EECS 113

PROCESSOR HARDWARE/SOFTWARE INTERFACES

BeeWare

Diego Torres, 12602118
Jed Marasigan, 15988273

OBJECTIVE

The goal of this project was to design an IOT device that acts to improve upon the household experience. We aimed to display the the humidity, the temperature, time, and date to the user when they are at their house. It also doubles as a security device because of the implemented security feature which allows the user to safely protect their home from intruders. The device is able to detect motion and change in light due to the entrance of the intruder into the users home. Another objective this project was designed for was to introduce us to the concept of embedded systems to help us get an idea of how our senior design projects are going to be.

METHODOLOGY

The project was realized as part of an assignment for EECS 113. This assignment is the final project and its purpose is to show us how to use our understanding of microcontroller architectures to create a device which can interact with the users, sensors, and internet. This type of device is known as an *IOT* device. IOT is a system of computing, mechanical, and digital devices that interact with people, animals, and objects. Data is transferred between the two in different types of mediums over a network. This idea is widely seen as the future of society, and currently there are already implementations of IOT on small scales. The CoastGuard247 is an introduction into this topic. With it we are able to explore the core concepts like data management, control systems, and data processing.

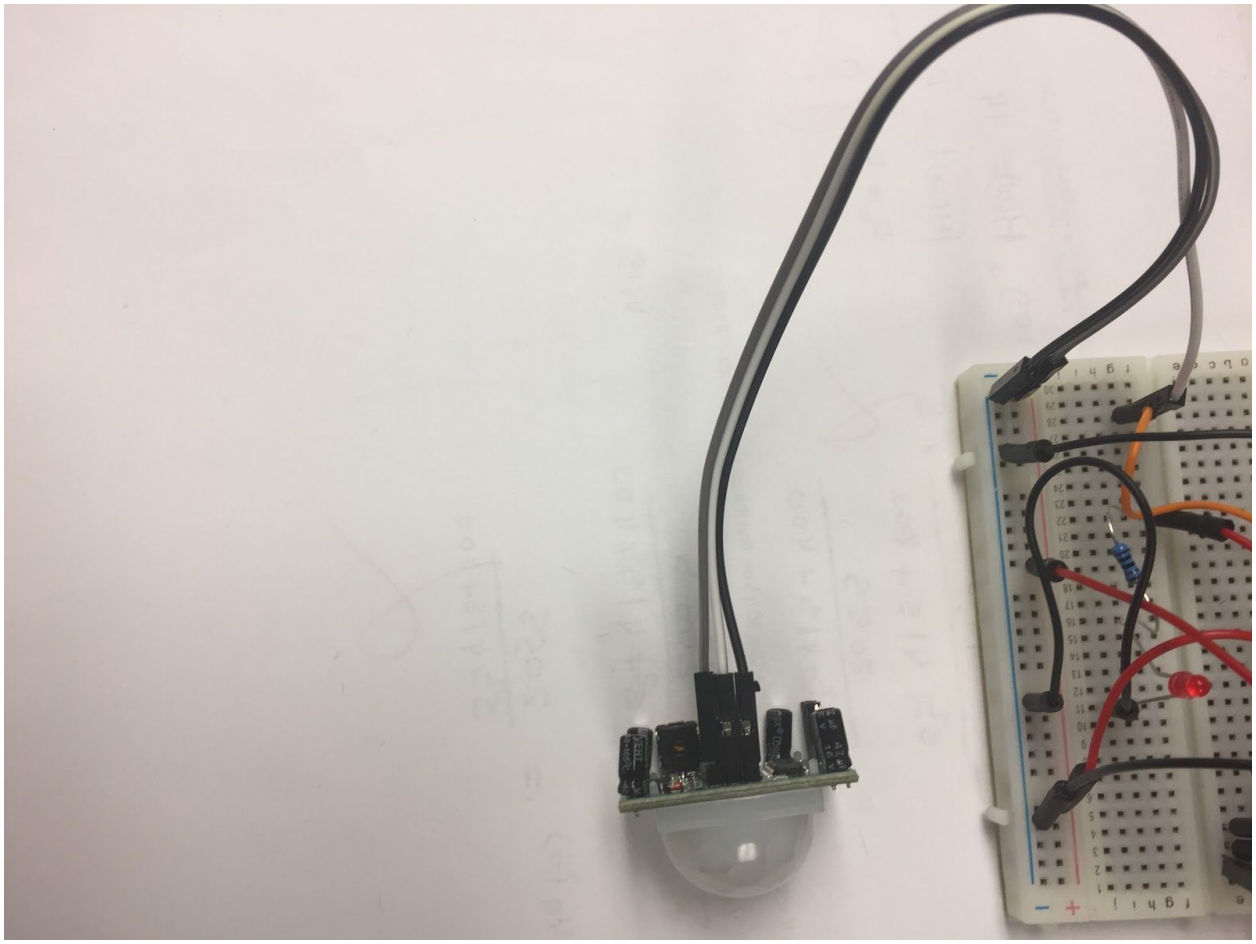

Figure 1: (*Digital Infrared MS*)Motion Sensor Circuit

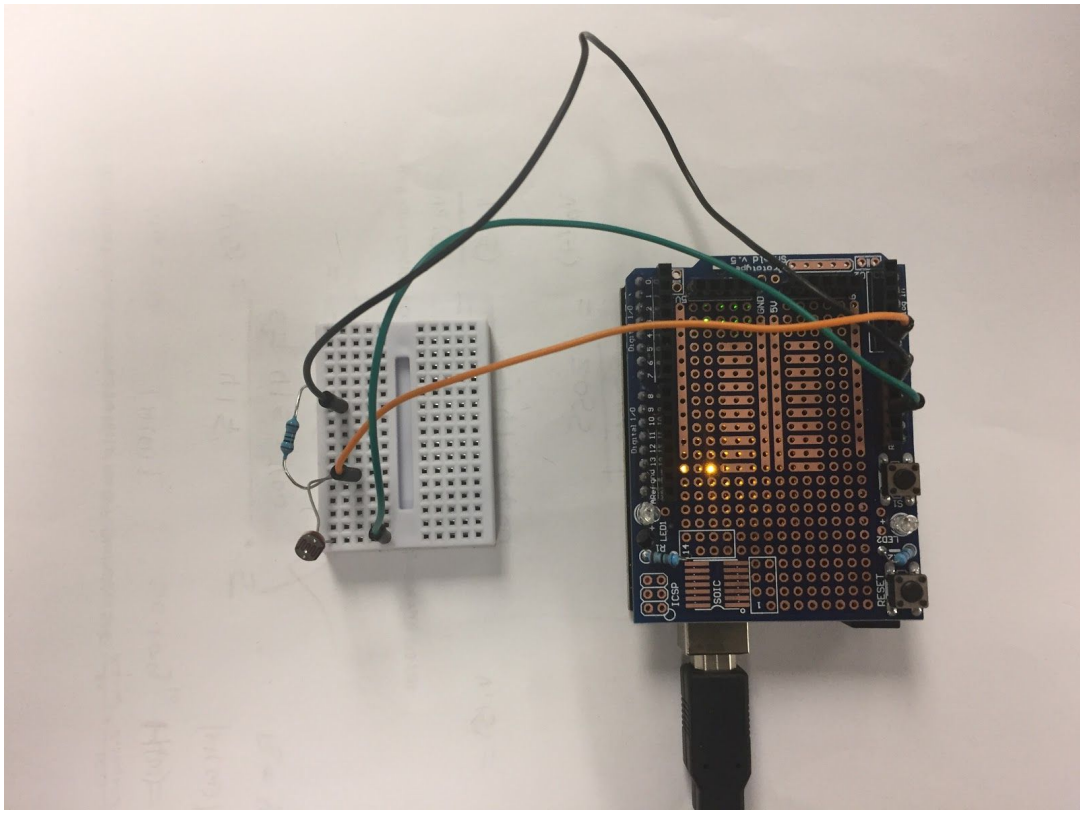

Figure 2: (*LDR*)Light Dependent Resistor Circuit

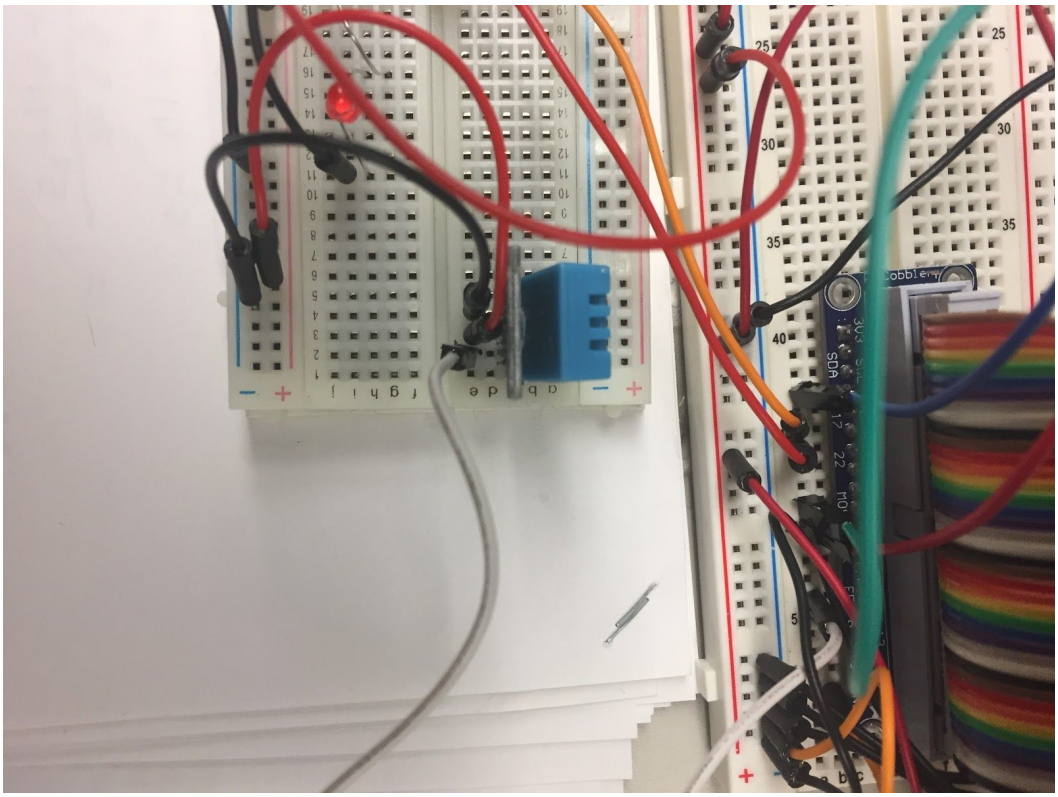

Figure 3: (*DHT11*)Temperature and Humidity Sensor Circuit

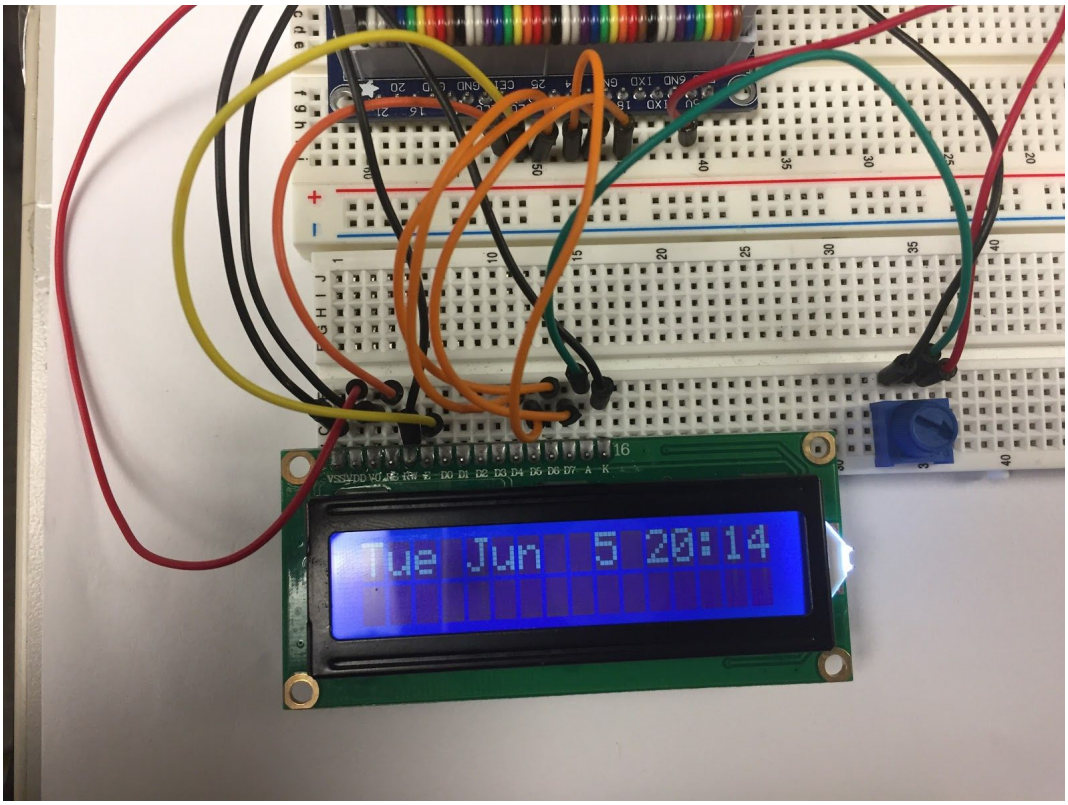

Figure 4: (*LCD1602 Module*)LCD Circuit

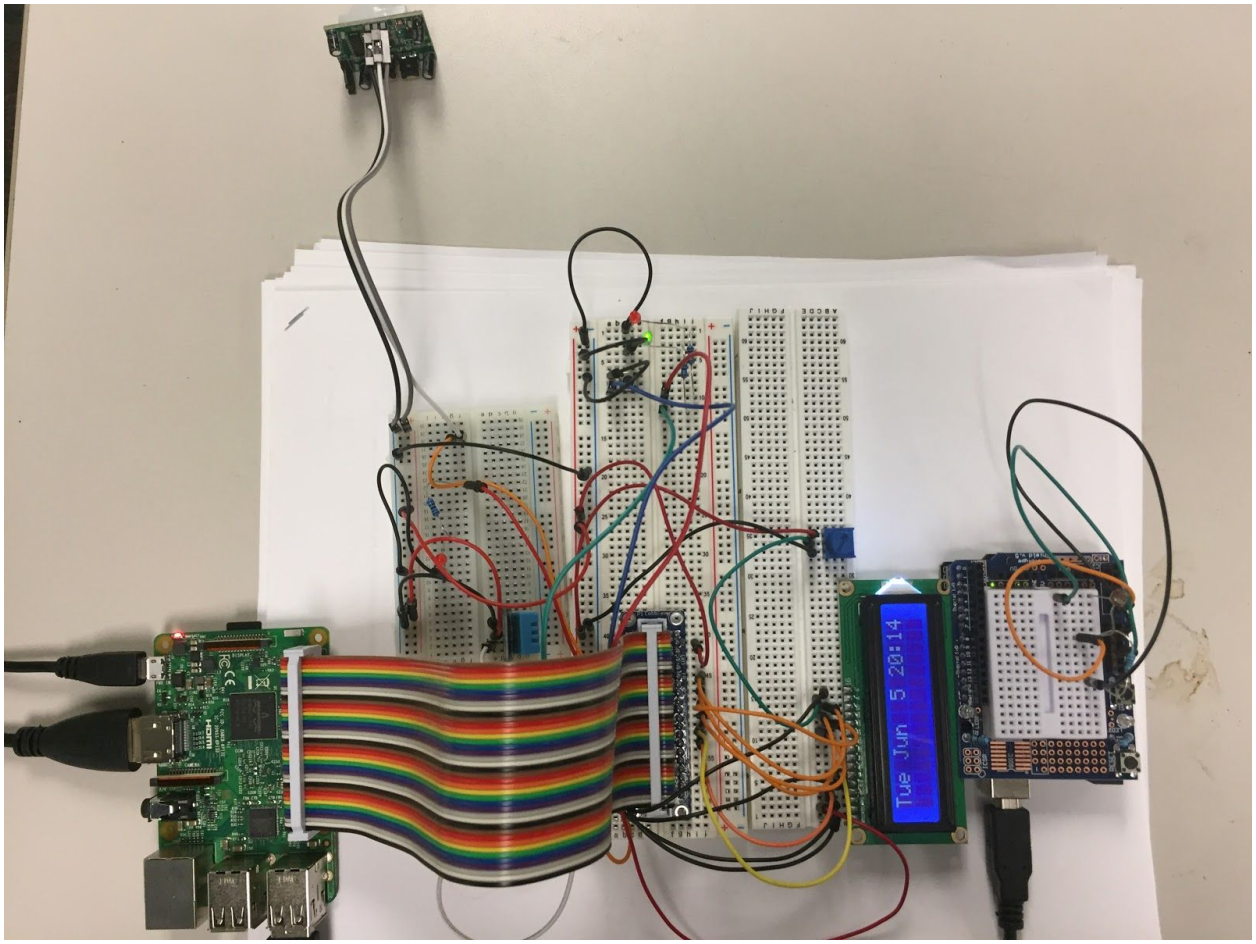

Figure 5: Arduino Uno Elegoo with Raspberry Pi 3

Code:

The code was written in Python, and compiled with Python v3.63. The code was split into different modules for handling different tasks for the IOT device.

Main Module:

- *main* function handles switching between the home mode and security mode. The switch is all held in a while loop
- *modeHome* function handles instantiating objects from the other modules that are needed for the lcd, and temperature and humidity sensors. Then the code enters a while loop to constantly retrieve data from the sensors, process the sensor data, and send the processed data to the user via LCD
- *modeSecurity* function handles instantiating objects from the other modules that are needed for the lcd, motion detector, and the LDR. Then the code enters a while loop to constantly retrieve data from the sensors, process the sensor data, and send the processed data to the user via email as an alert
- PIR Sensor code: The PIR sensor returns a 1 when it senses movement by infrared light. In security mode the code is implemented in a while loop that reads the input of the

sensor continuously. If the sensor is triggered it signals the LCD display to display “Motion Detected!!”, turns on a red LED, and sends an email to the owner warning them that there is an intruder within their home.

EmailNotify Module:

- *sendEmailIntruder* function handles sending an email to the user of the device in the case of either motion or change in light. This is achieved through the smtplib library.
- *receiveEmails* and *getCommands* functions receive and parse emails into commands that are understood by the main module as a data request. An email is then send back to the user depending on the type command that was sent. This is achieved through the easyimap library which allows us to retrieve the contents of a set of emails from the users inbox.

controlLCD Module:

- *getHumidity*, *getMaxTemperature*, *getLowTemperature*, and *getCurrentTemperature* provide additional information for debugging the temperature and humidity sensor by providing real time information on the outdoor temperature of Irvine.
- *lcd_init* function issue initialization commands to the LCD module like cursor direction font size, length, number of lines, clearing the display
- *lcd_byte* function takes the commands for the LCD and configures the GPIO aswell as the LCD for properly writing a string to it and includes code for the changing the screen every time the screen is updated.
- *lcd_toggle _enable* function toggles the LCD enable pin with the proper delays
- *lcd_string* function sends a string to the lcd via the *lcd_byte* function by transmitting each character in the string

Dht11 Module:

- The dht11 module took from an already provided python library that allows the user to retrieve raw data of temperature and humidity.

RESULTS

Figure 6: LCD Home Mode

The LCD displays the time, and the date on the first line and on the second line it displays the temperature or humidity.

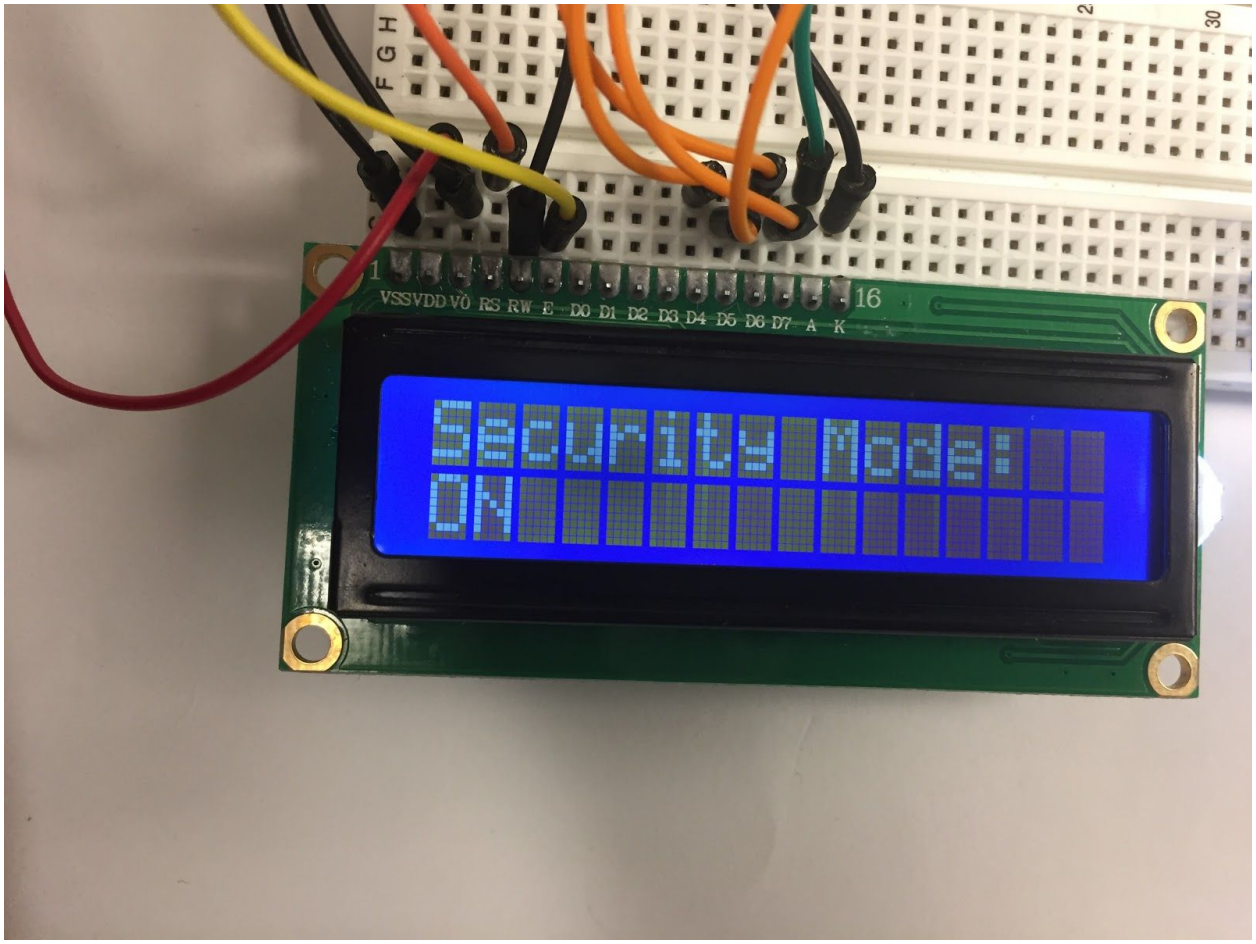

Figure 7: LCD Security Mode

The LCD displays the *security mode on*. When switching between the different modes the LCD displays Switch *mode*

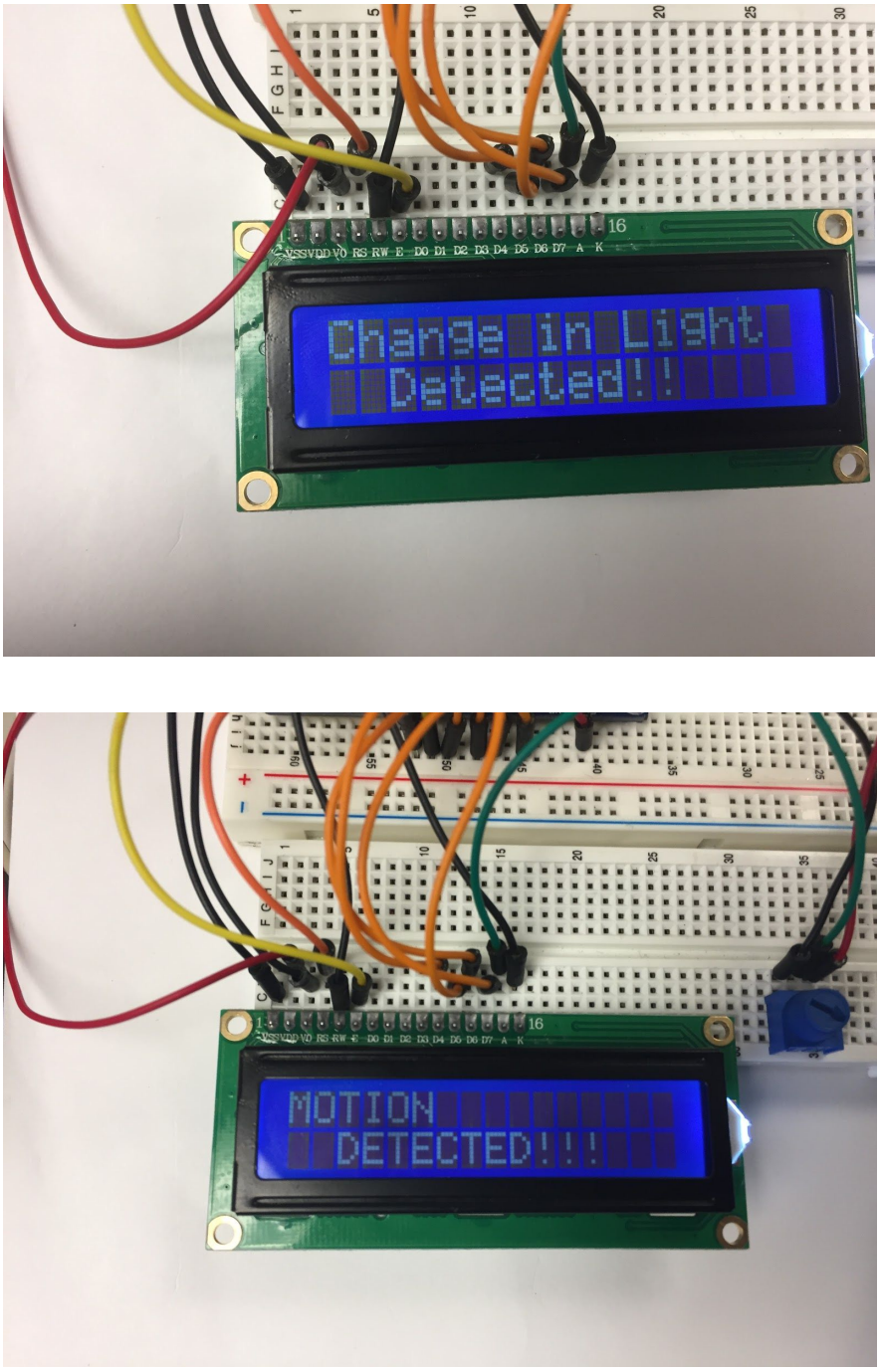

Figure 8: LCD Motion and Light Detected

The LCD displays motion detected when the light sensor senses a change in the motion of the environment or itself. It can also display when there is a change in the light level. These detections will automatically send an email to the user.

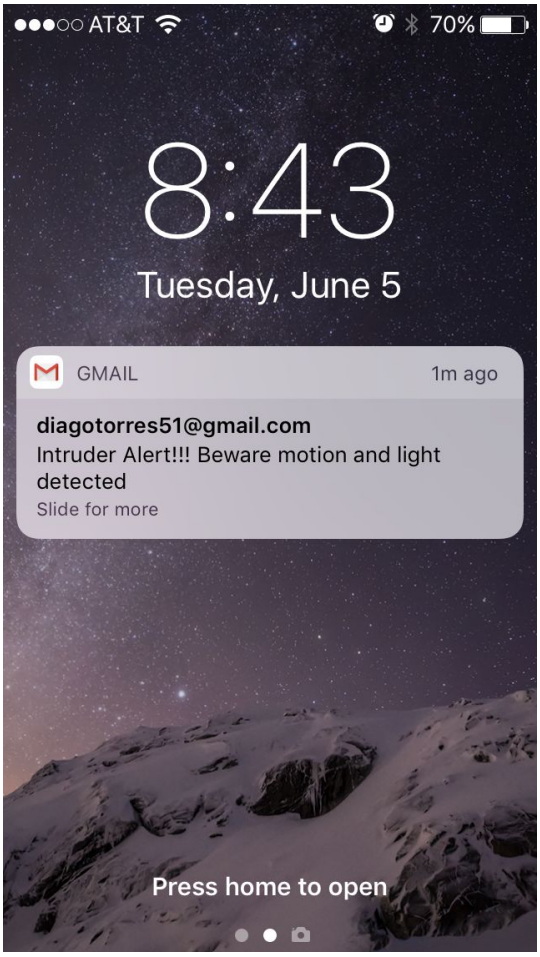

Figure 9: Email of Intruder Alert