A Report on

# Temperature based fan controller using Raspberry

For

**Mini Project 1-b(REV- 2019 'C' Scheme) of Second Year, (SE Sem-IV)**

In

**Electronics & Telecommunication Engineering**

by

1. Kshurugal Rawat (SEET1/ 31)

2. Sneha Jadhav (SEET1/63)

3. Hemangini Kundaram (SEET1/64)

4. Mansi Bole (SEET1/65)

Under the guidance of

Name of Mini Project Guide

**Mrs. Ruchi Chauhan**

**UNIVERSITY OF MUMBAI**

**AY 2021-2022**

**ATHARVA COLLEGE OF ENGINEERING**

**MALAD**

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

ATHARVA
COLLEGE OF ENGINEERING

**ATHARVA COLLEGE OF ENGINEERING**

**MALA D**

# CERTIFICATE

This is to certify that the project entitled **Temperature based fan controller using Raspberry** is a bonafide work of

1. Kshurugal Rawat (SEET1/ 31)

2. Sneha Jadhav (SEET1/63)

3. Hemangini Kundaram (SEET1/64)

4. Mansi Bole (SEET1/65)

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Mini Project 1-b (REV-2019 'C' Scheme) of Second Year, (SE Sem-IV)** in **Electronics & Telecommunication Engineering** as laid down by **University of Mumbai** during academic year **2021-22**

(______________________)

**Examiner/Reviewer-1**

(______________________)

**Examiner/ Reviewer -2**

Ruchi

**PROF. RUCHI CHAUHAN**

**NAME OF GUIDE**

**HEAD OF DEPARTMENT**

**PRINCIPAL**

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

# INDEX

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

# 1.) INTRODUCTION

## 1.) Need:

Automation can be defined as the technology by which a process or procedure is performed without human assistance. In other words, Automation or automatic control, is the use of various control systems for operating equipment such as machinery, processes in factories, boilers and heat treating ovens, switching on telephone networks, steering and stabilization of ships, aircraft and other applications and vehicles with minimal or reduced human intervention, with some processes have been completely automated.

Automation has been achieved by various means including mechanical, hydraulic, pneumatic, electrical, electronic devices and computers, usually in combination. Complicated systems, such as modern factories, airplanes and ships typically use all these combined techniques. The benefit of automation include labor savings, savings in electricity costs, savings in material costs, and improvements to quality, accuracy and precision. The term automation, inspired by the earlier word automatic (coming from automaton), was not widely used before 1947, when Ford established an automation department. It was during this time that industry was rapidly adopting feedback controllers, which were introduced in the 1930. , Raspberry Pi is used as the main node which is used for both monitoring and controlling purposes. The system is placed in a network so that different devices and components can communicate and interact among each other or with end-users or other entities in the network. In other words, different devices and the appliances in the industry, as a system, being connected to the Internet, can be controlled remotely or continuously monitored. Raspberry Pi, a fully functional mini-computer, are both cheap solutions for harnessing the Internet of Things at industry. Unlike your regular computer, the devices are very good at reading the world around them as they both include plenty of inputs and outputs for sensory add-ons to test light, temperature, humidity and more.

In various heating system for several application as is required by the Industry. In the hazardous working environment, human safety is an important concern. Coal mines is a place in which human lives are more dangerous and many workers are injured due to explosions and leakage of toxic gases. Fire accidents can also happen.

## 1.2) Definition:

Temperature Controlled DC Fan is a system which automatically turns on a DC Fan when the ambient temperature increases above a certain limit. Generally, electronic devices produce more heat. So this heat should be reduced in order to protect the device. There are many ways to reduce this heat. One way is to switch on the fan spontaneously.

This article describes two such circuits that automatically, switch the fan when it detects the temperature inside

### 1.3)Comparative Study

Temperature Based Fan Speed Controller-Biman Kr Pal , Sagar Ghosh ,Subhankar Paul, This project is a standalone automatic fan speed controller that controls the speed of an electric fan according to our requirement. Use of embedded technology makes this closed loop feedback control system efficient and reliable.

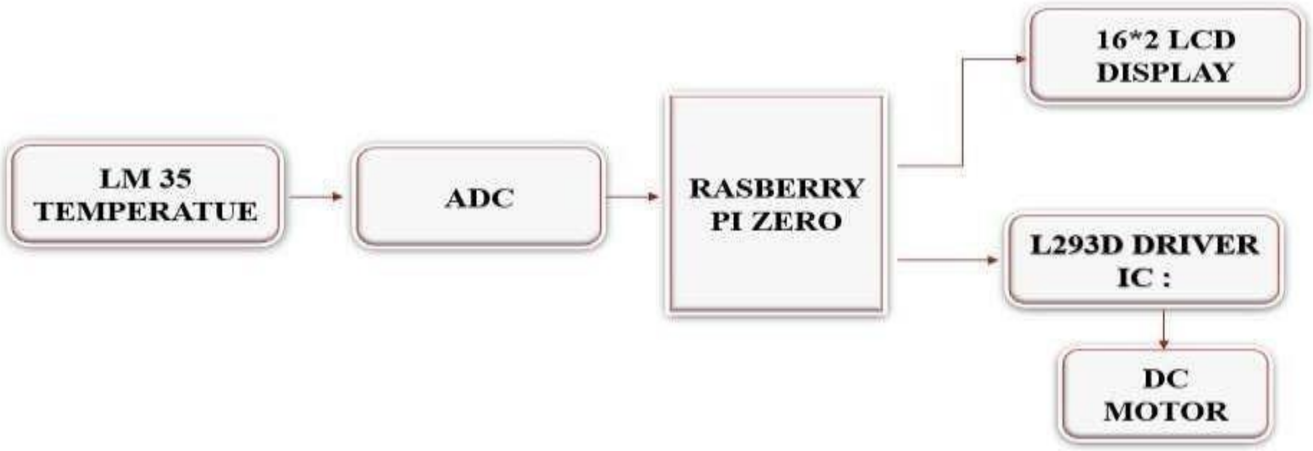
IoT based Temperature and Humidity Controlling using Raspberry Pi-Lalbihari Barik ,In this process they have used IOT as than main platform to link sensors to raspberry pi. They have used web servers to display the data which will be receive from sensors and raspberry pi .

## 2.) PROBLEM STATEMENT

This system is used to control the cooling system automatically based on the room temperature. The system uses an Raspberry board to implement a control system. Since this system is proposed to control the cooling system and it is very important to know raspberry controlled system well.

## 3.) MINI PROJECT DESIGN (PRINCIPLE AND WORKING)

### 3.1) BLOCK DIAGRAM:



**Fig; Block Diagram of Temperature Based fan Controller using Raspberry**

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

### 3.2) Circuit Diagram and working

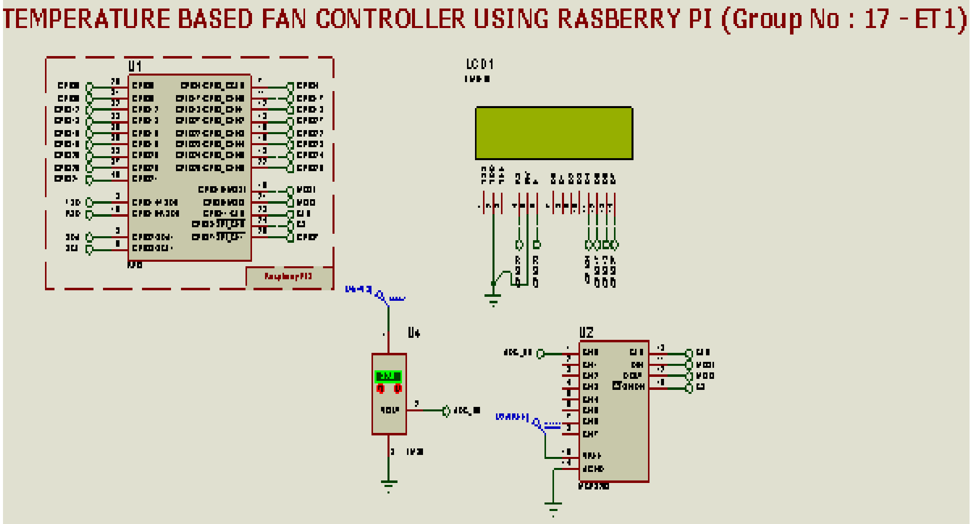

TEMPERATURE BASED FAN CONTROLLER USING RASBERRY PI (Group No : 17 - ET1)
U1
LCD1
U4
U2

**Fig: Circuit diagram of Temperature Based Fan Controller using Raspberry**

**WORKING OF TEMPERATURE BASED FAN CONTROLLER USING RASPBERRY:**

Lm36 out pin is connected to A to D converter (mcp3208). A to D converter is used to lm35 analog signal is converted into digital signal and then digital signal goes to raspberry Pi processor it is process the code and output display on 16x2 LCD display. When the temperature exceeds 30°C the fan starts rotating.

**<u>RESULT: -</u>**

We have created a project Temperature based Fan controller using Raspberry Pi Zero. In this project the as the temperature of room would increases above 45 degrees then the fan would increase the speed according to the temperature of the room.

## <u>4.) COMPONENTS/TOOLS TO BE USED</u>

### 4.1) Components:

| Sr. No. | Component | Quantity |
|---|---|---|
| **1** | 16X2 LCD DISPLAY | 1 |
| **2** | Raspberry Pi Zero | 1 |
| **3** | LM35 Temperature Sensor | 1 |
| **4** | Power Supply | 1 |
| **5** | L293D Driver IC | 1 |
| **6** | Red LED | 1 |
| **7** | Green LED | 1 |
| **8** | DC Motor | 1 |

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

## Raspberry Pi Zero :

The Raspberry Pi Zero is a computer the size of a stick of gum, which is designed to be as cheap as possible, with a starting price of $5 (or $10 for a version with Wi-Fi). With dimensions of just 2.6 x 1.2 x 0.2 inches (66.0mm x 30.5mm x 5.0mm), about the size of a USB Flash drive, the Pi Zero is small enough to be embedded in a project and cheap enough to be left there! The same Raspberry Pi OS and software runs on the Raspberry Pi Zero as on larger Pis such as the Raspberry Pi 4, though the smaller board cannot handle the new, 64-bit version of the OS.

## Raspberry Pi Zero Specification:

| | |
|---|---|
| CPU | 1-GHz, Broadcom BCM2835 |
| RAM | 512MB |
| Wireless (Pi Zero W only) | 802.11n / Bluetooth 4.1 / LE |
| Ports | Micro USB, Mini HDMI |
| I/O | 40 GPIO Pins, CSI camera connector (not on version 1.2) |
| Size | 2.6 x 1.2 x 0.2 inches (66.0mm x 30.5mm x 5.0mm) |
| Weight | 0.31 ounces (9 grams) |

## Power Supply :

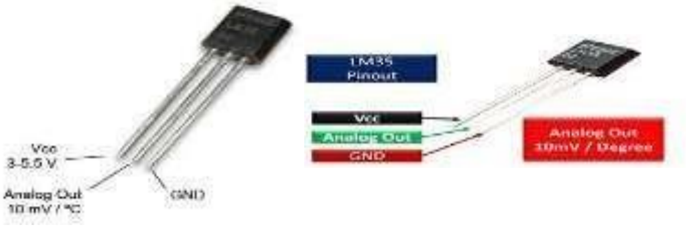
As mentioned already in the above theory portion, Raspberry Pi needs a 5V power supply. If supply exceeds 5V then it can’t be guaranteed to work properly. And the power supply also needs to supply at least 500 milliamps (mA), and preferably more like 1 amp (A). If the supply is 500 mA or less, it is likely to have the malfunction of the keyboard and mouse. It is not a good idea to power the Raspberry Pi from the USB port of the computer and hub as they mostly provide current less than required. Hence, the Raspberry Pi requires a Micro-USB connection that is capable of supplying at least 700 mA (or 0.7 A) at 5V.

## LCD 16×2:

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc.

These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

**LM35 Temperature Sensor:**



**LM35 Regulator Features:**

- Minimum and Maximum Input Voltage are 35V and -2V respectively. Typically 5V.
- Can measure temperature ranging from -55°C to 150°C.
- Output voltage is directly proportional (Linear) to temperature (i.e.) there will be a rise of 10mV (0.01V) for every 1°C rise in temperature.
- ±0.5°C Accuracy.
- Drain current is less than 60uA.

**L293D Driver IC:**

Motor Driver ICs are primarily used in autonomous robotics only. Also, most microprocessors operate at low voltages and require a small amount of current to operate while the motors require a relatively higher voltage and current. Thus current cannot be supplied to the motors from the microprocessor. This is the primary need for the motor driver IC.

The L293D is a 16-pin IC, with eight pins, on each side, dedicated to the controlling a motor. There are 2 INPUT pins, 2 OUTPUT pins, and 1 ENABLE pin for each motor. L293D consists of two H-bridge. H-bridge is the simplest circuit for controlling a low current-rated motor.

**DC MOTOR:**

A DC motor is an electric motor that runs on direct current power. In an electric motor, the operation is dependent upon simple electromagnetism. A current-carrying conductor generates a magnetic field, when this is then placed in an external magnetic field, it will encounter a force proportional to the current in the conductor and the strength of the external magnetic field. It is a device that converts electrical energy to mechanical energy. It works on the fact that a current-carrying conductor placed in a magnetic field experiences a force that causes it to rotate concerning its original position.

## 4.2) SOFTWARE

**Proteus:**

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by the electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit diagrams It was developed in Yorkshire, England by LaCenter Electronics Ltd and is available in English, French, and Spanish and Chinese languages.

## 6.) PROPOSED EXECUTIONS STEPS

**Implemented Steps for Simulation**

1. We Install Proteus software of version 8.11.

2. With the help of block diagram we mounted all component and check whether the circuit is working or not.

3. For raspberry we designed required program code. We check for the code execution.

4. After successfully mounting the circuit on software we started working on hardware

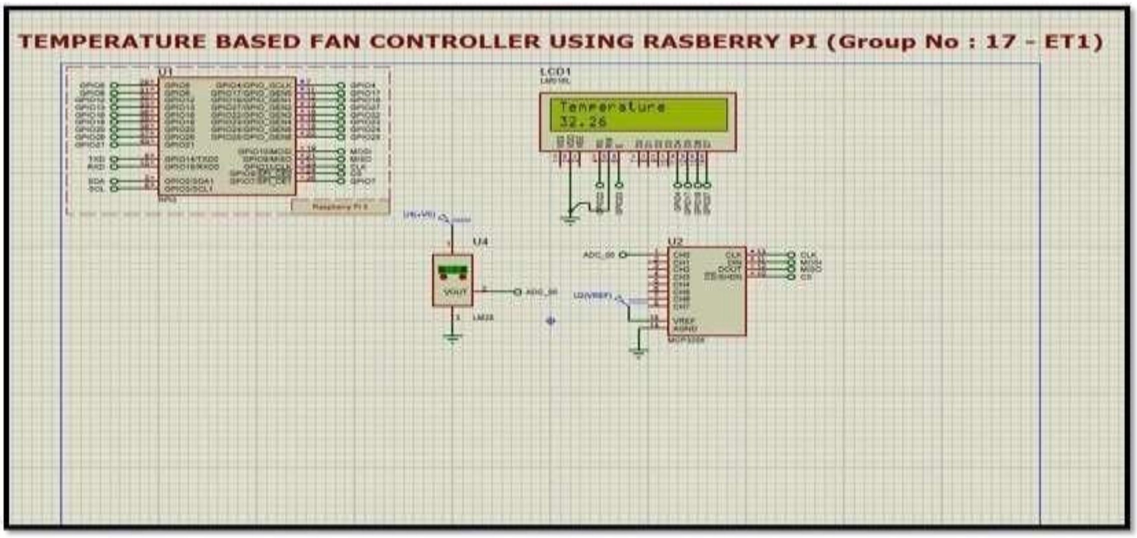
5. We gathered the components required for our project.

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

6. We have done literature survey on our project topic

## SCHEMATIC DIAGRAM:



## Code:

```
#!/usr/bin/python import spidev import time import os
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False)

# Open SPI bus spi = spidev.SpiDev() spi.open(0,0)

# Define GPIO to LCD mapping

LCD_RS = 15

LCD_E  = 16

LCD_D4 = 7

LCD_D5 = 11

LCD_D6 = 12

LCD_D7 = 13

 # Define sensor channels temp_channel = 0

''' define pin for lcd '''

# Timing constants

E_PULSE = 0.0005

E_DELAY = 0.0005

delay = 1
```

```
GPIO.setup(LCD_E, GPIO.OUT)  # E
GPIO.setup(LCD_RS, GPIO.OUT) # RS

GPIO.setup(LCD_D4, GPIO.OUT) # DB4

GPIO.setup(LCD_D5, GPIO.OUT) # DB5

GPIO.setup(LCD_D6, GPIO.OUT) # DB6 GPIO.setup(LCD_D7, GPIO.OUT) # DB7
# Define some device constants

LCD_WIDTH = 16    # Maximum characters per line

LCD_CHR = True

LCD_CMD = False

LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line

LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line

''' Function Name :lcd_init()

Function Description: This function is used to initialize LCD by sending the different commands '''

def lcd_init():  # Initialise display
  lcd_byte(0x33,LCD_CMD) # 110011 Initialise lcd_byte(0x32,LCD_CMD) # 110010 Initialise
lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction lcd_byte(0x0C,LCD_CMD) # 001100
Display On,Cursor Off, Blink Off lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines,
font size  lcd_byte(0x01,LCD_CMD) # 000001 Clear display  time.sleep(E_DELAY)
''' Function Name :lcd_byte(bits ,mode)

Function Name: the main purpose of this function is to convert the byte data into bit and send it to LCD port '''

def lcd_byte(bits, mode):  # Send byte to data
pins
  # bits = data

  # mode = True for character

  #        False for command

  GPIO.output(LCD_RS, mode) # RS

  # High bits

  GPIO.output(LCD_D4, False)

  GPIO.output(LCD_D5, False)

  GPIO.output(LCD_D6, False) GPIO.output(LCD_D7,
False)   if   bits&0x10==0x10:   GPIO.output(LCD_D4,
True)  if bits&0x20==0x20:  GPIO.output(LCD_D5,
```

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

```
True)   if bits&0x40==0x40:     GPIO.output(LCD_D6,
True)  if bits&0x80==0x80:
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin lcd_toggle_enable()
  # Low bits

  GPIO.output(LCD_D4, False)

  GPIO.output(LCD_D5, False)

  GPIO.output(LCD_D6, False) GPIO.output(LCD_D7,
False)   if   bits&0x01==0x01:   GPIO.output(LCD_D4,
True)   if   bits&0x02==0x02:   GPIO.output(LCD_D5,
True)   if   bits&0x04==0x04:   GPIO.output(LCD_D6,
True)  if bits&0x08==0x08:
    GPIO.output(LCD_D7, True)


  # Toggle 'Enable' pin lcd_toggle_enable()
''' Function Name : lcd_toggle_enable()

Function Description: basically this is used to toggle Enable pin '''

def lcd_toggle_enable(): # Toggle enable
time.sleep(E_DELAY) GPIO.output(LCD_E,
True) time.sleep(E_PULSE)
GPIO.output(LCD_E, False)
time.sleep(E_DELAY)
''' Function Name :lcd_string(message,line)

Function  Description :print the data on lcd  '''

def lcd_string(message,line): # Send string to
display message = message.ljust(LCD_WIDTH," ")

   lcd_byte(line, LCD_CMD) for i in
range(LCD_WIDTH):
    lcd_byte(ord(message[i]),LCD_CHR)

# Function to read SPI data from MCP3008 chip # Channel must be an
integer 0-7 def ReadChannel(channel):
  adc = spi.xfer2([1,(8+channel)<<4,0])  data = ((adc[1]&3) << 8)
```

ATHARVA COLLEGE OF ENGINEERING MUMBAI

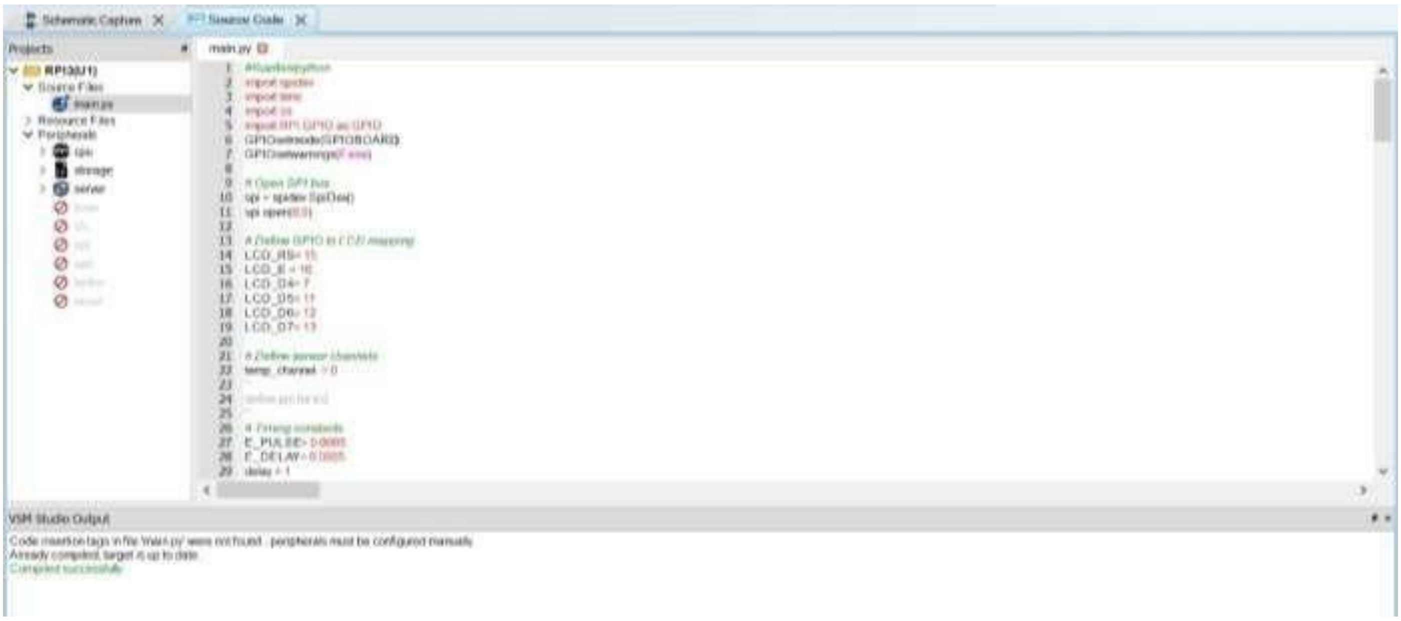
PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

```
+ adc[2]  return data
```

```
# Function to calculate temperature from
# TMP36 data, rounded to specified # number of decimal
places. def ConvertTemp(data,places):
  # ADC Value
  # (approx)  Temp  Volts
  #    0      -50    0.00
  #   78      -25    0.25
  #  155        0    0.50
  #  233       25    0.75
  #  310       50    1.00
  #  465      100    1.50
  #  775      200    2.50
  # 1023      280    3.30
    temp = ((data * 330)/float(1023)) temp = round(temp,places)
   return temp
 # Define delay between readings delay = 5 lcd_init()
lcd_string("welcome ",LCD_LINE_1) time.sleep(2) while 1:
  temp_level = ReadChannel(temp_channel)    temp       =
ConvertTemp(temp_level,2)
    # Print out results lcd_string("Temperature ",LCD_LINE_1) lcd_string(str(temp),LCD_LINE_2)
time.sleep(1)
```

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI



```
#!/usr/bin/python
import spidev
import time
import os
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

# Open SPI bus
spi = spidev.SpiDev()
spi.open(0,0)

# Define GPIO to LCD mapping
LCD_RS= 15
LCD_E = 16
LCD_D4= 7
LCD_D5= 11
LCD_D6= 12
LCD_D7= 13

# Define sensor channels
temp_channel = 0

define pin for lcd

# Timing constants
E_PULSE= 0.0005
E_DELAY= 0.0005
delay = 1
```

```
Code insertion tags in file 'main.py' were not found - peripherals must be configured manually
Already compiled, target is up to date
Compiled successfully
```

ATHARVA COLLEGE OF ENGINEERING MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI

# REFERENCES

- https://www.digikey.in/en/maker/projects/control-a-dc-fan-with-a-raspberry-pi/f3fa09ab84c049d08474b625bee8d8f2
- https://www.researchgate.net/publication/325478216_Controlling_the_temperature_reactor_based_on_Raspberry_Pi_system_control
- https://robu.in/working-principle-of-dc-motor/
- https://2020.robotix.in/tutorial/auto/motor_driver/#:~:text=How%20Motor%20Driver%20Operates%3F,apply%20voltage%20to%20the%20motors
- https://www.google.com/search?q=l293d+motor+driver+ic&tbm=isch&ved=2ahUKEwiz8p_EiKz3AhWVgGMGHQ42DlAQ2-cCegQIABAA&oq=l293&gs_lcp=CgNpbWcQARgDMgcIABCxAxBDMgcIABCxAxBDMgQIABBDMgQIABBDMgQIABBDMgUIABCABDIFCAAQgAQyBAgAEEMyBQgAEIAEMgUIABCABDoHCCMQ7wMQJzoKCCMQ7wMQ6gIQJ1D9B1iOP2DEV2gBcAB4BIAB8AGIAY8QkgEGMC4xMS4ymAEAoAEBqgELZ3dzLXdpei1pbWewAQrAAQE&sclient=img&ei=eO1kYrPWLpWBjuMPjuy4gAU&bih=656&biw=1536&rlz=1C1RXQR_enIN977IN977
- https://www.google.com/search?q=16*2+lcd+display&rlz=1C1RXQR_enIN977IN977&sxsrf=APq-WBsAiAu0_M0ufKGpg2c1pPFEzRKCiw:1650780580068&source=lnms&tbm=isch&sa=X&sqi=2&ved=2ahUKEwjYvc7whKz3AhXTGLkGHfujBzkQ_AUoAnoECAIQBA&biw=1536&bih=656&dpr=1.25#imgrc=DGxXZ_NXx9NujM

MUMBAI

PRINCIPAL
ATHARVA COLLEGE OF ENGINEERING
MUMBAI