the robot move to the worry position.

For the step which judge the types of robot's position, the below table shows the details of the judgment algorithm.

Table 5.1: The logic table of actions' algorithm of robot

IR sensor					Output	
Left	Left-Middle	Middle	Right-Middle	Right	Motion	Error [arb.unit]
N	N	Y	N	N	↑	0
N	Y	Y	Y	N	↑	0
N	N	N/Y	Y	N	→	1
N	N	N/Y	N/Y	Y	→	2
N	Y	N/Y	N	N	←	-1
Y	N/Y	N/Y	N	N	←	-2
N	N	N	N	N	Stop	NA
N/Y	Y	Y	Y	N/Y	Special situation	NA

The inputs are five IR sensors, each IR sensor has two values 1 and 0 (in the table, Y means 1, and N means 0). When the IR sensor detects black line, it will become 1. When the IR sensor does not detect black line, it will be 0. the output is a variable Error. When Error is 0, it means the line is in the middle of the robot, the robot will go forward. When Error is larger than 0, it means the line is a little bit right of the middle of the robot, the robot will turn right. When Error is smaller than 0, it means the line is a little bit left of the middle of the robot, the robot will turn left. When all the sensors have not detected line, the robot will stop. The actions of the robot depend on the speed difference between left-side and right-side motors. Figure 5.2 below shows the theory briefly.

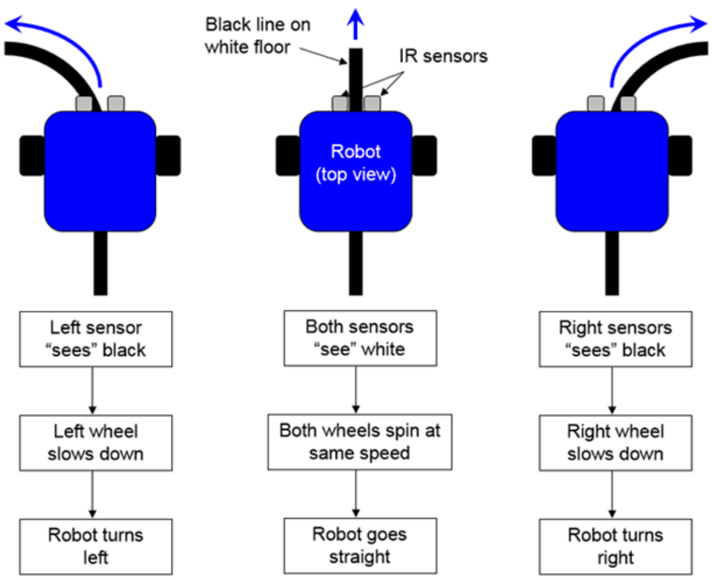


Figure 5.2: The actions of the robot

From the Table 5.1, the inputs' combinations have not been all used. There are some combinations left for some special situations. For example, when robot meets a crossroad, the inputs will be five Y, then the robot can be programmed for some special actions, like turn left or turn right or go straight.

The next important part is PID control. PID control will adjust the turning speed of robot and make the turning more smoothed and reduce the oscillation of robot. The function of PID is

$$U(t) = kp\left(err(t) + \frac{1}{T_I}\int err(t)dt + \frac{T_D derr(t)}{dt}\right)$$

It also can be written the equation below:

$$U(k) = K_p e(k) + K_i \sum_{n=0}^{k} e(n) + K_d(e(k) - e(k-1))$$

K_p, K_i and K_d are the hyper-parameters, they need to be set to the suitable values at the beginning. *e(k)* is the output Error in the Table 5.1.

$$U = K_p error + K_i \sum_{n=0}^{k} error + K_d(error - previous\_error)$$

U will be the output PWM value for the motors.

6. System Verification

In this part, the design requirements verification method we applied will be presented. Based on the design requirement verification diagram in our system requirement specification, there are 21 design requirements that need to be verified. In the design requirement attributes, there are four design requirement verification method that will be discussed in this part. The design requirement which will be verified is presented in Table 6.1

Table 6.1 Verified design requirements

Requirement	Verification	Completion
The robot shall independently work by dry cell batteries.	Remove all outer supply line and check if the car could still fulfil DR-01.	Completed
The robot shall follow the line including straight lines and curve lines.	Apply a testing line with 45° and 90° curve line routine and check if the car could follow the curve line and	Completed
The robot shall move at a minimum speed of 1cm/s on the target line.	Apply a testing line with a fixed length and measure the time that car could finish it and compute the speed.	Completed
The robot shall be dimensionally acceptable to house all components.	Measure the length, width and height of the car.	Needed to be discussed
The robot shall be created within a certain amount of budget, $400.	Sum the cost of all that we have bought and what we intend to purchase and check if it exceeds AUD$400.	Completed

The first main design requirement that needs to be verified is whether the car could work without extra power supply except battery assembled in the car. In the early detail design period, we avoid all outer power supply and power the vehicle with only batteries. Additionally, all our test on the car runs without any outer power supplies. Consequently, this design requirement is verified.

Next, the second design requirement is that the car shall be able to conduct a 45 degree and 90-degree turn. In a test, a routine created by black taps containing both a 90 degree and 45-degree turn is applied. The result shown in Figure 6.1 and Figure 6.2 indicates that the vehicle is able to conduct both 45 degree turn and 90-degree turn.

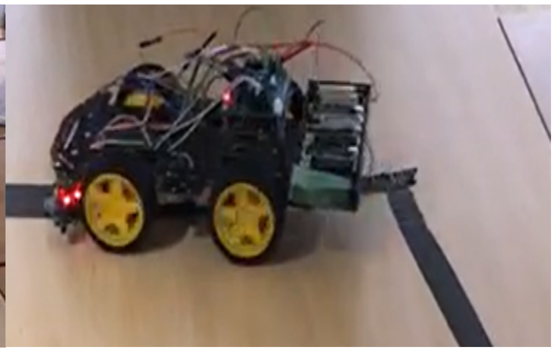
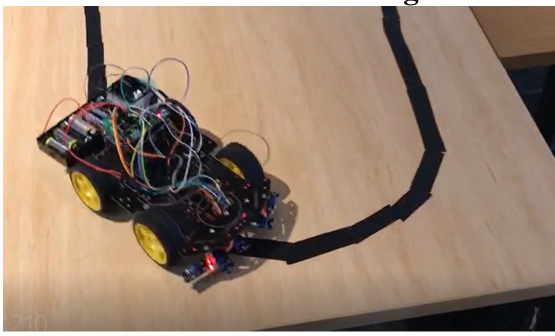



Figure 6.1: 90-degree turn



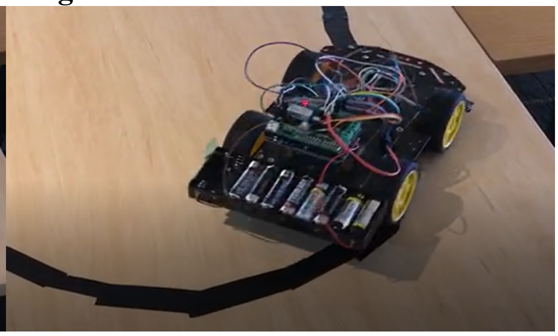


Figure 6.2: 45-degree turn

Then, the third design requirement which needs to be discussed is whether the car could move at least 1cm/s on the target line. To verify this requirement, two conditions are considered. The first verification routine includes only a straight line, while the second verification routine includes a 45-degree turn and a 90-degree turn (shown in Figure 6.3) aiming at clarifying the effect of turn. Based on the video length and the total length of test routine, the average speed of car in the straight line is 28.06 cm/s, and the average speed of 45-degree turn is 19.26 cm/s, while the average speed of 90-degree turn is 17.4 cm/s. Consequently, this design requirement is fulfilled.

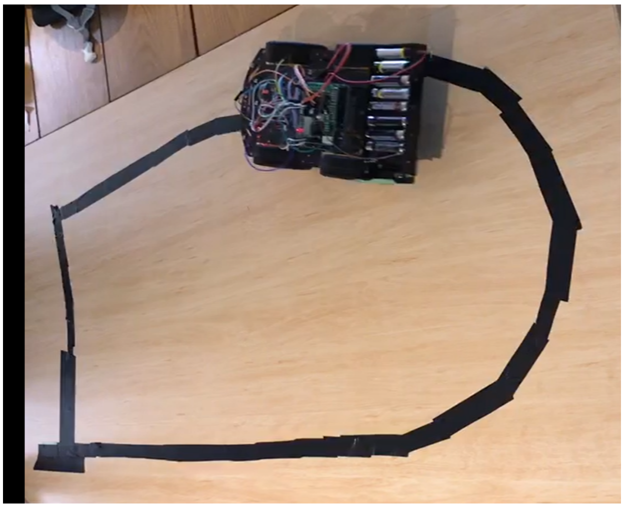


Figure 6.3: Test routine

Next, the fourth design requirement is the size of vehicle. In the requirement, the size of car should not exceed 10 cm * 5 cm * 5cm. The size of our robot is 22.4 cm* 16.3 cm *9.8 cm, which exceeds the design requirement. The reason is that the size of wheel exceeds the initial design and that is the only kind of wheel which could be accessed in the design period.
The last importance design requirement is the budget limit. In the requirement, our budget is limited to 400 Australian dollars. It can be concluded that we make use of all the budget we could have and do not exceed. Detail of budget allocation is explained in Chapter 8.

In this part, most of the main requirement in the design are fulfilled and verified. Although some of the main requirements are not perfectly fulfilled, they could be researched in future work.

7. Discussion

In this part, the completion of the project and comparison between the result and customer requirements will be discussed. The discussion will focus on how the system fulfils operational objectives which are derived from customer requirements.

In the project initiation document, derived from the seven customer requirements, there are five main operational including four functional objectives and two performance objectives in this project. During the project, there is also a new functional objective added. The four functional objectives are detecting visual black lines, working in different light conditions, detecting lines with different colours, and assembling sensors and device.

For the first functional objectives, as described in verification part, the car could successfully detect black line and allow the car to follow the line. For assembling sensors to the device, in the final design result, it is confirmed that all five IR sensors in the front of car are all assembled to this car with bolts and nuts. This functional objective also has a related performance objective, assembling sensors to the car with bolts and nuts makes it easier to remove it from the car with screwdriver, which makes the system easier to maintain. But assembly the IR sensors could also cause problems. IR sensors may not be strictly well connected and may be loose, causing position change of IR sensor, which could makes detecting lines with different width unreliable.

The third functional objective caused some problems in our project. Although a solution adjusting the resistance of IR sensor to avoid this problem is proposed, it is also clarified that the light condition may change quickly, and when it changes, the sensor resistance should be reset to a new appropriate value. The new solution needs to be discussed in future work.

For the fourth functional objective, in order to follow the lines, a simple plan is initially developed simply adjust the resistance of IR sensor is the direct solution based on the principle of IR sensor. However, in the test, it is found that simply changing the resistance of IR sensor is a suboptimal solution. It is found that the resistance range that the IR sensor could detect a specific colour is quite narrow. In addition, the effect is also not stable, which indicates that the sensor may not be able to detect a specific colour as expected even if it has been adjusted. In Figure 7.1, the effect of detecting lines with yellow colour is presented. When the IR sensor is adjusted to detect yellow line, it may not be able to work if the sensor hit something or be assembled to the car. If the resistance adjust knob is rotated with a slight degree less than 5-10 degree, it will be unable to detect the yellow line. To conclude, the method to fulfil the operational objective to detect various colours is not successful since the solution is still unreliable.

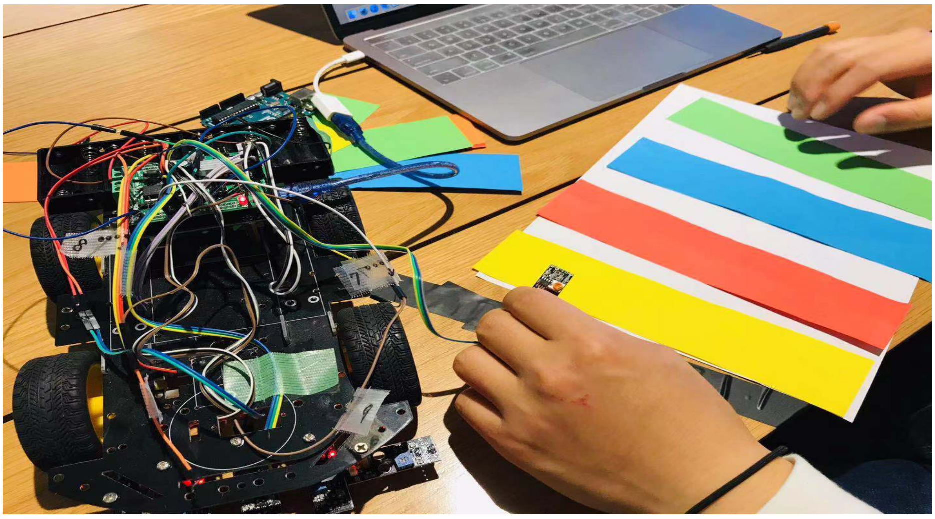


Figure 7.1: Test for detecting lines with different colour

The last performance objective is avoiding vibration while the car is moving. This goal is not achieved in our project. The car could run smoothly and keep at the middle of line. However, when it conducts 45-degree turn or 90-degree turn, especially in 90-degree turn, the car will slow down and vibrate, sometimes the wheels may be too close to the line. The vibration also decreases the speed when the car is turning. In this condition, one proposal is to apply a PID control method to our code to make the turn smoother by cancelling overturn every code execution period.

To conclude, four out of seven original operational objectives are well completed and the other three needs to be improved in future work.

8. Cost Analysis

The entire cost of the system is A$384.1, where a big chunk of the budget (around 48%) is for the procurement of the components, while commercial implementation (involving poster making, product propaganda, and product marketing etc.) accounts for about 21% of the total budget. In the case of saving costs related to human resources (our working) as much as possible, experiments (test site construction) and maintenance (spare parts and batteries) of the robot have been conducted reasonably. The pie chart in Figure 8.1 illustrates the percentage and the cost of each part.

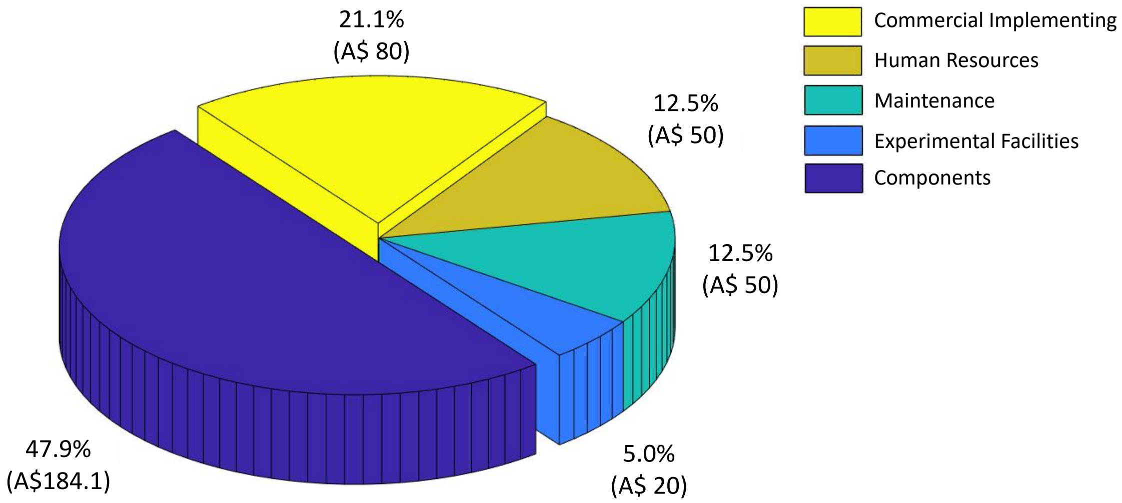


Figure 8.1: Project Cost Analysis

In order to illustrating a more detailed explanation of the cost, the cost and the number of each component applied in the line follower robot can be found in Table 8.1

Table 8.1 Cost of components

Name	Quantity	Cost
M3 screws	10	A$2
Motor fixed locking pieces	4	A$4
Copper columns	5	A$5
Pair of motor and car wheels sets	2	A$4
Bidirectional DC motor	4	A$16.2
20cm Jumper Wire	1	A$5.4
Infrared sensor	5	A$30.5
Arduino UNO R3 Compatible with USB Cable	1	A$32.5
Motor driver shield	1	A$14.3
Battery case	1	A$15
Colored tapes	4	A$8
Car chassis	1	A$12
Battery (AA Bulk Pack + 9V)	1	A$22.6
Tool Box	1	A$12.6
Total	**42**	**A$184.1**

9. Future Work

- In the future, in order to suit the commercial application, the line follower robot will require improvement and development. For instance, in logistics industry, the line follower robot can delivery packages along the specified line or route to significantly reduce manpower costs.

- Meanwhile, various cameras can be mounted on the line follower robot to effectively monitor the environment or workplace in real-time, combining with Internet of Things (IoT) technics. With further development, it can also be used in service industry. For example, in airports, not only it can guide people or people with disabilities to their departure gates, but also it can carry the luggage for them. Hence, it is an open platform that you can mount additional modules to complete different tasks.

- To conclude, there are two major parts to improve the line follower robot in the future. The first one is on the sensors part. You can attach different sensors on the line follower robot. Besides that, power supply is also a direction for improvement. For example, we can mount solar panels on the line follower robot to save more energy. Lastly, the line follower robot can be improved by algorithm, such as PID control algorithm, which reduces the moving error efficiently.

10. Conclusion

Systems Engineering was successfully implemented to achieve the completion of the Line Follower Robot. The fundamental stakeholder requirements were identified in the Project Initiation Document (PID) which helped generate system level requirements and subsequently functional baseline. The functional architecture and physical architecture helped build different concepts for the solution of the project. The detailed design phase was executed to produce the initial prototype of the system to understand the functionality and requirements. This was further developed into a robust and optimal design that incorporated the use of multiple IR sensors and efficient algorithms to achieve the tasks. Throughout the course of the project, the requirements were monitored and ensured that the needs of the customer are prioritized for the best output.

11. Bibliography

[1] Japanese Standard Association, *term about automatic guided vehicle*, retrieved from https://www.jisc.go.jp/app/jis/general/GnrJISNumberNameSearchList?toGnrJISStandardDetailList.

[2] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis*, Pearson Education, Ed., 2011.

[3] US DoD Systems Management College, *Systems Engineering Fundamentals*. United States of America, 2001.

[4]"AS 4024.3303:2017 | Safety of machinery Robots and rob... | SAI Global", *Infostore.saiglobal.com*, 2019. [Online]. Available: https://infostore.saiglobal.com/en-au/standards/as-4024-3303-2017-99432_SAIG_AS_AS_209045/. [Accessed: 22- Aug- 2019].

[5] "ISO/TC 10/SC 6 - Mechanical engineering documentation," *ISO*, 08-Dec-2016. [Online]. Available: https://www.iso.org/committee/46064.html. [Accessed: 22-Aug-2019].

[6] "IEC System of Conformity Assessment Schemes," *IEC Standard - Home*. [Online]. Available: https://www.iecee.org/dyn/www/f?p=106:49:0::::FSP_STD_ID:5275. [Accessed: 22-Aug-2019].

[7] *Saiglobal.com*, 2019. [Online]. Available: https://www.saiglobal.com/PDFTemp/Previews/OSH/As/as2000/2900/2939.PDF. [Accessed: 22- Aug- 2019].

[8] D. Wheel, "DC Geared Motor with Rubber Wheel | Jaycar Electronics", *Jaycar.com.au*, 2019. [Online]. Available: https://www.jaycar.com.au/dc-geared-motor-with-rubber-wheel/p/YG2900. [Accessed: 15- Oct- 2019].

[9] "L293D Motor Driver IC Pinout, Equivalent ICs, Features and Datasheet", *Components101.com*, 2019. [Online]. Available: https://components101.com/l293d-pinout-features-datasheet. [Accessed: 12- Oct- 2019].

[10]"L293D Motor Driver Shield - Wiki", *Wiki.sunfounder.cc*, 2019. [Online]. Available: http://wiki.sunfounder.cc/index.php?title=L293D_Motor_Driver_Shield. [Accessed: 15- Oct- 2019].

[11] A. R3, "Arduino Uno R3", Core Electronics, 2019. [Online]. Available: https://core-electronics.com.au/arduino-uno-r3.html. [Accessed: 26- Aug- 2019].

12. Appendix

(A) Coding

Programming for Arduino UNO is coded by C++ language on Arduino IDE platform as follows.

```
#include <Servo.h>

float Kp = 10, Ki = 0.5, Kd = 0;                    //PID control parameters
float error = 0, P = 0, I = 0, D = 0, PID_value = 0;//Initial the PID inputs and output
float previous_error = 0, previous_I = 0;           //The error of PID input.
static int initial_motor_speed = 150;
int sensor[5] = {0, 0, 0, 0, 0};
int leftMotor1 = 14;
int leftMotor2 = 15;
int rightMotor1 = 16;
int rightMotor2 = 17;
int turnspeed = 150;
int forwardspeed = 150;
int trac1 = 8; //The sensors sort from right side of the front car
int trac2 = 7;
int trac3 = 9;
int trac4 = 2;
int trac5 = 10;

int leftPWM = 3;
int rightPWM = 5;


void tracing(void);
void calc_pid(void);
void motor_control(void);

void setup() {
  // put your setup code here, to run once:
  //initial the serial port
  Serial.begin(9600);
  pinMode(leftMotor1, OUTPUT);
  pinMode(leftMotor2, OUTPUT);
  pinMode(rightMotor1, OUTPUT);
  pinMode(rightMotor2, OUTPUT);
  pinMode(leftPWM, OUTPUT);
  pinMode(rightPWM, OUTPUT);
  //initial the IR sensor port
  pinMode(trac1, INPUT);
  pinMode(trac2, INPUT);
  pinMode(trac3, INPUT);
```

```
  pinMode(trac4, INPUT);
  pinMode(trac5, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  tracing();
  calc_pid();
  motor_control();

}

void motorRun(int cmd, int value)
{
  analogWrite(leftPWM, value);  // set the PWM output (the speed of robot)
  analogWrite(rightPWM, value);
  switch (cmd) {
    case FORWARD:
      Serial.println("FORWARD"); //the state of the output
      digitalWrite(leftMotor1, LOW);
      digitalWrite(leftMotor2, HIGH);
      digitalWrite(rightMotor1, LOW);
      digitalWrite(rightMotor2, HIGH);
      break;
    case BACKWARD:
      Serial.println("BACKWARD"); //the state of the output
      digitalWrite(leftMotor1, HIGH);
      digitalWrite(leftMotor2, LOW);
      digitalWrite(rightMotor1, HIGH);
      digitalWrite(rightMotor2, LOW);
      break;
    case TURNLEFT:
      Serial.println("TURN  LEFT"); //the state of the output
      digitalWrite(leftMotor1, HIGH);
      digitalWrite(leftMotor2, LOW);
      digitalWrite(rightMotor1, LOW);
      digitalWrite(rightMotor2, HIGH);
      break;
    case TURNRIGHT:
      Serial.println("TURN  RIGHT"); //the state of the output
      digitalWrite(leftMotor1, LOW);
      digitalWrite(leftMotor2, HIGH);
      digitalWrite(rightMotor1, HIGH);
      digitalWrite(rightMotor2, LOW);
      break;
    default:
      Serial.println("STOP"); //the state of the output
      digitalWrite(leftMotor1, LOW);
```

```
      digitalWrite(leftMotor2, LOW);
      digitalWrite(rightMotor1, LOW);
      digitalWrite(rightMotor2, LOW);
  }
}

void motor_control()
{
  //int left_motor_speed = initial_motor_speed - PID_value;
  int right_motor_speed = initial_motor_speed + PID_value;
  if(right_motor_speed < -255){
    right_motor_speed = -255;
  }
  if(right_motor_speed > 255){
    right_motor_speed = 255;
  }

  if (error = 0)
  {
    motorRun(FORWARD,right_motor_speed);
  } else if (error > 0)
  {
    motorRun(TURNRIGHT,right_motor_speed);
  } else if (error < 0)
  {
    motorRun(TURNLEFT,right_motor_speed);
  }
}

void tracing()
{
  sensor[0] = digitalRead(trac5);
  sensor[1] = digitalRead(trac4);
  sensor[2] = digitalRead(trac3);
  sensor[3] = digitalRead(trac2);
  sensor[4] = digitalRead(trac1);

    if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3
] == 1) && (sensor[4] == 1)) {
      error = 0;// corner 1 1 1 1 1   forward
    } else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 1) && (
sensor[3] == 1) && (sensor[4] == 0)) {
      error = 0;//small corner 0 1 1 1 0   forward
    } else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (
sensor[3] == 1) && (sensor[4] == 1)) {
      error = 2;//right 0 0 1 1 1    turn right
    } else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (
sensor[3] == 1) && (sensor[4] == 0)) {
      error = 1;//right 0 0 1 1 0    turn right
```

```
  } else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 1) && (
sensor[3] == 0) && (sensor[4] == 0)) {
    error = -2;//left 1 1 1 0 0    turn left
  } else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 1) && (
sensor[3] == 0) && (sensor[4] == 0)) {
    error = -1;//left 0 1 1 0 0    turn left
  } else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[4] == 1))
{
    error = 2;//        0 0 0 0 1  turn right
  } else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (
sensor[3] == 1) && (sensor[4] == 0)) {
    error = 1;//        0 0 0 1 0  turn right
  } else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (
sensor[3] == 0) && (sensor[4] == 0)) {
    error = 0;//        0 0 1 0 0  forward
  } else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 0) && (
sensor[3] == 0) && (sensor[4] == 0)) {
    error = -1;//       0 1 0 0 0  turn left
  } else if ((sensor[0] == 1) && (sensor[2] == 0) && ( sensor[3] == 0) && (sensor[4] == 0))
{
    error = -2;//       1 0 0 0 0  turn left
  } else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (
sensor[3] == 0) && (sensor[4] == 0)) {
    motorRun(STOP, 0); //  0 0 0 0 0 Stop
  }

 Serial.print(sensor[0]);
 Serial.print("---");
 Serial.print(sensor[1]);
 Serial.print("---");
 Serial.print(sensor[2]);
 Serial.print("---");
 Serial.print(sensor[3]);
 Serial.print("---");
 Serial.print(sensor[4]);
 Serial.print("---");
 Serial.println();
}

void calc_pid()  //PID algorithm
 P = error;
 I = I + error;
 D = error - previous_error;

 PID_value = (Kp * P) + (Ki * I) + (Kd * D);

 previous_error = error;
}
```

(B) ANU maker space

ANU maker space is a facility in ANU and it was used to assemble each component and test each function. The place is fundamentally open for ANU student those who want to create novel and creative objects.

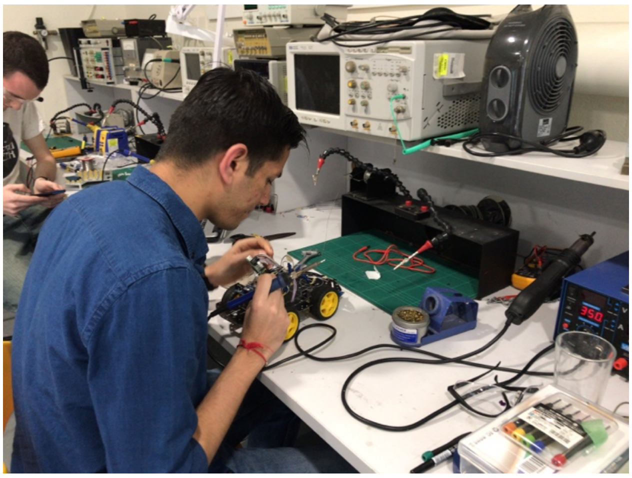


Figure 12.1 Work in ANU maker space