BEDROOM LIGHTING OF THE FUTURE

Module code and Name	**DE4-SIOT Sensing & IoT**
Student CID	01500108

Presentation URL (publicly accessible link): https://youtu.be/ujafhEVz7Ow
Code & Data (publicly accessible link): https://github.com/ShafaeAli/Bedroom-Lighting-Future-SIoT
Public Web App (Data Analysis and Interaction): https://bedroom-lighting-future.herokuapp.com/
Happy to share the video on Twitter/course YouTube?: YES

EXECUTIVE SUMMARY

Lighting is a major aspect of the bedroom which directly affects our mental and physical wellbeing. Artificial lighting has negatively impacted our sleep cycles. Smart bulbs can be scheduled to turn on/off, but do not truly sync based on the changing sunrise/sunset times to reflect the natural sun cycle. Lighting can also be made much more meaningful, particularly RGB which advertises itself as 'mood' lighting but is set manually by the user, whereas it has potential to be more naturally integrated. This report outlines development of Bedroom Lighting of the Future with two sub-systems. The first project aims to determine how misaligned my sleep pattern is to the natural cycle of the sun, and then making automated interventions through bedroom IoT devices that encourage my cycle to shift. It was found that my sleeping pattern was extremely unhealthy and improving it is important to avoid future health complications. Being able to visualise the misalignment of my cycle was more meaningful to illicit change, than just seeing the hours that I slept. It contextualised the data and showed me just how poor my schedule is. The second project helped me to visualise my emotions in an intuitive way, indicating expressions that I did not realise I was exhibiting. A potential route for this would be to aid non-verbal communication between individuals who struggle to communicate.

I learnt several skills in this work, achieving:

1. 7 time-series datasets collected through sensors setup on a RPi 3, APIs, FitBit sleep tracker and PC webcam.
2. Captured facial emotion data using OpenCV with an existing model and created my own personalised model.
3. Live data collection and storage on ThingSpeak, automatic .csv upload to Google Sheets, and local backups.
4. Performed data analysis using a scatterplot matrix and Pearson correlation coefficient matrix.
5. Created a public web app with data visualisation tools and novel interactions for understanding data.
6. Created 3 physical actuations linked to the live data – sun colour synchronised smart lights, Alexa controlled/daylight synchronised blinds and facial emotion triggered RGB lighting.

A comprehensive multi-part IoT system was created, fulfilling the objectives set out, and successfully exploring Bedroom Lighting of the Future through practical implementations.

COURSEWORK 1: SENSING

INTRODUCTION AND OBJECTIVES

In this project I am seeking to develop Bedroom Lighting of the Future, looking at 2 main areas - Circadian Rhythm lighting and Emotional-based lighting. These areas are linked as they form a complete lighting system but are two sub-systems in this report. that are connected to each other.

Circadian rhythms are 24-hour cycles [1,2] which our body synchronises to, using light and dark as the main timing factor. The biological interactions are extremely complex, but ultimately affect our metabolic rates, physical health, chemical signals, mental health, etc. For thousands of years, humans were synchronised to the sun, ensuring these cycles were optimal. With artificial light, these cycles have become negatively affected, disrupting our sleeping patterns, potentially increasing likelihood of cancers, cardiovascular diseases, and diabetes. Windows help alleviate this, but many city flats only have street level windows which are often frosted, increasingly reliance on bulbs. Smart bulbs with scheduling exist, but these are often for fixed times and do not synchronise to the cycle of the sun which changes throughout the year. I have a personal interest in lighting, inspired by the 2 summer placements at Dyson in the Lighting and Professional team, learning about the effects of light on our overall health. My motivation also stems from struggling with sleep patterns, often waking up groggy, at unsynchronised times with the sun.

The second part of the project seeks to develop the capability of smart lights by improving their 'mood' function. Instead of manual setting, it will be based on literal mood. This will be attempted with a webcam using a face emotion recognition model. Doing so may also aid individuals who struggle with non-verbal communication, by visually indicating emotion that someone may otherwise not pick up on, i.e., a parent in another room can know if their child is feeling sad since the home lighting changes to blue.

My project seeks to A) record my daily sleep states, the correlated colour temperature (CCT) of the sun, bedroom lighting and environmental conditions, then analyse the data and make insights on my patterns. Meaningful actuations will be made to synchronise my bedroom lighting with the natural sunlight cycle, but not disrupt my work by forcing any states, i.e., switching the light off. B) record my emotions by detecting my facial expressions and automatically change the light colour to indicate that my mood has changed. Project objectives and parameters are summarised (Table 1) – orange shows the circadian rhythm project, blue shows the emotional lighting, grey is general. A Gantt chart was created to organise my time and understand the overall tasks timeline (Appendix A).

	OBJECTIVE	PARAMETERS
1	Measure and record key bedroom environment parameters relevant to circadian rhythms and sleep.	Indoor light CCT (K), Indoor light level (lux), Indoor temperature (°C), Indoor humidity (%)
2	Measure and record sleep data to indicate circadian rhythm.	FitBit Tracker (Awake, REM, Light, Deep sleep).
3	Approximate the sun CCT outside using APIs as environment sensors are affected by artificial city lighting.	Astral Py Module (local sunrise/sunset times), PySolar API (solar elevation based on time), Elevation to CCT mapping function.
4	Measure facial emotion data through a webcam using an emotion recognition model.	Emotion prediction values for each class.
5	Develop a personalised emotion recognition model to improve detection for my face.	Emotion prediction values for each class.
6	Display results on a live web dashboard for visualisation and analysis.	/
7	Synchronise the lighting in my room with the natural cycle of the sun.	/
8	Actuate my blinds according to sunrise/sunset times.	/
9	Use an RGB bulb to visually indicate my emotions in real-time.	/

Table 1 – Project Objectives and Parameters.

DATA SOURCES AND SENSING SET-UP

CIRCADIAN RHYTHM LIGHTING

The Raspberry Pi (RPi) 3 was the full sensing, data collection and networking unit for the Circadian Rhythm element of the project. It collected data from environmental sensors (indoor CCT of light, indoor lux light levels, temperature, humidity) relevant to sleep and lighting, solar elevation data with post-processing to CCT using the Colour Science Library, and FitBit sleep state data to indicate sleep start/end times. In total there were 6 data sources. SSH was used to communicate with the RPi in headless mode. Since this opens a potential network vulnerability, the default username and password were updated. The RPi was running Raspbian, a Debian-based OS, and all code was ran using Python scripts. To enable the RPi to pull data and upload it, WiFi was required, however the chipset was limited to 802.11n at 2.4GHz, and our router range and speed on this channel was extremely slow, making it difficult to use. WiFi was also required to install packages on to the RPi. A 5GHz USB WiFi AC receiver was setup, installing necessary drivers manually, increasing internet speed from the Kb/s to Mb/s range.

BEDROOM ENVRIONMENT SENSORS

The RPi had a BH1745 luminance and colour sensor, and DHT22 temperature and humidity sensor (Figure 1). The RPi was placed on top of my PC as the light incident on the BH1745 here captured the most realistic estimate of CCT and brightness in my bedroom – the sources were from my window, ceiling lights and desk lamp. If I passed across the sensor then I cast a shadow over it, but this was a minimal trade-off as this did not happen very often and I avoided it by moving around it. The BH1745 sensor returned scaled RGB values to provide colour data unaffected by brightness. RGB was not the desired value, so the Colour Science Python package [3] was used to convert between colour spaces as there is no direct conversion between RGB and CCT. The raw values were converted to a NumPy array to use with the module. I experimented with different variations and found the most accurate conversion cycle to be RGB to XYZ → XYZ to xy → xy to CCT (using the Hernandez 1999 method in the final step) [4]. If the RGB values were 0, indicating darkness, the function was set to return a CCT of 0 - anything above this and the conversion algorithm was suitable.

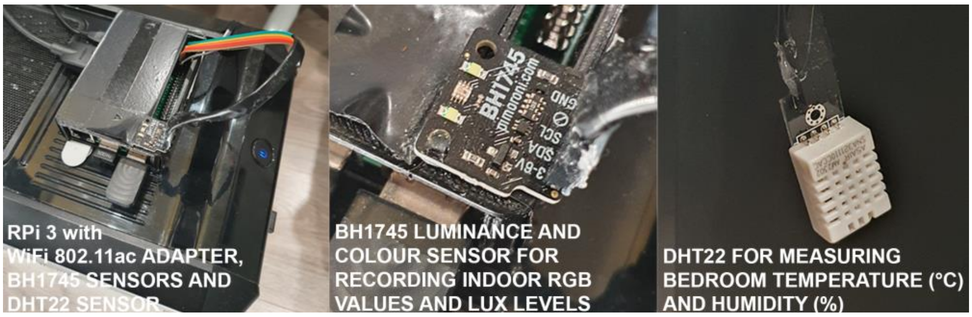


Figure 1 – RPi 3 Sensing Setup.

The BH1745 was only able to collect colour data in Red, Green, Blue and White (RGBW) with its Python library, and not lux (room brightness). Therefore, I adapted an alternative script from online [5] which directly accessed the I2C address. This used the System Management Bus (SMBus) function to write a value to a register on the I2C bus (which was found using the i2c detect function), to initiate the colour sensor. The data was then read from the White/clear channel block, but since these were raw bytes, it was multiplied by 256 to convert it to lux.

There were issues in setting up the sensing which were analysed and logically troubleshooted. The DHT22 would sometimes return NaN readings (i.e. 1 in every 500) which disrupted the continuity of data – this is a common issue with this type of sensor - so the function call was designed to ignore NaN and provide the last value read instead when writing data. This was suitable as bedroom temperature does not fluctuate massively and it ensures a complete data set without impacting data quality. The placement of the temperature sensor was in open air, away from the RPi and the PC as these generate noticeable heat if mounted directly or outside the fan outlets.

SUN CORRELATED COLOUR TEMEPRATURE (CCT) APIs & LIBRARIES

The RPi was also used to collect local sunset/sunrise times and solar elevation. This was advantageous because these times vary greatly throughout the year by up to 4 hours, so setting them as fixed times would not accurately reflect the sun, hence the use of an API. Solar elevation at these times were found and set as the min/max boundary for the CCT. The changing solar elevation throughout the day could now be mapped to CCT, better representing how the sun changes throughout the day – the height roughly corresponds to ideal CCT. The simplified model is based on [6,7], which maps the elevation values to maximum and minimum bounds for what the CCT should be at the corresponding sun elevation, e.g. at the peak of the sun elevation, CCT is set to 6000k [8] and at its lowest height it is estimated to be 0K. This model works well for the hours between sunrise and sunset, but outside of this loses its representativeness of the outdoor conditions as brightness becomes a factor. Trying to achieve this by interpolating between the times only would not be possible as you would struggle to create the correct curve function. The flow of this is demonstrated in Figure 2.

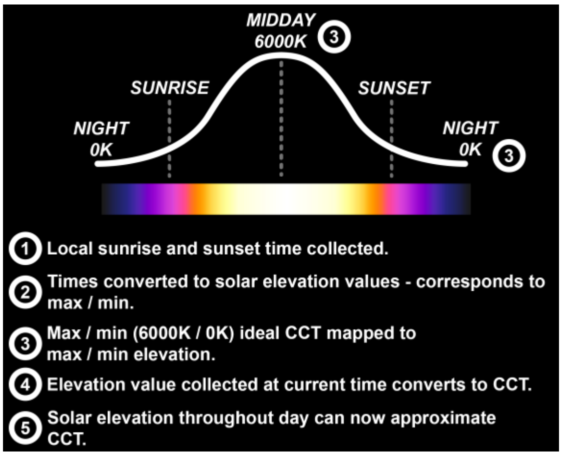


Figure 2 – Sun CCT Mapping Function.

[9,10] comprehensively covers a more complex, but accurate method to predict the solar CCT. But their model requires atmospheric conditions that would be difficult to acquire through public APIs or sensors. In this project, the focus was not to perfectly recreate the sun in my bedroom, but to evaluate how the CCT of the sun aligns with the CCT in my bedroom, and how this effects my sleep. Recreating the sun in my bedroom, with brightness included, would significantly alter my patterns as it would force darkness, which could have an adverse effect in the short-term whilst trying to meet deadlines.

FITBIT SLEEP DATA API

Different sleep sensors were researched such as Sleep As Android, Xiaomi Mi Band 6, Fitbit Charge 4, custom-built sensor, etc. The major limitation in sleep tracking is that it should not impact data collection, i.e., it should not cause my night to become restless. A wearable tracker is best suited for this as it can make measurements non-intrusively. Furthermore, sleep tracking is a combination of many sensors such as heart rate, accelerometer data, skin temperature, etc, and uses previous models to estimate the state of sleep, so a custom sensor may not be reliable. The 4 main states of sleep are awake, light, deep and REM. Awake is indicated by slight accelerometer movements and an increase in heart rate, whilst in deep sleep your heart rate and breathing can be 20-30% lower than normal [11]. A significant amount of post-processing happens before data can be labelled as 'light sleep' or 'deep sleep', and this happens on the application side. Most wearables work by collecting data locally on the device, then periodically syncing with a mobile app. Data can then be viewed by the user, helping to conserve battery life of the wearable as you are not constantly transmitting data. My focus was to gather information on start/end sleep times and sleep states for assessing sleep quality. Therefore, live sleep capture was not required only information after wakeup. The FitBit met these criteria with an accessible API. To collect data from the Fitbit server, a new application had to be registered to grant access to my personal data [12]. Implicit Grant Flow was used for authorization as this was best suited for web apps in terms of security, as these cannot keep client secrets. After enabling account access, an API endpoint URL was setup, allowing me to make requests to the server. The GET function in the Python requests library was used with the authorization token being placed in the header of the URL. The returned data was a JSON response and converted to a Python dictionary.

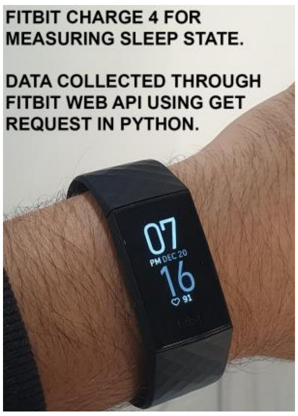


Figure 3 – FitBit Charge 4 Sleep Tracker.

EMOTION-BASED LIGHTING

FACIAL EMOTION DETECTION

The emotion detection script used the OpenCV Python module and Keras – a deep learning API in Tensorflow. The purpose of this script was to use my webcam to capture my emotion during work sessions and log them – the actuation on this would be changing the colour of my RGB lighting to match my mood. The detection code was based on the algorithm by Karan Sethi [13] and uses an off-the-shelf emotion recognition model trained on 27,261 faces from Kaggle [14]. Several models were tested beforehand, but this one minimised loss, making it stable during predictions. There are 5 classes of expression - Angry, Happy, Sad, Surprised and Neutral. The model was split into an 70/30 training and validation set, with the final result having an accuracy of 68% on the validation set. The model was run on my PC to provide the highest FPS. To install OpenCV and Tensorflow on Windows I created a virtual environment – this allowed me to use Python 3.8 and not affect other distributions or packages. The script works in 2 main parts, the first uses a classic facial recognition algorithm – a Haar Cascade classifier - to detect features on my face through edge detection and draws a box around the region. This is extracted and converted into a 48x48 grayscale image, so it matches the format of the h5 model. I optimised the detection for my lighting environment by changing the minimum number of neighbours from 5 to 8 for the classifier, reducing the number of false positives as it would occasionally detect my shirt. The data output was predictions of each emotion between 0 and 1, finding the index of the highest value and outputting the corresponding emotion. One of the drawbacks for a generic trained model is that emotions may have to be exaggerated to register, therefore I made and tested my own model to attempt to address this.

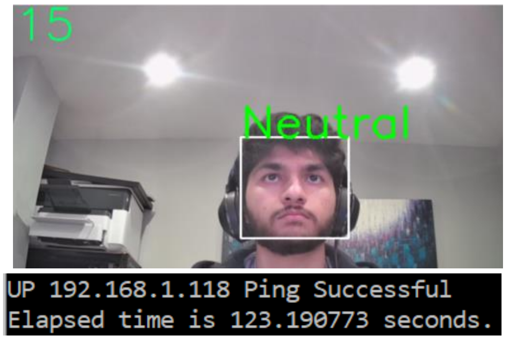


Figure 4 – OpenCV FPS and WiFi Presence Detection every 2 minutes.

PERSONAL EMOTION DETECTION MODEL

I explored training my own model solely on my face, to see if it could better capture subtleties in my expression. I had a selection of 386 images from my phone – 196 were removed as they were unsuitable. The facial detection algorithm was adapted to accept files rather than webcam frames as input. This took a input directory, iterated through the image files, converted them to grayscale, used the Haar Cascade classifier to detect and extract my face, rescaled to 48x48 in increase processing speed for the training phase, and then sequentially outputted as a jpg named 0,1,2, etc. Any images with multiple people were manually cropped beforehand. In future I would automate this process if the dataset was larger. The model was trained on a 7-layer CNN [13] with an ImageDataGenerator class to ensure new image variations were received at each epoch. The batch size was reduced to 10 to increase model accuracy as the dataset was small [15]. The number of epochs was set to 50, but early stopping was enabled if there was no improvement in loss function. The final model had an accuracy of 55%, which was not ideal. It was able to detect Neutral and Happy, being able to pick up subtleties in my expression as hoped but was ineffective at the rest. This is likely due to the dataset being 190 past images from my phone, which were biased to the detected emotions - taking up 75% of the dataset. As a result, I went forward with the existing model, but would be interested in retraining a personal model after collecting more emotions.

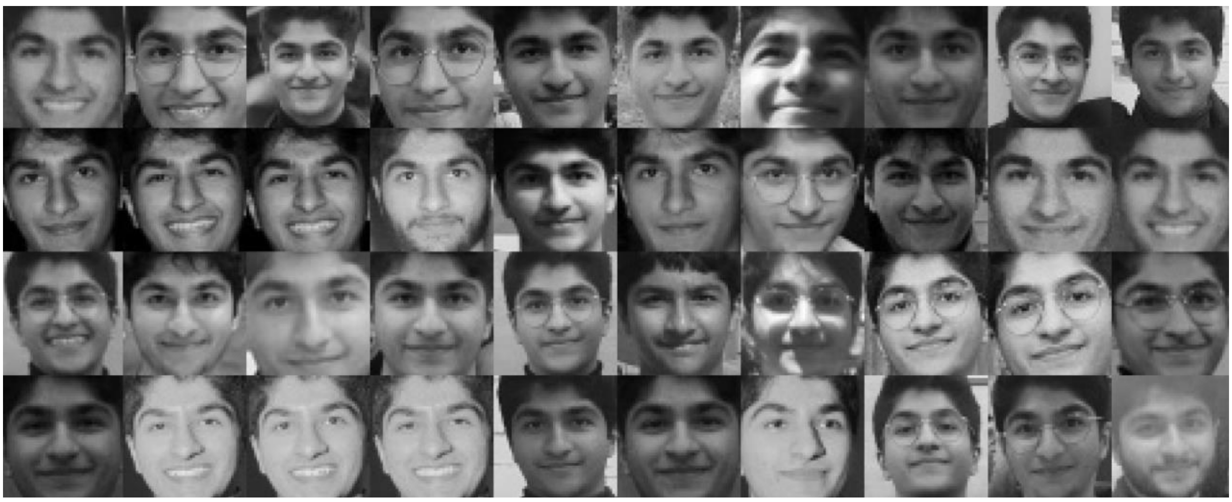


Figure 5 – Sample of processed 48x48 grayscale images from personalised emotional detection model.

DATA COLLECTION AND STORAGE PROCESS

CIRCADIAN RHYTHM LIGHTING

The sensing successfully ran for 8 days (18th – 26th November) uninterrupted, uploading all data to ThingSpeak. This demonstrates the robustness of the collection setup. The script has been setup to run from start-up, meaning if there is a power outage, although data would be lost for this period, the RPi should restart and continue logging. Data is uploaded using a POST request with the urllib.request command, with sensor data in the header. The upload limit for ThingSpeak was one request every 15 seconds, however during testing of the luminance sensor, this was excessive. Different sampling rates between once per second to 5 minutes were tested. The sampling rate was decided based on capturing events where I would switch on my lights for a short amount of time such as going to the bathroom – this was found to be usually 2 minutes. Hence, sampling rate for the environmental sensors was set to once per minute.

The Fitbit data was a challenge as it collects overnight, syncs to the phone in the morning, then uploads from the phone to the web server. Given the unpredictability of my sleeping patterns, a cronjob scheduler was used to run the fitbit.py script at 11pm to collect sleep from the night before. The limitation with this is how quickly I can access the data after it has been recorded, but this was not an issue as I was interested in the post-analysis and a live view would have no benefit as I would be asleep. Data was also saved offline in a local csv in case of internet outages.

EMOTION-BASED LIGHTING

The collection period for the Emotion system was on the 15th, 16th and 18th December in work sessions (approximately 11 hours) when I was at my desk. The camera frame rate of the OpenCV script was 15 FPS. This meant it can detect a change in emotion every 0.06 seconds. This was confirmed when writing data to a csv as 15 values were collected in 1 second. This rate allows every emotion prediction to be captured, but results in an undesirably large file size. As a result, I tested different averaging windows and found 30 to work best which logged data every 2 seconds. This data was saved in a local csv and automatically uploaded to Google Sheets using the gspread API. I built a privacy feature in the script that would only turn the camera on if my smartphone was on the WiFi network, indicating I was home. A Python subprocess was run every 2 minutes to ping the IP address and wait for a response to continue data collection. The data was not logged to ThingSpeak as the sampling rate exceeded the limit and was not bulk uploaded at intervals as it was not necessary to have a live view of my emotions online.

BASIC TIMESERIES DATA ANALYSIS

CIRCADIAN RHYTHM LIGHTING

A clear pattern of my circadian rhythm is shown by my sleep state in Figure 6. I get an average of 5 - 6 hours' sleep, starting early morning and ending before midday. A healthy circadian rhythm would be where I sleep at the troughs of the outdoor CCT, indicating night-time. Instead my sleep is almost opposite to this, demonstrating a highly out of sync cycle which can lead to decreased alertness, memory issues, tiredness, depression, etc [1,16]. My actuation will focus on addressing this by using smart bulbs to mimic the CCT function of the sun, to prepare a better sleeping environment. The CCT of my room closely follows my sleeping pattern, which is expected as I switch my lights off when I sleep. Instances where the CCT is 0K, but I am not asleep, show that I am not in my room, hence the sleep data is important for contextualising my circadian rhythm. The CCT of my room is also a stepped graph at a constant 4200K, whereas the natural colour of the sun fluctuates in a sinusoidal like pattern, with a slight phase shift occurring due to shifting sunrise and sunset times. This demonstrates how artificial light causes a binary lighting environment that can lead to unnatural patterns. There are 6 different data sources which relate to the overall bedroom environment (Appendix D). Overlaying these all over each other would not be practical for understanding correlations or trends, therefore I will create a Pearson correlation coefficient matrix and scatterplot matrix to help identify patterns across all 6 datasets.

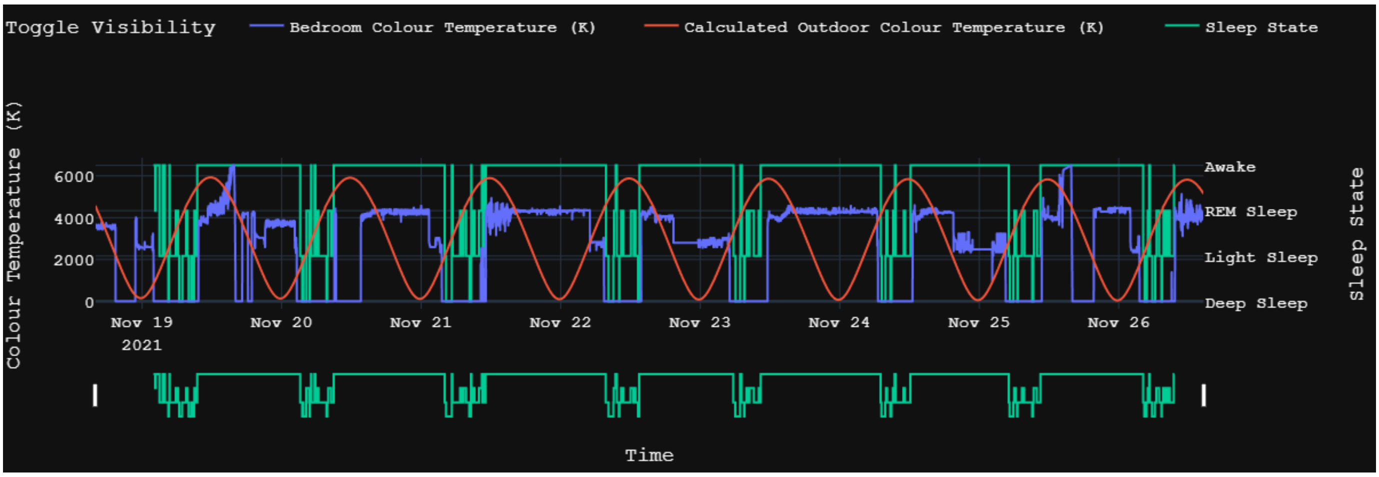


Figure 6 – Bedroom CCT, Calculated Outdoor CCT and Sleep State time series analysis.

EMOTIONAL-BASED LIGHTING

The raw time series data from the emotion detection is extremely noisy, despite averaging the past 30 readings into 2 second intervals (Appendix E). My main emotion was neutral which is expected as facial expressions may only appear briefly. The prediction model seems to bias Sad as this had the highest baseline prediction compared to the other emotions. To make the data easier to analyse, an interactive rolling average filter will be created to be able to tune the amount of smoothing live.

END-TO-END SYSTEM CHARACTERISTICS

An end-to-end system diagram, including actuation, is shown in Figure 7. See Appendix C for enlarged version.

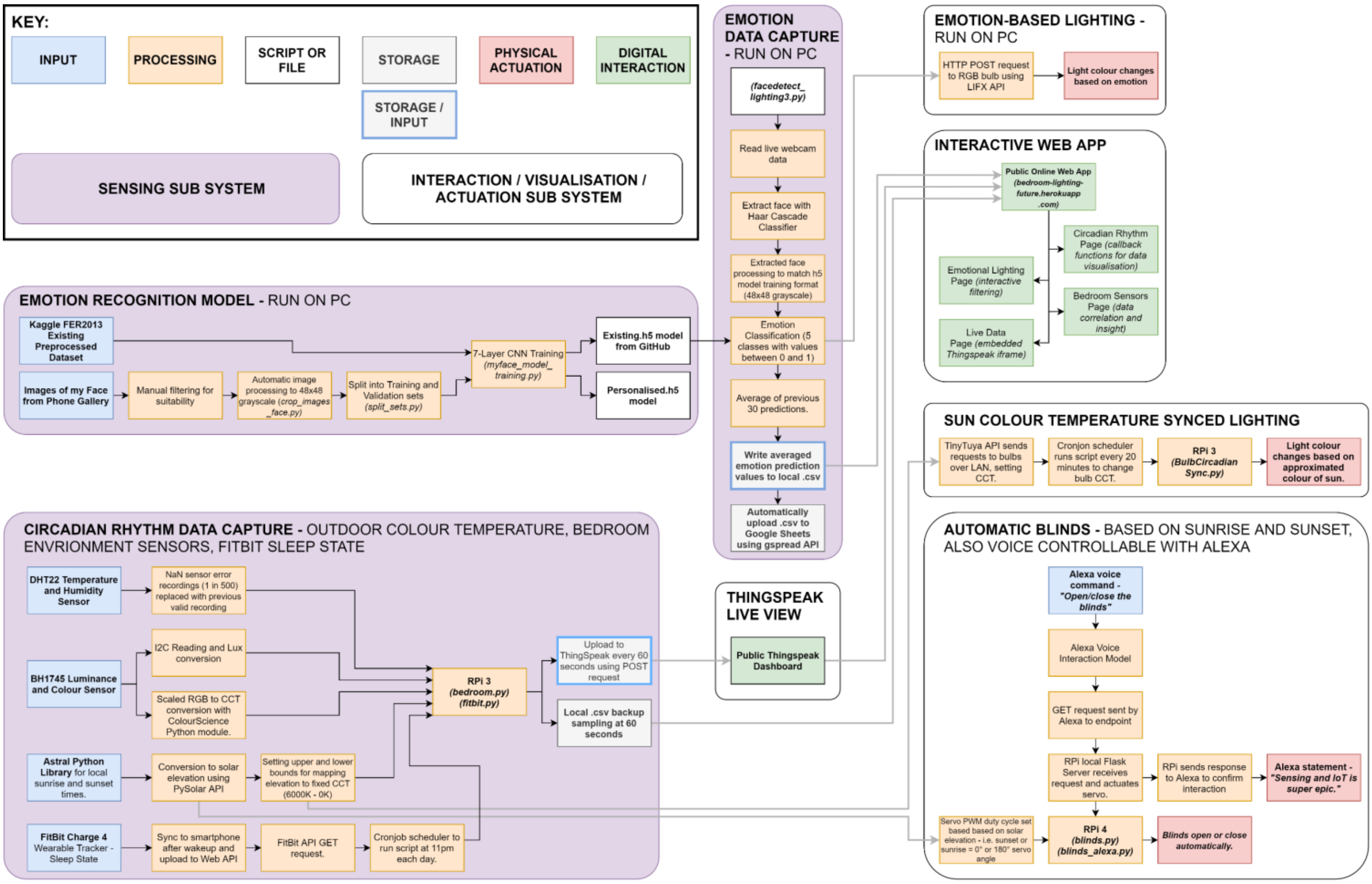


Figure 7 – Full end-to-end system diagram including actuation and data workflow. Large version in Appendix C.

COURSEWORK 2: INTERNET OF THINGS

DATA INTERACTION / VISUALISATION / ACTUATION PLATFORM

OVERVIEW – WEB APP (https://bedroom-lighting-future.herokuapp.com/)

A multi-page webapp was created to visualise and present data (Figure 8). The app was created in Dash on Python, which uses a Flask framework, allowing it to be built rapidly with HTML wrapper functions. It is deployed on Heroku making it publicly accessible. Several novel interactions help understand and contextualise data. Circadian rhythm is presented by graphing outdoor & indoor light CCT (K) and my sleep state. Users can hover over, and the light colour is shown as a swatch with the sleep state, allowing an instant indication of the light colour in my room and why, i.e., are the lights off because I am asleep or not at home. This was achieved using a callback function, where if the input property of a component changes, i.e., the hover location of the cursor on the graph, a property in an output component is updated, i.e., the colour of the swatch and sleep state text. These were used throughout my app to enable user interactions. The bedroom sensors page includes all the raw data plots, a scatterplot matrix and Pearson correlation coefficient matrix to understand trends. The emotional lighting page features emotions tracked during an 11-hour work session and has an interactive rolling average filter for tuning the window size. Finally, the live data page embeds ThingSpeak iframes to see real-time plots from the sensors. All pages can be seen in Appendix B. The web app could be optimised to improve loading times and also deploying it on an active web server. The Heroku app dyno falls asleep after 30 minutes, meaning it fresh loads if visited after a period of inactivity. This could be resolved by using a bot to keep the app active, or to host it directly on AWS.

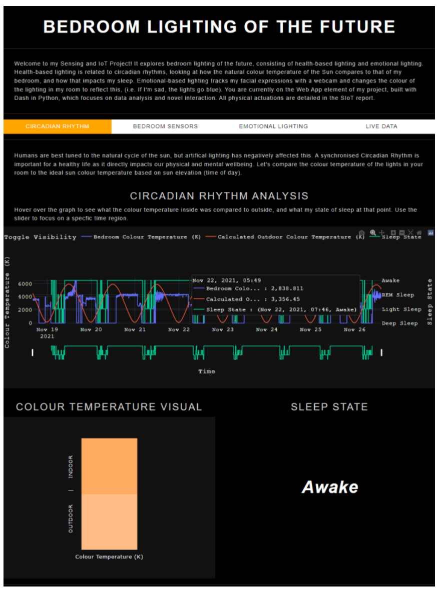


Figure 8 – Public Web App.

CIRCADIAN RHYTHM LIGHTING

SUN COLOUR TEMPERATURE SYNCED BULBS

Smart bulbs were installed in my room which can change CCT between 2700K and 6000K (Figure 9). There is a scheduling function to change the CCT over time, but this is static and does not interpolate between sunrise and sunset time. I implemented my CCT mapping function to dynamically update the smart bulbs colour so that my indoor environment matches the sun. This was achieved with the TinyTuya API running on my RPi 3 that accesses the bulbs using LAN. The bulb network address, Device ID (unique to Tuya), firmware version and local security key for encryption/decryption were required. The TinyTuya API has a built-in scanner which identifies the devices and returns connection information. A Tuya IoT account is required to setup a Service API to allow the security key to be fetched. The setup process returns a JSON file that contains information on each smart connected device that can be used for future scripts. Constantly updating the colour was distracting, so the script was set to run every 20 minutes with a cronjob scheduler. This was the minimum required to capture a sunset/sunrise event.

SMART BLINDS - SUNSET/SUNRISE SYNCED AND ALEXA CONTROLLED

My blinds physically actuate at sunrise/sunset by comparing current solar elevation to solar elevation at the time of these events. A MG946R servo was used, and a 3D printed housing and bar which draws the blinds, running on a Pi 4. The mechanism is simple and less prone to failure compared to earlier iterations which used a winch system to tighten or loosen the cord. Calculations for minimum servo torque and rod length were made by checking the force required to open the blinds, and the displacement of the string (Figure 9). The pigpio library was used to create accurate hardware-based timing signals rather than default software-based ones which can cause jittering in servos. Actuating the blinds encourages a bedroom environment more accurate to the sun.

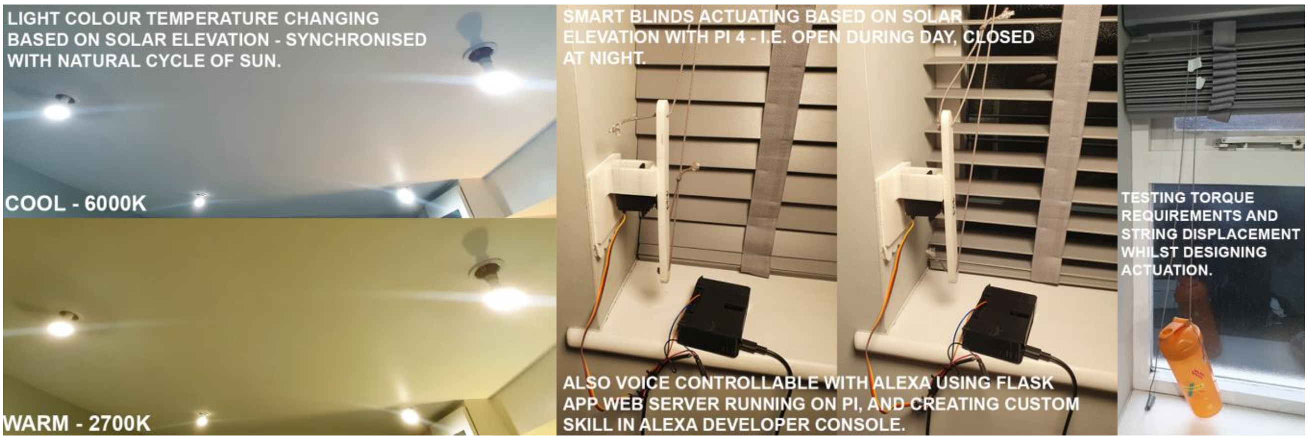


Figure 9 – Circadian Rhythm lighting physical actuation elements (sun synchronised bulbs and smart blinds).

Although the blinds actuate automatically, I wanted to expand its capabilities. I used the Amazon Alexa Developer console and the Pi 4 to enable voice activation. I created a custom Alexa skill where I defined a voice interaction model, created a Flask-ask server on the Pi to receive commands, and localtunnel to create a HTTP endpoint that Alexa could send requests to. Alexa processes my speech data, sends a request to the end point, that gets proxied to my local webserver, then a PWM signal is sent to the servo causing the blinds to actuate. The main issue was Alexa provided errors despite the blinds working. This was because no response was specified from the Flask sever, which was invalid. To fix this, a statement command was returned to Alexa to confirm the interaction had been received.

EMOTION-BASED LIGHTING

The emotion actuation changes the light colour based on the current emotion prediction. This works by sending a POST request to the LIFX API. The colour and emotion are stored as a dictionary, and requests are made at the request limit of the API, every 0.5 seconds. This was tested to be most appropriate for providing real-time responsiveness. The light changes everytime a new emotion is detected, except for neutral. This is because emotions only appear on our face for a brief period. The system could be improved by delaying the POST request so that there is less fluctuation when changing light – this happens as the emotion prediction model is jumping between values before settling - but this could equally be achieved by improving the prediction capability of the model. The final implementation is effective and achieves the fulfils the project objective.

Figure 10 – The OpenCV emotion detection script changes the colour of the bulb using the LIFX API in real-time.

DATA ANALYTICS, INFERENCES, AND INSIGHTS

CIRCADIAN RHYTHM LIGHTING

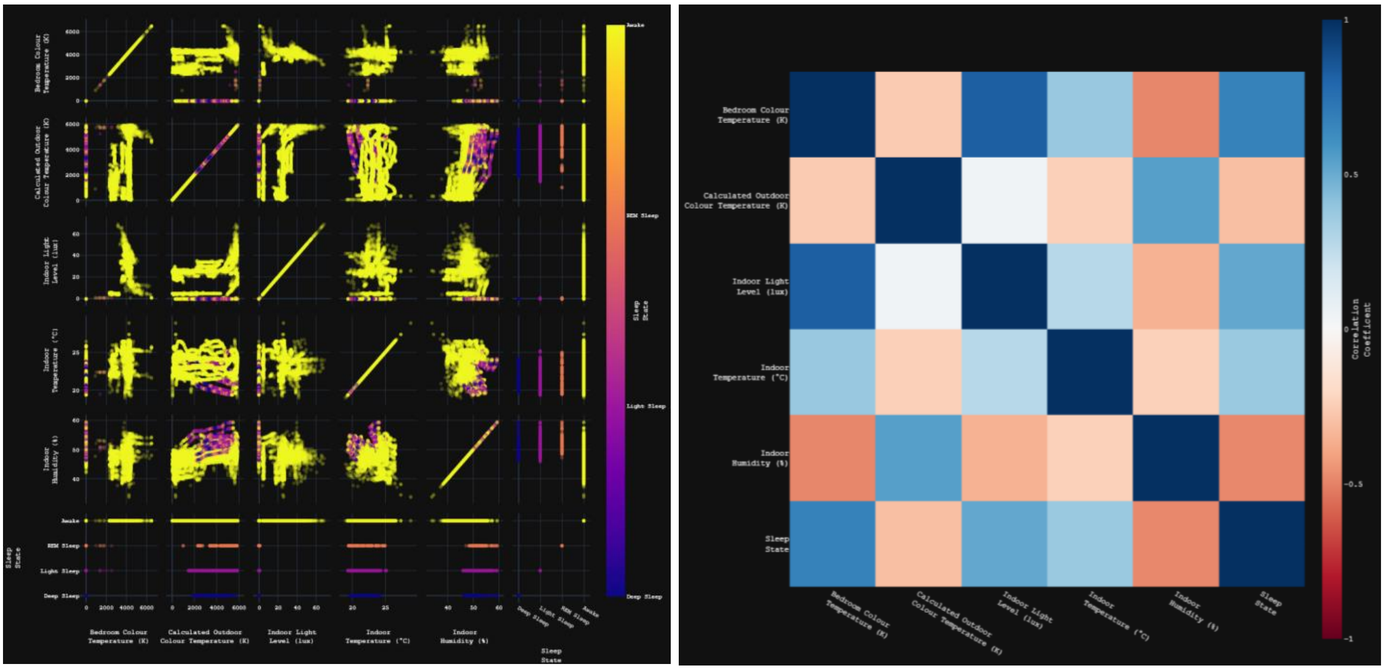


Figure 11 – Scatterplot matrix with sleep state colour scale to see data correlation between variables (left). Pearson correlation coefficient matrix to quickly visualise degree of correlation using a temperature scale (right).

An interactive scatterplot matrix was created to compare each of the measured values, and the sleep state shown as a colour axis to see how it changes across each comparison (Figure 11). The indoor light level and my sleep state clearly show correlation, where when I am awake, my lights are on, and whenever I am asleep, they are off – this is in line with expectation. A similar correlation occurs with my Bedroom CCT which is also in line with expectations. The Pearson coefficient supports this with moderate positive correlation of 0.52 and 0.67 respectively. The same pattern should be identified with the Outdoor CCT, to show I am sleeping at night and awake during the day, however the data is far more spread out across each sleep state, therefore indicating a highly poor circadian rhythm. The correlation coefficient for this is slightly negative at -0.3, supporting the conclusion of misalignment between my cycle and the natural sun cycle. This indicates to me that just getting the right number of hours of sleep is not enough but doing these at the right time is what I should be focusing on. I am potentially impacting my productivity and alertness with an asynchronous sun to sleep cycle. Further correlations show that the humidity is usually higher when I am asleep, and the temperature is lower - this is likely due to my ceiling lights, PC and monitors being off. There was a strong positive correlation between indoor light levels (lux) and bedroom CCT of 0.82, which is reassuring as these properties are directly linked.

Appendix D7 shows that my sleep cycle is consistent and follows a similar daily pattern. There is a 17-minute delay between switching my lights off and falling asleep, which is in line with the average amount of time it takes for a human to fall asleep [17]. This provides validation on the results, and shows that I likely do not have insomnia, where time taken to fall asleep increases significantly. Sleep data was important to record as my bedroom lighting is not enough to indicate information about my circadian rhythm. E.g. At 18:00 my lights turn off, but I am awake, showing that I am not in my room, therefore sleep data contextualises the results to ensure they are valid to the project objectives.

EMOTION-BASED LIGHTING

When I would be on a call to friends whilst working, and they mention something funny, the light would change to green – this showed me that I was smiling and laughing, making me much more self-aware of my expressions. We recently had an attempted break-in whilst in our flat. Following this, every time I would hear a noise that I thought was the front door, the light would change to blue, indicating ‘Sad’. This was extremely surprising, as I did not realise my facial expression changed during this – perhaps it was just for a split second (within the 0.06 second frame rate). Although this was not an ideal feeling to have, it demonstrated to me that the system works. The light was able to better identify and inform me of my emotions, even if I was not fully aware of them. Appendix E2 shows that the model is sensitive to sadness, so the prediction may have only lasted 1 frame and been a subtle expression, but this was enough to trigger the light. During the data collection period (11-hour work session), Sad peaked 3 times, but otherwise my facial expression remained relatively neutral. Given I was focused on work, this is not unrealistic.

DISCUSSIONS ON THE IMPORTANT ASPECTS OF THE PROJECT

Overall, I have met the project requirements that I initially set out. Out of all my projects to date, this one has taught me the widest range of technical skills and tested my ability to both design and fully implement a real-life working system. Although the timeline was dense, I was able to successfully achieve the goals I had originally set. There were however areas for improvement I would like to address.

For the project, I used ThingSpeak to store data online, due to ease of use and analysis capability, but this is inherently limited as it only accepts 15 requests per second. For the emotion data this was an issue given the required sampling rate, so decided to automatically upload it to cloud storage using the Google Sheets API. It would be better to instead use a platform like AWS or Google Cloud IoT Core which creates cloud storage buckets and is highly scalable. I explored Google Cloud briefly, setting up my own BigQuery table, a Pub/Sub topic to direct messages, and a MQTT protocol to allow the Pi to connect – but this was after already collecting a week of data with ThingSpeak, so decided to continue using the same platform.

The luminance sensor could have also been placed in a more reliable location as there were times where my body would cast a shadow over it, impacting the lux readings. The location was chosen as it best covered the three light sources in my room, but in future this should be mounted to a ceiling in the centre of the room, where it can record without being affected by shadows. Users can be highly unpredictable so creating robust sensing mechanisms is important within IoT systems, otherwise with unreliable inputs, it can be difficult to provide meaningful actuation.

The CCT function was idealised by directly mapping solar elevation at sunrise and sunset (the maximum and minimum values in a day) to the expected CCT at midday (6000K) and assuming no CCT at night (0K). This was effective in the project as it allowed the circadian cycle to be visualised, compared to my indoor CCT, and mapped to the CCT of smart bulbs. An improvement would be to perfectly replicate the CCT of the sun, as observed at a given location, by using the full equation definition [9] which considers scattering caused by atmospheric conditions and luminance gradients. From research, there are no easily accessible APIs that provide this data, but it will likely be collected by meteorologists in research. A simpler approach would be to use a colour luminance senor against the window to measure the outdoor CCT, but in a city, this would be impacted by cars and streetlights, preventing it from being natural.

The FitBit sleep data is collected by running a cronjob scheduler at the end of the day at 11pm to ensure all sleep data is captured from the previous night. Viewing sleep data live is not beneficial unless it is used to monitor someone else's health or during research experiments, hence this method is suitable as it is being used to evaluate personal circadian rhythm patterns over time. However, there is still a delay in collection, whereas it could be collected as soon as the FitBit has synchronised with the mobile app. This can be achieved using a Subscription API which notifies an application when new data is available with a POST request. The FitBit server would need to be polled at a rate at least half of when I would like to capture the data, i.e., to see the data within an hour of it being uploaded, I would poll in at least 30-minute intervals. If the request returns True for new sleep data being available, then it would trigger the existing sleep collection script, thereby eliminating the current latency in retrieving the data. Alternatively, the Device API can be used which would involve creating an app on the wearable directly, allowing data to be transmitted as soon as my sleep ends.

The mechanism for the blinds is larger than most existing solutions, but is far more satisfying, cost-effective, and robust. Typically, winch systems to wrap the string of the blinds are used, but this is prone to getting tangled. Alternatively, a teethed pully can be used to move the string whilst under tension, but this causes wear on the string and plastic components. The mechanism created is easy to mount and install on to a wall and interface with the blinds. It currently uses a Pi 4 but could be swapped to a Pi Zero W – this means the total cost is £8 + £1.04 + £9.30 (Servo + 3D Printed Parts + Pi) = £18.34. Other retrofitted motorized blind mechanisms cost anywhere between £50 - £115. The main issue of trying to market such a product are the safety concerns of the servo motor and rotating bar, which may be dangerous to children or during operation. I will however be posting the design as an open-source project due to its low-cost, allowing other DIYers to have smart blinds, but at their own risk.

The emotion-based lighting had a basic privacy feature built into it when capturing frames, which was presence of my phone on the WiFi, so that it only records data when I am in the house. The aim of this was so other flatmates would not unintentionally have their emotions recorded if they went into my room. Although no actual image data is captured, the presence feature was implemented to explore how privacy could be built into my algorithm. On a larger scale, this would be completely unsuitable as I pinged the device based on its IP address that was hard coded in my script. An improvement to this would be checking the local network for available devices, retrieving the data from this list, selecting the device to check presence for, and ensuring the connection was encrypted.

The bulb CCT synchronisation with the sun CCT can be marketed as a software plugin. A similar solution already exists for the Phillips Hue by a 3rd party, but the aim would be to make it available for a variety of bulbs. This would present challenges as each bulb is unique and have different workflows for creating apps for them. Also, some of these platforms may require a license to use at scale. The TinyTuya API used LAN to control the bulbs in this project, bypassing the Tuya Cloud development platform which costs $5000 dollars annually. The capabilities of the license are however excessive for a university project as it can handle up to 10,000 devices and 42 million API calls a month. Other platforms vary in cost and accessibility, but this variation does limit the devices. If the project did scale up and users were willing to pay for the capability, then this would not be an issue.

Data privacy is a major consideration for the sensing element as a lot of inferences can be made that can be harmful. Examples of this include knowing when people are out of the house or asleep, by simply using the light sensor levels as an indicator. Burglars could potentially use this information to enter the home when it is empty, or when an individual is not alert. For this project, the historical data and live feed has been made available, but this will only be temporary until January 2022. Other bedroom sensor data such as temperature and humidity pose less serious threats but should also be kept private to the user as it is still a form of personal information. Within the code, there are IP addresses, authentication codes, etc, so these have been removed before uploading on to GitHub. The calculated outdoor CCT was produced using already publicly accessible data, therefore this could be made into an API for others to use freely. The actuation can also be harmful if not secured as the lighting could be used maliciously to try and trigger seizures or the servo could be manipulated to damage the blinds.

AVENUES FOR FUTURE WORK AND POTENTIAL IMPACT

Several routes can be taken for the future work of Bedroom Lighting. With the circadian based system, it may be possible to treat seasonal affective disorder by recreating sunlight cycles from summer inside the home. Additionally, work from home is becoming increasingly common meaning we spend more time indoors, so better lighting environments are critical for our health to keep our cycles natural. This is particularly important for cities and flats where it may be difficult to get natural light due to street level windows, 24/7 city lights, etc. One of the limiting factors to using smart lights is the implementation cost and robustness. Many existing systems are difficult to setup and maintain, deterring users away, preventing them from accessing potential benefits. Existing bulbs also offer a higher degree of customisability that is unnecessary such as setting 255 RGB colours, syncing to music, etc. The problem is exacerbated by the lack of awareness of the importance of light on our overall health and wellbeing. Communicating the benefits of natural lighting, and creating a system solely based on health lighting will help progress this. Although this project uses IoT to adapt our circadian rhythms so that they are healthier, oftentimes societal pressures prevent us from fully automating these processes in the interest of our health, e.g., night shift workers, parties, work, etc. Therefore, flexibility in the system is required so that automation does not become counterproductive.

Emotional lighting could have positive impact in improving our ability for nonverbal communication. It could be used to help identify the emotions of those who struggle with this, e.g., children, pets, etc, and visually communicate it to someone else who may otherwise not have realised it or is in a different part of the home. This enables an intuitive human computer interface with instant visual cues that do not impede the attention of a user, such as a smartphone that sends a notification. It leverages the idea of a ‘calm computing’ environment. The RGB bulb is particularly suitable as colour is already heavily associated with mood, making it easy to understand with minimal instruction. Alternatively, the emotion recognition could be used to detect a negative change in mood, and enable a light colour to counter this, i.e., when starting to feel sad, a green light is produced to suggest feelings of happiness. Light therapy is already a form of treatment for depression, aiming to improve mood and sleep, so any future development of the system created would need to be reassessed to align with the findings in these formal studies.

I would be interested in further developing my personal emotion recognition model to include more emotions. The personal model built was a great learning experience, however, was limited by the number of emotions in my camera roll. The potential benefit of picking up subtleties in my facial expression were observed in the two datasets with the highest number of images, Neutral and Happy, demonstrating a case to further experiment. Generalised emotion recognition models can require quite exaggerated expression, limiting their use. An alternative would be to use biosensor data from a wearable tracker [18], monitoring heart rate, sweat, breathing variability, etc, rather than facial expression to indicate emotion. This could mitigate the issue of capturing subtleties through a camera, but introduces a latency, as the change in biodata would have to be monitored for a minimum period of time before indicating the emotion state. Another challenge would be transmitting the wearable tracker data live so that the lighting changes in real-time, as the battery life is limited, and constant uptime of wireless communication would cause this to drain much faster.

REFERENCES

[1] Brainard J, Gobel M, Scott B, Koeppen M, Eckle T. Health Implications of Disrupted Circadian Rhythms and the Potential for Daylight as Therapy. *Anesthesiology (Philadelphia).* 2015; 122 (5): 1170-1175. 10.1097/ALN.0000000000000596.

[2] Farhud D, Aryan Z. Circadian Rhythm, Lifestyle and Health: A Narrative Review. *Iranian Journal of Public Health.* 2018; 47 (8): 1068-1076. https://www.ncbi.nlm.nih.gov/pubmed/30186777.

[3] Colour Developers. *Colour Science Python.* https://colour.readthedocs.io/en/latest/index.html .

[4] Hernández-Andrés J, Lee J, R L, Romero J. Calculating correlated color temperatures across the entire gamut of daylight and skylight chromaticities. *Applied optics (2004).* 1999; 38 (27): 5703-5709. 10.1364/AO.38.005703.

[5] PIBITS. *BH1745NUC Luminance and Colour Sensor.* http://www.pibits.net/code/raspberry-pi-3-and-bh1745nuc-luminance-and-colour-sensor-example.php .

[6] Cambridge in Colour. *SUNRISE & SUNSET CALCULATOR.* https://www.cambridgeincolour.com/tutorials/sunrise-sunset-calculator.htm .

[7] Meeus J. *Astronomical algorithms.* 2. ed. 1998, with corr. as of August 10,2009 ed. Richmond, Virg: Willmann-Bell; 2009.

[8] Poirier G, Demers CMH, Potvin A. Experiencing Wooden Ambiences with Nordic Light: Scale Model Comparative Studies under Real Skies. *Bioresources.* 2017; 12 (1): 1924-1942. 10.15376/biores.12.1.1924-1942.

[9] Preetham A, Shirley P, Smits B. A practical analytic model for daylight. *Proceedings of the 26th annual conference on computer graphics and interactive techniques* : ACM Press/Addison-Wesley Publishing Co; Jul 1, 1999. pp. 91-100.

[10] Vollmer M, Gedzelman SD. Colours of the Sun and Moon: the role of the optical air mass. *European journal of physics.* 2006; 27 (2): 299-309. 10.1088/0143-0807/27/2/013.

[11] Corliss J. *How does sleep affect your heart rate?* https://www.health.harvard.edu/blog/how-does-sleep-affect-your-heart-rate-2021012921846 .

[12] Fitbit. *Fitbit Web API Developer Guide.* https://dev.fitbit.com/build/reference/web-api/developer-guide/ .

[13] Sethi K. *Emotion Detection Github.* https://github.com/karansjc1/emotion-detection .

[14] Kaggle. *Challenges in Representation Learning: Facial Expression Recognition Challenge.* https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data .

[15] Keskar NS, Mudigere D, Nocedal J, Smelyanskiy M, Tang PTP. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *5th International Conference on Learning Representations,* ; Sep 15, 2016.

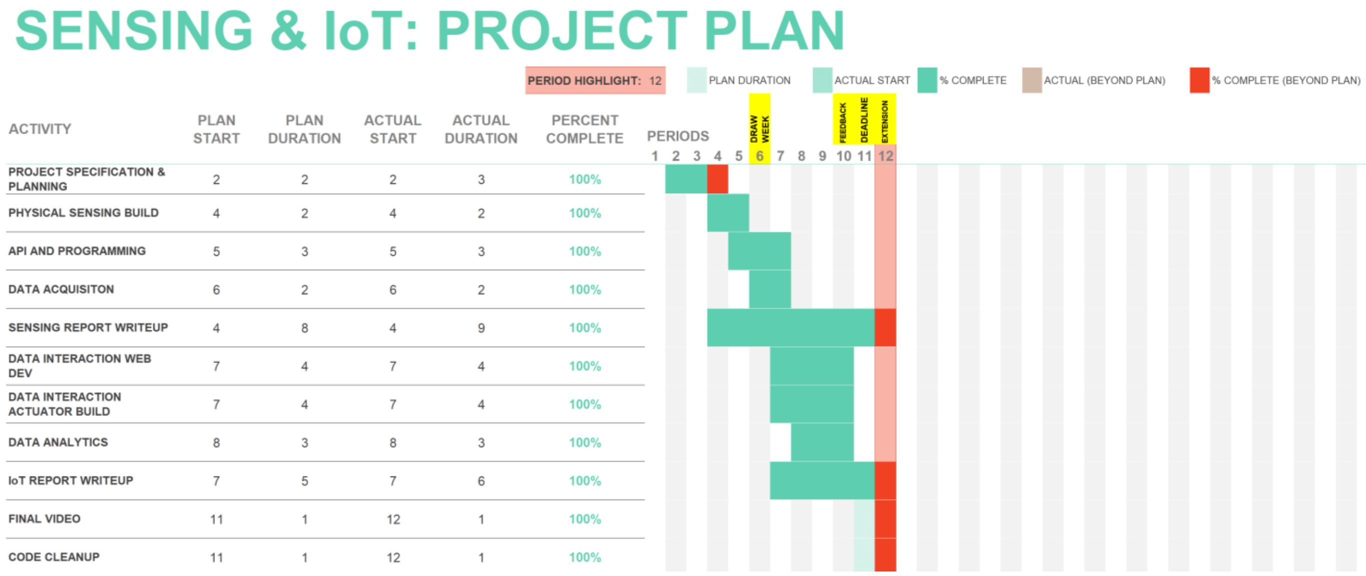
[16] National Heart, Lung and Blood Institute. *Circadian Rhythm Disorders.* https://www.nhlbi.nih.gov/health-topics/circadian-rhythm-disorders .

[17] Silver N. *How Long Does It Typically Take to Fall Asleep?* https://www.healthline.com/health/healthy-sleep/how-long-does-it-take-to-fall-asleep#normal-sleep .

[18] Shu L, Yu Y, Chen W, Hua H, Li Q, Jin J, et al. Wearable Emotion Recognition Using Heart Rate Data from a Smart Bracelet. *Sensors (Basel, Switzerland).* 2020; 20 (3): 718. 10.3390/s20030718.

APPENDIX

APPENDIX A: PROJECT GANTT CHART

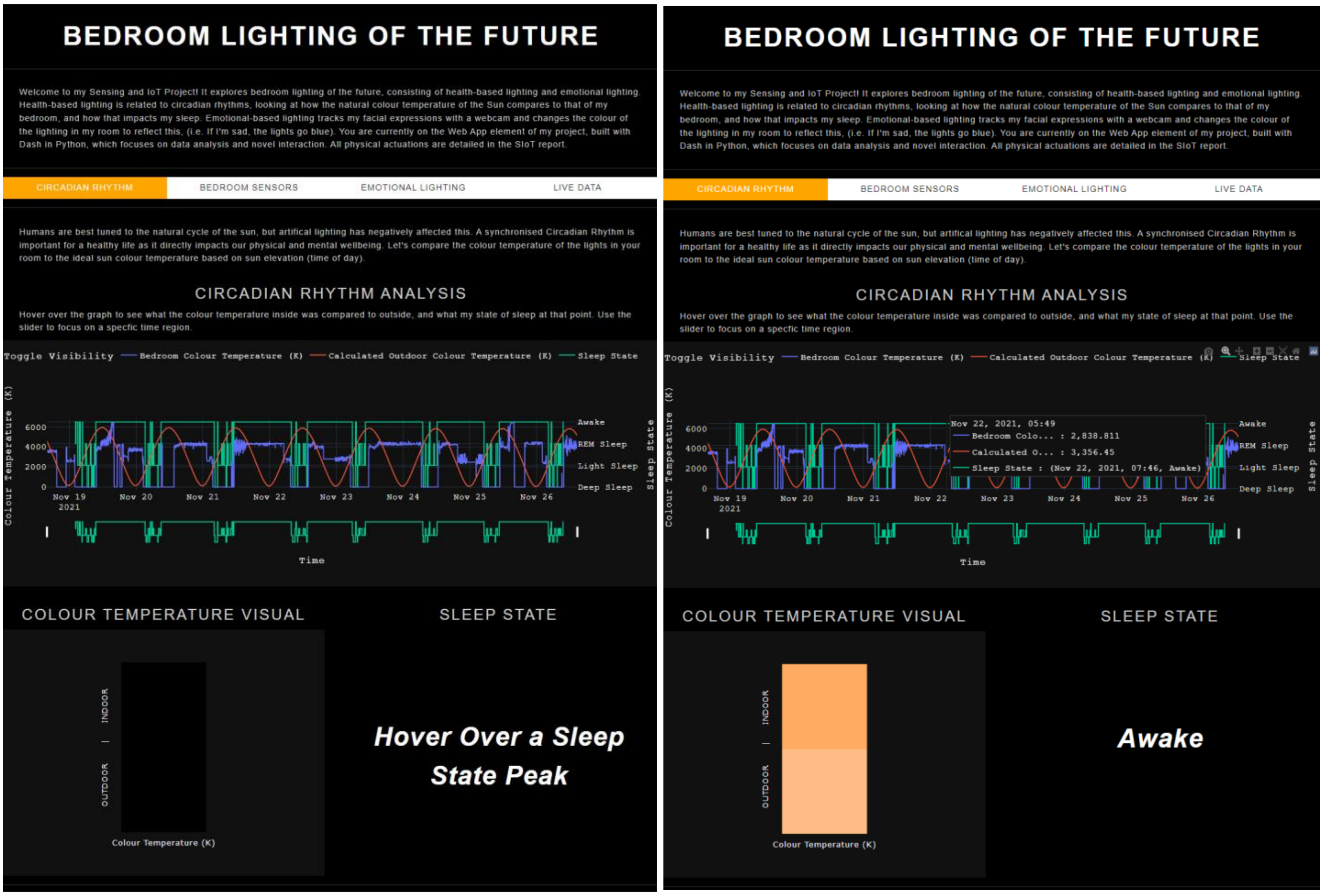


ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
PROJECT SPECIFICATION & PLANNING	2	2	2	3	100%
PHYSICAL SENSING BUILD	4	2	4	2	100%
API AND PROGRAMMING	5	3	5	3	100%
DATA ACQUISITON	6	2	6	2	100%
SENSING REPORT WRITEUP	4	8	4	9	100%
DATA INTERACTION WEB DEV	7	4	7	4	100%
DATA INTERACTION ACTUATOR BUILD	7	4	7	4	100%
DATA ANALYTICS	8	3	8	3	100%
IoT REPORT WRITEUP	7	5	7	6	100%
FINAL VIDEO	11	1	12	1	100%
CODE CLEANUP	11	1	12	1	100%

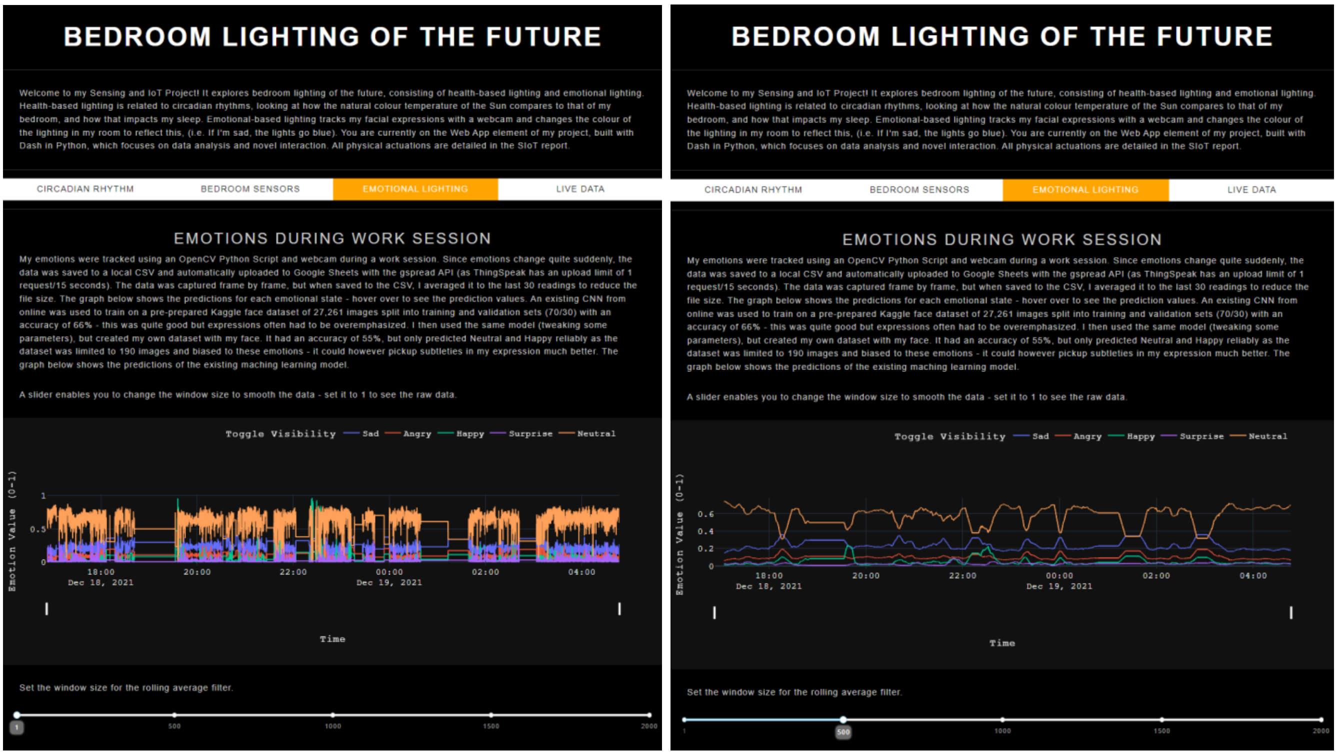
Appendix A – Project Gantt Chart detailing task timeline from start to end.

APPENDIX B: WEB APP SCREENSHOTS

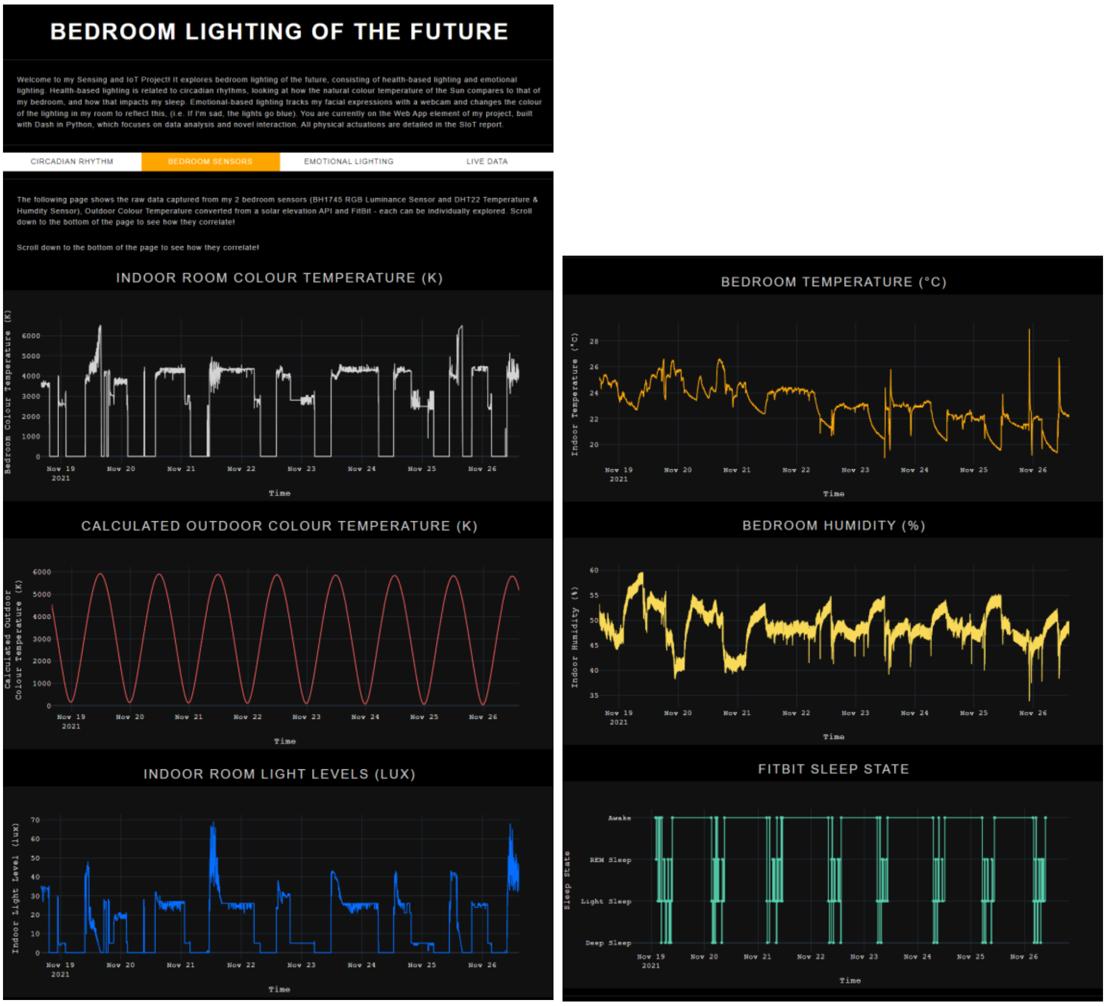
Below are screenshots from each page to show evidence of the web application in case the server goes down. The web app is also shown in the video.



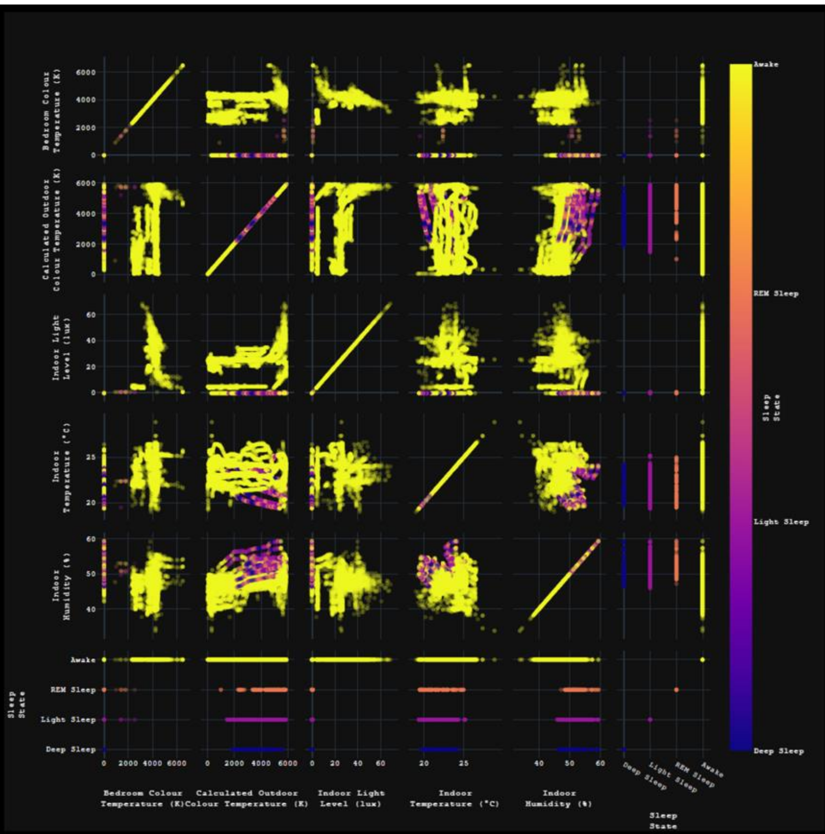
Appendix B1 – Circadian Rhythm Page contextualising data from the outdoor CCT, indoor CCT and sleep state. This page is the core culmination in terms of data visualisation and analysis, allowing me to see the alignment of my lighting environment and sleep cycle to the natural cycle of the sun. It indicates massive misalignment and that I should take steps towards fixing it – some of which have been achieved in this project. Variety of technical features implemented on this page which enables live user interaction.



Appendix B2 – Emotion Lighting Page with interactive data averaging (adjustment of rolling average window size using slider achieved through callback function).

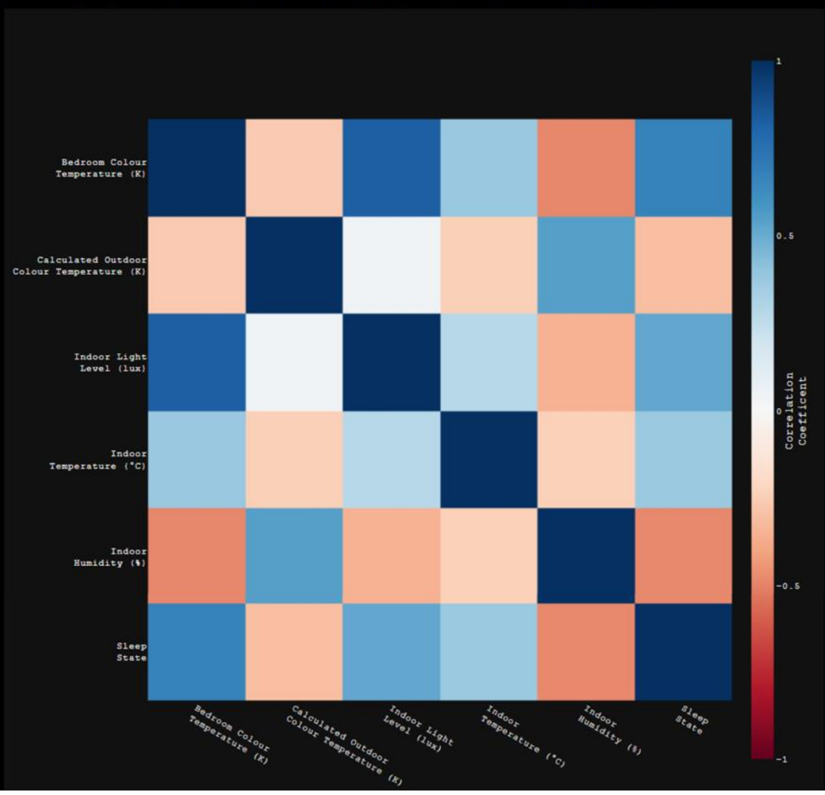


Appendix B3 – Bedroom sensors page showing raw plots of historical data.



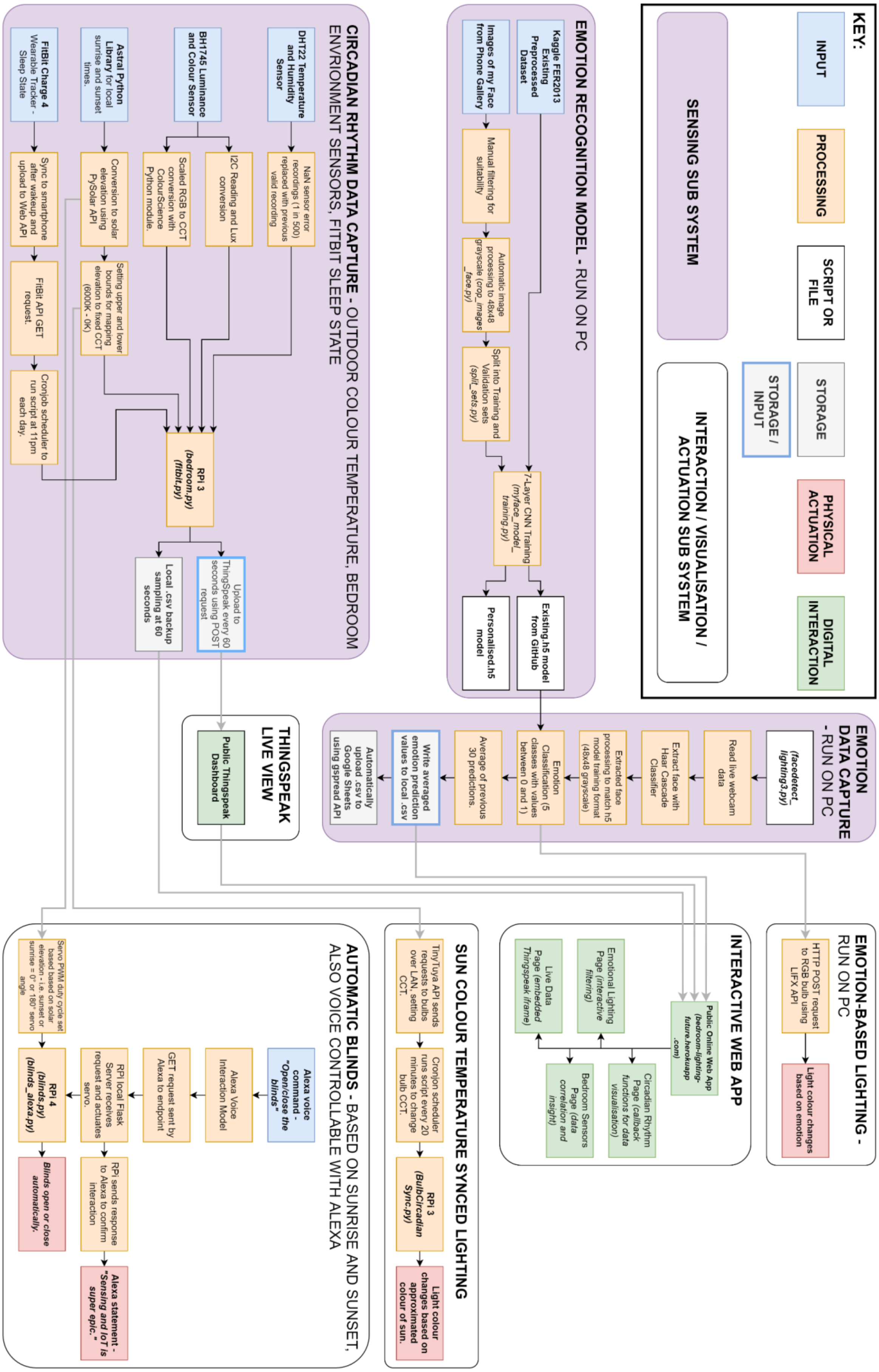
PEARSON CORRELATION COEFFICIENT PLOT

A feature correlation plot shows the Pearson correlation coefficients using a colour scale. This provides greater insight than the scatter matrix above, but doesn't allow as much exploration of the data. There positive correlation between the Bedroom Colour Temperature and Indoor Light Level, which makes sense as it shows my lights are on when light colour is detected. My sleep state has a negative correlation of -0.3 with the Outdoot Colour Temperature, which is quite bad given that this should be closer to 1, let alone negative. This indicates to me that just getting the right number of hours of sleep is not enough, but actually doing these at the right time is what I should be focusing on. I am potentially impacting my productivity and alertness with an asynchronous sun to sleep cycle.



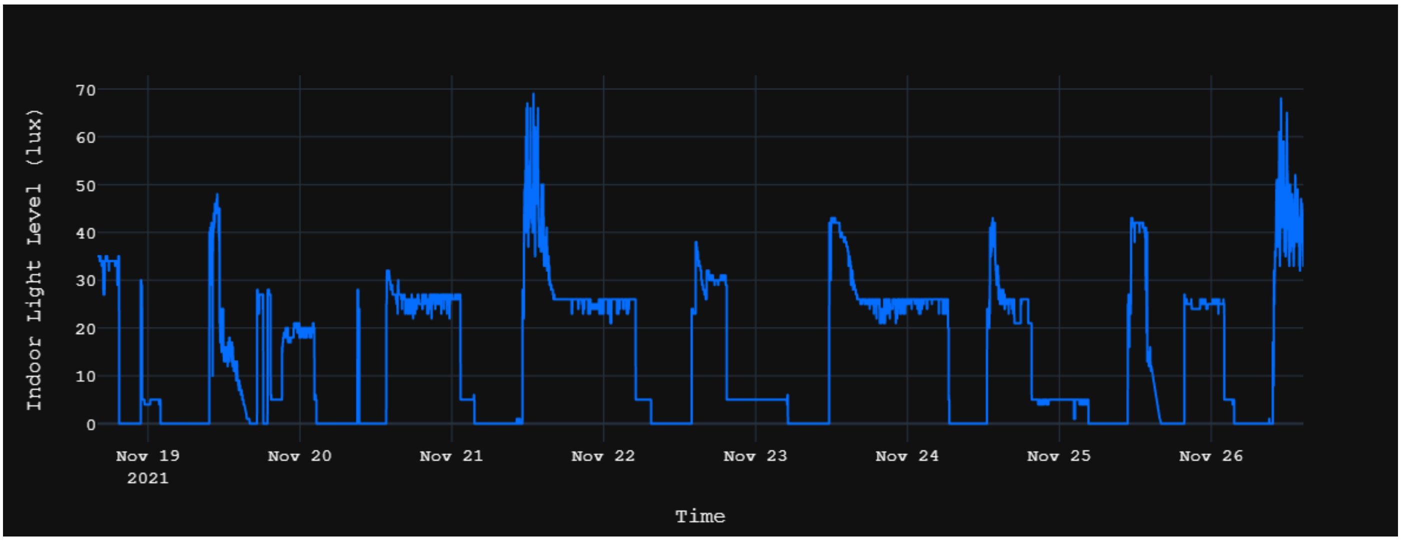
Appendix B4 – Interactive scatterplot matrix and Pearson correlation coefficient matrix graphs on Web App.

APPENDIX C: SYSTEM DIAGRAM

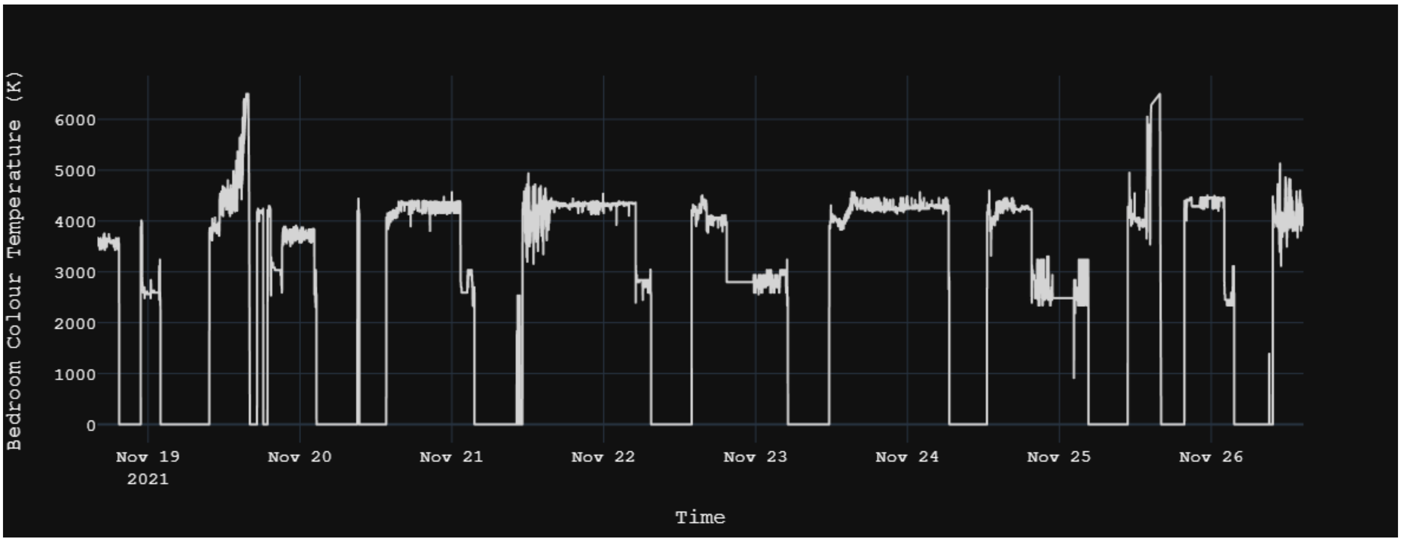


Appendix C – Full end-to-end system diagram including actuation and data workflow.

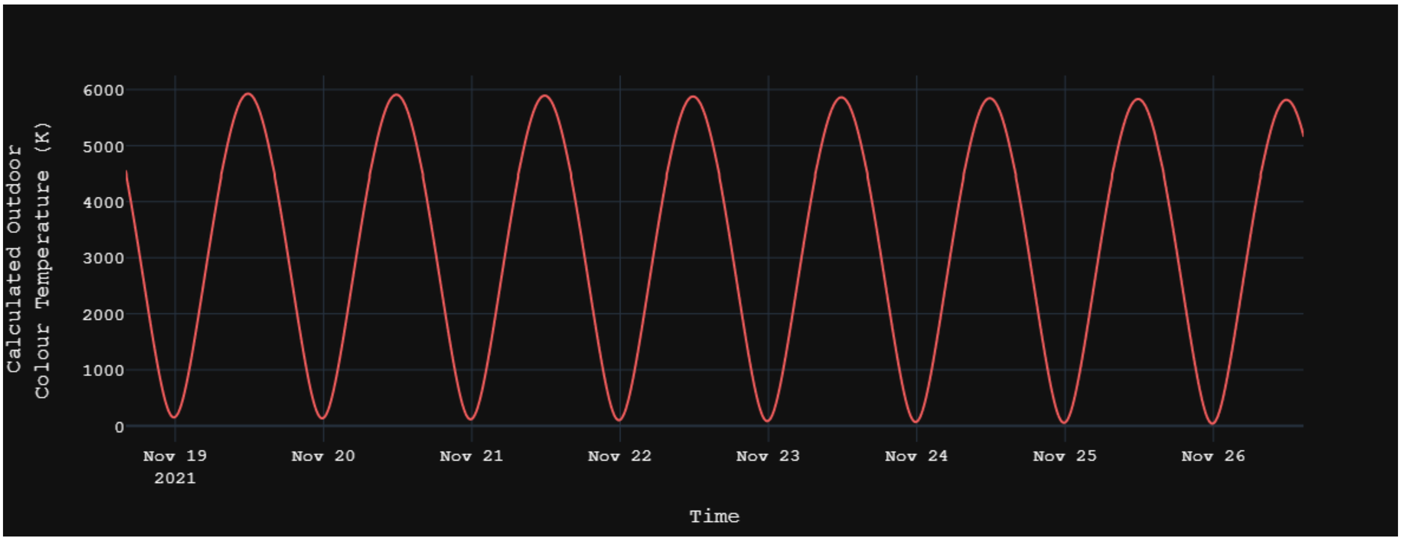
APPENDIX D: CIRCADIAN RHYTHM LIGHTING TIME SERIES DATA



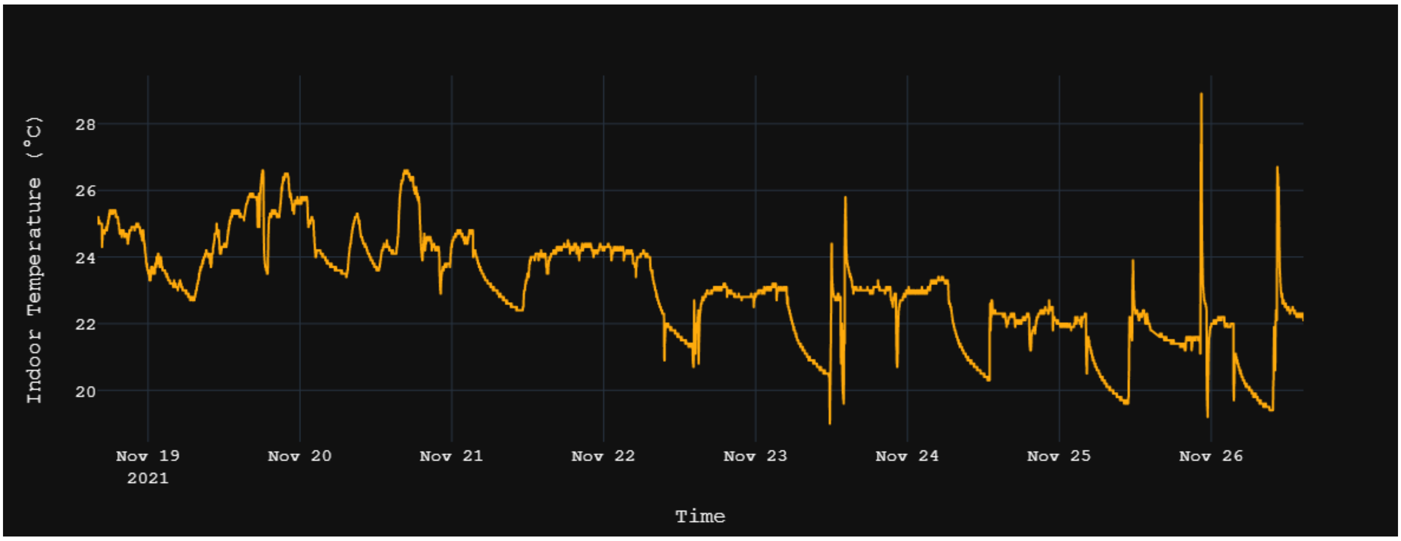
Appendix D1 – Indoor Light Level (lux) versus Time.



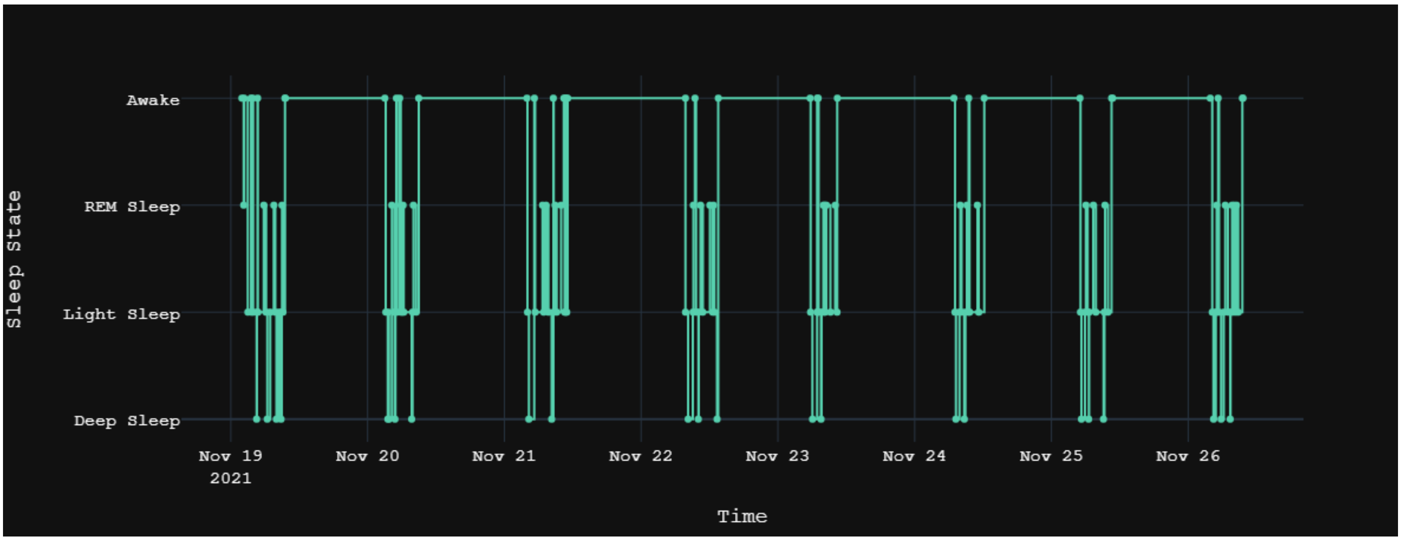
Appendix D2 – Bedroom Colour Temperature (K) versus Time.



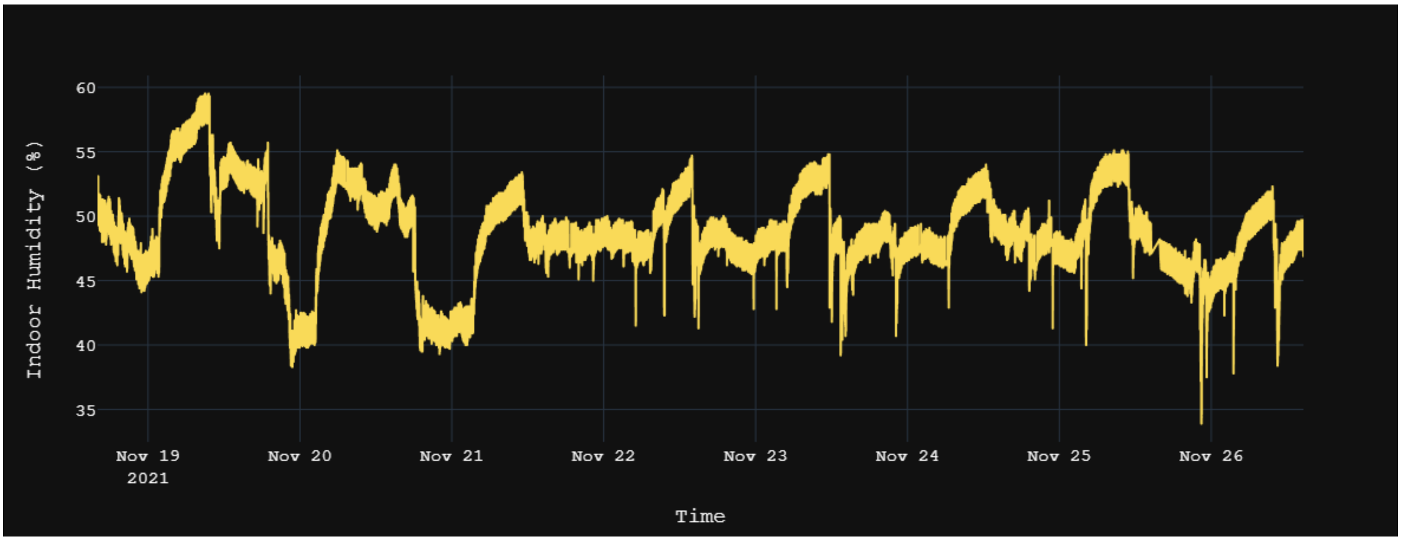
Appendix D3 – Outdoor Approximated Colour Temperature (K) versus Time.



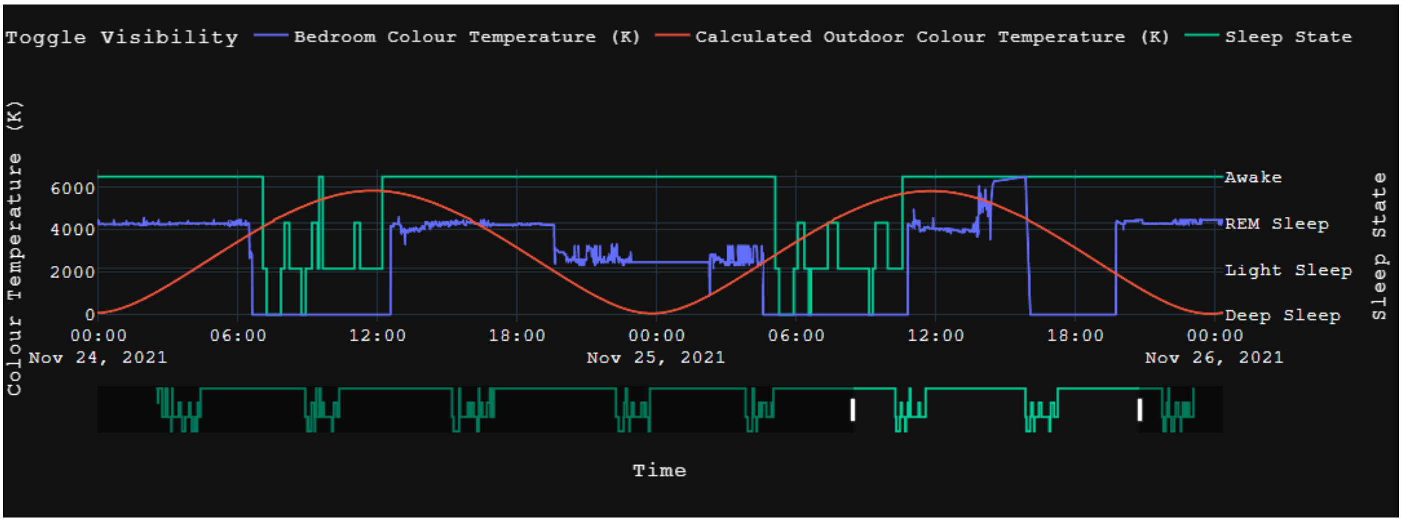
Appendix D4 – Indoor Bedroom Temperature (°C) versus Time.



Appendix D5 – FitBit Sleep State versus Time.

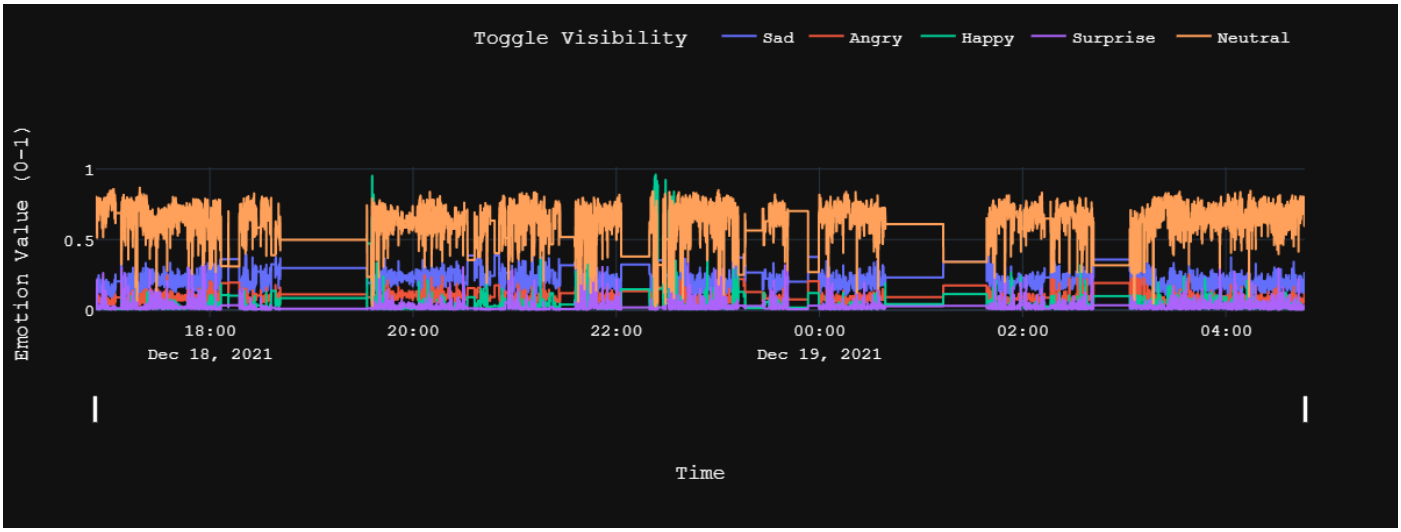


Appendix D6 – Indoor Bedroom Humidity (%) versus Time.

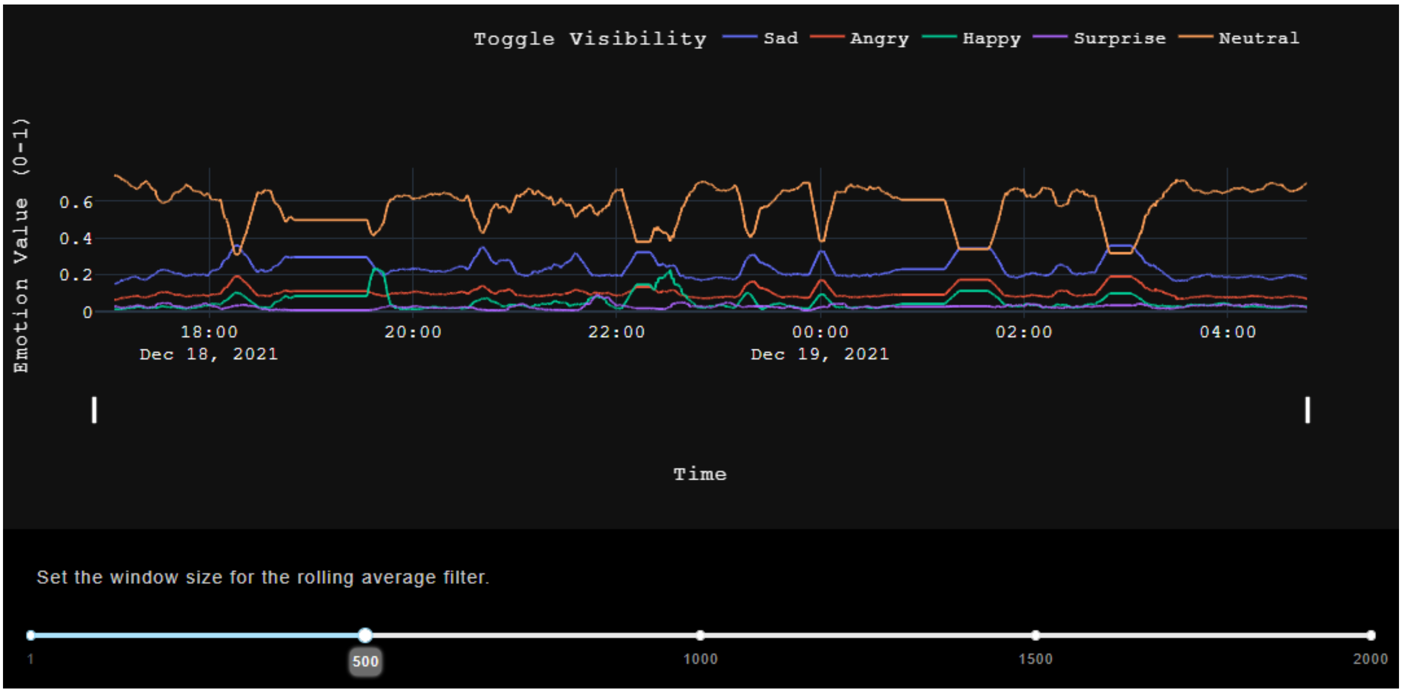


Appendix D7 – The graph has a built-in range slider to allow the user to focus on specific periods of time. The results provide great insight into my sleeping patterns. The CCT and the lux levels are sharp graphs indicating the state of bulbs as relatively binary, either on or off, and that I do not adjust brightness or CCT – it is a consistent 4200K and 30 lux which is in line with a typical bedroom. The lux drops down just before I sleep, which is in line with my behaviour of turning the main lights off and lamp on before bed.

APPENDIX E: EMOTION-BASED TIME SERIES DATA



Appendix E1 – Raw emotion data time series plot.



Appendix E2 – Plot of emotion predictions during a work session. Results show it remains relatively static at neutral but does detect sadness which peaks 3 times. The model tends to favour sadness predictions over all the other emotions, indicating a potential bias in the large dataset. There is also an interactive rolling average filter that can be changed live by the user on the website to see raw or filtered data (this works through a callback function).