ENHANCING THE CAPABILITIES OF SEARCH AND RESCUE ROBOT (BAMBOT)

by
Naqiatul Amirah binti Ab. Halim
16227

Supervisor
AP Dr. Tun Zainal Azni bin Zulkifli

Dissertation submitted in partial
fulfilment of the requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronic)
JANUARY 2016

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

ENHANCING THE CAPABILITIES OF SEARCH AND RESCUE ROBOT (BAMBOT)

by

Naqiatul Amirah binti Ab. Halim

A project dissertation submitted to the
Department of Electrical & Electronic Engineering
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

Approved by

_____

AP Dr. Tun Zainal Azni bin Zulkifli
Project Supervisor
UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

January 2016

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

_____

Naqiatul Amirah binti Ab. Halim

ABSTRACT

In modern technology, Search and Rescue Robots has been implemented and developed through the advancement of robotics industry. The purpose of this robot is to perform a search and rescue mission where the environment is not reachable by human and the difficulty of a rescuer to locate victims. The problem is due to the visibility and different environmental condition of the mission itself. The solution to this problem is by using an ideal platform type of an autonomous robot with the ability to locate victims and identify any obstacles in front of it.

This project entitled "Enhancing the Capabilities of Search and Rescue Robot (BamBot)" is focusing on enhancing the capabilities of Search and Rescue Robot (BamBot) that can maneuver autonomously to a desired location while identifying the presence of obstacles and avoid it and also to identify a method, algorithm and technique that is suitable for search and rescue mission. The paper describes the usage of a search and rescue platforms with different application and operate in various environments with the ability of planning the path of the robot whether it is using local motion planning or the global path finding. This paper also discusses a suitable algorithm that shall be used to achieve the most enhanced search and rescue robot capabilities. These capabilities include the speed, the system, and the duration of the robot to perform search and rescue mission. Improvements in navigation, vision, communication and sensory systems is also a crucial part to provide an efficient mobile robotic platform.

The proposed search and rescue robot is designed based on one of the previous technologies reviewed with several changes carried out to improve the system and to make the system more stable. Several testing have been done during the implementation schedule. The results from each testing turns out to be successful considering the total current of all components are less than 2.2A. Since the battery used in this project is LiPo type of battery, the robot can sustain and maintain its system for at least an hour.

ACKNOWLEDGMENT

First and foremost, I would like to praise and thanks Allah S.W.T that with His blessings and the strength He granted me, I am able to complete my final year project (FYP) entitled “Enhancing the Capabilities of Search and Rescue (BamBot)”. These two semesters of completing this project were full with ups and downs that dependence onto Him is among others that make me persistent until the last phase.

My utmost gratitude goes to my supervisor; AP Dr. Tun Zainal Azni bin Zulkifli for his support and guidance throughout the project execution. His full commitment in supervising and also giving motivation has been very significant to me. Other than that, I would also thank my sponsor which is Majlis Amanah Rakyat (MARA) that has been providing sponsorship throughout my five years of study in UTP.

My appreciation also goes to my classmates, we have been very supporting to each other and at certain point we even know the current progress and issues faced by each of the members and somehow they have assisted me directly or indirectly throughout the process of completing the project.

Lastly, to my parents, Ab. Halim bin Mohamed and Artinah binti Mohamed Salleh who are very supportive, their encouragement and motivation has been one of the keys that strengthen me in enduring these two semesters.

TABLE OF CONTENTS

CERTIFICATION OF APPROVAL ii

CERTIFICATION OF ORIGINALITY iii

ABSTRACT iv

ACKNOWLEDGMENT v

TABLE OF CONTENTS vi

LIST OF FIGURES ix

CHAPTER 1 1

INTRODUCTION 1

1.0 INTRODUCTION 1

1.1 PROJECT BACKGROUND 1

1.2 PROBLEM STATEMENT 2

1.3 OBJECTIVES OF STUDY 2

1.4 SCOPE OF STUDY 3

CHAPTER 2 4

LITERATURE REVIEW 4

2.0 INTRODUCTION 4

2.1 SEARCH AND RESCUE ROBOT 4

2.2 PREVIOUS TECHNOLOGIES FOR SEARCH AND RESCUE ROBOT 5

2.2.1 Path Planning 5

2.2.2 Path Planning Algorithm 6

2.2.3 Monitoring/Vision Algorithm 10

CHAPTER 3 13

METHODOLOGY 13
3.0 INTRODUCTION 13
3.1 PROJECT ACTIVITIES 13
3.1.1 Research and Study 13
3.1.2 Data Gathering and Analysis 14
3.1.3 Preliminary Design 14
3.1.4 Components Assembling 14
3.1.5 Circuit Building, Simulation and Coding 14
3.1.6 Testing 15
3.1.7 Project Demonstration 15
3.2 TOOLS 15
3.2.1 Robot Platform 15
3.2.2 Arduino MEGA 16
3.2.3 Raspberry Pi Model B 17
3.2.4 Camera Module 17
3.2.5 Ultrasonic Sensor 18
3.3 PROJECT PROGRESS, GANTT CHART AND KEY MILESTONE 19
CHAPTER 4 22
RESULTS AND DISCUSSION 22
4.0 INTRODUCTION 22
4.1 RESULTS 22
4.1.1 Ultrasonic Sensor 22
4.1.2 IR Sensor 23
4.1.3 Cytron Motor Driver MDD10A 25

4.1.4 Gripper System 26

4.1.5 Power Consumption 28

4.1.6 Image Processing Testing 29

CHAPTER 5 36

CONCLUSIONS AND RECOMMENDATION 36

REFERENCES 37

LIST OF FIGURES

Figure 1: Djikstra's Algorithm 7

Figure 2: Greedy Best-First-Search Algorithm 8

Figure 3: A* Search Algorithm 9

Figure 4: Concept of Image Processing 10

Figure 5: HSL in RGB colour 10

Figure 6: HSV in RGB colour 10

Figure 7: Project Flow 13

Figure 8: Dagu 4WD Wild Thumper 16

Figure 9: Logitech C210 17

Figure 10: Image Processing Implementation Flow 18

Figure 11: IR Sensor Connection and Code in Arduino IDE 24

Figure 12: Dual Channel 10A DC Motor Driver 26

Figure 13: Gripper Mechanism 26

Figure 15: Fully open and close gripper 27

Figure 14: Gripper Test Code 27

Figure 16: Current needed by Servo Motor (Gripper System) 27

Figure 17: Example of LiPo battery 28

Figure 18: Center of Contour Code in Python 30

Figure 19: Original Image 30

Figure 20: Threshold image 31

Figure 21: Step by step detection of center contour 31

Figure 22: Threshold value = 45 32

Figure 23: Threshold value = 60 32

Figure 24: Detecting Shapes 1 33

Figure 25: Detecting Shapes 2 34

Figure 26: Colour detection 35

CHAPTER 1
INTRODUCTION

1.0 INTRODUCTION

Chapter 1 provides a general overview of Search and Rescue Robot (BamBot). This chapter outlines some of the general aspects in the project including the project background. The problem statement of this project is clarified followed by the project objectives. Finally, the scope of study of this project is expressed at the end of this chapter.

1.1 PROJECT BACKGROUND

A search and rescue robot is a robot that is designed following the needs to rescue people. Based on the online Oxford's Dictionary, a robot is a machine that is capable of carrying out a complex series of actions automatically, especially one that is programmable by a computer [Robot, n.d.]. In other words, a robot can be defined as a machine that is programmable and is capable of achieving sets of action autonomously. Later in this chapter, the word autonomously will further be explained in the next section and its relationship to the project.

Robots can perform a variety of applications in daily life including carrying out heavy objects and repetitive work in order to replace manpower needed in certain industry [Engelberger, 1980]. In some of the cases, human is prohibited to enter certain dangerous conditions or situations where the loss of human lives is likely to occur. This is why robots are designed to take over human roles over this threatening cases. Some of the examples including military services which are Explosives Ordinance Disposal (EOD) robots, space exploration services where Remotely Automated Vehicle (RAV) and the Remote Manipulator System (RMS) where both of the robots are unmanned and can move freely to perform space mission. Nowadays, work has been shared between human and machine. Since robot can move autonomously and they are technically advanced than human, they shall learn how to do the work faster, have endless energy levels and their precision in performing work is the quality any employers would have hired [10 Things We Couldn't Do Without Robots, n.d.].

A robot that performs a specific task without the guidance of a person is known as the autonomous robot. Furthermore, an autonomous robot has the ability to think for itself based on the algorithm provided. Basically this autonomous robot has sensors to obtain information from the environment examples like cameras, IR sensors, sound sensors, ultrasonic sensors and so on [Bekey, 2005]. This project aims to develop a robot called Search and Rescue Robot (BamBot), which can maneuvre autonomously around the obstacles and complete the task given which is by bringing the selected object to a specific point. The purpose of developing this robot is to be able to be used in a rescue mission without endangering human's life. Hence the robot should be able to autonomously move avoiding the obstacle and head towards its desired locations.

1.2 PROBLEM STATEMENT

At present, due to technological advancement and promulgation in terms of robotic usage, autonomous robot shall be implemented in many applications worldwide. However, there are several problems which need to be taken care of and require profound study including the design of the robot which are;

i) How can the robot maneuver itself to a desired location with the presence of obstacles and is able to avoid it?
ii) What is the preferable method or algorithm that is suitable for search and rescue mission?
iii) How can the robot find victims, identify them and transport them to desired locations?

Based on the problem statements mentioned above, the project is further outlined and defined on the following chapters.

1.3 OBJECTIVES OF STUDY

The main objectives of this project are;

i) to enhance the capabilities of Search and Rescue Robot (BamBot) that can maneuver autonomously to a desired location while identifying the presence of obstacles and avoid it;

ii) to identify a method, algorithm and technique that is suitable for search and rescue mission.

1.4 SCOPE OF STUDY

This project focuses on implementing and enhancing the capabilities of a Search and Rescue Robot. In the past, there are some researchers that have come out with the idea on performing and delivering their Search and Rescue Robot in time but when it comes to the implementation part, this is where some minor technical issues arise. The idea here is to enhance their robot's capabilities in getting accurate and precise output, successfully with minimal error. Basically, going back to the objectives of the study, the research focuses on how to navigate and avoid collision with obstacles, find objects, identify them and transport them to a desired location with minimal duration of time using a specific method or algorithm. Thus, a profound research and selection of suitable components with high accuracy of result is also very crucial in this project.

CHAPTER 2
LITERATURE REVIEW

2.0 INTRODUCTION

Chapter 2 provides literature review on the past researches and studies related to the project. The literature review focuses on two of the main aspects for this project which are the importance of search and rescue mission and previous technology for Search and Rescue Robot including the path planning and the algorithm used in the past researches.

2.1 SEARCH AND RESCUE ROBOT

Urban search and rescue mission robots shall be capable of working in a number of environments such as crossing terrains that may be covered with debris. One of the design approach is that the robots must be able to pass through some obstacles or man-made structures such as stairs and thresholds around doors. The environment that requires this type of robots are the ones that may be difficult for humans or dogs to search [Sam et al., 2009]. The importance of Search and Rescue Robots shall be explained in the next section.

Search and Rescue Robotics is a new research field that deals with systems that support first response units in disaster missions. In the urban rescue missions, mobile robots can be extremely beneficial tools after catastrophes such as earthquakes, bomb explosions, gas explosions or even daily incidents, i.e. fires and road accidents which involves any hazardous materials and dangerous situations and conditions. The robot that are designed to fulfil this rescue missions can be used to investigate any collapsed structures, access the situation and the safety of surrounding areas will be able to search and locate any victims around them [Search and Rescue Robot: What is it all about?, n.d.]. Other than their contribution in any rescue missions that involves harsh environment, the advancements of robotics industry nowadays had contributed towards all kinds of aid and act as a helping hand in the heavy industry.

2.2 PREVIOUS TECHNOLOGIES FOR SEARCH AND RESCUE ROBOT

2.2.1 Path Planning

Path planning is one of the essential element for designing an autonomous mobile robot. Based on the statement before, path planning is defined as the determination of the best path that the robot must take in order to get to the target locations [Buniyamin et al., 2011].

Path planning from three literatures [Abiyev et al., 2010; Al-Taharwa et al., 2008; Buniyamin et al., 2011] are discussed in this section. These three literatures tell us the difference between global path planning and local motion planning and also the suitable path planning that is more practical to use.

R. Abiyev, D. Ibrahim, B. Erin stated that the navigation system task is to guide the robot towards a target point without any collision with any unwanted obstacles. They also stated that in robotics, a robot that is designed to be swift and efficient following the procedure for navigation of mobile robots is one of the crucial parts in robotics industry. Over the past years, many researches has come out with special algorithm which has helped the robot in avoiding obstacles [Abiyev et al., 2010]. There are two major classifications of this research field which are; i) the global path planning [Al-Taharwa et al., 2008] and ii) the local motion planning [Buniyamin et al., 2011].

In global path finding, the location of obstacles and the environment surrounding the search and rescue robot are well known ahead. So, this is where the robot is required to move around and avoid the obstacles and reach its destination. The three main things to take into account in completing the robot path is by knowing the coordinates of the starting point, coordinates of the destination point, and the obstacles coordinate [Al-Taharwa et al., 2008].

On the other hand, a local motion planning is a path planning that is being implemented during the movement of the search and rescue robot which in other words, the algorithm of the path may change in the response of environmental changes [Abiyev et al., 2010]. It will guide the robot dynamically in accordance to the locally sensed obstacles that appear to be blocking the robot's path [Buniyamin et al., 2011].

Based on these two motion planning that needs to be injected to the robot system, the local motion planning is more suitable and more practical to use. This is because the

location of every obstacle are unknown and it is hard to figure out if any unwanted obstacles may present at the time of search and rescue mission.

2.2.2 Path Planning Algorithm

There are several algorithms used by researchers which can be used to find the shortest path for a robot to maneuver around a specific area. A simple robotic application, for instances maze solving robot or obstacle avoidance robot may use the following algorithms. Some commonly used algorithms are as follows;

i. Wall-Follower Algorithm (Left-Turn and Right-Turn Algorithm)

As the name implies, the robot will only take left or right turn during its maneuvering time around the specific area of interest while finding the closest wall; whether it is right or left wall, until it reaches the desired locations. The robot will then move along the wall while keeping some distance away from the wall. This algorithm is very effective in designing a maze solving robot which the maze is wall-linked to the target point [Abdramane, 2015].

ii. Dijkstra's Algorithm

Dijkstra's algorithm is widely known as a single-source shortest paths problem solver. The algorithm works by broaden its coverage outwards from the starting point to the end point. Study shows that using Dijsktra's algorithm, it is guaranteed that the robot will travel the shortest path. Based on Figure 1, the starting point is labelled with a red star, the destination point is labelled with purple cross. The left side of the picture shows the shortest route for the robot to travel, which is 7 steps (including the count of obstacles). On the other hand, the picture on the right side shows the shortest route the robot shall take and to avoid all the obstacles placed on the map.

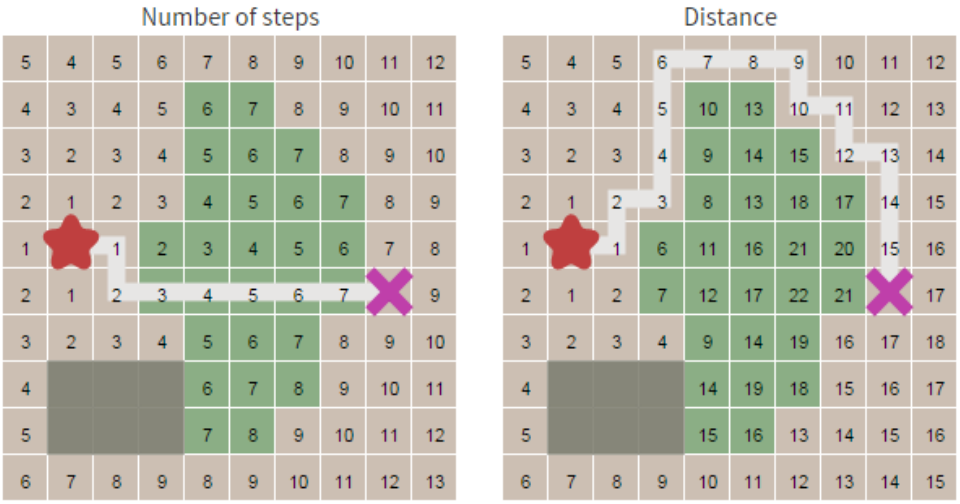


Figure 1: Djikstra's Algorithm

iii. The Greedy Best-First-Search Algorithm

The Greedy Best-First-Search Algorithm works pretty much similar to Dijkstra's algorithm. The only difference is that Greedy Best-First-Search has some *heuristic* (estimate) function. It shall estimate how far any vertex from the ending point is. Although selecting any vertex that are closest to the ending point is the main objective, it does not guarantee whether the robot shall go through the shortest path or not. The advantage of Greedy Best-First-Search is that, it runs much faster than Dijkstra's algorithm because of the applied heuristic function that will guide the robot to move rapidly towards the target point [Patel, 2014]. Figure 2 shows the algorithm works in finding the shortest path where;

i. the ● (blue dot) will represent the starting point;
ii. the ● (red dot) will represent the destination point;
iii. the ■ (blue square) will represent the obstacles;
iv. the ■ (green square) will be the 1st priority for the robot to travel;
v. the ■ (yellow square) will be the 2nd priority for the robot to travel;
vi. the ■ (purple square) will represent the shortest path for the robot to travel.

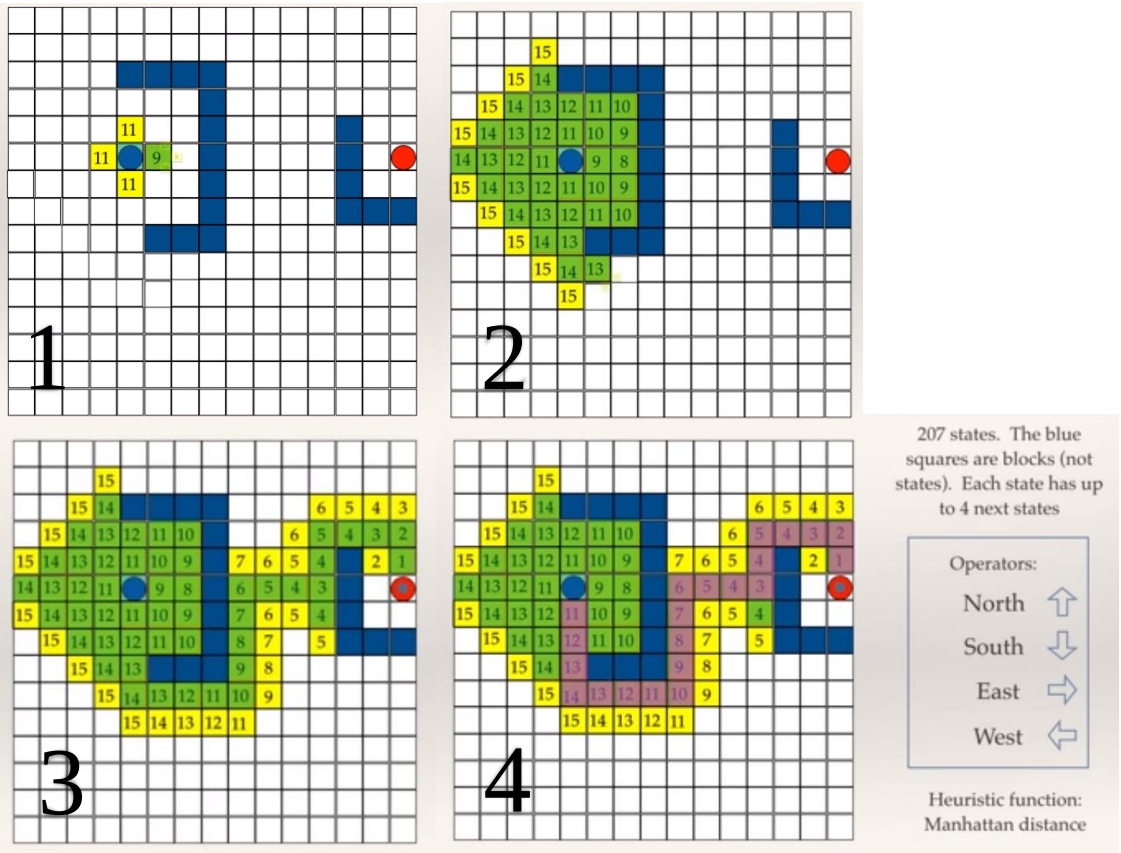


Figure 2: Greedy Best-First-Search Algorithm

Figure above shows the step-by-step on how the selection of path being made by Best-First-Search algorithm. At first, the robot chooses to move east but in the case of being blocked by obstacles, it shall make a U-Turn like the one being shown in 2. The green squares are the possible path that could be travelled by the robot, shown in 3. In the end, the robot will travel along the purple squares as it has been confirmed to be the shortest route to the destination point with the help of Greedy Best-First-Search algorithm.

Based on the explanations above, the Greedy Best-First-Search algorithm may not be the perfect method or algorithm that can be used by the robot in finding the shortest route. It is because the algorithm is 'greedy' to move to the destination point even though the path is not correct. However, this algorithm has some advantages which will help in accomplishing the objectives as stated in Chapter 1. One of the advantages is that, Greedy Best-First-Search algorithm requires short time in simulating or finding the shortest route for the robot to travel. Besides, it also minimizes the usage of memory used in running this algorithm since it uses only a simple calculation in determination of route.

iv. A* Search Algorithm

Basically, A* algorithm is the most popular choice of algorithm for path finding. It is flexible, adjustable and can be used in a wide range of applications. The function is very similar to Djikstra's algorithm where it can find the shortest path to be travelled by a robot and it uses the heuristic function from Greedy Best-First-Choice to guide itself. In other words, A*algorithm is the combination of Djikstra's algorithm and Greedy Best-First-Search algorithm. What differentiates between A* algorithm and Greedy Best-First-Search algorithm is that, it considers the distance that has already been travelled by the robot. The equation below shows the A* algorithm resolves the shortest path to travel.

$$\mathbf{F(n) = G(n) + H(n)}$$

Where,

- **i.** G represents the movement from the starting point following the path generated along the way;
- **ii.** H represents the heuristic function which will estimate the movement to the ending point;

Figure 3 illustrates how A* algorithm works. Based on figure below, orange square represents the starting point while blue square represents the ending point, H (yellow squares) represents the farthest node that is located from the ending point while G (teal squares) represents the farthest node that is located from the starting point. This is where A* algorithm plays it part in balancing G and H values as it moves from the starting point to the ending point. Here, if the value of H is 0 for all points, only G plays role, it is a special case to A* algorithm where it is actually Dijkstra's algorithm [Woo-Jin et al., 2009].

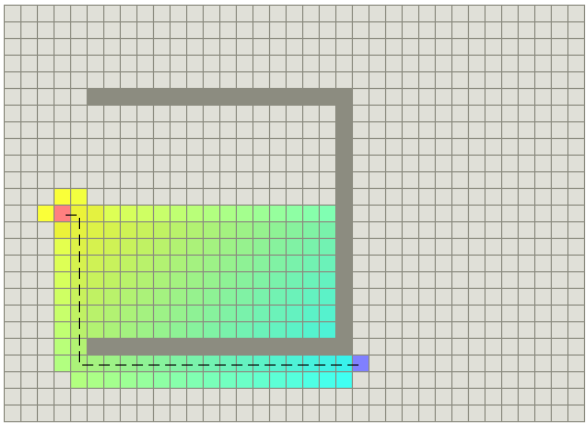


Figure 3: A* Search Algorithm

2.2.3 Monitoring/Vision Algorithm

Image processing is a form of analysing and manipulating a signal process where the input is taken from an image captured using a camera while the output is the processed image where it includes a set of specifications and parameters related to the image [Gonzalez, et al., 2008]. In acquiring a digital image, it has come to attention that the problem lies from an undesirable camera shakes due to random camera movement. To remove these unwanted camera shakes, image enhancement algorithms are required in overcoming this problem [Shilpashree et al., 2009].

i. Basic Concept of Image Processing

Figure 4: Concept of Image Processing

Based on figure above, it shows that the concept of image processing is a step of 3 stages; input, processor and output. Basically the input will be obtained by using a camera module connected to the Raspberry Pi. It will then be sent to focus on a given image's pixel then output it as a processed image.

ii. Image Processing Algorithm

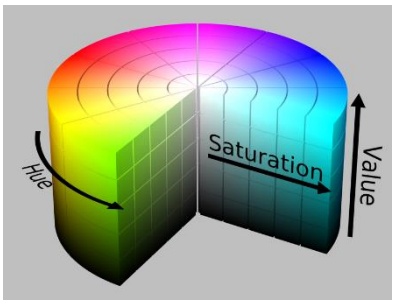


Figure 6: HSV in RGB colour

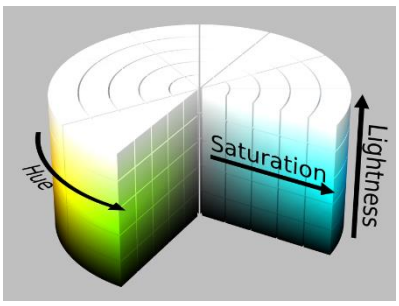


Figure 5: HSL in RGB colour

Figures above show cylindrical geometries where HSL stands for Hue, Saturation and Lightness while HSV stands for Hue, Saturation and Value, or HSB (brightness). Raspberry Pi will obtain an image from the camera in

either of these formats. It will then perform the image processing and send the processed image data to the robot. The angular dimension, 0° red, 120° green and 240° blue and wrapped back at 360°. Basically, there are three stages of processing an image. There are three main steps in processing an image. The steps will be explained in the next section.

iii. Steps in processing an image

a) Colour detection

Basically the goal is to detect colour. Using the values from the HSV within the threshold value, selected colour can be detected. The objective of this colour detecting and filtering approach is to apply a reference colour to an image for filtering out the range of colours which is too far off from the reference value. It will then compare the presence of specific colour within the image.

b) Contour detection

Canny algorithm, a multi stage algorithm is used to detect wide ranges of edges in images. The process of Canny algorithm can be broken down into five different steps [Canny. J, 1986]

- Removing noise: Gaussian filter is used to smoothen the image from noise but will miss some of the weak edges
- Intensity gradients of the image is to be found where it uses four filters to detect diagonal, horizontal and vertical edges in an image
- Edge thinning technique: Non-maximum suppression is applied to get rid of false response in the detection of edge of an image. Still, there will be some noise in the edge pixel
- To remove noise and colour variation from weak edges pixel, double threshold is applied to determine potential edges
- Binary Large Object (BLOB) is used to track the edge of the image by hysteresis. It suppresses all other weak edges that are not connected to the strong ones.

c) Shape detection

This algorithm shall perform some simple detection of geometrical shapes for example square, circle or rectangles. The algorithm will go through the list of all provided points in the BLOB analysis and checks

whether it fit correctly into the assumed shape. During the check the algorithms goes through the list of all provided points and checks how accurately they fit into assumed shape. For this project, the robot may require to find the obstacles and victims. Victims may be in term of square boxes while the obstacles may be in round or triangle shapes [Aforgenet, n.d.].

CHAPTER 3
METHODOLOGY

3.0 INTRODUCTION

This chapter discusses the project activities and the flowchart as the steps taken to achieve the objectives that have been mentioned in Chapter 1. The detailed explanation to every steps taken are further elaborated. It shall include the tools and components involved throughout the project. The following sections will clarify on the key milestone through Final Year Project 1 (FYP1) which is then followed by Gantt chart.

3.1 PROJECT ACTIVITIES

This project is a step-by-step basis that shall be conducted through few stages and activities. Project activities include research and study, data gathering and analysis, preliminary design phase, components assembling, circuit building, simulation and coding, testing phase and lastly the project demonstration. Figure 7 provides the information on the step-by-step procedure in completing the project within allocated time.

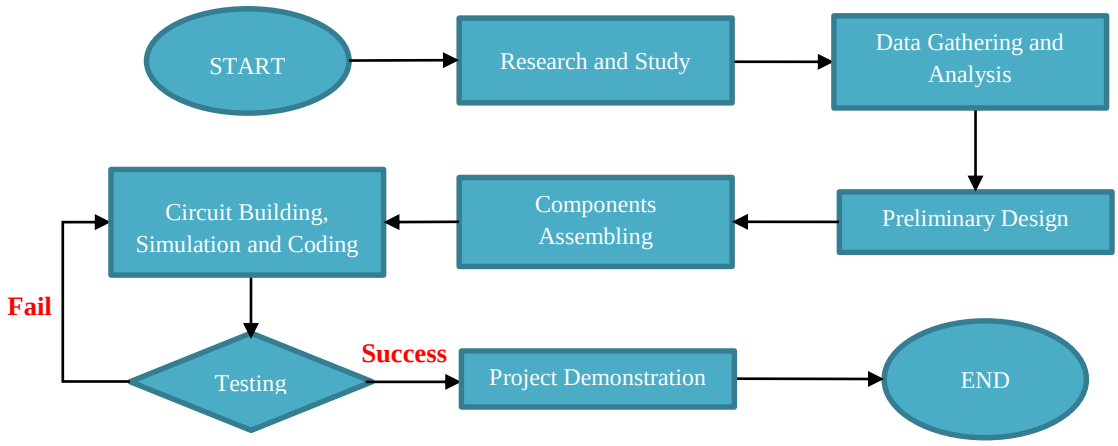


Figure 7: Project Flow

3.1.1 Research and Study

At first, quantitative research and study shall be done throughout the first phase of this project. This is to get some useful information to be taken into account before moving into the part where implementation of robot shall be done. Based on the objectives,

there is need to design a robot that can be used in search and rescue mission. Up to this point, some research has been made by reading through some research papers and journals published in the past to make sure extensive information can be obtained from the papers. The information obtained from past research papers including the importance of search and rescue robot in the world nowadays, the method in designing a search and rescue robot that is autonomous, while instantaneously perform the task given to it successfully without any technical issues arise afterwards.

3.1.2 Data Gathering and Analysis

There are some of the information that needs to be taken into account such as the type of sensors that need to be used in this project, the suitability and specifications of components that will be integrated into the full system and also the design approach of this robot to make sure that it can work properly.

3.1.3 Preliminary Design

The preliminary design phase of this project is one of the important phase in designing a fully functioning robot. The proposed design was presented during FYP1 which is before the proposal defense presentation. The design of the search and rescue robot shall take place after all the gathered data are taken into account. In preliminary design phase, the aim is to make sure that the robot will show an interesting appearance while maintaining the function at its best. It is optimum that the robot designed is user friendly, reliable, cheap yet effective and also uses less power consumption.

3.1.4 Components Assembling

The components shall be available in the Electrical and Electronics Department Store (EE Store), can be purchased online from Cytron Technologies website and any other shops that provide all the electronic parts of this project. During this time, the cost structure of this project is very important that it should be within the allocated budget given by University Teknologi Petronas (UTP).

3.1.5 Circuit Building, Simulation and Coding

After all the components have been assembled, a simple circuit shall be built at first that will integrate all the components together. This is to ensure that all the components can work and function properly. Initially, the coding and all the algorithm for this robot will be plugged in into the Raspberry Pi Model B and will be tested later whether it

can give the desired output. Then, the Raspberry Pi Model B will be integrated with all other components so to observe the performance of the robot.

3.1.6 Testing

During the testing phase of this project, the robot is expected to deliver all the objectives stated at the first chapter successfully. If it works properly, some modification can be applied during this process in making the performance of the robot at its peak. The project shall run through the circuit building phase again for any alterations made. On the other hand, if the robot does not deliver satisfactory results, the circuit building, simulation and coding phase will be executed one more time. An accurate and precise output is expected for this project where the robot can move autonomously and give outstanding results.

3.1.7 Project Demonstration

The project will lastly be demonstrated during FYP II.

3.2 TOOLS

Listed below are the tools that will be used throughout this project; Arduino MEGA, Raspberry Pi Model B, Camera Module, Ultrasonic Sensor and IR Sensor.

3.2.1 Robot Platform

Robot platform that will be used in this project is a rugged platform, 4-wheel drive chassis from Dagu Electronics. Basically, this type of chassis can move around any rough terrain, which is considered as the best platform to be used for search and rescue mission. This chassis consists of 4 powerful brushed DC geared motor with 75:1 steel gearboxes that drive large tyres which is 120mm in diameter. Other than that, this chassis comes together with a super-twist suspension system, where the advantage is to keep each wheel in contact with the ground at all times even when crossing over bumpy and uneven surfaces. When powered at 7.2 V, this chassis can reach a top speed of approximately 3 km/h (2 mph), and each motor has a stall torque of roughly 11 kg-cm. Below shows the specifications of Dagu 4WD Wild Thumper.

Specifications

Size: 280 × 300 × 130 mm (11" × 12" × 5")

Weight: 1.9 kg (4.1 lb)

Recommended motor voltage: 12V

Stall current at 12 V: 5.6 A per motor

No-load current at 12 V: 300mA per motor

No-load output shaft speed at 12 V: 130 RPM

Stall torque at 12 V: 17 kg-cm per motor

Figure 8: Dagu 4WD Wild Thumper

3.2.2 Arduino MEGA

The proposed system is modular with main processing unit resides in the Arduino Mega block. The approach of separated defined modules simplifies any troubleshooting task by isolating each module with and its functionalities. The communication between different modules and the main processing unit is managed by the I2C protocol, which would be responsible for transferring data between different sensing modules to the Arduino Mega, which in turn produces the correct commands to transfer them to the corresponding operation modules.

Sensing Modules

- Camera module and Raspberry PI
- IR Sensor
- Ultrasonic Sensors

Operational modules:

- Gripper
- Motor Driver Unit

Arduino Mega will store and execute the main algorithm. It will be integrated with ultrasonic sensor to detect any unwanted obstacles, control the speed of the motors for

the robot to move and get the signal from Raspberry Pi which is required to reach the desired location.

3.2.3 Raspberry Pi Model B

Raspberry Pi acts as the controller as it is defined as a low cost, credit-card sized computer that plugs into a computer monitor or a TV [“What is Raspberry Pi?”, n.d.]. In this case, Raspberry Pi will be connected to a Logitech C210 camera. The input from the camera will be processed by Raspberry Pi so the robot can detect the current location of objects or obstacles, preferred paths and so on based on the algorithm. The data of locations mentioned above serves as an environment mapping that is ready to be sent to the robot on request. The primary criteria of this project is the capability of the robot to recognize objects whether it is an obstacle, a victim, the starting point and the destination point. Therefore, the implementation of image processing is required to differentiate between object, utilizing colour filtering and shape recognition approach.

3.2.4 Camera Module

This model of camera is used to identify the objects based on colour and size using an image processing algorithm. This camera will act as the sensor and will be placed at the front side of the robot. This is because the camera will be attached together with a pan and tilt kit which consists of mini servos. The camera will scan the entire game field and perform environment mapping. Figure below shows the camera module that will be used in this project.

Image processing will be performed by combining a Raspberry Pi and a camera module as its vision system to capture whether it is a still image or a video. Figure 10 shows the proposed method of implementation.

Figure 9: Logitech C210

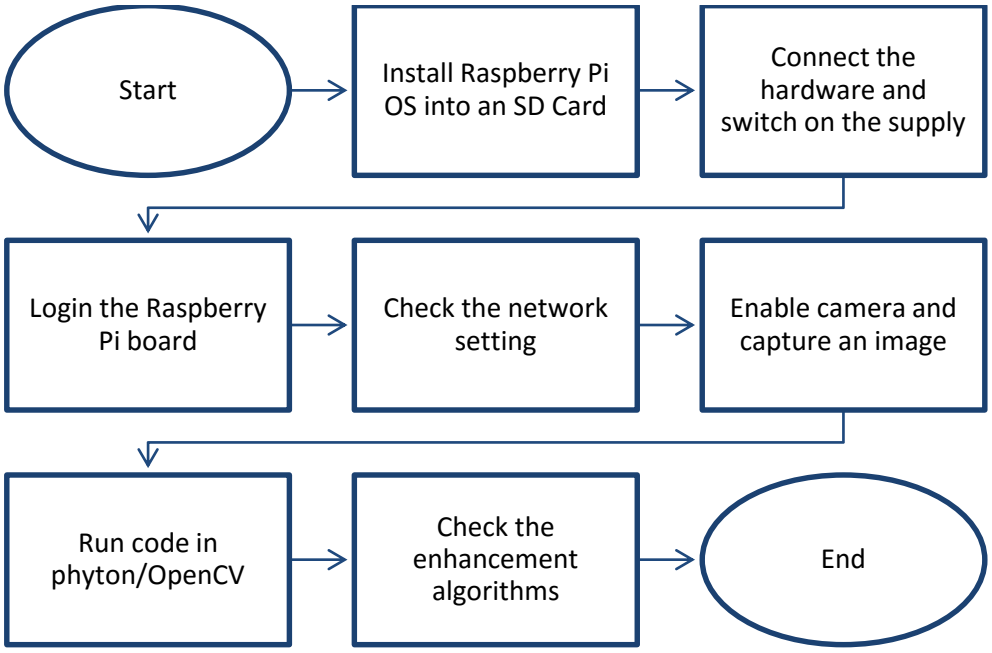


Figure 10: Image Processing Implementation Flow

3.2.5 Ultrasonic Sensor

With the use of the camera stated above, to plan the robot's path where it will show all the obstacles along the way, a secondary system to detect any obstacles and line detector is needed. In this case, it is advised to use four ultrasonic sensors which will be placed at each four corners of the robot. Ultrasonic sensor uses sonar to determine obstacles and works by interpreting the echoes of sound wave (transmitting wave and receiving waves). This is to ensure the path will not contain any unexpected obstacles and if it does, the robot will still detect the obstacles within the range of 15cm and stopped from moving.

The ultrasonic sensor will detect the obstacles in front of it and sends the feedback to Arduino and it takes about 5 seconds for the feedback to be processed. If the obstacles are still placed in front of the robot, the ultrasonic once again send the feedback to the camera to recalculate the path for the robot.

3.2.6 Motor Driver

Usage:

- Typically used in controlling motors speed and direction,
- Acts as a current amplifier, where it takes a low-current control signal and then turn it into a higher-current signal that can drive a motor.

How it works:

- An H-Bridge is a circuit that can drive a current in either polarity and be controlled by *Pulse Width Modulation (PWM).
- The speed of the motors can be adjusted by connecting PWM outputs from your robot's microcontroller to the ENA and ENB input pins on the motor driver board. The ENA pin controls Motor A and the ENB pin controls Motor B. When these pins are HIGH, power is output to the motor.
- By using PWM, we are turning power on and off very quickly to adjust the speed of the motor. The longer the PWM duty cycle is, the faster the motor will turn. Recommended PWM duty cycle is 90% or less.

3.2.7 IR Line Sensor

The robot shall be programmed not to move exceeding the white/black line of a search and rescue field. So, this sensor will be located at the bottom centre of the robot platform where it will detect the white line of the field. IR LED is used to emit infrared light. IR photodiode current is small (uA), a large resistor is used to maximize the voltage. For the interrupt input a switching concept using transistor is used to detect bright or dark surface.

3.3 PROJECT PROGRESS, GANTT CHART AND KEY MILESTONE

The progress of the project is currently right on track. Based on the Gantt chart, currently in week 14, interim report should be submitted. Presently, the research in overcoming some problems is still ongoing. The testing done during FYP1 will be continued and will be integrated as one system in FYP2.

	DETAILS	WEEK													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	Title Selection and Confirmation														
2.	Preliminary Research Work on Bambot														
3.	Preparation of Extended Proposal														
4.	Initial Preparation on Bambot and Gather All Information on Components To Be Used														
5.	Extended Proposal Submission														
6.	Finalizing The List Of Components Needed And Drafting Price Quotation														
7.	Preparation for Proposal Defence														
8.	Proposal Defence and Progress Evaluation														
9.	Purchase Components and Tools														
10.	Preparation of Interim Report														
11.	Draft of Interim Report Submission														
12.	Interim Report Submission														

Key Milestone

Table 1: Project Gantt Chart and Key Milestones for FYP1

	DETAILS	WEEK													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	Testing components individually and independently														
2.	Data gathering and analysis of method														
3.	Start to integrate the components together														
4.	Testing whole system altogether														
5.	Progress Report Submission														
4.	Modification in design/algorithm used (if any)														
5.	Upgrade, enhance capabilities and improve design and system														
6.	Pre-EDX														
7.	Draft Report														
8.	Final Report														
9.	Viva / Deploy the robot in field /														

Key Milestone

Table 2: Project Gantt Chart and Key Milestones for FYP2

CHAPTER 4
RESULTS AND DISCUSSION

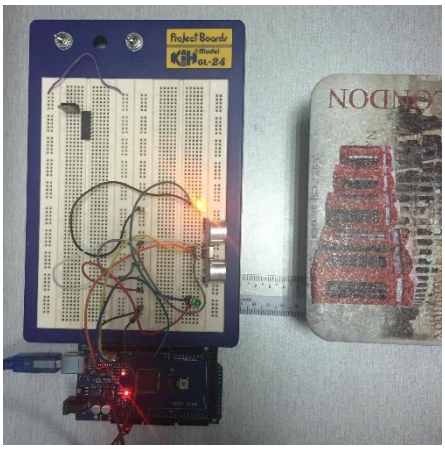
4.0 INTRODUCTION

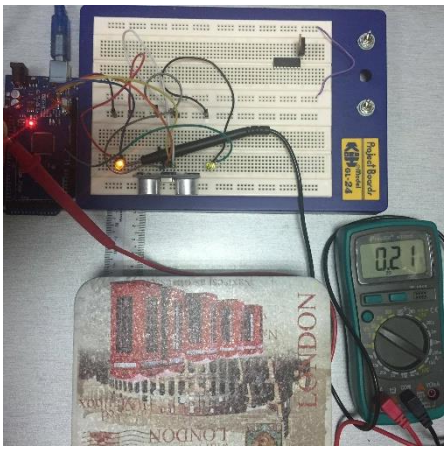
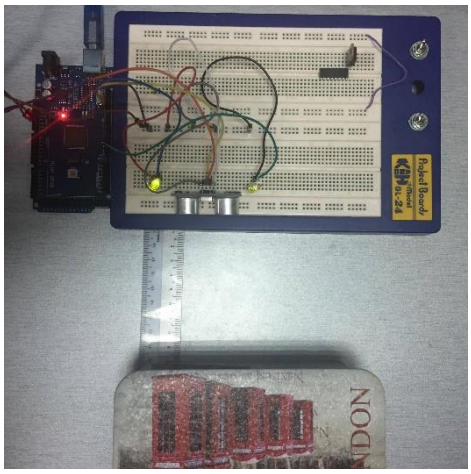
This chapter provides the results from the testing and simulation the proposed design using Arduino IDE software. In FYP 1, the aim is to implement each subsystem one by one before integrating it into one system. This way, any unwanted malfunctions in a subsystem can be prevented and troubleshooting will be easy.

4.1 RESULTS

4.1.1 Ultrasonic Sensor

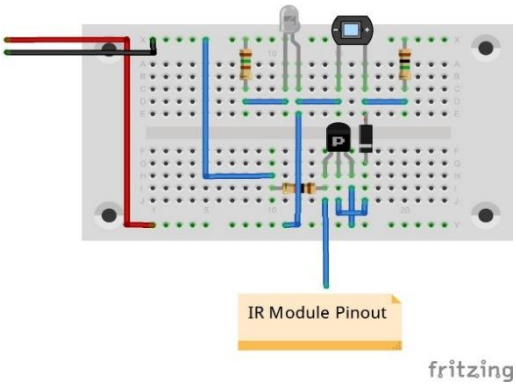
To ensure ultrasonic sensor effectively operate in certain conditions, a test using ultrasonic sensor which is connected with Arduino was conducted. For testing purposes, the safe range is selected to be at 5cm. There are 2 cases which are; a) obstacle is in range, b) obstacle is not in range. Table below shows the result obtained from the testing of the ultrasonic sensor.

Testing Equipment	Result
	i. Obstacle Detected Yellow LED is lighted up when there are obstacles detected within the range of 5cm.

<table>
<tr>
<td> 
 </td>
<td>i. Obstacle Not Detected
➢ The first picture shows when the obstacles is 6cm away from the sensor.
➢ The second picture shows when the obstacles is 10cm away from the sensor.
➢ LED is not lighted up when when there are no obstacles detected within setting distance.</td>
</tr>
</table>

4.1.2 IR Sensor

IR Sensors work by using a specific light sensor to detect a select light wavelength in the Infra-Red (IR) spectrum. By using an LED which produces light at the same wavelength as what the sensor is looking for, the intensity of the received light can be obtained. When an object is close to the sensor, the light from the LED bounces off the object and into the light sensor. This results in a large jump in the intensity, which is known can be detected using a threshold.



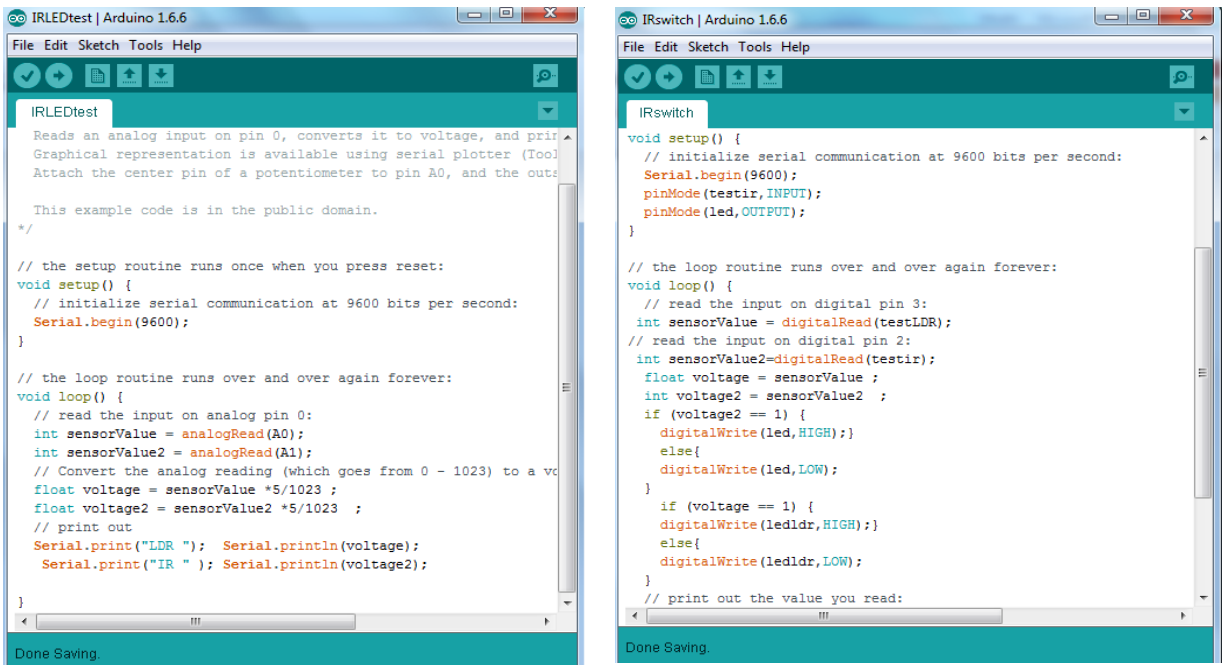


Figure 11: IR Sensor Connection and Code in Arduino IDE

Condition	Without Reflection surface	White Reflection Surface	Dark Reflection Surface
Inside the room (No sun) **Resistor 10K**	**1.3 – 1.4 V**	**1.5 V**	**1.3 V**
Outside (Under the sun) **Resistor 10K**	**1.6 V**	**1.7 V**	**1.4V**
Inside the room (No sun) **Resistor 10M**	**2.6 V**	**1 V**	**3.4 V**
Outside (Under the sun) **Resistor 10M**	**0 V**	**1 V**	**2-3 V**

Table 3: IR Sensor Result

4.1.3 Cytron Motor Driver MDD10A

Motor driver is used to drive the 4 tyres that will be attached onto the robot for movement purpose. This type of motor driver can drive up to 2 brushed DC motors bi-directionally. Since DAGU platform already included 4 rubber tyres which are being attached to 4 geared DC brushed motor, the motor driver will control both right tyres together and both left tyres together. The movement of the robot can be simplified using the table below. For forward movement, all 4 tyres will move forward, same goes for reverse direction. In case of turning left and right, the robot will apply differential turning method. This can be illustrated in the table below.

Clockwise	**Right Front Tyre**	**Right Back Tyre**	**Left Front Tyre**	**Left Back Tyre**
Forward	Clockwise	Clockwise	Clockwise	Clockwise
Reverse	Anti-Clockwise	Anti-Clockwise	Anti-Clockwise	Anti-Clockwise
Left Turn	Anti-Clockwise	Anti-Clockwise	Clockwise	Clockwise
Right Turn	Clockwise	Clockwise	Anti-Clockwise	Anti-Clockwise

Table 4: Tyres Movement

Features:

- Bi-directional control for 2 brushed DC motor.
- Support motor voltage ranges from 5V to 25V 30V.
- Maximum current up to 10A continuous and 30A peak (10 second) for each channel.
- Solid state components provide faster response time and eliminate the wear and tear of mechanical relay.
- Fully NMOS H-Bridge for better efficiency and no heat sink is required.
- Speed control PWM frequency up to 20KHz. Support both locked-antiphase and sign-magnitude PWM operation.
- Activation buttons for fast test on each channel.
- Dimension: 84.5mm x 62mm

Figure 12: Dual Channel 10A DC Motor Driver

4.1.4 Gripper System

Figure 13: Gripper Mechanism

For controlling the gripping mechanism, two servo motors will be used where one of them is used for open-and-close gripper while the other one will be used for tilting up and down. A servo motor has three wires - Vcc, Ground and Signal. Red wire which is usually red will be connected to 5V pin on the Arduino while black or brown wire will be connected to the GND pin. The signal cable is usually white, yellow, or orange and it shall be connected to Pulse Width Modulation (PWM) on the Arduino to obtain the signal. For generating a PWM waveform a microcontroller is often used where in this project, Arduino Mega is chosen since it has the capabilities in performing and processing the signal onto the servo motors with only few lines of coding. Based on the program shown below, the

gripper will be fully open when the digital value is set to be 150; [myservo.write(150)] and will be fully close when the digital value is set to be 0; [myservo.write(0)].

Figure 15: Gripper Test Code

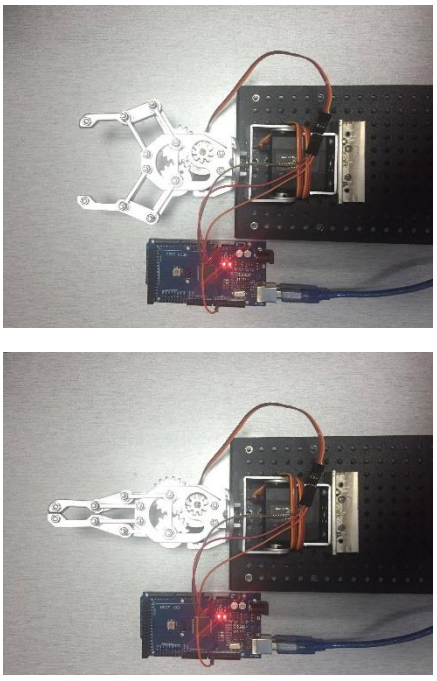


Figure 14: Fully open and close gripper

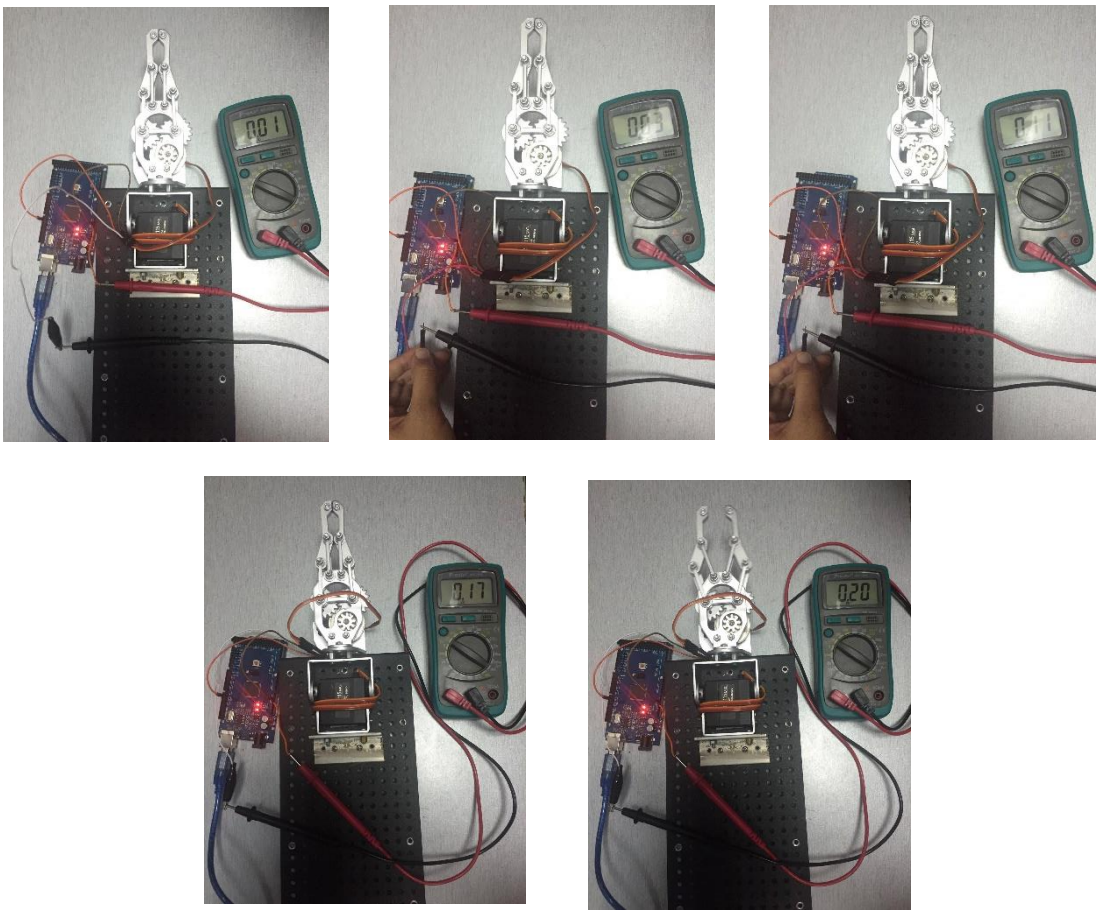


Figure 16: Current needed by Servo Motor (Gripper System)

4.1.5 Power Consumption

Basically, the load of the project is being calculated and measured. This can be obtained by looking at the load's power rating label in their respective datasheets. Formula below shows the calculation on how to obtain the wattage of a load. The wattage obtained is the power needed per hour, with base unit, Watt-Hour (Wh).

$$Power = Voltage \cdot Current$$

Table below shows the overall power consumption of each load.

Components	Voltage - Current	Power
1. IR Sensor	5V, 33mA	0.165W
2. Ultrasonic sensor	5V, 15mA	0.075W
3. Camera	2.8V, 200mA	0.56W
4. Gripper System	5V, 200mA	1.0W
5. Arduino Mega 2560	5V, 50mA	0.25W
6. Raspberry Pi	5V, 700mA	3.5W
7. Cytron Motor Driver	12V, 1A	12W

Table 5: Power Consumption of the Project Different Modules

Figure 17: Example of LiPo battery

Lipo battery is commonly used in mobile platform or robot prototype due to its small size and lightweight. For this project, one LiPo battery with the capacity of 2200mAh and 11.1V output voltage is sufficient to run the robot for at least an hour. This can be proven in the calculation below.

Total current: 2.198A (from table 5)

Battery lasting hours: $2.2\ AH\ /\ 2.198A\ \cong\ \mathbf{1\ Hour}$

4.1.6 Image Processing Testing

One of the deliverables of this project is the image processing part. The robot must be coded to perform image processing where it can detect and recognize objects while distinguishing it whether it is an obstacle, target object or collection point. By performing image processing, it is easier for the robot to do an environment mapping and successfully achieving the objective. Basically table below shows the assigned colours and shapes for certain object.

Object	Colour	Shape
Target object (victim)	Green, Yellow, Blue and Red	Cylinder rod
Obstacles	White or black	Boxes – Rectangular and Square-shaped
Collection point	White tape	-
Game field boundaries	Black tape	-

Table 6: Colours and Shapes Assigned

i) Center of contour

Finding the center of contour of certain objects or shapes is the first step in accomplishing image processing using Raspberry Pi. In order to achieve the final image processing, image pre-processing which in this case, finding the center of contour must be done. There are three steps in finding the center of contour of objects and shapes, which are:

- Converting a normal image to grayscale
- Image blurring which is to reduce the high frequency of noise
- Image binarization which includes edge detection or thresholding of an image

Figure below shows the code being executed in Python and the original image that will be used as sample for this testing.

*center_of_shape.py - F:\PMI\1-center-of-contour\center-of-contour\center_of_shape.py (2....

File Edit Format Run Options Window Help

```
# USAGE
# python center_of_shape.py --image shapes_and_colors.png
# import the necessary packages
import argparse
import imutils
import cv2

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
	help="path to the input image")
args = vars(ap.parse_args())

# load the image, convert it to grayscale, blur it slightly,
# and threshold it
image = cv2.imread(args["image"])
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
thresh = cv2.threshold(blurred, 60, 255, cv2.THRESH_BINARY)[1]

# find contours in the thresholded image
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
	cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if imutils.is_cv2() else cnts[1]

# loop over the contours
for c in cnts:
	# compute the center of the contour
	M = cv2.moments(c)
	cX = int(M["m10"] / M["m00"])
	cY = int(M["m01"] / M["m00"])

	# draw the contour and center of the shape on the image
	cv2.drawContours(image, [c], -1, (0, 255, 0), 2)
	cv2.circle(image, (cX, cY), 7, (255, 255, 255), -1)
	cv2.putText(image, "center", (cX - 20, cY - 20),
		cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)

	# show the image
	cv2.imshow("Image", image)
	cv2.waitKey(0)
```

Ln: 2 Col: 57

Figure 18: Center of Contour Code in Python

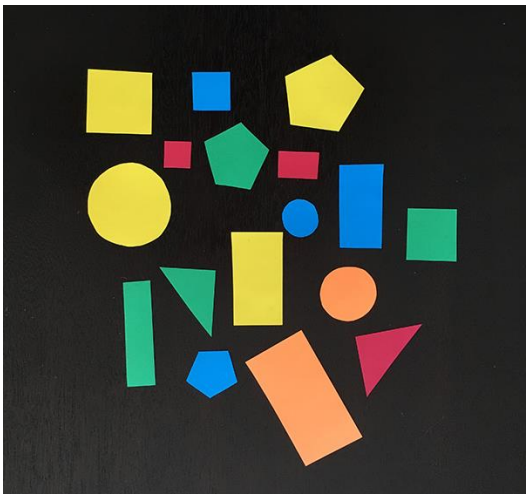


Figure 19: Original Image

The first step is to load the image from the disk and apply the grayscale conversion. For this conversion, Gaussian function will be used to blur or smoothen out the edge of pictures where it is typically used to reduce the noise that are present in the image. According to the python code shown above, the image will be grayed out, smoothen using 5x5 Gaussian blur and lastly will be thresholded. Figure below shows the thresholded image after performing such functions. The image returns a binary image where it has two colours; black in the background and shapes will appear white in colour.

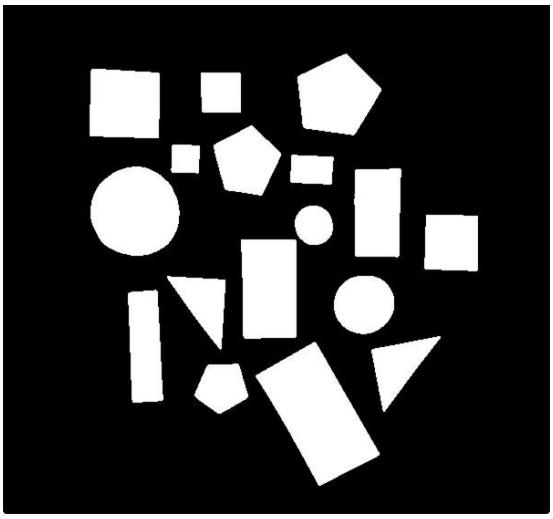


Figure 20: Threshold image

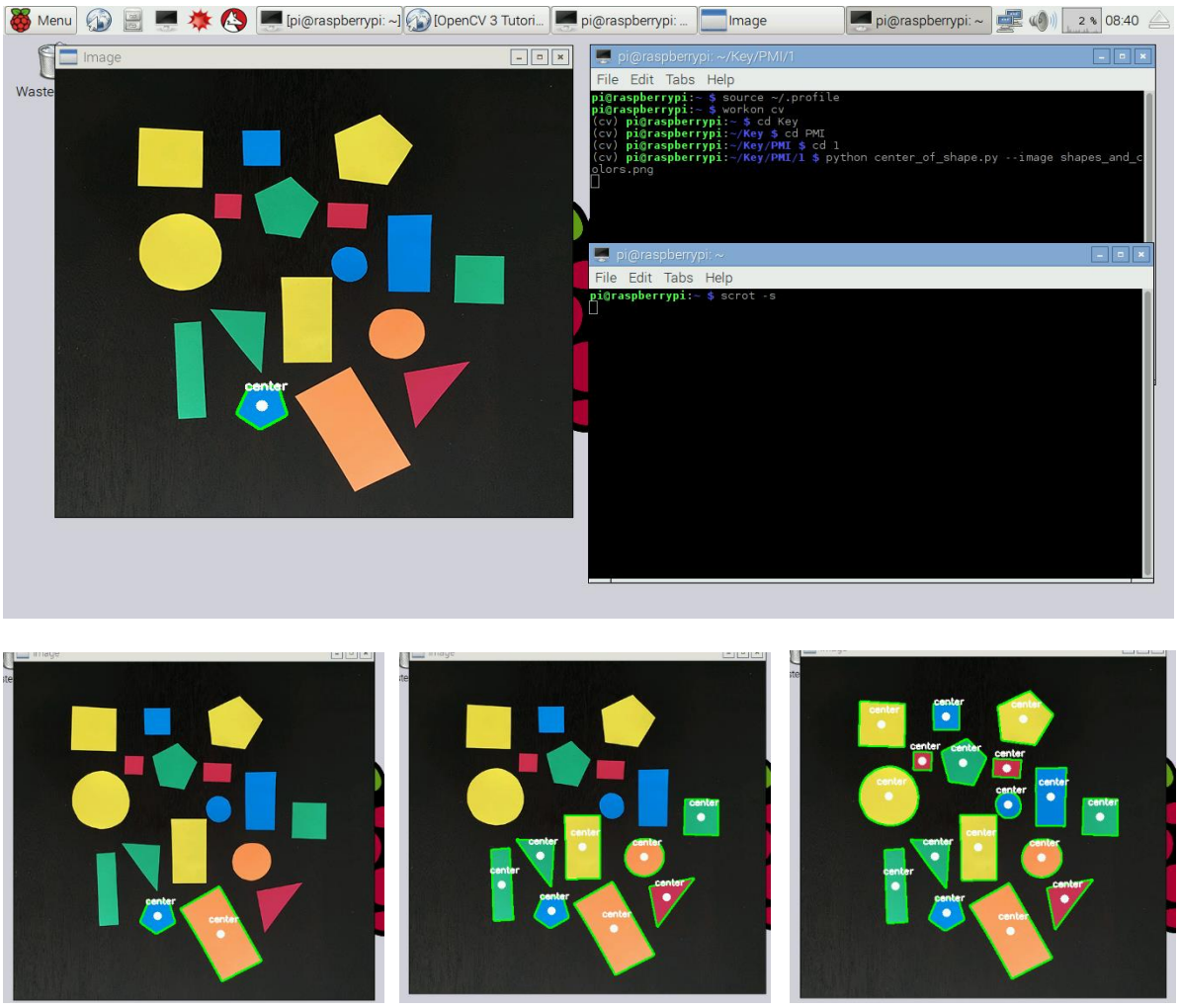


Figure 21: Step by step detection of center contour

Figure above shows the detection of center of contour is done one by one until it finishes scanning. To run the code in cv environment, the folder of the python file must be located. Execute the script using command below in the terminal.

```
(cv) pi@raspberrypi: ~/Key/PMI/1 $ python center_of_shape.py --image
shapes_and_colors.png
```

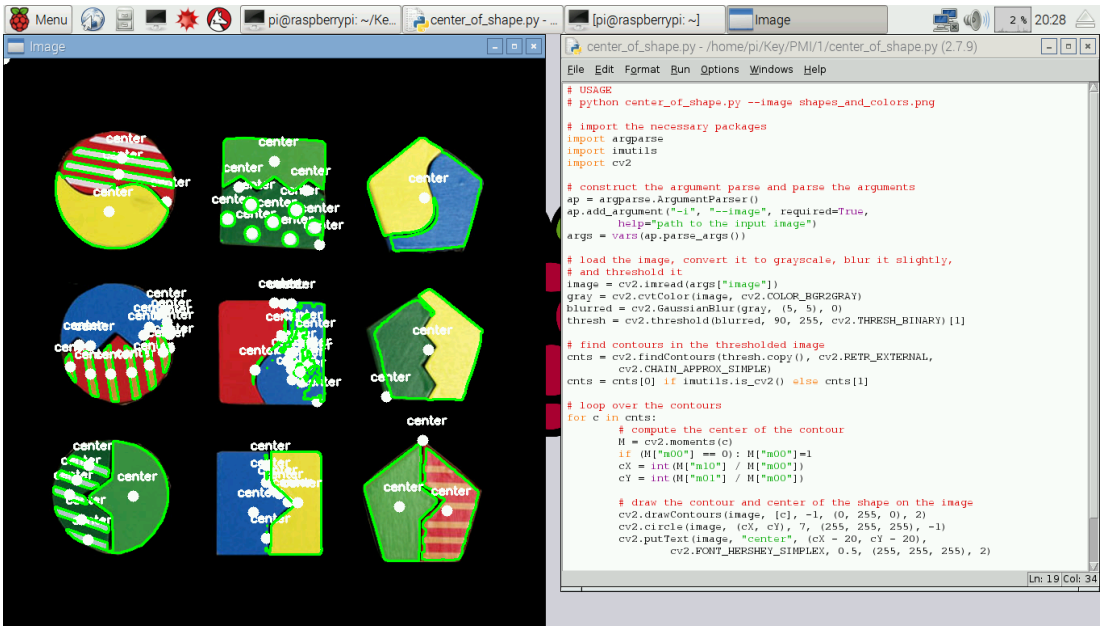


Figure 23: Threshold value = 60

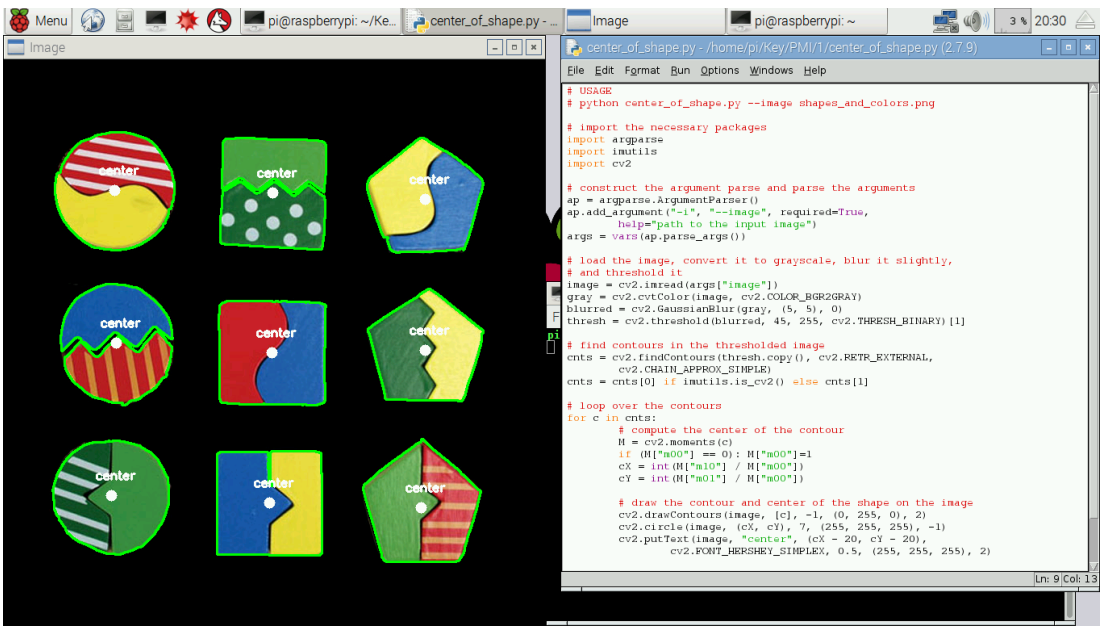


Figure 22: Threshold value = 45

Figure above shows the program with a different image. The first picture shows that the result of maintaining the threshold value at 60. Since it detected so many center in one object, the threshold has been changed to a lower value which in this case 45. The result can be observed in the second picture where the detected centers are in the middle of all objects.

ii) Shape detection

Shape detection is the second step in achieving image processing using Raspberry Pi. In order to perform shape detection, contour approximation algorithm is used. As the name implies, contour approximation is an algorithm where it reduces the number of points in a curve. It is known as Ramer-Douglas-Peucker algorithm or also known as split-and-merge algorithm. Basically, according to this algorithm, approximating a contour is predicated by the assumption of having a series of short line segments in a curve.

In order to perform contour approximation, the perimeter of a contour is computed, followed by constructing the actual contour approximation of a certain shape. Contour is actually consisting of a list of vertices, where basically, as for example, if the contour has three vertices, it is automatically being detected as a triangle. That is when the case is very solid. If a contour is having four vertices, it is either a square or a rectangle. To distinguish between one another, the aspect ratio of the shape is computed. If the aspect ratio is ~1.0, the shape is definitely a square. Otherwise, the shape is rectangle. Figures below show two different sample of shape detection.

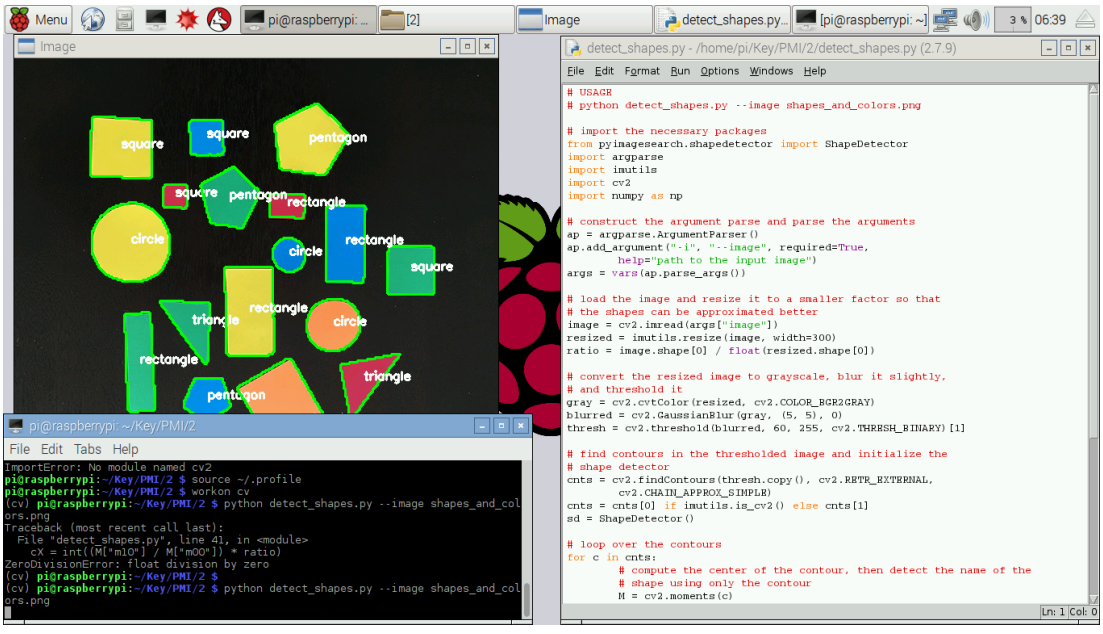


Figure 24: Detecting Shapes 1

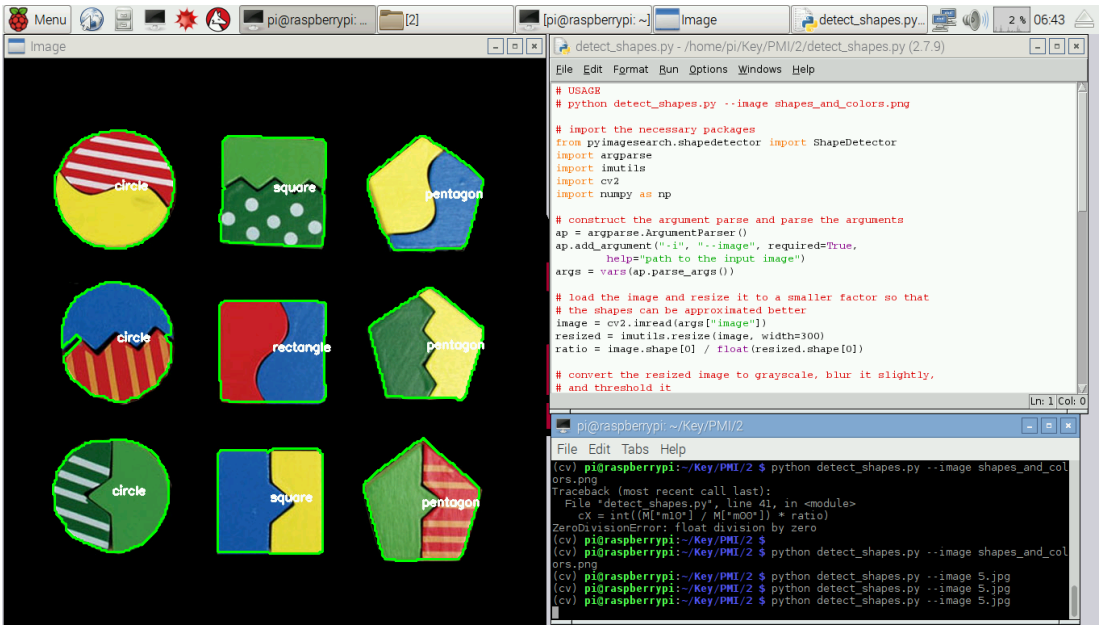


Figure 25: Detecting Shapes 2

iii) Colour detection

The third step of image processing part is the colour detection. It is important as the colour will differentiate between the obstacles and objects to be rescued. An image can be characterized by basic colour channel statistics and also colour histograms. Since the statistics can be computed, but still it cannot give us the actual colour label such as “red”, “green”, “blue” and so on.

For this project, L*a*b colour space is leveraged along with the Euclidean distance which is to tag, to label and also to determine the object colour using OpenCV and Python. L*a*b colour space is a space with dimension of L (lightness) and ‘a’ and ‘b’ for the colour-opponent dimension. The reason L*a*b colour space is used to be leveraged with RGB colour space is because L*a*b colour space is designed to be approximately similar to human vision. In order to label and tag each of the shapes in an image which has its individual colours, the Euclidean distance is calculated.

Basically, these RGB web colours have been assigned to each colour that is going to be used in this project. At first, colour dictionary is being initialized in the program so that the obtained imaged can be processed and provide with correct colour label. The colour dictionary that has been initialized is as follows.

```
colors = OrderedDict({
"red": (255, 0, 0),
"green": (0, 255, 0),
"blue": (0, 0, 255),
"yellow": (255, 255, 0),
```

```
"orange": (255, 175, 0),})
```

One of the primary drawbacks in using this method is the lighting conditions of an image. Since each image has various hues and saturations, the colours that will be detected will hardly look like pure colours – red, green, blue and etc. If the image has small sets of colours, this method can be implemented but in case of a larger colour palette, this method will most likely to provide an incorrect result where it is dependent on the complexity of the images. Figure below shows the shapes and the colours are being detected using this method.

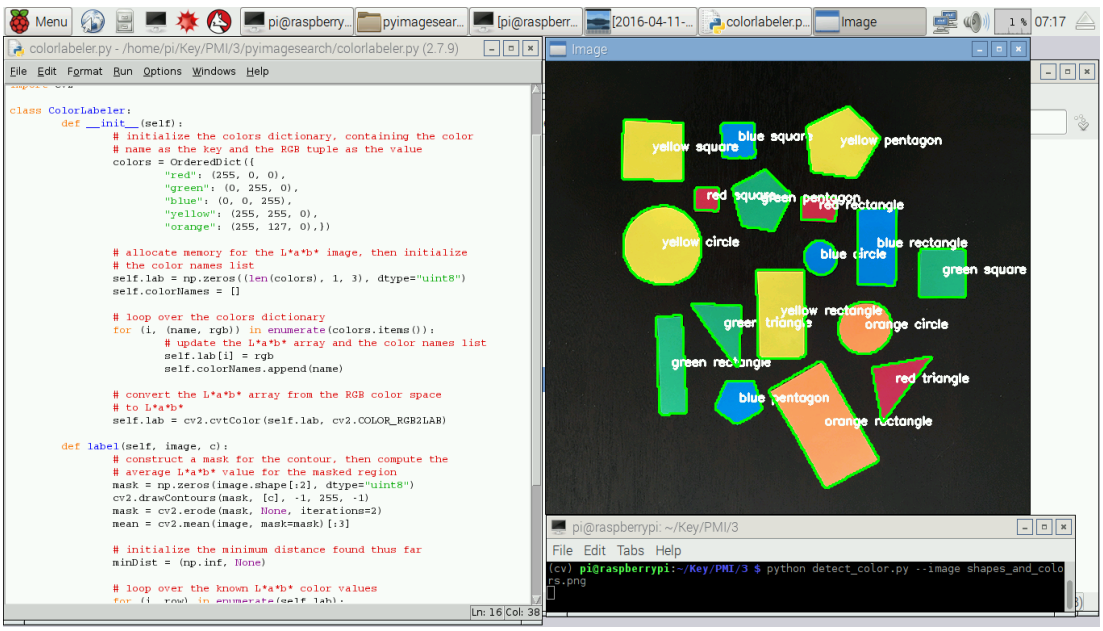


Figure 26: Colour detection

CHAPTER 5
CONCLUSIONS AND RECOMMENDATION

This paper provides the overall details and discussion on the project entitled "Enhancing the Capabilities of Search and Rescue Robot (BamBot)". As discussed earlier, the selection of components is very crucial in this type or project. Wrongly chosen components will give negative effects in delivering all the objectives successfully. The concept for this project is to design a robot that can find objects, identify them and transport them to desired locations. The concepts suggested beforehand will be tested and improvised in order to ensure that the system design meet the expectation.

In this design project, the main objective is to enhance a search and rescue robot that can find objects, identify them and carry all the objects to a desired location. The main controller which is the Raspberry Pi will provide a vision system and connect to the slave controller. The slave controller which is Arduino is used to read the sensors feedback status, update the motor speed and direction and implement the obstacle avoidance algorithm and the maneuvering algorithm where the object will be safely transported by the robot. As mentioned in the tools section, the sensors that will be used in this project are ultrasonic sensors which aims to provide feedback on any unwanted obstacle along the robot's path. A camera which will assist in image processing also will be used to generate a map where it will show the current location of the robot, objects and obstacles together with planning on the preferred path. So basically, with the help of all sensors being integrated together, the robot will automatically know about the environment of the area and also detect any obstacles and objects in the specified area of mission.

REFERENCES

[1] Abdramane, I., "Robot In A Reconfigurable Maze," Undergraduate, Electrical & Electronic Engineering Department, Universiti Teknologi PETRONAS unpuplished, 2013.

[2] Abiyev, R., Ibrahim, D., and Erin, B., "Navigation of mobile robots in the presence of obstacles," *Advances in Engineering Software,* vol. 41, pp. 1179-1186, 2010.

[3] Al-Taharwa, I., Sheta, A., and Al-Weshah, M., "A mobile robot path planning using genetic algorithm in static environment", Journal of Computer Science, pp. 341–344, 2008.

[4] Bekey, G.A., (2005). "Autonomous Robots: From Biological Inspiration to Implementation and Control", England: The MIT Press, Cambridge, Massachusettes.

[5] Buniyamin N., Wan Ngah W.A.J., Sariff N., and Mohamad Z., " A Simple Local Path Planning Algorithm for Autonomous Mobile Robots," International Journal Of Systems Applications, Engineering & Development, Issue 2, Volume 5, 2011.

[6] Canny, J., (1986) *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence.

[7] Engelberger, J. F., (1980). "Robotics in Practice: Management and Applications of Industrial Robots", Amersham: Avebury Publishing Company.

[8] Gonzalez, R.C., Woods, R.E., (2008). *Digital Image Processing*. Prentice Hall. pp. 1–3.

[9] Kirillov, A., "AForge .NET framework," 2009. [Online]. Available: http://www.aforgenet.com. [Accessed 24 December 2015]

[10] Robot, Oxford's Dictionary, n.d. [Online]. Available: http://www.oxforddictionaries.com/definition/english/robot. [Accessed 27 October 2015].

[11] Patel, A., "Pathfinding" in Pathfinding vol. 2014, A. s. A. P. F. R. B. Games, Ed., ed. http://www-cs-students.stanford.edu/~amitp/: Computer Science Department, Stanford University, 2015.

[12] Sam, H., Nathaniel, B., Andrew D., and Nikolaos, P., "A Search and Rescue Robot", Department of Computer and Science and Engineering, University of Minnesota, May 2009.

[13] "Search and Rescue Robots: What is it all about? ", Robotics: Jacobs University, n.d. [Online]. Available: http://robotics.jacobs-university.de/node/236. [Accessed 27 October 2015].

[14] Shilpashree, K.S., Lokesha.H , Hadimani Shivkumar, "Implementation of Image Processing on Raspberry Pi", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015.

[15] Woo-Jin, S., Seung-Ho, O., Jin-Ho, A., Sungho, K., and Byungin, M., "An Efficient Hardware Architecture of the A-star Algorithm for the Shortest Path Search Engine", in INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on 2009, pp. 1499-1502.

[16] "What is a Raspberry Pi?", Raspberry Pi Foundation, n.d. [Online]. Available: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/. [Accessed 27 October 2015].

[17] "10 Things We Couldn't Do Without Robots", Web Design School Guide, 2015. [Online]. Available: http://www.webdesignschoolsguide.com/library/10-things-we-couldnt-do-without-robots.html. [Accessed 27 October 2015].