



**ARDUINO BASED PEDESTRIAN CROSSING CONTROL SYSTEM
FOR GWERU GENERAL HOSPITAL
PROTOTYPE BASED RESEARCH**



By

Nyasha B. D. Mambondiani | R113039F | February 2015

Supervisors: Mr. V. Mazheke and Mr. Sachingoma

A Dissertation Submitted to the Midlands State University in Partial Fulfilment of the
Requirements for a Bachelor of Science in Telecommunications Honours Degree

Department of Applied Physics and Telecommunications, Faculty of Sciences

Midlands State University

Gweru, 2015



Abstract

Pedestrian visitors at Gweru General Hospital are facing challenges crossing the Shurugwi road into the hospital premises due to high vehicular traffic influx to and from town. Thus this Arduino based pedestrian crossing control system has been designed to give priority to these pedestrians so that they can safely cross this perilous main road. The system incorporates the concept of passive pedestrian detection to sense the presence of pedestrians and in turn automatically controls the traffic lights. Relevant information gathering and electronic designing were incorporated into the designing of the system and some tests made on the effects of temperature on the detection range of the PIR sensor. Results obtained proved the system to be quite viable thus the project was a success.



Dedication

To my dear parents Mr. Charles and Mrs. Emily Mambondiani. No words can express how highly indebted I am to your unwavering support. Thank you for the strong foundation that you have laid up for me. You are greatly appreciated!



Acknowledgements

No single author could prepare a comprehensive work on a science dissertation without relying heavily upon the ideas and constructive criticisms of others. To try and mention them individually may only serve to underestimate my appreciation. All those whose published writings and researches I have quoted in this volume, my parents, my close relatives and friends this work would have been a futile exercise if it were not for your immense contribution. You are greatly appreciated!

Although acknowledging the contributions and assistance of others, I reserve to myself whatever mistakes and shortcomings may have found their way into these pages. Finally, Mr. V. Mazheke and Mr. Sachingoma, my advisors and supervisors for this work, I have no better words for you. Thank you.



Declaration

I do hereby solemnly declare that this submission is my own work and that to the best of my knowledge and belief, it contains no material neither already published or unpublished, nor submitted for the award of a certificate, diploma or degree in any institution of higher learning, except where due acknowledgement is made.

////////////////////////////////////

Nyasha B. D. Mambondiani

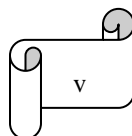




Table of Contents

Abstract	ii
Dedication	iii
Acknowledgement	iv
Declaration	v
Table of Contents	vi
List of Figures	ix
Table of Figures	xi
Glossary of Abbreviations	xii
Chapter 1	1
1.0 Introduction	
1.1 Background	
1.2 Justification	
1.3 Aim of Study	2
1.4 Purpose of Study	
1.5 Objectives of Study	3
1.6 Hypothesis	
1.7 Problem Statement	
1.8 Sub Problems	
1.9 Assumptions	4
Chapter 2 – Literature Review	6
2.0 Introduction	
2.1 Arduino Uno	



2.2 Arduino Software	12
2.3 USB Cable	13
2.4 Passive Infrared Sensor	14
2.5 Light Emitting Diodes	16
2.6 Liquid Crystal Display	18
2.7 Resistors	20
2.8 Solderless Breadboard	23
2.9 Jumper Wires	24
2.10 Pedestrian Crossings	25
2.11 Location of Pedestrian Crossing Facilities	26
2.12 Types of Pedestrian Crossing Facilities	
2.13 Crosswalk Markings	
2.14 Advantages of Marked Crosswalk Facilities	27
2.15 Disadvantages of Marked Crosswalk Facilities	
Chapter 3 – Methodology	31
3.0 Introduction	
3.1 System Structure	
3.2 Design of the PIR Sensor System	32
3.3 Design of Pedestrian and Traffic Lights System	33
3.4 Design of LCD System	34
3.5 Programming the Arduino Uno	35



3.6 Complete Circuit Diagram	36
3.7 Key Crosswalk Design Elements	37
3.8 Specifications	
3.9 System Flow Chart	38
Chapter 4 – Results	41
4.0 Introduction	
4.1 Presentation of Results	
4.2 Analysis and Interpretation of Results	42
4.3 Overall System Analysis	43
Chapter 5 – Conclusion	45
5.0 Introduction	
5.1 Discussion	
5.2 Limitations	
5.3 Recommendations	46
5.4 Recommendations for Future Research	
References.....	
Appendix	48
A. Crosswalk Lights Control Sketch	
B. LCD Sketch	



List of Figures

Figure 2.1: Front of the Arduino Uno Board	
Figure 2.2: Back of the Arduino Uno Board	
Figure 2.3: The USB and Power Connectors	
Figure 2.4: The ATMEGA 328 Microcontroller	
Figure 2.5: Arduino Uno R3 Systems	
Figure 2.6: The Power and Analog Sockets	
Figure 2.7: The Digital Input/output Pins	
Figure 2.8: The On-board LEDs	
Figure 2.9: The Reset Button	
Figure 2.10: The Arduino IDE	
Figure 2.11: A to B USB Cable	
Figure 2.12: PIR Sensor Side View	
Figure 2.13: Top and Inside of a PIR Sensor	
Figure 2.14: LEDs in 8mm, 5mm, and 3mm sizes	
Figure 2.15: Parts of an LED	
Figure 2.16: LED Symbol and Basic Structure	
Figure 2.17: Front View of a 16x2 LCD	
Figure 2.18: 16x2 LCD Back View	
Figure 2.19: 16x2 LCD Pin Configuration	
Figure 2.20: Typical Resistors	



Figure 2.21: Resistor Symbol and a Typical Resistor

Figure 2.22: Resistor Colour Codes

Figure 2.23: Typical Potentiometer

Figure 2.24: Pot Symbol and a Typical Potentiometer

Figure 2.25: Solderless Breadboard

Figure 2.26: Solderless Breadboard Pin Configuration

Figure 2.27: Internal Design of a Solderless Breadboard

Figure 2.28: Spool of 20AWG Wire and Jumper Wire

Figure 2.29: Crosswalk Pavement Markings

Figure 3.1: System Block Diagram

Figure 3.2: PIR Sensor Schematics

Figure 3.3: Lighting Schematics

Figure 3.4: LCD System Schematics

Figure 3.5: Arduino IDE with C++ Code Waiting for Upload

Figure 3.6: Complete Circuit Diagram

Figure 3.7: Expected Responses for Each Input and Output

Figure 3.8: System Flowchart

Figure 4.1: Temperature Effect on PIR Sensor Distance Detection

Figure 4.2: Arduino Based Pedestrian Crossing Control System.....

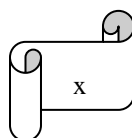




Table of Figures

Table 2.1: Technical Specifications of the Arduino Uno

Table 3.1: System Input and Output Devices

Table 4.1: Detection Distance of PIR Sensor and Corresponding Ambient Temperature



Glossary of Abbreviation

LCD – Liquid Crystal Display

IDE – Integrated Development Environment

PIR – Passive Infrared Sensor

I/O – Input Output

USB – Universal Serial Bus

RISC – Reduced Instruction Set Computer

ADC – Analog to Digital Converter

LED – Light Emitting Diode

TX – Transmit

RX – Receive

PC – Portable Computer

IR – Infrared

ASCII – American Standard Code for Information Interchange

DR – Data Register

RAM – Random Access Memory

RW – Read/Write

RS – Register Select

Pot – Potentiometer

GND – Ground

VO – Voltage Output

DC – Direct Current

MSU – Midlands State University

APS – Accessible Pedestrian Signals



CHAPTER 1

1.0 Introduction:

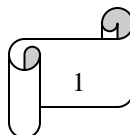
This project is based on the development of an Arduino based pedestrian crossing control system which can be used by pedestrian traffic (visitors) at Gweru General Hospital during visiting hours. The automated system will assist in giving priority to visitors wanting to cross the Shurugwi Road into the hospital premises considering the limited time they have to reunite with their beloved ones who are not in full health.

1.1 Background:

Pedestrians are legitimate users of the transportation system, and they should, therefore be able to use this system safely and without unreasonable delay. Pedestrian needs in crossing streets should be identified, and appropriate solutions should be selected to improve pedestrian safety and access. Deciding where to mark a pedestrian crossing is only one consideration in meeting that objective [1].

On several occasions on his way to town, the writer observed that during visiting hours (weekdays 6-7 a.m., 1-2 p.m., 6-7 p.m. and weekends 6-7 a.m., 3-6 p.m.) quite a number of visitors at Gweru General Hospital faced problems crossing the Shurugwi road into the hospital premises. This was due to high traffic volumes to and from town especially during the afternoon visiting hour. Traffic would pour in considerably from neighbouring communities like Senga and Nehosho and the majority from Shurugwi. Out of desperation, some pedestrians both young and old would end up resorting to jay walking just to get to the refugee island dividing the dual carriageway. Others would opt to wait for some time for the roads to clear up first before crossing cautiously. Thus in the writer's eyes this matter needed urgent attention since it left many pedestrians vulnerable and caused unnecessary time delays. So in line with this problem the author has designed an automated pedestrian crossing control facility which he hopes will improve ease of access for these visitors.

1.2 Justification:





Most of the pedestrian crossings used in Zimbabwe are the Zebra crossings incorporating black and white stripes across the road surface. In the case of Gweru General Hospital these ordinary systems do not suffice since most motorists disrespect them and have an evident desire not stop regardless of the fact that a pedestrian has the right of way once on the crossing. Again the Zebra crossings are not automated thus they are not time conscious. Furthermore, these crossing facilities are not so clearly visible to a majority of motorists especially when they are speeding as a result pedestrians are more prone to road accidents. Thus the visitors require a crossing facility that is clearly visible and is smart enough to spare some crossing time for pedestrian traffic without necessarily inconveniencing vehicular traffic.

1.3 Aim of Study:

To design a prototype of an automatic digital controller for a pedestrian crossing system used by visitors at Gweru General Hospital incorporating the concept of passive pedestrian detection. Passive pedestrian detection is the detection of pedestrians in a stationary or moving state at the kerbside of or in a pedestrian crossing by means other than those requiring physical response by the pedestrian [2]. In this system a passive infra-red sensor which measures change in heat energy of the surrounding area will be used as the sensing element or detector. Pedestrians will have to step directly underneath the detector on the detection zone for it to trigger the light control sequence prior to crossing. The system will control both vehicular and pedestrian traffic signals. It enhances ease of access and safety for pedestrians as well as reduce elements of jaywalking thus ensuring order on roads.

1.4 Purpose of Study:

To design a prototype of an automated digital pedestrian crossing control system used by visitors at Gweru General Hospital. The system will keep pedestrians together where they can be seen by motorists whilst they are moving across the flow of vehicular traffic thus improving efficiency and safety on the road transport system. This facilitates coexistence of vehicles and pedestrians safely and conveniently. Pedestrians will not have to face unnecessary delays prior to crossing the Shurugwi Road.



1.5 Objectives of Study:

- To give visitors priority of crossing the Shurugwi road during visiting hours.
- To employ the concept of Passive Pedestrian Detection in the prototype.
- To increase drivers' yielding rate at pedestrian crossings by making it even more visible.
- To observe the variation between detection distance of the PIR sensor with ambient temperature.
- To reduce vehicle-pedestrian crash rates on roads since most pedestrians will be using the crossing facility for convenience.
- To design advance warning signs for motorists in form of liquid crystal display units (LCDs),

1.6 Hypothesis:

The pedestrian crossing control system will automatically assist pedestrians to safely cross the Shurugwi road without any physical response coming from them. Thus increasing convenience and ease of access for them.

1.7 Problem Statement:

The current road set up at the entrance of Gweru General Hospital along the Shurugwi Road makes pedestrian traffic susceptible to road accidents or injuries especially during visiting hours when both traffic levels are relatively high. The system is highly favouring vehicular traffic leaving pedestrian traffic slightly vulnerable when they wish to cross the road to get into the hospital premises. Pedestrians have to wait for some time for vehicular traffic to clear up at times endangering themselves by making miscalculated moves whilst crossing the road. This is due to the fact that there is no appropriate crossing facility to cater for them. Pedestrians are legitimate users of the transportation system and they also require safe crossing facilities.

1.8 Sub Problems:



- Propose and design a pedestrian crossing prototype using an Arduino Uno microcontroller.
- Programme the microcontroller using the Arduino Integrated Development Environment (IDE) to facilitate the operation of the proposed circuitry.
- Writing a C++ assembly language that controls the variables.
- Designing and calibrating the passive infra-red (PIR) sensor.

1.9 Assumptions:

- The crosswalk facility is used during visiting hours only so as not to constantly inconvenience vehicular traffic.
- Pedestrians do not wear heat-insulating full-body suit or Styrofoam material for them to be detected by the PIR sensor.
- There are no sudden, drastic environmental changes like hot/cold air flow or sunlight which may trigger the sensor.
- Users have been educated on how the pedestrian crossing control system works for them to properly use it.



References

- [1] C. Zeeger, "Safety Effects of Marked Versus Unmarked Crosswalks at Uncontrolled Locations: Executive Summary and Recommended Guidelines," Federal Highway Administration, United States of America, 2002.
- [2] D. M. Beckwith and K. M. Hunter-Zaworski, "Passive Pedestrian Detection at Unsignalised Crossings," *Journal of the Transportation Research Board*, vol. 1636, pp. 96-103, 2007.



CHAPTER 2

Literature Review

2.0 Introduction:

This chapter tends to come up with a detailed examination of all the hardware and software incorporated into the crosswalk model. The theoretical aspect is explained in such a way that any system designer will ultimately gain an understanding of how the components work and how to best interface them. This understanding enhances troubleshooting methods.

2.1 Arduino Uno:

It is an Arduino based board also known as a microcontroller, which is the same thing as a small computer on a single, integrated circuit with a processor, memory, and controllable output and input pins. On this board, one can write programs and create interface circuits to read switches, sensors, lights and to control motors as well as other electronic components with very little effort [1]. It supports C and C++ programming languages.

Arduino is an Open Hardware project started in 2005 that tries to bring the world of digital electronics to education, research, and the maker community [2]. Arduino Uno is Italian and means “one”. It was released publicly in September 25, 2010. It is inexpensive and reliable, \$30 or €22.



Figure 2.1: Front of the Arduino Uno Board [3]



2.1.2 Technical Specifications of the Arduino Uno:

Table 2.1: Technical Specifications of Arduino Uno

Microcontroller	ATmega328 (8-bits Architecture)
Operating Voltage	5 V
Input Voltage (recommended)	7–12 V
Input Voltage (limits)	6–20 V
Digital I/O Pins	6–20 V
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3 V Pin	50 mA
Flash Memory	32 KB (ATmega328) 0.5 KB for bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Figure 2.2 is the back view of the Arduino Uno board with inscriptions written on it and four surface mounting holes.

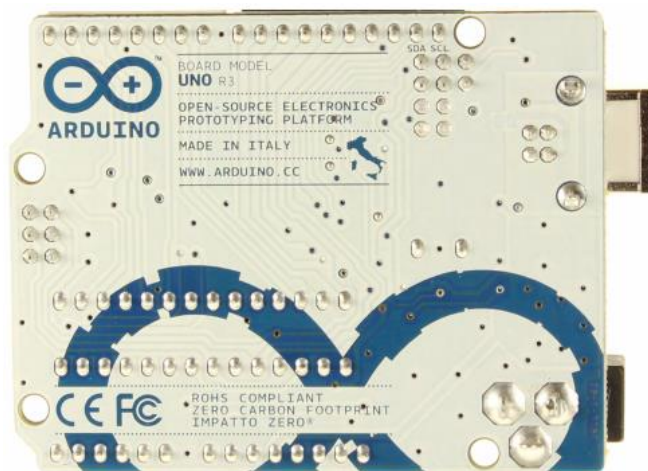
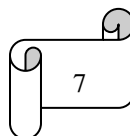


Figure 2.2: Back of the Arduino Uno Board [3]

2.1.3 Parts of the Arduino Uno:





Two connectors are found on the left side of the Arduino Uno. The one on the far left is the Universal Serial Bus (USB) connector. It connects the board to the computer for three reasons:

- To supply power to the board.
- To upload instructions to the Arduino Uno.
- To send data to and receive it from a computer.

The power connector on the right powers the Uno with a standard mains power adapter.

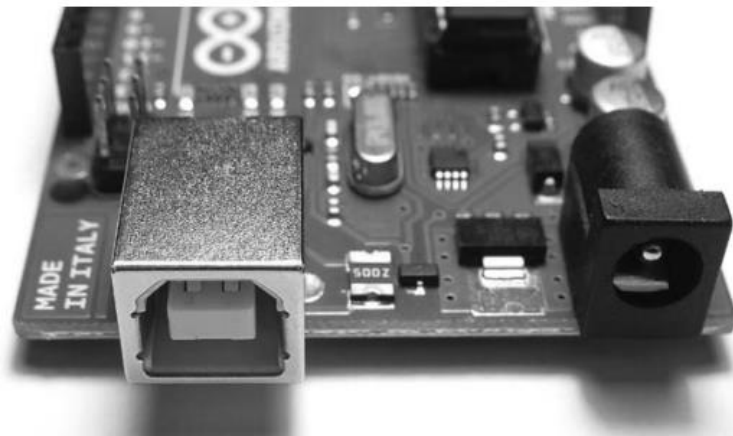


Figure 2.3: The USB and Power Connectors [4]

The microcontroller is at the lower middle and acts as the “brains” of the board. It is a tiny computer that contains a processor to execute instructions, includes various types of memory to hold data and instructions from sketches, and provides various avenues of sending and receiving data [4].



Figure 2.4: The ATMEGA 328 Microcontroller [4]



The “Atmel Atmega328” is a 28 pin, 8-bit microcontroller [5]. The Reduced Instruction Set Computer (RISC) concept forms its architecture. When operating at 20 MHz the processor completes 20 million instructions per second. The microcontroller is equipped with features which can be categorized into the following systems:

- Memory system,
- Port system,
- Timer system,
- Analog-to-digital converter (ADC),
- Interrupt system,
- And the Serial communications.

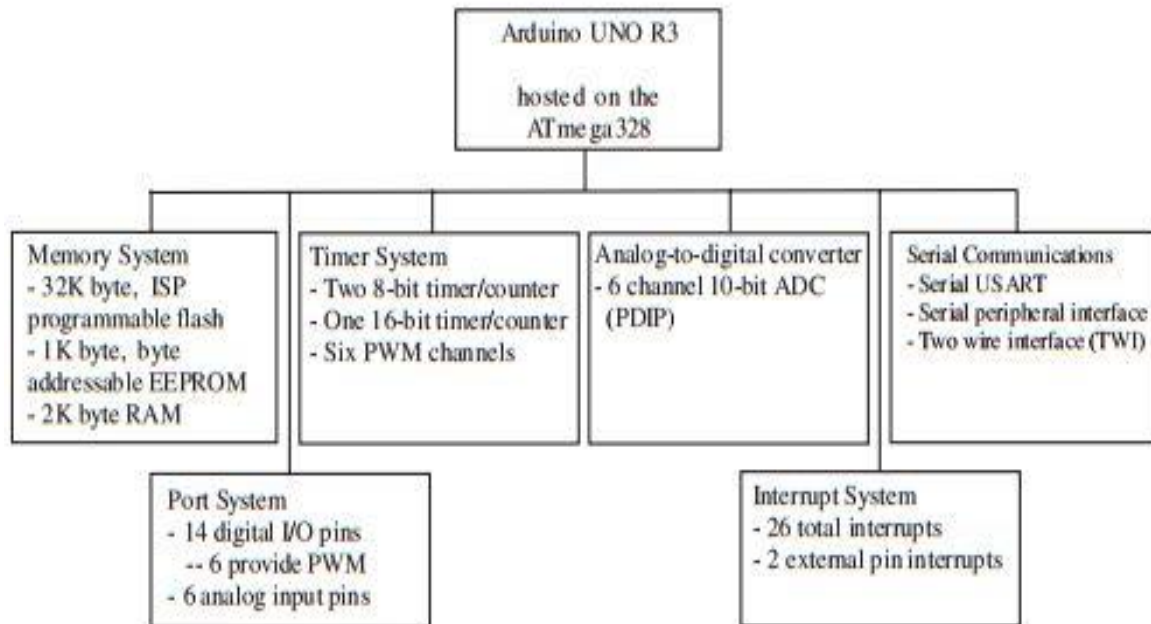


Figure 2.5: Arduino Uno R3 Systems [5]

Two rows of small sockets are found below the microcontroller. The first one has power connections and the ability to use an external RESET button. The second one has six analog inputs, used to measure electrical signals that vary in voltage. Pins A4 and A5 can be used for sending data to and receiving it from external devices.

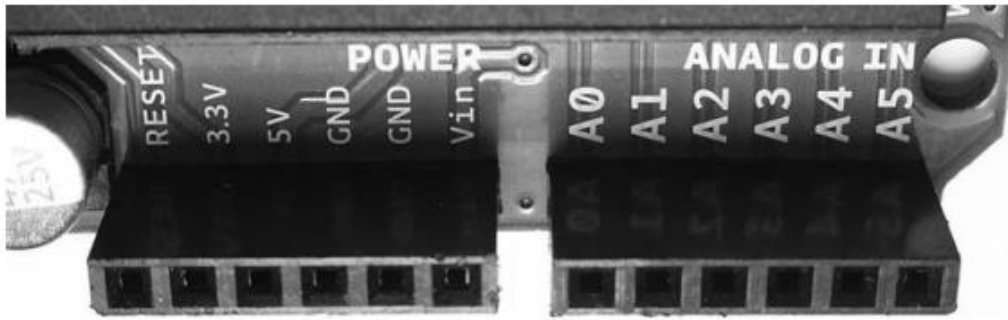


Figure 2.6: The Power and Analog Sockets [4]

Along the top, sockets numbered 0 to 13 are digital (I/O) pins. They can detect the presence of an electrical signal or generate a signal on command. Pins 0 and 1, also known as the serial port, can be used to send and receive data to external devices, via the USB connector circuitry. The pins labelled with a tilde (~) can also generate a varying electrical signal, which can be useful for such things as creating lighting effects or controlling electric motors [4].



Figure 2.7: The Digital Input/output Pins [4]

The board has four LEDs, one labelled ON that indicates when the board has power, and three others TX, RX and L. TX and RX light up when data is being transmitted or received between the Arduino Uno and attached external devices through the USB and serial port. “L” is for the operator’s use and is linked to the digital I/O pin 13. The little black square part to the left of the LEDs in Figure 2.8 is a tiny microcontroller that controls the USB interface that allows Arduino Uno to send data to and receive it from a computer [4].

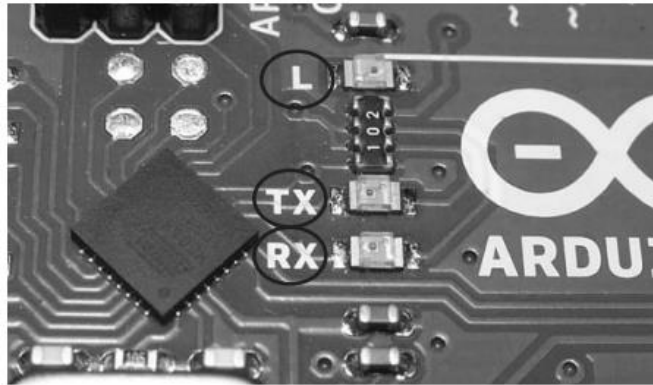


Figure 2.8: The On-board LEDs [4]

The RESET button is used to reset the system and restart the Arduino Uno to resolve the pending problems.

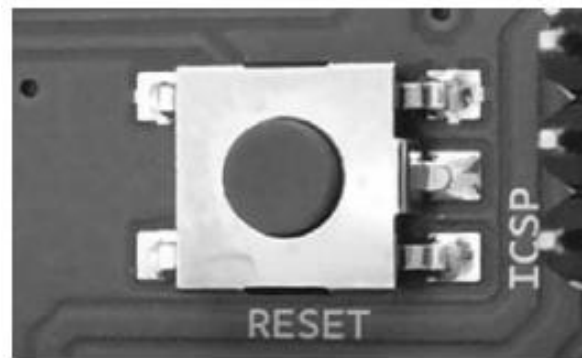


Figure 2.9: The Reset Button [4]

2.1.4 Advantages of Arduino Uno:

- Ease of expandability – it is easy to add more hardware functions. The two rows of sockets along each side of the Arduino allow the connection of a shield that is another circuit board with pins that allow it to plug into the Arduino.
- Programmable USB chip on board - enables one to flash the chip in such a way that when plugged into the PC it will show up as any USB device such as a mouse. Thus enabling one to use the Arduino as an interface for creating USB devices.

2.1.5 Disadvantages of Arduino Uno:



- Cannot communicate directly with an Android device - it lacks the USB Host functionality. However, shields are available that can bring the USB Host functionality and add Bluetooth communication.

2.2 Arduino Software:

It is a collection of instructions that tell the hardware what to do and how to do it. Two types of software are used:

- The integrated development environment (IDE).
- The Arduino sketch that one creates.

The IDE software is installed on the personal computer and is used to compose and send sketches to the Arduino Uno board [4].

2.2.1 Arduino Integrated Development Environment:

The Arduino IDE resembles a simple word processor and is divided into three main areas:

- The command area.
- The text area.
- The message window area.



Figure 2.10: The Arduino IDE [4]



The Command Area - It includes the title bar, menu items, and icons. The title bar displays the sketch's filename as well as the version of the IDE. Below it is a series of menu items (File, Edit, Sketch, Tools, and Help) and icons.

The Text Area - This is where one creates a sketch and the name of the current sketch is displayed in the tab at the upper left of the text area. Contents are entered as in any text editor.

The Message Window Area - Messages from the IDE appear in the black area. The messages vary and include messages about verifying sketches and status updates.

At the bottom right of the message area is displayed the name of the Arduino board as well as its connected USB port.

2.2.2 Arduino Sketch

It is a set of instructions that you create to accomplish a particular task [4]. In other words, a sketch is a program. It possesses three main attributes which are:

Comments – these are notes of any length in a sketch, written for the user's benefit. They are useful for adding notes to yourself or others, for entering instructions, or for noting miscellaneous details. They begin with two forward slashes “//” or “/*”.

The Setup Function - The void setup() function is a function that contains a set of instructions for the Arduino to execute once only, each time it is reset or turned on.

The Loop Function – The void loop() function tells the Arduino to execute an instruction over and over until the power is shut off or someone presses the RESET button.

2.3 USB Cable:

The A to B USB cable is used for programming and providing power to the Arduino Uno. This type of USB cable is mostly used to connect printers to computers.



Figure 2.11: A to B USB Cable [6]

2.4 Passive Infra-red Sensor (PIR):

It is a type of sensor that is used for detecting changes in infrared radiation in its field of view thus can detect changes in temperature. It can measure up to a maximum distance of 9 m (30 feet) field of view in which case its sensitivity has to increase. It uses 3-6 VDC and uses 12mA at 3V and 23mA at 5V [7]. “Passive” in this case implies that the PIR sensor does not send any signals but only reads in infrared light or heat signals [6]. Typically it calibrates for a short period of time to create a map of the infrared energy in front of it, and once the calibration is done, it triggers a signal when the amount of infrared changes suddenly. Everything emits some low-level radiation, and the hotter something is, the more radiation it emits. PIR sensors come in a variety of styles and are sensitive over different distances and angles [8]. The IR sensor is typically covered by a dome-shaped lens that helps to condense and focus light so that it is much easier for the sensor to detect infrared variations [9].

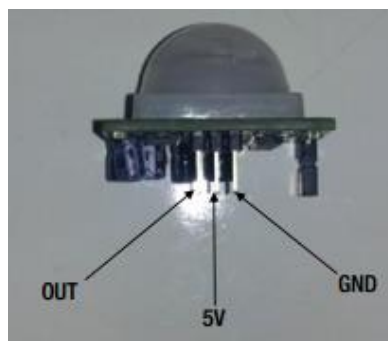


Figure 2.12: PIR Sensor Side View [6]



Decreasing the sensitivity makes the sensor more immune to random events, while still allowing it to detect actual movement if a person is within its operational field of view. A high pulse-count setting will decrease its ability to detect very slow movement which is why the setting should generally be left low.

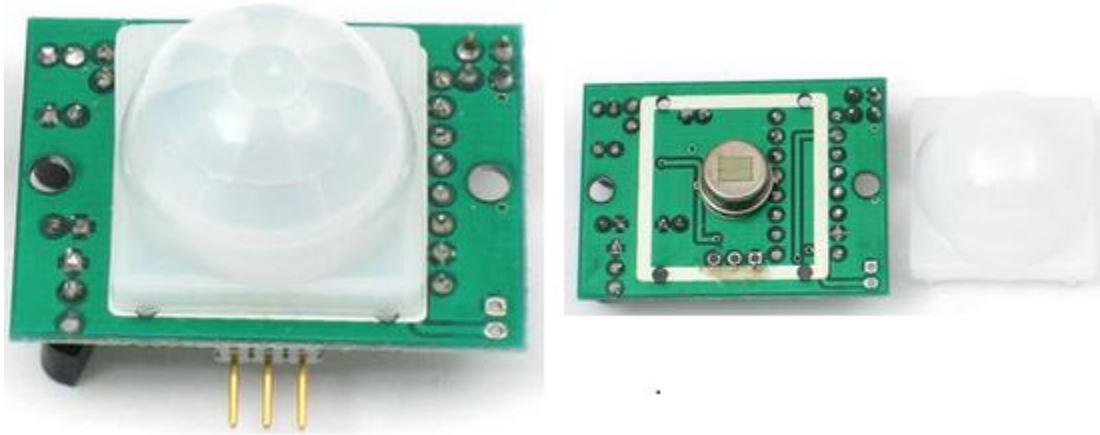


Figure 2.13: Top and Inside of a PIR Sensor [10]

2.4.1 Detection of Movement:

At the center of a rectangle on the sensor's face is a crystalline material which detects the infrared radiation. The PIR sensor is split in two halves to detect not the radiation itself, but the change in condition that occurs when a target enters its field. Changes in infrared radiation amount on the sensor result in changes in generated voltages. An on board amplifier measures these voltages. The sensor actually detects the broken field for a "normal" temperature. The field does not have to be broken by an object with a different temperature in order to register change. Highly sensitive PIR sensors will activate from the movement alone. Designed to work in temperatures of 15 to 20 degrees Celsius. The field of view narrows at higher temperatures and at below 15 degrees the field of view widens thus distant objects can activate the sensor. When the PIR detects motion it outputs a high signal on its output pin which can be read by a microcontroller [11]. PIR sensors are mostly used in security lighting and indoor burglar alarm systems.

2.4.2 Advantages of PIR Sensors:



- It is small, light and easy to connect.
- They are sensitive over different angles and distances.
- They are very cheap in terms of cost.

2.4.3 Disadvantages of PIR Sensors:

- They can be easily triggered by insects and animals so careful positioning and calibration is required.
- They require line of sight.

2.5 Light Emitting Diodes (LEDs):

An LED is a semiconductor forward biased p-n junction diode with two leads, an anode and a cathode. When the voltage on the anode is more positive than that on the cathode (forward voltage) the device emits light (photons). It emits light through spontaneous emission by a phenomenon termed electroluminescence [12]. Most LEDs will provide a reasonable level of light output when a forward current of between 5 mA and 20 mA is applied [13]. The anode is usually the longer lead, and there is often a flat spot on the housing to indicate the cathode. The LED color and the exact value of the forward voltage depend on the construction of the diode [8].

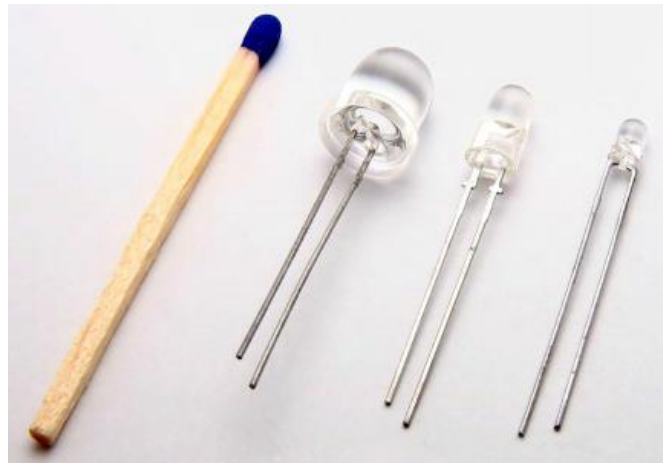


Figure 2.14: LEDs in 8 mm, 5 mm, and 3 mm sizes [3]



Light emitting diodes are available in various formats with the round types being most popular. Round LEDs are commonly available in the 3 mm and 5 mm diameter plastic packages and also in a 5 mm×2mm rectangular format [13].

2.5.1 Advantages of LEDs:

- Efficiency – they emit more lumens per watt and are not affected by shape and size.
- Size – they are very small and can be incorporated into smaller devices.
- Response Time – they light up very quickly.
- Lifetime – they can last relatively long.
- Shock Resistant – they are solid state thus difficult to damage with external shock.

2.5.2 Disadvantages of LEDs:

- Temperature Dependence – their performance depends on the ambient temperature of operating environment.
- Voltage Sensitivity – must be supplied with voltage above the threshold and current below the rating.
- Electrical Polarity – they only light with the right polarity connections.
- High Initial Price – they are currently more expensive on an initial capital cost basis.

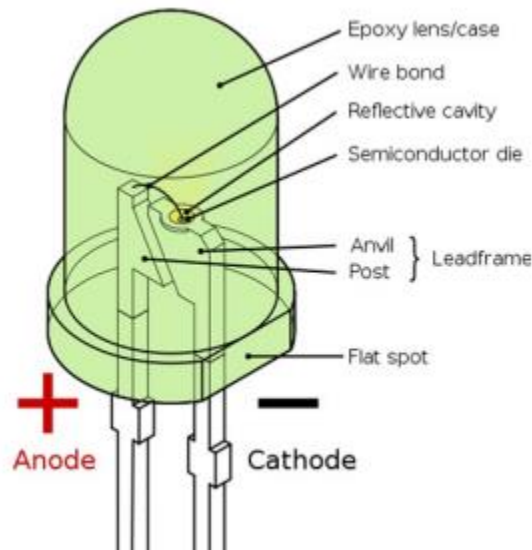


Figure 2.15: Parts of an LED [14]



2.5.3 Uses of LEDs:

- Alphanumeric displays in seven-segment and dot-matrix displays
- Aviation lighting
- Automotive headlamps
- Advertising
- Camera flashes
- Advertisements
- General lighting

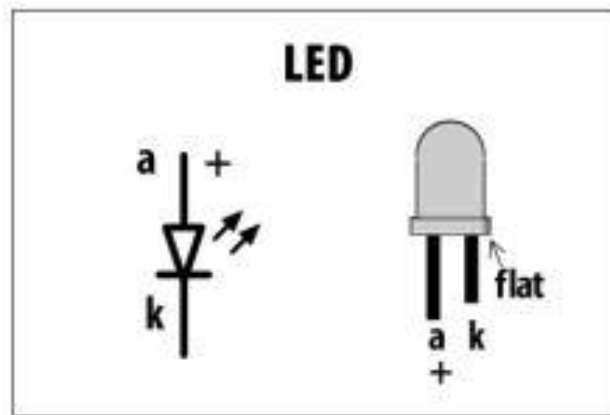


Figure 2.16: LED Symbol and Basic Structure [8]

2.6 Liquid Crystal Display (LCD):

It is a module used to display information to the user. Two types of modules exist, the character LCD module and the graphic LCD module. The first one displays custom characters, text and numeric data whereas the latter displays graphics. Character LCDs are inexpensive and easier to use. Liquid crystal displays are manufactured using materials which have both properties of a liquid and also those of a crystal. They are of various sizes and shapes and are measured by the number of rows and columns used to display characters. Some have a backlight and allow selection of the colour of the character as well as the background. A trimpot is used to adjust the display contrast.

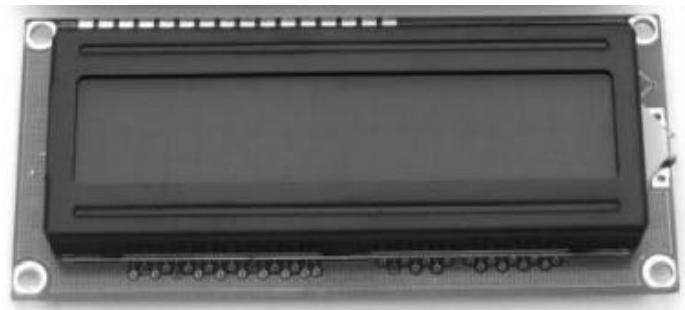


Figure 2.17: Front View of a 16x2 LCD [4]

Characters are sent to the LCD via American Standard Code for Information Interchange (ASCII), one character at a time. Each character is displayed in a 5x7 pixel matrix. The LCD display Module is built in a LSI controller, the controller has two 8-bit registers, an instruction register (IR) and a data register (DR). The Instruction Register stores codes such as display clear and cursor shift, and it also addresses information for display data RAM (DDRAM) and the character generator (CGRAM). The Data Register is a temporary location for storing data to be written or read from DDRAM or CGRAM.

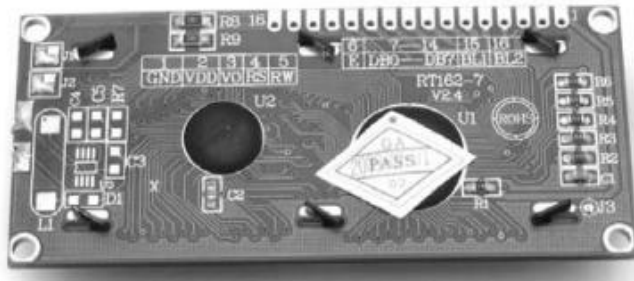


Figure 2.18: 16X2 LCD Back View [4]

2.6.1 LCD Pin Configuration:

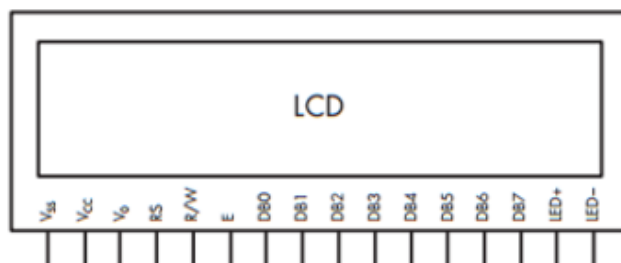


Figure 2.19: 16x2 LCD Pin Configuration [4]



VCC / VDD – The positive supply 5V.

VSS – Ground of the LCD.

VO (V Open) – LCD display contrast.

RS (Register Select) - Instructs the LCD to consider data either as a register address or instruction code.

RW (Read/Write) – Text can be written to the screen when it is low. When high, data from the LCD can be read back into the Arduino.

E (Enable) - Acts as a switch to notify the LCD that data is available on the data pins.

DB0-DB7 – They are 8-bit parallel data pins. DB7 is the most significant bit (MSB) going down to DB0 the least significant bit (LSB). Four data pins can be used when operating in 4-bit mode to save pins.

LED+ - Anode for the LED backlight.

LED- - Cathode for the LED backlight.

2.7 Resistors:

A resistor is an electronic component designed to oppose an electric current by producing a voltage drop between its two ends in relation to the current. In other words, a resistor slows down the flow of current that will lower the voltage going into the resistor. They are made from a variety of materials some metal and others carbon. Metal ones are more precise and have less noise as compared to carbon ones which however are less expensive. There are several types of resistors for example photoresistors (photocells) and potentiometers.

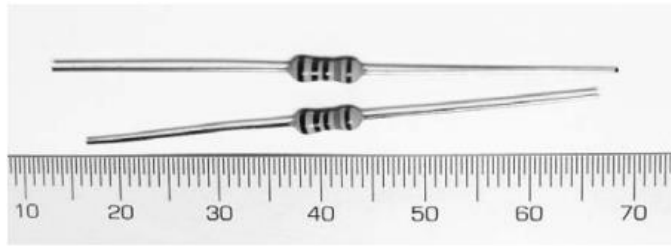


Figure 2.20: Typical Resistors [4]

Resistance is always measured in ohms and can also be presented as the Ω symbol. The resistor value is calculated using the following formula:

$$R = \frac{V_s - V_f}{I} \dots \dots \dots (2.1)$$

Where (V_s) is power supply voltage, (V_f) is the LED forward voltage and (I) is the amount of current that you want to flow through the LED.

Equation 2.1 was derived from Ohm's Law which says that:

- The current flowing through a conductor (I) is directly proportional to the potential difference (V) across the two ends. The constant of proportionality is called resistance (R) [15].

Simply summarised by an equation it becomes:

$$R = \frac{V}{I} \dots \dots \dots (2.2)$$

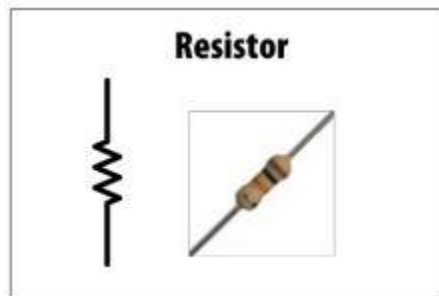


Figure 2.21: Resistor Symbol and a Typical Resistor [8]



Resistors are so small thus values are encoded using coloured stripes. All resistors are color-coded with four, five, or six color bands. These bands tell the resistance.

Color	1 st Band	2 nd Band	3 rd Band (multiplier)	4 th Band (tolerance)
Black	0	0	$\times 10^0$	
Brown	1	1	$\times 10^1$	$\pm 1\%$
Red	2	2	$\times 10^2$	$\pm 2\%$
Orange	3	3	$\times 10^3$	
Yellow	4	4	$\times 10^4$	
Green	5	5	$\times 10^5$	$\pm 0.5\%$
Blue	6	6	$\times 10^6$	$\pm 0.25\%$
Violet	7	7	$\times 10^7$	$\pm 0.1\%$
Grey	8	8	$\times 10^8$	$\pm 0.05\%$
White	9	9	$\times 10^9$	
Gold			$\times 10^{-1}$	$\pm 5\%$
Silver			$\times 10^{-2}$	$\pm 10\%$
None				$\pm 20\%$

Figure 2.22: Resistor Colour Codes [14]

2.7.1 Potentiometer (Pot):

It is a variable resistor that has two outside terminals which act as a fixed resistor and a movable contact called a wiper or slider that moves across the resistor, producing a variable resistance between the center terminal and the two sides.



Figure 2.23: Typical Potentiometer [4]



2.7.2 Uses of Pots:

- Volume controls on audio equipment.
- Control inputs for electronic circuits, for example dimmer switch for lights.

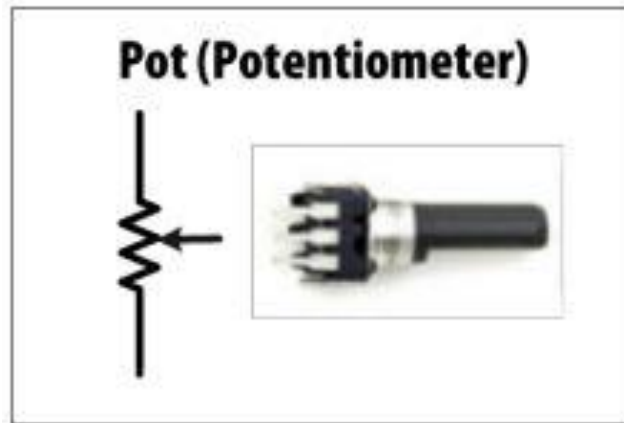


Figure 2.24: Pot Symbol and a Typical Potentiometer [8]

2.8 Solderless Breadboard:

A plastic board studded with pins that can hold jumper wires to make designing of circuits easier. It allows one to quickly assemble components without soldering the pieces together.



Figure 2.25: Solderless Breadboard [6]

Inside at the base are metal strips that connect the small holes together in rows. The rows of holes are both connected in vertical and horizontal lines. No soldering of electronic components is required when using this board thus it is very easy to use.

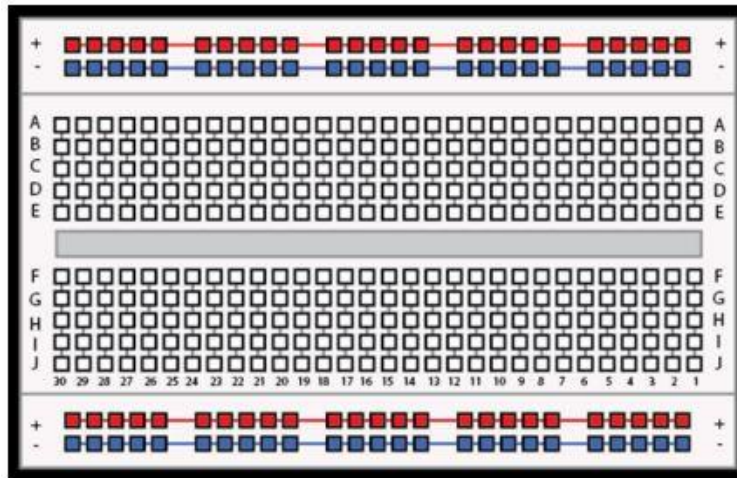


Figure 2.26: Solderless Breadboard Pin Configuration [15]

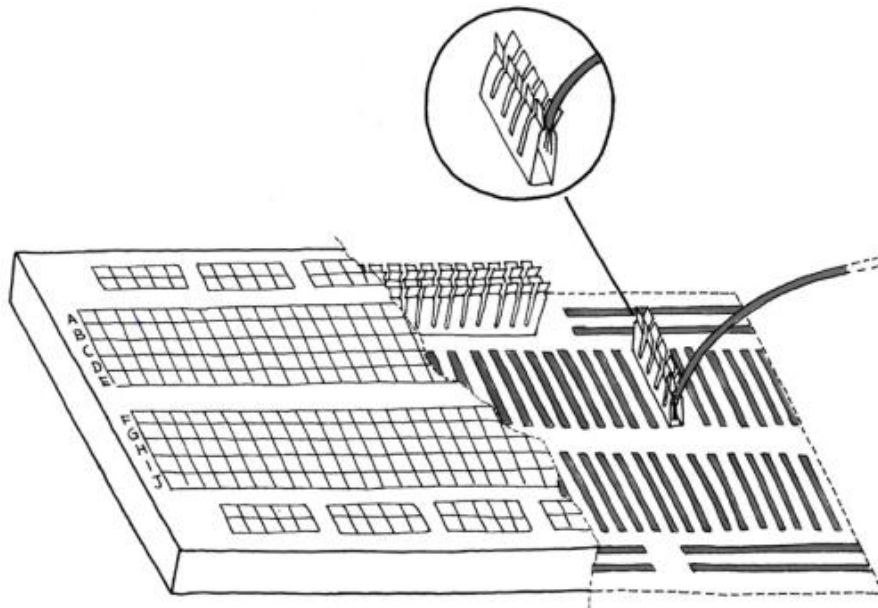


Figure 2.27: Internal Design of a Solderless Breadboard [16]

2.9 Jumper Wires:

Jumper wires also known as hook-up wires 1mm core diameter (AWG 20 or AWG 22) are used to connect the electronic components, the Arduino, and the breadboard [6]. They have solid metal tips perfectly sized to fit into Arduino and breadboard pins, and come sheathed in various colours of insulation.

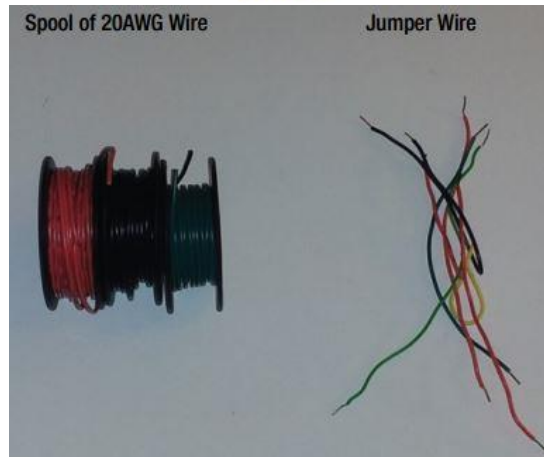


Figure 2.28: Spool of 20AWG Wire and Jumper Wire [6]

2.10 Pedestrian Crossings:

Generally a pedestrian crossing is a place designated for pedestrians to cross the road. It is designed to keep pedestrians together where they can be seen by motorists, and where they can cross most safely across the flow of vehicular traffic. The Institute of Transportation Engineers in 1998 defined the crosswalk as:

- The portion of a roadway designated for pedestrians to use in crossing the street [17].

However, the Manual for Uniform Traffic Control Devices 2009 Edition defines the crosswalk as:

- That part of a roadway at an intersection included within the connections of the lateral lines of the sidewalks on opposite sides of the highway measured from the curbs or in the absence of curbs, from the edges of the traversable roadway, and in the absence of a sidewalk on one side of the roadway, the part of a roadway included within the extension of the lateral lines of the sidewalk at right angles to the center line;
- Any portion of a roadway at an intersection or elsewhere distinctly indicated as a pedestrian crossing by pavement marking lines on the surface, which might be supplemented by contrasting pavement texture, style, or color [18].



2.11 Location of a Pedestrian Crossing Facility:

The practice of where to install a pedestrian crossing facilities differs considerably from one jurisdiction to another across the globe. Thus engineers have been left with using their own judgement sometimes influenced by political and/or public pressure in reaching decisions. Some cities have developed their own guidelines on where marked crosswalks should or should not be installed [19]. However many factors contribute in choosing the perfect location such as traffic volume, pedestrian volume, number of lanes, presence or type of median and so on.

2.12 Types of Pedestrian Crossing Facilities:

Listed below are some of the crosswalk systems currently in use [20]:

- Pelican (Pedestrian Light Controlled crossing) – One pushes a button on a pole and waits for the green man to light then they cross the road cautiously.
- Pegasus crossing – this system is provided for horse riders. A green horse is provided above the button as well as a holding area for the rider and his horse.
- Puffin (Pedestrian User-Friendly Intelligent crossing) – on this system the button is below the green/red man on the same kerbside the pedestrian is on. The pedestrian pushes the button and waits for the green man to light then cautiously crosses the road.
- Toucan ('two can') – the system is meant for both cyclists and pedestrians. A green cycle display is provided next to the green man and can be above the button like the Puffin or on the opposite kerbside like the Pelican.
- Zebra crossing – They incorporate black and white stripes across the road and are mostly used at uncontrolled crossings.

2.13 Crosswalk markings:

They are used to define the pedestrian path of travel across the roadway and alert drivers to the crosswalk location [17]. Six types of pavement markings are normally used globally to mark the road surface on the designated crossings. There is the Solid, Dashed, Standard,



Continental, Zebra and Ladder pavement markings. According to Knoblauch and his colleagues' research in 1988, the continental design is recommended because it is the most visible to drivers [17].

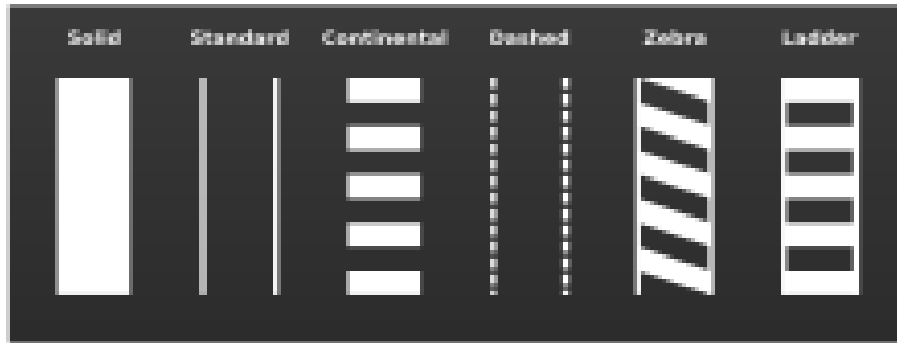


Figure 2.2910: Crosswalk Pavement Markings [18]

Use of the continental design for crosswalk markings improves crosswalk detection for people with low vision and cognitive impairments. It is recommended that the continental design be used consistently to mark all crosswalks; otherwise the impact of less visible markings may be weakened by comparison [17].

2.14 Advantages of Marked Crosswalk Facilities:

According to the Virginia Department of Transportation [21]:

- Designate the shortest path for pedestrians.
- They assist pedestrians in finding their way across complex intersections.
- They direct pedestrians to a location of best sight distance.
- They help create reasonable expectations for motorists with regards to where a pedestrian may cross the roadway and predictability of pedestrian actions and movement.

2.15 Disadvantages of Marked Crosswalk Facilities:

According to the Virginia Department of Transportation [21]:

- They possibly create a “false sense of security” for pedestrians.



- They have higher maintenance cost.
- They generate a higher pedestrian crash rate at uncontrolled locations that is locations not controlled by signals, stop signals and yield lines.



References

- [1] T. Olsson, *Arduino Wearables*, New York: Apress, 2012.
- [2] A. Goranson and D. C. Ruiz, *Professional Android Open Accessory Programming with Arduino*, Indianapolis: John Wiley & Sons, 2013.
- [3] P. Di Justo and E. Gertz, *Atmospheric Monitoring with Arduino*, California: O'Reilly, 2013.
- [4] J. Boxall, *Arduino Workshop*, San Francisco: No Starch Press, 2013.
- [5] S. F. Barrett, *Arduino Microcontroller: Processing for Everyone*, Wyoming: Cley pool Publishers, 2012.
- [6] J. F. Kelly and H. Tunis, *Arduino Adventures*, New York: Apress, 2013.
- [7] G. McComb, "Learn.parallax.com," Parallax Inc, 05 October 2012. [Online]. Available: <http://www.learn.parallax.com>. [Accessed 29 May 2015].
- [8] M. Margolis, *Arduino Cookbook*, California: O'Reilly, 2012.
- [9] M. Riley, *Programming Your Home*, North Carolina: The Pragmatic Bookshelf, 2012.
- [10] Adafruit, "PIR Motion Sensor Tutorial," Instructables, 2010. [Online]. Available: <http://www.instructables.com/id/PIR-Motion-Sensor-Tutorial/>. [Accessed 21 April 2015].
- [11] C. Mathas, "Sensing Motion with Passive Infrared (PIR) Sensors," *Article Library*, 28 June 2012.
- [12] E. Phillip and A. Laplante, *Electrical Engineering Dictionary*, Boca Raton: CRC Press LLC, 2000.



- [13] J. Bird, Electrical and Electronic Principles and Technology, Third Edition, New York: Newnes, 2007.
- [14] M. McRoberts, Beginning Arduino, 2nd Edition, New York: Apress, 2013.
- [15] E. R. Melgar and C. C. Diez, Arduino and Kinetic Projects, New York: Apress, 2012.
- [16] M. Banzi, Getting Started with Arduino, 2nd Edition, California: O'Reilly, 2011.
- [17] F. H. Administration, "Part II of II: Best Practices Design Guide - Sidewalk2 - Publications - Bicycle & Pedestrian Program - Environment - FHWA," U.S. Department of Transportation/Federal Highway Administration, 02 October 2014. [Online]. Available: http://www.fhwa.dot.gov/environment/bicycle_pedestrian/publications/sidewalk2/sidewalks. [Accessed 04 April 2015].
- [18] Federal Highway Administration, Manual on Uniform Traffic Control Devices for Streets and Highways, United States of America: U.S. Department of Transportation, 2009.
- [19] C. Zeeger, "Safety Effects of Marked Versus Unmarked Crosswalks at Uncontrolled Locations: Executive Summary and Recommended Guidelines," Federal Highway Administration, United States of America, 2002.
- [20] Lincolnshire County Council, Pedestrian Crossings - A User's Guide, United Kingdom: County Council Lincolnshire, 2009.
- [21] V. D. o. Transportation, Guidelines for the Installation of Marked Crosswalks, Virginia: Traffic Engineering Division, 2004.



CHAPTER 3

Methodology

3.0 Introduction:

This chapter takes a detailed examination of all the building blocks that make up the pedestrian crossing control system. It comes up with a detailed description of how these sub-elements are interfaced with the Arduino Uno microcontroller and displays the block diagram and flow chart of the overall system.

3.1 System Structure:

The prototype consists of four main building blocks:

- A set of traffic lights – to control vehicular traffic flow.
- A set of pedestrian lights – to control pedestrians.
- A pair of LCDs – to alert vehicular traffic of the presence of the crosswalk.
- A PIR sensor – to detect pedestrians intending to cross.

The traffic lights, pedestrian lights and PIR sensor are all connected to a separate Arduino Uno microcontroller whilst the LCDs are connected to the other one. This is due to the fact that the Arduino Uno has a minimum number of pins. Figure 3.1 below is the system block diagram.

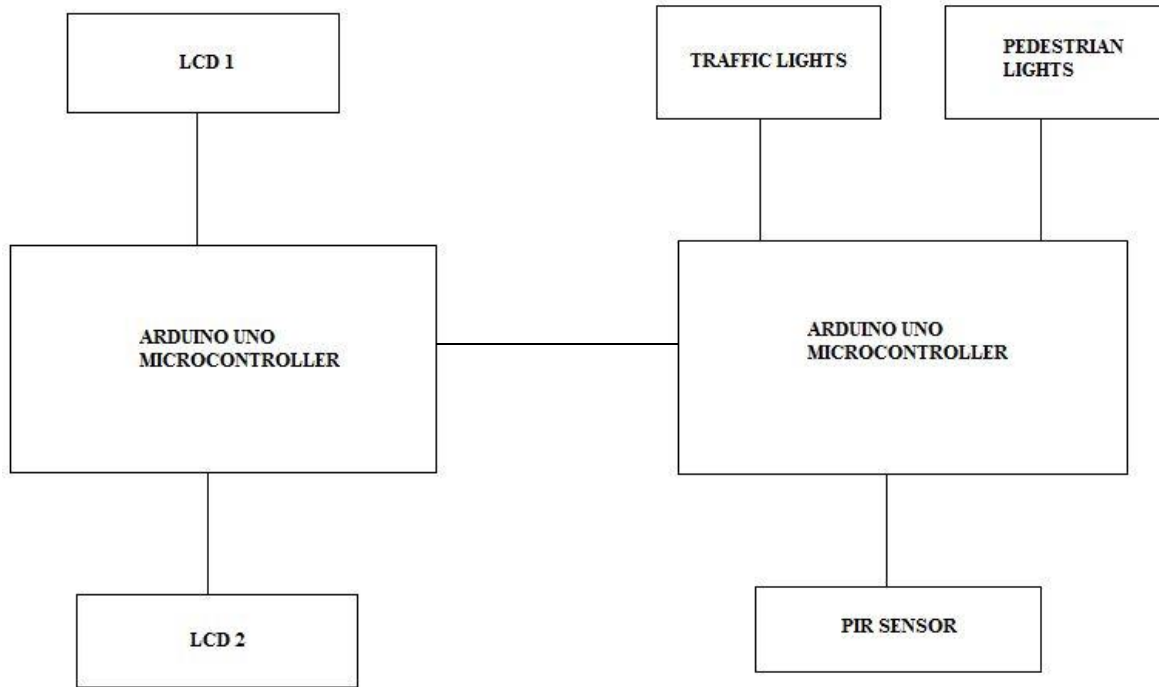


Figure 3.1: System Block Diagram

3.2 Design of the PIR Sensor System:

The PIR sensor pin configuration is shown in figure 2.12. It has three pins, GND connected to the ground pin of the microcontroller, OUT connected as input to the Arduino Uno digital input 2 and VCC connected to the positive 5 volts supply of the microcontroller. The PIR sensor checks for changes in infra-red levels thus sends a signal to the Arduino digital pin 2 as HIGH or LOW [1]. Figure 3.2 below shows the PIR sensor schematic diagram to the microcontroller.

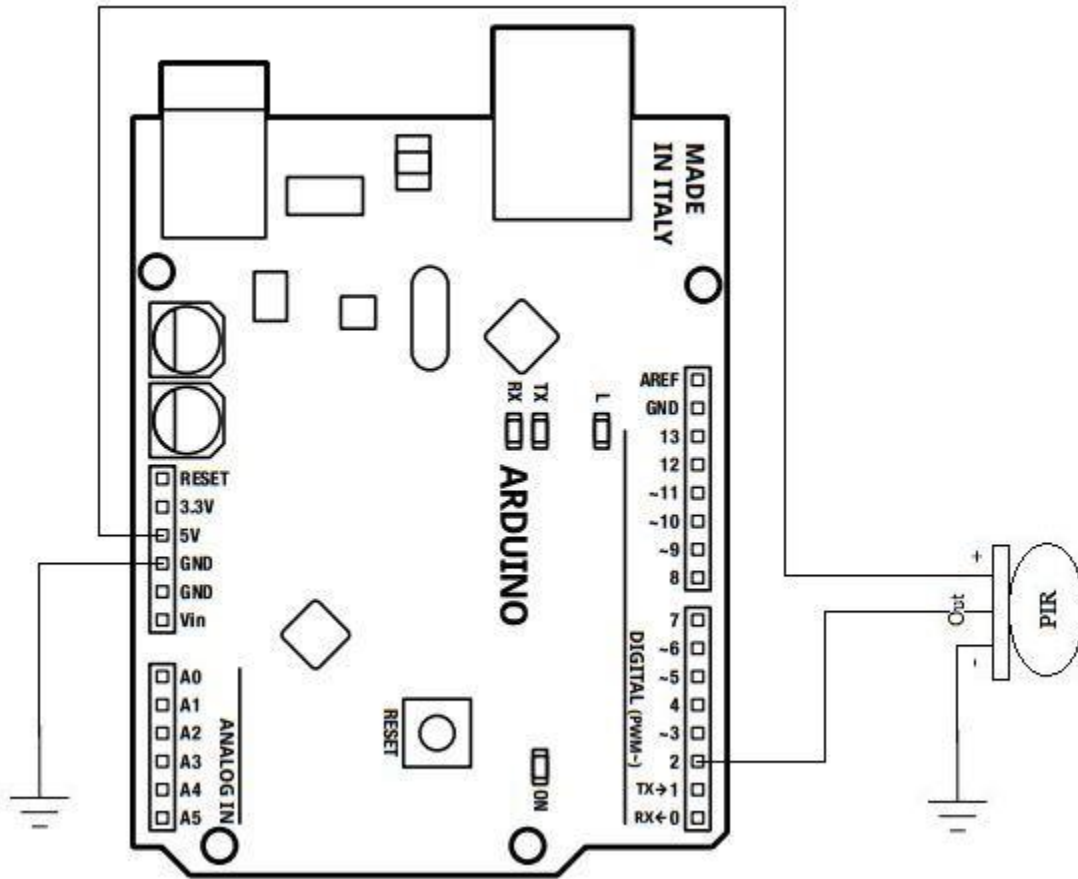


Figure 3.2: PIR Sensor Schematics

3.3 Design of Pedestrian and Traffic Lights System:

The lighting system is made up of eight 10mm LEDs, of which three pairs (2 RED, 2 AMBER and 2 GREEN), each forming a parallel connection is in series with a 50k Ω resistor. These LEDs act as traffic lights for the system. Their anodes are connected to Arduino digital output pins 12, 11 and 10 respectively and the cathodes to ground, this is because LEDs can only light with the right polarity connections made. The remaining two (RED and GREEN) control pedestrian traffic and are each connected in series with a 100k Ω resistor. These are connected on digital output pins 9 and 8 on the microcontroller and the cathodes to ground. Figure 3.3 shows the schematics of the lighting system.

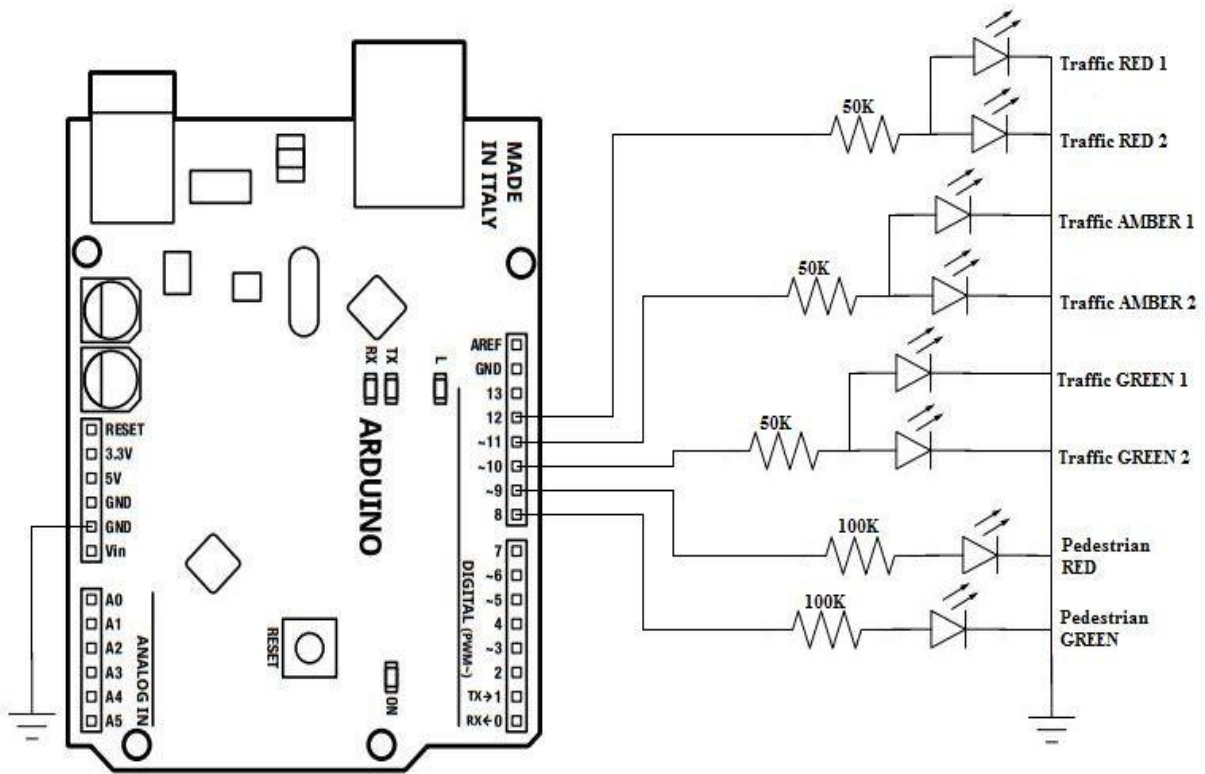


Figure 3.3: Lighting Schematics

3.4 Design of LCD System:

The pin configuration for the LCD is shown in figure 2.19. It has 16 pins but the 4-bit mode can be employed to save pins. For this system a parallel connection was made for the two LCDs. Two 10k Ω potentiometers were used to control the contrast levels, one for each LCD. (VCC and Led+) are connected to the positive 5V of the pot, (VSS, R/W and Led-) are connected to the pot's ground terminal and VO is connected to the slider of the pot for each LCD. The E pins are connected to separate digital pins 4 and 5 for each LCD. The remaining pins RS, DB4, DB5, DB6 and DB7 are connected in parallel on digital output pins 3, 6, 7, 8, 9 of the microcontroller respectively.

Three jumper wires were looped from vehicular traffic lights red, amber and green to digital input pins 12, 11 and 10 respectively of the other Arduino Uno microcontroller with LCDs connected. These pins act as input to this microcontroller such that whenever they are digitally high, a corresponding message is displayed on the two LCDs simultaneously. Figure 3.4 below shows the LCD schematics.

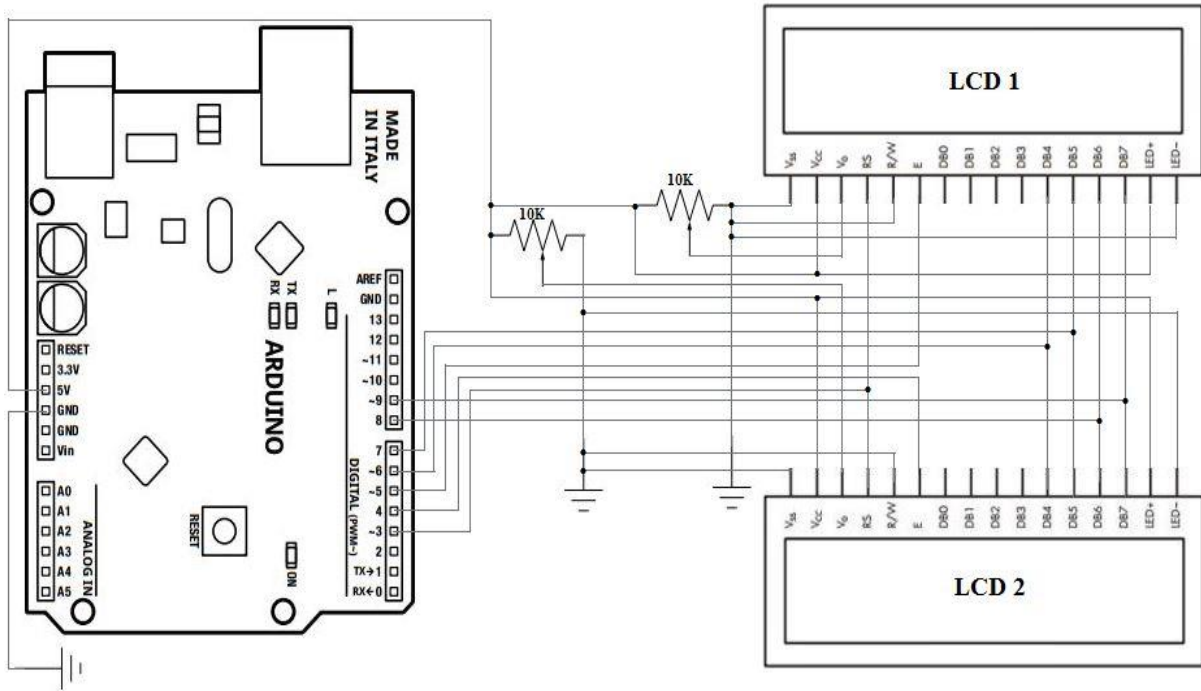


Figure 3.4: LCD System Schematics

3.5 Programming the Arduino Uno:

The IDE was used to write sketches to the Arduino Uno so that it controls these sub components. The A to B USB cable was used to upload sketches to the microcontroller.

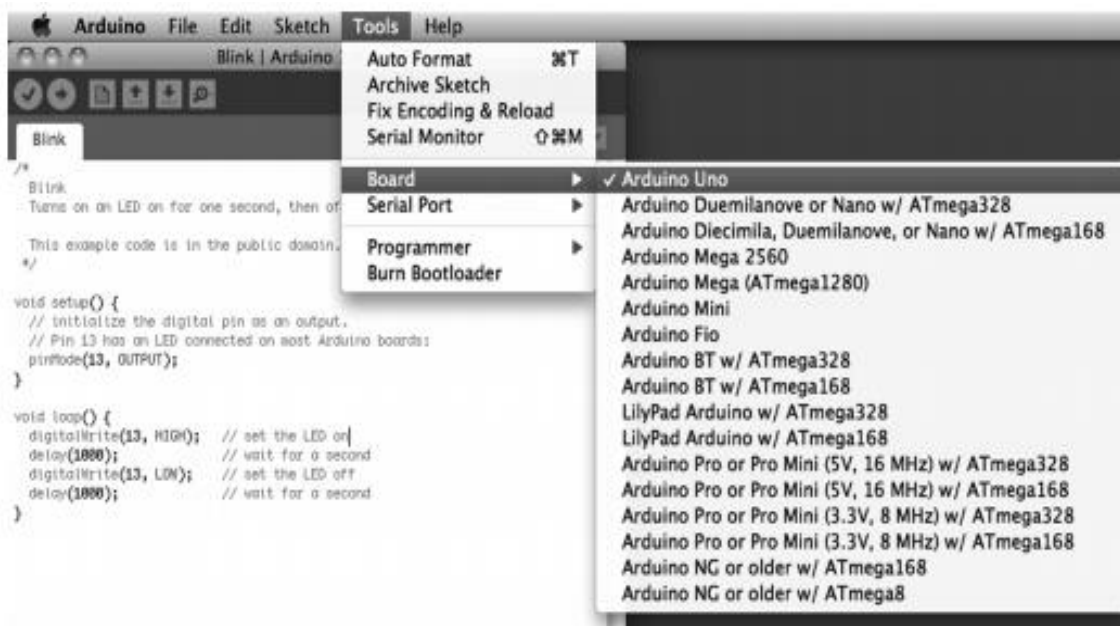


Figure 3.5: Arduino IDE with C++ Code Waiting for Upload [2]



3.6 Complete Circuit Diagram:

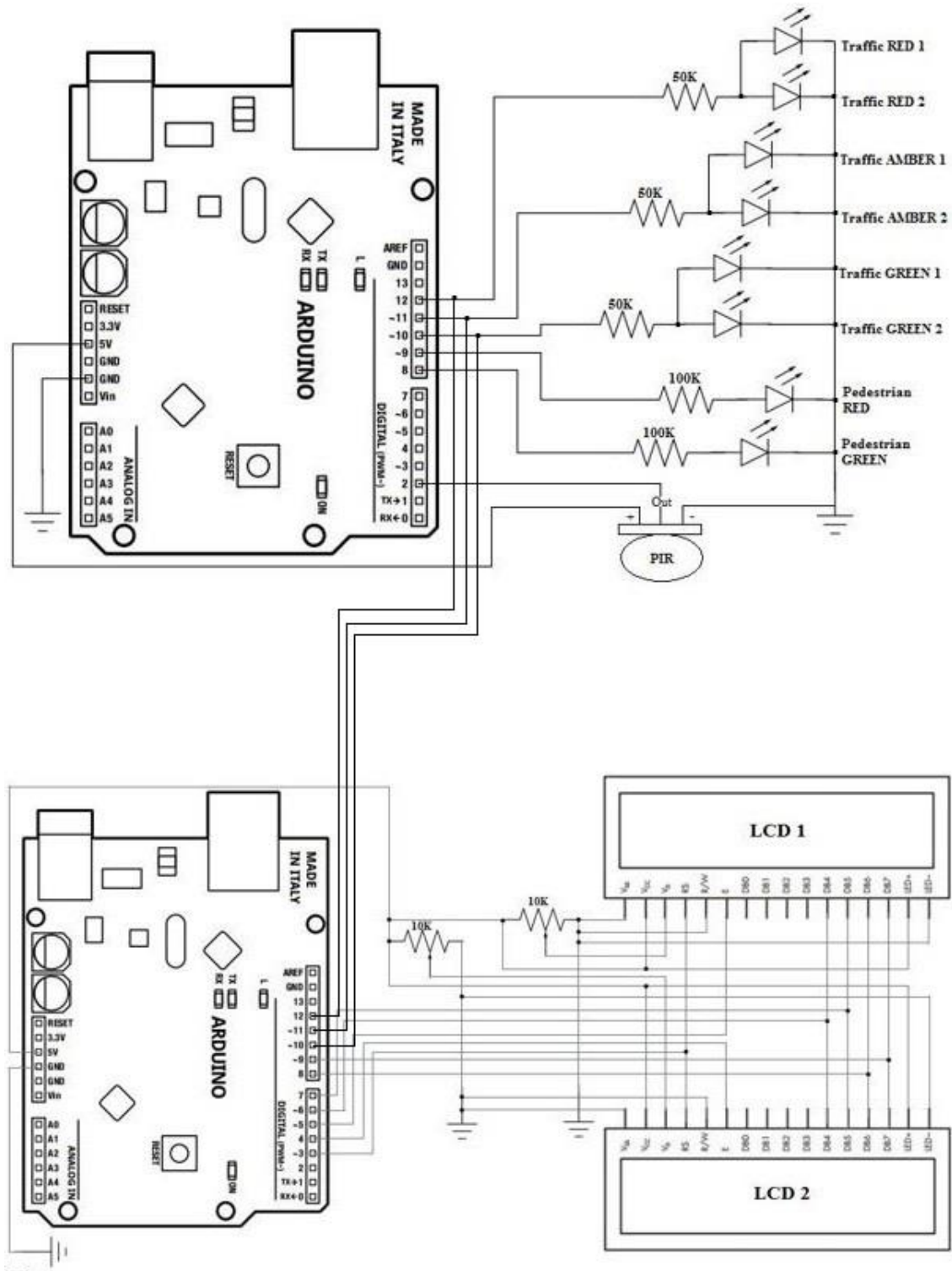


Figure 3.6: Complete Circuit Diagram



3.7 Key Crosswalk Design Elements:

According to the Federal Highway Administration, designing an effective pedestrian crossing involves the correct layout of a variety of standardised elements such as information/signs, signals and markings, crosswalks, crossing times, sight lines and medians [3].

- Advance warning signs were placed 15m from the stop line.
- The heights for the crossing light posts were 2.5m
- A continental crossing with dashed lines was used since it is highly visible.
- Longitudinal lines were 2mx0.3m and the distance between dashed lines was 2.4m.
- Line spacing between crosswalk longitudinal lines were 0.3m to reduce distortion due to tyre marks.
- Crossing time was made 15 seconds using the following equation;

$$\text{Crossing Time} = \frac{\text{Crossing Distance}}{\text{Pedestrian Average Speed}} \dots \dots \dots (3.1)$$

The crossing distance was 15m and pedestrian average speed 1.065m/s.

3.8 Specifications:

Table 3.1: System Input and Output Devices

Input Devices	Output Devices
PIR Sensor	Vehicular Traffic Lights
	Pedestrian Lights
	LCDs

- In the idle state the lights will show red for pedestrians and green for vehicular traffic; LCDs will display a message to alert motorists of the presence of the crossing and to exercise caution.



- When a pedestrian wishes to cross they step on the detection zone and the traffic lights switch from green to amber and waits for three seconds with the pedestrian light red; LCD message simultaneously changes to notify motorists to prepare to stop fifteen meters ahead.
- After the three seconds both lights turn red and have a time delay of three seconds; LCD message now instructs motorists to yield to pedestrians at the crossing.
- After three seconds, the traffic lights remain red, whilst the pedestrian lights turn green for fifteen seconds (pedestrian crossing time).
- After the above time delay, the pedestrian green light blinks for six seconds.
- Both lights switch to the all red state for three seconds again.
- The system is returned to the idle state once again, having the traffic lights green and the pedestrian lights red. .

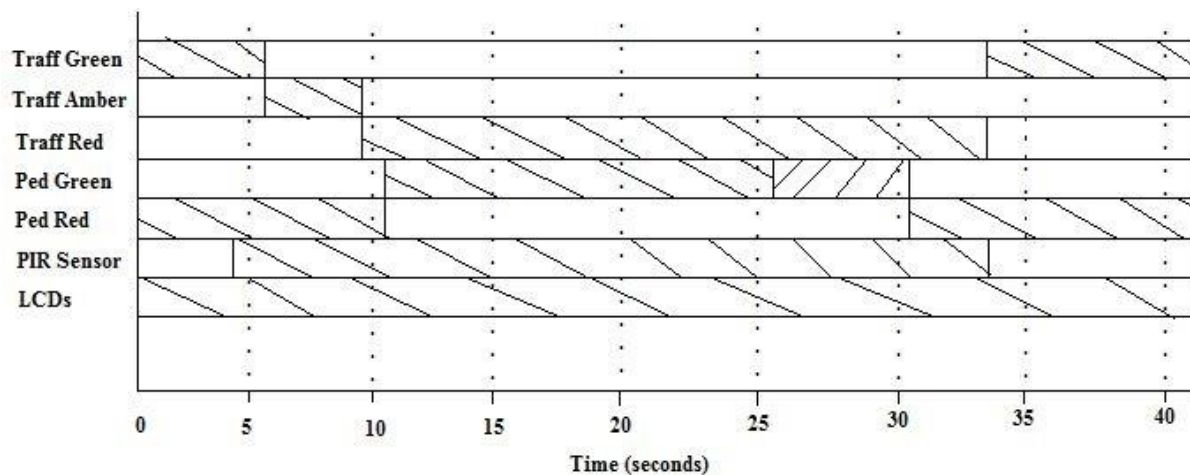


Figure 3.7: Expected Responses for Each Input and Output

3.9 System Flowchart:

Summarises all the stages when the system is functioning.

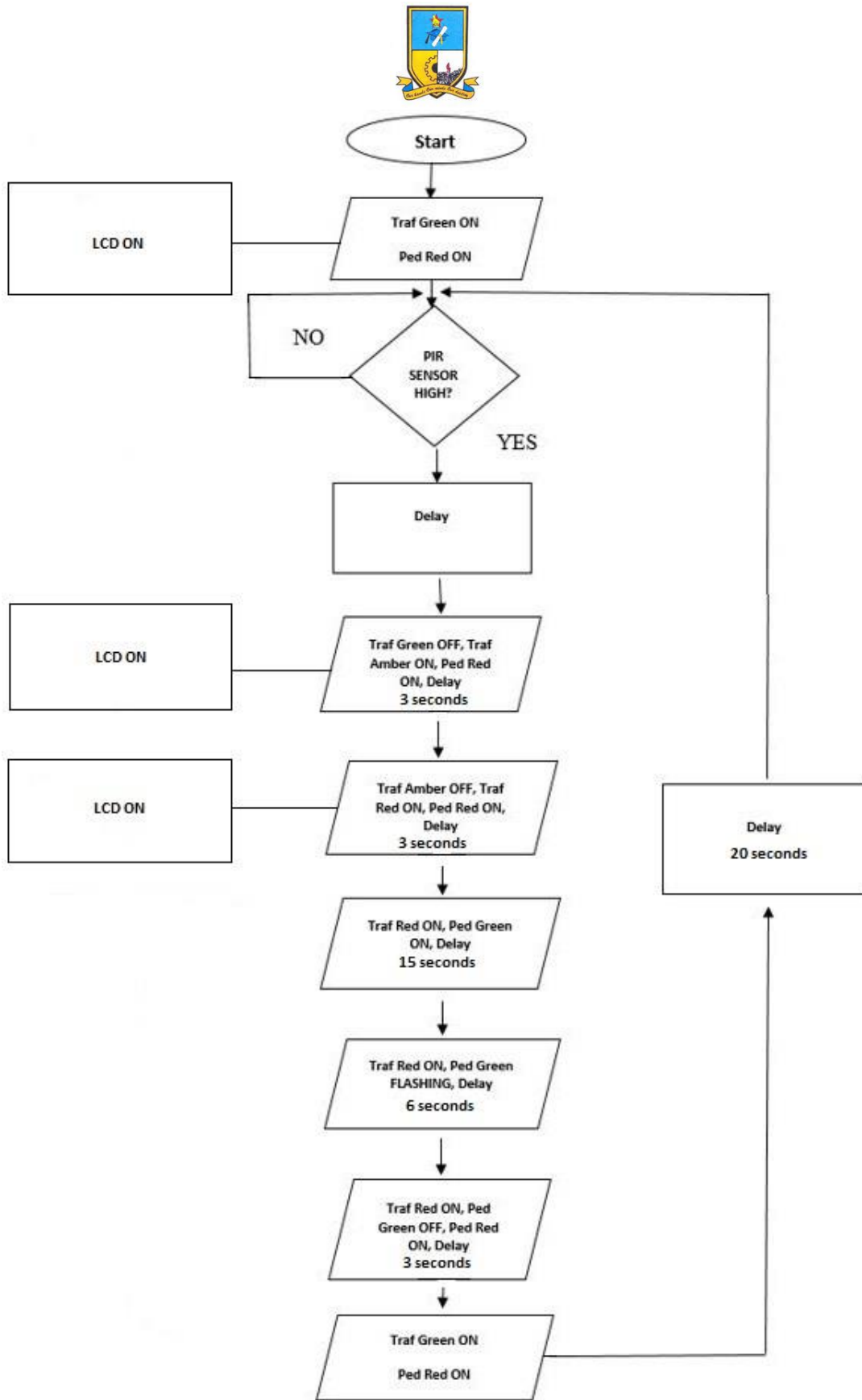


Figure 3.8: System Flowchart



References

- [1] M. Riley, Programming Your Home, North Carolina: The Pragmatic Bookshelf, 2012.
- [2] B. Evans, Beginning Arduino Programming, New York: Apress, 2011.
- [3] F. H. Administration, Designing Sidewalks and Trails for Access Part II of II, Washington ,DC: US Department of Transportation Federal Highway Administration, 2014.



CHAPTER 4

Results

4.0 Introduction:

This chapter focuses on the results derived from the project research, system modelling and preliminary circuit tests.

4.1 Presentation of Results:

Table 4.1: Detection Distance of PIR Sensor and Corresponding Ambient Temperature

Distance(m)		Temperature (°C)
Normal Operation	Reduced Sensitivity	
9.1	4.5	18.4
8.5	3.9	21.1
7.9	3.6	23.8
7.3	3.5	26.7
6.7	3.2	29.4
6.1	3.0	32.2
5.5	2.7	34,8

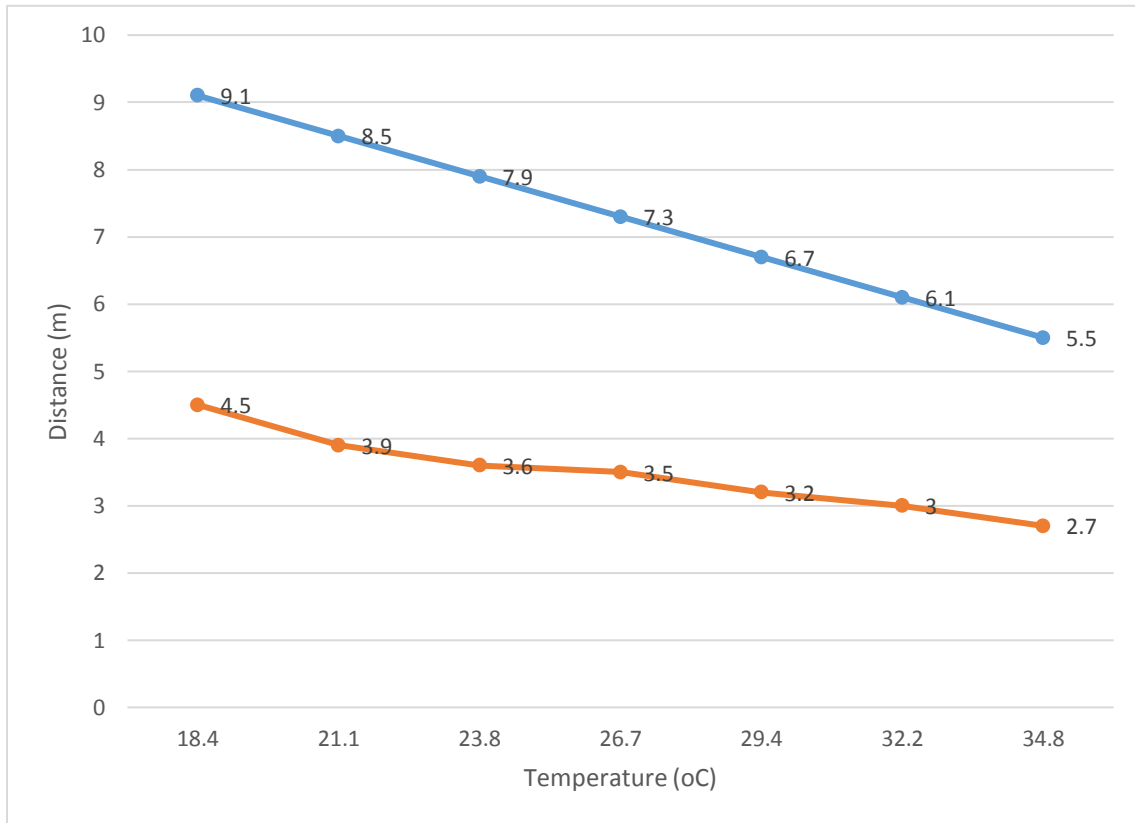


Figure 4.1: Temperature Effect on PIR Sensor Distance Detection

4.2 Analysis and Interpretation of Results:

The results shown in table 4.1 reveal an inverse proportionality relationship between ambient temperature and detection distance. Figure 4.1 goes on to reveal that proper adjustments to the sensitivity of the PIR sensor can yield to longer detection ranges despite the temperature range. This is clearly shown by the plot on normal operation revealing a detection range of 5m to 10m as compared to that of reduced sensitivity ranging from 2.5m to less than 5m on the same ambient temperature range. On the graph the plot for normal operation values has a scaling factor of $-0.22 \text{ m}/^{\circ}\text{C}$ and that of reduced sensitivity is $-0.11 \text{ m}/^{\circ}\text{C}$. The negative sign implies that for every rise in unit temperature the detection distance falls by the allocated margin. The obtained results indeed agree with the theoretical analysis that the range of the PIR sensor is affected by conditions such as:



- Sensitivity setting of the sensor - Turning the sensitivity potentiometer to a minimum results in shorter range about four meters and switching to normal operation results in longer range about nine meters.
- Thermal properties of nearby objects and their sizes.
- Environmental conditions incorporating ambient temperature and light sources.

This relationship between ambient temperature and detection distance is very important for the designer to know the effects of varying seasons thus varying ambient temperatures on the performance of the pedestrian crossing control system. Thus during maintenance the sensitivity of the system should be checked from time to time.

4.3 Overall System Analysis:

- The 16x2 LCD cannot accommodate many word at one go thus needs a bigger one to show more detail nonetheless it displays quite clearly.
- The system proved effective for it was able to allocate crossing time for pedestrians as well as vehicular traffic automatically.
- The PIR sensor's sensitivity must be checked from time to time so that it remains on optimum levels.
- If pedestrian remains stationery in the detection zone whilst the sensor is calibrating itself they might not be detected.
- Pedestrians require a visual display unit or a timer to keep track of the crossing time lapses and not only depend on crosswalk lights.
- Pedestrians should not wear heat-insulating full-body suit or Styrofoam material for them to be detected by the PIR sensor.
- The project results proved that the system is quite practicable.



Figure 4.2: Arduino Based Pedestrian Crossing Control System



CHAPTER 5

Conclusion

5.0 Introduction:

The designing of a prototype for the automated pedestrian crossing control system was successful. The prototype worked perfectly using the concept of passive pedestrian detection and allocated pedestrians some crossing time without an interference from vehicular traffic. The LCDs functioned perfectly displaying the alert message to the motorists.

5.1 Discussion:

Incorporating passive pedestrian detection and LCDs in pedestrian crossing facilities proves to be very effective. Less material is required to set up the whole controlling system and it is automated as well as clearly visible thus human error is greatly minimised. In turn probability of pedestrian crash rates significantly reduced. In this project only one PIR sensor was used to detect pedestrians and a set of ordinary traffic lights to regulate vehicular and pedestrian traffic as well as a pair of LCDs to act as an alert signal to motorists. The setup worked quite well however, on the actual location the researcher found out that there is an island demarcating the carriageway in two. Thus this set up might mean that two crossing facilities need to be installed to cater for each dual lane to further enhance ease of access for the visitors at the hospital.

5.2 Limitations of the PIR Sensor:

- The sensing can be misinterpreted if
 - Motion too close to sensor.
 - Motion is passed through the sensor all at once and not one at a time.
 - Object is slow moving object and stationary objects cannot be detected.



- The sensor is approached straight on.
- Most sensors have limited range and are temperature sensitive.

5.3 Recommendations:

- The crossing facility with a few more tweaks or upgrades can be used by school children at Senga Primary School when crossing the Senga – MSU road.
- Interfacing Arduino Uno with a DC motor for the boom gates posed a major challenge for the researcher because the Arduino library does not provide tilt angles for DC motors but only servo motors which are rarely found in Zimbabwe. Thus the researcher recommends that servo motors be readily made available for future projects.
- Also in order to make maximum use of Arduino, a variety of auxiliary shields need to be readily available for example the researcher could not use the time library because a special shield which works with it was not available.
- A module supporting C++ programming language only confined to Arduino IDE should be made available for future researchers to be able to improve designs.

5.4 Recommendations for Future Research:

The prototype can be improved by:

- Incorporating Accessible Pedestrian Signals (APS) that include speakers near the sensor to alert the blind.
- Using tactile cone surfacing for the visually impaired.
- Use approach radar/ electromagnetic detectors buried in the road surface to measure speed of oncoming traffic and turn lights red to stop vehicular traffic (enforce speed limit).
- Installing count down timers to notify drivers and pedestrians on the time remaining on the crossing.
- Introduce web cameras to see those who violate the law.
- Link the crosswalk system with Tawanda Mugwiji's system to reduce crime rate.



- Incorporate boom gates to stop vehicular traffic.
- A timer to continuously monitor the crossing times.
- Incorporate crosswalk lights to illuminate the crossing area maybe at night.
- LCDs to alternate colour on messages as well as the text itself.
- Incorporating real time chips so that the crossing facility can be used precisely during visiting hours.



Appendix

A. Crosswalk Lights Control Sketch:

```
// Project Synergy - Crosswalk Lights

int trafRed = 12; // assign traffic lights

int trafYellow = 11;

int trafGreen = 10;

int pedRed = 9; // assign pedestrian lights

int pedGreen = 8;

int pirSensor = 2; // sensor pin

int pirState = 0; // pirSensor status

int crossTime = 15000; // time for pedestrians to cross

unsigned long changeTime = 0; // time last pedestrian cycle completed

void setup() {

pinMode(trafRed, OUTPUT);

pinMode(trafYellow, OUTPUT);

pinMode(trafGreen, OUTPUT);

pinMode(pedRed, OUTPUT);

pinMode(pedGreen, OUTPUT);

pinMode(pirSensor, INPUT); // sensor on pin 2
```



```
}
```

```
void loop() {
```

```
  pirState = digitalRead(pirSensor);
```

```
  /* check if sensor is triggered and it is over 20 seconds since last trigger */
```

```
  if (pirState == 1 && (millis() - changeTime)> 20000)
```

```
  { // Call the function to change the lights
```

```
    changeLights();
```

```
  }else
```

```
  {
```

```
    digitalWrite (trafGreen, HIGH);
```

```
    digitalWrite (pedRed, HIGH);}
```

```
  }
```

```
void changeLights() {
```

```
  digitalWrite(trafGreen, LOW); // green off
```

```
  digitalWrite(trafYellow, HIGH); // yellow on
```

```
  delay(3000); // wait 3 seconds
```

```
  digitalWrite(trafYellow, LOW); // yellow off
```

```
  digitalWrite(trafRed, HIGH); // red on
```



```
delay(3000); // wait 3 second till its safe

digitalWrite(pedRed, LOW); // ped red off

digitalWrite(pedGreen, HIGH); // ped green on

delay(crossTime); // wait for preset time period

// flash the ped green

for (int x=0; x<7; x++) {

digitalWrite(pedGreen, HIGH);

delay(500);

digitalWrite(pedGreen, LOW);

delay(500);

}

// turn ped red on

digitalWrite(pedRed, HIGH);

delay(3000);

digitalWrite(trafRed, LOW); // red off

digitalWrite(trafGreen, HIGH);

// record the time since last change of lights
```



```
changeTime = millis();  
  
// then return to the main program loop  
  
}
```

B. LCD Sketch:

```
#include <LiquidCrystal.h> //include library code  
  
LiquidCrystal lcd1(3,5,6,7,8,9);//initialise library with numbers of interface pins  
  
LiquidCrystal lcd2(3,4,6,7,8,9);  
  
int trafRed = 12; // Red traffic lights  
  
int trafYellow = 11; // Amber traffic lights  
  
int trafGreen = 10; // Green traffic lights  
  
int trafR =0;  
  
int trafY =0;  
  
int trafG =0;  
  
void setup() {  
  
pinMode(trafRed, INPUT);  
  
pinMode(trafYellow, INPUT);  
  
pinMode(trafGreen, INPUT);  
  
lcd1.begin(16,2);//set up the LCD columns against rows  
  
lcd1.clear(); //clear LCD
```



```
lcd2.begin(16,2);

lcd2.clear();//clear

}

void loop() {

  trafG =digitalRead(trafGreen);

  if(trafG ==1)

  {

    lcd1.setCursor(3,0);

    lcd1.print("PEDESTRIAN");

    lcd1.setCursor(1,1);

    lcd1.print("CROSSING AHEAD");

    delay(1000);

    lcd1.clear();

    lcd1.setCursor(4,0);

    lcd1.print("PROCEED");//print message to LCD1

    lcd1.setCursor(2,1);//start text at row 2 column 1

    lcd1.print("WITH CAUTION");

    delay(1000);

    lcd1.clear();

    lcd2.setCursor(3,0);
```




```
lcd2.print("PEDESTRIAN");  
  
lcd2.setCursor(1,1);  
  
lcd2.print("CROSSING AHEAD");  
  
delay(1000);  
  
lcd2.clear();  
  
lcd2.setCursor(4,0);  
  
lcd2.print("PROCEED");//print message to LCD2  
  
lcd2.setCursor(2,1);//start text at row 2 column 1  
  
lcd2.print("WITH CAUTION");  
  
delay(1000);  
  
lcd2.clear();  
  
}
```

```
trafY =digitalRead(trafYellow);  
  
if(trafY ==1)  
  
{  
  
lcd1.setCursor(3,0);  
  
lcd1.print("PREPARE TO");  
  
lcd1.setCursor(1,1);  
  
lcd1.print("STOP 15M AHEAD");  
  
delay(3000);
```



```
lcd1.clear();
```

```
lcd2.setCursor(3,0);
```

```
lcd2.print("PREPARE TO");
```

```
lcd2.setCursor(1,1);
```

```
lcd2.print("STOP 15M AHEAD");
```

```
delay(3000);
```

```
lcd2.clear();
```

```
}
```

```
trafR =digitalRead(trafRed);
```

```
if(trafR ==1)
```

```
{
```

```
lcd1.setCursor(1,0);
```

```
lcd1.print("PLEASE YIELD TO");
```

```
lcd1.setCursor(1,1);
```

```
lcd1.print("PEDESTRIANS");
```

```
delay(3000);
```

```
lcd1.clear();
```

```
lcd2.setCursor(1,0);
```



```
lcd2.print("PLEASE YIELD TO");
```

```
lcd2.setCursor(1,1);
```

```
lcd2.print("PEDESTRIANS");
```

```
delay(3000);
```

```
lcd2.clear();
```

```
}
```