PROJECT: lightSense

Implementation of Project along with Configuration Steps

Summary of Project:

Seasonal affective disorder, or SAD, is a mental health disorder that affects people in the winter months [Torres]. Generally, a person absorbs very little light during winter, and in some people, this can develop a condition where their brain becomes chemically and structurally imbalanced, especially as related to the natural circadian rhythm and the sleep cycle. This causes depression, anxiety, mood swings, and sleep problems in people affected by SAD. In the US alone, SAD affects approximately 5% of the adult population, which is close to 13 million people. In addition to medications, one of the most effective therapy for SAD is light therapy. The therapy is to simply being in the presence of a box that provides full spectrum light, which would sort of mimic sunlight. The only problem with this therapy is extremely low patient compliance. According to research published last year by Professor Faulkner from the University of Manchester, UK, the reasons range from a lack of knowledge about the importance of light to simple forgetfulness. The current commercially available models of light therapy boxes do not take patient interaction and compliance into consideration. Therefore, my project aims to bring the benefits of IoT devices, such as user interactivity, convenience, data manipulation/analysis and data storage, to the light therapy box, making the therapy easily compliable for patients.

Design of Solution:

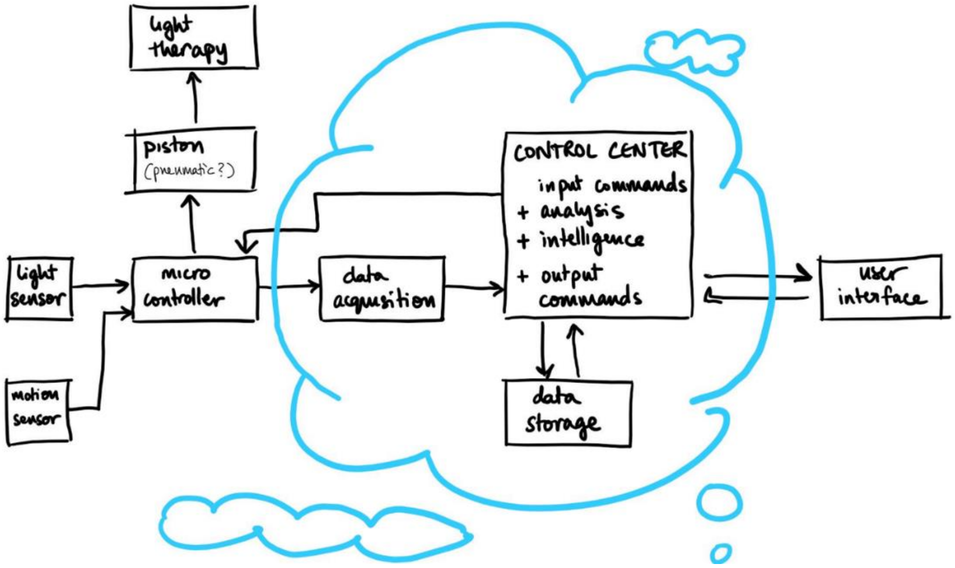

Figure 1

This is my overall design for lightSense. Data flows from the light and motion sensors to a microcontroller, then to the cloud. The user can configure the IoT to automatically turn on the light box based on various parameters: whether a person is near the light box, times when it is acceptable to turn on the light box and when it is not, duration of a single light therapy session, threshold of ambient light below which the light box will turn on, and so on. Commands are then sent to the solenoid via the microcontroller about pushing the button on the light box. This design has a threefold benefit. The user or patient only has to set up the initial instructions, and lightSense will automatically administer therapy. This reduces accidental forgetfulness and increases compliance. We are storing a comprehensive history of ambient light levels and light box usage. Within days of implementing lightSense, we can present to the user stark, clear-cut numbers about how much light is naturally available and how much light the box is providing. The user, upon feeling the benefits of the light box, and the statistics provided by lightSense, would become galvanized to keep a high compliance with the therapy. Lastly, with the user's consent, we could gather all this data region-wise to launch an educational and marketing campaign to raise public awareness of the benefits to having more light during the winter months and difficulties due to mental health disorders such as SAD. This has the potential to change light boxes from a simple treatment that a person is prescribed to a preventative method that almost everyone uses.

Implementation:

<u>List of materials to create prototype:</u>

(Suggested vendors for electronic equipment: Adafruit, Arrow Electronics and Chip One Stop Global)

- Ambient Light Sensor
    - High Dynamic Range Digital Light Sensor
    - Chose this because it has the capacity to sense up to 20,000 lux light intensity, and I need light sensors that can approximately sense 10,000 lux because light therapy lamps provide 10,000 lux of light.
- Motion Detector
    - Passive Infrared Motion Sensor
    - I required a motion sensor that can sense living objects within 6m of the sensor.
- Solenoid
    - Adafruit Mini Push-Pull Solenoid - 5V
- Microcontroller with Bluetooth connectivity

    - Raspberry Pi 3 Model B
    - Associated keyboard, mouse, display, power supply, SD card with Noobs (Raspian) software
- MOSFET transistor
- 10k Ohm resistor
- Diode rectifier
- Breadboard and wires (various types such as female-to-male and female-to-female jumper wires)
- Access to following software packages
    - Python 3.7
    - AWS Greengrass and IoT
    - AWS Lambda functions
    - AWS Elastic Computing 2 (EC2)
    - NOOBS (Raspbian)
    -
- Auxiliary materials – not needed but would add value to prototype
    - Light therapy box (my reason for not purchasing this item was a constrained budget)

In my design, the solenoid moving forward turns on the light therapy box, and the solenoid moving forwards turns off the same light therapy box. So, I only need to prove the solenoid to moves forward in appropriate conditions to prove my prototype works. While waiting for the equipment, I have determined the general ranges of voltage inputs I may receive from the light sensor and motion detector. To set up the infrastructure for processing the inputs I receive from the raspberry pi, I created a greengrass group in AWS. However, I could not yet set up the raspberry pi as the core, so I set up another greengrass group with an EC2 instance as the core. I wrote a python program to have the core perform the following tasks.

- publish voltage values to the "light" topic
- publish voltage values to the "motion" topic
- receive voltage values from the "therapy" topic

I wrote a lambda function that takes input from the "light" topic and prints to the S3 bucket whether it is necessary to turn on the light therapy box. It is a 1 ("yes") if the lux level is lower than 5000 lux. It is a 0 ("no") if the lux level is greater than 5000 lux. I wrote a lambda function that takes input from the “motion” topic and prints to the S3 bucket whether there is a person near the light therapy box. It is a 1 (“yes”) if the voltage detected is above 2V. It is a 0 (“no”) if the voltage level is lower than 2V. I wrote a lambda function that takes input from the S3 bucket about the “light” and “motion” topics. If both have 1 (“yes”) as their data points, then the function publishes “5” (representing 5V) to the “therapy” topic. Otherwise, it publishes 0. I was able to observe the Greengrass group working successfully through subscribing all 3 topics in the AWS IoT Console and observing the messages.

This is only my first iteration of the project, and it will contain more components as described in the “planned tasks” section. I had difficulty finding Raspberry Pi because many sellers were sold out of it, but I found it at last in Adafruit. I also got the rest of my parts from Adafruit. It was extremely difficult to have 2 inputs to the same lambda function. In the “rules” for this lambda function, I tried querying from 2 topics, and parsing the output through the “event” variable in the lambda function. It became too complicated and difficult to achieve. So, I decided to create 2 separate lambda functions, 1 for each input.

- Solder the light sensor to the pins provided and then insert them into the breadboard. I have not done this because I do not have the equipment, but I would strongly recommend performing this step because I had trouble with maintaining electrical connectivity with the light sensor.
- Connect the light sensor, motion sensor, raspberry pi, and solenoid to the breadboard as shown in figure 2. It should end up looking like figure 3.
- The solenoid is also a technical challenge because of the springs around the piston. Initially it worked well, but with multiple uses and many cycles of heating and cooling of the solenoid with the electrical current running through it, the spring itself expanded and contracted multiple time and lost it capacity to work properly. It made the piston stay in a position even when a voltage change would indicate the solenoid should exert enough force on the piston to change its position. I periodically slightly agitated the solenoid to allow the solenoid’s force to overcome the spring’s resistive force.

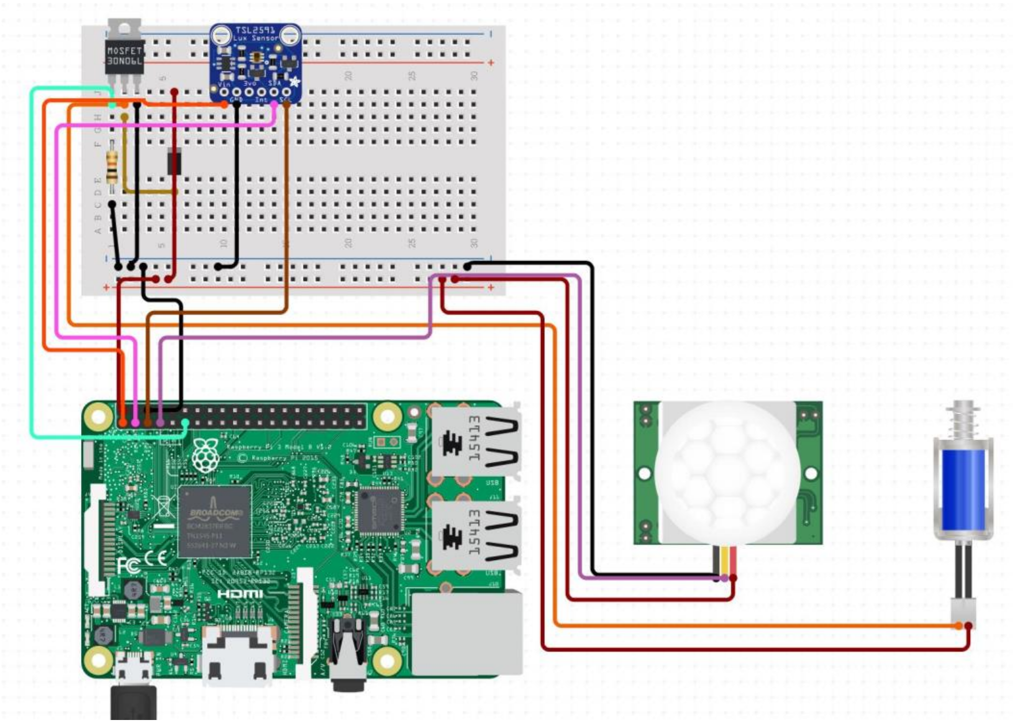

Figure 2

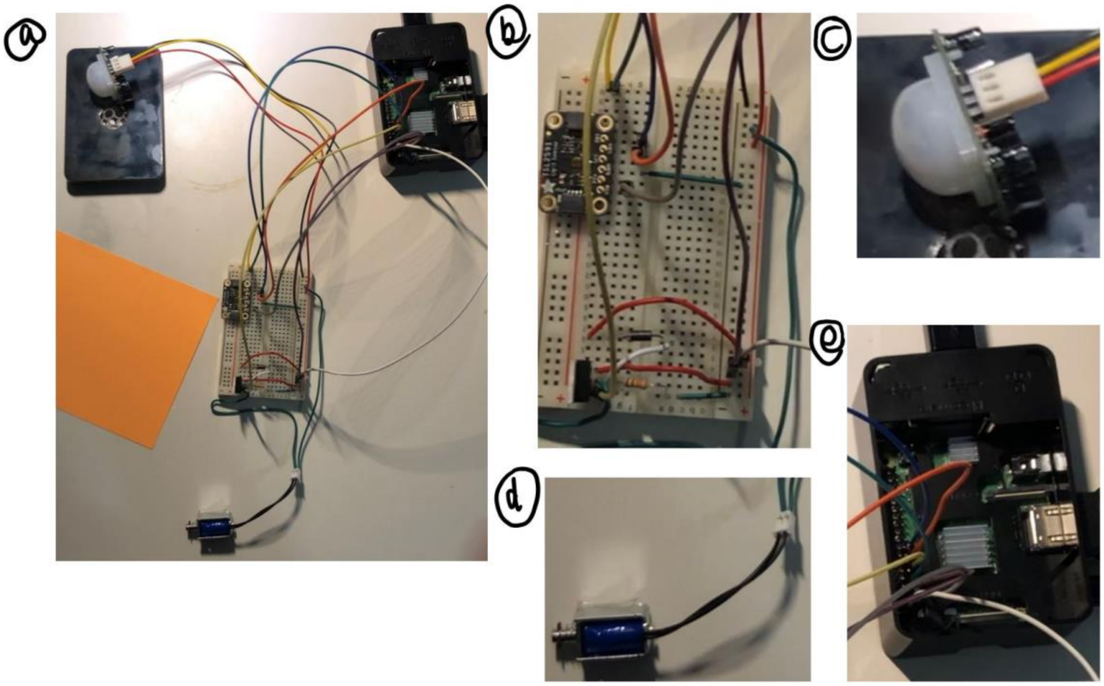

Figure 3: a) is the entire lightSense prototype. b) shows all the connections between the various components, and the ambient light sensor is on the top left of the bread board. c) shows the PIR motion sensor. d) shows the solenoid. e) shows the Raspberry Pi 3b

Configuration Steps:

- Register the raspberry pi as a core device in AWS Greengrass.
- Program a script to acquire information from this core. It must be able to publish 2 types of data: one would be voltage information concerning lux values to the topic “light” and the other information concerning IR readings to the topic “motion”. It must also listen to the topic “therapy” to receive the voltage information that could turn on the solenoid for a moment. The solenoid automatically springs back once pushed forward. To turn off the light therapy, the same voltage needs to be outputted again to push the solenoid forward (this turns off the light therapy box).
- Acquire user input through connecting this Greengrass group to an AWS EC2 Instance.
- Create a group of lambda functions that take this user data, messages from the “light” and “motion” topics to publish to the “therapy” topic and to save some relevant data in a S3 bucket.
- Whenever the solenoid is pushed linearly, the web app must be notified and an appropriate (“light therapy turned ON/OFF”) message displayed to the user.

References

Torres, F. (2020, October). Seasonal affective disorder (SAD). American Psychiatric Association. Retrieved November 2, 2021, from https://www.psychiatry.org/patients-families/depression/seasonal-affective-disorder.

Faulkner, S. M., Dijk, D.-J., Drake, R. J., &amp; Bee, P. E. (2020, March 12). Adherence and acceptability of light therapies to improve sleep in intrinsic circadian rhythm sleep disorders and neuropsychiatric illness: A systematic review. Sleep health. Retrieved November 3, 2021, from https://pubmed.ncbi.nlm.nih.gov/32173374/.