Individual Sensors & Components

Temperature Sensor - BME280

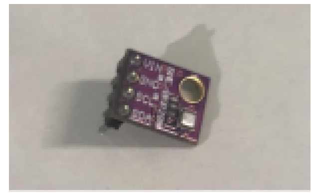

Figure 1: BME280 sensor

Component Name: Adafruit *BME280* Humidity + Barometric Pressure + Temperature Sensor

Requirements

The success of the BME280's ability to measure will be in whether it can sense the temperature to the nearest 5℉ after every 3 minutes of the stove or oven being on with 8 repetitions or the oven reaching the target temperature. We do not need absolute precision, but rather to sense a change in the temperature when the stove/oven is on. Hence, the BME280 should be able to sense at least a 5℉ difference from when the appliance is off or on at a minimum of one location. Each location will have its own fail or pass. The purpose of this study is to determine the optimal location for the BME280 and the optimal minimum temperature for it to detect as unsafe because users will be instructed to use the optimal location and the device should be able to accurately detect temperatures that are unsafe.

Independent and Dependent Variables

The independent variables will be the orientation in relation to the stove/oven, and which appliance (oven or stove) is on. The dependent variable will be the temperature outputted by the BME280. Not only does this testing confirm the sensor works, but this testing also finds the optimal location to place the BME280.

Hypothesis

The BME280 should be able to sense the real temperature to the nearest degree. The BME will also fare better at sensing changes in temperature closer to the appliance, although this capability would be nearly the same within perhaps 2 feet of the appliance. The best orientation of the device may also be above the stove/appliance.

Methods

Materials:

- Stove
- Oven
- Temperature Sensor - BME280
- Arduino Uno
- Wires (to connect Arduino to Sensor)

Building Instructions:

1. Connect 5V on the Arduino Uno to VIN on the BME280 sensor.
2. Connect GND on the Arduino Uno to GND on the BME280 sensor.
3. Connect A5 on the Arduino Uno to SCL on the BME280 sensor.
4. Connect A4 on the Arduino Uno to SDA on the BME280 sensor.

Note: Does not require Breadboard

Testing Instructions:

1. Use the Kitchen Kare BME280 Testing Code. Connect the device to a computer to run the code. Temperature information shows up on the Serial Monitor.
2. Place the BME280 sensor beside the stove.
3. Take the starting temperature before any appliance is turned on.
4. Turn on one burner.
5. Take the temperature every 3 minutes until the temperature stops increasing.
6. Wait until the area around the stove and oven cools down.
7. Place the sensor above the stove.
8. Repeat steps 5 to 8.
9. Set the oven to 500℉.
10. Repeat steps 5 to 7.

Testing

Test Scenario 1: Placing BME280 **Beside** Stove with **1 Stove Burner On**

Purpose: Checking effectiveness of the BME280 sensing change in temperature from beside the stove.

Location and Stove/Oven Information	Beside Stove, 1 Burner On							
Time (min.)	0	3	6	9	12	15	18	21
BME Temperature (℉)	74.53	74.73	75.52	75.78	76.51	76.78	77.14	77.41

Table 1: The BME280 sensor was placed on the counter beside the stove. The temperature does not vary much when the sensor is placed beside the stove.

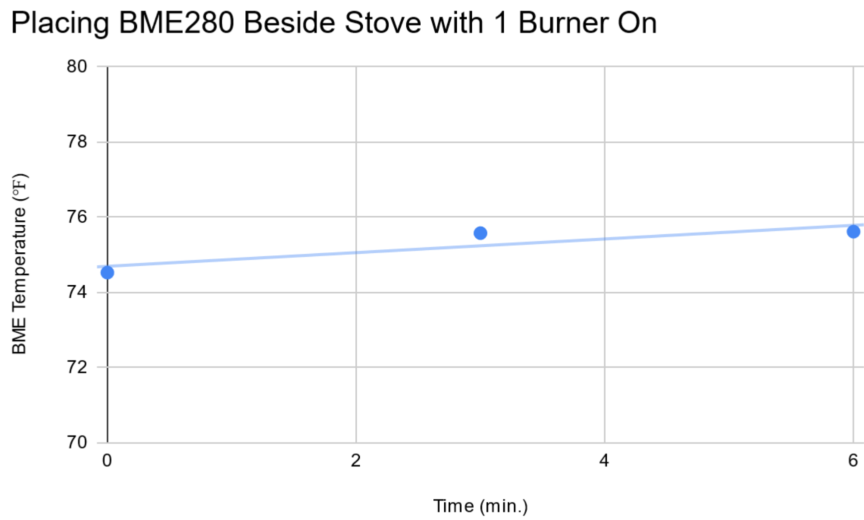

Figure 2: The temperature increased slightly by 1-2℉ when the BME280 was placed beside the stove with 1 burner on.

Test Scenario 2: BME280 **Above** Stove with **1 Stove Burner On**

Purpose: Checking the effectiveness of the BME280 sensing change in temperature from above the stove.

Location and Stove/Oven Information	1 m. Above Stove, 1 Burner On							
Time (min.)	0	3	6	9	12	15	18	21
BME Temperature (℉)	75.52	76.51	77.14	77.41	78.42	79.63	81.68	82.24

Table 2: The BME280 sensor was placed 2 feet above the stove, and one burner was turned on. Once the temperature reached 82℉ about 18 minutes after the burner was turned on, the area surrounding the stove remained at that temperature.

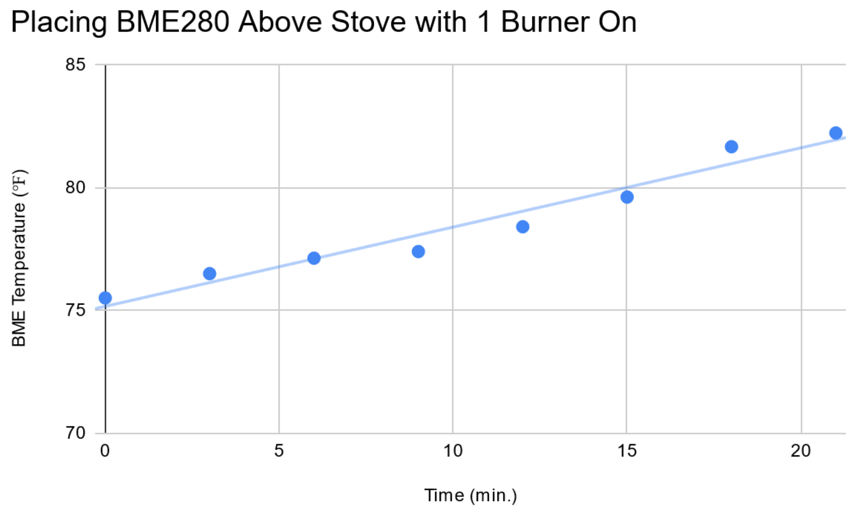

Figure 3: The temperature increased by about 7℉ when the BME280 was placed above the stove with 1 burner on.

Test Scenario 3: Placing BME280 **Above** Stove with **Oven** at 500℉

Purpose: Checking the effectiveness of the BME280 sensing change in temperature from above the stove when the oven is turned on at 500℉.

Location and Stove/Oven Information	1 m. Above Stove, Oven Preheated to 500℉				
Oven Temperature (℉)	200	235	450	500	500
Time (min.)	0	3	7	10	30
BME Temperature (℉)	86.05	90.48	121.42	169.76	179.5

Table 3: The BME280 sensor was placed 2 feet above the stove, and the oven was turned on to 500℉. The initial starting temperature was 86℉. Once the temperature was at 500℉ for 20 minutes, the area surrounding the stove was 179.5℉ and the temperature stopped climbing.

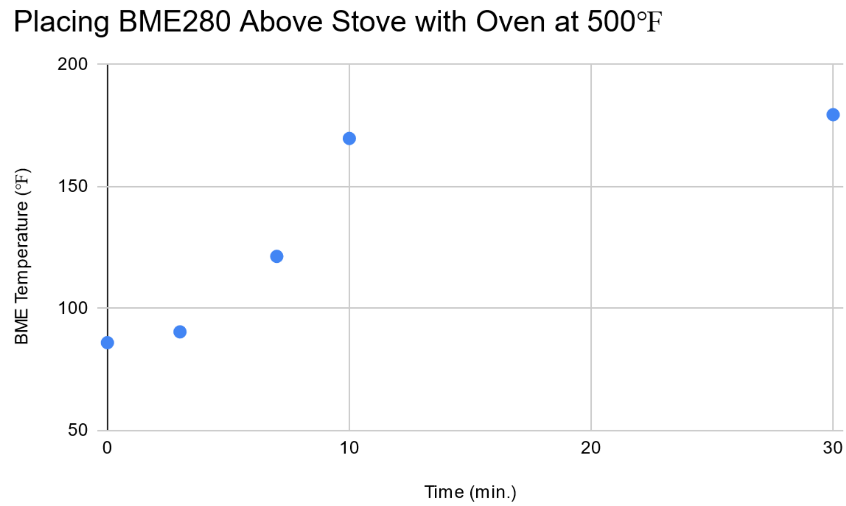

Figure 4: The temperature increased by about 100℉ when the BME280 was placed above the oven while it was preheating. After that, the increasing temperature rate slowed down.

Analysis

Through testing, it was determined that the **optimal location** for the BME280 sensor would be **above** the stove. As seen in Scenario 1, there was a very small temperature change during the time that the burner was on. In comparison, in Scenario 2, the temperature increased by about 7℉ while the temperature only increased by about 1℉ in Scenario 1. The larger increase in temperature indicates that the BME280 sensor would better detect the temperature of the stove or oven when it is located above the stove. Additionally, in Scenario 3, the BME280 sensor is placed above the stove while the oven is on and shows an increase in temperature of about 100℉. Through the testing of the BME280 sensor in Scenarios 2 and 3, the optimal location (above the stove) was determined. Additionally, an approximate optimal temperature that the sensor needs to reach to send the alert was found. The maximum temperature that the sensor reached when one burner was on was around 80℉, so the approximate **optimal temperature** was also determined to be about **80℉.**

Conclusion

After comparing the 3 test scenarios, it was found that **Test Scenarios 2 and 3** gave the best location for the Temperature Sensor (BME280) to properly sense the change in temperature (above the stove with the oven or stove on). This range exceeded the 5℉ benchmark initially set in the requirements and confirms the functionality of this component. The testing also showed that the sensor should alert the user that the stove or oven is on when it reaches about 80℉.

Digital Screen - OLED I2C SSD1306

Component Name: OLED I2C SSD1306 display module

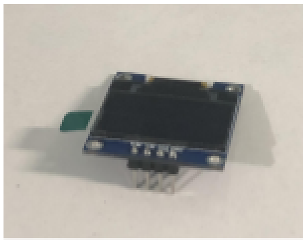

Figure 5: OLED

Requirements

This device must be able to push a message to the OLED and be able to refresh that message periodically so that the user can easily find out what temperature the device perceives. The purpose of this study is to determine if the OLED is capable of refreshing messages rapidly because this functionality is needed for the final device (to display the temperature detected by the BME280).

Independent and Dependent Variables

The independent variables will be the frequency of the messages that are pushed to the OLED sensor. The dependent variable will be whether the OLED sensor is able to effectively portray that message.

Hypothesis

The OLED may need to have an initial grace period to start up, so it may miss a few initial changes, but it should eventually catch up.

Methods

Materials:

- Digital Screen - OLED screen
- Wires
  - Two small blue wires
  - 1 long blue and 1 long yellow wire
  - 1 long red and 1 long black wire
- Breadboard (optional, but required for this set-up)

Building Instructions:

1. Place the 4 pins of this Screen in g24-27, with the screen above the pins directly into the breadboard.
2. On the bottom side of the Arduino, take a blue male-to-male wire and connect it from A4 on the Arduino to i27.
3. On the bottom side of the Arduino, take a yellow male-to-male wire and connect it from A5 on the Arduino to j26.
4. Connect two smaller wires from i25 to the positive (+) row on the bottom of the breadboard and i24 to the negative (-) row on the bottom of the breadboard.

5. Connect the black wire from GND or ground from the bottom of the Arduino to the negative (-) row
6. Connect the red wire from 3.3V from the bottom of the Arduino to the positive (+) row

Testing Instructions:

1. Use the Kitchen Kare OLED Testing Code. Connect the device to a computer to run the code.
2. At first, the message on the screen will change due to a variable screenDuration, representing the frequency of the messages displayed (independent variable)
    a. This will also change the message after this defined period so it can easily be observed
3. For the values of 20 seconds, 10 seconds, 5 seconds, 2 seconds, and 1 second, run the program for 6 repetitions and observe how many times the message changes. If all 6 repetitions work, it can be established that the OLED is functional, and can periodically update the screen.

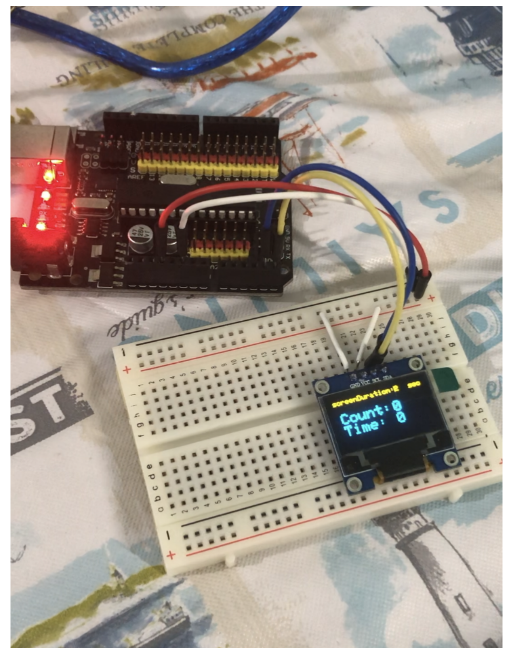

Figure 6: OLED Testing with Arduino

Testing

Test Scenario: Testing **OLED** Changing Message Rate

Purpose: Checking whether the OLED is capable of changing the message displayed constantly.

Screen Duration (s)	20	10	5	2	1
Successes/Total Repetitions	6/6	6/6	6/6	6/6	6/6
Success?	Yes	Yes	Yes	Yes	Yes

Table 4: The OLED was successful in changing the screen every second (and at slower rates).

Analysis

This testing concluded that the sensor is able to quickly **change** between screens. The OLED was tested several times during decreasing intervals. The intervals began at 20 seconds then went to 10 then 5 then 2 and then 1. In each of these intervals the sensor was tested 6 times and worked each of the times. Since the sensor was able to quickly change between screens, it will be able to show different temperatures that the BME280 sensor will detect.

Conclusion

This test shows that the OLED will be able to accurately **display** to the user the changing temperatures detected by the BME280 sensor. As the temperature changes, the OLED will be able to display the different temperatures to alert the user of any change and possibility that the stove or oven was left on.

Human Sensor - Sparkfun Infrared Sensor

Component Name: Sparkfun Grid-Eye Infrared Sensor

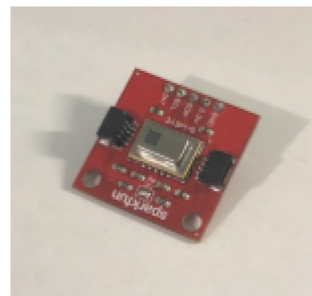

Figure 7: Sparkfun Infrared Sensor

Requirements

The success of the device will be measured exclusively on its capability to send a message, indicating it successfully sensed a person. The purpose of this study is to determine whether the sensor can accurately detect a person's presence because it will tell if the sensor is reliable to use to detect motion in the kitchen.

Independent and Dependent Variables

The independent variables will be the distance and speed of a human being sensed by the device. The dependent variable will be whether a message is pushed to the console screen on the computer when a human is present. Although in practice the device will wait half an hour before turning the appliance off, it is more efficient to reduce this delay for instantaneous testing. For each configuration of independent variables, there will be 4 trials to build a holistic analysis of the device under those conditions.

Hypothesis

The device will have higher accuracy at sensing people from shorter distances, about less than 1 or 2 meters. The device should be able to handle most frequencies of human motion and may grow increasingly inaccurate as the human moves at higher speeds.

Methods

Materials:

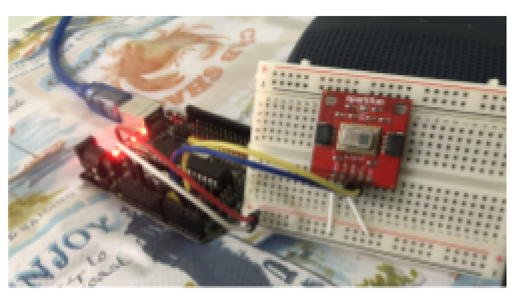

Figure 8: Sparkfun Infrared Sensor testing with Arduino

- Person
- Infrared Sensor
- Arduino Uno
- Wires
    - Two small white wires
    - 1 long blue and 1 long yellow wire
    - 1 long red and 1 long white wire
- Ruler or tape measurer
- Timer
- Breadboard (optional, but required for this set-up)

Building Instructions:

1. Build the following device:
    a. Place the 5 pins of this Sensor in f16-20, with the Sensor above the pins directly into the breadboard.
    b. On the bottom side of the Arduino, take a blue male-to-male wire and connect it from A4 on the Arduino to g18.
    c. On the bottom side of the Arduino, take a yellow male-to-male wire and connect it from A5 on the Arduino to g17.
    d. Connect two smaller wires from g19 to the positive (+) row on the bottom of the breadboard and g20 to the negative (-) row on the bottom of the breadboard.
    e. Connect the white wire from GND or ground from the bottom of the Arduino to the negative (-) row
    f. Connect the red wire from 3.3V from the bottom of the Arduino to the positive (+) row
2. Use the Kitchen Kare Sparkfun Infrared Testing Code. Connect the device to a computer to run the code. Information shows up on the Serial Monitor.
3. Walk in front of the sensor at different speeds (0.5, 1, 1.5, and 2 m/s) each three times at varying distances (1, 1.5, 2 m) and record whether the sensor detects human presence (from Arduino Serial Monitor).

Testing

Test Scenario: Testing Sparkfun Infrared Rate of Sensing Human Presence

Purpose: Checking whether the Sparkfun Infrared sensor is capable of detecting varying human motion at varying distances from it.

Speed (m/s)	0.5			1			1.5		
Distance (m)	Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3
1	Y	Y	Y	Y	Y	Y	Y	Y	Y

1.5	Y	Y	Y	Y	N	Y	Y	Y	N
2	Y	N	Y	N	N	Y	N	N	N

Table 5: The Sparkfun Infrared was able to detect human motion at many distances and speeds. However, it struggled to detect human motion when the human was moving at 1.5 m/s and when the human was 2 m away from the sensor.

Analysis

The majority of the testing of the infrared sensor shows that the sensor is able to detect people from a distance of **2 meters**, moving at speeds up to **1 m/s**. From speeds .5 to 1 m/s, the sensor was able to detect people with some slight contradictions. At .5 m/s at a distance of 2 m, the sensor detected the human in 2 out of the 3 trials. Since the sensor detected the human a majority of the time, it can be concluded that when a person is moving at .5m/s a distance of 2 meters from the sensor, they will be detected. Similarly, when the human was moving at 1 m/s a distance of 1.5 meters, the sensor detected them 2 out of the 3 trials. Since the sensor detected the human a majority of the time, it can be concluded that the sensor will detect a human moving at 1 m/s a distance of 1.5 meters. Similarly, when the person was 2 meters from the sensor, they were detected 1 out of 3 of the time. Since the human was detected, the sensor should be able to detect humans moving at a speed of 1 m/s a distance of 2 meters from the sensor. There might be some slight errors, but the human should be detected.

At a **speed of 1.5 m/s**, a human should be detected when they are **1 or 1.5 meters** from the sensor. In contrast, when the human is 2 meters from the sensor, they will most likely not be detected. Out of the 3 trials completed during the test, the human was detected 0 out of the 3 trials, indicating that the sensor would not be able to detect humans moving at a speed of 1.5 m/s a distance of 2 meters from the sensor.

Conclusion

The sensor will be able to detect humans that are 2 meters from the sensor moving at speeds up to 1.5 m/s. The sensor will also be able to detect humans that are at most 1.5 meters from the sensor moving at speeds up to 1.5 m/s. This shows that the infrared sensor will be able to detect if someone is in the kitchen and nearby the stove or oven.

Controllable Outlet - 4 Outlet Wall Plug

Component Name: Controllable Four Outlet Power Relay Module version 2 - (Power Switch Tail Alternative)

Requirements

The Arduino should be capable of controlling power to at least half of the devices tested for controllable durations. The purpose of this study is to determine whether the wall plug is capable of controlling power to appliances, particularly kitchen appliances, which will tell if the sensor is useful to include in our prototype.

Independent and Dependent Variables

The independent variable will be the kitchen appliance used for testing. The dependent variable will be whether the device can turn the appliance on and off every 5 seconds. This specific wall plug cannot support the needs of ovens or stoves, but rather proves as a proof-of-concept that Arduino can control power to kitchen appliances.

Hypothesis

The device will work with most, if not all, of the kitchen appliances tested with this methodology. It will be capable of interpreting and utilizing a signal to turn the device off or on.

Methods

Materials:

- Arduino Uno
- Wall Plug
- Wires
    - 1 long black wire
    - 1 long yellow wire
- Breadboard (optional, but required for this set-up)

Building Instructions:

1. Connect 1 long yellow and 1 long black wire from pin 1 to the right hole of the green connector and ground (GND) to its left hole.
2. In order to connect the wires to the green connector portion of the wall plug, unscrew the tops then clamp them when inserted.

Testing Instructions:

1. Use the Kitchen Kare 4 Outlet Wall Plug Testing Code. Connect the device to a computer to run the code.
2. Run the program and record if the wall plug can turn off and on the power flow for each of the following 5 times in a row.
3. Apply to the varying appliances: 2 toasters, rice cooker, computer, and lamp.

Testing

Test Scenario: Testing If Wall Plug can Turn Varying Appliances On and Off

Purpose: Checking whether the 4 Outlet Wall Plug can turn varying appliances on and off.

Appliance Name	Toaster 1	Toaster 2	Rice Cooker	Computer proof of concept	Lamp
Successes/Total Repetitions	5/5	5/5	5/5	5/5	5/5
Success?	Yes	Yes	Yes	Yes	Yes

Table 6: The wall plug was able to successfully turn the varying devices on and off for each repetition.

Analysis

This testing concluded that the wall plug is able to successfully **power and discharge** kitchen appliances. The Wall Plug initially turns off the appliance when the Arduino is starting up but soon starts implementing the code. Every 5 seconds, the Wall Plug turns the appliance on and off. After 5 successful repetitions, the next appliance was examined. All appliances worked successfully without any errors.

Conclusion

The wall plug can properly give and restrict power to the appliances for given durations of time. This experiment also shows it works for a large range of smaller voltage appliances and is able to restart the appliance even after it has been turned off. For more practical applications of the device, the 5 seconds can be changed to a customizable cool down in future applications.