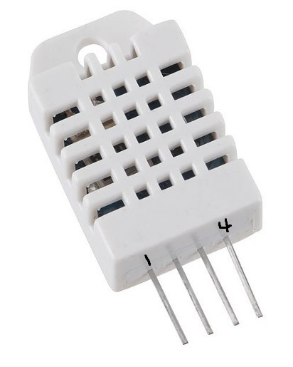
**Methods and Testing Procedures**

**DHT22 Temperature and Humidity Sensor**

The DHT22 sensor contains both a thermistor and humidity sensor to provide both temperature and humidity readings within a single data line. The sensor outputs a digital signal; therefore it can be sent straight to any digital I/O pins. To test this sensor, the sample Arduino code provided by Adafruit Industries was used to output temperature and humidity readings onto the serial monitor. This was done with the following pin connections:

* Pin 1 to 5V on Mini
* Pin 2 to 7 on Mini
* 10kΩ Resistor from Pin 2 to Pin 1
* Pin 4 to GND on Mini

Pin 3 was ignored, as it is classified as a NULL pin on the datasheet. The example code was then uploaded to the Mega Mini. This code used the provided header file from Adafruit, which automatically calculates temperature into Celsius and humidity on a percentage scale. The code can be found in Figure 1A of the Appendix.

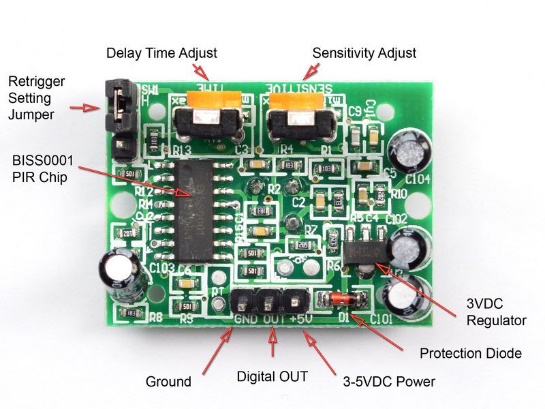
When executed, the serial monitor provided the readings from the sensor. This output can be found in Figure 1B. To test the validity of these reading, a home temperature system was used as a comparison. These values were close within a few degrees Celsius. General conclusions were also made given that the sensor showed average room temperature readings. The sensor was confirmed to be functional and was imported to the prototype design.

**PIR Sensor**

The Passive Infrared Sensor tracks motion from up to seven meters. The lenses found around the sensor help amplify the range of the sensor to reach this distance. To test this sensor, a simple program was written to illuminate the on-board LED whenever motion was detected. This code can be found in Figure 2A. This was done with the following pin connections

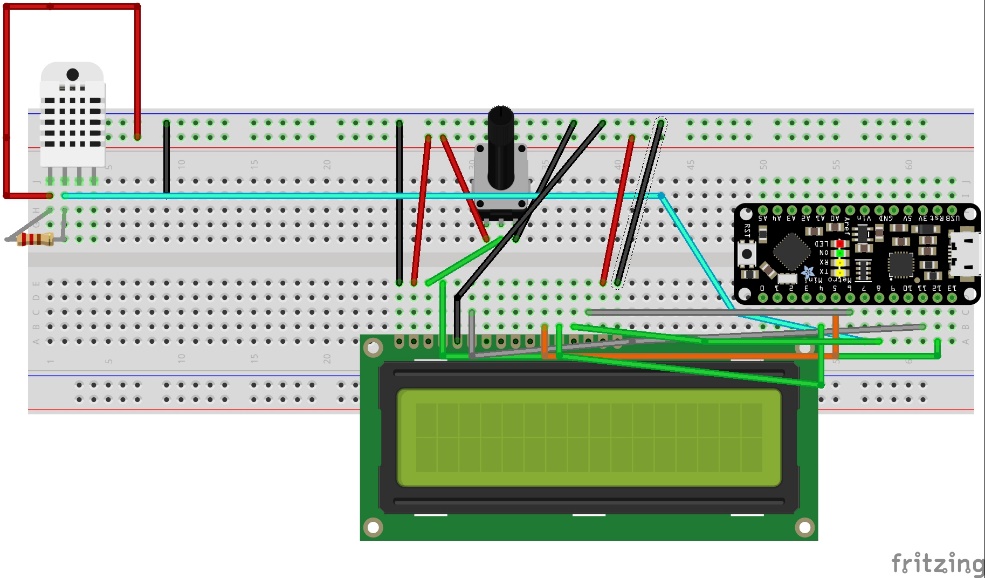
* +5V to 5V on Mini
* OUT to 10 on Mini
* GND to GND on Mini

Before uploading the code onto the Mini, the on-board potentiometers of the PIR sensor were adjusted to provide a minimum sensitivity and a five second output delay. This delay would keep the output high for that duration. Once the code was executed and the sensor was given time to calibrate, motion was provided in its path. The circuit responded by illuminating the on-board LED for about five seconds. After this duration, the LED was driven low. In addition to this test, a simple counter was implemented to increment a set value whenever the PIR turned high. This test was executed, but the outcome was not desired. The counter would continuously increment on the serial monitor for the given duration delay. To adjust this outcome, the previous state and current state of the output was stored and compared to ensure that the counter variable would only increase once the output drove low. The sensor was confirmed to be functional and was imported to the prototype design.



**LCD**

The 16x2 character LCD provides an alphanumeric display based on a specified output. To test this component, the DHT procedure was modified to print out on the LCD rather than the serial monitor. Using the Liquid Crystal header file, the LCD was initialized and was sent temperature and humidity readings. This code can be found in Figure 3A. The following pin connections were made to utilize the LCD.

* VCC to 5V on Mini
* VDD to GND
* VO to potentiometer wiper (10kΩ)
* RS to Pin 12
* RW to GND
* E to Pin 11
* D4 to Pin 5
* D5 to Pin 4
* D6 to Pin 8
* D7 to Pin 6
* A to 5V
* K to GND
* (See DHT procedure for pinout)

When the code was executed, the LED showed null values for the readings. This is because the DHT22 requires two seconds for a valid data output. Once two seconds passed, the temperature and humidity was displayed on the screen. These values were equivalent to the previous ones shown on the serial monitor. The LCD was confirmed to be functional and was imported to the prototype design.

**Push Button**

The final component to evaluate was a push button. When the button is pressed, the internal switch is turned on, allowing current to flow through the terminals. To test this simple component, a button counter program was written. The program would count how many times the button was pressed. This value was printed on the serial monitor. The following pin connections were made:

When the code was first executed, the counter variable showed a similar reaction found in the PIR test. The variable was incrementing more times than the button was pressed. This bouncing issue was also resolved through comparing the previous and current states. When the button state was low (signifying the button was pressed), the button counter was changed and the previous state was saved. A fifty millisecond delay was applied to ensure that the device had sufficient time to debounce. Therefore, the counter would only increment once every time the button was triggered.

**Climate Control and Room Occupancy Prototype**

Once all individual components were tested, they were applied to the prototype board. This design contained the same pinouts detailed in the previous procedures. The code from these procedures was organized and configured for the prototype build. The last two components that was included were 12VDC case fans. Since the relay was inoperable, the power pin was fed to an I/O pin on the mini. The analog pins were used since the other digital pins were cluttered with jumper cables. These fans will be fed through the digital I/O pins for the final design. A simple function was written to control each fan based on a comparison between the current and set temperatures. The fans were oriented in a push-pull configuration. When the set temperature was less than the current temperature, the push fan was activated. When the set temperature was greater, the pull fan was activated. The final code for the prototype can be found in Figure 4.

The prototype system was given a set of user inputs to test its response. Button response was immediate whenever a set temperature was configured. The case fans responded accordingly to the input data. There was no visible time lag between a change in temperature and the fan response. The PIR sensor also responded well in the system. There seemed to be a slight delay between motion and the output signal. This was expected, as many motion controlled lighting systems do indeed contain a small delay before turning the light on. A chart of the various testing procedures can be found below.

|  |  |
| --- | --- |
| **System Inputs** | **System Response** |
| Set temp exceeds current temp | A0 goes high and A1 goes low. Pull fan turns on |
| Temperatures are equal | A0 goes low, pull fan turns off. |
| Current temp exceeds set temp | A1 goes high and A0 goes low. Pull fan turns off, push fan turns on |
| Both input buttons are pressed | Set temp stays the same |
| Hand waved near PIR | Light turns on, off after 5s |
| Hand waved and input button is pressed | Light turns on and set temperature is changed simultaneously |