

EMBEDDED SYSTEMS ASSIGNMENT

Power Systems Engineering & Instrumentation
Engineering

HNC Electrical/Electronic Engineering, Power Systems Engineering
& Instrumentation Engineering

PCEE4004

Griffiths, Tomas (Contractor)

Contents

Introduction	2
Hardware Design.....	2
Piezo Buzzer	2
Ultrasonic Distance Sensor	4
80×60×10mm Breadboard	5
Arduino Uno Rev 3	5
16×2 LCD	6
Breadboard Jumper Wire.....	7
RGB LED.....	7
250Ω Resistor	8
Software Design	9
Implementation	11
Testing.....	14
Discussion, Evaluation and Conclusions	17
Hardware Design Datasheets.....	18
Appendix	19

Introduction

An Embedded System is a microcontroller that forms part of a device or machine used to control a certain function within it. Embedded systems are used in many items such as dishwashers, tumble dryers and ovens, mobile phones and network receivers. Their purpose is to control the device it is utilised in and to allow the user to interact with it. This assignment will entail a simple alarm system design that will set off an alarm if the ultrasonic distance sensor detects an object within 2m. An LED will also be used to indicate that an object has been detected.

Task 1 – Hardware Design [15 marks]

Hardware Design

As part of designing a microcontroller-based system, certain parts and components are required. These parts and components are based off the fact that the embedded system design is an alarm system that is used to detect an object within 2m of the sensor which buzzes and has LED indication. The parts required for this system are as follows. The datasheets for these components can be found through the hyperlinks at the back of the report.

Piezo Buzzer

The piezo Buzzer is a device used to provide an audible cue when the appropriate conditions have been changed. In this embedded system, the piezo buzzer is used to emit sound when an object has been detected within 2m of the ultrasonic distance sensor. The specifications for the buzzer chosen can be seen in figure 1 and show that they meet the criteria for the embedded system that will be used.

Attribute	Value
Mounting Type	Panel Mount
Minimum Supply Voltage	1.5V dc
Maximum Supply Voltage	20V dc
Sound Level	96dB
Drive Type	Internal
Tone Type	Continuous
Diameter	41.8mm
Height	16mm
Dimensions	41.8 x 16mm
Minimum Operating Temperature	-20°C
Maximum Operating Temperature	+60°C
Maximum Frequency	3.3kHz
Minimum Frequency	2300Hz
Supply Current	10mA

Figure 1 - Shows the specifications of the buzzer



Figure 2 - Shows the Piezo Buzzer

Ultrasonic Distance Sensor

An Ultrasonic Distance Sensor uses ultrasonic sound waves to measure the distance of an object. Ultrasonic Distance Sensors work by sending out a sound wave which determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse. In the embedded system being used an object needs to be detected if it comes within 2m of the sensors vicinity. Once an object has been detected the sensor will then enable the buzzer, RGB LED and the LCD to carry out their roles in the system. The Key Features for the sensor chosen can be seen in figure 3 and show that they meet the criteria for the embedded system that will be used. When deciding on what sensor to use, it was taken into consideration that the distance in which an object needed to be detected was 2m. From the Key Features in figure 3 it can be seen that the sensor will be capable of detecting an object within 2m.

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**

Figure 3 - Shows the Key Features of the Ultrasonic Sensor Module



Figure 4 - Shows the Ultrasonic Distance Sensor

80×60×10mm Breadboard

Breadboard is used so that it is possible to create and test a circuit before any soldering has been done. Breadboard enables connections between parts and components such as LED's, Resistors, Diodes and capacitors.



Figure 5 - Shows the 80×60×10mm Breadboard

Arduino Uno Rev 3

An Arduino Uno is a microcontroller board based on the ATmega328. Arduino is an open source electronics platform consisting of hardware and software. With the embedded system that will be used the Arduino Uno will be plugged in to a computer and programmed to the set specifications that are needed for the embedded system. The reason that an Arduino Uno is being used as the microcontroller in this embedded system is because Arduino is an easy to use straightforward platform when it comes to electronics and coding.

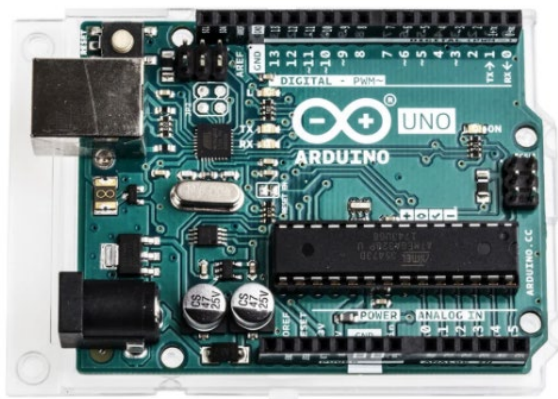


Figure 6 - Shows the Arduino Uno Rev 3

16×2 LCD

An LCD or Liquid Crystal Display is a type of flat panel display which uses liquid crystals as its main form of operation. LCD's use less power than other components as they work on the principle of blocking light rather than emitting it. In the Embedded system the LCD will be used to provide a visual display reading when an object has been detected by the Ultrasonic Distance Sensor, where only until the object is out of range will the LCD display change. The LCD specifications can be seen in figure 7.

Attribute	Value
Display Type	Alphanumeric
Character Matrix	2 Rows by 16 Characters
Characters per Row	16
Character Rows	2
Character Width	3mm
Character Height	5.6mm
Viewing Area Width	65.6mm
Viewing Area Dimensions	66 x 16mm
Viewing Area Height	16mm
External Depth	11mm
Background Colour	Green
Illumination Colour	Yellow
Backlight Type	LED
External Width	84mm
External Height	44mm
External Dimensions	84 x 11 x 44mm
Controller Type	ST7066U
Display Matrix Technology	STN
Display Mode	Transflective
Maximum Operating Temperature	+60°C
Minimum Operating Temperature	-10°C
Operating Temperature Range	-10 → +60°C

Figure 7 - Shows the LCD Specifications

Breadboard Jumper Wire

Breadboard jumper wire or jumpers are used to link different parts and components together on the breadboard being used without the need to solder.



Figure 8 - Shows the Jumper Wires

RGB LED

An RGB LED combines the three primary colours to create different colours in the light spectrum. In the embedded system the RGB LED will be used to indicate whether an object has been detected within 2m of the ultrasonic sensor. If no object has been detected then the LED will indicate a green light, whilst if an object has been detected within the 2m the LED will indicate red. Figure 9 shows the specifications for the RGB LED, as you can see the forward voltage is up to 4.1V which means that it is suitable for the Breadboard and circuit being used.

Attribute	Value
LED Colours	RGB
Package Type	5mm (T-1 3/4)
Mounting Type	Through Hole
Number of LEDs	3
Forward Voltage	2.5 V, 4 V, 4.1 V
Luminous Intensity	300 mcd, 700 mcd, 1300 mcd
Number of Pins	4
Viewing Angle	60 °
Lens Shape	Round
Dominant Wavelength	465 / 525 / 630 nm
Dimensions	5 (Dia.) x 8.6mm
Lens Colour	White

Figure 9 - Shows the RGB LED Specifications

250Ω Resistor

A resistor is an electrical component that is primarily used to limit the flow of electrical current within the circuit. A 250Ω resistor will be used within the circuit for the RGB LED, Buzzer and the LCD, the reason being is to protect these components from over current which could potentially damage them and cause damage to other components with in the prototype circuit constructed. Since the LCD uses a 4-20mA output a 250Ω resistor is needed. The resistor specifications are shown below in figure 10.

Specifications:

Type	: MCMF
Rated Power	: 0.25W at 70°C
Max. Working Voltage	: 250V
Max. Overload Voltage	: 500V
Dielectric Withstanding Voltage	: 500V
Rated Ambient Temp	: 70°C
Operating Temp. Range	: -55°C to +155°C
Resistance Tolerance	
MCMF0W4B Series	: ±0.1%
MCMF0W4F Series	: ±1%
Resistance Range	: 10Ω to 1MΩ

Figure 10 - Shows the 250Ω Resistor specifications

Task 2 – Software Design [15 marks]

Software Design

The Arduino Uno is programmed using Arduino IDE. The Arduino is the microcontroller which means that it is programmed to perform a specific task which in the case of this assignment is to detect an object within 2m and set off other components such as the Red LED, the LCD and a buzzer. The flowchart as seen in figure 11 shows a visual representation of how the code should function from start to finish. Following the flowchart, the first sequence is for the Arduino Uno to be powered up so that it has a constant power supply, this is done by plugging it into a computer or a USB source. Once powered the Arduino Uno will begin its programmed sequences that will enable the other components to function correctly. The LCD will state that the 2m area is clear, the buzzer will be silent and the RGB LED will show green at first once powered, the Arduino however will immediately send an output signal to the ultrasonic distance sensor to send an ultrasonic wave to detect if an object is present. If an object is present, then the ultrasonic sound wave will be reflected back to the sensor where it will be received, and the time taken to return will be calculated into a distance. The Arduino code will then determine what the other components output based on whether an object has been detected within 2m. IF statements have been used to determine this, so IF an object is $\leq 2\text{m}$ then the LCD will read "Object Detected Within 2m", the buzzer will go off and the RGB LED will turn red. Once the object is not within the vicinity of the sensor then the LCD will read "No Object within 2m", the buzzer will silence and the RGB LED will return to green. This will continue to happen based on whether the sensor has detected any objects.

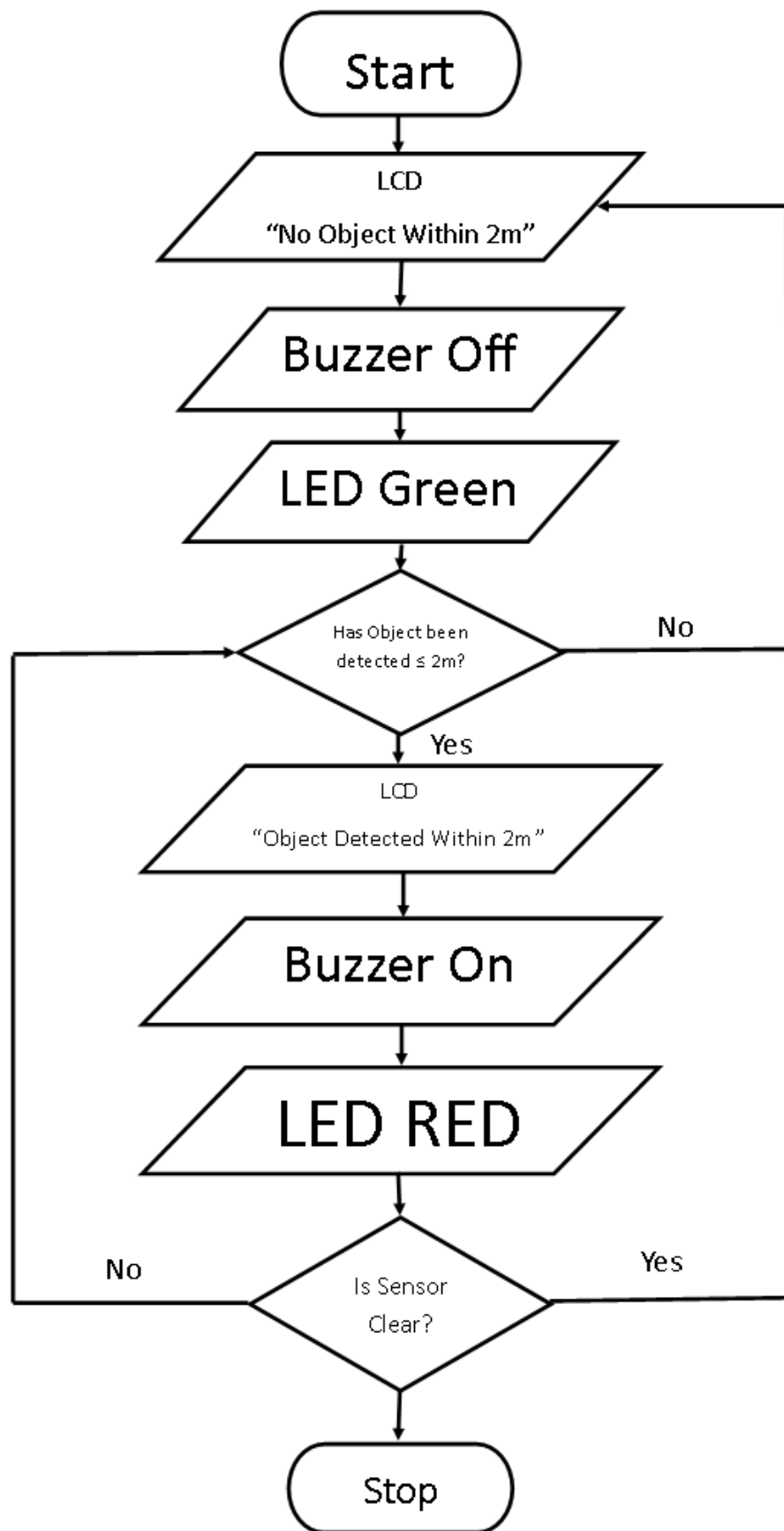


Figure 11 - Shows the Code Flowchart

completed the const int and ints were done, these are the read and write (sensor and buzzer) and the read only (RGB LED) section. The set up (void setup()) upon putting power onto the Arduino was then completed, this was carried out so that an initial message was displayed and the programme could do a run through beforehand. The final stage of the coding was to do the void loop () which ultimately enables the simulation to run continuously, this was based on the else and if statement within the code. As seen in figure 13 the in-comments did not overflow onto the next line and were clear as to what the code meant. Upon start-up of the simulation, the code appeared to have some errors. One of the errors was with the lcd.setCursor on multiple separate lines of the code. After investigating, the problem was fixed when it had been discovered that the lcd part of LiquidCrystal lcd(12,8,5,4,3,2); was in uppcase when it needed to be in lowercase. The second error on the code was with the inst NOTE_G3 400 which needed to be #define NOTE_G3 400 instead. This was because inst is a variable and variables can't have a number. The functionality of the of the alarm system can be seen in figures 13,14 and 15. When the program is turned on the lcd screen initially reads "2m Distance Detection Online" for 2 seconds, once this 2 second delay is up the sensor detects whether an object is within 2m, if so then then the display reads "Object Detected Within 2m", the buzzer alarms and the RGB LED turns red. As soon as the object is out of the 2m vicinity then the lcd will read "No Object Within 2m", the buzzer will mute and the RGB LED will go green.

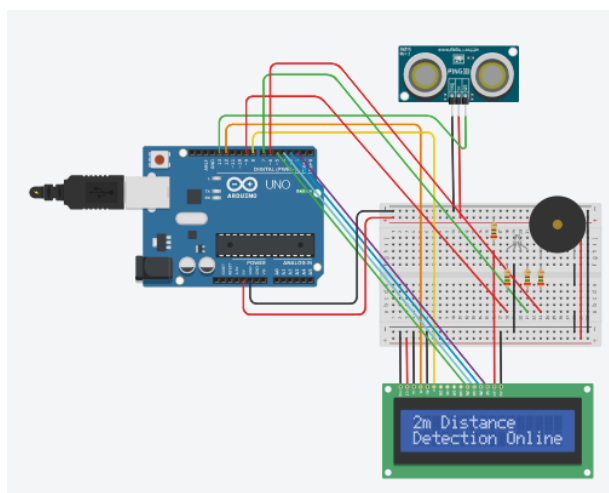


Figure 14 - Shows the Initial LCD display

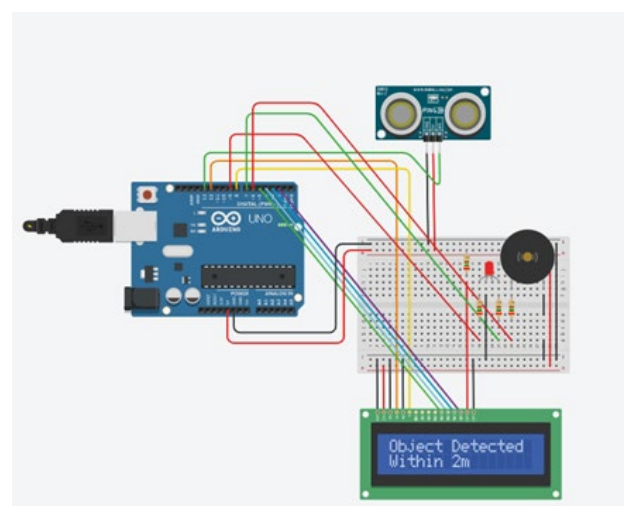


Figure 13 - Shows that the circuit has detected an object

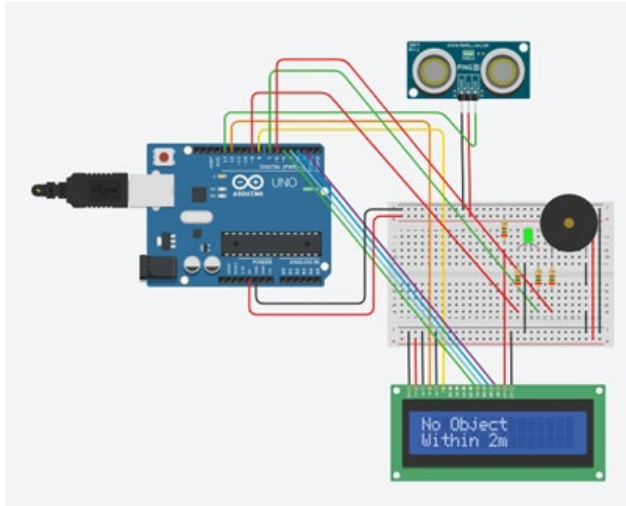


Figure 15 - Shows that no object has been detected within 2m

```

1 #include <LiquidCrystal.h>
2 #define NOTE_G3 400
3 const int COLUMNS= 16; // LCD Columns
4 const int ROWS= 2; // LCD Rows
5
6 LiquidCrystal lcd(12,8,5,4,3,2); // LCD pin numbers
7
8 const int SENSOR_PIN=13; // Ultrasonic Sensor pin number
9 const int BUZZER=6; // Buzzer pin number
10 int LEDRED=9; // RED LED pin number
11 int LEDGREEN=7; // Green LED pin number
12
13
14 void setup()
15 { // Start of group statement
16   Serial.begin(9600);
17
18   pinMode(BUZZER,OUTPUT); // Buzzer Output
19   pinMode(LEDRED,OUTPUT); // RED LED Output
20   pinMode(LEDGREEN,OUTPUT); // GREEN LED Output
21
22   lcd.begin(COLUMNS,ROWS);
23   lcd.print("2m Distance"); // LCD Display Reading
24   lcd.setCursor(0,1);
25   lcd.print("Detection Online"); // LCD Display Reading
26   delay(2000); // 2 Second Delay
27   lcd.clear();
28 } // End of group statement
29
30 void loop()
31 { // Start of function declaration
32   int distance,microseconds;
33   pinMode(SENSOR_PIN,OUTPUT);
34   digitalWrite(SENSOR_PIN,LOW);
35   delayMicroseconds(1500); // 1.5 Second Delay
36   digitalWrite(SENSOR_PIN,HIGH);
37   delayMicroseconds(1500); // 1.5 Second Delay
38   digitalWrite(SENSOR_PIN,LOW);
39   pinMode(SENSOR_PIN,INPUT);
40   microseconds=pulseIn(SENSOR_PIN,HIGH); // Sensor microseconds
41   distance=microseconds/29/2; // Sensor microseconds to Distance
42   Serial.println(distance);
43   if (distance <= 200 && distance >1) // If distance is <= 2m di
44   { // Start of group statement
45
46     digitalWrite(LEDRED, HIGH); // Red LED will be displayed
47     digitalWrite(LEDGREEN, LOW); // Green LED will not show
48     lcd.clear();
49     lcd.print("Object Detected"); // LCD Display Reading
50     lcd.setCursor(0,1);
51     lcd.print("Within 2m"); // LCD Display Reading
52     tone(BUZZER,NOTE_G3); // Buzzer Tone will be heard
53     delay(1000); // 1 Second Delay
54   } // End of group statement
55   else // If above code does not apply the code below applies
56   { // Start of group statement
57     digitalWrite(LEDRED,LOW); // Red LED will not show
58     digitalWrite(LEDGREEN,HIGH); // Green LED will be displayed
59     lcd.clear();
60     lcd.print("No Object"); // LCD Display Reading
61     lcd.setCursor(0,1);
62     lcd.print("Within 2m"); // LCD Display Reading
63     noTone(BUZZER); // Buzzer Off
64     delay(1000); // 1 Second Delay
65   } // End of group statement
66 } // End of function declaration

```

Figure 16 - Shows the code lines 1-66

Task 4 – Testing [10 marks]

Testing

As part of the testing phase of this assignment a series of screenshots will be shown below with an explanation of what is happening. Figure 17 below shows the program before the simulation had started.

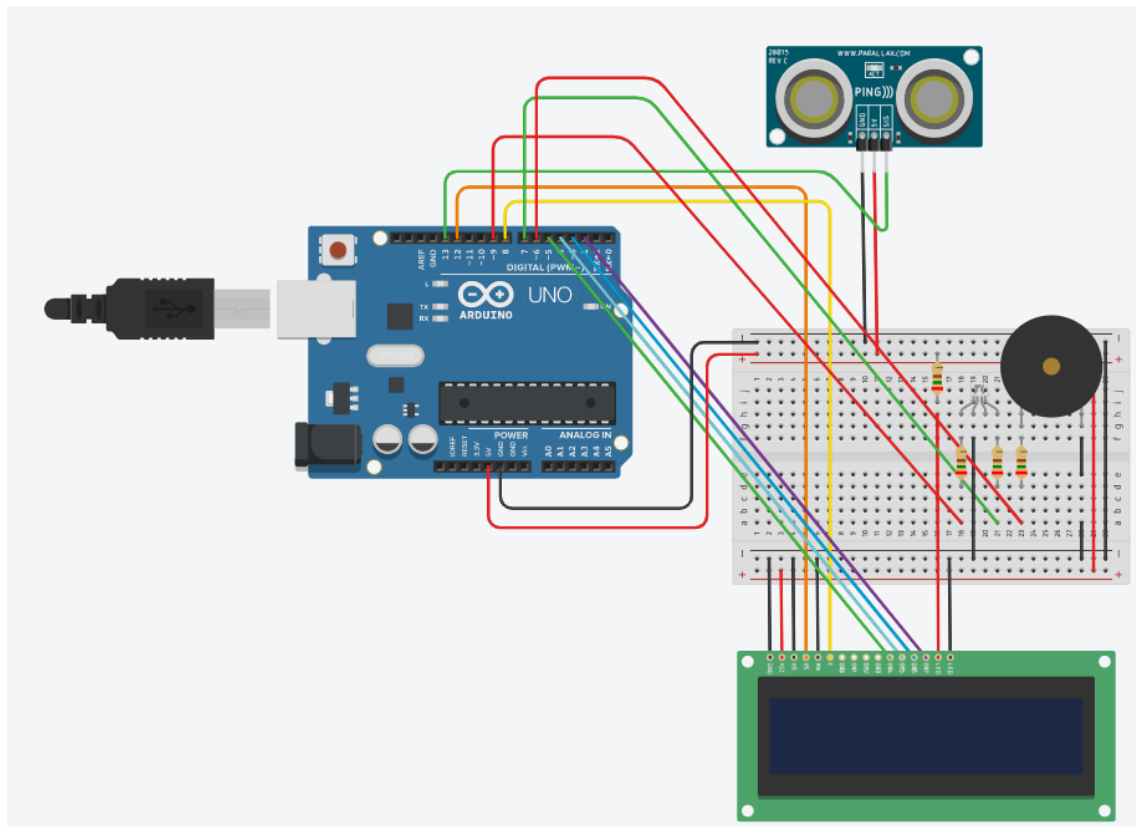


Figure 17 - Shows the Sensor Circuit

Once the simulation has started it can be seen that the RGB LED is green, the buzzer has no tone, and the LCD tells us that the “2m Distance Detection Online”, after 2 seconds the LCD will then read “No Object Within 2m” as long as no object has been detected. This can be seen in figure 18 and 19.

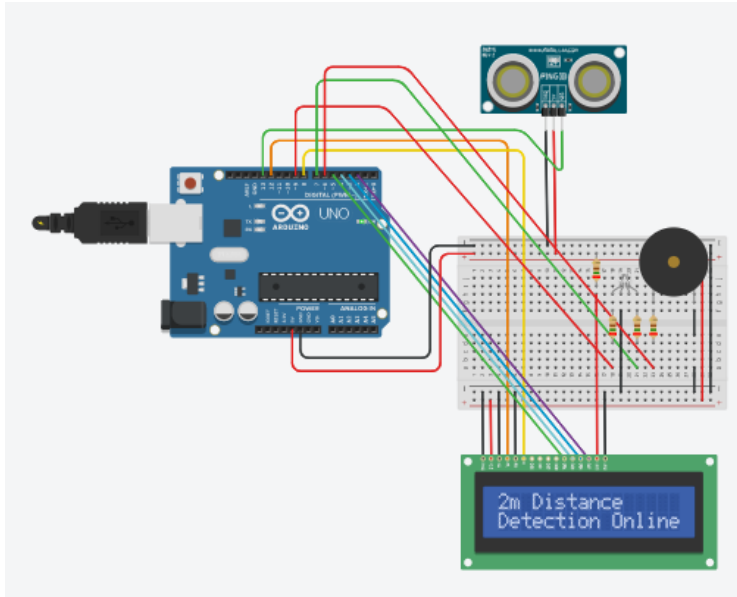


Figure 18 - Shows the Initial LCD power on message

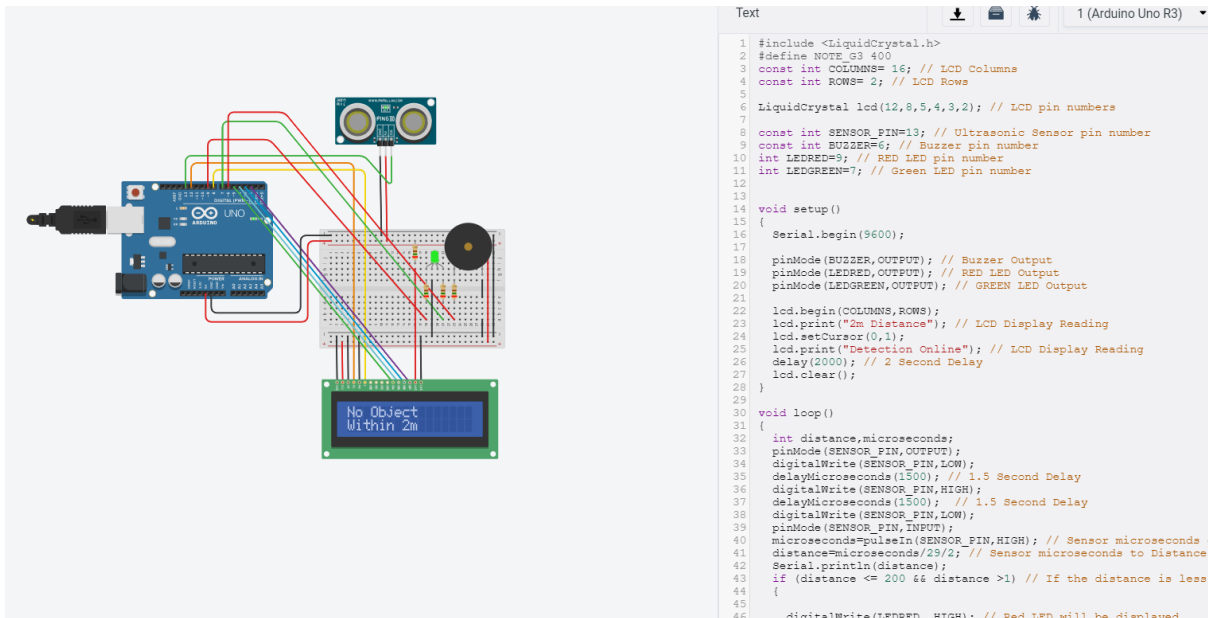


Figure 19 - Shows the Simulation once started up

When the object is moved within 2m of the sensor, the buzzer gives off a continuous tone, the RGB LED turns red and the LCD gives a display which reads “Object Detected Within 2m”. Figure 20 shows the circuit when an object has been detected.

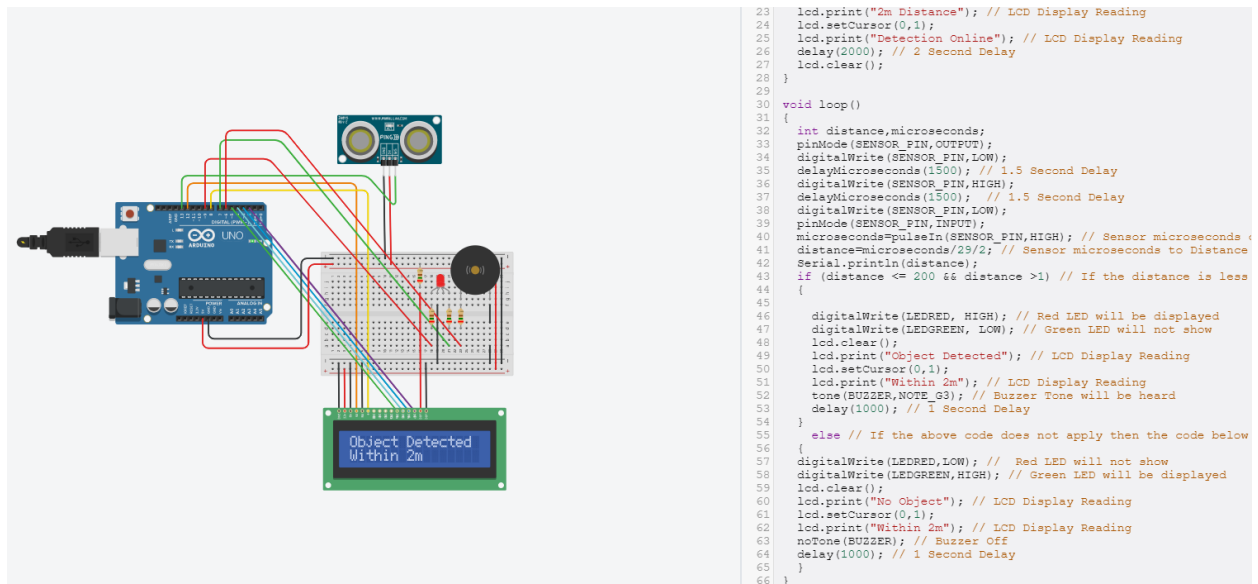


Figure 20 - Shows the Circuit when an object has been detected

The simulation will continue to run as long as there is power to the Arduino. The circuit condition is based off of the sensor detecting an object.

Task 5 – Discussion, Evaluation and Conclusions [15 marks]

Discussion, Evaluation and Conclusions

C is a high-level programming language where the interface between the user and the processor is easy to work with and human friendly. C is used for microcontrollers to operating systems as it is very flexible allowing maximum control with minimal commands. The results gained from my system is clear to see that the circuit functions well completing what is needed for this assignment, whilst having a simple and tidy design overall. A modification that could make this system more useful could be a keypad that enabled you to mute the buzzer and turn the RGB LED Amber to show that you are aware that an object has been detected and you have silenced it but it will still show on the LED that the object is still within 2m. By adding the keypad, this could then potentially make this a security system for a house or a shed. In conclusion I believe that I have met the requirements of the system as the simulation works correctly as it should, altering the components as and when needed. The code is simple as having too many excessive lines and functions could end up overcomplicating the code causing it to work incorrectly or to not work at all.

Hardware Design Datasheets

RS PRO Piezo Buzzer - <https://docs.rs-online.com/10b8/0900766b815810c6.pdf>

Parallax Inc 28015 Ultrasonic Distance Sensor Module - <https://docs.rs-online.com/ca69/0900766b81382974.pdf>

80×60×10mm Breadboard - <https://docs.rs-online.com/5834/0900766b81582f7f.pdf>

Arduino Uno Rev 3 - [0900766b80e8ba21.pdf](https://docs.rs-online.com/0900766b80e8ba21.pdf) ([rs-online.com](https://www.rs-online.com))

Displaytech 162B-BC-BC Alphanumeric LCD - [series](https://docs.rs-online.com/series/0900766b812bf0ed.pdf) ([rs-online.com](https://www.rs-online.com))

Breadboard Jumper Wires - <https://docs.rs-online.com/285e/0900766b812bf0ed.pdf>

RGB LED - <https://docs.rs-online.com/a4f6/0900766b8139d70e.pdf>

250Ω Resistor - <http://www.farnell.com/datasheets/2872372.pdf>

Appendix

Figure 1 - Shows the specifications of the buzzer.....	3
Figure 2 - Shows the Piezo Buzzer	3
Figure 3 - Shows the Key Features of the Ultrasonic Sensor Module.....	4
Figure 4 - Shows the Ultrasonic Distance Sensor.....	4
Figure 5 - Shows the 80×60×10mm Breadboard	5
Figure 6 - Shows the Arduino Uno Rev 3	5
Figure 7 - Shows the LCD Specifications	6
Figure 8 - Shows the Jumper Wires	7
Figure 9 - Shows the RGB LED Specifications.....	7
Figure 10 - Shows the 250Ω Resistor specifications	8
Figure 11 - Shows the Code Flowchart	10
Figure 12 - Shows the circuit design	11
Figure 13 - Shows that the circuit has detected an object	12
Figure 14 - Shows the Initial LCD display	12
Figure 15 - Shows that no object has been detected within 2m	13
Figure 16 - Shows the code lines 1-66	13
Figure 17 - Shows the Sensor Circuit	14
Figure 18 - Shows the Initial LCD power on message	15
Figure 19 - Shows the Simulation once started up	15
Figure 20 - Shows the Circuit when an object has been detected	16

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 in system, explaining why you chose these items. Consideration of alternatives available would have given you higher marks in this section. Software design 9/15 You start with a clear coherent description of how the software will control the hardware. Flowchart considers main loop of the program but fails to acknowledge the setup code required. Appropriate shapes are used for flowchart and it is mostly logically sound. However, it is customary not to implement the stop feature as the loop function in Arduino continues until the unit is powered down. It does not make sense to have a yes, no, AND another arrow coming from a decision box!	30	20
3. Cognitive and Intellectual Skills		
Discussion, evaluation and conclusion A sound evaluation of the underlying concepts and principles, you have considered the effectiveness of your solution in relation to the original brief, given some good suggestion on how to further improve it. You have given a brief overview of the usefulness of C programming language for an embedded system.	15	9
4. Practical Application Skills		
Implementation 25/40 Code is well written with self-annotating variables, use of constants where appropriate, and very well commented. Where you use constants, use either const or #define, rather than both to show consistency. You would gained higher marks for using some suitable functions, such as a function to calculate the distance from the sensor to the object detected. testing 6/10	50	31

An 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. It would have helped to show where the object was placed in the simulation. An inclusion of a testing table gives a more structured way to performs tests.		
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 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)		64%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		