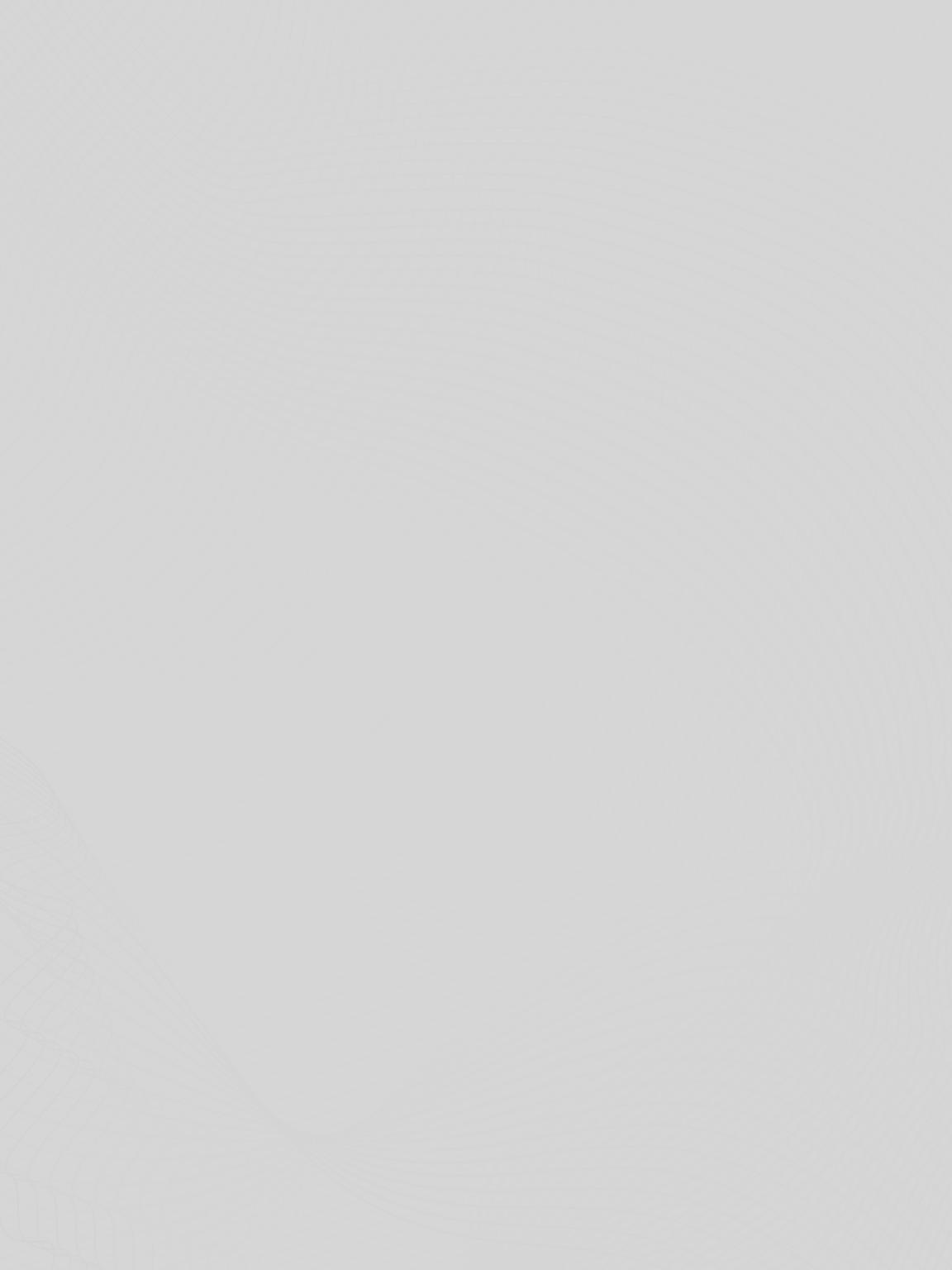


Intelli - Pantry

Table of Contents

[Summary/Abstract](#page3) [3](#page3)

[Introduction](#page4) [4](#page4)

[Experimental Methods](#page5) [5](#page5)

[Results](#page11) [12](#page11)

[Conclusions and recommendations](#page15) [15](#page15)

[Acknowledgements](#page16) [16](#page16)

[Appendices](#page17) [17](#page17)

[References](#page17) [18](#page17)

**Safe Haven** **PAGE 2**

**Summary/Abstract**

Safe Haven is a technology solution for enforcing social distancing measures in public places.

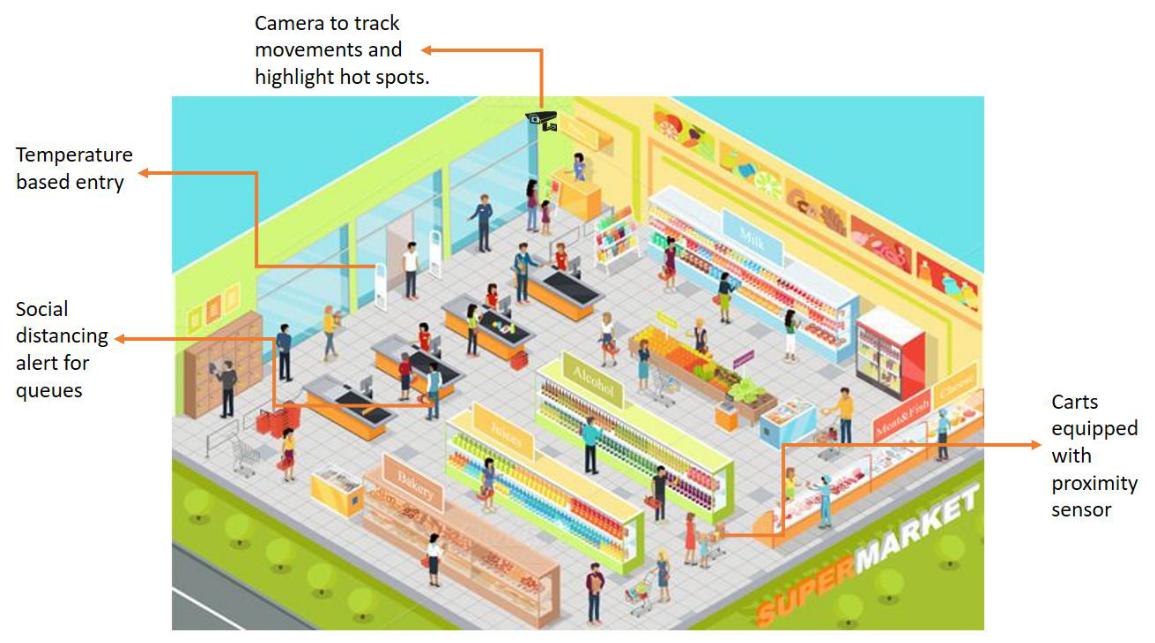
One day I visited the supermarket with my parents during the global pandemic. The supermarket seemed to have implemented some measures for social distancing, but it was still not able to enforce them or monitor them. This got me thinking as to how I could improve upon the measures. I made a list of additional measures that can be implemented:

* Automatic body temperature detection at the door and entry allowed if it is within norms
* Proximity based sensor for all shopping trolleys to provide alert in case another cart is too close
* Monitoring of number of spots in check out queue & number of people standing to enforce proper distance between shoppers
* Motion capture via close circuit camera and identify red zones where improvement can be done

**Safe Haven** **PAGE 3**

**Introduction**

The purpose of this project was to build and test a mock-up of a public place like office or school cafeteria, post office, supermarkets etc. that is equipped with various technologies needed to make it compliant to social distancing norms and other COVID 19 preventive measures like temperature screening etc.



**Safe Haven** **PAGE 4**

**Experimental Methods**

I built a Raspberry Pi based system coupled with a temperature sensor. This setup was then used in a model of a supermarket to demonstrate the usefulness of the ´Safe Haven´ technological package.

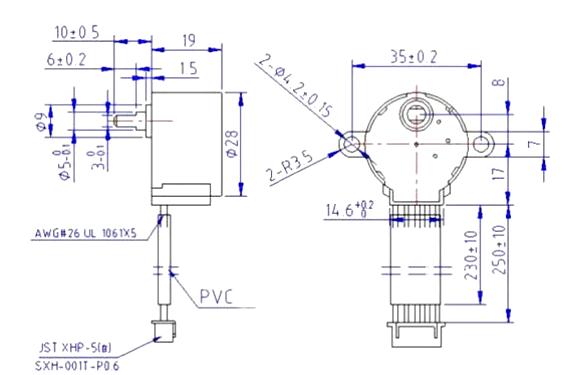
The supermarket doors open only if the temperature detected is optimal.

The temperature sensor MLX3064 was connected to the Raspberry Pi. A DC 5v 4 phase 5 wire stepper motor (28 BYJ – 48) and its driver board (ULN 2003) was also connected to the Pi.

**Component# 1: Stepper motor**

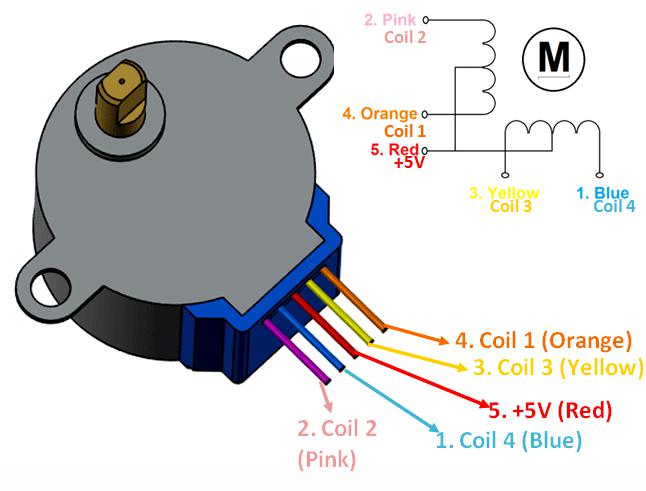
The most commonly used stepper motor is the **28-BYJ48 Stepper Motors**. The motor has a 4 coil unipolar arrangement and each coil is rated for +5V hence it is relatively easy to control with any basic microcontrollers. These motors has a stride angle of 5.625°/64, this means that the motor will have to make 64 steps to complete one rotation and for every step it will cover a 5.625° hence the level of control is also high.

28BYJ-48 Dimensions



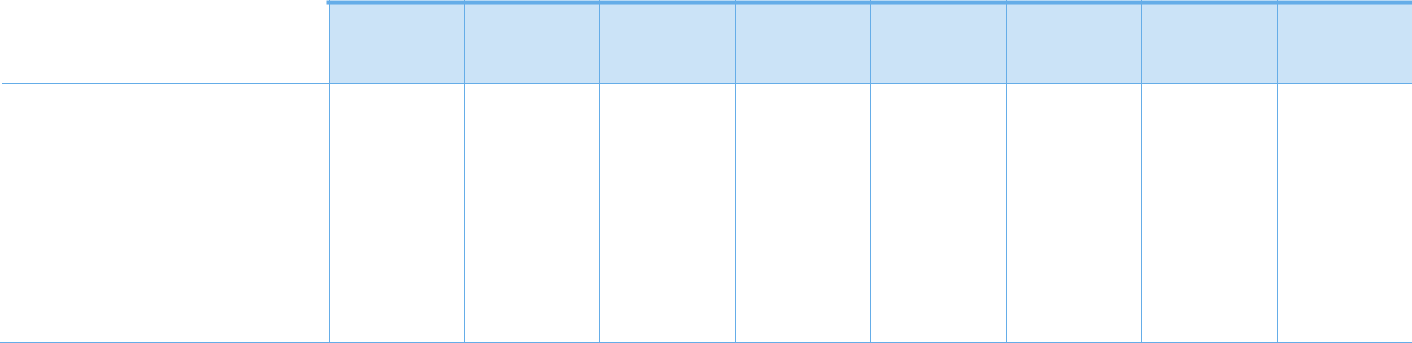
These stepper motors consume high current and hence a driver IC like the [ULN2003](https://components101.com/stepper-motor-driver-ic-uln2003-pinout-datasheet) is mandatory. To know how to make this motor rotate we should look into the coil diagram below.

**Safe Haven** **PAGE 5**



There are four coils in the motor and one end of all the coil is tied to +5V (Red) and the other ends (Orange, Pink, Yellow and Blue) are taken out as wires. The Red wire is always provided with a constant +5V supply and this +5V will be across (energize) the coil only if the other end of the coil is grounded. A stepper motor can be made to rotate only if the coils are energized (grounded) in a logical sequence. This logical sequence can be programmed using a microcontroller or by designing a digital circuit. The sequence in which each coil should be triggered is shown in the table below. Here “1” represent the coil is held at +5V, since both the ends of coil is at +5V (red and other end) the coil will not be energised. Similarly “0” represents the coil is held to ground, now one end will be +5V and the other one is grounded so the coil will be energised.

**Motor Wire Color** **Sequence to Rotate in clockwise Direction**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Step 1** | **Step 2** | **Step 3** | **Step 4** | **Step 5** | **Step 6** | **Step 7** | **Step 8** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Orange** | 0 | | 0 | 1 | 1 | 1 | 1 | | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |
| **Yellow** |  | 1 | 0 | 0 | 0 | 1 |  | 1 | 1 | 1 |
|  |  |  |  |  |  |  |  |  |  |  |
| **Pink** | 1 | | 1 | 1 | 0 | 0 | 0 | | 1 | 1 |
|  |  | |  |  |  |  |  | |  |  |
| **Blue** |  | 1 | 1 | 1 | 1 | 1 |  | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |

**Safe Haven** **PAGE 6**

**Red**

1

1

1

1

1

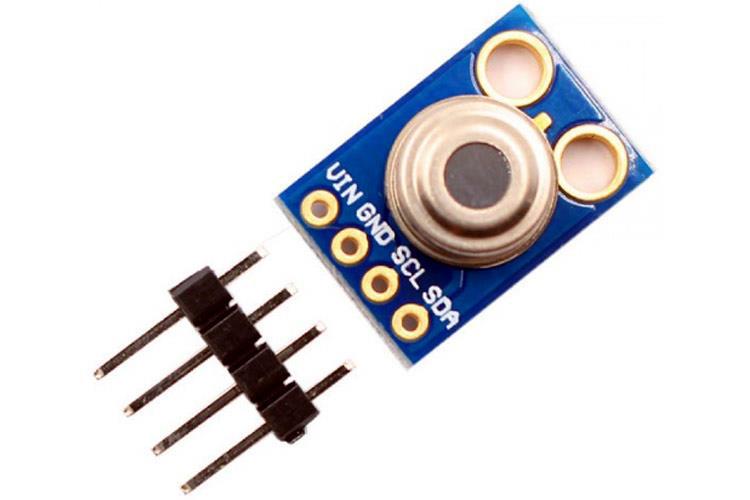
1

1

1

**Component# 2: Non-contact IR Temperature Sensor**

The MLX90614 is a **Contactless Infrared (IR) Digital Temperature Sensor** that can be used to measure the temperature of a particular object ranging from -70° C to 382.2°C. The sensor uses IR rays to measure the temperature of the object without any physical contact and communicates to the microcontroller using the I2C protocol.



MLX90614 Temperature Sensor Specifications

* Operating Voltage: 3.6V to 5V (available in 3V and 5V version)
* Supply Current: 1.5mA
* Object Temperature Range: -70° C to 382.2°C
* Ambient Temperature Range: -40° C to 125°C
* Accuracy: 0.02°C
* Field of View: 80°
* Distance between object and sensor: 2cm-5cm (approx.)

**Safe Haven** **PAGE 7**

Where to use the MLX90614 Temperature sensor

The key feature of MLX90614 is that it is a contactless IR temperature sensor with high accuracy. So it can be used in industries to **measure the temperature of moving objects** like a rotating motor shaft. Due to its high accuracy and precision, it is also used in a wide range of **commercial, health care, and household applications** like room temperature monitoring, body temperature measurement, etc.

Working Principle of MLX90614

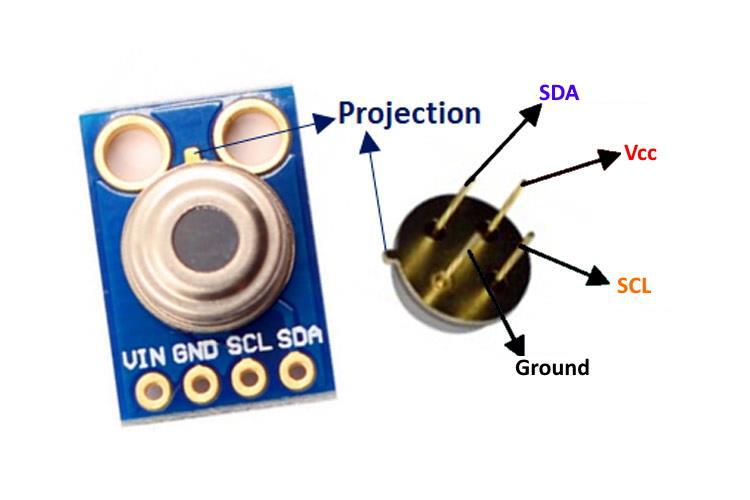
As mentioned earlier, the MLX90614 sensor can measure the temperature of an object without any physical contact with it. This is made possible with a law called **Stefan-Boltzmann Law**, which states that all objects and living beings emit IR Energy and the intensity of this emitted IR energy will be directly proportional to the temperature of that object or living being. So the MLX90614 sensor calculates the temperature of an object by measuring the amount of IR energy emitted from it.

How to use MLX90614 Thermometer Sensor

The MLX90614 Temperature sensor is manufactured by a company called **Melexis**. The sensor is factory calibrated and hence it acts like a **plug and play sensor module** for speeding up development processes.

The MLX90614 consists of two devices embedded as a single sensor, one device acts as a sensing unit and the other device acts as a processing unit. The sensing unit an **Infrared Thermopile Detector** called **MLX81101** which senses the temperature and the processing unit is a **Single Conditioning ASSP** called MLX90302 which converts the signal from the sensor to digital value and communicates using I2C protocol. The MLX90302 has a low noise amplifier, 17-bit ADC and a powerful DSP which helps the sensor to have high accuracy and resolution.

**Safe Haven** **PAGE 8**



The sensor requires no external components and can be directly interfaced with a microcontroller like Arduino. As you can see above the power pins (Vdd and Gnd) can be directly used to power the sensor, typically 5V can be used, but there are other versions of this sensor which can operate on 3.3V and 7V as well. The capacitor C1 is optional and is used to filter noise and provide optimum EMC. The signal pins (SCL and SDA) for used for I2C communication and can be connected directly to microcontroller operating on 5V logic.

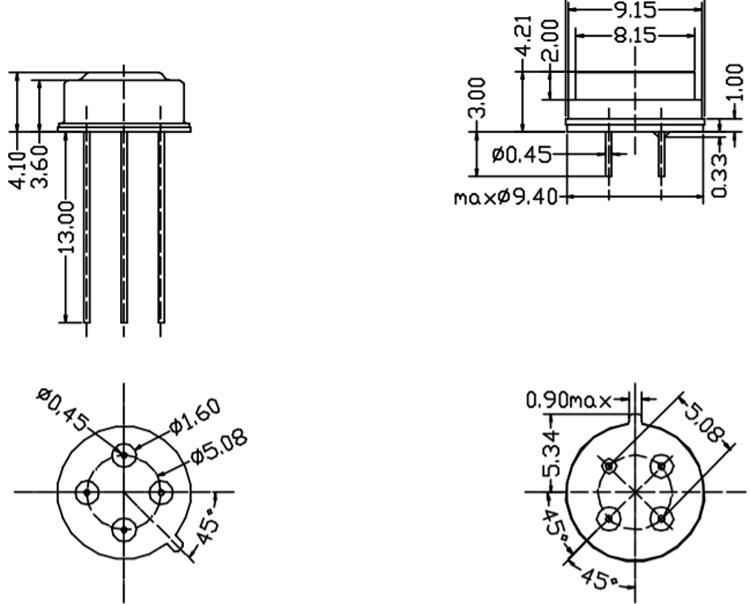
The sensor is also sold as a module as you can see in the pinout image. But the sensor module is very similar to the sensor itself and does not have any additional components other than the sensor itself.

|  |  |  |  |
| --- | --- | --- | --- |
|  | MLX90614 Pinout Configuration | |  |
|  | **Pin No.** | **Pin Name** | **Description** |
|  |  |  |  |
|  | 1 | Vdd (Power supply) | Vdd can be used to power the sensor, typically using 5V |
|  |  |  |  |
|  | 2 | Ground | The metal can also act as ground |
|  |  |  |  |
|  | 3 | SDA – Serial Data | Serial data pin used for I2C Communication |
|  |  |  |  |
|  | 4 | SCL – Serial Clock | Serial Clock Pin used for I2C Communication |
|  |  |  |  |

**Safe Haven** **PAGE 9**

MLX90614 Dimensions

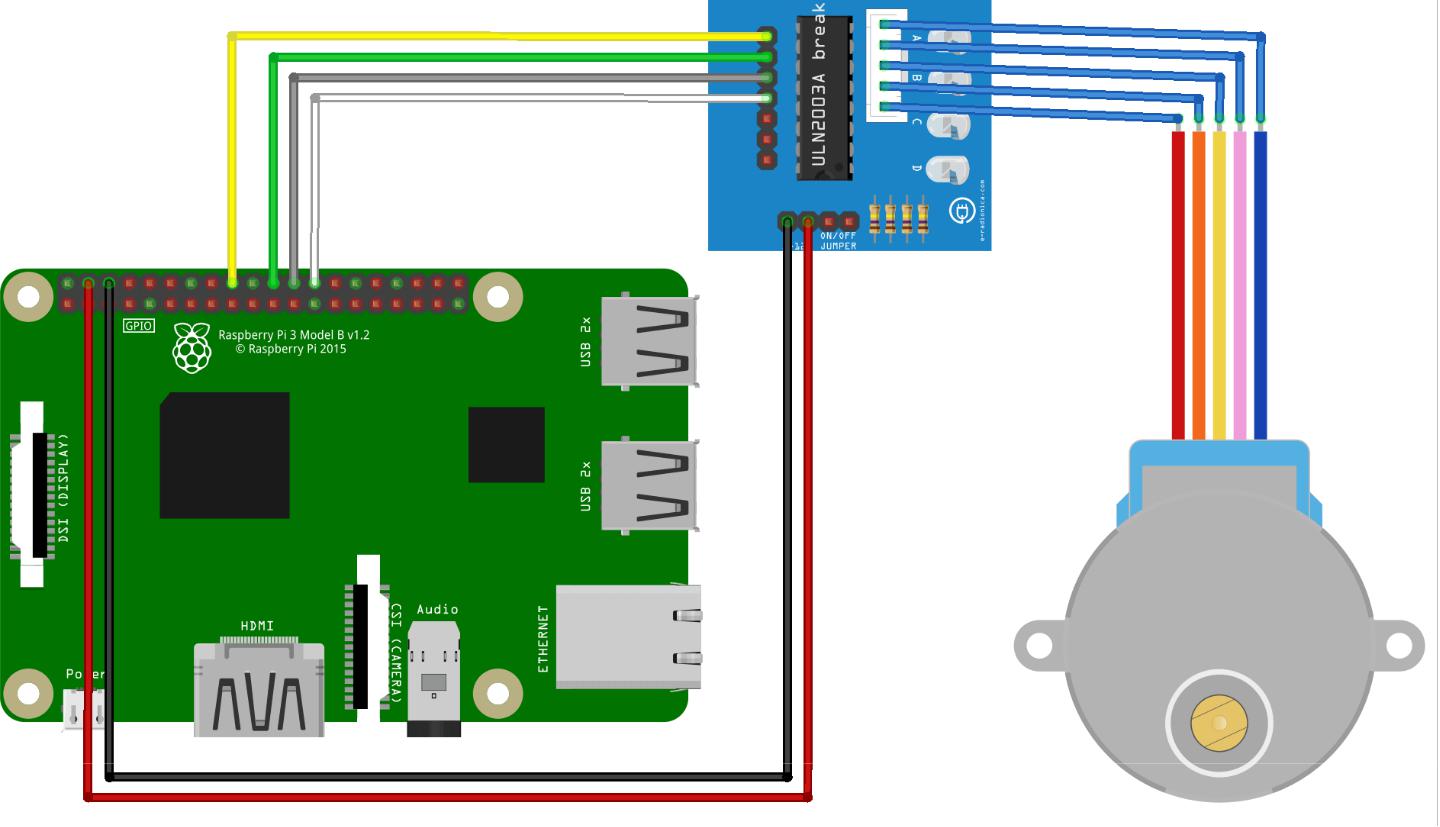
The MLX90614 is available in the standard TO-39 package making it easy to mount on a breadboard. Do note that this is a 4-pin TO39 package and hence the footprint will differ. The dimensions of MLX90614 is shown below.



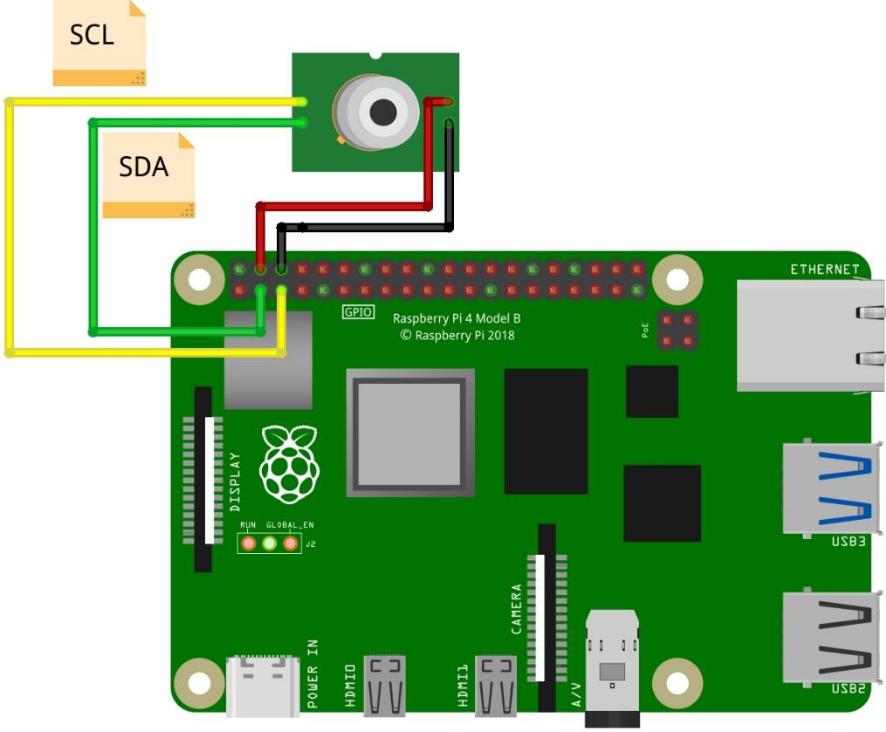
**Safe Haven** **PAGE 10**

**Connections to Raspberry Pi**

1)Motor



2) Temperature Sensor



**Safe Haven** **PAGE 11**

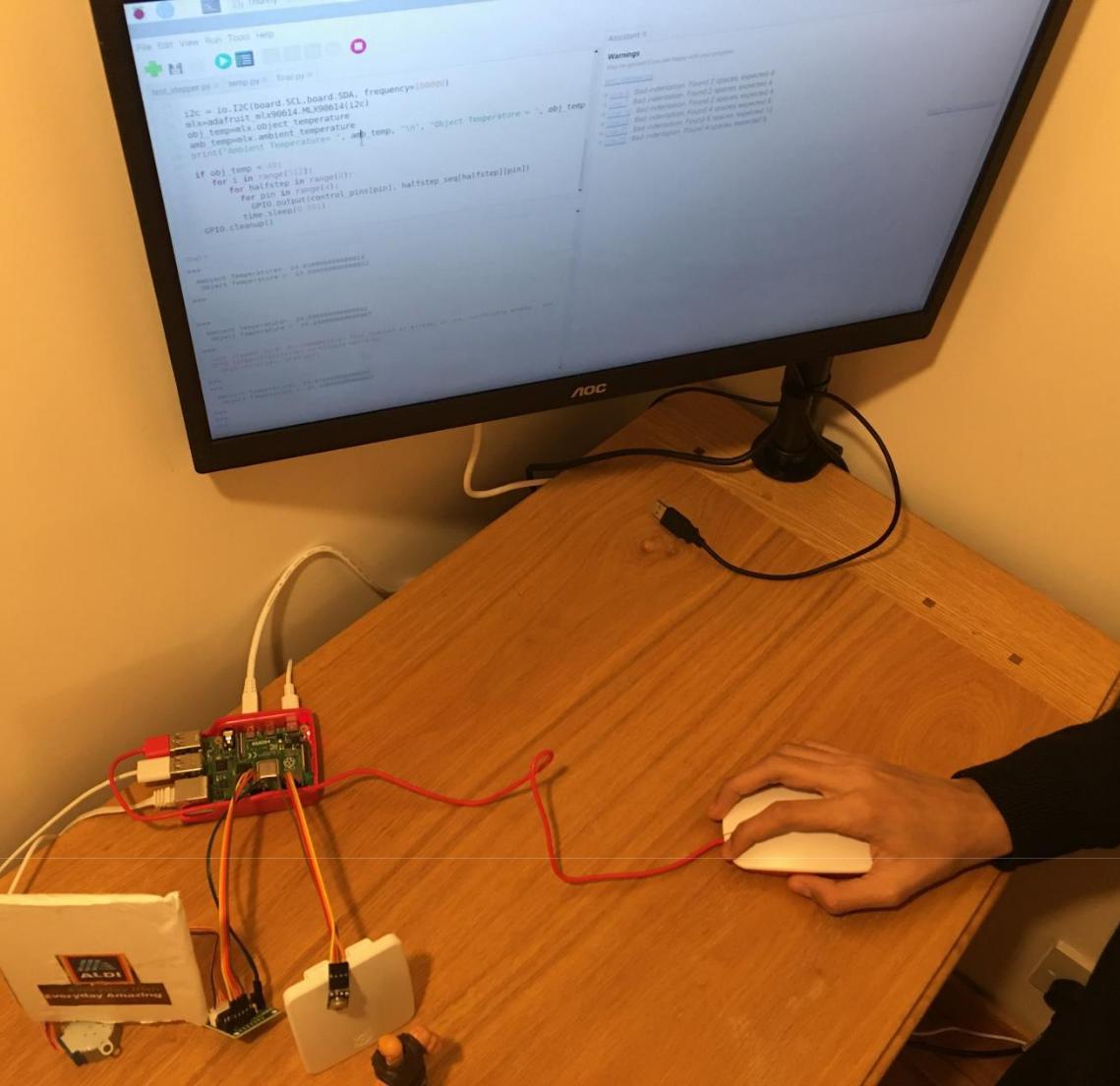
**Software:**

Program written in Python to read object temperature and rotate motor shaft if temperature < 40 degrees Celsius

**Safe Haven** **PAGE 12**

**Results**

Experimental Setup:



**Safe Haven** **PAGE 13**

Temperature Check at the Supermarket Entry:



Supermarket Door rotated by motor connected to the temperature sensor



**Safe Haven** **PAGE 14**

**Conclusions and recommendations**

This project helped me understand the integration of a sensor and a motor with Raspberry Pi. It demonstrated the real life application of a micro-computer especially with respect to health applications.

The current setup that I demonstrated can be triggered in two ways:

* Motion sensor – this can trigger the temperature sensor which in turn can turn the motor.
* Loop – an infinite loop can be inserted into the Python code coupled with a delay timer of a few seconds to gather the object temperature and then drive the motor if required

**Safe Haven** **PAGE 15**

**Acknowledgements**

I consulted the software-programming teacher at my school to arrive at variables needed for scripting the algorithm.

**Safe Haven** **PAGE 16**

**Appendices**

**Safe Haven** **PAGE 17**

**References**

<https://www.omega.co.uk/technical-learning/infrared-temperature-measurement-theory-application.html>

[https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi\_DATA\_2711\_1p0](https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf) [\_preliminary.pdf](https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf)

**Safe Haven** **PAGE 18**