

**SMART SIGN DETECTION ROBOT**

**CHIA CHEE HAW**

**Submitted to the Faculty of Electrical and Electronic Engineering  
in partial fulfillment of the requirement for the degree of  
Bachelor in Electrical Engineering (Electronic)**

**Faculty of Electrical and Electronic Engineering  
University Malaysia Pahang**

**NOVEMBER 2010**

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : \_\_\_\_\_

Author : CHIA CHEE HAW

Date : 21 OCTOBER 2010

Dedicated, in thankful appreciation for all the support, encouragement and understanding to my family and friends.

## ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people. First and foremost, I wish to express my sincere appreciation to my supervisor, En Reza Ezuan Bin Samin, for guidance, advices, motivation and enthusiasm given throughout the progress of this project. Without the continued support and interest from my supervisor, this thesis would not have been the same as presented here. Besides, I would like to express my sincere appreciation for his valuable advices, guidance and encouragement. This has inspired me to be more confident in trying new things.

My appreciation also goes to my family who has been so tolerant and supports me all these years. Thanks for their encouragement, love and emotional supports that they had given to me.

Nevertheless, my sincere appreciation also extends to all my course-mates and other whom have provided assistance directly or indirectly in helping me to complete this project. Their view and tips are useful indeed. There is no such meaningful word than.....Thank You.

## ABSTRAK

Robot digunakan secara meluas di kilang-kilang dalam bidang pembuatan serta menggantikan kerja manusia dalam persekitaran yang berbahaya. Idea projek ini adalah untuk memecahkan sensor tradisional dengan menggunakan lebih maju dan sensor cerdas disebut sensor penglihatan. Mobile robot pembangunan navigation adalah salah satu aplikasi yang paling kompleks dalam bidang robot. Dalam projek ini, program akan mengembangkan untuk mengawal pergerakan robot berdasarkan gambar yang diperolehi. The CMUcam mempunyai mikrokontroler sendiri, SX28 mikrokontroler untuk menangani maklumat data yang diperolehi oleh kamera. Sebuah program akan dibina menggunakan BASIC Stamp dengan bahasa pengaturcaraan PBASIC. Program PBASIC ini akan membina satu rangkaian secara bersiri di antara robot dengan CMUcam. Ini akan memerintahkan kamera untuk mengunci dan menjejaki pada tanda panah. Stamp kemudian akan member arahan kepada BoeBot untuk bergerak berdasarkan data anak panah yang ditunjuk pada kamera. CMUcam adalah untuk bertindak sebagai mata iklan yang dapat mengembalikan output maklumat data seperti warna dan kedudukan objek. Sebagai kesimpulan, projek ini menggunakan Boe-bot terintegrasi dengan CMUcam untuk mengembangkan program automasi.

## ABSTRACT

Robots are now widely used in factories area for manufacturing purposes and replacing human work in dangerous environment. The idea on this project is to solve the traditional sensor by using more advance and intelligent sensor called vision sensor. Mobile robot navigation development is the one of most complex application in robotic field. In this project, a program will develop to control the movement of the robot based on the images acquired. The CMU cam has its own microcontroller, SX28 microcontroller to handling the data information acquired by the camera. A program will be built up using BASIC Stamp with PBASIC programming language. The PBASIC programs software will set up a serial interface connection between the robot and the SX28 microcontroller. This will instruct the camera to lock on and track on arrow sign. The stamp will command the Boe-Bot to move according to the arrow direction based on the imaging data send by the camera. The used of CMU cam is to acts as an eye ad which can return output data information such as color and the position of the object. As a conclusion, this project uses the Boe-bot integrated with CMU cam to develop an automation sensing program.

**TABLE OF CONTENT**

<b>CHAPTER</b>	<b>TITLE</b>	<b>PAGE</b>
	<b>DECLARATION OF THESIS</b>	ii
	<b>DEDICATION</b>	iii
	<b>ACKNOWLEDGEMENT</b>	iv
	<b>ABSTRACT</b>	v
	<b>ABSTRAK</b>	vi
	<b>TABLE OF CONTENT</b>	vii
	<b>LIST OF TABLES</b>	x
	<b>LIST OF FIGURES</b>	xi
	<b>LIST OF SYMBOLS</b>	xiii
	<b>LIST OF APPENDICES</b>	xv

<b>1</b>	<b>INTRODUCTION</b>	
1.1	Background	1
1.1.1	Introduction to the project	2
1.1.2	Problem statement	3
1.2	Objective of Project	3
1.3	Scope of Project	3
1.4	Thesis Outline	4
<b>2</b>	<b>LITERATURE REVIEW</b>	
2.1	Introduction	5
2.1.1	Phoenix & Dragon: Parallax BoeBot / Lego Techniques Hybrid Solutions to the Reactive Maze Traversal and Deliberative “Cheese” Location Problems	6
2.1.2	Vision-Based Navigation of Mobile Robot with Obstacle Avoidance by single Camera Vision and Ultrasonic Sensing	6
2.1.3	A Low Cost Embedded Color Vision System	7
2.1.4	Dynamic Omni-directional vision for mobile robots	7
2.1.5	Road Sign Detection and Recognition	8
2.1.6	Road sign detection and recognition using matching pursuit method	8
2.1.7	Road Sign Classification without Colour Information	9
2.2	Project Review and Methodology	9



<b>3</b>	<b>METHODOLOGY</b>	
3.1	Introduction	10
3.2	Robot Description	11
3.2.1	Board of Education Robot (BoeBot)	11
3.2.2	Board of Education Carrier Board	13
3.2.3	BASIC Stamp 2	14
3.2.4	CMU cam	16
3.2.4.1	Board Layout of CMU camera	18
3.2.4.2	CMU camera App Module	20
<b>4</b>	<b>SOFTWARE IMPLEMENTATION</b>	
4.1	Introduction Software Implementation Process	22
4.2	Flow chart of project	23
4.2.1	Initialization	24
4.2.2	Setting window and sign lock-on	26
4.2.3	Perform sign tracking	28
4.2.4	Robot Movement based on the arrow direction	29
4.3	Result	32
4.4	Discussion	33
4.4.1	Choosing Object to Track	34
4.4.2	Camera Adjustment	34
4.4.3	Power Ups the Robot	35
4.4.4	Locking on Sign	35
4.4.5	Tracking Sign	35

<b>5</b>	<b>CONCLUSION AND RECOMMENDATION</b>	
	5.1 Introduction	36
	5.2 Conclusion	37
	5.3 Future development	37
	5.4 Costing and Commercialization	38
	<b>REFERENCE</b>	39
	<b>APPENDICES</b>	41

**LIST OF TABLES**

<b>TABLE</b>	<b>TITLE</b>	<b>PAGE</b>
3.1	Sample CMU camera commands	18
5.1	Robot Costing	38

**LIST OF FIGURES**

<b>FIGURE</b>	<b>TITLE</b>	<b>PAGE</b>
2.1	Flow chart of entire project task	10
3.1	Board of Education robot ( BoeBot )	13
3.2	Board of Education Carrier Board	14
3.3	BASIC Stamp 2	15
3.4	CMU Camera	17
3.5	Camera bus and its pin description	19
3.6	CMU camera App Module	20
3.7	AppMod Header	21
4.1	Flow chart of whole performing detection and robot movement	24
4.2	Flow chart of initialization	25
4.3	Flow chart of setting window and object lock on.	27
4.4	Flow chart of performing sign tracking and robot movement	30
4.5	Flow chart of full process	31

**LIST OF APPENDIX**

<b>APPENDIX</b>	<b>TITLE</b>	<b>PAGE</b>
A	Robot Pictures	41
B	Code Listing	43

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 Background**

Vision is a powerful robot sensing capability. Vision is not used in an exploratory sense but is use to confirm measure or refine existing data. Hence, sensors and hardware associate with the robot vision are more complex than other types of sensors. With this capability, we are able to perceive, identify and recognize objects.

Robot Vision is use to extract or interpret information from images of a three dimensional worlds where it provide robot with the capability of sensing which present in human vision. This computer vision process may be subdivided into six principle areas which are sensing, preprocessing, segmentation, description, recognition and

interpretation. These operations will process image data and matching the image features against stored prototypes.

In robotic world, mobile robot can be used at home as a pet or toy even home equipment like vacuum. The ability of moving from a place to another makes the robot be able to complete task like searching over victims that buried under destroyed buildings, or defusing bombs. Once a vision system has been programmed, it will perform particular inspection task and return result as through any other sensor.

### **1.1.1 Introduction to the project**

BoeBot was also known as Board of Education robot, introduced by Parallax Inc. It is a complete reprogrammable robot kit. It used software simulator and digital trainer board in conjunction to bridge gap between hardware and software. In this project, CMU cam is applied to the board to enhance the capability of vision sensing. CMU camera is a device that specially modified by Parallax to seamlessly integrated with Boe-Bot platform.

In order to develop a base prototype, Basic Stamp Editor is used to as it is the software simulator for the BoeBot. The programs build-up will be stored in BoeBot's brain, the BASIC Stamp 2 microcontroller. Once the program is transferred to the microcontroller, the mobile robot will perform autonomous tasks based on the program. This project development includes detection of arrows using CMU camera through shape recognition and mobile robot moves according to the program developed.

### **1.1.2 Problem Statement**

Human errors are seems to be the main cause of traffic accident happen. By implement advance technology to the vehicle, we are able to reduce accident happening. Vision system is more intelligent rather than using traditional sensors. Thus, sign detection system is applied to vehicle to reduce human cause errors.

## **1.2 Objective**

The aim of this project is to move a vision system for mobile robot navigation. The main objective of this project is to develop a program of vision camera robot for detect shape.

## **1.3 Scopes of Project**

This research will be the implementation of a software tracking system to induce the mobile robot to track a desired shape. The robot will track and lock on an arrow which it will then move based on the arrow direction. As if the robot detects another sign, it will change direction and move according to the latest sign detected. The robot will stop at a specific distance after the last detection.



The scope of this project is:

- i. Study on the Structure and operation of Boe-Bot.
- ii. Construct a shape tracking and moving system using Basic Stamp Editor.
- iii. Build up an Image recognizer module development for sign detection.

#### **1.4 Thesis Outline**

This chapter explains the introduction of robot vision and the principle area involved. Vision may perform several particular inspection tasks which leads vision system were group into several types.

Chapter 2 discuss on literature review of related studies. It gives brief review on vision system and its application on images processing. Besides, it also reviews on information about tracking process. This chapter also focuses on methodology for entire task.

Chapter 3 discuss about the background information of the mobile robot, Boe-Bot and CMUcam. It discuss brief on Board of Education Robot description, BOE Carrier Board, BASIC Stamp function, and operation, layout and information of CMUcam.

Chapter 4 explains the software implementation of this project. It gives details review about the process of sign detection task which include initialization, window setting, sign tracking, and performing robot movement. This chapter also contains the result and discussion.

Chapter 5 discuss on the conclusion of development of this project. It also focuses on the recommendation for this project and future development or modification for this project.

## **CHAPTER 2**

### **LITERATURE REVIEWS**

#### **2.1 Introduction Literature Review**

This chapter focuses on the literature review of the related subject. This chapter consists of summary of related study materials and component required in this research. It also gives brief reviews on vision system and its application in images processing.

### **2.1.1 Phoenix & Dragon: Parallax BoeBot / Lego Techniques Hybrid Solutions to the Reactive Maze Traversal and Deliberative “Cheese” Location Problems**

This paper involved building an autonomous robots using the Parallax BoeBot kit as a base and making any structural or electronic changes required to complete the task. The first task, reactive maze traversal, was completed utilizing two versions of BoeBot Phoenix, one of which used touch sensors as the sole receptor of information, with the final version utilizing binary IR sensors in lieu of side touch sensors, but still employing a central touch sensor to detect frontal objects. By incorporating minor directional changes to move around obstacles, the BoeBot was able to move back to the vertical coordinate of the goal state [2].

### **2.1.2 Vision-Based Navigation of Mobile Robot with Obstacle Avoidance by single Camera Vision and Ultrasonic Sensing**

This paper describes a vision-based navigation method in an indoor environment for an autonomous mobile robot which can avoid obstacles. In this method, the self-localization of the robot is done with a model-based vision system, and a non-stop navigation is realized by a retroactive position correction system. Stationary obstacles are avoided with single-camera vision and moving obstacles are detected with ultrasonic sensors. This research reports experiment in a hallway using the YAMABICO robot [3].

### **2.1.3 A Low Cost Embedded Color Vision System**

This paper describes a functioning low cost embedded vision system which can perform basic colour blob tracking at 16.7 frames per second. This system utilizes a low cost CMOS colour camera module and all image data is processed by a high speed, low-cost microcontroller. This eliminates the need for a separate frame grabber and high speed host computer typically found in traditional vision systems. The resulting embedded system makes it possible to utilize simple colour vision algorithms in applications like small mobile robotics where a traditional vision system would not be practical [4].

### **2.1.4 Dynamic Omni-directional vision for mobile robots**

This paper describes a new algorithm for Omni-directional vision navigation. A prototype Omni-directional vision system and the implementation of the navigation techniques using this modern sensor and an advanced automatic image processor are described. The significance of this work is in the development of a new and novel approach—dynamic Omni-directional vision for mobile robots and autonomous guided vehicles [5].

### **2.1.5 Road Sign Detection and Recognition**

In this paper, road signs are detected by means of rules that restrict colour and shape and require signs to appear only in limited regions in an image. They are then recognized using a template matching method and tracked through a sequence of images. The method is fast and can easily be modified to include new classes of signs [6].

### **2.1.6 Road sign detection and recognition using matching pursuit method**

This paper describes an automatic road sign recognition system by using matching pursuit (MP) filters. The system consists of two phases. In the detection phase, it finds the relative position of road sign in the original distant image by using a priori knowledge, shape and colour information and captures a closer view image. Then it extracts the road sign image from the closer view image by using conventional template matching. The recognition phase consists of two processes: training and testing. The training process finds a set of best MP filter bases for each road sign. The testing process projects the input unknown road sign to different set of MP filter bases (corresponding to different road signs) to find the best match [7].

### **2.1.7 Road Sign Classification without Colour Information**

Colour may be used as a useful attribute for the decomposition of classification problem into several priori defined road sign groups or sub-problems. In this paper, the colourless method for the road sign classification is presented working on gray-level images and allowing the same problem decomposition as its colour-based counterpart. The method may be used in combination with the colour-independent sign detection algorithms. The road sign recognition system then works entirely without the colour which may be used as an alternative procedure when the input traffic scene images lacks good colour information [8].

## **2.2 Project Review and Methodology**

In this project, the major task is to study the robot navigation and study on the control to the navigation of the robot using Basic Stamp. The studies of PBasic software and programming languages from website (<http://my.fit.edu/~jtrifiletti/BasicStamp> ), Basic Stamp 2 chip architecture from Basic Stamp User Manual, Parallax Inc. website (<http://www.parallaxinc.com>), and Carnegie Mellon University website (<http://www.cs.cmu.edu> ). The development of software program for the mobile robot movement become handy by knowing the relationship between Boe-Bot, CMUcam and Basic Stamp from their documentation.

Second task is to study about the basic knowledge of CMUcam that connected to Board of Education for detection purpose. The third would be the integration the robot with program developed. In this project, a program was develop that make the mobile robot to detect sign and it will response by moving based on the direction of the arrow pointed. From given documentation, certain syntax of Basic Stamp 2 is used to program the mobile to grab the first colour that appears in front of the camera. Then the CMU

cam will grab all the data information and sent to Basic Stamp for memorize as a data base. As the colour of the sign arrow is memorized, the robot then will react only when it detect the arrow with the colour only. Although there are multicolour object to be appear around the mobile robot, but the robot only response to the first colour that appear in front of the camera when its operates. The implementation of the entire task is shown in Figure 2.1.

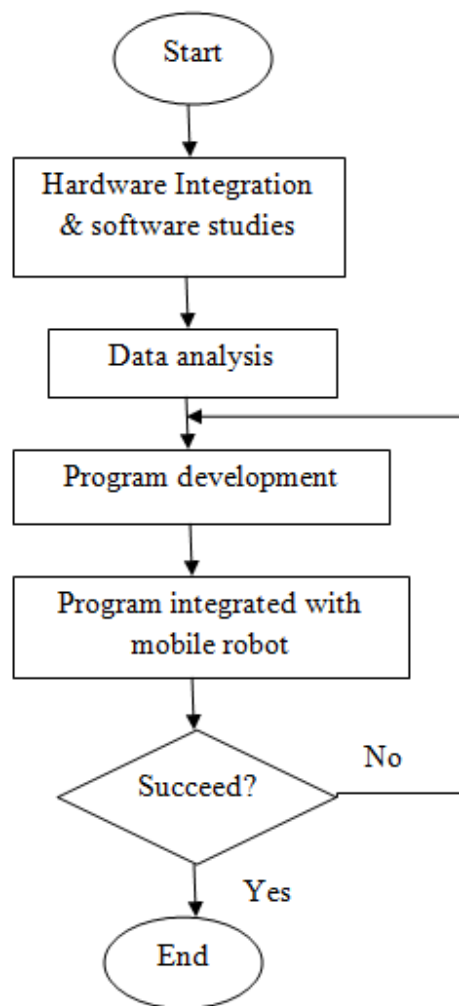


Figure 2.1: Flow chart of entire project task

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Introduction**

The development program of a vision system is one of the complex applications in robotic field. In this research, the program developed is to control the movement and navigation of the robot based on the images acquired. The robot kit used, Boe-Bot is embedded with an imaging system which consist of CMU camera to detect image correctly. Once the vision system is applied to the robot, it will induce to follow the desired target shape. When the CMU camera capture images, the information will be sent to the stamp and will be compared to the based prototype. The robot will respond by move according to the direction pointed once the image captured is 'appropriate' with the based data.



## **3.2 Robot Description**

This project uses Board of Education Robot (Boe-bot) integrate with CMU cam to perform an automation task. The advantages of Boebot is it can integrated with many devices and it can implement more than one extra circuit to the board itself such as whisker, infrared devices, and etc. In this robot description, this chapter will be reviews on BoeBot, BOE carrier board, Basic Stamp and CMU cam documentation.

### **3.2.1 Board of Education Robot ( Boebot )**

Board of Education is built on a high quality brushed aluminum chassis that provides a study platform for the servo motors and printed circuit board. The rear wheel is a drilled polyethylene ball held in place with a cotter pin. This parallax Boe-Bot kits sets include breadboard, battery pack, two servos and wheels along with others small components. Boe-bot is structured with PBASIC 2.5 source code support.

This mobile robot is basically a rolling BASIC Stamp on a carrier board (Board of Education). All I/O projects are built on the breadboard, without a need for soldering, the circuit board is mounted on an aluminium chassis that provides a sturdy platform for the electronics, and the servo motors. Mounting holes and slots on the chassis may be used and benefit to add some custom robotic equipment.

The unique of Boe-Bot is the BASIC Stamp 2 microcontroller brain where BASIC Stamp 2 is both powerful and easy to used, especially with robot. By writing the program using BASIC programming language, the software is loaded into the stamp and used to control the Boe-Bot. This will set up the serial interface connection between the robot and the CMU cam's microcontroller. Boe-Bot can be programmed to perform many functions. Once the assembly is made, testing must be done to ensure all components work properly. The Boe-bot includes two servos or wheel motors which will be used to control the path of the robot. An electric engine

also known as a pulse train will control the servos. The software in the BASIC stamp will control how much each wheel should turn in order to advance the robot towards the object that the camera is tracking.

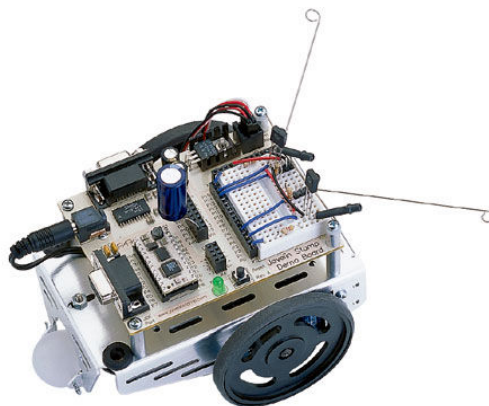


Figure 3.1 : Board of Education robot ( BoeBot )

### 3.2.2 Board of Education Carrier Board

Board of Education carrier board is a breadboard consists of BASIC Stamp2 and Board of Education. It includes a handy power switch and servo jumper making it surpass the features of previous Boe-Bot versions. The jumper near the servo connectors will select either Vdd (5V) OR Vin ( unregulated input voltage ) to power the servos [9].

The three-position power switch is either off (0), power to everything but servo connectors (1), or power to everything (2). The Board of Education carrier board in has mechanically interlocked power supply to prevent dual connection of wall-pack and 9 volts battery [9]. The DB9 connector is for BS2-IC programming and serial communication during run-time. Besides, it has the P0-P15 I/O pins, Vdd, and Vss [9]. The female 10- pin dual row connector for optional AppMods means for more breadboard space.

The power provided from battery source or any other source to servo sockets X4 and X5 can be connected by using a jumper that is located between these two sockets, the default position is Vdd (5++ regulated). In order to gain extra power to the servo, this jumper can be move to Vin position. Oversupplied or exceed the servo specification may damaged to the servo.

The reset button can be used to restart the BASIC Stamp IC without having to cycle the power. This saves wear-and-tear on the power switch for simple program restarts.

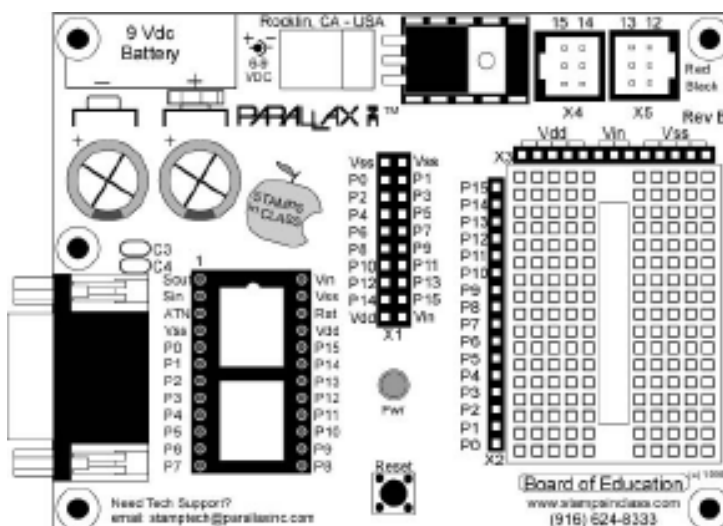


Figure 3.2: Board of Education Carrier Board

### 3.2.3 BASIC Stamp 2

BASIC stamp serve as the microcontroller for the project. The stamp provides I/O pins, which can be fully programmed to interface with other components such as LED speakers and other devices. The stamp will include a 5 volts regulator, serial EEPROM and PBASIC interpreter.

BASIC Stamp 2 is a microprocessor which has a PIC16C57 chip with a PBASIC interpreter running at 20MHz. It sports a 2048 byte EEPROM for about 500 instructions and performing 4000 instruction per seconds. The stamp 2 is very versatile and popular microprocessor building block that can be used to develop many different electronic sensing and control system. In addition, stamp can be interface with many different types of sensors or external device quit.

The BASIC program will be written using a BASIC Stamp editor supplied y the parallax vendor and can be transferred directly from the PC to the EEPROM through the serial connection. User can erase or rewrite the program by reloading new program into EEPROM. The PBASIC interpreter fetches and writes the instructions to the EEPROM. Since the EEPROM is non-volatile, it stores the program even when power off.

The programming language used by the BASIC Stamp is the BASIC programming language. The software loaded into the stamp used to control the Boe-Bot. The Stamp is plugged into the Boe-bot. The software set up the serial interface connection between the robot and the CMU camera's own SX28 microcontroller. The software instructed the robot to track a shape. The stamp command the Boe-Bot to follow the sign being detected based on imaging data sent from the CMU camera and stop when it comes within a specified distance after last sign [1].

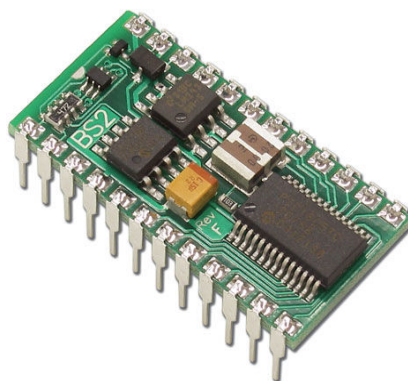


Figure 3.3: the BASIC Stamp 2

### 3.2.4 CMU Camera

The CMU camera system was originally designed by Carnegie Mellon University and is specially modified by Parallax to seamlessly integrate with Boe-bot platform. The connection method to the robot only requires a couple of screws and plugging in the serial components so that the stamp onboard the robot can interpret the data sent to it. The CMU camera already contains the hardware necessary for image tracking, pattern locking onto a specific target, and the interface necessary for control the wheel servos for movement. With a few modifications, the CMU camera may be able to track specified shape rather than having to use the pattern lock feature that is already implemented in the hardware.

The CMU camera is a SX 28 microcontroller interfaced with an OV 6620 Omni-vision CMOS camera on a chip that allows simple high level data to be extracted from the camera's streaming video [1]. CMU camera is a new low-cost, low-power sensor for mobile robots. CMU camera vision system can perform many different kinds of on-board, real-time vision processing. Because CMU camera uses a serial port, it can be directly interfaced to low-power processors. Using CMU camera, it is easy to make a robot head that swivels around to track an object.

The camera can be used to output information such as the size or position of object. The information will be sent to the stamp, which will use the image coordinates to control the robot and advance to the object, the stamp will send instruction commands to the SX28. Each command will be sent using visible ASCII characters. The commands used to communicate with the CMUcam. The will be integrated into the BASIC program downloaded onto the stamp.

At 17 frames per second, CMU camera can do the following:

1. track the position and size of a colorful or bright object
2. measure the RGB or YUV statistics of an image region
3. automatically acquire and track the first object it sees
4. physically track using a directly connected servo
5. dump a complete image over the serial port
6. dump a bitmap showing the shape of the tracked object

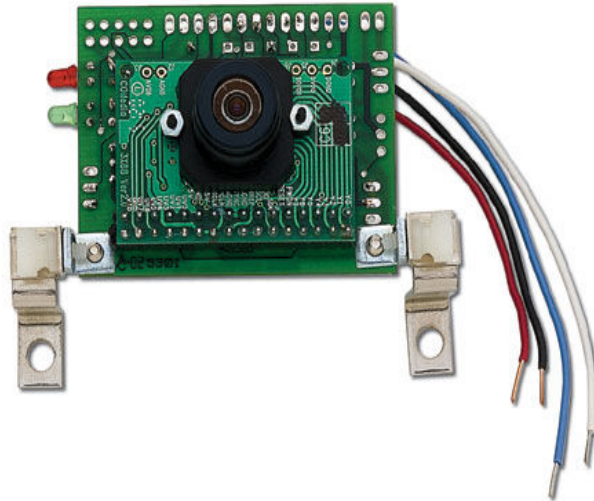


Figure 3.4: CMU Camera

Table 3.1: Sample CMU camera commands

<b>Command</b>	<b>Description</b>
RS\r	This command reset the vision robot
TW\r	This command will track the object found in the central region of the current window. This can be useful for locking onto and tracking object held in front of the camera.
DF\r	This command will dump a frame out the microcontroller via serial port. This dumped data will consist of raw video data column by column and also consisting of a frame synchronize byte and column synchronize byte.
L1 value\r	This command is used to control the green LED tracking light. The three possible values are 0, 1, and 2. 2 is default auto mode. 0 is disables the tracking light and 1 turns the tracking light on.
DM value\r	This command sets the delay before packets that are transmitted over the serial port. The possible delay value range from 0-255.

#### 3.2.4.1 Board Layout of CMU camera

The camera consists of two major components, the CMU camera board and the CMOS camera module. The module must be attached with CMU camera all the times for the system to function correctly. CMU camera board has a level shifted serial port which provides full shifting for communication with the computer. Though it only uses 3 of the 10 pins it is packaged in a 2x5 pin configuration to fit standard 9 pin ribbon cable clip-on serial sockets and 10 pin female clips on serial headers that can both attach to a 10 wire ribbon cable.

CMU camera board also has a TTL serial port where it taps into serial I/O before it goes through MAX 232 chip. This port is ideal for communication with a microcontroller that does not have any built in level shifting. In this project, the CMU camera board does not have a MAX 232 because it communicates with microcontroller [1].

In this camera module, it has 32 pins of camera buses. This bus interfaces with the CMOS camera chip. The CMOS camera board is mounted parallel to the processing part of the board and connects starting at pin 1. Figure 3.5 shows the description of the pins.

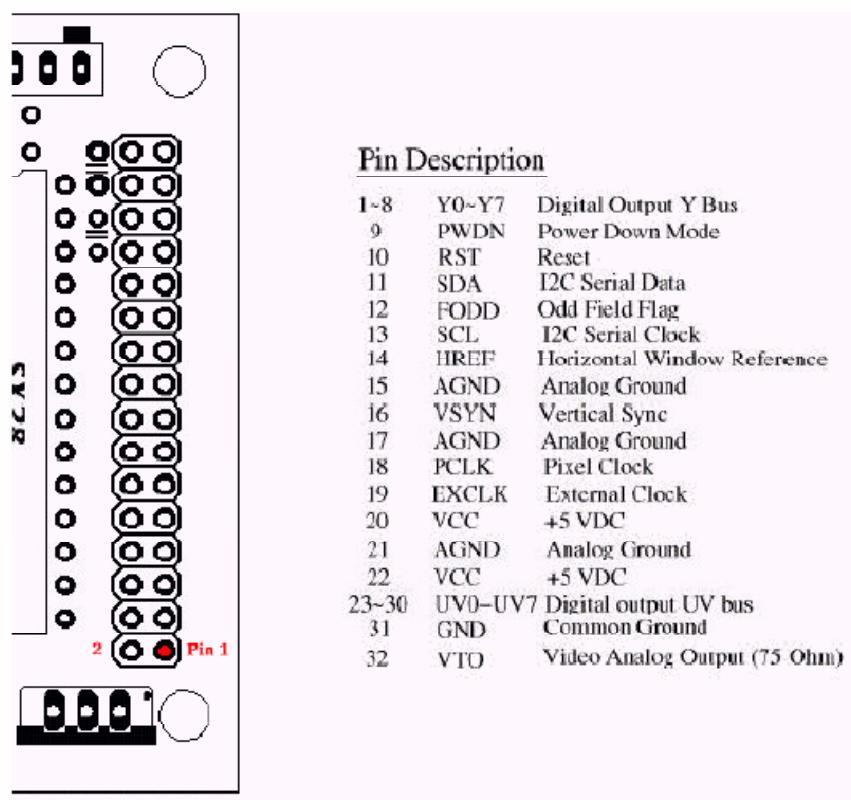


Figure 3.5: Camera bus and its pin description



### 3.2.4.2 CMU camera App Module

In this project, CMU camera App Module (see Figure 3.6) is used where it is specially designed for the Parallax Boe-bot robot. CMU camera easily bolts onto the front of the Boe-bot. It plugs to the AppMod header (see Figure 3.7) that provides a connection and signal routing for additional modules. All pins are routed through the AppMod connector.

The CMU camera has four wires plug into breadboard on Boe-bot's board of education. As the red wire plugged into Vdd connector, black wire plugged into Vss connector, and blue and white wires are used for serial communication. In this project, blue wire is plug to pin P7 and white wire is plug to pin P9, where P7 as a transmit wire and pin P9 as a receive wire.

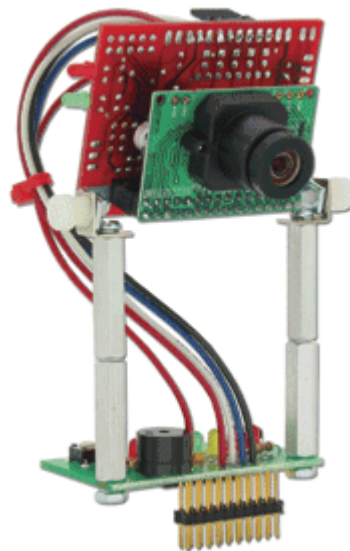


Figure 3.6: CMU camera App Module

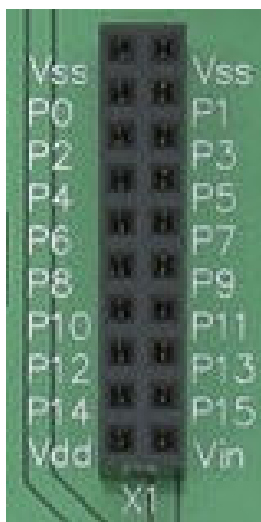


Figure 3.7: AppMod Header

The lens become pre-focused and set for use on the Boe-Bot. Before opening the plastic bag containing the CMUcam1, be careful to discharge any static charge from body before handling the CMUcam1. Although the CMUcam1 is quite rugged static discharge can damage it. When adjusting the robot focus, we need to take extra care because the high sensitivity of the CMUcam. Besides, we need to keep the detecting object in range because the robot is difficult to identify and tracking objects when robot gets far out of focus.

As a conclusion, the robot holds two parts which consists of microcontroller which is BASIC Stamp 2 and SX 28 that used for handling data processing and CMU camera as sensor that for output information such as length of a sign.

## **CHAPTER 4**

### **SOFTWARE IMPLEMENTATION**

#### **4.1 Introduction Software Implementation Process**

In this chapter, we may know the basic flow of whole operation and pictures us how the robot process from the beginning of the operation, response, and react over the input which is sign detection. Once we know the flow of the software, we may gain knowledge how every step and form a full process.

## 4.2 Flow chart of project

There are two major parts that need to be complete along the research which are the software development and online simulation. In this project, the Boe-Bot robot and CMU cam is used and apply on the implementation of a video tracking system. From the software implementation, we are able know the basic flow of the operation on how the operation flows, response and react over the input. Understanding the flow of the whole system is the most important steps to form a full process.

The program flow on the main system is shown as below.

- a) Initialization
- b) Setting window and sign lock-on
- c) Perform sign tracking
- d) Based on the arrow pointed: move forward, turn left or turn right
- e) Return to main

The following diagram, Figure 4.1 illustrates the flow of the software.

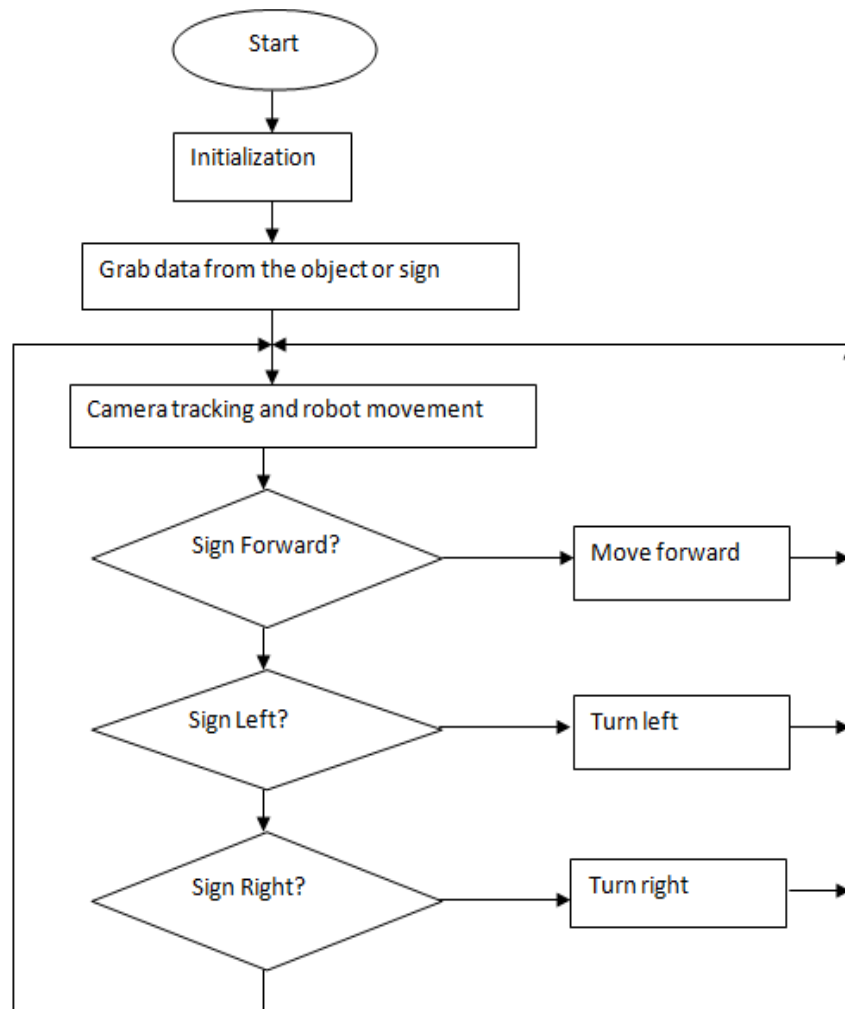


Figure 4.1: Flow chart of whole performing shape detection and robot movement

#### 4.2.1 Initialization

In software implantation, the first task is initializing the CMU camera with do the calibration process. It is important to calibrate the CMU camera to the lighting environment so that it can be operates in certain required field. The calibration process is important to make sure it successfully identifies and track shape using CMU camera.

In this project, as the CMU camera started, it takes one second before a reset sent to synchronize the CMU camera. The BASIC Stamp that controls the CMU camera also will be synchronizing.

To identify the current lighting environment, the auto white-balance is adjusted to turn on for five seconds for its setting time. This is important so that CMU camera can operates in required field. White-balance attempts to correct the camera's color gains based on the ambient light conditions and the environment.

In this process, the command L1 is sent in order to control the tracking light. Generally it has 3 inputs. For beginner, the command L1 with value 1 is sent to turn in the tracking light. But at the end of this step, the value 1 is changed to 2. This value puts the light into its default auto mode. The '0' value state used for disable the tracking light.

In order to properly set the auto-gain and white balance, the Boe-bot is place at where it will be operates without any object. The green LED illuminates for 5 seconds. During this time the auto gain and auto white balance are being set. Now, CMU camera is ready to track the object. The flow of initialization is shown in figure 4.2.

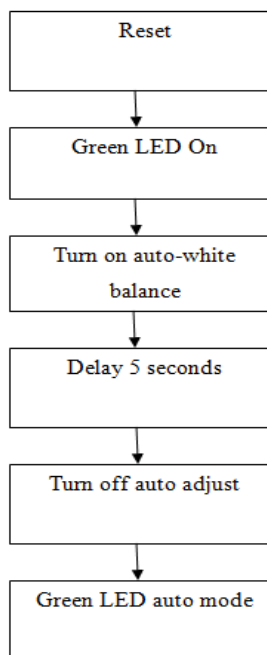


Figure 4.2: Flow chart of initialization

#### 4.2.2 Setting window and sign lock-on

In this project, two types of mode were implemented to setting the CMU cam window which is Pole Mode and Raw Mode. When pole mode function is engaged, only 1 packet data is returned when an image processing function is called. This is useful when we rapidly change parameters or we had a slow processor that cannot keep up with a given frame. When camera sent with pole mode off, the camera will continuously send data packet at 17 fps. This can exhaust most of the processor resources. When engaged with pole mode, the camera only sends one data packet of information. Thus, the processor has time to do other operations and can control when it wants to receive a packet of information from the camera.

Another type of mode setting is Raw Mode. Raw mode configures the camera to return packets in raw data bytes rather than in printable ASCII characters. By using raw mode, it saves time because we do not have to go to the trouble of converting the ASCII to a raw data byte because this has already been done in raw data mode command.

In this project, Boe-Bot is integrated with CMUcam on top of the robot. Since there are proper connection between the board of education and CMU camera, there is no necessary for servo movement. Other than initializing commands, there were window tracking command and sign detection commands that been used where both of this commands are the main commands that has been used to accomplish my project scope which is detecting the sign in front of the CMUcam.

The Track Window (TW) command is used to track the color found in the central region of the current window. After the color is grabbed, a tracking function is called with those parameters. This is useful for locking onto and tracking an object held in front of the camera.

Track window command calls the get mean command for color information and statistics and passes this to the track color command. Which mean it saves parameters used in a previous call. So it can easily track color blob on the centre of the window.

The flow of this process is shown in flow diagram at Figure 4.3. After the initialization process, an object is place in front of the camera. At this moment, pole mode command is sent. Setting pole mode command with value of 1 enables the camera to only send back one packet data of information.

Then raw mode command is sent to configure the input to the camera and camera returns packets of data information in raw data bytes. After setting window for proper location to look for object, then track window command tracks and grab the color information of the object on the centre of the window. These parameters then send to track color function and save as a data base for further process.

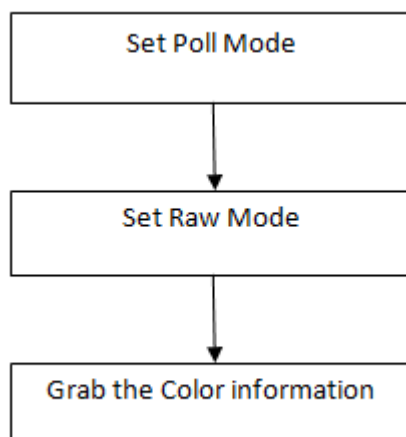


Figure 4.3: Flow chart of setting window and object lock on.



### 4.2.3 Perform sign tracking

The third part is the main procedure in this program. In this part, the robot will save the color and all the other data that has been grabbed from track window command and then compare to the data base as to identify the presence of arrow occur in front of the camera.

First, command SEROUT 7, 84, ["TC", CR] is sent. This command will track color grabbed by the track window once it an arrow appears in front of CMU cam. Next, the command SERIN 9, 84, [STR Arrdata\10] is sent. By using this command, it will returns certain information such as x and y-coordination of middle mass, pixel value of the frame grabbed, confidence level and other information to the stamp in form of 10 array length, called Arrdata.

The subsequent calls to track color with no parameter will return a data packet of M type which in a raw data bytes and the format is arranged as below:

Type M packet

*M mx my x1 y1 x2 y2 pixels confidence\r*

The M indicates that the camera is returning a middle mass TC packet. Mx is the middle mass x coordination of the tracked region and My is the middle mass y coordinate. The next four bytes give the coordinates of the tracked rectangular. Pixels are the number of pixels in the tracked reguin capped at 255. Confidence shows how well the camera has lock on the desired color. A confidence >50 indicates a good lock, while a confidence < 8 indicates a poor lock.

The next line of Cnfd\_Level=ArrData(9), sets the variable confide equal to the confidence level returned by the camera bytes, which are the ninth data of the return data packet. The array ArrData contains several important variable including the y coordinate located at ArrData(3) and the x-coordinate located at ArrData(2). The range of the x-coordinate is from 0 to 80 while range of y-coordinate is from 0 to 143.

#### 4.2.4 Robot Movement based on the arrow direction

This is the final module of this program. In this module, movements of the robot are based on the arrow pointed. As an example if it is pointed to the right or left, the robot will turn right or left variable amount based on the program. In this project, when the arrow is detected, it will be compared to the data base value to check on the true arrow. When the arrow is identified, the program will only start tracking the direction of the arrow. The tracking window will focus on first quadrant as we divided the tracking window into 4 quadrants. As this step can be done by giving out the command Set Window (SW) which set the window size of the camera. It accepts the x and y Cartesian coordinates of the upper left corner followed by the lower right of the window as we wish to set. The origin is located at the upper left of the field of view.

As the beginning of the project, the robot moves forward until it “found” an arrow. Then it started to identify the arrow pointed. As the arrow is a left arrow, the y coordinate will be less than 70. The robot will move the left. When the arrow is a right arrow, the x-coordinate will be greater than 65 and y-coordinate will be greater than 80. When the robot detects the backward arrow, the x-coordination will be less than 60, the robot move backward. Despite on what type of movement is executed, the software flow to the top of the main module and coordinates are reacquired from the camera starting the sequence over again. All steps is shown in figure 4.4. The combination of entire step is shown in Figure 4.5.

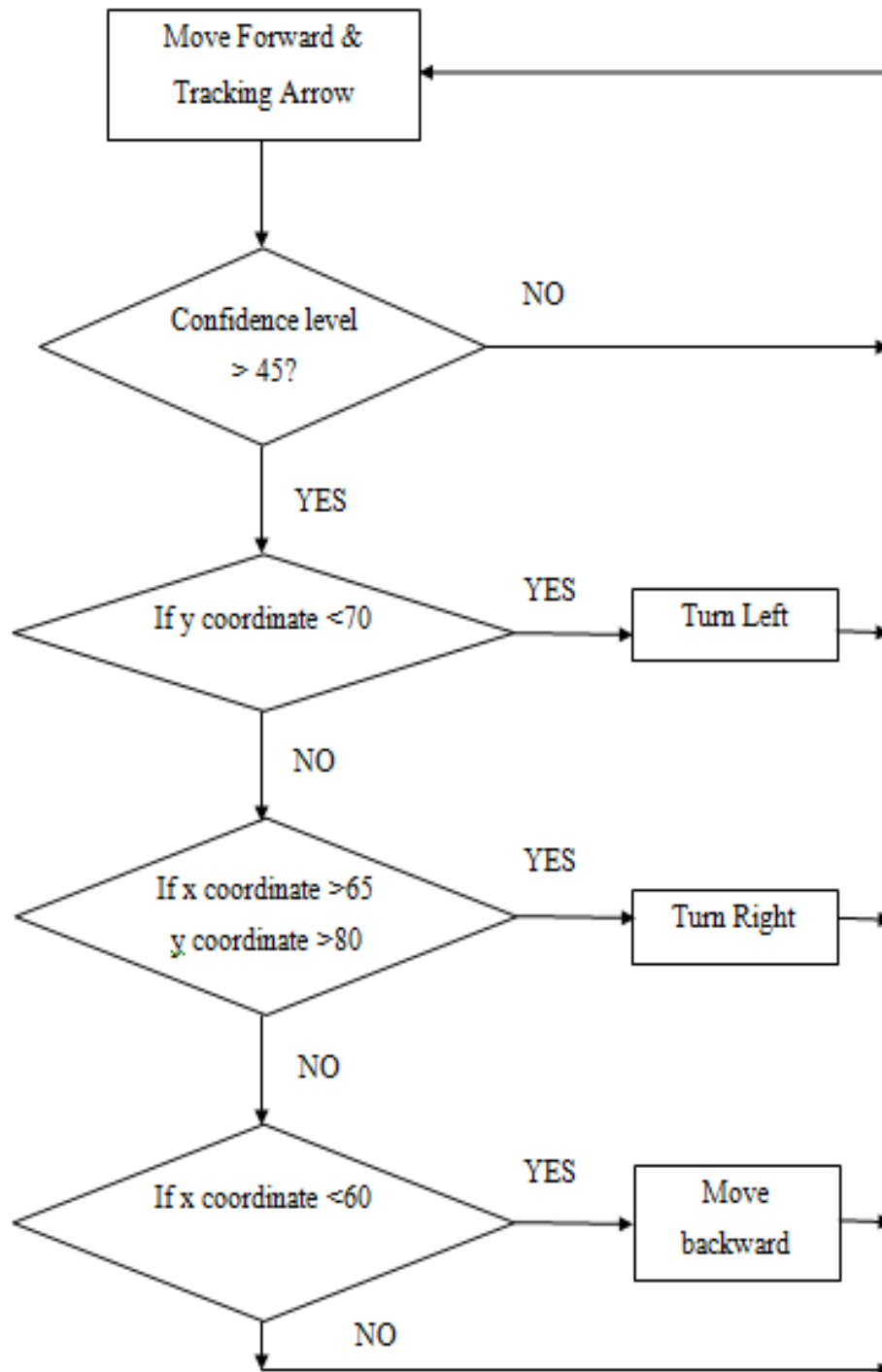


Figure 4.4 Flow chart of performing sign tracking and robot movement

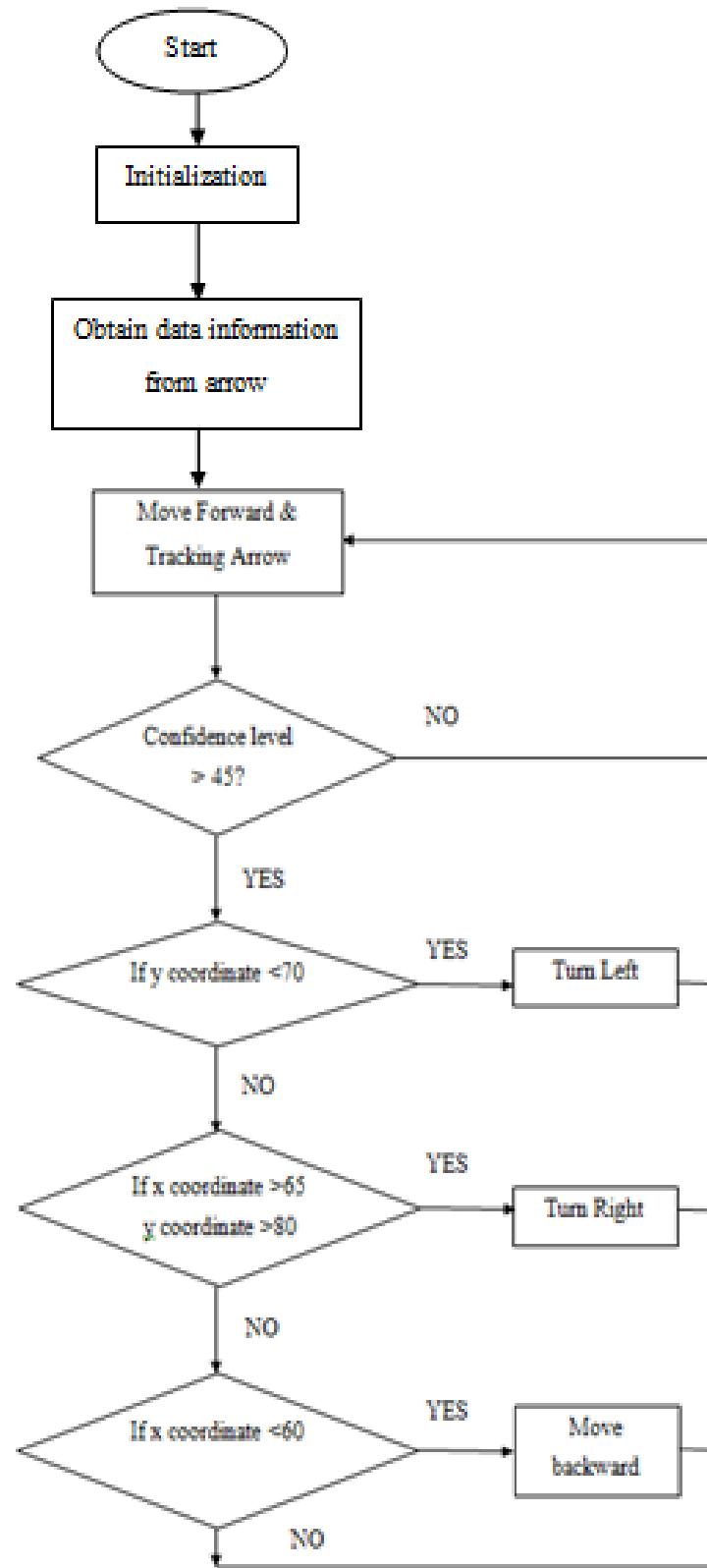


Figure 4.5 Flow chart of full process

### 4.3 Result

After the software program had been loaded to the program, the BoeBot will execute the program and operation according to be program developed. The robot is successfully moved to the first arrow that had been seen. The robot movement is depends on the arrow direction. When the arrow is pointed right, the robot will turn right. When the arrow is pointed left, the robot will turn left and if the arrow pointed back, the robot will move backward. When there is no arrow, the robot will moving forward several distances for detection purpose.

The project is expected to successfully when the mobile robot can operate smoothly in detecting sign. The implementation of software tracking system will induce the robot to follow the arrow and performing movements. The main objective of this project is to make UTEC's Vision Robot detect to sign, which will operates according to the arrow pointed within the operation range. The robot will move forward and constantly tracking for the presence of an arrow. When the robot detects the target arrow, the robot will perform movements based on the arrow pointed.

As stated before, the basic operation of this robot is color detection on an object by CMU cam. The camera has a resolution of 80x143 for the image pixel. When CMU cam grabs data from the window, window will return a packet of data with 80x143 resolution frames. The window of the camera has 1 to 80 of the x-coordinates values and 1 to 143 of y coordinates values. Once an arrow is detected, the resolution if the window will restore back basic resolution of 80x143.

When the colored of the sign is in front of the camera, the entire frame of data will received and the camera will returns data consists of the X, Y components of centroid, the four coordinates of the color bounding box, number of pixels and the confidence value of the object tracked.

Before the tracking system begins, the mobile robot will move forward until it detects an arrow. The camera then will lock on the color of the sign detected and check on the confidence value to identify whether is the "correct" sign for the system.

As the correct sign is detected, the frame now is set for smaller frame which for this system, the x coordinates is set from 40 to 80 while y coordinates is set from 60 to 143. As this small frame area, the details on the arrow will be taken for tracking operation. As we divided to four quadrants, the frame taken would be first quadrant.

For left operation, the value related is the coordinates of y values. When a left sign is tracked, the frame now is focusing on the tail of left arrow. As the camera now will detect the resolution value is less than y coordinates. For this condition, the value is set to 70. Respond to this, the robot will move to left.

For right operation, the value related is x coordinates and y coordinates value. When the sign is detected, the upper head appears in first quadrant. It will detects the resolution is larger than both x and y coordinates values. For this condition, x value is set as 65 and y value is set as 80. Respond to this, the robot will move to right.

As for backward operation, the value related is x coordinates. When backward sign is detected, the resolution is smaller than x value. For this condition, x is set as 60. Respond to this, the robot will move backward.

#### **4.4 Discussion**

In this project the robot will grab the color of the first object appear in front of the camera. The sign is placed in front of the CMU cam after the robot is turned on. After certain delays, the robot will perform movement according to the direction of the sign. If any other colored sign in front of the robot, the robot will not response to it. The robot will only follow the first color sign because the robot has only the previous information saved in its memory. So when there is other different arrow seems to appear in front of the camera, the robot will ignore them.

UTEC's Vision robot has several operate steps. The process included integrating CMU cam with the BoeBot in order to perform object detection. The operation starting from choosing object to detected, setting window resolution for better detection, powered the robot, identify the correct colored object and start tracking system.

#### **4.4.1 Choosing object to track**

In this process, before start up the robot system, a brightly colored object is chosen for tracking process. In this project, colored sign with sharp contrast to the surrounding environment is chosen for the best result of CMU cam to capture the image and makes them easily tracked by CMU cam. The size of the sign should be at least four inches square.

#### **4.4.2 Camera Adjustment**

To perform the best detection, the angle of the camera needs to be adjusted to the best angle for the detection. For an effective detection, the robot is going to track and the sign is placed on the ground. The floor surface must be smooth and clean so that the robot would not have any difficulty to moving around.

#### **4.4.3 Power Ups the Robot**

To start the system operation, we need to power up the robot by plugging the cord from battery source located on the bottom of the robot which needs 4 of AA's 1.5V batteries or another battery source located on the top of the robot which needs a 9V battery.

#### **4.4.4 Locking on colored sign**

After power up the robot, there will be 5 seconds delay for the lighting condition. At this moment, the indicator green LED on the circuit board of the CMU cam will be illuminated. At this moment, the camera is adjusting to setting to match to the environment for best resolution for the tracking system. After that, the green LED will turn off for 5 seconds. This delay gives time to position the sign in front of the camera. After that a 5 seconds delay for the camera to describe the arrow's color and save all the characteristics and data information needed in memory. When the camera begins color lock-on process, the green light will turned off. This indicates the camera is operating a lock-on color process. When the camera has finished locking-on to the color, the green LED will turn back on. The LED on indicates the camera detects the target color within its operating range.

#### **4.4.5 Tracking sign**

In order to observe how the robot responses to the object within its operating range, robot is set for moving forward until it found a sign at the beginning of the process. The target object also has to be center of the window frame so can only perform a high accurate operation. From previous explanations, the movement of the robot is based on coordinate position.



## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **5.1 Introduction**

This chapter discuss about the conclusion and future development of this project research. There are 2 parts consisted in this project which are hardware description and software implementation.

## **5.2 Conclusion**

As a conclusion, this thesis consist of 5 chapters which are introduction, literature review, methodology, hardware and software implementation, result and discussion, and conclusion have been discuss for the development of this final year project.

The implementation of this project was successful since the main objective has been achieved. The Boe-Bot operates smoothly through software implementation, PBASIC language which is the robot will grab the all the details of the object and followed the program generated.

A vision system technology can easily applied with a low startup cost and tiny technical support from the designer. As we know, vision system can detect an object that occur in very odd situation as it can detect all the details of the object where most of other sensors only detect blur image without details of the object.

## **5.3 Future Development**

Considering the rarity of vision robot projects, I felt that my project was both challenging and innovative. The project touched a variety of engineering areas including integrating hardware and testing software. Hopefully a particular design of this project can become the backbone for other researcher's field study.

For future project, extra circuit or devices can be implemented to the Boe-Bot such as LCD display for showing the status of the processor execution and be together integrated it with CMU cam for industry application.

## 5.4 Costing and Commercialize

For this project, whole body kit of BoeBot had to order from overseas. Hence for the pricing for BoeBot and CMUcam are shown in US dollar. For BoeBot Body Kit packs, it comes with all assemble parts but users have to assemble the robot themselves. Mean while, The Boe-Bot CMUcam uses the same PCB as the general purpose package but does not include the parts that convert the serial port to RS232C levels compatible with a PC serial port. It also doesn't include the voltage regulator and jumper headers since they aren't required when using the BASIC Stamp microcontroller. This version can be modified for use with a PC by adding the parts that were left off the board but the Parallax warranty is voided if modify the board. While cannot directly view a picture from the camera on your PC with the Boe-Bot CMUcam setup, a demo program is included that allows to display the tracking data packets from the CMUcam on the PC debug screen. For the costing for this project is shown in Table below:

Table 5.1: Robot Costing

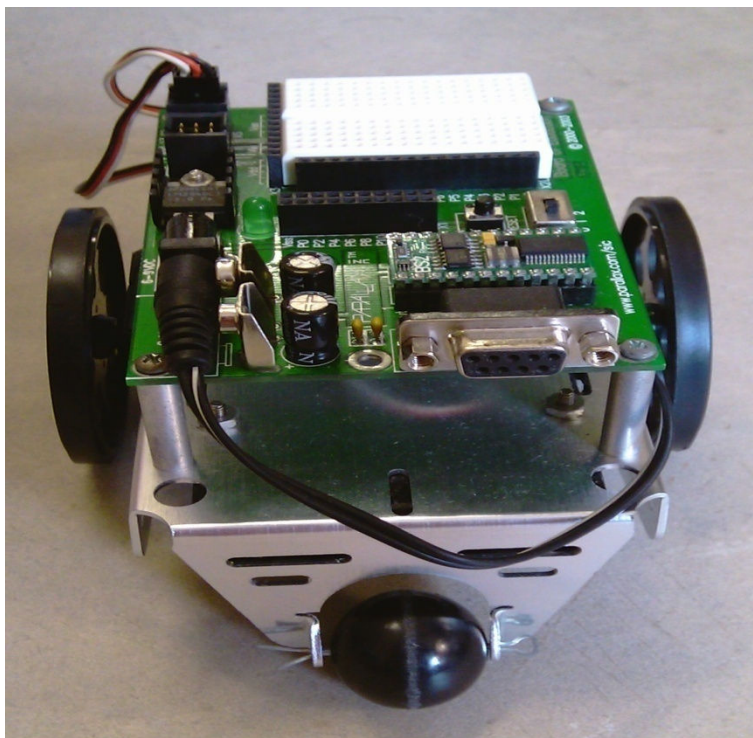
<b>Item</b>	<b>Cost (US dollar)</b>
BOE-Bot Full Kit	159.99
BOE-Bot CMUcam	139.99

## REFERENCES

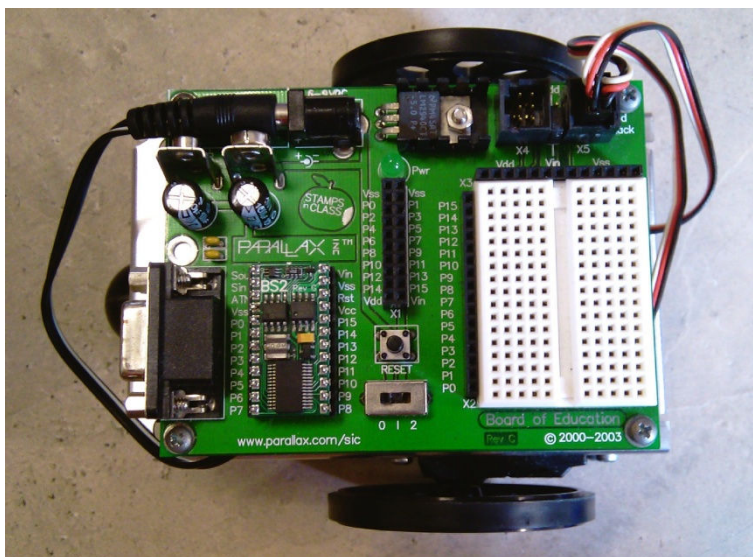
- [1] Seattle Robotic CMUcam Vision System for Boe-Bot User Manual, Carnegie-Mellon University, 2002.
  
- [2] D.Boucugnani, D. Barnhard, and D. Caveney. “Phoenix & Dragon:parallax BoeBot/ Lego Technics Hybrid Solutions to the Reactive Maze Traversal and Deleberative “Cheese” Location Problems”, the University of Georgia.
  
- [3] O. Akihisa, K. Akio, and K. Avi. “ Vision-Based navigation of Mobile Robot with Obstacle Avoidance by single Camera Vision and Ultrasonic Sensing”, University of Tsubaka and Purdue University.
  
- [4] A Rowe, C. Rosenberg, and I. Nourbakhsh (2002). “Proceedings of IROS 2002”
  
- [5] Cao, Z. L., Oh, S. J. and Hall, E. L. (1986), Dynamic omnidirectional vision for mobile robots. *Journal of Robotic Systems*, 3: 5–17

- [6] S. Michael (2005). "Road Sign Detection and Recognition", National Institute of Standards and Technology.
  
- [7] C.L Huang, and S.H Hsu. Road Sign Interpretation using Matching Pursuit Method. IEEE International Conference on Pattern Recognition (ICPR2000), 1329-1333.2000. Barcelona, Spain.
  
- [8] P. Paclik, and J. Novovicova. Road Sign Classification Without Colour Information. Proceedings of the 6th Conference of Advanced School of Imaging and Computing. 2000. Lommel, Belgium.
  
- [9] Andy Lindsay. "Robotic with the Boe-Bot, Version 2.2". Parallax Inc.

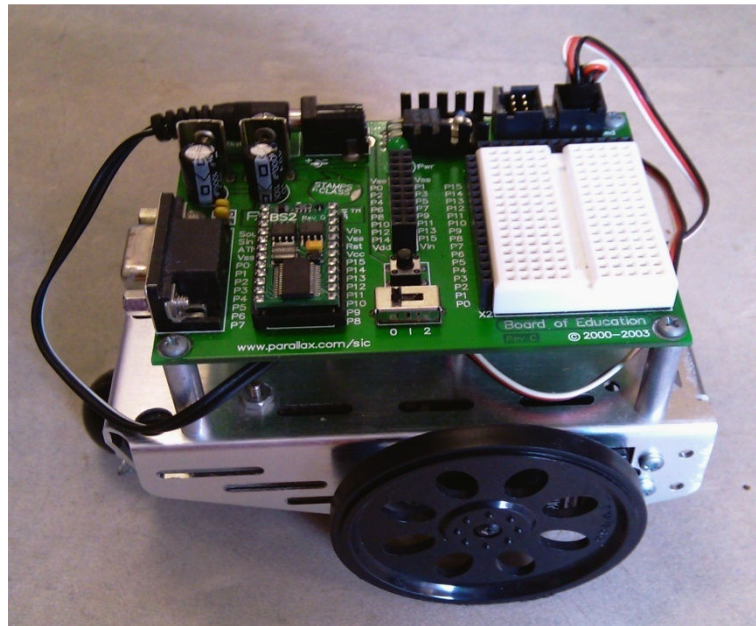
## APPENDIX A: Robot Pictures



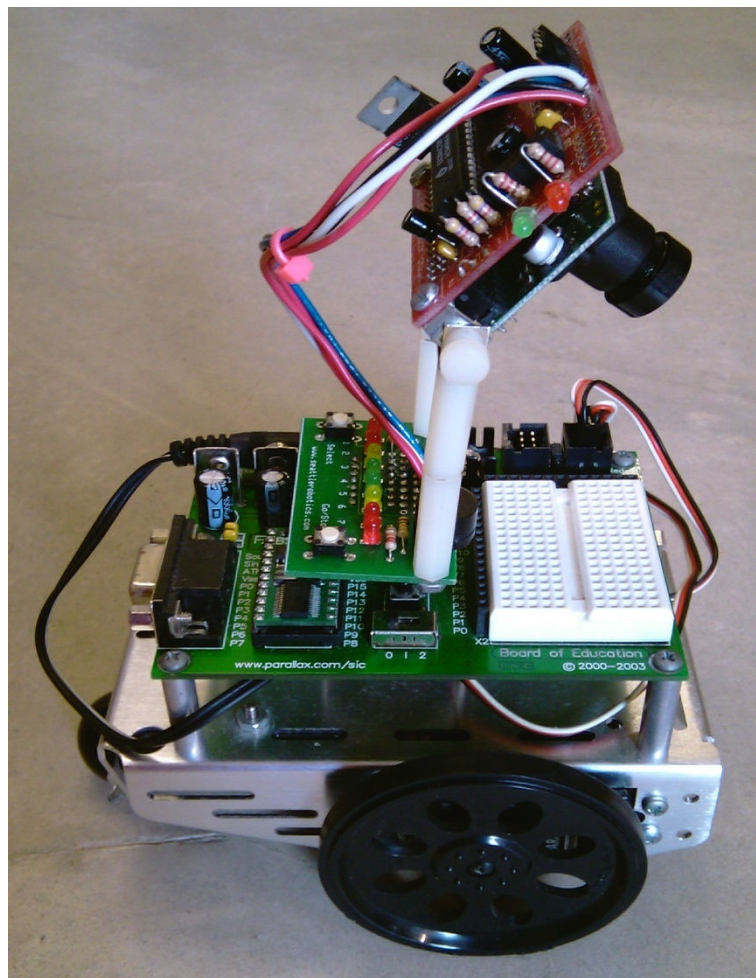
Front View of Boe-Bot



Top view of Boe-Bot



Side View of Boe-Bot



UTEC Vision Robot

## APPENDIX B: Code Listing

```
' {$STAMP BS2}
```

```
' {$PBASIC 2.5}
```

```
=====Variable=====
```

```
ArrData VAR Byte (10)
```

```
n VAR Byte
```

```
Cnfd_Level VAR Byte
```

```
=====Initialization=====
```

```
'Pause 1 second for CMU cam to startup
```

```
PAUSE 1000
```

```
'Send "Reset" to synchronize CMU cam and Basic Stamp
```

```
SEROUT 7, 84, ["RS",CR]
```

```
SERIN 9, 84, [WAIT(":")]
```

```
PAUSE 1000
```

```
'Green LED On
```

```
SEROUT 7, 84, ["L1 1",CR]
```

```
SERIN 9, 84, [WAIT(":")]
```

```
PAUSE 100
```

```
'Adjust Lighting
```



SEROUT 7, 84, ["CR 18 44",CR]

SERIN 9, 84, [WAIT(":")]

PAUSE 100

‘Pause 5 seconds for CMU cam to auto adjust to lighting conditions

PAUSE 5000

‘Turn off auto adjust

SEROUT 7, 84, ["CR 18 44 19 32",CR]

SERIN 9, 84, [WAIT(":")]

PAUSE 100

‘Auto Tracking mode, Green LED On

SEROUT 7, 84, ["L1 2",CR]

SERIN 9, 84, [WAIT(":")]

PAUSE 100

‘Pause 5 seconds for user to place target close in front of the camera

PAUSE 5000

=====Setting camera and Object Lock-on=====

‘Setting Pole Mode

SEROUT 7, 84, ["PM 1",CR]

SERIN 9, 84, [WAIT(":")]

PAUSE 100

‘Setting Raw data Mode

SEROUT 7, 84, ["RM 3",CR]

PAUSE 100

‘Track Window command

SEROUT 7, 84, ["TW",CR]

PAUSE 2000

=====MAIN PROGRAM=====

=====Perform Sign Tracking=====

MAIN:

‘Track Color Command

SEROUT 7, 84, ["TC",CR]

SERIN 9, 84, [STR ArrData\10]

‘Set Cnfd\_Level equals to 9<sup>th</sup> array of ArrayData

Cnfd\_Level=ArrData(9)

‘Moving Forward

FOR n= 1 TO 10

  PULSOUT 13,900

  PULSOUT 12,550

  PAUSE 20

NEXT

‘Perform Tracking sign

IF Cnfd\_Level>35 THEN TRACKING

GOTO MAIN

TRACKING:

‘Setting window command

SEROUT 7, 84, ["SW 40 65 80 143",CR]

IF ArrData (6)>65 AND ArrData (7)>80 THEN RIGHT

```
IF ArrData (7)<70 THEN LEFT
IF ArrData (6)<60 THEN BACK
GOTO MAIN
```

```
=====Robot Movement=====
```

```
LEFT:
```

```
FOR n =1 TO 10
  PULSOUT 13,650
  PULSOUT 12,650
NEXT
GOTO MAIN
```

```
RIGHT:
```

```
FOR n= 1 TO 10
  PULSOUT 13,850
  PULSOUT 12,850
NEXT
GOTO MAIN
```

```
BACK:
```

```
FOR n= 1 TO 10
  PULSOUT 13,550
  PULSOUT 12,900
  PAUSE 20
NEXT
GOTO MAIN
```