Mobile Robotic Design

Robotic Colour and Accelerometer Sensor

Euclid Weatley MILLS

Submitted for the degree
of Master of Philosophy

Department of Engineering and Technology

University of Bradford

2010

Abstract

This thesis investigates the problem of sensors used with mobile robots. Firstly, a colour sensor is considered, for its ability to detect objects having the three primary colours Red, Green and Blue (RGB). Secondly, an accelerometer was investigated, from which velocity was derived from the raw data using numerical integration. The purpose of the design and development of the sensors was to use them for robotic navigation and collision avoidance. This report presents the results of experiments carried out on the colour sensor and the accelerometer. A discussion of the results and some conclusions are also presented. It proved feasible to achieve the goal of detecting colours successfully but only for a limited distance. The accelerometer proved reliable but is not yet being applied in real time. Both the colour sensor and the accelerometer proved to be inexpensive. Some recommendations are made to improve both the colour sensor and the accelerometer sensors.

Key words: Colour Sensor; Accelerometer; Navigation; Collision Avoidance; Mobile Robotics.

Dedication

In Memory of Paul Francis, my friend, who did not live to witness my goals that he encouraged. Rest in peace my friend.1965 to 2003.Also to my friend Clive Claxton, 1954 to 1994, Rest in peace my friend and god bless, we all miss you both. Who would have thought a descendant of slaves 200 years ago would get this far and further god willing.

Acknowledgements

I would like to thank the library staff, especially Ellie Clement and the laboratory staff especially Terry and John for their help and co-operation. To the three supervisors: Dr H. S. Rajamani (The primary supervisor); Dr J. C. Readle and Dr R. Wyatt-Millington (The secondary supervisors) for their help and assistance. Thank you all, also especially Michelle Benn and Margaret Errington, my tutors, Thank you.

Table of Contents

Figures List

Table List

Glossary

Xflt	X axis filter
Xa	X axis designation a
Xb	X axis designation b
Yflt	Y axis filter
Ya	Y axis designation a
Yb	Y axis designation b
Triangulation	A method of determining co-ordinates for a location
Trilateration	
DCM	Duty cycle modulation
RGB	Red Green Blue
LDR	Light Dependant Resistor
Cds	Cadmium Sulphide
Polysilicon	Refer Figure 2.7
Mcd	milli candela
Heuristically	By trial and error
LED	Light Emitting Diode

CHAPTER 1 Introduction

1.1 Introduction

The development of robotic systems has led to an equal rise in the development of sensors. There are various types of sensors that are possible due to modern technology. The idea behind this research is to see how successfully a sensor can be used to detect objects and also be able to locate the robot's position also its acceleration and velocity. The sensors come in all shapes and sizes. They have many different properties for measuring various parameters, such as velocity and acceleration which can be derived from an acceleration device by numerical integration. Other advance sensors such as imaging devices can recognise colour scales according to temperature variations. They can also be used to view a physical object and visualise hot spots or heat emanating from an object. From the most sophisticated to the basic robot, they all use some sort of sensor. In this report the robot used was a basic mobile robot, with two different types of sensors (acceleration device) and one other type the (colour sensor); they measure acceleration/velocity and colour respectively. The list of sensors is summarised according to their different properties and how they would fit into the robotic control environment. The first criterion was to navigate and identify target objects, such as using colours and motion signals for location data. Thus how to navigate a way around a coloured object by reversing and turning left or right, in order to steer around objects in its path. Secondly, the advantages and disadvantages in the summaries helped in narrowing down to two main types of sensor(s) to meet the criteria. Finally another criterion was cost since the project was limited in budget. This chapter covers various sensor

types on the market. Further on in this chapter, is an overview of the project, looking at the properties behind the chosen sensors.

1.2 Survey of Sensor(s)

A summary of a range of appropriate sensors are given below.

1.3 LDR

Light sensors are inexpensive and robust; they work by light, being shone on a Light Dependant Resistor (LDR), which produces a current proportional to light intensity. Other photo devices such as photo diodes and photo transistors, which are also light sensitive and work by the same principle as the LDR. They can be used in various applications. But photo resistors are suitable for ambient or room lighting detection, whilst the photo-transistor proved the better option.

Advantage: good for ambient room light, inexpensive, robust, easy to implement. (Martin, 2001)

Disadvantage: not unique, not overly sensitive in other light.

1.4 Infra Red Proximity Sensor

This type of sensor is expensive. It operates by reflecting its light source off a white flat surface and is then detected by an Infra Red detector. With a light source shining, directly on the IR detector and a rotating disk with slots in the disk to interrupt the IR beam reflected by the white surface, giving rise to pulses that represent the detector, detecting an obstacle.

Advantage**:** faster response to changes in ambient light. It is suitable for break beam shaft encoder application. The signal generated by the detector can be modulated and demodulated.

Disadvantage: not unique, commonly used.(Martin, 2001)

1.5 Ultrasonic/Sonar Sensor

Sonar sensors are expensive to buy. But it is possible to construct your own and acquire a circuit design to do so. The principle of its operation is that it generates an echo, which is picked up by a receiver. The time it takes to transmit and receive, is proportional to the distance between the robot and the obstacle. Because of the nature of the sonar circuit used in more expensive versions, when component is failed, they are expensive to replace them and add to the overall expense.

Advantage**:** although expensive, cheaper versions available, robust and reliable. (Seattle, 2009)

Disadvantage: expensive versions are prone to parts failure, inexpensive versions if an LC tank circuit is used, it can cause blanking, where the first few centimetre cannot be measured.

1.6 Colour Sensor

The colour sensor is inexpensive to buy and implement. It is also simple to construct. An LDR (light Dependant Resistor) is surrounded by three different colour LED’s, whilst the LDR is surrounded by heat shrink sleeve, to prevent the LED’s from shining directly on to the LDR, thereby making it more sensitive.

The LED's are Red, Green, and Blue (RGB) also the same applies to the cards which are placed in front of the sensor, so that its colour can be identified by the sensor. The sensor is calibrated by the coloured light reflecting off a card's flat surface at various distances.

Advantage: Easy to construct, inexpensive, can be effective.

Disadvantage**:** some types are expensive and more complex.

1.7 Active Beacon Sensor

Active beacons are used on ships and aeroplanes and mobile robots. They have a high cost and require regular maintenance. They operate by having three fixed beacons placed accurately at different points on a circle or at opposite ends of a straight line on a circle. Whilst a receiving beacon on the mobile robot picks up the signal from the active beacons to determine position. There are two methods; these are is Trilateration or Triangulation, which simply means the beacons are placed in a linear or on a triangular format respectively.

Advantage**:** accurate positional determination. As long as active beacons are placed accurately, position can be derived.

Disadvantage**:** expensive, requires regular maintenance.

1.8 Rate Gyro

Rate gyros or just gyros are expensive if they come with inertial navigation system at the top end of the gyros market. Therefore they are not suitable for mobile robots, until now they have been mass produced at the lower end of the market. However a different version exists and is called the laser gyro. These

are very effective in terms of accuracy and are far cheaper at the lower end of the market. This can be used with mobile robots.

Advantage**:** laser gyro cheaper, accurate, suited to mobile robots.

Disadvantage**:** other versions, except for laser gyro, are costly and require maintenance regularly. Some types not suitable for a mobile robots. (Liu, 2001)

1.9 Magnetic Compasses

There are five types of magnetic compasses; the Fluxgates compass; of which the type most suitable for mobile robots. It is the most suitable for the navigation needs for autonomous robot platforms. It is also inexpensive, used in commercial application as well as the military positional accuracy is around ±0.5 degree(x, y, and theta).

Advantage: low cost, accurate, gives positional heading.

Disadvantage**:** due to the nature of the earths' magnetic field and the distortion caused by power lines indoors as well outdoors they can give erroneous headings.

1.10 Landmark Navigation (not a sensor)

Landmarks are distinct features that a robot can recognise from sensory inputs. Landmarks can be any geometric shape e.g. (rectangles, lines and circles). It operates by recognising land marks reliably and calculates their position, relative to the landmark in question. There are two main types, natural and artificial landmarks.

1.11 Natural Landmarks Navigation (not a sensor)

The main problem with natural landmark navigation is to detect and match characteristic features, from sensory inputs. Proper selection of features will reduce the chances of ambiguity and increase positioning accuracy. (Liu, 2001)

1.12 Artificial Landmarks Navigation (not a sensor)

Detection is much easier with artificial landmarks. Most artificial type's position systems are based on computer vision. A variety of landmarks are used in conjunction with non vision sensors. (Liu, 2001)

Advantage: inexpensive.

Disadvantage: only limited commercial support for natural landmarks.

1.13 Accelerometer Sensor

The accelerometer sensor is used to measure acceleration and velocity from this velocity can be derived by integrating with respect to sample data. The measuring device is placed on the mobile robot platform. It can be combined by other measuring sensors, such as a gyroscope to form a dead reckoning position device. The accelerometer device is a solid state and is also inexpensive. It must be noted that there are many different types of accelerometers on the market. The velocity is calculated from the function please refer to (equation 3.1).

Advantage**:** inexpensive to buy.

Disadvantage**:** can have random bias drift due to temperature.

1.14 Inertial Navigation Sensor

The inertial navigation sensor is very expensive and has room for development. It can be combined with other device mechanisms to form positional accuracy, by providing more position information. Current research is looking into how it can be used with other sensor types, which are suitable for mobile robots, However the right error model has to be generated for the bias drift error before it can be solvable by using a Kalman filter also for noise**.**(Liu, 2001**)**

Advantage: Requires no external reference.

Disadvantages**:** expensive

1.15 Overview

The robot used in this project is not the highly sophisticated, advance robot used in industry or sold to domestic consumers, e.g. as a vacuum cleaner or other house hold appliances. It is instead a robot on wheels (mobile) and made out of LEGO blocks and sensors, with microprocessor based controller, referred to as Handy Board. These basic building blocks act as a basic test-bed for mounting motors and sensors. The controller or (Handy Board) is initially loaded by down loading program code, using a language, called Interactive “C” or (IC), this is not to be confused with “C” language, but similar allowing for similarities the basic functions such as, arrays, pointers etc...The program code is down loaded to the Handy Board via a serial data cable, using RS232C, system. The processor then executes the program code and carries out the necessary

operations, in response to the instructions the language is not a powerful one, but has simple commands. For the robots to go forward, backwards, turn or to carry out a series of simple operations, it does so according to the controller (microprocessor) and the codes executed by it. Depending on the size of the program and the size of memory on Handy Board, mounted on the robot can accommodate most programs. The robot is autonomous, so long as the serial cable does not get in the way of its manoeuvres. The Handy Board is capable of supporting sensors of various types, such as touch, light, infra red (invisible light), sonar, ultra sound and laser sensors etc. It is these sensors (colour and accelerometer) and the program code that allows the mobile robot to move in a way limited only by the length of the data cable, by which, data transfer takes place between the console and the Handy Board. Obstacles can be detected by the colour sensor mentioned above so long as the colours are RGB. The mobile robot can then navigate its way around an area and successfully operate, where ever it is commanded to go by the on-board program, on the Handy Board. As well as the program in memory on the Handy Board, the memory has to be maintained with power, since the memory is volatile (loses information if power is lost). Therefore the Handy Board has its own power pack. However sophisticated or intelligent, (artificially) depending on sensors and program interacts making the robot appear in its manoeuvres, to be intelligent in behaviour. This research overview, involves mounting sensors on the Lego chassis platform with electric motors. This will have sensors along with the program software, both interacting, to “home in” on different colours (R, G and B). The sensors will be constructed and the program will be written, to test the theories, using the Handy Board system. From a summary of ten different sensor types available, two were selected based on cost, technical feasibility

and etc., and then the final one was chosen based on its ability to achieve the goal of coloured obstacle detection and robot acceleration and velocity. The two chosen sensors were colour and accelerometer sensors. The colour sensor has the capability to detect objects of different colours especially (R, G, and B).Whilst the accelerometer sensor determines acceleration, from the raw data and velocity by numerically integrating once. Both sensors are inexpensive to buy and construct. The colour sensor and the accelerometer are the means by which detection and velocity information are derived respectively.

1.16 Aims and Objectives

The proposed aim of using a Colour and acceleration/velocity sensors is:-

1. Using the sensor for, the acceleration device that was used to determine the velocity and acceleration respectively of the Robot, on which the sensor are mounted. The sensor (ADXL330) are of the similar type as the previous, and will monitor the acceleration of the robot, then from this, the acceleration and velocity can be evaluated by the computer program using:

$$velocity = initial\ velocity + h\sum_{k=1}^{n}\ (Xa(k) - offset)$$ equation 1.1

where

h is the sample interval

Also (Xa, Ya) is the first axis (Xa) and second axis (Ya) of the accelerometer is the data samples of which (Xa, samples) and (Ya, samples), were plotted as shown in (Figures 3.5 to Figure 3.10) respectively.

2. The second sensor is a colour sensor, which works by reflecting and/or absorbing R, G and B light which is detected by a phototransistor. The coloured light being reflected by the coloured card or coloured obstacle, which in this case should be Cherry Red, Lime green and Navy Blue obstacle in front of the

robot i.e. (R, G, and B). When the robot detects these colours in its path and therefore the coloured obstacle, the robot responds by executing a routine in its program, that causes the robot to reverse, turn left or right, in an attempt to steer around the obstacle. This technique could be used where an application, e.g. requiring sorting boxes with R, G and B colour stickers on them. Once a colour of RGB is detected the routine in the program can be commanded to push the boxes in a particular direction, left/right or forward or backwards, hence sorting boxes.

1.17 Organisation of the Report

This report begins with Chapter 1, introduction and continues with a series of summaries and survey of various sensors, from which one type of sensor was chosen. Also as part of chapter 1, the justification, for the choice of research project is given. Chapter 2 discusses the design of the colour sensor is also presented and how the accelerometer was used. In chapter 3, the results of the experiments are the pseudo-code for calibrating the colour sensor and the basic template for the sensor program. Since the accelerometer, came in as a ready built package, that was already built, just requiring connecting wires to be attached and is self contained. The accelerometer did not require calibration due to it being factory set. The rest of chapter 3 present the details of the acceleration device, by using the acceleration and velocity plots. Chapter 4 discusses results and data analysis. Chapter 5 discusses further work and a conclusion. The penultimate section is references and bibliography, which represent the source of contribution for the sensors used, in this project. Finally the report ends with appendices, A and B. Appendix A describes the Handy

Board control computer used for the project. Appendix B lists the Interactive C code used in the project to control, test and monitoring the mobile robot.

CHAPTER 2 Design

2.1 Design of Colour Sensor

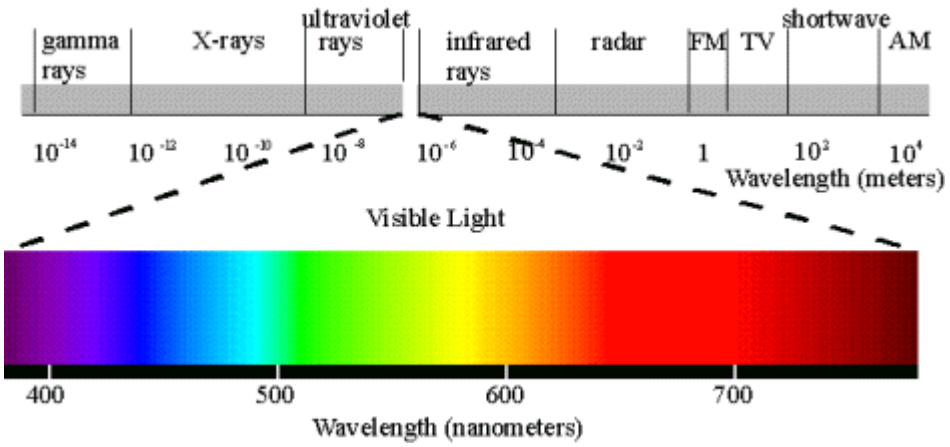
According to the Society of Robots, colour does not really exist. This is due to the fact that, it is part of the Electromagnetic spectrum which emits a wave length of a given value, e.g. typically for green 520nm. When the wave length hits the back of the eye, millions of specialized neurons in the brain translate the wavelength into colour. The best way of looking at colour, is to see colour as invisible and visible, thus you can see colours but not infra red, this is invisible. If for example the sensor, detects red, the colour that is detected is, e.g. red, and therefore the sensor detects difference according to reflection and the reflected colour is detected. If a red apple and a green apple were placed in front of a red light, for example, the green apple would absorb the light, whilst the red apple would reflect the red light, thus the sensor would detect the red apple. The opposite would happen if a green light was shone upon the two apples. The red apple would absorb the light whilst the green apple would reflect it. Thus the green apple would be identified by the sensor. It is clear that distance is a factor: if the robot sensor is physically close to a coloured object, the coloured object can be more effectively detected. Since the robot would have to be very close in order to detect the coloured object, at the distance quoted earlier, the robot would collide with the object due to the closeness for detection to occur. To avoid this, the average data from each colour test, were used to improve colour distinction between each colour (R, G and B) as it can be seen in Tables 3.1 to 3.6 in chapter 3 where percentages were calculated and placed in tables according to 3 different environments of ambient light.

2.2 Colour Sensor Implementation

The colour sensor was constructed, from three LED's, Red, Green and Blue or (R G B) and a Photo-Transistor, with an internal pull up resister. These were in a cluster, so that when coloured cards, (R G B) are placed in front of it, so that the coloured lights (RGB) are reflected back into the photo-transistor and/or absorbed by the RGB coloured cards, and reflected on to the photo-transistor, therefore detected and identified. This is then converted into a voltage by the sensor circuit and supplied to the Handy Board as a data sample. The data sample values represent coloured light intensity are due to the distance of the coloured cards. The numerical data produced by the sensor (photo-transistor) were sampled via the program (colv2110.c and colv2111.c appendix B) which was written to identify a particular range of colours, (R G B) cards, which when detected and identified, the name of the colours displayed on the Handy Board LCD display, so long as the coloured card is held in front of the Colour sensor, i.e. (R, G, and B).

In order to prevent stray light sources, black P.V.C insulation tape is wrapped round the cluster as shown in Figure 2.2. This is very important since external light can cause problems, in the detection of colours; any surrounding light is referred to as ambient light. Because the R G B LED's of a high brightness, not the standard LED's, but the specialist bright types: 1200mcd to 3000mcd's. In the case of the R G B LED's, red brightness is 3000mcd's; green brightness is 3000mcd; blue brightness 1200mcd. To control the brightness of the LED's, fixed resistors were used to fix the current whilst three variable resistors (10kohm's) was used to vary the current through the LED, and thereby controlling the brightness. The same applies to each RGB LED.

To detect the coloured light, the Photo-transistor had to be of a sensitive type. To set the brightness coming from the colour sensor it has to be calibrated. The sample data from the Sensor (10 Samples) were averaged to produced a value that is distinct enough to allow colours to be detected. The colour spectrum of the colour sensor is shown in (Figure 2.1) which represents the visible light spectrum.



.

Figure 2.1 The colour spectrum. (Liu, 2001)

The ability of the photo-transistor to detect colours is dependent on the wave length of the colours, and on this basis, the range of the wave length value, i.e. if the photo-Transistor can detect 200nm to 800nm; therefore the R G B fits in this range, as shown in (Figure 2.2), the colour spectrum.

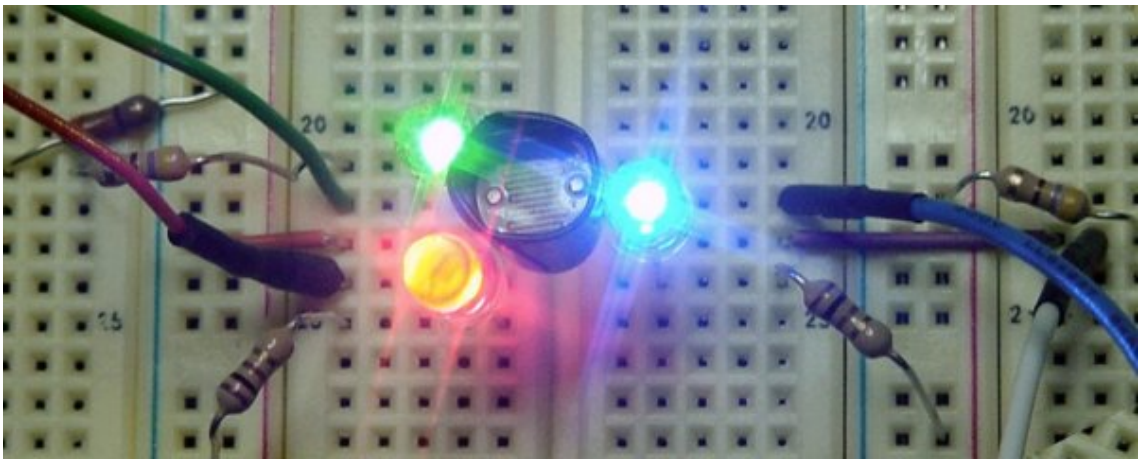


Figure 2.2 Colour sensors

As can be seen in the pseudo code, Table 2.1 all three LED's are initially "off" to start with. The next stages cause each individual LED to be switched "on" at different stages, whilst the others are off. At each stage the phototransistor data values are stored, using an array with-in the program (colv2110.c and colv2111.c appendix B). The pseudo code is an algorithm that can be used to identify an R G B colour, with further development. The use of a 50ms delay, between turn "on" the LED and record its ^{i}value is used as a technique for intensifying the brightness of the LED's, by maintaining the brightness of the LED for 1/20th of a second. This means that the LED is on and delayed for 50ms, then it takes about 10 to 20 times more current and therefore shines for about 10 to 20 times brighter for one third of a second. The exact numbers would depend on the thermal cooling rate of the LED, something that can only be determined by burning out the LED through testing (Phuyal, 2004) which is not recommended.

2.3 Photo-Transistor

The photo-transistor is more sensitive than the LDR and was used in this project. As shown, this particular sensor can detect the variation of light and hence get the best readings at 0.3cm away according to the graph below. This information can be obtained from data sheets as can be seen in (Figure 2.3). If the object is at a different distance, or change of angle of the object, then the reading will also change due to the amount of coloured light being reflected back to the sensor (photo-Transistor). In later tests the distance was found to be 2cm.

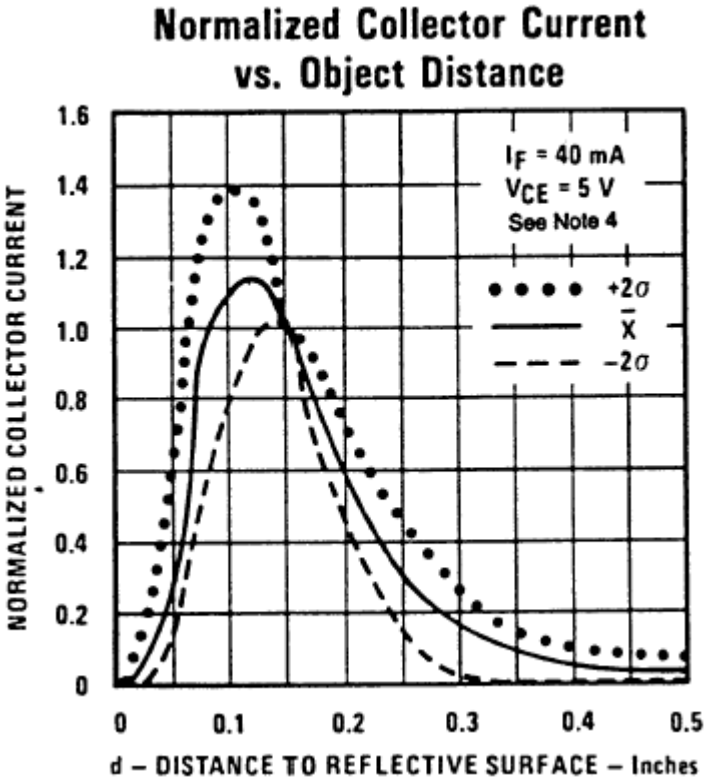


Figure 2.3 Distance vs. ambient light current of phototransistor. (Devices, 2009)

As shown in the normalized graph of collector current vs. object distance (Figure 2.3), the distance is 0.12 inches or 0.3cm. Whilst distance is a factor, the separation of the colours is also a factor. If there is no separation of the colours, the colour sensor would not be able to distinguish adjacent colours in the spectrum. Where the trace ($\bar{X}$) in Figure 2.3 represents an average of the reflected ambient light. The *dotted* line is the positive mean deviation 2 standard deviation and the *dash* line is the negative mean deviation of reflected ambient light. Essentially this is a bell curve for determining the relationship between natural light reflection and the phototransistor collector current.

2.4 Calibration

Switch “on” red LED.
Wait 50ms.
Record sensor reading or reflected LED light value.
Switch “off” red LED.
Switch “on” green LED.
Wait 50ms.
Record sensor reading or reflected light value.
Switch “off” green LED.
Switch “on” blue LED
Wait 50ms.
Record sensor reading or reflected light value.
Switch “off” blue LED.

Table 2.1 Pseudo-code

Table 2.1 lists the sequence in the program code in which the LED’s are switched “on and off”, for the purpose of controlling the colour sensor to detect R, G, and B.

Colour sensor calibration is achieved by adjusting, the variable resistors, each in turn, being switched-on and placing a coloured cards (R G B) in front of the clustered LED’s. Then adjusting the variable resistor R1, R2 and R3, each, in turn with LED being turned on respectively until the hue of colours is of a

reasonable size until it's detectable. Then the programs (colv2110.c and colv2111.c see appendix B) are run in order to sample 10 values, which are stored in an array within the program, where the data values can be accessed after the program is run, by typing the designated name of the arrays, e.g. red array, Green array and blue array allows data values to be displayed for analysis. The average of the 10 samples are determined in order to differentiate between each colour (R G and B), so that distinct average values, from the samples, can be used to differentiate between colours, RGB. When the average values match the common sensor values, so long as there is enough separation, to avoid errors by indicating the wrong colour for the wrong average value. This is the key to colour detection to overcome this problem of detection by altering the distance between sensor and the coloured cards. Figure 2.4 shows the colour sensor circuits from left to right R, G and B also the photo-transistor on the right.

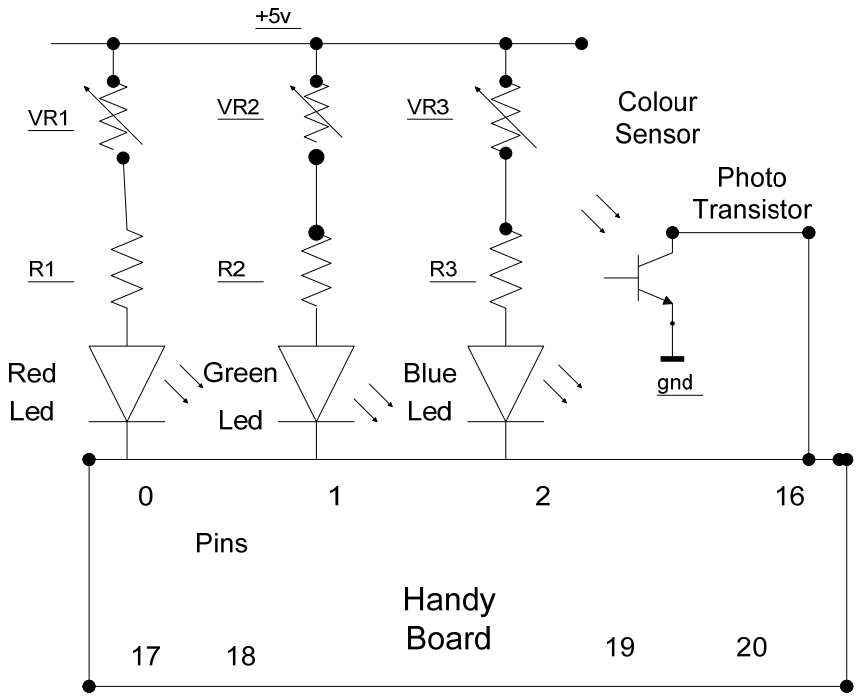


Figure 2.4 Colour sensor schematic

Pin(s) 16 is the phototransistor connection. Pin 0 is connected to the red LED, pin 1 to the green LED and finally pin 2 to the blue LED. The variable resistor(s), VR1, VR2 and VR3 have a value each of 10kΩ. Each of the fixed resistors in series is 100Ω, for each. The resistors are, from left to right R1, R2, and R3. The logic level is active low to control for each LED device. The D.C supply is +5 volts. For the accelerometer a +3.3 volts regulator is required to supply it. The implemented controlling system can be seen in the appendix A in the back of this document.

2.5 Acceleration Device

The accelerometers on the other hand, were used to measure acceleration and velocity information. In the case of the accelerometers no physical calibrations were needed, since the two of them are factory set. Most of the developments with these sensors were in software i.e. the program for control and monitoring

of the accelerometer (accor2.c) which can be found at the back of this document, in appendix B. Both the colour sensor and the accelerometers were mounted on a mobile robot, which was connected to the Handy Board, which in turn was connected to a computer (pc) as shown in Figure 2.4 and appendix A.

2.6 Theory of Operation

The ADLX330 is fully assembled. It is a 3- axis acceleration measurement system on a single monolithic integrated circuit which contains a polysilicon surface micro machined sensor and signal conditioning circuitry to implement open loop acceleration measurement architecture for each axis; an output circuit converts the analogue signal to a proportional voltage output which is connected to the Handy Board. The device is capable of measuring both positive and negative accelerations to at least +/-3g. The size of the assembled device (accelerometer) is shown in (Figure 2.6), the breakthrough board (ADXL330). (Devices, 2009) The normal operation of the device is described as follows: structure of the wafer is deflected, causing a differential capacitor, which consists of independent fixed plates attached to a moving mass. The fixed plates are (Devices, 2009) driven by 180 degrees out of phased square waves. Acceleration will deflect the beam and unbalance the differential capacitor, resulting in an output square wave whose amplitude is proportional to acceleration. Phase sensitive demodulation techniques are then used to rectify the signal and determine the direction of the acceleration (Olney, 2009). The output of the demodulator is driven by the duty cycle modulator (DCM) stage through a resistor having a value of 32k ohms. A pin for each channel is available to set the signal bandwidth of the device by adding a capacitor. By doing this, filtering improves measurement resolution and prevents aliasing.

Once low passed filtered the analogue signal is converted to a duty cycle modulated signal by a DCM stage. A sole resistor sets the period for a complete pulse which is settable in a range from 0.5ms to 10ms. Zero g acceleration causes a voltage output of 300mv/g. The acceleration signal is determined by measuring the level of the pulses from a timer/counter or by polling a loop using the Handy Board microcontroller. The analogue output voltage can be obtained by buffering the signal from Xflt and Yflt or by passing the pulse's signal through an RC filter to reconstruct the D.C. value.

Synthetic detail of a micro manufactured integrated circuit through four layers of planarized copper interconnect, down to the polysilicon (pink), wells (greyish) and substrate (green) as shown in Figure 2.5.

The acceleration device is a solid state, micro machined, polysilicon architecture integrated circuit. This can be used as a relative positioning system; using dead reckoning to find the position of the robot. As opposed to absolute position, the currently calculated position does not depend on the previous position. An example of this is Global positioning system (GPS). The acceleration device output data is integrated, once for velocity or, twice for position respectively, both are integrated with respect to sample data (Liu, 2001). A solid state acceleration device also has the advantage of being small sized, being low cost and being self contained.

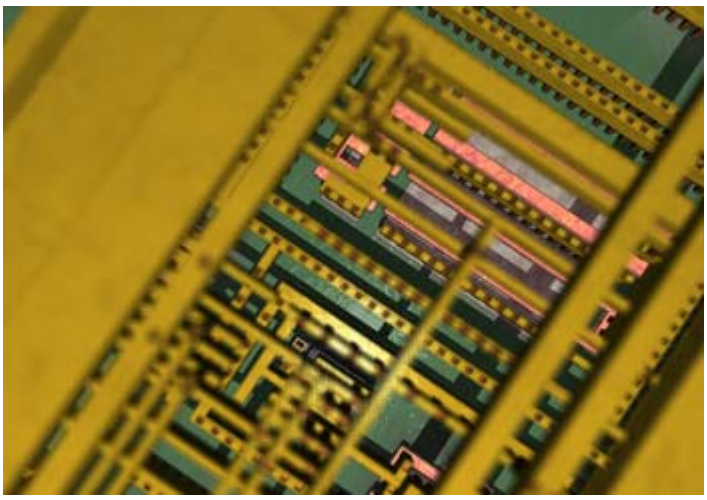


Figure 2.5 Polysilicon use example

2.7 Design of Accelerometer

One of the main reasons for using the acceleration device, for this research project, was economic and technical considerations, such as size, cost, and low power. The first acceleration device was used to produce acceleration data. This was integrated once to produce velocity, through the process of numerical integration, which was done within the program (accor2.c), which can found in appendix B. The acceleration device was used to give acceleration and velocity data by numerically integrating, respectively (Liu, 2001).

The specification of the acceleration device can be seen in (Table 2.2 a specification summary).This device, of which a functional diagram is shown in (Figure 2.7) shows the internal schematic of the system (ADXL330). This kind of acceleration device can be a viable solution as a short duration distance measuring device for a mobile robot. (Liu, 2001)

# of Axes	3
Range	+/- 3g
Sensitivity	300 mV/g
Sensitivity Accuracy (%)	±10n/a
Output Type	Analogue
Typical Bandwidth (kHz)	1.6kHz
Noise Density (°/s/rtHz)	280
Voltage Supply (V)	1.8 to 3.6
Supply Current (max)	0.18mA
Temp Range (°C)	-25 to 70°C
Package	4mm x 4mm LFCSP

Table 2.2 Summary of specification

The summary of the specification shows the key features to look at the parameter that allows the choices to be made about the accelerometer. These features are number of axes, 2 or 3 can be considered; range around 2cm to 3cm which is appropriate and voltage per gravity; output type, in this case analogue; supply current 0.14mA is acceptable; supply voltage 3.3v; bandwidth and finally size or package as shown in Figure 2.6.

Figure 2.6 +/-3g 3 - axis acceleration device ADXL330 (Devices, 2009)

Functional Block Diagram for ADXL330
Low-Cost ±3 *g* 3-Axis Accelerometer with Voltage Output

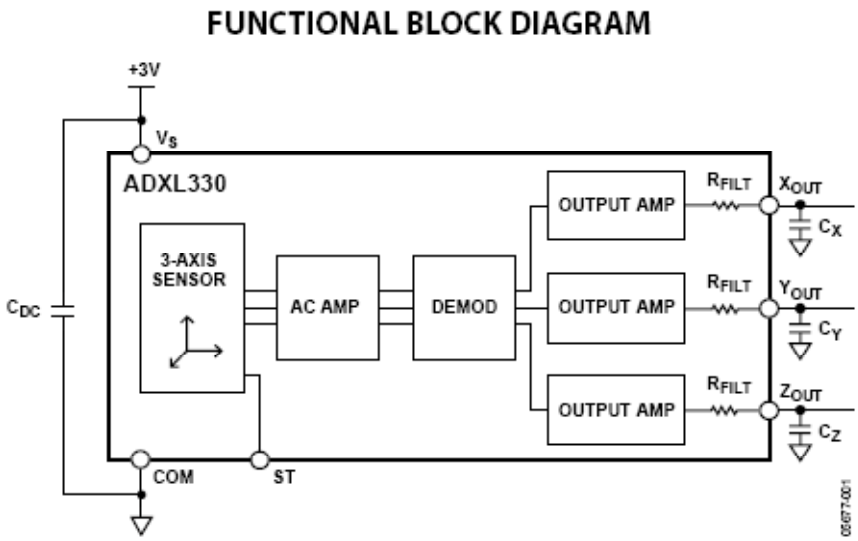


Figure 2.7 Accelerometer internal schematic (Devices, 2009)

SPECIFICATIONS

T_A = 25°C, V_S = 3 V, C_X = C_Y = C_Z = 0.1 µF, acceleration = 0 g, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		±3	±3.6		g
Nonlinearity	% of full scale		−0.3		%
Package Alignment Error			−1		Degrees
Interaxis Alignment Error			−0.1		Degrees
Cross Axis Sensitivity[1]			−1		%
SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	V_S = 3 V	270	300	330	mV/g
Sensitivity Change Due to Temperature[3]	V_S = 3 V		±0.015		%/°C
ZERO g BIAS LEVEL (RATIOMETRIC)	Each axis				
0 g Voltage at X_{OUT}, Y_{OUT}, Z_{OUT}	V_S = 3 V	1.2	1.5	1.8	V
0 g Offset vs. Temperature			±1		mg/°C
NOISE PERFORMANCE					
Noise Density X_{OUT}, Y_{OUT}			280		µg/√Hz rms
Noise Density Z_{OUT}			350		µg/√Hz rms
FREQUENCY RESPONSE[4]					
Bandwidth X_{OUT}, Y_{OUT}[5]	No external filter		1600		Hz
Bandwidth Z_{OUT}[5]	No external filter		550		Hz
R_{FILT} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF TEST[6]					
Logic Input Low			−0.6		V
Logic Input High			−2.4		V
ST Actuation Current			−60		µA
Output Change at X_{OUT}	Self test 0 to 1		−150		mV
Output Change at Y_{OUT}	Self test 0 to 1		−150		mV
Output Change at Z_{OUT}	Self test 0 to 1		−60		mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	V_S = 3 V		320		µA
Turn-On Time[7]	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		−25		+70	°C

[1] Defined as coupling between any two axes.
[2] Sensitivity is essentially ratiometric to V_S.
[3] Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.
[4] Actual frequency response controlled by user-supplied external filter capacitors (C_X, C_Y, C_Z).
[5] Bandwidth with external capacitors = 1/(2 × π × 32 kΩ × C). For C_X, C_Y = 0.003 µF, bandwidth = 1.6 kHz. For C_Z = 0.01 µF, bandwidth = 500 Hz. For C_X, C_Y, C_Z = 10 µF, bandwidth = 0.5 Hz.
[6] Self-test response changes cubically with V_S.
[7] Turn-on time is dependent on C_X, C_Y, C_Z and is approximately 160 × C_X or C_Y or C_Z + 1 ms, where C_X, C_Y, C_Z are in µF.

Table 2.3 Full accelerometer (ADXL 330) specification (Devices, 2009)

The sensor types selected for this project were a colour sensor and an acceleration device, used for determining acceleration and velocity sample data information. The colour sensor on the other hand is used to detect RGB and therefore a coloured target, from a distance of 2cm. Once the colour is detected and identified, the name of the colour is displayed on the Handy Board display.

2.8 Justification of Sensor Choice.

The main justification for using a colour sensor of this type was that it was simple to implement, inexpensive to buy and easy to construct. The robot system on the other hand, uses one accelerometer for the purpose of creating an acceleration and velocity system, which proved to be reasonably cost effective. Both the colour sensor and the accelerometer system were chosen in order to create a solid state system for the mobile robot to navigate its way around an obstacle course with minimal number of sensors. The accelerometer has three axes (x, y and z), but only (x and y) were used. The program listing for the acceleration sensor can be seen in appendix B (accor2.c).

CHAPTER 3 Colour and Accelerometer Sensor Results

3.1 Introduction

The construction of the colour sensor, involved three LED's, three fixed value resistors, three trim pots resistors and finally one photo transistor. Originally the colour light sensor was an LDR (light dependant resistor), but it was not sensitive enough, therefore the LDR was replaced by a photo-transistor, which proved to be more suitable. The circuit was built on Vero board and the inputs/outputs were connected to the Handy Board. Test programs (colv2110.c and colv2111.c see appendix B) were written in order to control certain aspects of the LED's: such as using delays in the program statements. To get extra brightness without using more power, when switching on, the LED's for (1/3) of a second, saves power in order to achieve this the sensor was calibrated in ambient light. Then each coloured card was placed in front of the sensor, at fixed distances, for a period of a few seconds, for each LED colour that matched the colour of the card. This proved successful. Another aspect of the colour sensor was that, data was required for normal light (ambient light) and the coloured light from each LED, red, green and blue. This was achieved by using arrays, for each light source and logging the data. This was used to plot data samples vs. distance, in different light environments. The colour sensor was tested in each environment, day light, dull light and fluorescent light. The data picked up from the coloured light array, Red, Green and blue, data were each averaged. This was distinct for each colour Red, Green and Blue light. The separation from each other was small but reasonable, so that no data values were too close or overlapping in the value of colour data that was

averaged. The graph of the combined colour traces and each individual trace of R G B is shown in Figure 3.1.

3.2 Colour Sensor Experiment

The test programs (colv2110.c and colv2111.c see appendix B) were run, in the following order, Red, Green and Blue, one at a time, with a 50ms between each LED being on and off. Firstly a cherry Red card was held at the fixed distance 2cm away from the clustered LED with the photo-transistor in the middle of the cluster. Secondly it was held in front of the sensor at 2cm for at least 20 seconds, so that the program ran its course and stored the data values for the Red LED. The same procedure was repeated for a Lime Green and a Navy Blue card. To access each LED's data at the console level the name of the colour and array were combined and typed into the console, e.g. Red array, Green array, Blue array, and Normal array. At each array stored ten samples were taken and were plotted on one graph using Excel. As shown in (Figure 3.1), once the RGB data values were averaged, they were then inserted in program's line statement (colv2110.c and colv2111.c see appendix B) which were the programs used for recognising RGB cards using two line statements one in each respective programs.

3.3 Colour Sensor Plots

This graph of sample data in fluorescent light vs. distance (Figure 3.1) from the colour sensor for each individual colour LED showing in the case of the RGB has been combined and shown below. Here it can be seen that the traces settle down as the sample rate levels out and the distance increases. The RGB

colours traces have very little spacing between them; this is not quite enough to be distinctive when detecting colours. The blue and the ambient light trace are close together to start with, causing the sensor to display “blue” erroneously, when the sensor is exposed to ambient light. This problem was solved by calculating the average value of the 10 samples for the (LED/phototransistor) from their data samples, as it was done for each of the three coloured LEDs.

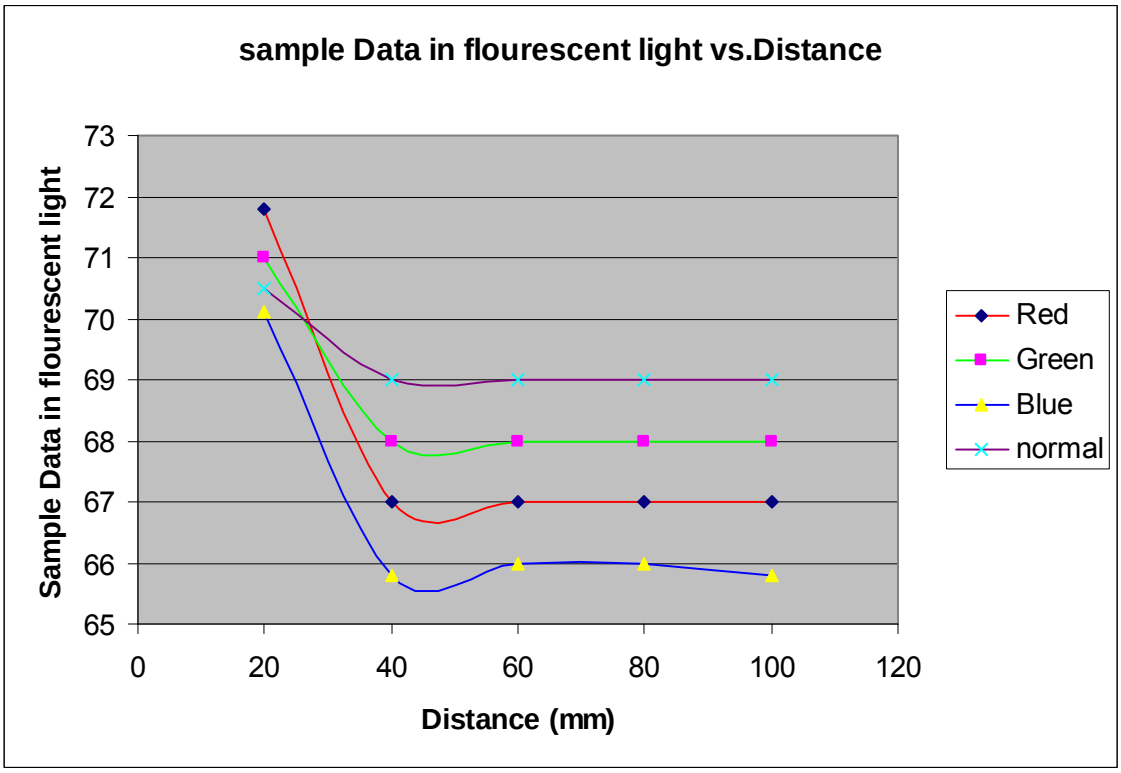


Figure 3.1 RGB colour sensor tested under fluorescent light conditions

One set of plots were produced. One set was based on the program line statement in program (colv2110.c), e.g. If ((Red=>60) && (70<=Red)) to detect colours for Red, Green and Blue. This is the heart of the colour detection system. The other line statement for the detection system is e.g. IF (Red ==70) used in program (colv2111.c). These two examples of line statement were used to develop the two sets of graphs and also tables of the outcome of colour

detection tests in three different light environments shown further on and also both sets of graphs were combined and individual traces are shown in Figure 3.1. The individual plots in Figure 3.1 for R, G and B are reproduced separately in Figures 3.2 to 3.4 for further discussion.

The data samples axis represents the data derived from the phototransistor/red LED (Figure 3.2). The sample rate represents the time taken to accumulate the data samples. The rise of the red trace is caused when the colour card, red, in this case, is brought closer to the sensor; therefore it increases in steepness towards a high value indicating that the light reflection is intense therefore close or bright or both.

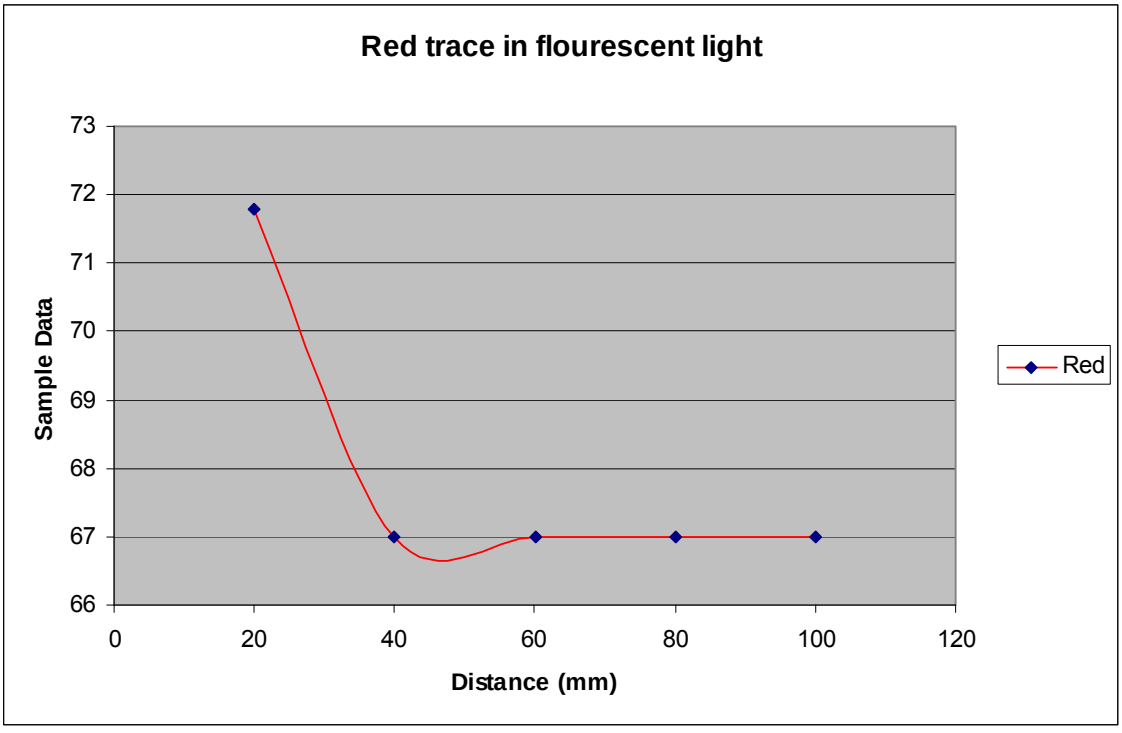


Figure 3.2 Red LED responses under fluorescent light conditions

The similar description can be said about Figure 3.2 to 3.4 above and below where Figure 3.1 is the combined traces followed by the three RGB traces which all follow the same trace pattern to the combined trace. The blue trace rises gently then levels out near the 40 mark (distance), as the coloured blue

card is moved closer to the phototransistor/blue LED (Figure 3.4). This is due to blue light being reflected from a blue card back to the sensor, hence the resultant trace. The lower ends of trace represent some distance from the sensor e.g. as the trend increases; this indicates the card is getting further away.

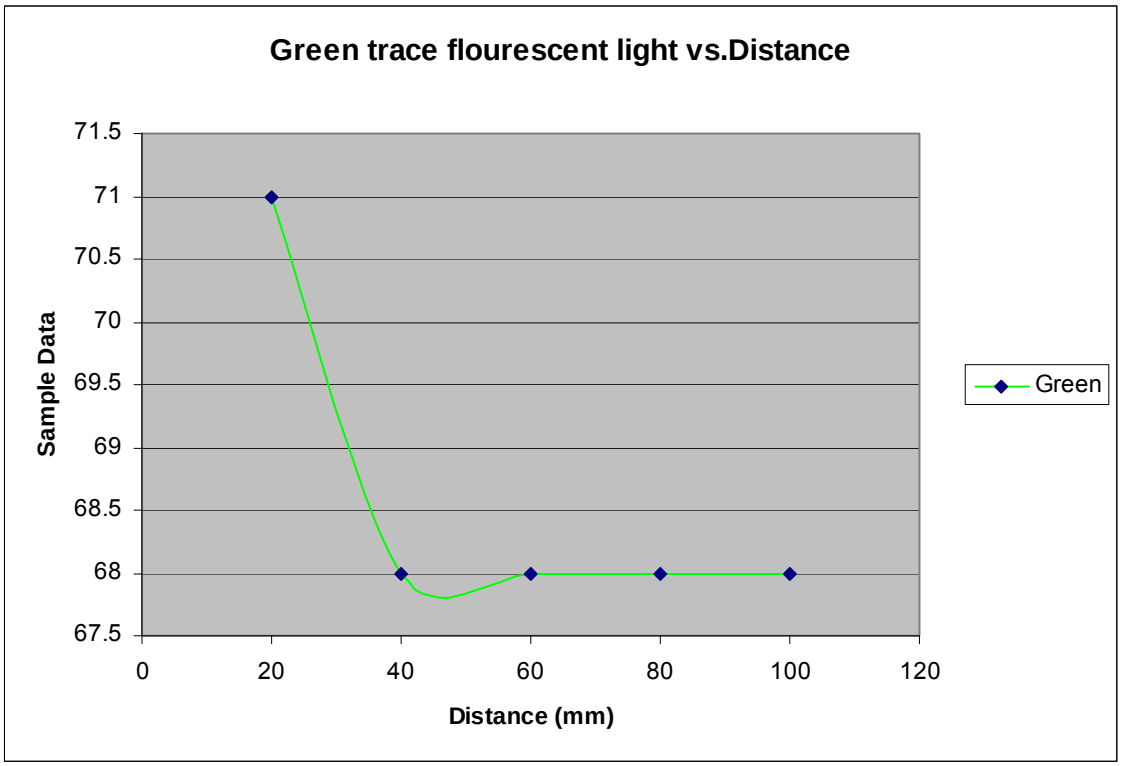


Figure 3.3 Green LED responses under fluorescent light conditions

In Figure 3.3 the pattern is the same for the other traces Red and Blue, both individually and combined plots have the same pattern as the other traces.

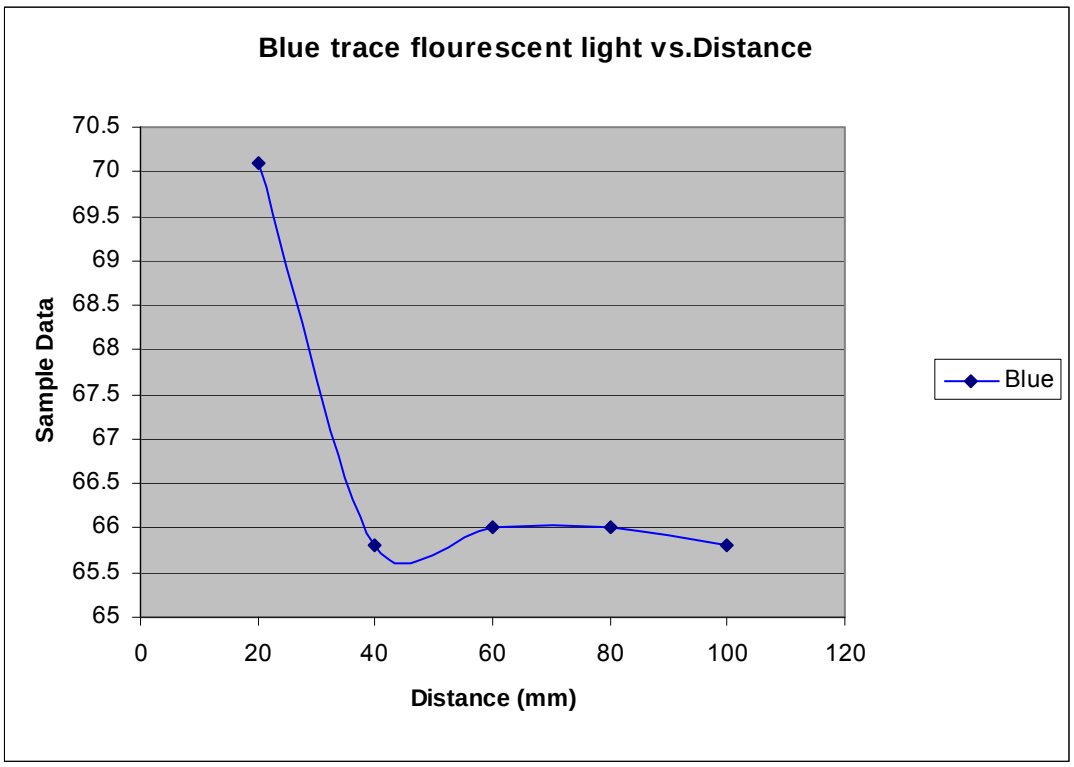


Figure 3.4 Blue LED tested under fluorescent light conditions

As for the three traces and the combined traces above were tested in order to see whether, as an example of how detection of colour is achieved, two examples are used (Red ==70) line statement in the program (colv2110.c) was better in detecting colours more effectively than using If ((Red=>60) && (Red<=70)) line statement in the program (colv2111.c).This is discussed further in this chapter and is proved to be promising, when it comes to near reliable colour detection Therefore the first line statement was the most promising in colour detection in percentage terms. The line statement e.g. (red==70) will now be referred to as the direct method and e.g. If ((red=>60) && (Red<=70)) will be referred to as the range method. It seems that the environmental light effects had little or no effect in some instances and only a slight influence on the overall outcome. But the direct method made an impact as can be seen in below in Tables 3.4 to 3.6. These tables below represent three different light

environments. Each of the LED's is (3000mcd) of brightness and the cards were held 2cm away from the coloured LED's sensor, since this was determined to be the optimum distance, which is determined by the combined colour graph traces plots. Previously the data for different distances were measured and then averaged to arrive at the value of 2cm as optimum.

3.4 Programs 2 & 3 Description for the Colour Sensor (colv2110.c & colv2111.c)

The program begins, in the same way for both program 2 and 3 where all variables are defined as integers, ready for them to be called into the body of the program, as integers. Since there are no floats in these two programs, all variables are defined as (int), short for integers. In the case of these programs some of the variables are initialised and the rest is set by the size of the arrays. These are better known collectively as declarations. The next stage void main is the start of the program when being executed by the Handy Board microprocessor (appendix A). The while (1) causes the program to be executed over and over again, so long as while is equal to one. After the while statement the three LED's as a block are initialised to zero to switch off for a few seconds defined by sleep (0.05). It is noted that Clear is switch on LED's and Set is switch off LED's.

The next stage is where Red Green and Blue are assigned to the same input source, namely Red=analog (16), this is done for all three colours, so that colour name = analog (16). This is the assignment and connection to the phototransistor input. The individual LED's are switched on, with a few seconds delay before progressing and repeating the routine to the next LED. At the end of the program loop and end of the program code the colours are assigned to

arrays to store the data from the photo transistor. When the program is running and a coloured RGB card is held, at varying distances, for test purposes, the LED colour light is reflected by the card on to the phototransistor. This causes it to input a data value to the Handy Board. A combination of this with the coloured LED generates a value from the phototransistor as expressed above. These tests were repeated 10 times and stored as numerical data. The data values, associated with these individual colours were then averaged and placed in e.g. range method If ((Red=>60) && (Red<=70)) for the purpose of determining the averaged data value discussed earlier, was tested to determine if it is either in or out of range. If the condition is met, then print red or the colour RGB and display on the LCD screen of the Handy Board. This was repeated for each RGB colours. Program 2 and 3 are the same, but for the two line statements shown below, where each line statement is included in each respective program whose names are (colv2110.c and colv2111.c) and can be found in Appendix B.

These first 3 tables represent the range method of determining colour,

Results	RED Card	GREEN Card	BLUE Card
Red	30%	0%	0%
Green	15%	65%	0%
Blue	55%	35%	100%

Table 3.1 Colour detection tests - range method for fluorescent light (intensity 6.5)

This method of testing involved using the line statement range method above, so that when the colour sensor value within the range 60 to 70, the named colour, red, would be displayed on the Handy Board display. The environmental light in this case is fluorescent light.

Results	RED Card	GREEN Card	BLUE Card
Red	100%	100%	70%
Green	0%	0%	30%
Blue	0%	0%	0%

Table 3.2 Colour detection tests - range method for daylight (intensity 8.5)

Table 3.2 above uses the range method of colour testing. Ideally all boxes would be 100% for all colours; this would mean that the colour detection would be consistently reliable. Wherever there is 100%, in the diagonal in the back slash fashion, this means that the corresponding colour is reliable for the colours that are the same. For colours that are different, the percentage value in the box for like colours represents the ideal for successful colour detection so long as all the surrounding box cells are zero (0%).

Results	RED Card	GREEN Card	BLUE Card
Red	85%	0%	0%
Green	0%	15%	0%
Blue	15%	85%	100%

Table 3.3 Colour detection tests - range method for semi-darkness (intensity 2.0)

This marks the end of the percentage tables of colour detection tests, Figures 3.1 to 3.6, represent the test's involving the direct method of colour detection. The tables comprise of RGB colours against RGB coloured cards. When a red LED is "on" the red card is placed in front of it i.e. the colour sensor the display should indicate red, but occasionally the display would indicate green or blue. Whatever the colour indicated on the display it was recorded and the number of times different colours occur, they were totalled and averaged from 20 tests for each colour. As a result the percentage per colour was calculated.

Results	RED Card	GREEN Card	BLUE Card
Red	30%	50%	15%
Green	20%	25%	20%
Blue	50%	25%	65%

Table 3.4 Colour detection tests - direct method for fluorescent light (intensity 6.5)

The same tests were done using the direct method for the colour detection. The direct method above, works by printing red or any of RGB colours, whenever the value e.g. 70 is detected. Compared to the range method, there is marked difference, since where there is 0% in the direct method in some boxes; this is due to colours not matching when tested for like colours therefore this test was unreliable (Table 3.4).

Results	RED Card	GREEN Card	BLUE Card
Red	95%	40%	25%
Green	5%	50%	10%
Blue	0%	10%	65%

Table 3.5 Colour detection tests - direct method for daylight (intensity 8.5)

The table above use the direct method of colour testing. When any of the column percentage value is tallied, they are 100% in value. The zero 0% means that for the combination of blue LED light and the coloured red card produces no response, which is also good for this combination, since this outcome should have no response. This test was done in day light.

Results	RED Card	GREEN Card	BLUE Card
Red	35%	40%	5%
Green	35%	30%	0%
Blue	30%	30%	95%

Table 3.6 Colour detection tests - direct method semi-darkness (intensity 2.5)

This table uses the direct method of colour detection tests. Where each colour matches itself, therefore if the colour in the table is 100%, this is good since, that colour and only that colour, is true every time therefore reliable colour detection. This test was done in dull, low light.

3.5 Accelerometer Type

The main aim of using the accelerometer was to measure acceleration and velocity whilst the velocity and distance is derived, from the sensor data. The choice of type of accelerometer was made, based on size and sensitivity. To get a signal response from the device, it must respond to a low “g” force, with a low frequency response and proportional to voltage when measuring acceleration and velocity data. For zero “g” the output for the device should be zero also, but in actual fact it is 300mv/g, which represents the device sensitivity. The cost was also a factor, when it came to the specifications. The specification for this is shown in (Table 2.2) previously. The physical size can be seen in (Figure 2.6).

3.6 Accelerometer as a System

This sensor, as the name suggests measures acceleration, but its use in this case estimates velocity which is derived from acceleration data. It is this data generated from the accelerometer, which was plotted on a graph, using Excel. Before this, though, a test program was written, for the purpose of testing it experimentally. From the data samples generated by (Xa, Ya):

$$velocity = initial\ velocity + h\sum_{k=1}^{n}\ (Xa(k) - offset) \qquad \text{equation 3.1}$$

where

h is the sample interval

Calculates the velocity for each axis, e.g. (Xa, Ya) stored as data values in an array, each containing 100 samples (0 to 99). Each sample was sampled every 0.1 seconds, as shown equation 3.1, for Xa and Ya. From the initial results, which were plotted, the offset was determined and was then used as the offset value in equation 3.1. Before the data values are processed, raw data

(acceleration) was plotted, as shown in Figures 3.5 to 3.6. The resultant plot shows rapid activity traces of acceleration on each graph. Once processed using equation 3.1 the resultant processed data, comes out as a velocity data from which velocity can be plotted. All the plots have bursting peaks of noise, starting at values that represent the offset. This is different for each plot, including velocity, which represents the acceleration and velocity of the robot from the pair of axis (Xa, sample) and (Ya, sample). The plots represent the response of the accelerometer. Acceleration comes directly from the device and then the data was used to determine velocity by numerically integrating the data. The acceleration plots are shown in this chapter 3.

3.7 Acceleration and Velocity Plots

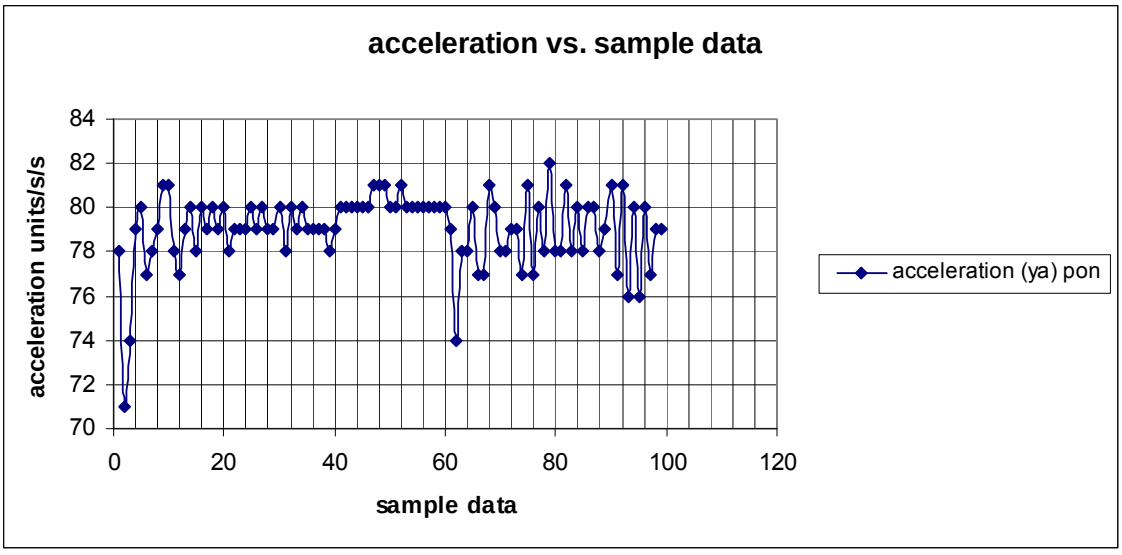


Figure 3.5 Data plot of acceleration with power on (Ya)

Figure 3.5 and 3.6 are raw data plots of acceleration using the ADXL330, accelerometer. This is a device which was used to acquire raw data for acceleration. The accelerometer is a tri-axis device, but only two of the axes were used (x and y). The device was placed on the central part of the mobile robot, and connected to the Handy Board. The mobile robot was manually pushed on a flat surface of approximately half a metre, which ensured that the device would produce raw acceleration data. This in turn was stored in the program array. To retrieve the data, the word Xaarray or Yaarray was typed into the console, for the array printout on the console screen. The data was then plotted on, using Excel as shown in Figure 3.6. The plot shows a trend where the acceleration generates a rapid response, sometime during motion since it is slowing down. The data from the acceleration data was used in the program to integrate the data once. Then (Xa) were plotted using Excel with respect to samples. (Sample, Xa). The same applies to (Ya) the other axis (sample, Ya) as shown in Figure 3.8 and 3.10 respectively.

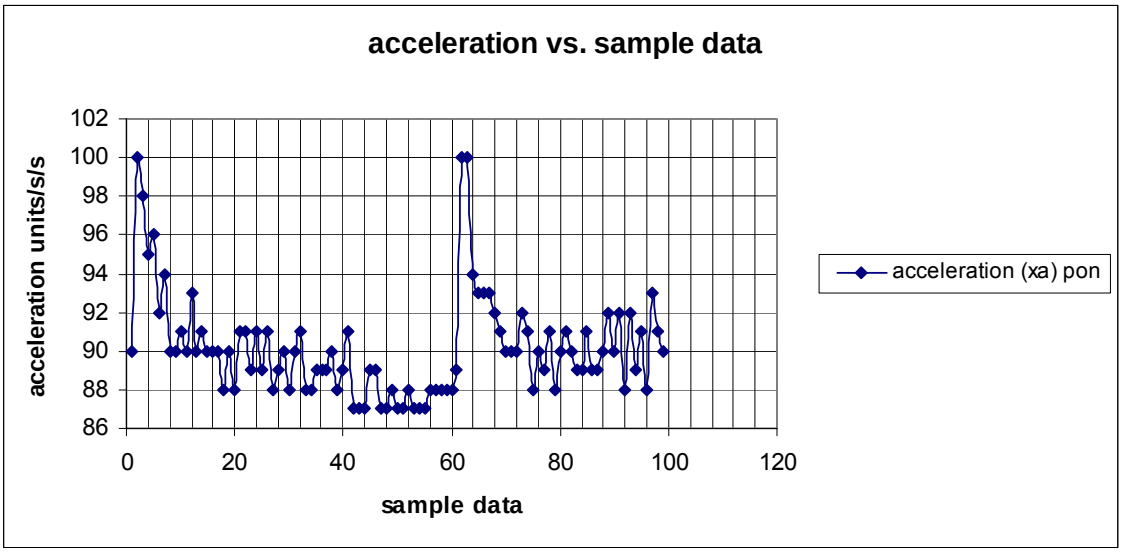


Figure 3.6 Data plot of acceleration on with power on (Xa)

Figure 3.7 reflects the velocity plot data derived from the acceleration data by numerically integrating once. From this plot the velocity is rising, reflecting increasing speed or velocity, until it stops and falls gradually to zero to a dead stop for the mobile robot. The offset was calculated by taking the average of the data values for each axis, i.e. the value of each axes offset were Xa =130.17, Ya =128.7, these were averaged for the raw data (acceleration) and velocity that was collated from the accelerometer and used to calculate velocity by the program. As shown in Figure 3.7.

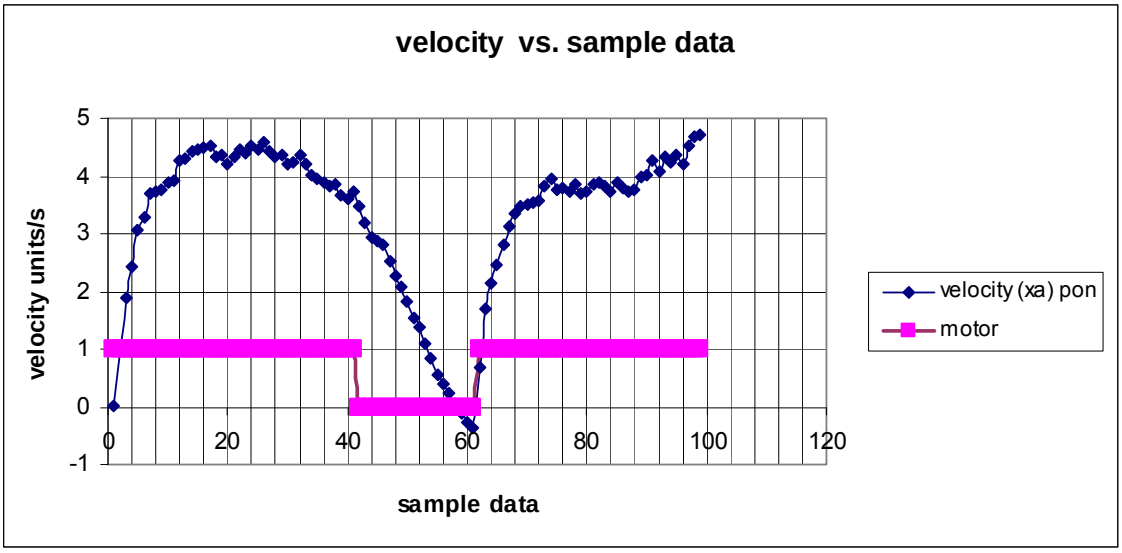


Figure 3.7 Data plot of velocity with power on (Xa)

Figure 3.7 shows the motor trace signal, in pink, superimposed on the velocity trace representing motor drive with the motor being powered “on” for motion. Further forward and above there are three of the traces that were produced with the motor power “off” and motion was done by hand powered movement in order to be compared with the power “on” version, for both acceleration and velocity as shown section 3.7. In Figure 3.8 again, the velocity plot shows data (velocity) from the accelerometer data, but on this occasion this is (sample data, Ya) axes. This was also integrated, once, numerically to produce velocity for

(Ya) from the two axis of the accelerometer, e.g. (Xa, Ya), both plotted with respect to sample data. Figure 3.8 represents the trace of velocity with the motor powered “on”.

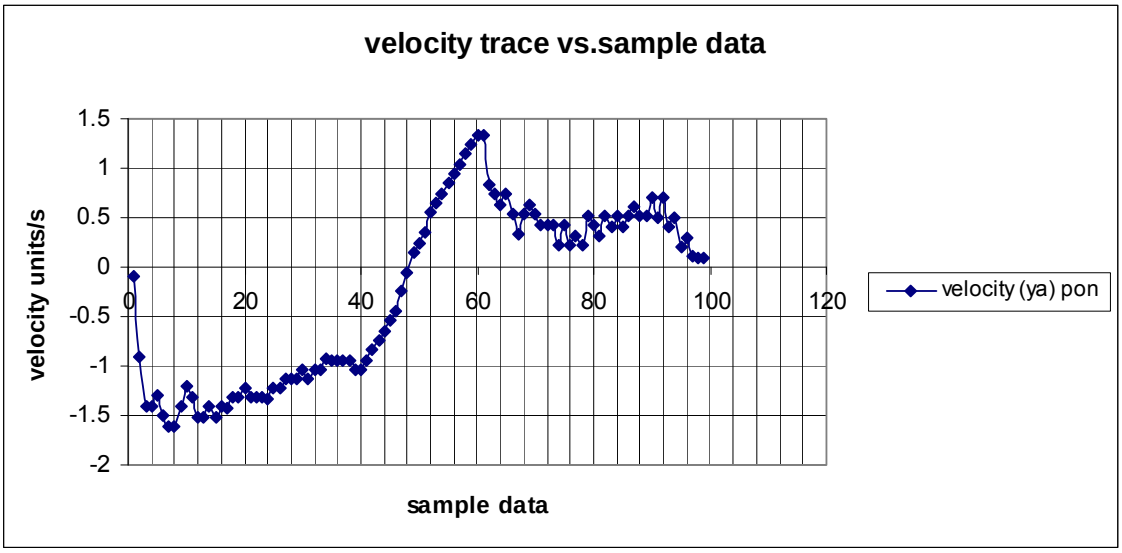


Figure 3.8 Data plot of Velocity with power on (Ya)

3.8 Acceleration with the Power Off Plots

The acceleration shown in Figure 3.9 shows rapid activity or fluctuations and one large spike as it comes to rest. Again the velocity was derived from the accelerometer raw data(acceleration) and numerically integrated to produce the velocity plot in Figure 3.10, but this time the (Xa) axis is plotted with respect to sample data and the velocity out-come is negative, indicating that (Xa) in this case is deceleration which has produced a negative velocity. This can be explained as travelling in the opposite direction.

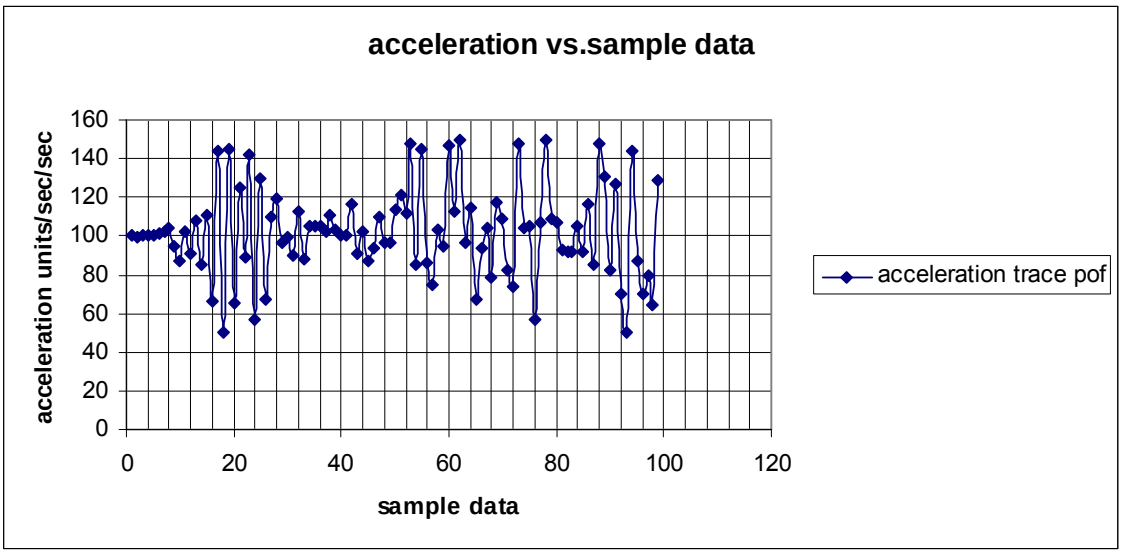


Figure 3.9 Data plot of acceleration (Xa) with the power off

In Figure 3.9 the raw data produced, from the accelerometer, shows the rapid fluctuation at the beginning and towards the end, just before it comes to a full stop. From the data the raw data (acceleration) was integrated once, numerically, in order to derive the velocity data for (Xa and Ya), with respect to sample as shown in Figure 3.5 and 3.6 the plot respectively. The peaks on the plots for velocity represent the level of “g’s” being exerted, when they are positive/negative going either side of the offset value. The offset value for Figure 3.9, from the trace in this case is approximately (100).

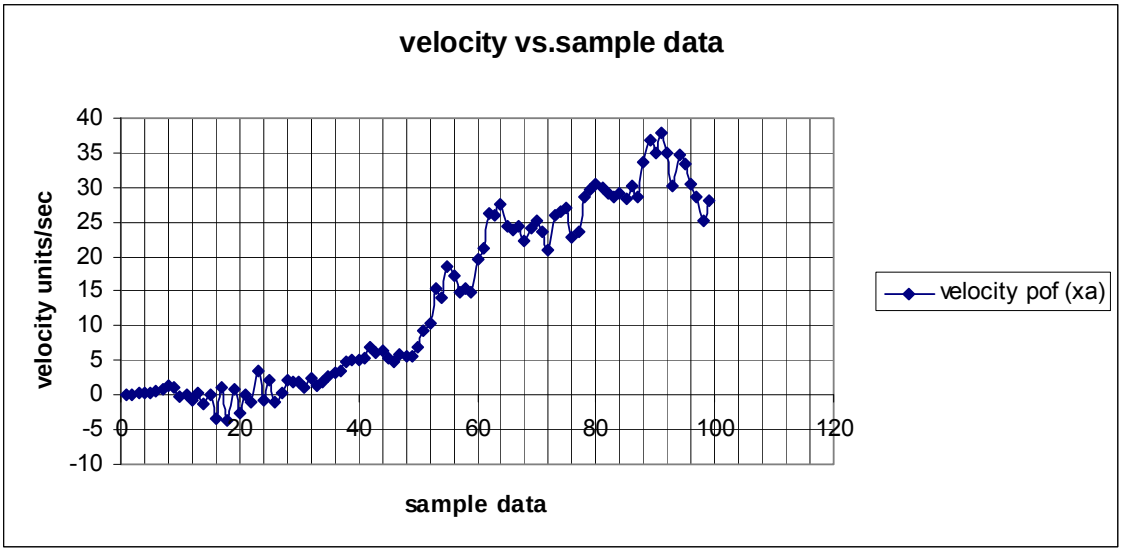


Figure 3.10 Data plot of velocity with the power off (Xa)

In this Figure 3.10 the plot is of a velocity data that was derived from the data shown in Figure 3.9. The acceleration of the mobile robot data was integrated to produce velocity. Figure 3.10 and Figures 3.7, 3.8 were all derived from accelerometer, which was mounted on the motor block of the mobile robot. There are six sample plots, three of which are acceleration and three are of velocity plots Figures 3.7/ 3.8/3.10. The other three are of velocity plots Figures 3.5 and 3.6. /3.9 each plot is one axis verses sample (sample, Xa). This trace was derived by accelerating the mobile robot by hand to simulate motor power and comparing it with the motor powered mobile robot to make a comparison between the motor being powered “off” and “on”.

3.9 Program 1 Description for the Accelerometer (accor2.c)

In this program, the variables are declared as (int) integers and floats for the array’s as well as initialisation of other variables. This program is partly, for test purposes as well as control of the mobile robot, involving, the accelerometer for

acceleration, raw data, which is used to calculate velocity. The next stage of the program, starts with void main (), where the offset is calculated by averaging the data being inputted by the accelerometer, for axes (Xa vs. samples) and (Ya vs. samples), for velocity. Both the raw data (acceleration) and velocity were stored in the array's having 100 (0 to 99) locations. The next step is when the motors are powered "on" as well as displaying or printing a cue to start then stop. The motor is switch on for a count of i==40 and off i==60 in the line statement. As shown in Figure 3.7, the pink trace represents the motor drive. The same process is repeated, but without the motor line statement for switching "on" the motor is disabled (off). In the situation of testing by hand, visual cue's are used, so that the mobile robot can be pushed by hand to simulate the motor, between the start and stop cue,(i==40 and i==60) respectively. The data for these tests was used to plot acceleration and velocity using (Xa, sample) and (Ya, sample) in both respective case's. Thus Figure 3.5 and 3.6 are acceleration plots and Figure 3.7 is velocity. All three figures were done with the motor "on". The other plots 3.8 and 3.9 are velocity and acceleration, whilst Figure 3.10 is velocity and were achieved by using hand power, motor "off". When this program is run, it then executes the loops and then stops. The data is retrieved from the arrays (for both acceleration and velocity) by using xaraw and, yaraw also xaarray and yaarray, by typing these names into the computer console. The data was then plotted using the Excel spread sheet.

Chapter 4 Discussion and Analysis

4.1 Colour

The result shown in chapter 3 for the colour sensor represents all the data that was collated as a result of the tests performed. From the combined RGB graphs (Figure 3.1), it can be seen that the ambient light trace follows the same trace pattern. The graph shows sample data in fluorescent light vs. distance. From a distance of 2cm,the trace begins high and falls gradually to a low point, before levelling out, indicating that the coloured cards of RGB causes the fall due to distance being increased. Therefore the light intensity reflected back to the photo sensor is diminishing, thus, the sample data values fall. The ambient light (fluorescent light) environment was bright, but was not intense, therefore had negligible effect. Since very little of the fluorescent light was reflected back to the photo sensor, this means the correct response, as far as the lighting environment is concerned, is due to the fact that if the fluorescent light had any effect, the sensor would not distinctly detect the RGB light. But if the fluorescent light had any effect it would blank out any other light source including the RGB light and drive the photo- transistor into saturation and therefore no detection.

4.2 Tables of Colour Detection Response

In this section a discussion of pro's and con's will be considered. The six tables will be compared and discussed in pairs such as putting the two fluorescent light tables together to determine the better of the two. The other light environments were, daylight and semi darkness, since these were the conditions in which the colour detection tests were carried out. As well as this another factor was the line statement in the two programs (colv2110.c and

colv2111.c) which are identical, except for the line statement. These statements will be referred to as the direct and range methods e.g.

Direct method

```
If (red==70)
```

Range method

```
If ((red=>60) && (red<=70))
```

These two factors determined why the six tables were divided, so that three of the tables are direct and the other three were of the range methods in the same three environments. These were to be paired off, so that comparisons could be made. Therefore the fluorescent light environment and the direct method will be discussed first. In Table 3.1, range method was used and the colour cards RGB and the LED's RGB for the same colours were 100% in some places in the tables. When this occurs, it means 100% detection of colours of the same colour. In other places where there was 0%, according to where it is in the table are both correct. If 0% or 100%, depending on where these two percentages are, means that they are correct, i.e. if the following criterion is met for the ideal, then this would be the ideal as shown in Table 4.1.

A colour detection system was designed and tested, using different lighting environments, such as day light, semi-darkness and fluorescent light. The key to the determination of coloured cards (RGB) is the line statement such as the range method if ((Red=>60) && (Red<=70)). This line statement is interpreted as if Red is equal to or greater than the value 60 and Red is less than equal to the value 70, then prints the colour name, in this case Red. If any value generated by the phototransistor, in the range from 60 to 70. Therefore they are 60, 61, 62,..., 70. The other method is the direct method, such as if(Red==70).

This method interprets the line statement, as if Red is equal to the value, in this case 70 exactly. To print the name of the colour would be the response.

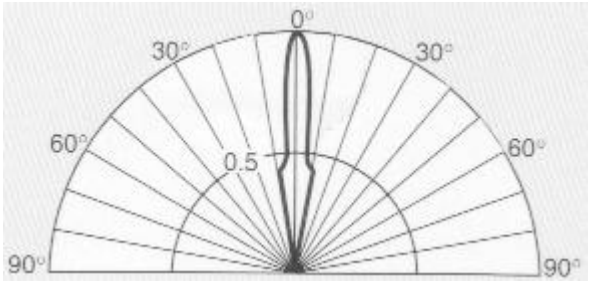


Figure 4.1 Laser-LED beam plot

"This radiation diagram shows the output of a blue LED with a water-clear case (Photron PL-BA31). Most of the light is shooting straight out the front of the package." (LASER–LED, 2009)

This represents what is required to prevent radiation scatter or light scatter. The radiation diagram shows how the beam should be focused when being emitted from the LEDs, so that the beam can be reflected back to the phototransistor and can detect RGB successfully as suggested in Section 5.2.

Results	RED Card	GREEN Card	BLUE Card
Red	100%	0%	0%
Green	0%	100%	0%
Blue	0%	0%	100%

Table 4.1 The ideal outcome for colour detection in day light

On the other hand in Table 3.4, the percentages of the direct method are all above the table boxes, which indicate that any of the colours will indicate the wrong colour, where they are a mismatch between colours in some combination instead of RGB, being detected 100% for colours that are alike. The fluorescent light again was found to be negligible in effect. The next two tables (Tables 3.2 and 3.5) can be addressed in the same fashion as the previous two tables (3.1 and 3.4). The light environment was different as the detection tests were tested in day light.

The methods, direct and range, tests showed that the direct method performed better than the range method. This can be seen when compared with Table 3.2 and 3.5. Where there is a mismatch in colours the boxes should be 0% as in Table 4.1. The final pair of tables (Tables 3.3 and 3.6) was tested in a semi-darkness environment. The reasoning can be attributed to the other two pairs of

tables. But again the direct method (Figure 3.6) proved more reliable than the range method (Figure 3.3) since blue = blue card =100%. All the other boxes in Figure 3.6 have various percentages that indicate that, as well as the correct colour being detected, other colours are detected too, and therefore different percentages occur e.g. 15% or 95% etc.

4.3 Accelerometer

The accelerometer chosen was the ADXL330, a three axes device(x, y and z). But in this case, only two of the axes were required. These were (x and y).The purpose of using the accelerometer was, that the output voltage is proportional to the amount of “g’s” the device can produce. The total “g’s” for this device was +/-3g’s, as it can be seen in Table 2.2 the summary of specification. In order to produce accurate data from the accelerometer, for acceleration and velocity the Handy Board had to convert the signal from the device which was supplied from a 3.3 volt regulator. The ADC (Analogue to Digital Converter) was 0-5volts. Since the ADC used has an 8 bit digital output, its accuracy was limited by its resolution. Therefore the following can be seen in terms of resolution percentages:

8 bit ADC Resolution:

$$1/ (2^8\text{-}1)*100 = (1/255)*100 = 0.392\% \qquad \text{equation 4.1}$$

12 bit ADC Resolution

$$1/(2^{12}\text{ -}1)*100 = (1/4095)*100 = 0.0244\% \qquad \text{equation 4.2}$$

From the accelerometer the range is 0 to 3.3v which represents -3g to +3g.The operating voltage range for the Analogue to Digital Converter is 0 to

5v.The analogue to Digital Converter is 8 bits wide, having a range of 0 to 255.The maximum range of the ADC in relation to the supply voltage of the accelerometer and the ADC will therefore be (3.3/5)*255 = 168.3. This represents the maximum range of values that are possible digitally. Therefore -3 to +3 is represented by 0 to 168 digital values each digital value is therefore 6g/168.3 =0.03565. The main issue for this application 3g is a very high acceleration that is rarely achieved with the robot used. A lower range sensor would have given better resolution.

It can be seen that when the resolution for an 8bit ADC and a 12 bit ADC, the difference is greatly improved, with 12 bit ADC the resolution would be more improved accuracy. Since the percentage value for the 12 bit ADC is 16 times more accurate than the 8 bit ADC, for the signal to swing between plus and minus when the acceleration data is being generated or measured there must be a point where it can do this successfully. This point is known as the offset. The offset point was determined in the program to compensate for the acceleration and therefore velocity swings without clipping the signal. To calculate the offset meant averaging the acceleration data and inserting the value into the integral equation 3.1, the offset in the equation. But later a line statement was used to determine the offset for different conditions when calculation takes place within the program (accor2.c). Without the offset the curve would be clipped in either of the plus or minus direction. The accelerometer output is +3g’s to -3g’s which means to do the measurement successfully the offset point must be in-between the two points and since the voltage supply is 3.3volts, the measurement will be proportional to the +/-3g’s swing.

4.4 Acceleration and Velocity Graphs

The data measured from the accelerometer, for acceleration and velocity were used to plot traces of (acceleration vs. sample data) and (velocity vs. sample data). There were six graphs in total, as can be seen in Figures 3.5 to 3.10. These figures are three acceleration graphs in (Figure 3.5, 3.6 and 3.9).The other three are velocity graphs in Figure 3.7, 3.8 and 3.10. The acceleration plots or raw data plots were integrated, using equation 3.1 to produce velocity and the curves. In the case of Figure 3.7 a plot of the motor trace (in pink) is shown on the velocity plot. This shows that the motor is “on” from and interval, characterised by a fall at 40 to 60 point on the plot as shown in Figure 3.7, motion of the mobile robot tying in with decrease in velocity then rising for the next cycle for acceleration/deceleration. The acceleration data from Figure 3.5 was used to derive the velocity data for Figure 3.8 and Figure 3.6 was used to derive the velocity, through integration, resulting in Figure 3.7. All figures of acceleration and velocity which were measured by the motors for the mobile robot, were powered “on” under program control. However Figure 3.9, acceleration and Figure 3.10, velocity were tested by physically pushing the mobile robot by hand to simulating the motor being “on” but in fact there was no power to the motor. These two plots of acceleration and velocity performed just as well as the powered “on” but the small and large spikes on the curves were caused by noise and vibrations, since the accelerometer was taped to the body of the mobile robot.

Chapter 5 Conclusion and Further Work

5.1 Conclusion

It has been stated and shown earlier that, in the case of the lighting environment, (Figure 3.2) daylight proved reasonable. But in the case of the methods (Figure 3.2) the direct method proved the most promising when it comes to successful 100% colour detection in the day light environment of RGB. As a result of further investigations, of the sensor, the principle of colour detection is feasible, without a too high a cost. These colour detectors are even more accurate than the human eye, also they can detect up to 22 different colours faster than the human eye, and a scan distance of 4cm, double than what was achieved in this project. It can also detect colours at high speeds, such as scanning colours on a conveyer belt. (SAIMC, 2009)

In order to get a more accurate measurement of acceleration and in turn an accurate velocity measurement, since velocity is derived from acceleration by integration, the resolution of the ADC (Analogue to Digital Converter) hardware device has to be improved, from 8 bits to 12 bits. This will improve the accuracy by up to 16 times for the 12 bit ADC as discussed in chapter 4.

The offset, in equation 3.1, was used in the program, (accor2) in the back of this document. The purpose of the offset is to keep a balance between the positive and negative going data or swing, e.g. as the acceleration swings between positive and negative, there is a mid point, which is maintained to keep a balance between the two polarities this midpoint and the values that keeps it in the middle is known as the offset. Velocity is achieved by numerically

integrating the acceleration data as well as the offset values, which was part of the equation, for determining velocity.

The noise generated by acceleration measurement, is caused by the ADC having poor resolution due to the number of bits used by the ADC and as part of this process. When the acceleration data is numerically integrated, the noise present in acceleration signal is doubled due to the integration. The way to improve this is to use an ADC with a larger number of bits such as a 12 bit to improve a good resolution. As a result, better resolution and accuracy would also be improved.

5.2 Further Work

The need to improve the sensors is borne out by the fact that these existing sensors have limitations that can be improved. In the case of the colour sensor the distance it requires is 2cm from experimentation, to detect a coloured object in front of it. The proposal is to improve this by using laser type LED's and a sensitive photo device in order to lengthen the distance over which the mobile robot detects the coloured object for the purpose of collision avoidance. The situation, as it stands, is that the mobile robot has to be up close, before it can respond. The results of this are that the mobile robot would collide with the object, due to its momentum. Therefore at a reasonable distance, when it detects light and steers clear, without the need to reverse or collide with an object. In a recent investigation (LASER-LED, 2009) it is feasible to use laser diodes, or LEDs to achieve the goal of collision avoidance as just outlined in the discussion before.

The limitations of the colour sensor are just as applicable to the accelerometer, with respect to acceleration and velocity as stated earlier. The two parameters are not readily available, since data has to be collated and processed using equation 3.1 to determine, acceleration and velocity data. Therefore, to overcome this, so that these two parameters are displayed in real time, it can be achieved by using hardware/software combined to implement, the two parameters so that the data is processed in real time using the existing accelerometer used in this project.

References

CAZO, R. M. (2008) Six Degree Freedom Optical Fiber Accelerometer. *1ST WORKSHOP ON SPECIALTY OPTICAL FIBERS AND THEIR APPLICATIONS,* 1055**,** 125-128.

CORSARO, R. D. (1996) Smart actuator with integrated co-formed accelerometer. *INDUSTRIAL AND COMMERCIAL APPLICATIONS OF SMART STRUCTURES TECHNOLOGIES - SMART STRUCTURES AND MATERIALS 1996,* 2721**,** 436-442.

CORSARO, R. D. (1997) Integrated smart actuator containing a monolithic co-formed accelerometer. *INDUSTRIAL AND COMMERCIAL APPLICATIONS OF SMART STRUCTURES TECHNOLOGIES: SMART STRUCTURES AND MATERIALS 1997,* 3044**,** 397-405.

DEMARS, K. J. (2008) Precision descent navigation for landing at the moon. *ASTRODYNAMICS 2007, PTS I-III,* 129**,** 1027-1050.

DEVICES A.D.(2009) Accelerometer ADXL 330. http://www.analog.com/static/imported-files/data_sheets/ADXL330.pdf

DING, M. L. (2005) Design of an intelligent sensor. *ISTM/2005: 6th International Symposium on Test and Measurement, Vols 1-9, Conference Proceedings***,** 3546-3548.

GAJJAR, B. (2002) Model based roll axis control of a terrain adapting Mars rover. *ROMOCO'02: PROCEEDINGS OF THE THIRD INTERNATIONAL WORKSHOP ON ROBOT MOTION AND CONTROL***,** 237-242.

GUL, F. (2005) Correction technique for velocity and position error of inertial navigation system by celestial observations. *IEEE: 2005 International Conference on Emerging Technologies, Proceedings***,** 7-12.

HAAPASAARI, M. (1998) Position measurement system for virtual reality and mobile multimedia. *PROCEEDINGS OF THE ICMA'98 - ADVANCED MECHATRONICS: FIRST-TIME-RIGHT, VOLS 1 AND 2***,** 557-568.

LASER-LEDS (2009) http://www.horrorseek.com/home/halloween/wolfstone/Lighting/litled_LightEmittingDiodes.html#LASER-LEDs

LIU, H. H. S. (2001) Accelerometer for mobile robot positioning. *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS,* 37**,** 812-819.

MARTIN, F. G. (2001) *Robotic explorations*, Prentice Hall International.

ONELY, B., DAVE AND SWANSON BOB ARKELL (2009) selecting the for yourapplication.http://www.archive.evaluationengineering.com/archive/articles/0501data.htm

PHUYAL, B. (2004) An experiment for a 2-D and 3-D GPS/INS configuration for land vehicle applications. *PLANS 2004: POSITION LOCATION AND NAVIGATION SYMPOSIUM*, 148-152.

SCHOEBERLEIN, H. C. (1998) Coherent noise cancellation of motion contamination in near-surface velocity measurements. OCEAN COMMUNITY Society of Robots http://www.societyofrobots.com/sensors_color.shtml CONFERENCE'98: CELEBRATING 1998 INTERNATIONAL YEAR OF THE OCEAN, PROCEEDINGS VOLS 1 AND 2, 1088-1092.

SAIMC, T. O. J. O. T. (2009) SA Instrumentation & Control http://www.instrumentation.co.za/news.aspx?pklNewsId=8274&pklCategoryID=69.The official journal of samic.issued October 2002.

WIEGERT, R. (2002) Magnetic anomaly sensing system for mine countermeasures using high mobility autonomous sensing platforms. *OCEANS 2002 MTS/IEEE CONFERENCE & EXHIBITION, VOLS 1-4, CONFERENCE PROCEEDINGS*, 937-944.LASER-LED

SEATTLE, http:www.seattlerobotics.org/encoder/apr97sonar.html

Appendix A The Handy Board

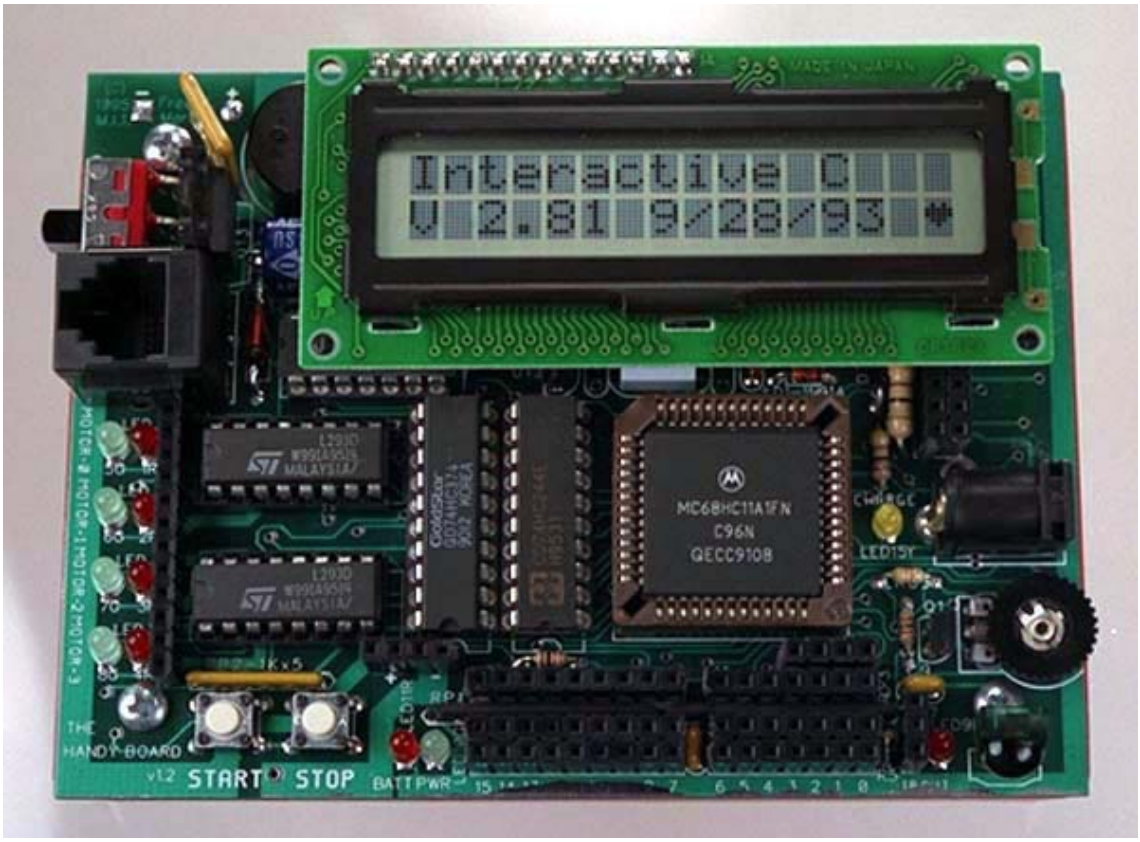


Figure A.1 Handy Board with LCD display

The Handy Board is based on the 52-pin Motorola MC68HC11 processor, and includes 32K of battery-backed static RAM, four outputs for DC motors, a connector system that allows active sensors to be individually plugged into the board, an LCD screen, and an integrated, rechargeable battery pack. This design is ideal for experimental robotics project, but the Handy Board can serve any number of embedded control applications.

Appendix B Programs

The programs for testing and control of the mobile robot

Program 1 interactive “C” language/accelerometer (accor2.c)

```
/*by Euclid Mills 14-09-2009*/
int xa,ya,xb,yb;
int xaraw[100];  /*acceleration data */
int

/*acceleration and velocity data test Program accor2.c */
yaraw[100];
int xbraw[100];
int ybraw[100];
float xaarray[100];
float yaarray[100];  /* velocity data */
float xbarray[100];
float ybarray[100];
float sum1,sum2;
float sum3,sum4;
float offset1,offset2;
float offset3,offset4;
int sumx=0,sumy=0;
int sumv=0,sumw=0;
int i=0;

/*The 2 pairs of axes represent forward velocity*/
/* and positional data, global declarations*/

void main()
/* no values are returned during any execution cycle*/
{
   /* First loop to find accelerometer offset while robot
      is stationary */
   for(i=0;i<99;i++)
   {
      sumx=sumx+analog(17);
      sumy=sumy+analog(18);
      sumv=sumv+analog(19);
      sumw=sumw+analog(20);
      sleep(0.09);
   }
   offset1=(float)sumx/99.0;
   offset2=(float)sumy/99.0;
   offset3=(float)sumv/99.0;
   offset4=(float)sumw/99.0;

   /* Main stage where robot moves */
   fd(1);fd(3);
   printf("start accel now\n",i==40);sleep(0.5);
   for(i=0;i<99;i++)
   {
      /* switch of motor near middle */
      if (i==40)
      {
         off(1);
```

```
         off(3);
      }
      /* switch on motor after short delay */
      if (i==60)
      {
         fd(1);
         fd(3);
      }
      printf("stop accel now \n",i==60);
      xa=analog(17);
      ya=analog(18);
      xaraw[i]=xa;
      yaraw[i]=ya;
      /*x linear co-ordinate*/
      sum1=sum1+((float)0.1)*((float)xa-(float)offset1);
      /*y linear co-ordinate*/
      sum2=sum2+((float)0.1)*((float)ya-(float)offset2);
      xaarray[i]=sum1;
      yaarray[i]=sum2;
      /*printf("xo=%f,yo=%f \n" ,sum1 ,sum2);*/
      /*linear co-ordinates*/
      xb=analog(19);
      yb=analog(20);
      xbraw[i]=xb;
      ybraw[i]=yb;
      sum3=sum3+((float)0.1)*((float)xb-(float)offset3);
      sum4=sum4+((float)0.1)*((float)yb-(float)offset4); /*velocity*/
      xbarray[i]=sum3;
      ybarray[i]=sum4;
      /*printf("x1=%f,y1=%f \n" ,sum3 ,sum4);*/
      /*velocity and positional info*/
      /*sleep(0.0625);*/
   }
   off(1);off(3);
}
```

Program 2 colour sensor (colv 2110.c)

```
/* program colv2110.c was developed by Euclid Mills 10/10/09 */
/* colour sensor detection program */

int i=0;           /*initialization*/
int redarray[10];
int greenarray[10];/* global declarations*/
int bluearray[10]; /* of arrays to store red,green and blue.*/
int normalarray[10];
int red,green,blue;
int normal;

void main()
{
   while(1)
   {  /* start of loop*/
      set_digital_out(0);
      set_digital_out(1);   /* switch off LED's*/
      set_digital_out(2);
      sleep(0.05);
      clear_digital_out(0); /*switch on red LED*/
      sleep(0.5);
      red=analog(16);
      set_digital_out(0);
```

```
      sleep(0.05);
      clear_digital_out(1); /*switch on green LED*/
      sleep(0.5);
      green=analog(16);
      set_digital_out(1);
      sleep(0.05);
      clear_digital_out(2); /*switch on blue LED*/
      sleep(0.5);
      blue=analog(16);
      /* colour photo sensor input for R-G-B)*/
      set_digital_out(2);
      sleep(0.05);
      normal=analog(16);/* ambient light input*/
      if(i<9)i++;
      redarray[i]=red;
      greenarray[i]=green;
      bluearray[i]=blue;    /*array values storage*/
      normalarray[i]=normal;
      printf("n=%d,r=%d,g=%d,b=%d ,\n",normal,red,green,blue);

      if ((red>=60)&&(red<=64))
      {
         /*compare sensor values*/
         printf("colour red=%d\n",red);
      }
      sleep(0.5);

      if ((green>=67)&&(green<=69))
      {
         /*compare sensor output for R-G-B*/
         printf("colour green=%d\n",green);
         sleep(0.5);
      }

      if ((blue>=70)&&(blue<=71))
      {
         printf("colour blue=%d\n",blue);
         sleep(0.5); /*delay before starting the loop again*/
      }
   }
}
```

Program 3 (colv2111.c)

```
/* program colv2111.c was developed by Euclid Mills       10/10/09 */
/* colour sensor detection program */

int i=0;            /*initialization*/
int redarray[10];
int greenarray[10]; /* global declarations*/
int bluearray[10];
/* of arrays to store red,green and blue.*/
int normalarray[10];
int red,green,blue;
int normal;

void main()
{
   while(1)
   {  /* start of loop*/
      set_digital_out(0);
```

```
        set_digital_out(1);   /* switch off LED's*/
        set_digital_out(2);
        sleep(0.05);
        clear_digital_out(0); /*switch on red LED's*/
        sleep(0.2);
        red=analog(16);
        set_digital_out(0);
        sleep(0.05);
        clear_digital_out(1); /*switch on green LED*/
        sleep(0.2);
        green=analog(16);
        set_digital_out(1);
        sleep(0.05);
        clear_digital_out(2); /*switch on blue LED*/
        sleep(0.2);
        blue=analog(16);
        /* colour photo sensor input for R-G-B)*/
        set_digital_out(2);
        sleep(0.05);
        normal=analog(16);/* ambient light input*/
        if(i<9)i++;
        redarray[i]=red;
        greenarray[i]=green;
        bluearray[i]=blue;    /*array values storage*/
        normalarray[i]=normal;
        printf("n=%d , r=%d, g=%d, b=%d ,\n",normal,red,green,blue);

        if (red==74)
        { /*compare sensor values*/
           printf("colour red=%d\n",red);
           sleep(0.5);
        }

        if (green==72)
        { /*compare sensor output for R-G-B*/
           printf("colour green=%d\n",green);
           sleep(0.5);
        }

        if (blue==71)
        {
           printf("colour blue=%d\n",blue);
           sleep(0.5); /*delay befor starting the loop again*/
        }
     }
}
```