APHRODITE SMART MIRROR PROJECT

By: Jose Antonio Castro Teodoro

A Kainotomia Company Project

Table of Contents

Abstract Of What We Are About

The Aphrodite project will be a smart mirror. With the use of an android application, Arduino uno, raspberry pi and a firebase database. The user will be able to use this smart mirror for everyday use and has a variety of features. Having voice recognition, sensing the humidity, temperature of the washroom, having different kinds of personal mirror display layouts for everyone and having a built-in proximity sensor to have contactless activatability with a user. Going further into how these personal user layouts work is that this will allow users to have different custom widgets on the mirror for each individual person. In addition to the proximity sensor, my module will work as proposed. The main function of the proximity sensor is that the mirror will turn on and off automatically when it senses any thermal or light readings because the sensor used is exactly the PIR Sensor. Which will be further discussed later into the report.

CREDIT:

Alyssa Gomez n01042777

Daniel Moore n01354875

Ryan Black n01305403

Jose Antonio Castro Teodoro n01384776

Introduction To the Project

How the software App will work with each other into the hardware project or specifically how it will work with the Arduino uno is the use of a firebase database. The phone App currently has a firebase to collect the layout data of a user and transfer that data into the Arduino uno. Reading those values as Boolean values to determine what widgets are wanted from the user or not. For example, if someone wants to only see the time and weather. Those widgets will be true in the Arduino uno and the rest of the widgets will be read as false, therefore only allowing the time and weather to be shown in the display.

Project Proposal

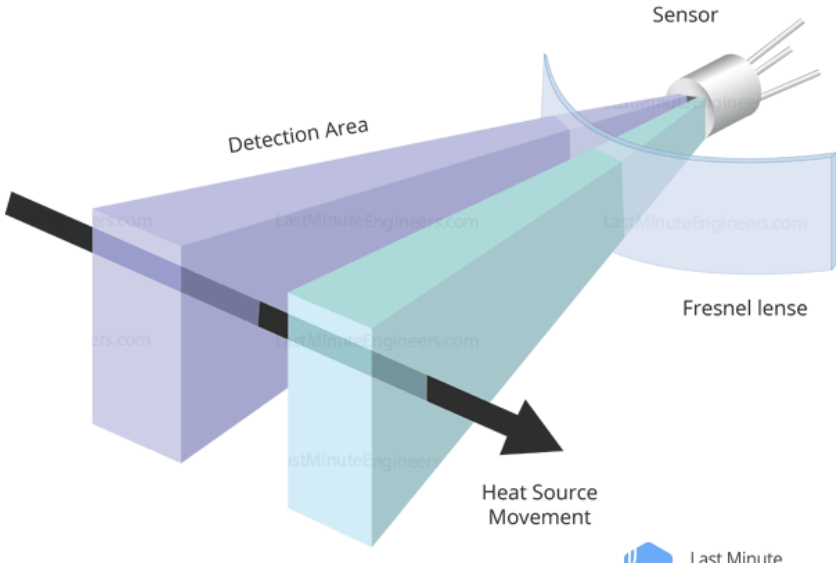
The world is changing every single day. Innovating with the use of computers and wireless connectivity. Only a few years ago the smart phone was introduced by Apple as the iPhone, the idea of having a computer in such a small package that it can fit in your pocket. Now here in the project made our company *Kainotomia,* we are here to introduce the Aphrodite project. A smart mirror that allows people to see things on your mirror like time, weather, emails, and any other widgets that the user desires. Smart homes are becoming more common these days in the 21st century with computer home assistants like the amazon Alexa, automatic doors, and security cameras. Now here is our contribution to the smart home market. Users will also have a phone app to interact with the smart mirror and make different display layouts for each person. Let's give an example in a couple standpoint. A couple lives together but likes to see different things in the morning when they wake up, the boyfriend like looking at the time, news, and stocks in the morning to keep himself updated. But the girlfriend only wants to see the time, weather, and social media. With the app, this will give users the ability to switch for layout to layout. So, the girlfriend and the boyfriend can see their own things with no problems, meaning less arguments with each other. And with this idea, we hope that the Aphrodite project will help the world making people's lives easier.

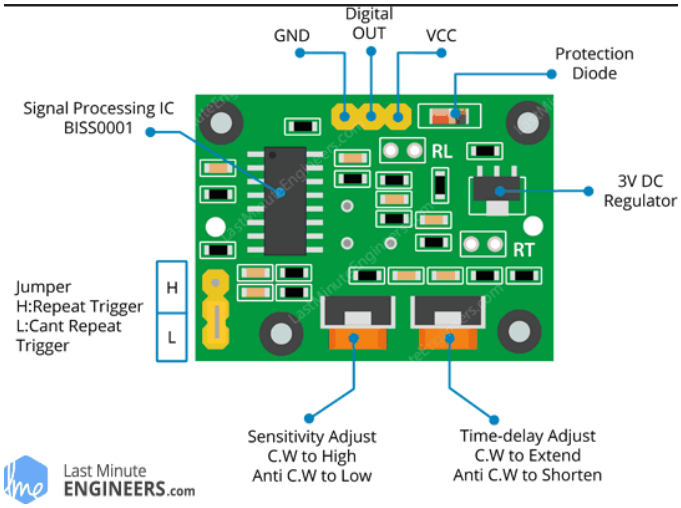
CPU board & Sensor Modules

The CPU board chosen was the Arduino uno, but it will be interacting with a raspberry pi for wireless connectivity. The sensors chosen was the HR-SR501 PIR Sensor, a DHT22 Temperature/Humidity sensor, a voice recognition sensor and a LED strip used with a logic converter so that all the modules use the same voltage throughout the system. The PIR sensor is used for contactless activatability so that the mirror can turn on when someone comes on front of the mirror and if there is no one present, the mirror will turn off automatically. The DHT22 sensor, to check the temperature and humidity of the room that the mirror is present in, but has the option to turn on and off with the use of the app. The use of voice recognition so that the user can use google voice recognition to search stuff up, to turn off the mirror, or to turn on/off certain widgets in the display of the mirror. Finally, a customizable LED strip for cosmetic back lighting for the smart mirror.

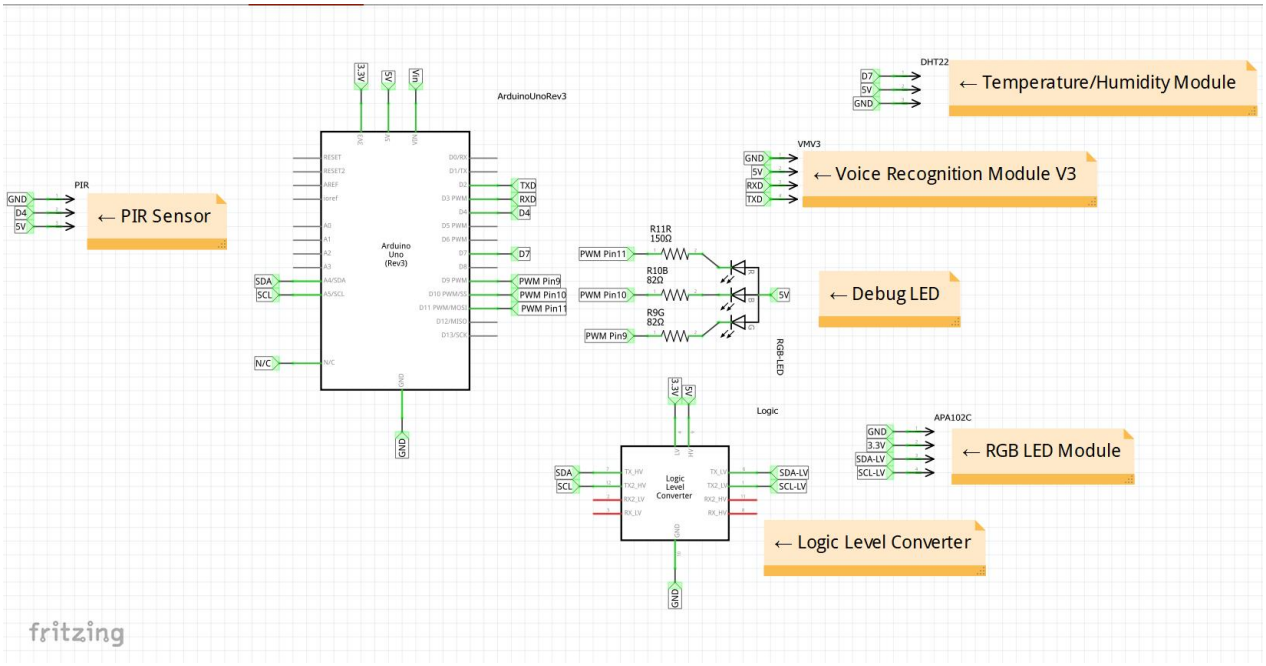
HR-SC501 PIR Sensor

- PIR sensor was designed to detect such levels of infrared radiation. It basically consists of two main parts: A Pyroelectric Sensor and A special lens called Fresnel lens which focuses the infrared signals onto the pyroelectric sensor.
- HC-SR501 PIR sensor has three output pins VCC, Output and Ground, built-in voltage regulator so it can be powered by any DC voltage from 4.5 to 12 volts, typically 5V is used.
- There are two potentiometers on the board to adjust a couple of parameters:
    - Sensitivity– This sets the maximum distance that motion can be detected. It ranges from 3 meters to approximately 7 meters
    - Time– This sets how long that the output will remain HIGH after detection. At minimum it is 3 seconds, at maximum of 5 minutes.
- The diagram below will show what the PIR Sensor has and how it works.

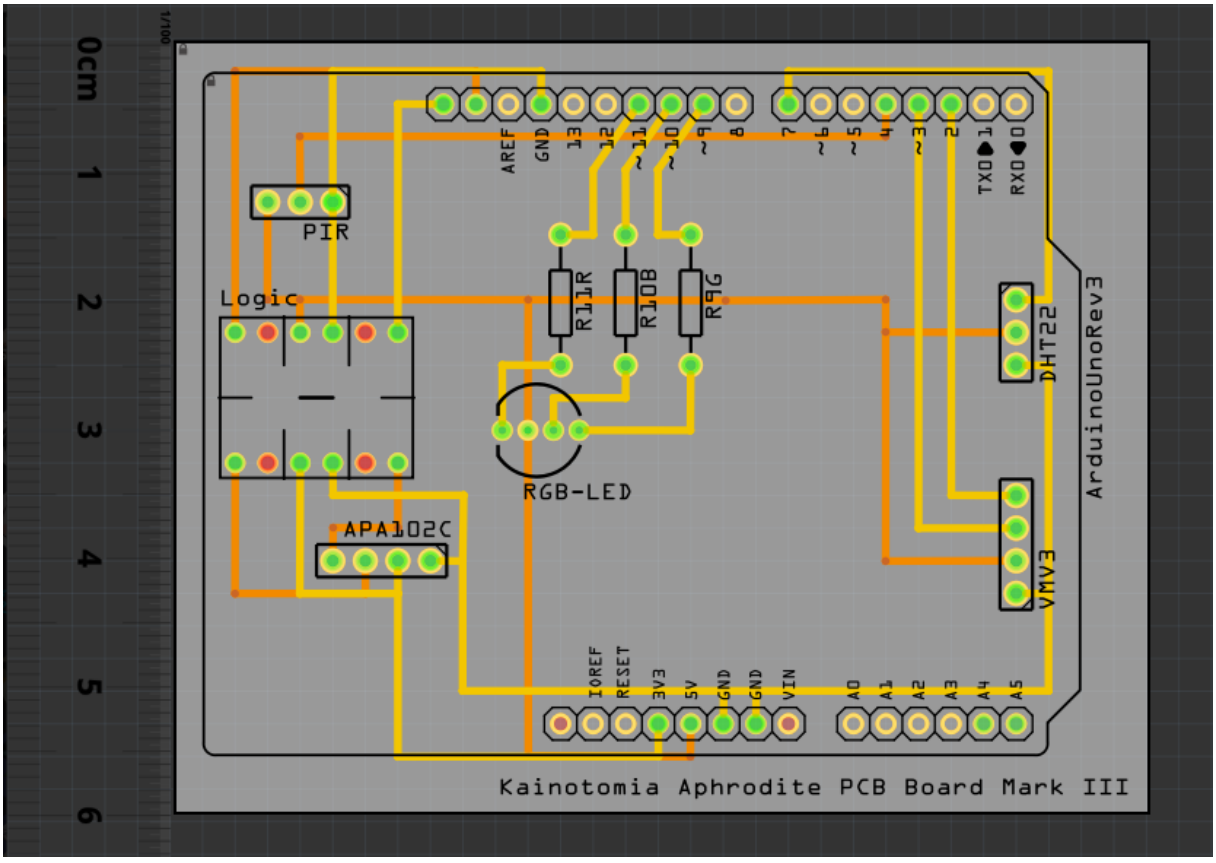




Schematic Design

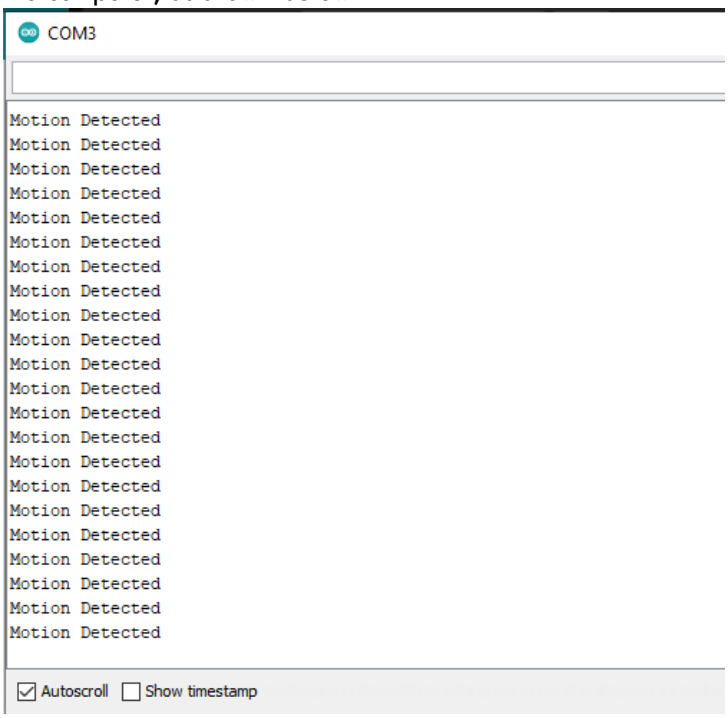


The PCB Design



Testing procedures of PCB

- Testing different module locations on the PCB
- Factoring if the modules will still work properly if they are positioned the way they are.
- Soldered the pin headers onto the PCB
- Found out that there was a power issue with the PCB and Arduino
    - Arduino Uno Controller was turning off when the PCB went on top of the Arduino.
    - The way how we fixed the issue was putting electrical tape on top of the metal D port on the Arduino because it was making contact with the soldered PCB pins from under, causing power issues.
- Before soldering the output LED to the PCB, I had to make sure that the PCB worked properly with the PIR Sensor module.
    - I put the LED diode to the ground and pin 13 so no soldering was needed.
- Testing Arduino Uno with PCB
    - Put the PIR sensor input pin into pin 4
    - Changing the potentiometers to account for the time delay and range sensitivity of the module and figuring out what was the best settings for the PCB.
    - Setting the Jumpers to High, to repeat the trigger of the PIR sensor
    - Coding with Arduino IDE so the sensor can read on the LED diode and the serial monitor on the computer, as shown below.



    - Finally figured out what worked with trial and error and finalized the design and code.
        - Soldered the LED diode when the design was final
        - The code being shown below

TeodoroCENG317PIRS

```
/*
PIR HCSR501
by Jose Antonio Castro Teodoro
*/
int ledPin = 11;                // LED
int pirPin = 4;                 // PIR Out pin
int pirStat = 0;                   // PIR status
void setup() {
 pinMode(ledPin, OUTPUT);
 pinMode(pirPin, INPUT);
 Serial.begin(9600);
}
void loop(){
 pirStat = digitalRead(pirPin);
 if (pirStat == HIGH) {            // if motion detected
   digitalWrite(ledPin, HIGH);  // turn LED ON
   Serial.println("Motion Detected");
 }
 else {
   digitalWrite(ledPin, LOW); // turn LED OFF if we have no motion
 }
}
```

```
/*
PIR HCSR501
by Jose Antonio Castro Teodoro
*/
int ledPin = 11;                // LED
int pirPin = 4;                 // PIR Out pin
int pirStat = 0;                   // PIR status
void setup() {
 pinMode(ledPin, OUTPUT);
 pinMode(pirPin, INPUT);
 Serial.begin(9600);
}
void loop(){
 pirStat = digitalRead(pirPin);
 if (pirStat == HIGH) {            // if motion detected
   digitalWrite(ledPin, HIGH);  // turn LED ON
   Serial.println("Motion Detected");
 }
 else {
   digitalWrite(ledPin, LOW); // turn LED OFF if we have no motion
 }
}
```

Troubleshooting procedures

- Having the wrong pins being coded into the Arduino IDE
- Tweaking and changing the potentiometers of the PIR Sensor because sometimes the readings are delayed with the LED and the serial monitor output.
- Changed some of the spare PIR sensors because some sensors worked better than one another
- Bought various kinds of proximity sensors such as IR sensors and ultrasonic sensors. These are the following sensors bought:
    - An infrared obstacle avoidance sensor will not work with the project because the sensor range was too short. For around 30cm or 12 inches.
    - An HC-SR04 ultrasonic module, did not work because this module was mainly used to measure distances with high precision. With the range of 4 meters.
    - A Seeed Grove 80cm infrared proximity sensor, this sensor did not work because it was not compatible with the current board being used. Which was the Arduino uno. It was only compatible with a grove base shield, which would not work for this project because the custom PCB was already being used as a shield for the Arduino.
- What was Learned During The Process
    - There is a lot of trial-and-error working on anything that involves hardware
        - Trying to figure out if the issue was with the code or the hardware of the project.
    - Learned how to solder properly
    - New to the fritzing program and learned how to make a hardware schematic and PCB from scratch.
    - Accounting for a lot of factors if the PCB will short circuit, not turn on, or if the wiring on the PCB are properly aligned to have no cross wiring.

Work Cited

Work Cited: Last Minute Engineers. (2020, December 18). *How HC-SR501 pir sensor works & how to interface it with Arduino*. Last Minute Engineers. Retrieved December 13, 2021, from https://lastminuteengineers.com/pir-sensor-arduino-tutorial/.