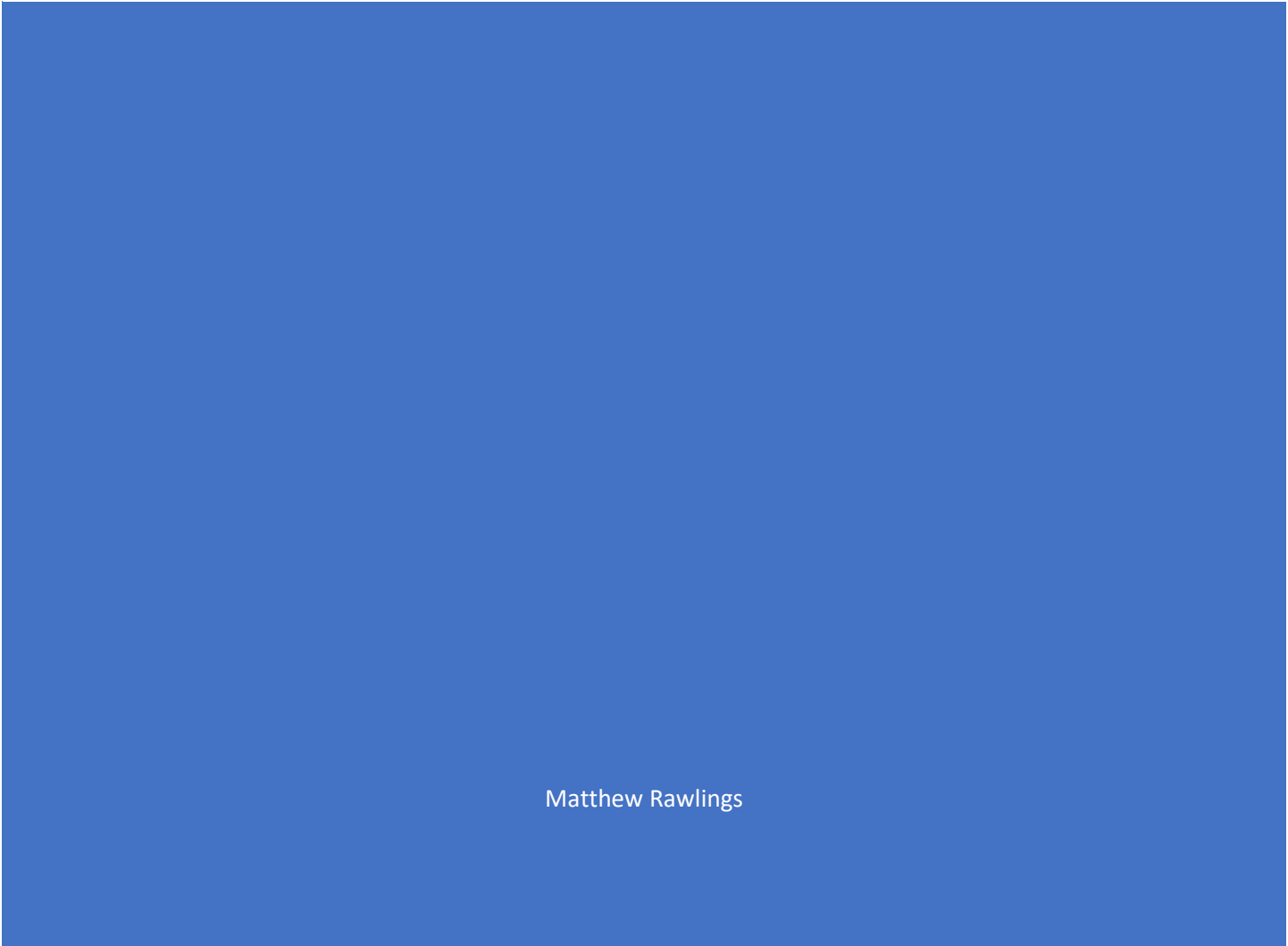




HNC ELECTRICAL/ELECTRONIC ENGINEERING, POWER SYSTEMS ENGINEERING & INSTRUMENTATION ENGINEERING

EMBEDDED SYSTEMS



Matthew Rawlings

Contents

Introduction	2
Task 1 - Hardware Design and components	2
Arduino UNO	2
RGB LED/LED	2
Piezo Buzzer	3
16 X 2 LCD	3
Ultrasonic Distance Detector	3
Task 2 – Software Design	4
Task 3- Implementation	6
Task 4 - Testing.....	9
Task 5 - Evaluation	11
Bibliography	12
Appendices.....	13
ArduinoUNO.....	13
LED	14
Piezo Buzzer	15
16 x 2 LCD.....	17
Ultrasonic Distance Detector	18

Introduction

Embedded systems explore a variety of microcontrollers and processors which have the ability to perform a large number of functions. The software functions can operate a number of hardware components including electrical, mechanical and instrumentation. An Embedded system is a software which is embedded in a computer hardware, dedicated to either a single action/application or multiple applications for a larger system. This paper will explore the application of an Arduino Uno and the hardware used to build a program to a single specification. The program, software and hardware will be implemented and tested under expected operating conditions. The outcome and response of the system will be evaluated, comparing to the specification. Any improvements which could be made to the system will be noted and expanded on how it will improve the system.

Task 1 - Hardware Design and components

Arduino UNO

The Arduino UNO is a microcontroller which can be used in a variety of applications for single or multiple actions at the same time. It is a compact device which can be used as an embedded system. It uses inputs and outputs to perform its functions with the correct coding. The Arduino board is able read analogue and digital inputs, as well as provide these two as outputs. The Arduino is useful as it is relatively inexpensive and easy to access, using a USB as its communication. The use of the Arduino UNO in this project is to read the code and control multiple outputs such as the Piezo Buzzer, LCD and RGB LED all will be outlined below.

<https://uk.rs-online.com/web/p/arduino/7154081/> - data sheet in Appendix

RGB LED/LED

An LED is a component which can be used as an indication. As a diode, it allows current flow through one direction. This can be an indication of power to a larger component or whether a supply has failed. An LED will only be able to show one colour, red, orange, green etc. The RGB LED or Red Green Blue LED has the ability to show all colours, using one component. This will maximise the space on a breadboard and one RGB LED can be used to show for example red, amber and green. In the assignment the LED utilised will be RGB in order to save space on the breadboard and will smoothly move from red through to green or the colours chosen.

<https://uk.rs-online.com/web/p/leds/9047367/> - data sheet in Appendix

Piezo Buzzer

A buzzer can also be used as indicationⁱⁱ. For example, in industrial environments, when a process variable changes, an alarm is used to notify an operator when the variables become too high or too low. In this assignment, a buzzer will; be utilised in order to notify when an intruder is within 2 metres of the front door of a house. The buzzer will imitate a house alarm and will not activate when nothing is detected within 2 metres.

<https://uk.rs-online.com/web/p/piezo-buzzer-components/1812658/> - data sheet in Appendix

16 X 2 LCD

The LCD display can be used to inform on whether someone is within 2metres or not. It can do this with a live feed of information on the screen, for example, “206cm, OK!”. This is how the LCD screen will be utilised in this assignment.

<https://uk.rs-online.com/web/p/lcd-monochrome-displays/5326385/> - data sheet in Appendix

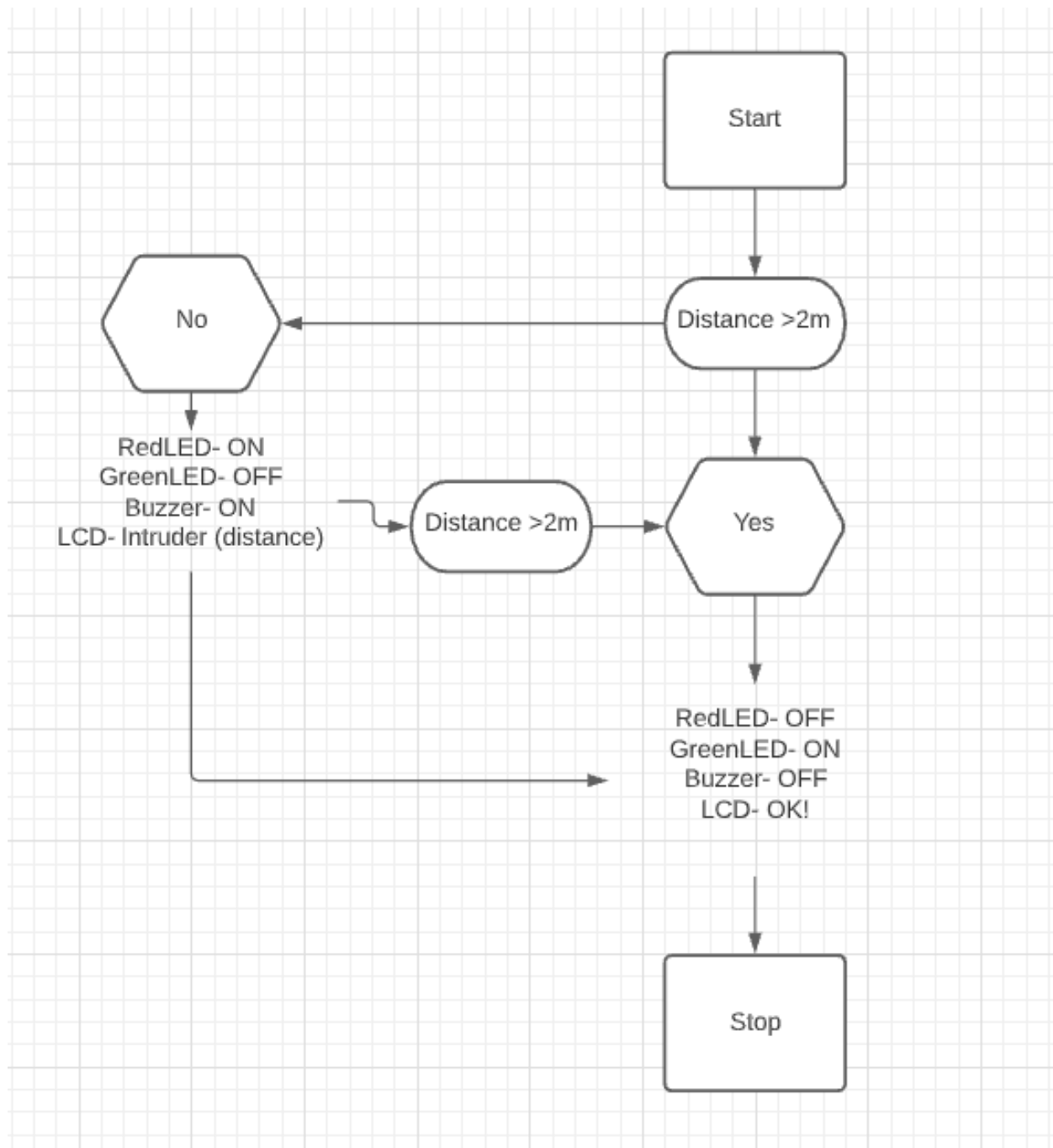
Ultrasonic Distance Detector

An Ultrasonic Distance Detector is a component which can have a number of uses. It can measure process level and distance which each can be used in a variety of areas. The Ultrasonic Detector bases its operation on time of flight so initially gives the signal to the Arduino in microseconds. The correct code has to be written in order to convert this to distance. When done correctly, the detector will give accurate readings on a given position in its range. In this assignment, the detector will be used to detect movement/an object within 2 metres. Upon detection, it will send a signal to the RGB LED and Piezo Buzzer, both previously mentioned.

<https://uk.rs-online.com/web/p/sensor-development-tools/7813020/> - data sheet in Appendix

Basic components include a breadboard to fix wires , resistor and other components to.

Task 2 – Software Design

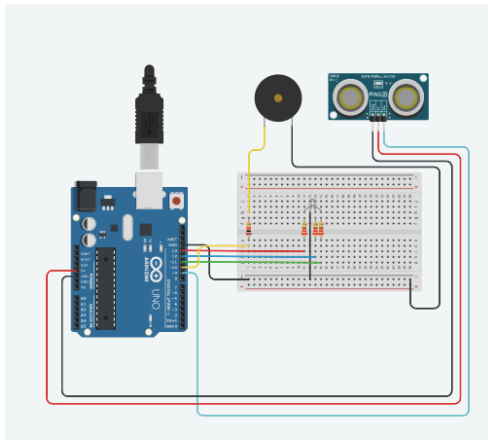


The software used to control the hardware and the Arduino UNO is C Program. C Program is a simple and flexible software which can operate a number of functions in different applications and areas. The flexibility of C program means that it is 'machine independent' so it can be run on any machine while the code stays the same. It is one of thousands of programming languages utilised but is used mostly for its efficiency and control. There are multiple pieces of hardware in this circuit which operate as a result of a change in environment. For example, in this assignment, when an object or person is detected within 2 metres of a front door which will be the Ultrasonic Distance Detectorⁱⁱⁱ, a list of security devices will operate to notify when this happens. When the program is set to run, the Distance Detector will constantly send ultrasonic signals, waiting for a reflection from an object. Using `"cm = microsecondsToCentimetres(duration)"` in order to convert the time taken for a response into a distance. The speed of sound is approximately 340 m/s or 29 microseconds per centimeter. The ping travels out and back, so to find the distance of the object half of the distance travelled must be calculated. This was done using the following line of code `"long microsecondsToCentimeters(long microseconds) { return microseconds / 29 / 2;"`.

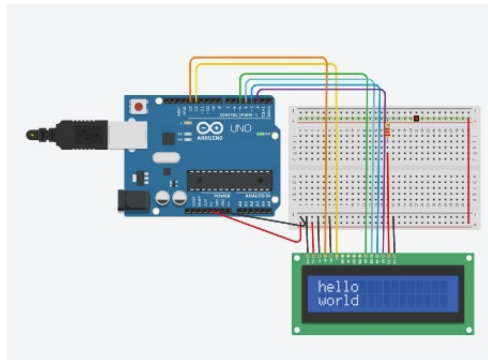
An 'if' function can be used in this scenario so that when a person is detected, the security devices will activate. The security devices which will be included in the hardware are LED^{iv}, buzzer and LCD screen. As stated in task 1, an RGB LED was planned to be used but through investigation, two separate devices is more acceptable. Each of these are planned to activate under an 'if' statement so a 'digitalWrite' and 'analogWrite' can be sent to the outputs of the Arduino UNO. An 'else' statement is also made so that when the ultrasonic signal is not reflected back, the hardware will be in a healthy state.

Task 3- Implementation

The alarm system was created by looking at other small pieces of code which accompanied pieces of hardware. Once the appropriate hardware components were selected, they were put onto a breadboard with the Arduino UNO and wired as initially desired.



Pictured is the initial design of the circuit. Using a RGB LED, changing between red, through to green when the distance exceeded 2 metres. This design was acceptable and functioned as expected with the accompanying code. Upon improving and redesign, other components and hardware were either replaced or included. One example is an LCD display to indicate the distance of the object detected. A replacement made is to add 2 LED's in the place of the RGB.



Shown is an addition to be implemented into the design. This will display the distance of the 'intruder' or tell whether the area is clear. The RGB LED improvement will also be included in the final design.

The void setup () within the program will initialize variables and begin to use libraries. This function will only occur once after the power up and/or reset of the Arduino. Functions and variables used in the code will be initialised in the setup and named with the pin before use, otherwise the name will not be recognised. For example in the code below, ledPin=13;, this names the pin and determines the pin in use.

Line 1 - #include <LiquidCrystal.h> this was taken from the Tinker CAD library

Lines 2 – 9 in the code below are naming and creating constants for the hardware in the circuit.

Creating constants allows a value to be declared in a code and be used multiple times throughout.

Lines 11- 22 include the void setup(). As previously mentioned the setup is where the variables used in the code will be initialised in the setup and named with the pin before use, otherwise the name will not be recognised.

The void loop takes the rest of the code and this sequence will repeat until the code is stopped.

As seen in the comments on lines 24-37, the distance sensor will be triggered by a high pulse of 2 or more microseconds so these lines will allow the ultrasonic sensor to see a change in HIGH to LOW, triggering the security systems when needed.

Lines 45 – 55 in the code start with an ‘if’ statement, followed by (cm<200.0). This means that if the ultrasonic detector receives a distance less than 2 metres, the security devices will operate, the buzzer will sound using tone (BUZZER_PIN) followed by a recognisable note of choice, the red LED will turn on using digitalWrite, and the LCD will display the distance and a message, using lcd.print, these actions will be carried out until the “else” statement is achieved.

The following lines 58- 67 start with an “else” function meaning that if the previous condition is not true, the functions within the “{ }” should be performed. The functions will result in the red LED turning off and the green LED on, the buzzer will silence and the LCD^v will display a neutral message such as area clear or OK. The distance will also be printed on the serial monitor using Serial.print and Serial.println. Used in line 73 long is a data type used in C. A constant or variable defined as long can store a single 64-bit signed integer. In line 78 the return statement is used to terminate the execution of a function and transfer program control back to the calling function. It is able to specify a value returned by the function.

```

1 #include <LiquidCrystal.h>
2 const int SENSOR_PIN = 9;
3 const int LED_PIN=13;
4 const int GREEN_LEDPIN=12;
5 const int BUZZER_PIN=10;
6 const int NoteC = 261; //creates a recognisable note
7 const int COLUMNS=16;
8 const int ROWS=2;
9 LiquidCrystal lcd (2,8,4,5,6,7);
10
11 void setup()
12 {
13   Serial.begin(9600);
14   pinMode(LED_PIN,OUTPUT);
15   pinMode(GREEN_LEDPIN,OUTPUT);
16   pinMode(BUZZER_PIN,OUTPUT);
17   lcd.begin(COLUMNS,ROWS);
18   lcd.print("System");
19   lcd.setCursor(0,1);
20   lcd.print ("Activated");
21   delay(1000);
22 }
23
24 void loop()
25 {
26   long duration, cm; //converting duration to distance
27   pinMode(SENSOR_PIN, OUTPUT); // The sensor is triggered by a HIGH pulse of 2 or more microseconds.
28   digitalWrite(SENSOR_PIN, LOW);
29   delayMicroseconds(3); //to allow time for the HIGH pulse to be detected
30   digitalWrite(SENSOR_PIN, HIGH);
31   delayMicroseconds(4);
32   digitalWrite(SENSOR_PIN, LOW);
33
34   //the pulseIn function will wait for the input to go from LOW to HIGH
35
36   pinMode(SENSOR_PIN, INPUT);
37   duration = pulseIn(SENSOR_PIN, HIGH);
38
39   // converting the time taken for response into a distance
40
41   cm = microsecondsToCentimeters(duration);
42   //Lines below check for a condition
43   //when distance to the sensor is less that 200cm or 2m, the red LED is HIGH and the other LOW
44
45   if (cm<200.0){
46     digitalWrite(LED_PIN,HIGH);
47     digitalWrite(GREEN_LEDPIN,LOW);
48     lcd.clear();
49     tone(BUZZER_PIN,NoteC, 300);
50     lcd.print("Intruder Alert");
51     lcd.setCursor(0,1);
52     lcd.print(cm);
53     lcd.print("cm");
54     delay(500);
55     lcd.clear();
56   }
57   |
58   else {
59     lcd.clear();
60     digitalWrite(LED_PIN,LOW);
61     digitalWrite(GREEN_LEDPIN,HIGH);
62     analogWrite(BUZZER_PIN,0);
63     lcd.print("Ok!");
64     delay(1000);
65     lcd.clear();
66     delay(1000);
67   }
68   Serial.print(cm);
69   Serial.print("cm");
70   Serial.println();
71   delay(100);
72 }
73 long microsecondsToCentimeters(long microseconds) {
74
75   // The speed of sound is 340 m/s or 29 microseconds per centimeter.
76   // The ping travels out and back, so to find the distance of the object take half of the distance travelled.
77
78   return microseconds / 29 / 2;
79 }

```

Task 4 - Testing

Testing is an important step when designing a system similar to the system in this assignment. If designing for a customer, it must work as per the specification.

Condition	Expected	Result
TEST 1 Distance to sensor <2m	Green LED – OFF Red LED – ON Buzzer – ON LCD – Intruder Alert! "distance cm" All of the above to continue until Distance to sensor >2m.	Green LED – OFF Red LED – ON Buzzer – ON LCD – Intruder Alert! "distance cm" All of the above to continue until Distance to sensor >2m. As expected
TEST 2 Distance to sensor >2m	Green LED – OFF Red LED – ON Buzzer – ON LCD – OK!	Green LED – OFF Red LED – ON Buzzer – ON LCD – OK! As expected
TEST 3 Distance = 200.2cm	Green LED – OFF Red LED – ON Buzzer – ON LCD – OK!	Green LED – OFF Red LED – ON Buzzer – ON LCD – OK! As expected
TEST 4 Distance = 199.8cm	Green LED – OFF Red LED – ON Buzzer – ON LCD – Intruder Alert! "distance cm" All of the above to continue until Distance to sensor >2m.	Green LED – OFF Red LED – ON Buzzer – ON LCD – Intruder Alert! "distance cm" All of the above to continue until Distance to sensor >2m. As expected

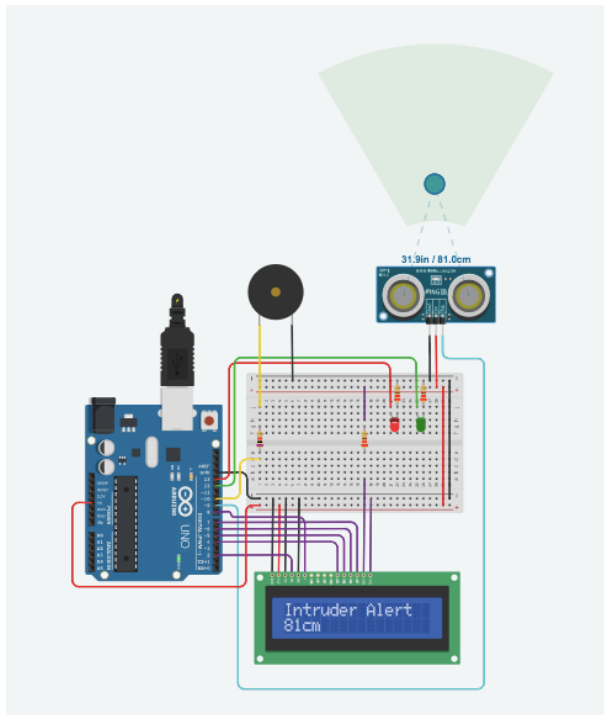


Figure 1 - Test 1

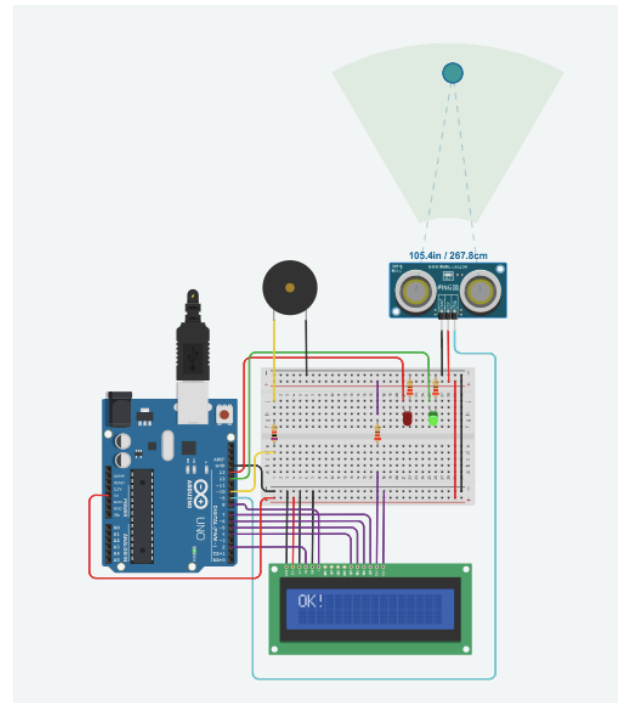


Figure 2 - Test 2

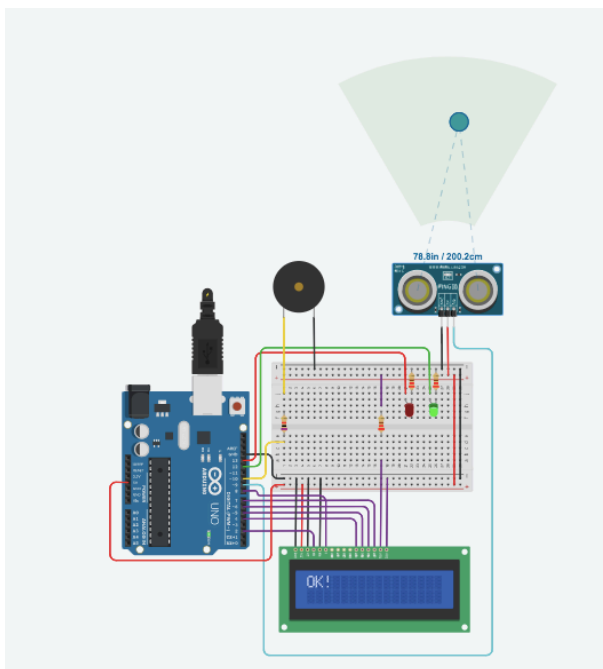


Figure 3 - Test 2

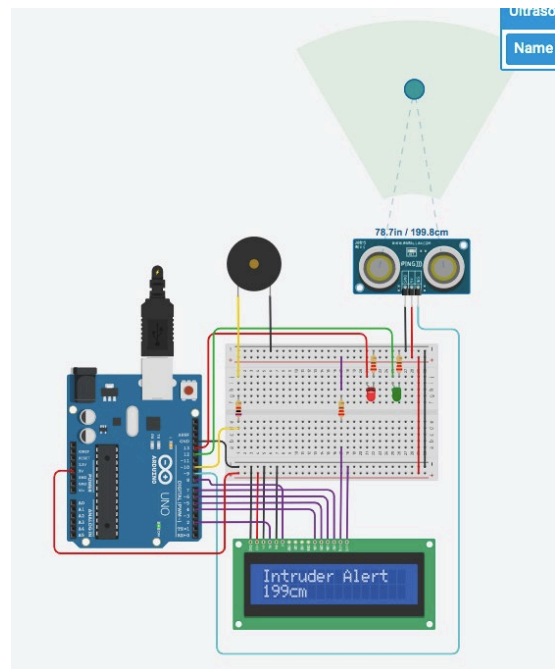


Figure 4 - Test 4

Task 5 - Evaluation

Through the design stages and testing, I have been able to develop an understanding of ArduinoUNO, C programming and a basic software to put these together, TinkerCAD. Through testing I am also able to conclude that I have met the parameters of the design specification. I am pleased with the final design and functionality of the system. Through testing in task 4, I was able to identify that the code and hardware operated as expected. The C programming language was useful in this project as it is a general purpose program which can work efficiently with a number of different applications such as operating systems, data buses and modern programs. Through designing and testing I have decided that in performing this project again, I would incorporate a disarming system. This may be a latching button which disables the system for a duration of time. This could be an option, although, a more appealing system would be a keypad with a password which would disable the system when the correct code is entered.

Bibliography

Guru99. (2021). *What is C Programming Language? Basics, Introduction, History*. Available: <https://www.guru99.com/c-programming-language.html> . Last accessed 27/4/21.

Marshall Brain & Chris Pollette. (). *The Basics of C Programming*. Available: <https://computer.howstuffworks.com/c1.htm> . Last accessed 30/4/21.

DataFlair. (). *Advantages and Disadvantages of C Programming*. Available: <https://data-flair.training/blogs/advantages-and-disadvantages-of-c/> . Last accessed 1/5/21.

Programiz. (). *Learn C Programming*. Available: <https://www.programiz.com/c-programming> . Last accessed 3/5/21.

Clare Hopping. (2021). *What is an embedded system*. Available: <https://www.itpro.co.uk/hardware/30317/what-is-an-embedded-system> . Last accessed 3/5/21.

Appendices

ArduinoUNO

Technical Specification

EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board

The image shows a photograph of an Arduino Uno board with various components labeled with colored boxes and arrows. The labels include:

- TX/RX Leds (blue)
- "Test" Led 13 (yellow)
- digital pins (green)
- USB Interface (red)
- External Power Supply (red)
- ATmega328 (red)
- Power Led (yellow)
- ICSP Header (red)
- Reset Button (purple)
- I2C (green)
- power pins (red)
- analog pins (yellow)

LED

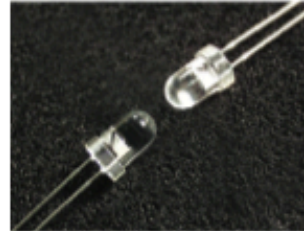
15

Cree® 5-mm Red and Amber Round LED

C503B-RAS/RAN/AAS/AAN

C503B-RBS/RBN/ABS/ABN

C503B-RCS/RCN/ACS/ACN



PRODUCT DESCRIPTION

Round LEDs offer superior light output for excellent readability in sun-light and dependable performance. They provide extremely stable light output over long periods of time.

These lamps are made with an advanced optical-grade epoxy offering superior high-temperature and high-moisture-resistance performance in outdoor signal and sign applications.

FEATURES

- Size (mm): 5
- Color and Typical Dominant Wavelength:
Red (624nm)
Amber(591nm)
- Luminous Intensity (mcd)
C503B-RAS/RAN:
(5860 - 23500)
C503B-RBS/RBN:
(3000 - 12000)
C503B-RCS/RCN:
(3000 - 12000)
C503B-AAS/AAN:
(5860 - 23500)
C503B-ABS/ABN:
(3000 - 12000)
C503B-ACS/ACN:
(4180 - 12000)
- Viewing angle:
C503B-RAS/RAN/AAS/AAN:
15 degree
C503B-RBS/RBN/ABS/ABN:
23 degree
C503B-RCS/RCN/ACS/ACN:
30 degree
- Lead - Free
- RoHS Compliant

APPLICATIONS

- Electronic Signs & Signals (ESS)
- Motorway Signs
- Variable Message Sign (VMS)
- Advertising signs
- Petrol Signs
- Amusement

Piezo Buzzer

Datasheet**RS PRO Piezo Audio Indicator**

EN

**A. SCOPE**

This specification applies piezo audio indicator, 1812658

B. SPECIFICATION

No.	Item	Unit	Specification	Condition
1	Resonant frequency	Hz	3800~4800	
2	Operating Volt. range	VDC	4.0 ~ 8.0	
3	Current consumption	mA	MAX 10	at 5VDC
4	Sound pressure level	dB	MIN 83	at 10cm/5VDC
5	Rated Voltage	VDC	5	
6	Tone		Continuous	
7	Operating temp.	℃	-35 ~ +85	
8	Storage temp.	℃	-40 ~ +90	
9	Dimension	mm	φ 13.9 x H7.3	See appearance drawing
10	Weight (MAX)	gram	1.0	
11	Material		PBT (BLACK)	
12	Terminal		Pin type (Plating Au)	See appearance drawing
13	Environmental Protection Regulation		RoHS2.0	

16 x 2 LCD

Displaytech Ltd**LCD MODULE****162B SERIES****Version : 1.2****P.3 of 17****■ ABSOLUTE MAXIMUM RATINGS** (Ta = 25°C)

Parameter	Symbol	Min	Max	Unit
Supply voltage for logic	VDD	-0.3	7.0	V
Supply voltage for LCD	VDD - VO	-0.3	VDD+0.3	V
Input voltage	VI	-0.3	VDD+0.3	V
Normal operating temperature	TOP	0	50	°C
Normal storage temperature	TST	-10	60	°C
Wide operating / storage temperature (except FSTN)	TOP / TST	-30	80	°C
Wide operating / storage temperature (FSTN)	TOP / TST	-30	70	°C

■ ELECTRICAL CHARACTERISTICS (VDD = +5V±10% , VSS = 0V, Ta = 25°C)**DC Characteristics**

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply voltage for logic	VDD	---	4.5	5.0	5.5	V
Supply current for logic	IDD	---	---	1.38	3	mA
Operating voltage for LCD	VDD - VO	25°C	4.5	4.8	5.1	V
Input voltage 'H' level	VIH	---	2.2	---	VDD	V
Input voltage 'L' level	VIL	---	-0.3	---	0.6	V

● Backlight operating information (Ta = 25°C)

LED Backlight	Supply voltage VF (V)			Supply current IF (mA)		
	Min	Typ	Max	Min	Typ	Max
Light box Y/G (-2)	---	4.2	4.6	---	80	120
White (-3LP)	---	3.4	3.5	---	20	25
Blue (-4LP)	---	3.4	3.5	---	20	25
Green (-5LP)	---	3.4	3.5	---	20	25
Amber (-6LP)	---	1.8	1.9	---	20	25
EL Backlight	EL Enable voltage EON (VAC)			EL frequency LF (Hz)		
	Min	Typ	Max	Min	Typ	Max
EL (B)	---	100	150	---	400	1000

AC Characteristics

● Write mode

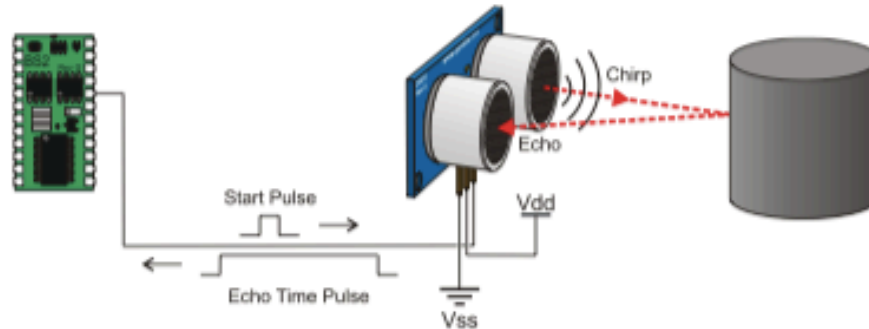
Characteristic	Symbol	Min.	Typ.	Max.	Unit	Test pin
E cycle time	t _c	500	---	---	ns	E
E rise time	t _r	---	---	25	ns	E
E fall time	t _f	---	---	25	ns	E
E pulse width (High, Low)	t _w	220	---	---	ns	E
R/W and RS set-up time	t _{su1}	40	---	---	ns	R/W, RS
R/W and RS hold time	t _{h1}	10	---	---	ns	R/W, RS
Data set-up time	t _{su2}	60	---	---	ns	DB ₀ ~ DB ₇
Data hold time	t _{h2}	10	---	---	ns	DB ₀ ~ DB ₇

Ultrasonic Distance Detector

PING))) Ultrasonic Distance Sensor (#28015)

The Parallax PING)))™ ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to microcontrollers such as the BASIC Stamp®, Propeller chip, or Arduino, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.



Features

- Range: 2 cm to 3 m (0.8 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2 μ s min, 5 μ s typ.
- Echo pulse: positive TTL pulse, 115 μ s minimum to 18.5 ms maximum.
- RoHS Compliant

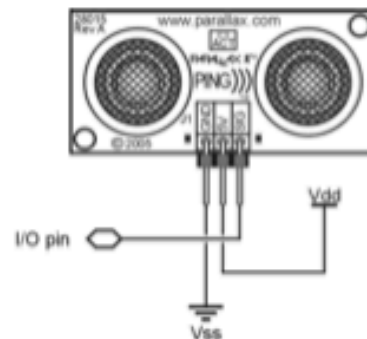
Key Specifications

- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged directly into solderless breadboard, or into a standard 3-wire extension cable (Parallax part #800-00120).



ⁱ <https://uk.rs-online.com/web/p/arduino/7154081/>

ⁱⁱ <https://uk.rs-online.com/web/p/piezo-buzzer-components/1812658/>

ⁱⁱⁱ <https://uk.rs-online.com/web/p/sensor-development-tools/7813020/>

^{iv} <https://uk.rs-online.com/web/p/leds/9047367/>

^v <https://uk.rs-online.com/web/p/lcd-monochrome-displays/5326385/>

STUDENT FEEDBACK FORM

This section details the *extent* to which the assessment criteria are demonstrated by you, which in turn determines your mark. The marks available for each category of skill are shown. Lecturers will use the space provided to comment on the achievement of the task(s), including those areas in which you have performed well and areas that would benefit from development/improvement.

Generic Assessment Criteria	Marks available	Marks awarded
1. Engagement with Literature Skills		
	0	
2. Knowledge and Understanding Skills		
Hardware design 11/ 15 Excellent knowledge demonstrated of the hardware used with clear coherent reasons given for choices made Software design 8/15 Flowchart is missing any indication of setup code required, and does not use the appropriate shapes for decision/process etc. However, an overview of the software's function can be ascertained from the flowchart. You have a good coverage of how the software operates the hardware, but this section reads in the past tense, and software design should be done prior to software development.	30	19
3. Cognitive and Intellectual Skills		
Discussion, evaluation and conclusion Reasonable attempt at evaluation of the underlying concepts and principles, not as full a discussion as other parts of the assignment. Possibly due to word count limitations at this point as generally this assignment shows that it has been very well considered in the design phase. Some valid suggestions are made for improvement along with consideration of the usefulness of C programming language.	15	8
4. Practical Application Skills		
Implementation 30/40 Code is well written with self-annotating variables, use of constants where appropriate and comments.	50	37

You have gained further marks in this section for the inclusion of a function that converts time to distance for the sensor. testing 7/10 The inclusion of a test outside just outside the 2m range on the boundary and just inside the 2m is a useful boundary test to ensure you program operates to the given brief. I would also consider testing what would happen if the alarm is triggered at the 2m and then the intruder moves out of range, I feel this is implied in your testing- but not explicitly stated. A testing table is a useful way to show what areas are being tested in a more compact fashion, well done.		
5. Transferable Skills for Life and Professional Practice		
Structure of report Work is coherent, very fluent and is presented professionally. Datasheets for the components selected are included in the appendices . Code layout is indented on the most part correctly and makes good use of self-annotation techniques.	5	4

Assessment Mark (Assessment marks are subject to ratification at the Exam Board. These comments and marks are to give feedback on module work and are for guidance only until they are confirmed.)	Late Submission Penalties (tick if appropriate)		68%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		