

Module Assessment 2021/22 _ Communication 2.0
Emma Sidey 3722216 BA (Hons) Architecture PT4 2021/22

Manipulating Architectural Design using Artificial Intelligence

Introduction to Robotics Report

Contents

Introduction	03
Section 01	04-13
An Investigation into the Field of Robotics	
3.1 Chapter 01: Understanding Tools and Components	
3.2 Chapter 02: Online with the Arduino	
3.3 Chapter 03: Measuring the Physical Environment using the Arduino	
3.4 Chapter 04: Capturing data off the Arduino on a Liquid Crystal Display (LCD)	
3.5 Chapter 05: Building a Robot with Sensors	
Section 02	14-15
The Future of Civic Cyber Space	
4.1 Develop and Validate a Prototype	
4.2 Research: Human Rights Commission and Government Digital Service	
Section 03	16-22
Building the Prototype	
5.1 Stage 01: Raspberry Pi (RPi) 4 Computer (Model B)	
5.2 Stage 02: Raspberry Pi OS	
5.3 Stage 03: Circuit Experiments using Python3	
Conclusion	23
Bibliography	24

An evolving interaction between advanced automation and architectural design profoundly impacts the technological possibilities demonstrated within this report. Undertaking preparation of a Raspberry Pi (RPI) to incorporate artificial intelligence (AI), computer vision (CV) and image processing. Further developed enabling perception of fundamental aspects of RPI, Raspberry Pi OS and Python producing a replication of biological vision as a mechanical system with the ability to read and rewrite data converting [INPUT] to the programmed [OUTPUT] through decision making.

An Investigation into the Field of Robotics

Chapter 01: Understanding Tools and Components

Conducting an investigation into the field of robotics, and evaluating each sector, Chapter 01 demonstrates an understanding of the tools and components involved in fundamental development.

An elemental, but imperative skill, to begin with, is stripping wires. A wire, as a component, is a critical contributor in any form of robotics with its primary purpose to transport electricity. In order to ensure a safe, durable circuit, removing the outer layer of plastic insulation without harming the wire underneath is paramount. If this is not performed accurately it may result in electrical shorts. To perform this exercise you need side cutters in one hand, with the chamfered edge facing the wire and the appropriate wire in the other hand gripped with your finger exposing one end. Ensuring you apply light pressure to the layer of plastic insulation to slice the shell but not cut entirely. This enables you to slide the end of the plastic insulation off the wire exposing the end, ready for connections.

The most common method of connection is soldering wires either together or to a board, this ensures the electrical connection is both effective and reliable. As a conducting material, solder ensures the bond between the two wires is permanent and enables the current to flow continuously. Another fundamental tool studied is the glue gun, this ensures all connections and components are sealed, and the glue is moldable and therefore optimal for organising the board as demonstrated within the final circuit of section 01. This provides a clear viewpoint for all wires, connections and components allowing the device to be read with ease, and ensuring any mistakes can be noticed and resolved.

Following analysis of the tools required, an induction to the electronic components took place offering an in-depth understanding of the principle and purpose of each element. While building our test circuit we experimented with a breadboard, LEDs, an Arduino board, transistors, resistors and environmental sensors. A breadboard was chosen as it enables a solder-free connection allowing a 'plug-and-play' platform ideal for a test model as it provides the ability to adapt placement and experiments with an array of board layouts. Experiment number one was to understand the polarity of an LED, an LED is a diode that only accepts a current flow in one direction determined by negative and positive. Like the majority of diodes, an LED is polarity-sensitive which means the flow of current must also be controlled by resistors to limit the value and ensure the voltage isn't too high otherwise the LED will not light. A resistor resists the electrical current to a component measured in Ohms, its primary purpose is to adjust signal levels, divide voltages, bias active elements and terminate transmission lines.

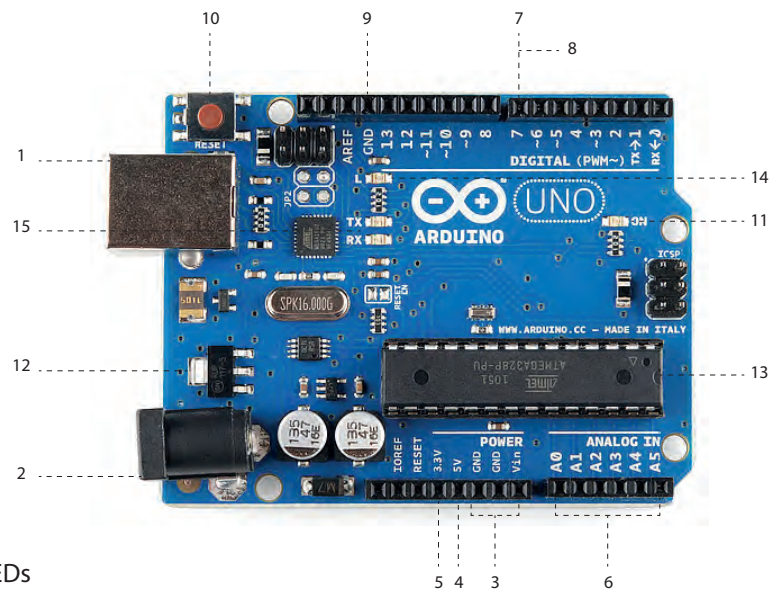
Electricity is the movement and accumulation of electrically-charged particles, all components within an electrical circuit consist of atoms. + protons, - electrons, neutrons. The amount of charge in an object is measured in coulombs (C), + repels +, - repels -, + attracts - and - attracts +. When referring to the material as a conductor it represents how freely electrons move within making it highly conductive, an insulating material would refer to material that is low conductivity as electrons cannot pass through. The movement of electrons refers to the electrical current that flows, this is measured in Amperes (A), 1 Amper = 1 Coulomb per second.

Expanding the experiment, Arduino as an open-source microcontroller development board was introduced. The decision was made to select Arduino over Raspberry Pi (SBE) because the Arduino board contains the CPU, RAM and ROM required for additional programming. As a microcontroller, the Arduino board senses the environment by receiving inputs from sensors and converts the data to output to control the device.

Arduino Board Diagram

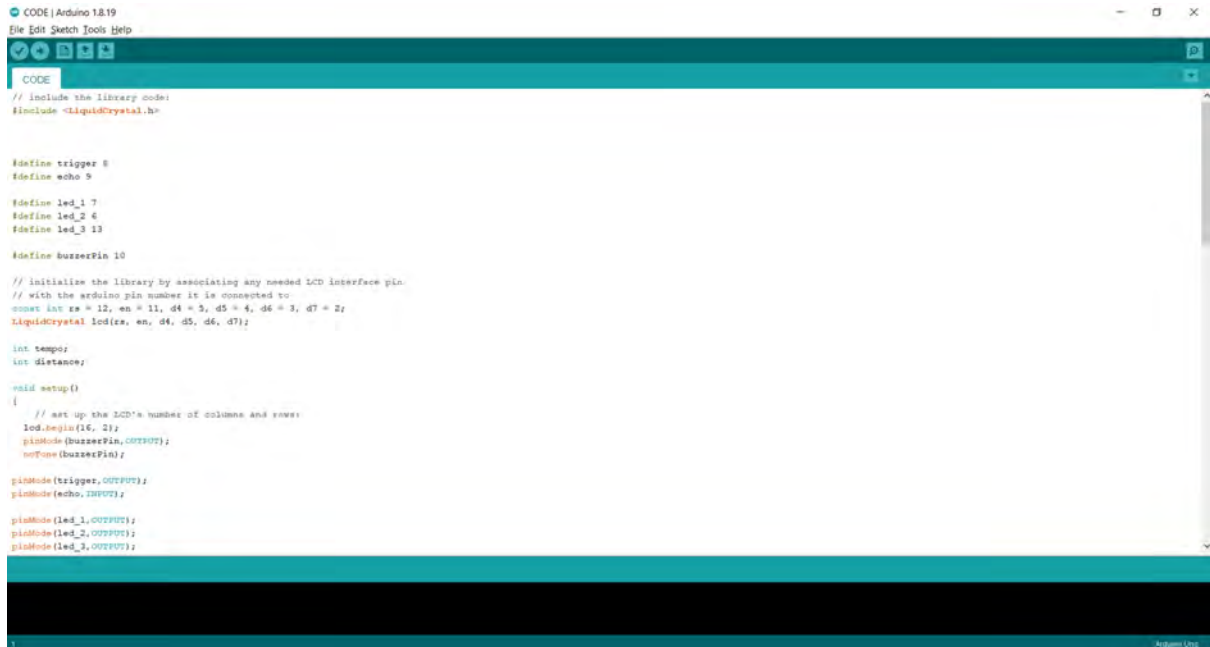
Diagram Key:

1. USB Connection
2. 6V~12V Barrel Jack
3. GND
4. 5V Pin Supplies
5. 3.3V Pin Supplies
6. Analog Pin
7. Digital Pin
8. (Pwm~)
9. GND
10. Reset Button
11. Power LED Indicator
12. Voltage Regulator
13. Main IC
14. Pin D13 Indicator LED
15. Serial Communication Indicator LEDs



Specifically, designed with a user-friendly interface enabling a clear introduction to integrate electronics and programming. Analysing key features, we downloaded and installed Arduino IDE (Integrated development environment), as demonstrated within figures 01-04 (pg.06-07) further advancement included adapting the Arduino pins and pinMode, analysing and assessing the digital input signals from the sensors previously installed on the test circuit and ensuring the code concludes to 'blink' the LED on/off.

The pins highlighted within the diagram enable the connection between the Arduino and Breadboard to create the circuit. GND refers to (-) Ground, this pin is the reference point in an electrical circuit where the VOLT is measured. 5V and 3.3V (+) are pin supplies, the first 5 VOLTS and the other 3.3 VOLTS of power. Analog (A0-A5) are pins that are programmed with the ability to read signals from an analog sensor and convert the data into a digital value that a human can understand. Digital, (0-13 UNO) are pins that are used for both digital input and output.



```
CODE | Arduino 1.8.19
File Edit Sketch Tools Help

CODE

// include the library code:
#include <LiquidCrystal.h>

#define trigger 8
#define echo 5

#define led_1 7
#define led_2 4
#define led_3 13

#define BuzzerPin 10

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int tempo;
int distance;

void setup()
{
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  pinMode(BuzzerPin, OUTPUT);
  digitalWrite(BuzzerPin, LOW);

  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);

  pinMode(led_1, OUTPUT);
  pinMode(led_2, OUTPUT);
  pinMode(led_3, OUTPUT);
}
```

Figure 01 - Arduino IDE Code



```
CODE | Arduino 1.8.19
File Edit Sketch Tools Help

CODE

pinMode(led_1, OUTPUT);
pinMode(led_2, OUTPUT);
pinMode(led_3, OUTPUT);

lcd.print("distance: ");
lcd.setCursor(14,0);
lcd.print("cm");
digitalWrite(trigger, LOW);
tempo=337*58;
}

void loop()
{
  digitalWrite(trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger, LOW);

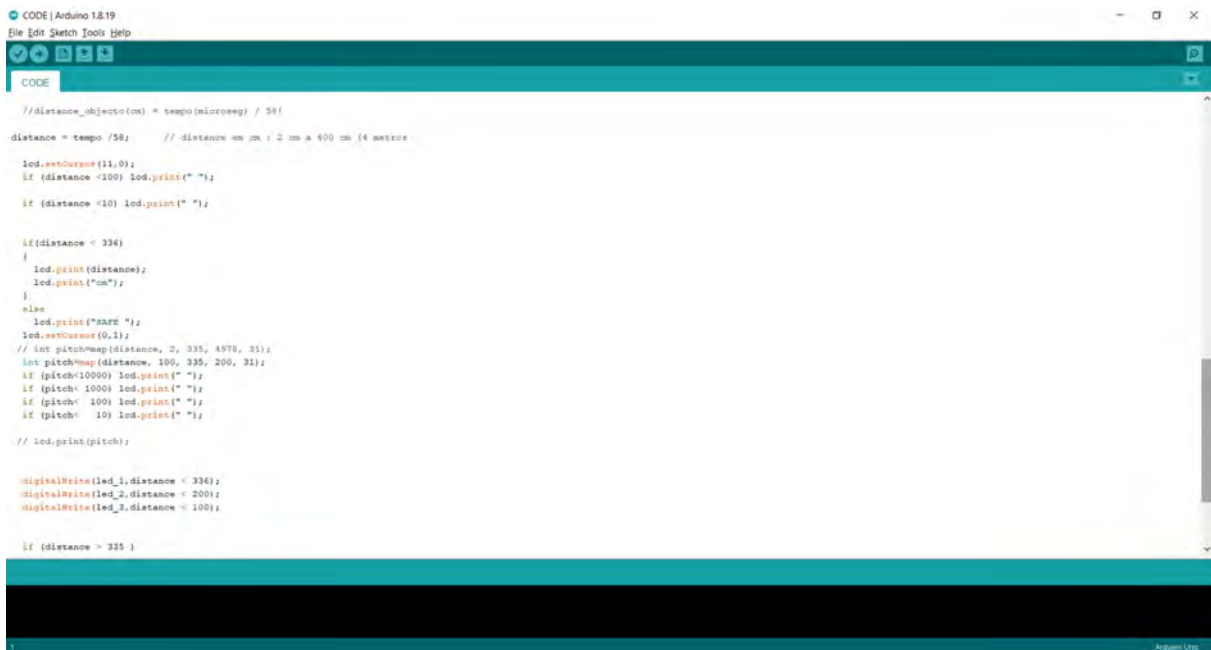
  tempopulsado(echo, HIGH); //microsegundos

  //distance = velocidade x tempo
  //velocidade_son = 343 m/s
  //tempo(seg) = tempo(microseg)/1000000
  //distance(m) = 343 x tempo(seg) -> (ida e volta)
  //distance(cm) = distance(m) x 100
  //distance_objeto = distance(cm) / 2 -> (apenas ida)

  //(343 x 100 x (tempo/1000000)) / 2 = tempo x 0,01715 +/- tempo / 58
  //distance_objeto(cm) = tempo(microseg) / 58;

  distance = tempo / 58; // distance em cm : 2 cm x 400 cm (4 metros)
```

Figure 02 - Arduino IDE Code



```
CODE

//distance_object(cm) = tempo(microseg) / 58f

distance = tempo / 58;    // distance on cm : 2 cm a 400 cm (4 metres)

led.setBuzzer(11,0);
if (distance < 100) led.print(" ");

if (distance < 10) led.print(" ");

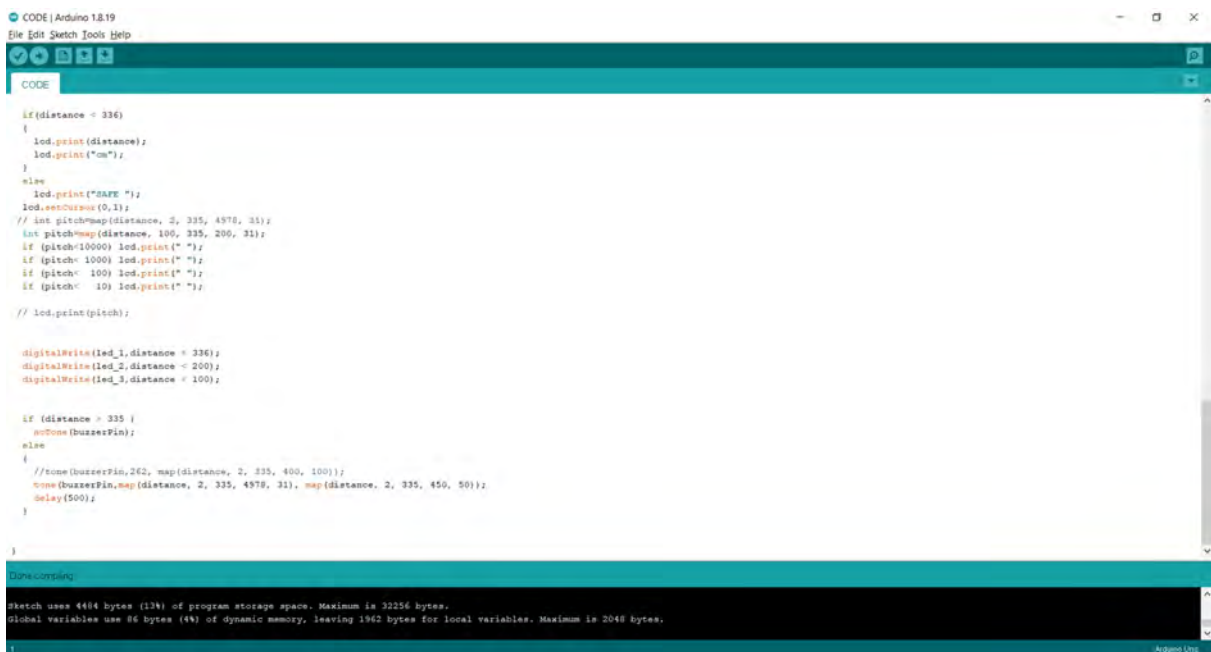
if (distance < 336)
{
  led.print(distance);
  led.print("cm");
}
else
{
  led.print("NARE ");
  led.setBuzzer(0,1);
  // int pitch=map(distance, 2, 335, 4978, 31);
  int pitch=map(distance, 100, 335, 200, 31);
  if (pitch<10000) led.print(" ");
  if (pitch< 1000) led.print(" ");
  if (pitch< 100) led.print(" ");
  if (pitch< 10) led.print(" ");

  // led.print(pitch);

  digitalWrite(led_1,distance < 336);
  digitalWrite(led_2,distance < 200);
  digitalWrite(led_3,distance < 100);

  if (distance > 335 )
```

Figure 03 - Arduino IDE Code



```
CODE

if (distance < 336)
{
  led.print(distance);
  led.print("cm");
}
else
{
  led.print("NARE ");
  led.setBuzzer(0,1);
  // int pitch=map(distance, 2, 335, 4978, 31);
  int pitch=map(distance, 100, 335, 200, 31);
  if (pitch<10000) led.print(" ");
  if (pitch< 1000) led.print(" ");
  if (pitch< 100) led.print(" ");
  if (pitch< 10) led.print(" ");

  // led.print(pitch);

  digitalWrite(led_1,distance < 336);
  digitalWrite(led_2,distance < 200);
  digitalWrite(led_3,distance < 100);

  if (distance > 335 )
  {
    noTone(buzzerPin);
  }
  else
  {
    //tone(buzzerPin,262, map(distance, 2, 335, 400, 100));
    tone(buzzerPin,map(distance, 2, 335, 4978, 31), map(distance, 2, 335, 450, 50));
    delay(500);
  }
}

}

Done compiling.

Sketch uses 4684 bytes (13%) of program storage space. Maximum is 32256 bytes.
Global variables use 86 bytes (4%) of dynamic memory, leaving 1962 bytes for local variables. Maximum is 2048 bytes.
```

Figure 04 - Arduino IDE Code

Chapter 02: Online with the Arduino

Following a thorough investigation and analysis of the tools and components required, Chapter 02 demonstrates an understanding surrounding the fundamental skills and knowledge required to operate an Arduino UNO. Including implementing software programming through the functions `setup()` and `loop()`, as well as, understanding the key principles to writing a structured programme, in three parts, Structure, Values (Variables and Constraints) and Functions.

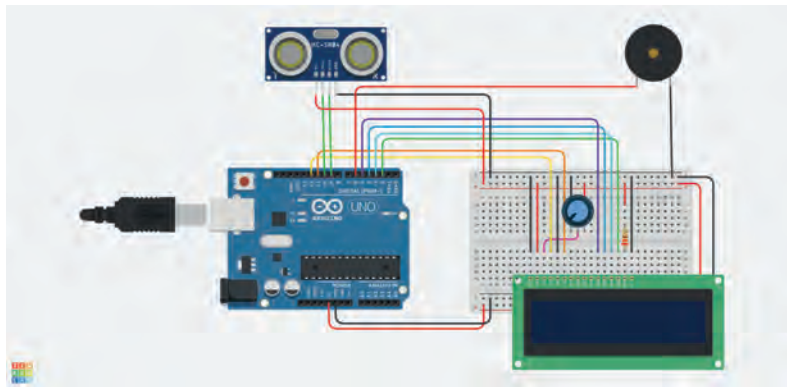


Figure 05 - Circuit with A B USB Cable

An Arduino board requires power to operate, after installing the Arduino software (IDE) in chapter 01, connecting the board requires an A B USB cable to provide power as demonstrated within figure 05. This produces force to move the electrons, measured in VOLT. Once the board has been connected successfully the Arduino will display a light to represent live power. Once the power has been established the `setup()` function can begin, power is required as it enables you to trail run code on variables and pin modes.

Example:

```
"int buttonPin = 3;

void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  // ...
}"
```

Arduino (2019).

When referring to Structure, Values (variables and constants), and Functions, the structure is the key element (C++) code, as demonstrated with the example sketch code, structural elements are `loop()`, `setup()`, as well as elements of control and command such as `'break'`, `'>'` (greater than), `'='` (equal). A Function within Arduino programming is the element of code that controls the board and performs computations. Variables are the elements of data types and constants, the most common example of a variable would be INPUT and OUTPUT.

Chapter 03: Measuring the Physical Environment using the Arduino

The evolving interaction between advanced automation and architectural design enables environmental responsive projects through encapsulating data. Chapter 03 introduces an ultrasonic sensor interfaced into the Arduino Uno connected through the 4 terminals, +5V, Trigger, Echo and GND. The function incorporates, detects and measures the distance to an object, typically used for navigation, object avoidance and home security.

The 4 terminals involved are Vcc (+5V) which provides power enabling the production of the ultrasonic pulses. GND which connects the sensor to the ground, Trigger which automates the [OUTPUT] sending the pulse and Echo, which reads the [INPUT] and processes the data connected to the interface.

The ultrasonic sensor used in this report is the "HC-SR04, which can measure distances from 2 cm up to 400 cm with an accuracy of ± 3 mm." Arduino | 60, C.B. | (2015). In order for the sensor to measure the distance to an object, it emits a pulse of ultrasonic sound on [OUTPUT] in a line of direct travel until an object has been detected. Once detected, the pulse is emulated back to the sensor, captured as [INPUT] and processed through programmed formulas.

$\text{Speed} = \text{Distance} / \text{Time}$

To calculate the distance in order to display the data on the interface, the sensor programs:

$\text{Distance} = \text{Speed} \times \text{Time}$

When referring to time, this is measured through the time it takes for the pulse to be released on [OUTPUT] and returned on [INPUT], this is then divided into 2 to measure the time to the object, before returning. The sensor consists of 2 transducers, which are the elements that convert mechanical forces into electrical signals, the ultrasonic pulse on [OUTPUT] is sent from the transmitting transducer and when returned on [INPUT] it is through the receiving transducer.

"To initiate a distance measurement, we need to send a 5V high signal to the Trig pin for at least 10 μ s. When the module receives this signal, it will emit 8 pulses of ultrasonic sound at a frequency of 40 KHz from the transmitting transducer." Arduino | 60, C.B. | (2015). The receiving transducer observes until the 8 pulses are returned after detecting an object within range. The data captured through the process is then displayed on the connected interface in inches and cm, via the Arduino computer serial port.

Chapter 04: Capturing data off the Arduino on a Liquid Crystal Display (LCD)

Developing the knowledge acquired within Chapter 03 surrounding the ultrasonic sensor and data processed on the interface through Arduino, chapter 04 extends the circuit by incorporating further coding to capture the data on a Liquid Crystal Display (LCD).

```
#include <LiquidCrystal.h>
#define trigPin 10
#define echoPin 13

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  float duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) * 0.0344;

  if (distance >= 400 || distance <= 2){
    lcd.print("Out of range");
    delay(500);
  }
  else {
    lcd.print(distance);
    lcd.print(" cm");
    delay(500);
  }
  delay(500);
  lcd.clear();
}
```

The Liquid Crystal Display (LCD) can come in many forms, however, this circuit connects a 16-pin interface. The code incorporated enables the microcontroller to manipulate multiple pins displayed on the interface at the same time. The pins are derived into 4 elements each with a different purpose, a register select (RS) pin holds the data memory for the liquid crystal display prior to display, a read/write (R/W) pin that determines the mode chosen to display data, and there is an enable pin that allows the data to register on the interface. The last element is the 8 data pins (D0-D7) programmed as high or low that will appear as either reading or writing depending on the requirement.

"There's also a display contrast pin (Vo), power supply pins (+5V and GND) and LED Backlight (Bkl+ and Bkl-) pins that you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively." Liquid Crystal Displays (LCD) with Arduino (2018).

As demonstrated within figure 06 the following pins are connected enabling the Liquid Crystal Display to function in the circuit:

- LCD Register Select (RS) pin to Digital (PWM~) pin 12
- LCD Enable (E) pin to Digital (PWM~) pin 11
- LCD D4 pin to Digital (PWM~) pin 5
- LCD D5 pin to Digital (PWM~) pin 4
- LCD D6 pin to Digital (PWM~) pin 3
- LCD D7 pin to Digital (PWM~) pin 2
- LCD Read/Write (R/W) pin to GND
- LCD VSS pin to GND
- LCD VCC pin to 5V

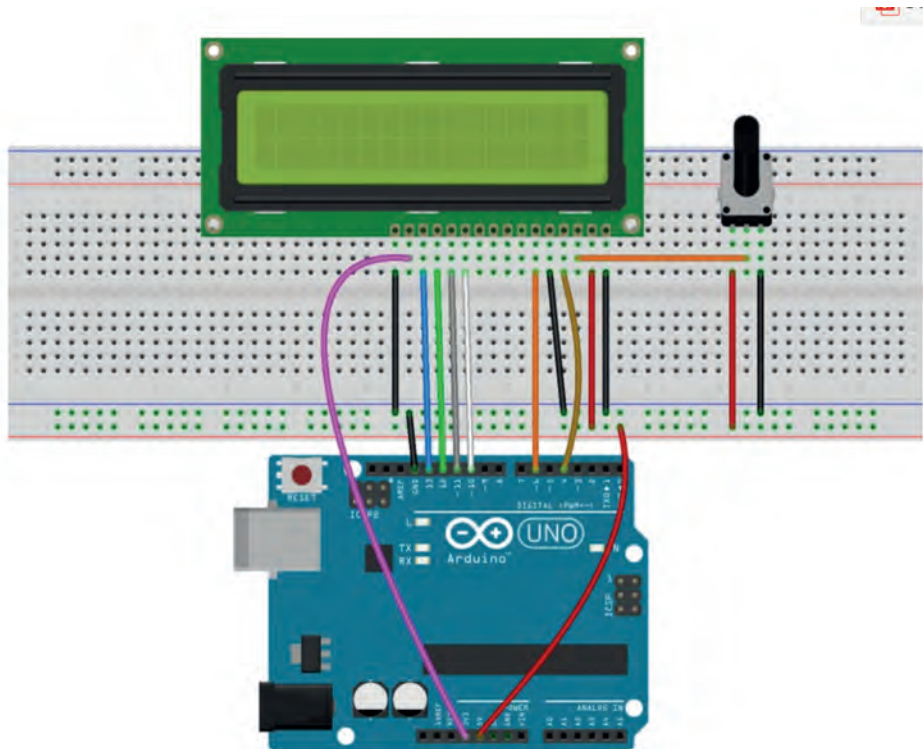
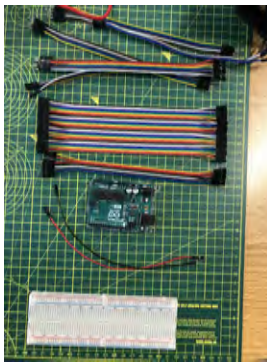


Figure 06 - Liquid Crystal Displays (LCD)

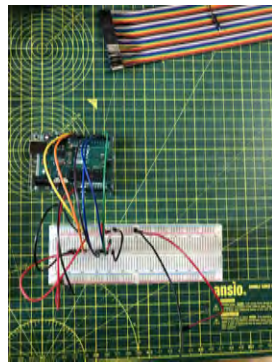
Chapter 05: Building a Robot with Sensors

To finalise the short course experiment as an introduction to robotics, chapter 05 concludes the evaluation, completing a functional circuit and performing as an ultrasonic parking sensor demonstrating how sensors and motion within advanced automation are incorporated into built space. Throughout everyday life, modern technology is continuously evolving and within architecture specifically, robots appear in many forms, and when referring to a 'robot' it could be a robotic arm, machinery, printers, or any sub-object related to programmable automated mechanisms. Many have been introduced into the construction and materials sector, providing alternatives to movement, coordination, design and programming. Many researchers are experimenting with robotic fabrication in architecture analysing the possibilities within digitalisation, virtualisation, and computerisation.

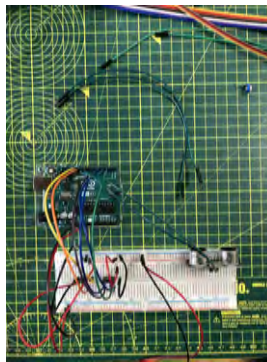
01



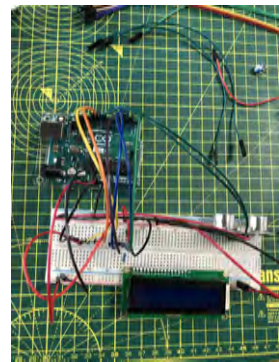
02



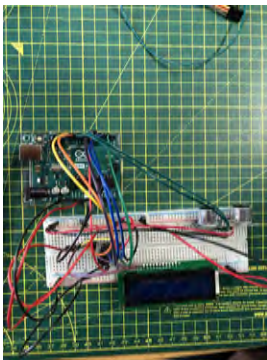
03



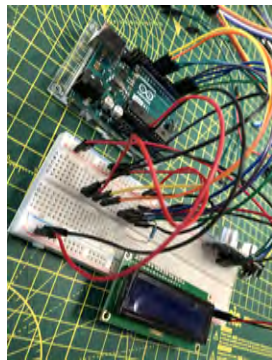
04



05



06

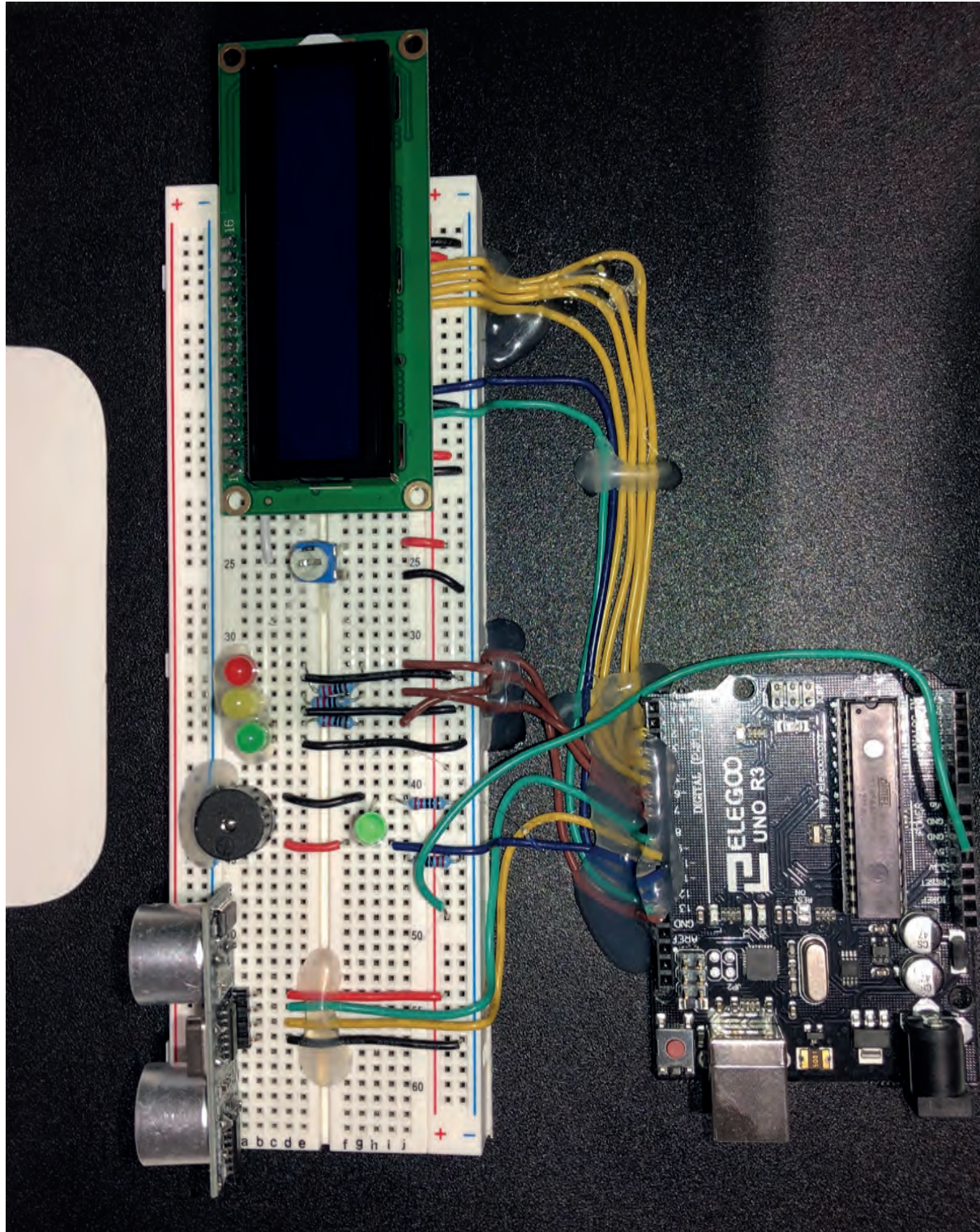


07



Following the conduction of a test in step 07 to ensure the circuit functioned accordingly, I then rebuilt the circuit to provide a clear demonstration for all wires, connections and components allowing the device to be read with ease.

Final Circuit Concluding Section 01

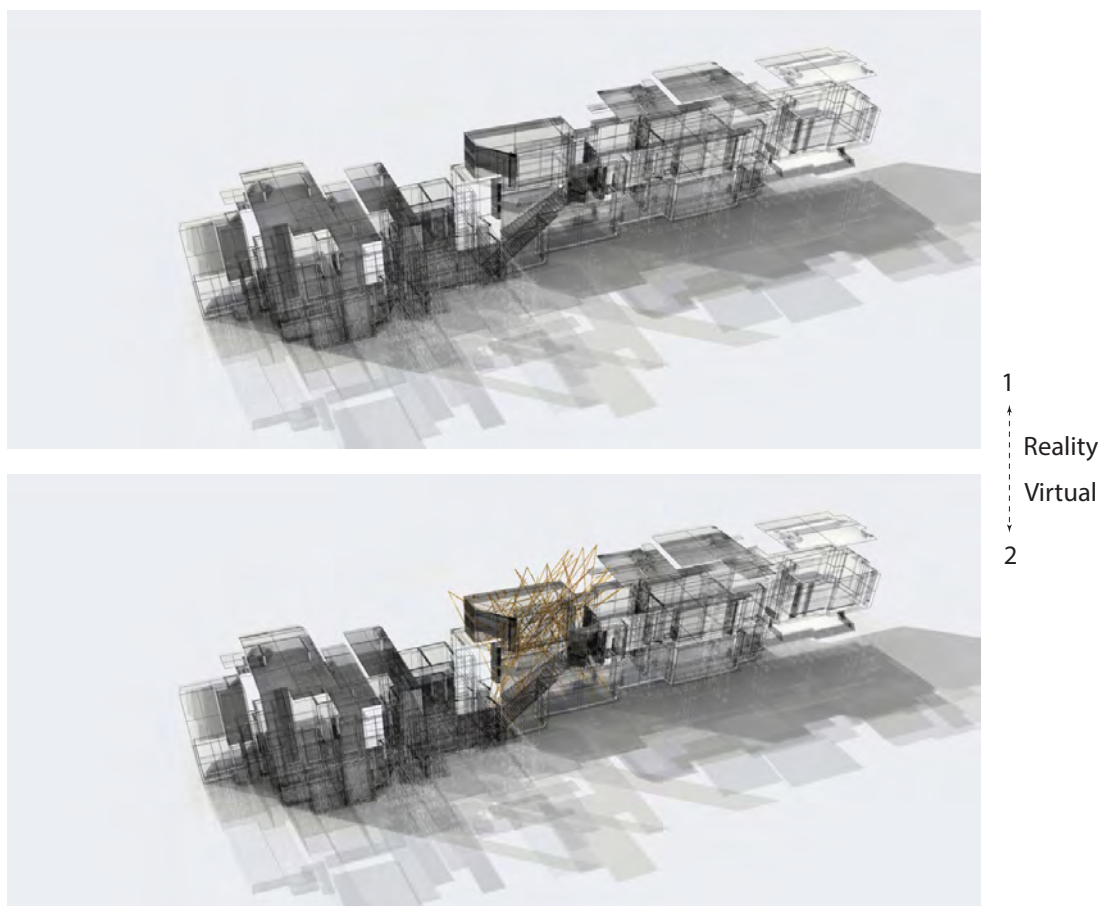


The Future of Civic Cyber Space

"The Future of Civic CyberSpace is a structure of symbolic activism, responding to the political and social issues of racial abuse within modern society. Developing a Civic Community Centre of visual representation and interpretation, enforcing reflection through experience inflicting awareness.

The threshold between two definitive spaces, virtual and physical, significantly decreased through the transformation of a post-pandemic world of dual existence. Investigating the concept of community, and the experience of spatial and temporal dimensions of human interaction age, race, religion, and crime are interconnected. Particularly demonstrated within younger generations of technological understanding, a targeted group 10-18 years old fall victim to abhorrent acts of online racial abuse and cyber attacks." Sidey, E. (2022).

Section 02 will develop and validate a prototype to demonstrate how advanced automation can be incorporated by manipulating architectural design using artificial intelligence alongside computer vision and processing. The structure within The Future of Civic CyberSpace will identify all forms of offensive content displayed as either text or image, once identified the data will be programmed on input to transform from a negative to a positive experience. Current research projects discuss the ability of Artificial Intelligence (AI) to identify toxic online content, the prevalence of online racial abuse has rapidly increased due to the global pandemic enforcing more social users and online interaction. Although this form of racial abuse stems from online use, it results in physical and mental abuse offline.



Develop and Validate a Prototype

To begin calculating the required elements to form the structure incorporated into The Future of Civic Cyber Space, thorough research took place to understand the current limitations as mentioned above and how the structure can overcome these restrictions.

Artificial Intelligence (AI) is the field currently being explored to address the global issue, as an extremely advanced section of computer science, AI is used to provide mechanical systems with human intelligence.

Computer vision as a subfield of artificial intelligence (AI), provides the ability to mechanical systems to process selected data displayed as a form of digital inputs and convert it as programmed. Implementing this element within the circuit would enable the performance of analysing data displayed on social media once a graphic user interface (GUI) in the form of a mobile phone has been connected through an A B USB cable. The computer vision is programmed to recognise racial abuse while analysing data, with the ability to understand that Example 1A should = Example 2A. Programming a file with examples enables the data to be read and rewritten/displayed accordingly.

Incorporating image processing, a subset of computer vision as a method of signal processing in which [INPUT] is a digital image and [OUTPUT] is a related characteristic associated with the data programmed. Through reading the image on input the circuit analyses and manipulates the input producing an output as desired. Implementing this element enables computer vision (CV) to understand Example 1A on [INPUT] as racial abuse and process on [OUTPUT] using image processing converting Example 1A as data to Example 2A and an image (positive).

Research: Human Rights Commission and Government Digital Service

Programming Example 1A, the following report published by the Australian Human Rights Commission in association with the Government Digital Service provides an overview of examples of conducting racial material over the internet against the Racial Discrimination Act 1975. Unfortunately, the nature of the internet is rapidly changing, including the configuration of online racist groups and the definitions used. "The material posted on such sites has the capacity to disseminate degrading notions of racial inferiority and cause offence, humiliation and social division." HumanRights.Gov.au. (2019). The following example used within the report published by the Australian Human Rights Commission has been extracted from a site illustrating the domestic relevance of the problem connecting cyber-racism:

Example: "...it is a world run by the Zionist Jewish Influence and Race Tainting Paedophiles that are only here to rape our heritage and destroy the qualities that make us White People great..." HumanRights.Gov.au. (2019).

Programmed Words/Phrases extracted from example: 'Zionist Jewish', 'Paedophiles', 'rape', 'White People'

Understanding the common characteristic within cyber-racism, compared with documented examples, a file will be script recognising the language used as well as intention, repetition and power imbalance. The file will comprise topics considered inherently controversial, such as politics, religion, gender, race and sexual orientation. The following section of this report will conduct building the prototype, once complete the file will then be developed, required and implemented through a script to begin machine learning.

Building the Prototype

Stage 01: Raspberry Pi (RPi) 4 Computer (Model B)

In order to develop and programme AI, Computer Vision and Incorporate image processing, the first step is to build the mechanical system these elements will be applied to.

To begin, the project will be using a Raspberry Pi (RPi) 4 Computer (Model B) With 8GB RAM as a compact computer board this will enable the use of the processor, graphics, RAM, Wi-Fi, Bluetooth, USB Port, Ethernet and HDMI, all aspects that will be required further into the project.

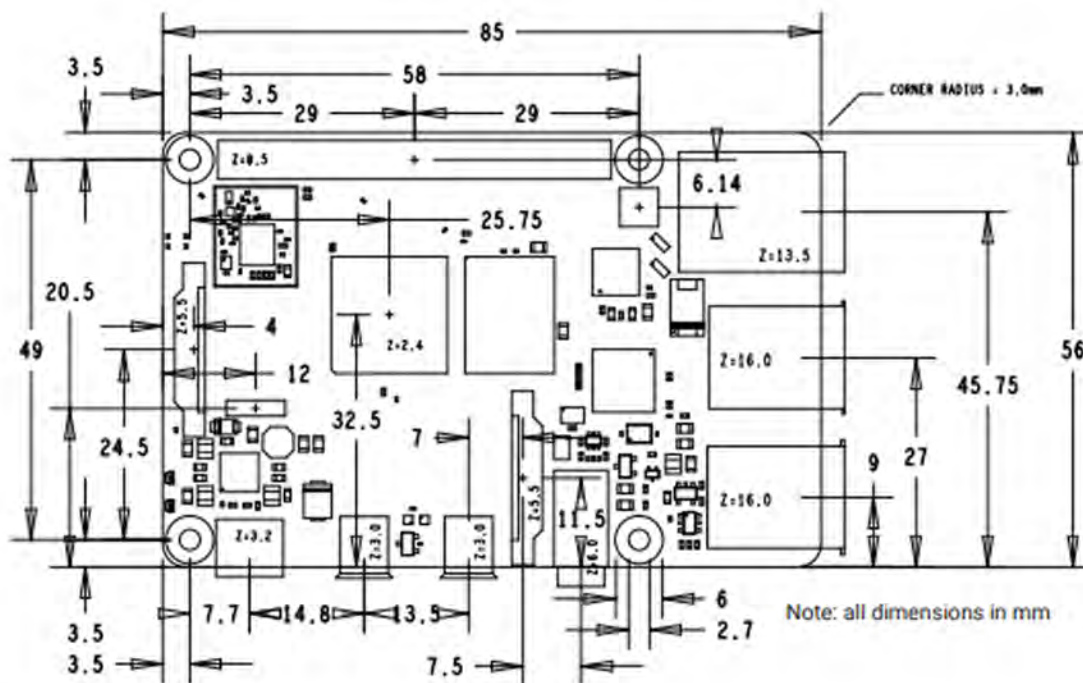


Figure 07 - Physical Specifications RPi
Raspberry Pi 4 Computer Model B. (2019).

When computers are being used, the electronic components generate a substantial amount of heat that can be detrimental to the system if it is not controlled. Particularly when developing a programme onto the board through resource-intensive applications, overheating components will compromise the performance and processing time. I have connected a Heatsink Case Armour Passive Cooling Case for Pi4B - Model P122 to maintain thermal control.

To comply with the recommended application of the case, 3 thermal pads are stuck onto the Raspberry Pi (RPi), the board is then placed on the bottom cover of the case with the screw holes visible ensuring the board and case are aligned. The top case is then aligned and the unit is secured using the recommended screws. Given the case dimensions, the pins become inaccessible and unreadable, by connecting a GPIO Extension Board and Breadboard the pin names are displayed centrally which will be required as pins are defined in code by their physical position, typically read from top to bottom and left to right, when writing the code, the pin is referred to as 3V3 (1), 5V0 (2), SDA1 (3).

01



02



03



04



05



06



07



08



09



10



11



Providing 32GB and 120mb/s transfer speeds a SanDisk Ultra Memory Card will be inserted within the Raspberry Pi (RPi) using a microSDXC Adapter.

12



By connecting a power cable, a Rii Screen (GUI) to one USB Port, and attaching a Rii Wireless Keyboard 2.4G and Mouse the physical structure is complete. This step will be confirmed by the green LED blinking on the board.

13



14



Stage 02: Raspberry Pi OS

Following stage 01, ensuring the structure loads once power has been connected stage 02 moves on to internal development. A window requesting the basic settings will appear, 'Country', 'Language' and 'Time-zone'.

Setting up the operating system by downloading and installing Raspberry Pi OS using Raspberry Pi Imager, onto the SanDisk Ultra Memory Card inserting within Stage 01.

15



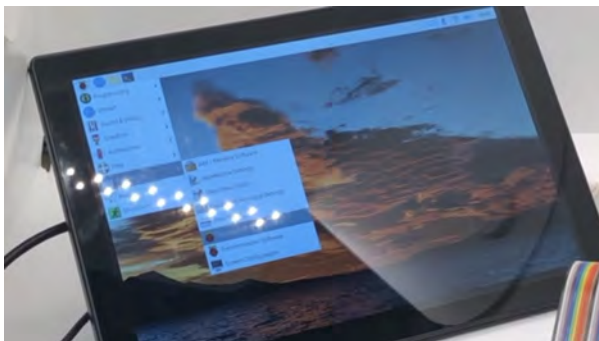
Login Details for Raspberry Pi:
User: pi
Password: Lsbu1354

RPi IP Address: 172.21.0.235

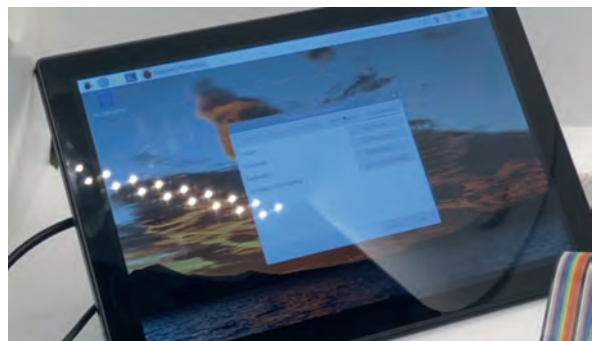
Continuing to explore the Raspberry Pi (RPi) Configuration, this section is found under 'preferences' and provides access to the settings of the Raspberry Pi Board divided into 4 categories, 'System', 'Interfaces', 'Performance' and 'Localisation'. It is important to ensure these settings are up to date as they will be required later in the project enabling aspects such as computer vision to function accurately.

In preparation for Computer Vision, remotely logging into the Raspberry Pi (RPi) is required by installing SSH through PuTTY Configuration. Entering the RPi IP Address (172.21.0.235), the user (pi) and the password (Lsbu1354).

16



17



Stage 03: Circuit Experiments using Python3

Raspberry Pi OS has Python3 pre-installed, this is the programme that will be used to write the script. In order to conduct some basic experiments, I connected an LED, 220 Resistors and an ASAIR AM2303 Temperature Humidity Sensor to the GPIO Extension Board and Breadboard expanding the circuit.

The AM2303 is a sensor that provides digital temperature and humidity readings, by connecting it to the Raspberry Pi (RPI) [OUTPUT] readings will be coded to display on an SSH Terminal.

To program the AM2303 with Python3, using the command prompts below, I installed Adafruit DHT11 Python Library.

COM1: sudo apt-get install git-core

COM2: git clone https://github.com/adafruit/Adafruit_Python_DHT.git

COM3: cd Adafruit_Python_DHT

COM4: sudo apt-get install build-essential python-dev

COM5: sudo python setup.py install

To enable [OUTPUT] to an SSH Terminal, I entered the following code:

```
#!/usr/bin/python
import sys
import Adafruit_DHT
```

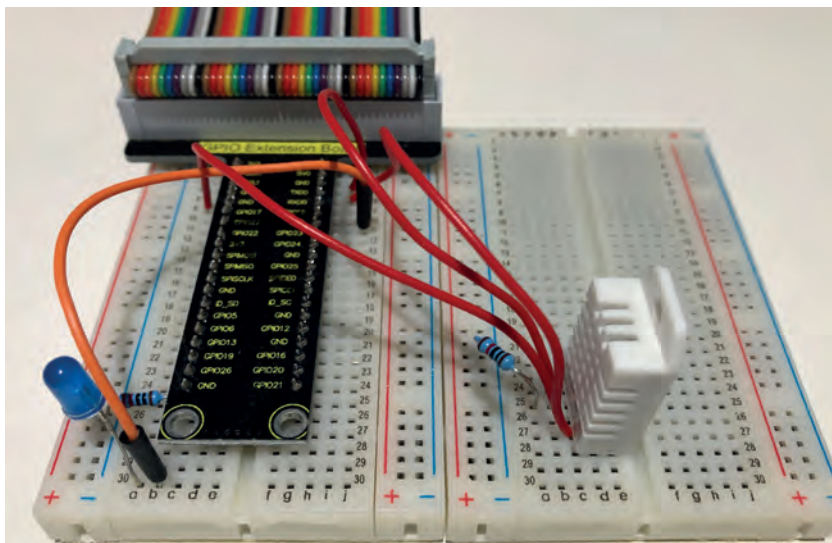
while True:

```
    humidity, temperature = Adafruit_DHT.read_retry(11, 4)
```

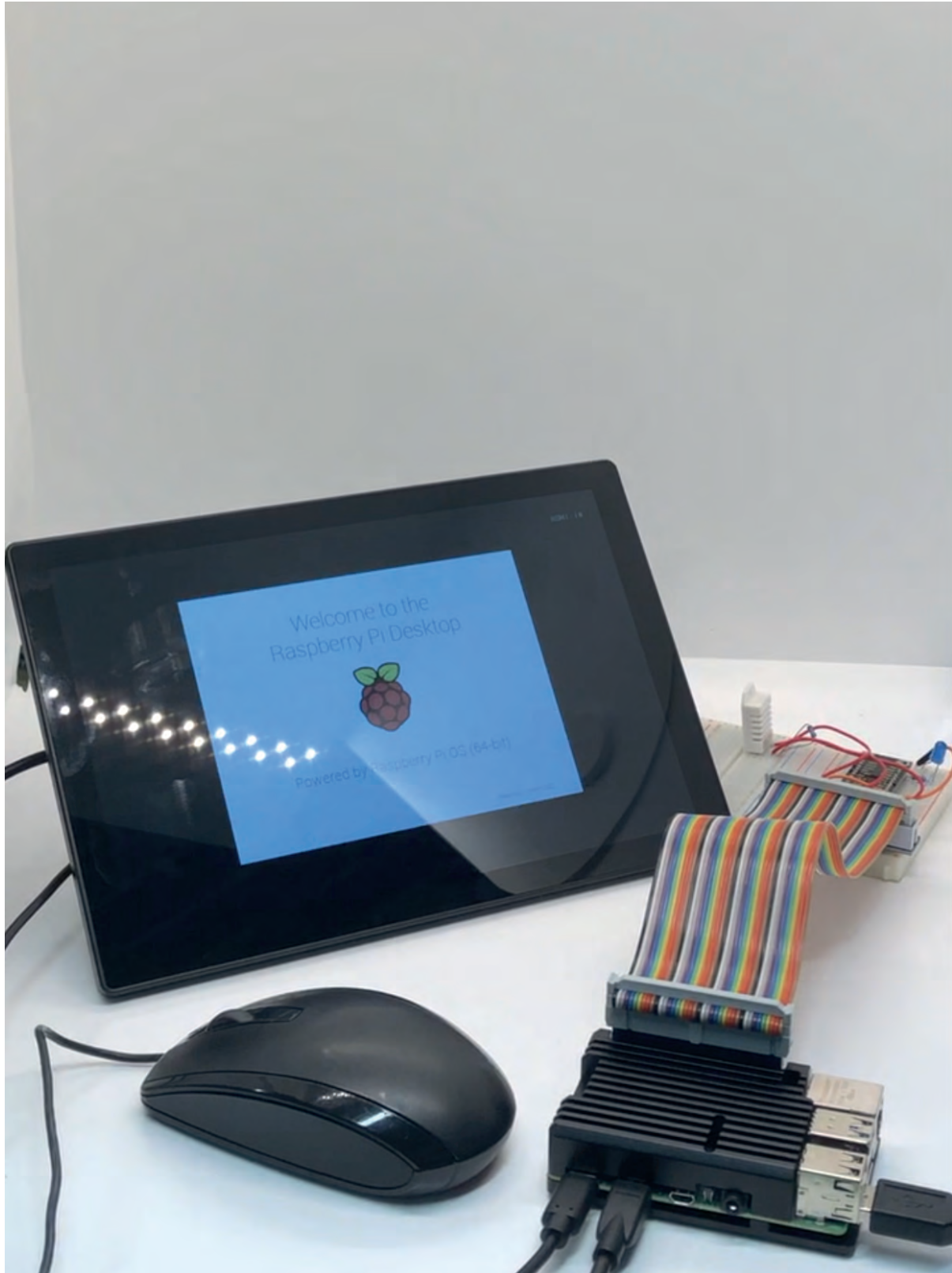
```
    print 'Temp: {0:0.1f} C Humidity: {1:0.1f} %'.format(temperature, humidity)'
```

Pi | 161, C.B. | R. (2015).

18



Final Structural Prototype to Conclude Section 03



Conclusion

Undertaking an investigation into the field of robotics, followed by conducting research into cyber racism and understanding the potential possibilities to respond within architectural design and advanced automation. The report concludes with section 03, as an introduction to building the prototype. Preparing the Raspberry Pi (RPI) for artificial intelligence (AI), computer vision (CV) and image processing enabled perception of the fundamental aspects of RPI, Raspberry Pi OS and Python.

Continuing development, section 04 will introduce CV, AI and Image processing using OpenCV providing a powerful programming framework. Incorporating Python3 as the script language written on OpenCV enables the RPI to capture, process and analyse both written and visual data. The overall aim and objective will be to replicate biological vision as a mechanical system with the ability to read and rewrite data converting [INPUT] to the programmed [OUTPUT] through decision making.

Understanding the purpose of OpenCV as an open-source library with the capability to connect CV, and AI with a GUI the next objective will be to introduce image processing by analysing image arithmetic operation, adding, subtracting and blending alongside performing logical operations, a function that provides the manipulation of an image displayed which will be required when developing the [OUTPUT] of data.

As demonstrated within this report, the field of robotics provides significant technological advancement when applied accordingly to Architectural Design. By introducing a method of interactive design utilising algorithmic machine learning intelligence connecting both physical and virtual space. Architectural possibilities are interminable, achieving an experience manipulating extracted data integrating digital design elements to produce a physical outcome.

Bibliography

Arduino (2019). Arduino - Introduction. [Online] Arduino.cc.
Available at: <https://www.arduino.cc/en/guide/introduction>.
[Accessed 04 April. 2022]

Arduino (2019). Arduino - Setup. [Online]
Available at: <https://www.arduino.cc/en/reference/setup>.
[Accessed 05 April. 2022]

Arduino | 60, C.B. | (2015). How to Set Up an Ultrasonic Range Finder on an Arduino. [Online] Circuit Basics.
Available at: <https://www.circuitbasics.com/how-to-set-up-an-ultrasonic-range-finder-on-an-arduino/>.
[Accessed 06 April. 2022]

GPIO 40pin Breakout Expansion Board - Wiki. (2020) [Online]
Available at: http://wiki.sunfounder.cc/index.php?title=GPIO_40pin_Breakout_Expansion_Board [Accessed 24 May 2022].

HumanRights.Gov.au. (2019). Examples of Racist Material on the Internet | Australian Human Rights Commission. [Online Report]
Available at: <https://humanrights.gov.au/our-work/publications/examples-racist-material-internet>.
[Accessed 23 May. 2022]

Liquid Crystal Displays (LCD) with Arduino (2018). | Arduino Documentation. [Online]
Available at: <https://docs.arduino.cc/learn/electronics/lcd-displays>.
[Accessed 09 April. 2022]

Raspberry Pi 4 Computer Model B. (2019). [Online Report]
Available at: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>.
[Accessed 24 May. 2022]

Pi | 161, C.B. | R. (2015). How to Set Up the DHT11 Humidity Sensor on the Raspberry Pi. [Online] Circuit Basics.
Available at: <https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/>.
[Accessed 25 May. 2022]

Robots In Architecture | RoboticsTomorrow. (2013). [Online]
Available at: <https://www.robotictomorrow.com/article/2013/07/robots-in-architecture/180/#:~:text=In%20architecture%2C%20robots%20have%20mostly>
[Accessed 10 Apr. 2022].

Sidey, E. (2022). Technology. [Submission Report]. EBB_5_511 BA(Hons) Architecture - Technology 2 / 2021/22