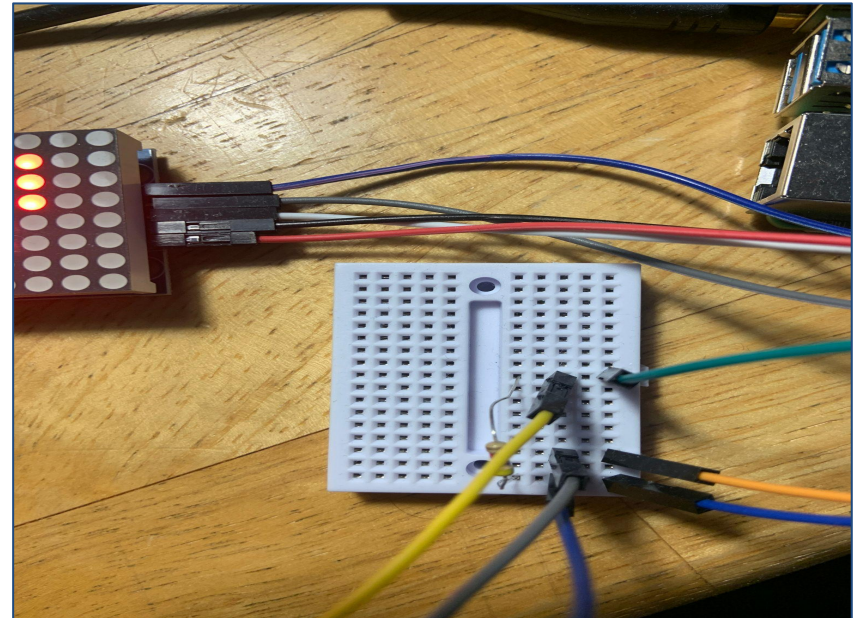
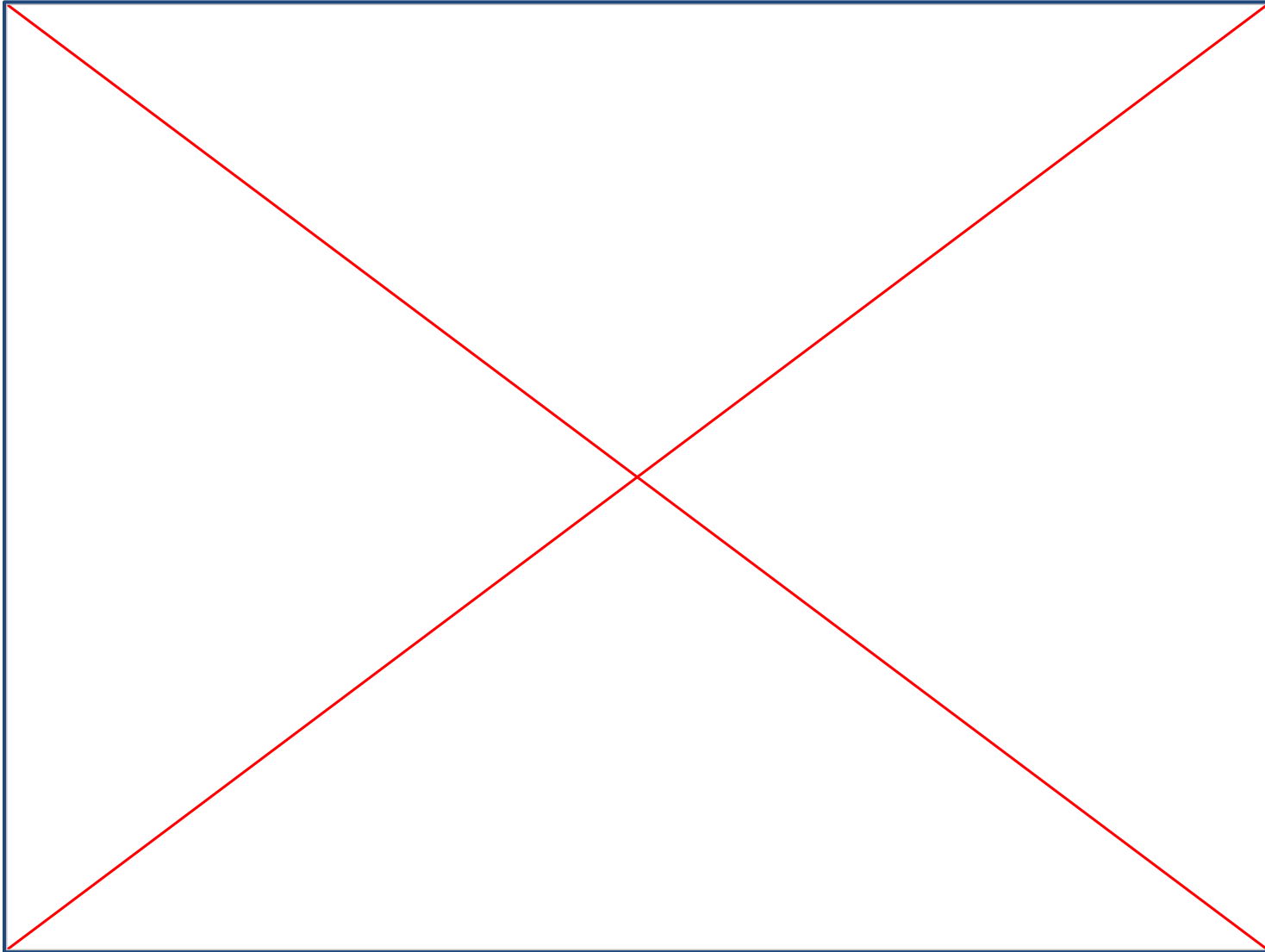


# SmartTemp Control: LED-Display Temperature Sensing with Automated AC Activation

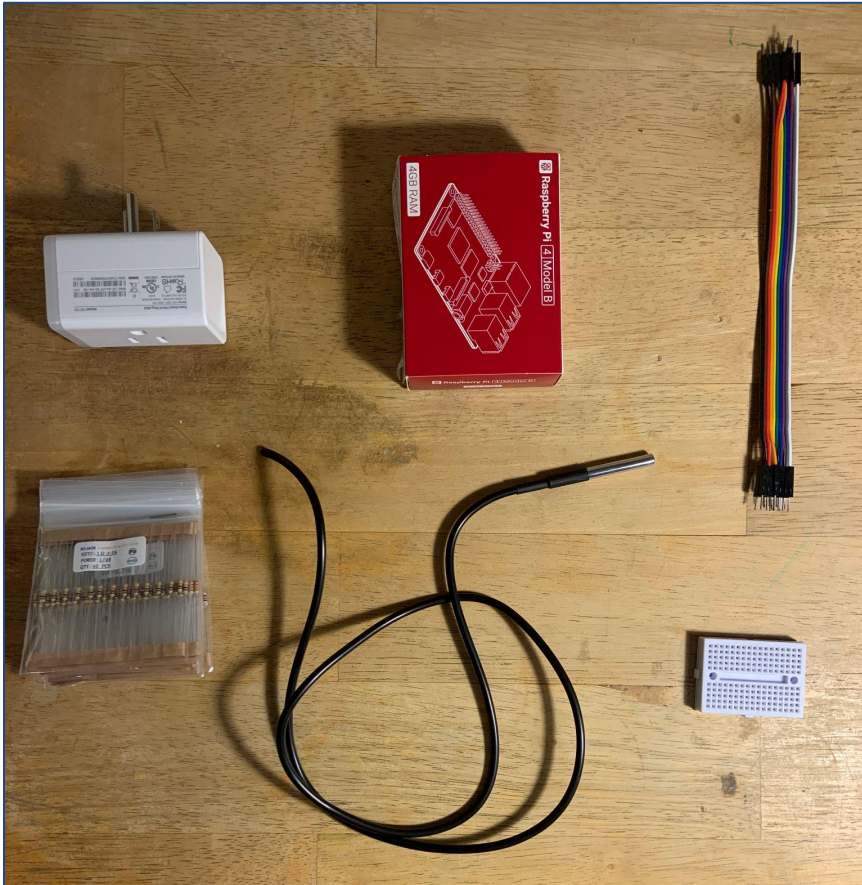


Wilson Heath

# Project Demonstration



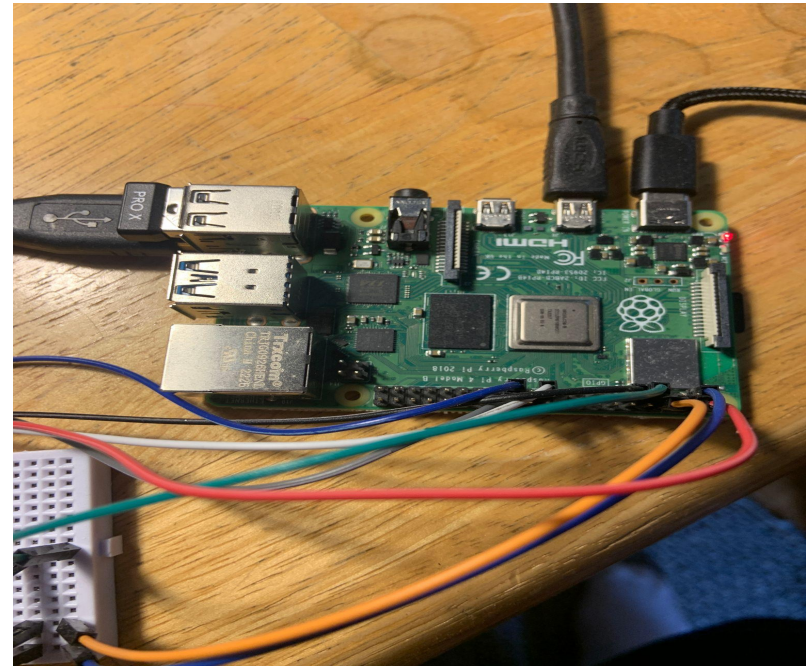
# Materials



- Microcontroller (Raspberry Pi4b)
- Jumper wires
- DS18B20 Temperature Sensor
- Kasa Smart Plug
- Mini breadboard
- MAX7219 LED Display

# Process

- 1.) Research
- 2.) Circuit Design
- 3.) Select Materials
- 4.) Hardware Assembly
- 5.) Software Development
- 6.) Testing + Integration
- 7.) Reflection & Improvement



```
# Loop: Read Temp, Display on LED, and Control Smart Plug
while True:
    celsius, fahrenheit = read_temp()
    if celsius is not None:
        # Prepare the temperature display
        temp_display = f"{celsius:.1f}°C"
        print(f"Temperature: {celsius:.2f}°C / {fahrenheit:.2f}°F")

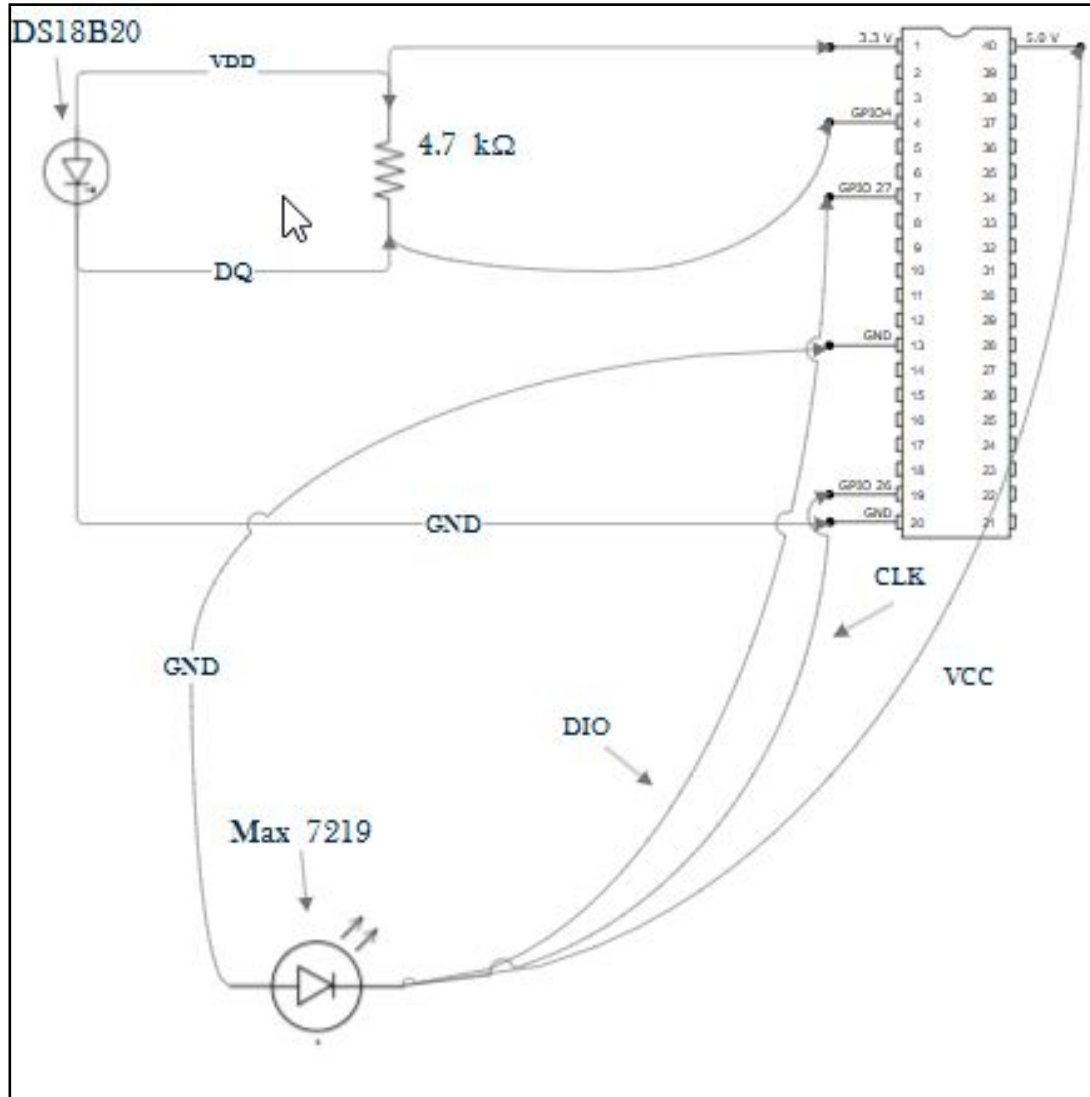
        # Update the MAX7219 LED display
        with canvas(device) as draw:
            draw.text((0, 0), temp_display, font=font, fill="white")

        # For instance, if the temperature exceeds 25°C, turn the plu
        otherwise, turn it OFF.
        if celsius > 25:
            plug.turn_on()
        else:
            plug.turn_off()
    else:
        print("Failed to read temperature data.")

    time.sleep(2)
```

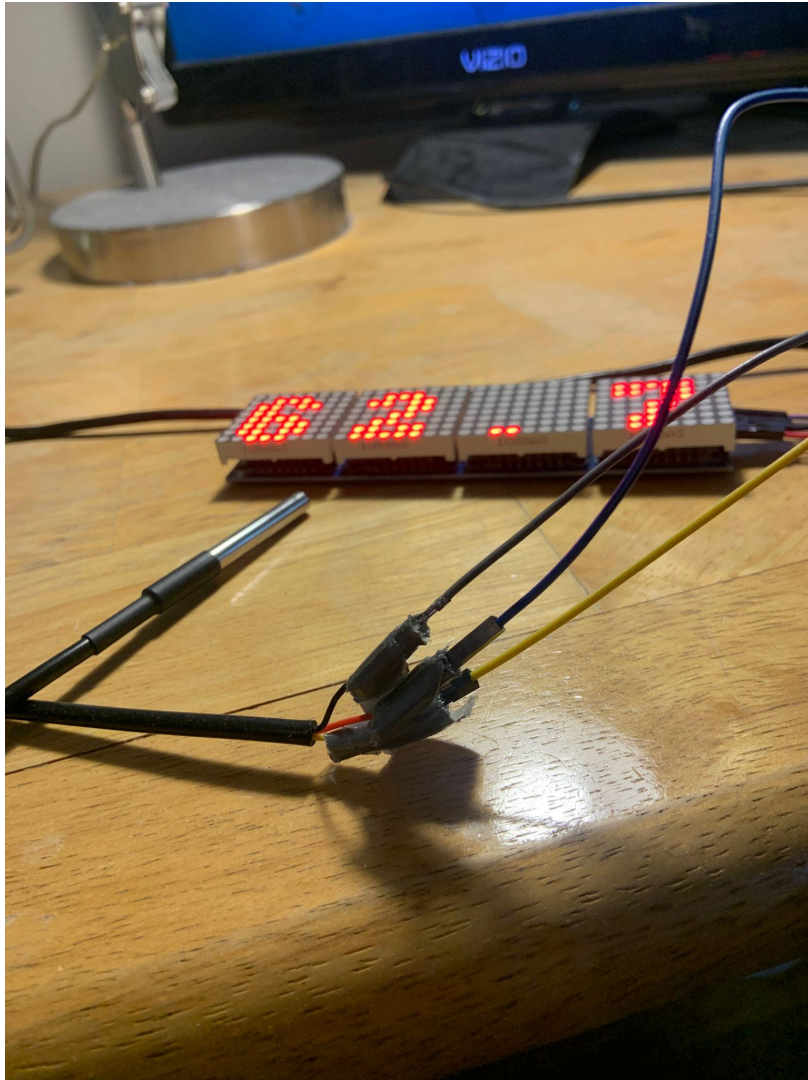


# Circuit



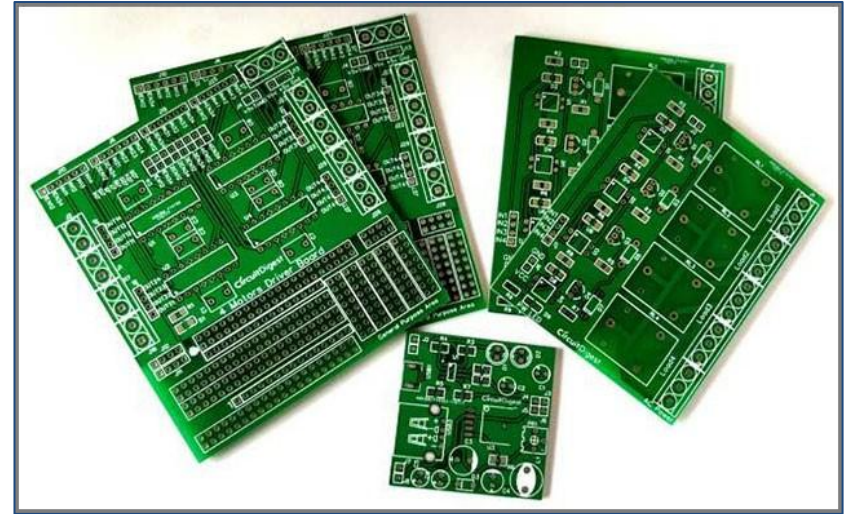
- Can not read data without 4.7k resistor in parallel with data cable and voltage source
- Current flowing through the resistor is approximately 0.7 mA

# Challenges Along the Way



- Connecting DS18B20 wires to jumper wires
- Getting temperatures to display correctly on LED display
- Troubleshooting hardware compatibility with python
- Debugging code

# Possible Improvements



- Custom PCB or solder wires
- Run the python program on boot (possible switch to Arduino)
- Develop a custom chassi with a CAD software