

Pandemic Team report

March 2021

1 Challenge Solution Statement

Our prototype solution, Abor.io, has been designed to address the issues of company employees' mental and physical wellbeing by encouraging physical activity. Our system encourages simultaneous activity by giving rewards for joint workouts, encouraging sustainability in the company through a tree theme on the website. To achieve this, we designed two different devices that can communicate to a server and web page.

The primary device (held in a pocket) has a heart rate monitor, a pedometer and an LED which notifies the user of inactivity through a timer. The secondary device (worn on your wrist) has an accelerometer and a timer, used to measure various exercises. It also uses the on-board Hall effect sensor for other exercises.

The devices both use an ESP32 microcontroller, which has built-in Wi-Fi. These are powered using rechargeable lithium-ion batteries, reducing the waste from single-use batteries, and providing sustainability in our prototype. The batteries are regulated to 3.3V. LEDs will be used to show inactivity and the status of other users. If another user is working out, an LED will toggle to show this, and joint exercise will double the points reward.

The accelerometer sensor is used in both the primary and secondary device. Output voltages representing the directions (with a 50Hz bandwidth) are amplified, rectified and summed together to get a value for acceleration. Using the ADC on the ESP32, this result is read and analysed. On the primary device, its peaks are analysed to determine steps (as a pedometer), and on the secondary device it is used to measure arm movement for shadow boxing. The data is published as points to the server.

The ESP32 has an on-board Hall effect sensor, which is used to measure other exercises, such as sit ups. This will be done by detecting the proximity to another magnet on the other device. The peaks are analysed as repeated movements, and points will be published to the server.

The primary device will have a heart rate monitor on it too, which will gather data and interface with the microcontroller using I2C. On this sensor, the heart rate is sampled via an internal ADC, and then signal processing will be used to analyse this result. The information will be all analysed on the device, so no health data is transferred to the server.

An inactivity timer on the primary device will toggle an LED if the user hasn't done any movement for 30 minutes. The timer will also be used to time the activity duration, once movement has been detected on the accelerometer, and store this information as a string to aid in calculating points.

The server uses the AWS IoT core, using the MQTT pub/sub method for the data. JSON variables are used to send data to the cloud, getting data from each device which has been instantiated. Libraries provided in the Arduino IDE will be used to create a connection between the ESP32 and the internet, and AWS IoT libraries connect to the server. Once connected, data transfer can be published from the ESP32s and the website will subscribe to this, integrating all layers of devices, server and website together.

The website will be designed to have a tree page, with a tree avatar for an individual user which grows based on the points gained by a user. The company page will show every company user trees anonymously, creating a "garden". The donations page will show the money donated by a company, and the number of trees which should be donated, with links to tree charities. Every user will have access to the website.

2 System Design

The system has multiple users. Each user has a primary device and a secondary device. These devices interface with the AWS IoT broker via MQTT which then uses Node RED to host a website and parse JSON values to the website. Both devices contain sensors to determine the user's physical activity.

The primary device includes a pedometer, heartrate monitor and an inactivity timer. The pedometer is implemented via an accelerometer, analogue circuitry and digital processing. The heartrate monitor is connected directly to the microcontroller and outputs pulse rate data after digital processing. Timer inactivity detection is done via software by routinely checking for a new step and is configured to toggle an LED if no step was detected for a fixed amount of time.

The secondary device is implemented with a shadow boxing sensor and a hall effect sensor. The shadow boxing sensor is realized with an accelerometer at its core, analogue circuitry and adequate digital processing to reliably detect punches. The hall effect sensor uses magnetic field proximity to count repetitions of various exercises like sit ups, rope-climbs, and any other workout which requires the user to move from one place to another.

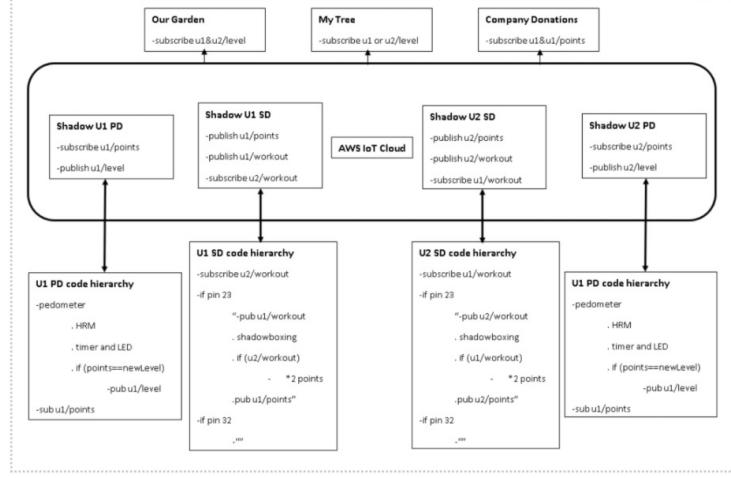


Figure 1: Full system diagram

The full system diagram is illustrated in 1. The users interact with the AWS IoT cloud via an MQTT communication protocol, which allows for full duplex, bidirectional communication between all clients (web pages and users). Our specification required user to user communication to realize the workout notification functionality on the secondary devices. We initially looked at potentially using HTTP POST/GET coupled with a database to realize our system. However, the half-duplex functionality of HTTP and the ease of data identification in MQTT (with the use of topics) encouraged the use of MQTT instead of HTTP. HTTP also did not allow for direct remote connectivity between users, hence reducing our ability to realize the workout notification function.

We investigated using multiple MQTT brokers, from mosquito and azure to google, however settled with AWS. This is due to the hornbill IoT open-source library which increases the ease of connectivity to AWS along with the added security that AWS allows (PEM encrypted certificates and TLS configuration). AWS IoT's MQTT broker supplied a reliable connection to the ESP32's (which could be enhanced by increasing the QoS) and it's simple pub/sub protocol allowed us to create a direct remote connection between all users and the website, greatly enhancing the potential for expansion of our product to accommodate more users.

3 Design Evaluation

The attempted design was fairly conservative in terms of the number and challenge of elements, and could have been more ambitious given the timeframe. This decision was consciously made in order to ensure that a fully working system could be demonstrated, but it may have been more effective to set more stretch goals for additional sensors etc to incorporate so that we could fully use

the time available. This may have also been partly a result of the division of labour, which was strongly weighted towards working on the hardware/firmware when in hindsight it was probably necessary to allocate at least one or two more people to the networking section of the project.

When designing the system, we thoroughly researched the different options of sensors available for each sub-system. However, we did not do as much research into the different options for the microcontroller (only selecting between two options, the ESP32 and the PSoC and choosing the ESP32). More research at this stage may have helped find something that better fit our requirements and made development easier. In the design spec for the firmware, there was a good definition of what data should be generated by each sensor-specific piece of code, but the exact format of that data and how functions should be called by the main program were less well defined resulting in some minor changes being required.

The primary device (the pedometer and heart rate monitor) was designed to go into the pocket, because we felt that it would be the easiest to use and therefore most likely to actually be used. It was also designed to perform its functions automatically when appropriate without any buttons for maximum ease-of-use. The secondary device is handheld and does use buttons to indicate workouts, but that is because it is focused on much more “deliberate” workouts, when the user is exercising for a defined time.

The designed product was aiming to increase the physical activeness of office workers. Since a key aim is encouraging sustainability, we have incorporated growing virtual trees in a virtual garden while office workers conduct their physical workout. By doing so, they are encouraging their company to donate directly to non-profit organizations to grow actual trees on Earth.

The aesthetic of the product is mild, considering it as a prototype. The product did not have any casing, and it only has the electronic components on the breadboard. The goal was to focus on the functionality and operability of the product instead of aesthetic properties. Moreover, the resources to improve the aesthetic of the product is not readily available. For example, a 3D printer to produce a casing for the product is not available.

The total cost to manufacture the two devices is £70. That includes the cost of electronic components, casing cost and assembly cost. The predicted selling price for the product (per pair) is £90. That will generate a £20 profit from the product. The generated profit of the product is a good amount for the company to achieve quick return on investment.

The devices have medium reliability when users use them. The reason for it is that sometimes the microcontroller in the devices could not connect to the server. The microcontroller needed to be regularly reset to counter that problem. When developing the prototype there were some networking issues arising from needing to use a trial version of AWS-IoT.

4 Ethics, Sustainability, Conformance Marking, and Marketing

Because we have designed a system which will monitor users' activity in their own homes, ethical considerations need to be examined. One of the largest problems was ensuring that the user's personal information was kept secure. We achieved this by making the trees shown on the "Our Garden" page anonymous. The website uses https and runs on a TLS configured network and each device has their own individual private pem keys and certificates when sending information from the devices to the server to increase security. In addition, any data measured by the devices is used to calculate the number of points the user has earned before being sent to the server. Therefore, only an arbitrary points value is stored on the server, rather than medical or personal data, such as resting heart rate, which the user might not want to disclose to their employer.

We had to ensure we took a sustainable design approach for our D4 project, both in terms of physical design choices and system behavior. We only considered components which are advertised as being power efficient, improving the battery life of each system and therefore having to recharge the system less frequently. We used rechargeable, lithium-ion batteries to eliminate electronic waste from conventional batteries (such as coin-cell). Alongside this, our fundamental project idea revolves around promoting sustainability in businesses, resulting in the planting of trees and donations towards sustainable charities. We also only used locally sourced components for our system, cutting down on shipping costs and not promoting air travel for the delivery of components. If we were to try and make the system more sustainable, we would use custom PCBs (produced in the UK), and source components from the same suppliers, removing increased shipping demand and adverse production effects on multiple components that could be manufactured together.

Before being allowed on the market, both devices will need to comply with requirements for CE or UKCA conformance marking for wearable fitness trackers. Currently, the circuits of both devices are built on breadboards, which are too large to fit easily on the user's wrist or in their pocket. However, the final product would be built on custom PCBs, which will be able to be made much smaller and therefore be comfortably worn. The wrist strap on the secondary device will also need to be designed to not cause discomfort or irritation when being used. The devices also won't have any exposed wires touching the user's skin, which will reduce the risk of causing an electric shock when being used. We also need to ensure that all medical information about our devices is correct and the reliability of the measurements made by the device, specifically the heart rate monitor, is disclosed.

We would be advertising our system specifically to businesses, who have been affected by the COVID-19 pandemic with many employees working from home. We would therefore design our advertisements around a specific business audience. We can do this by advertising at business conferences (non-specific),

both in person and online, trying to reach the largest audience as possible. These would start as specifically UK based, but could expand to European conferences in the future. Alongside this, we would use online advertising, specifically through LinkedIn, to once again aim at the largest possible audience in our target market. We will market our system as improving the sustainability of the company, whilst encouraging physical activity and collaboration between colleagues. Businesses will be able to themselves advertise how much they have been contributing to sustainable causes or growing trees while using the system, whilst also acting as another source of advertising for us.

5 Final Product

Our system has two LEDs; one to indicate inactivity for longer periods of time and one to notify if other users are exercising at the moment. Users are encouraged to start their workout after noticing one of those. To start a workout, the user needs to get BPM measured by holding his finger over the sensor for a couple of seconds. Once the BPM has been measured, both devices can be used for their respective measurements: primary device with pedometer, secondary device with accelerometer and hall effect sensor. After finishing the workout, user can measure heart rate again to calculate the points factor. The points are then transmitted to the server, where the user's tree grows proportionally to the number of points collected. This can be observed on the website. Users can also view the company's garden and see the number of points collected by their colleagues. There is also a donations page which is the total number of trees donated and money raised.

When comparing the primary device and secondary specifications versus what we achieved, most functionality is implemented successfully. Heart rate monitor, step detector and gyroscope which are the main components of the device work with a degree of success because at times it suffers from low accuracy. The points system, inactivity timer, digital processing, and transmits data at the end of workouts were implemented on both devices successfully with no issues. However, the timer for the duration of workouts was never implemented. Wearability of the device is not up to par as it cannot be placed in your pocket or worn on your wrist yet as it does not have a case to protect the devices and wiring.

The website and server side of the product achieved most if not all the objectives as the devices share information consistently. The website is displaying the growth of the trees and the garden of multiple people through the data from the devices. It also implements the company donations page which shows the total number of trees planted and money raised through working out without issues. However, the website layout is somewhat too simple.

Although our initial design is working correctly, there is much space for further improvements. Those include performance issues and additional features. Some of things we could work on:

- Designing custom casings and/or PCBs for current system.

- Handling situations when wireless connection is lost, for example storing the data to be sent later.
- Shortening the time required for heart rate monitor to collect and process the data by optimizing the algorithm.
- Adding support for other accelerometer-related activities other than shadow boxing, for example playing table tennis or jumping a rope.
- Further development of the website and server, adding features such as personal dashboard with workout statistics.
- Reducing power consumption of the devices by introducing power management techniques.
- Introducing more advanced signal processing techniques to improve the quality of sensor data.

6 Appendix A:

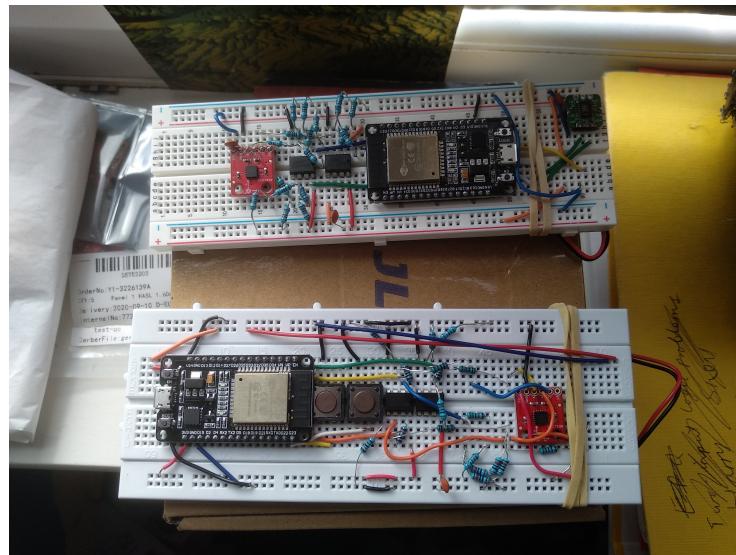
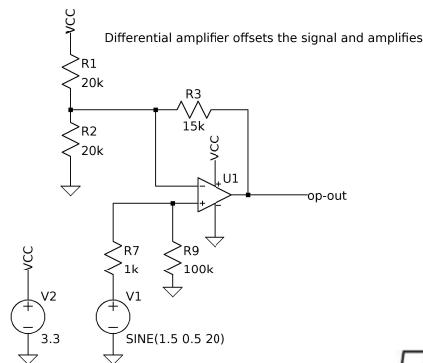
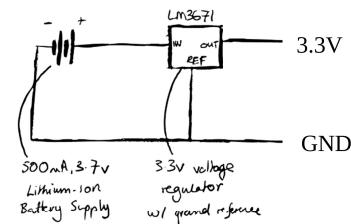


Figure 2: Primary (top) and Secondary device (bottom)

3x amplifier circuits,
one for each accelerometer channel



op-out-0 ————— ADC1_0
op-out-1 ————— ADC1_1
op-out-2 ————— ADC1_2



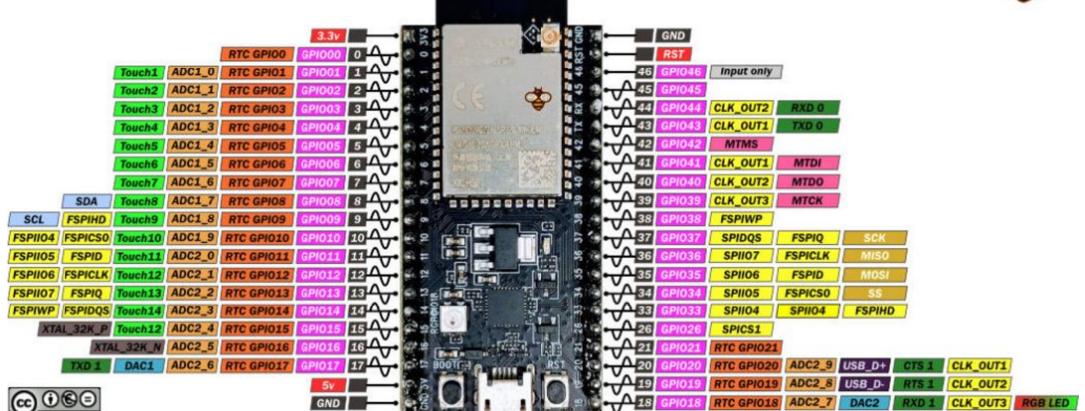
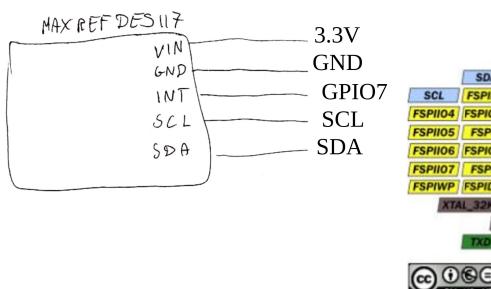
ESP32 S2 Saola 1MI PINOUT

www.mischianti.org (CC BY-NC-ND)



V1 used as a stand-in for the accelerometer output

.tran 0.2



7 Appendix B:

Code Listings

Listings

1	User 1 Primary device server Aduino file	10
2	User 1 Secondary device server Aduino file	13
3	User 2 Secondary device server Aduino file	16
4	Header file for credentials	19
5	Pedometer header file	19
6	Pedometer C file	20
7	Hall effect header file	24
8	Inactivity timer header file	25
9	Inactivity timer C file	25
10	Global Time header file	27
11	Global Time C file	28
12	Heart rate header file	28
13	Heart rate C file	29
14	Acclerometer for secondary device header file	31
15	WiFi connection header file	33
16	AWS header file	33
17	Final json file	34

U1PD.ino

```
1 #include <AWS_IOT.h>
2 #include <WiFi.h>
3 #include <ArduinoJson.h>
4 #include "wifiConn.h"
5 #include "awsPubSub.h"
6 #include "credentials.h"
7 #include "heartrate.h"
8 #include "Pedometer.h"
9 #include "localTimer.h"
10 #include "Globaltime.h"
11 #define SERIAL 1
12
13 AWS_IOT connection;
14
15
16 char HOST_ADDRESS []= "a2uci4q0y80ahc-ats.iot.eu-west-2.amazonaws.
17 com";
18 char CLIENT_ID []= "User1PD";
19 char SUB_TOPIC []= "User1SD/user1/points";
20 char PUB_TOPIC []= "User1PD/user1/level";
21 char PUB_TOPIC1 []= "User1PD/user1/points";
22
23 int status = WL_IDLE_STATUS;
24 int msgReceived=0;
25 char payload[512];
26 char reportpayload[512];
27 char rcvdPayload[512];
28
29 int points = 0;
30 int level = 0;
31
32 MAX30105 sensor;
33 MAX30105 *ptr;
34 int BPM;
35 float resting_bpm = 65;
36 float active_bpm = 65;
37 float factor = 1;
38 int *ptrBPM;
39 bool active = false;
40 bool old_active = false;
41
42 pedometer_struct pedometer_data;
43
44 int led = 2;
45
46 String time_str;
47
48 void mySubCallBackHandler (char *topicName, int payloadLen, char *
49 payLoad)
50 {
51     strncpy(rcvdPayload,payLoad,payloadLen);
52     rcvdPayload[payloadLen] = 0;
53     msgReceived = 1;
54 }
```

```

55 if(SERIAL) {
56     Serial.begin(115200);
57 }
58
59 // Setup for Heartrate monitor
60 ptr = &sensor;
61 ptrBPM = &BPM;
62 // while(!Serial);
63 init_sensor(ptr);
64
65 -----
66 //Setup timer and Pedometer
67 pinMode(led, OUTPUT);
68 if(SERIAL) Serial.println("start timer");
69 pedometer_data = init_pedometer_timer();
70 init_timer();
71 //digitalWrite(led, HIGH);
72 //delay(500);
73 //digitalWrite(led, LOW);
74 //connect to wifi
75 wifi_init(status, WIFI_SSID, WIFI_PASSWORD);
76 //digitalWrite(led, HIGH);
77 //delay(500);
78 //digitalWrite(led, LOW);
79 //connect to AWS and subscribe to topic
80 if(connection.connect(HOST_ADDRESS,CLIENT_ID)== 0) //Connect to
    AWS IoT Core
81 {
82     if(SERIAL) Serial.println("Connected to AWS");
83     delay(1000);
84
85     if(0==connection.subscribe(SUB_TOPIC, mySubCallBackHandler)) ///
        Subscribe to Accepted GET Shadow Service
86     {
87         if(SERIAL) Serial.println("Subscribe Successfull");
88     }
89     else
90     {
91         if(SERIAL) Serial.println("Subscribe Failed, Check the Thing
            Name and Certificates");
92         while(1);
93     }
94 }
95 else
96 {
97     if(SERIAL) Serial.println("AWS connection failed, Check the
            HOST Address");
98     while(1);
99 }
100 //digitalWrite(led, HIGH);
101 //delay(500);
102 //digitalWrite(led, LOW);
103 delay(300);
104
105 aws_pub(level, payload, connection, PUB_TOPIC);
106 digitalWrite(led, HIGH);
107 delay(500);

```

```

108     digitalWrite(led, LOW);
109 }
110
111 void loop() {
112     //digitalWrite(led, HIGH);
113     //delay(100);
114     //digitalWrite(led, LOW);
115
116     int init_level = level;
117     int init_points = points;
118     // Read heartrate monitor
119     if(read_bpm(ptr,ptrBPM)){
120         digitalWrite(led, HIGH);
121         while(read_bpm(ptr,ptrBPM) != 1);
122         digitalWrite(led, LOW);
123         if(!active){
124             resting_bpm = static_cast<float>(BPM);
125         } else{
126             active_bpm = static_cast<float>(BPM);
127             aws_pubPoints(points, payload, connection, PUB_TOPIC1);
128         }
129         factor = (active_bpm/resting_bpm < 1) ? 1.0 : active_bpm/
130             resting_bpm;
131         if(SERIAL) Serial.print("Heart rate ratio = ");
132         if(SERIAL) Serial.println(factor);
133     }
134     // -----
135     //Pedometer:
136     pedometer_data = detect_step_and_count(pedometer_data, 34, 35,
137     32);
138     if(pedometer_data.step_detected){
139         if(SERIAL) Serial.printf("\nSteps incremented , steps = %d\n",
140             pedometer_data.steps);
141     }
142
143     old_active = active;
144     check_steps(pedometer_data.step_detected);
145     if(pedometer_data.step_detected){
146         active = true;
147     }
148     if(!inactivity()){
149         active = false;
150     }
151     if(!active && old_active){
152         if(SERIAL){
153             Serial.print("activity finished. Points: ");
154             Serial.println(pedometer_data.total_points);
155             Serial.print("adjusted for BPM: ");
156             Serial.println(factor*pedometer_data.total_points);
157         }
158         points += factor*pedometer_data.total_points;
159         if(SERIAL) {
160             Serial.print("Total points: ");
161             Serial.println(points);
162             Serial.println("User ID: 000001");
163         }
164     }
165     // global time

```

```

162     time_str = get_time(WIFI_SSID, WIFI_PASSWORD);
163     if(SERIAL) {
164         Serial.print("Time and date is: ");
165         Serial.println(time_str);
166     }
167     aws_pubPoints(points, payload, connection, PUB_TOPIC1);
168 }
169
170 if(msgReceived == 1) {
171     int received_points = aws_sub(rcvdPayload);
172     msgReceived = 0;
173     // while(read_bpm(ptr,ptrBPM) != 1);
174     // active_bpm = static_cast<float>(BPM);
175     // factor = (active_bpm/resting_bpm < 1) ? 1.0 : active_bpm/
176     // resting_bpm;
177     points += received_points*factor;
178     if(SERIAL) {
179         Serial.print("Current Points: ");
180         Serial.println(points);
181     }
182     aws_pubPoints(points-init_points, payload, connection,
183     PUB_TOPIC1);
184
185     (points < 100) ? level = 0 :
186     ((points >= 100 && points < 300) ? level = 1 :
187     ((points >= 300 && points < 600) ? level = 2 :
188     ((points >= 600 && points < 1000) ? level = 3 :
189     ((points >= 1000 && points < 1500) ? level = 4 :
190     ((points >= 1500 && points < 2100) ? level = 5 :
191     level = 6)));
192
193     if(init_level != level) {
194         if(SERIAL){
195             Serial.print("init_level: ");
196             Serial.println(init_level);
197             Serial.print("level: ");
198             Serial.println(level);
199         }
200         aws_pub(level, payload, connection, PUB_TOPIC);
201         aws_pubPoints(points-init_points, payload, connection,
202         PUB_TOPIC1);
203     }
204     delay(5);
205 }
```

Listing 1: User 1 Primary device server Aduino file

U1SD.ino

```

1 #include <AWS_IOT.h>
2 #include <WiFi.h>
3 #include <ArduinoJson.h>
4 #include "wifiConn.h"
5 #include "awsPubSub.h"
6 #include "hall.h"
7 #include "credentials.h"
```

```

8 #include "accel.h"
9 #define SERIAL 1
10
11 AWS_IOT connection;
12 Accelerometer acc;
13
14 char HOST_ADDRESS []= "a2uci4q0y80ahc-ats.iot.eu-west-2.amazonaws.
15 com"; //AWS endpoint
16 char CLIENT_ID []= "User1SD";
17 char PUB_TOPIC1 []= "User1SD/user1/workout"; //topic used to
18 publish to workout
19 char PUB_TOPIC2 []= "User1SD/user1/points"; //topic used to
20 publish to points
21 char SUB_TOPIC []= "User2SD/user2/workout"; //topic used to
22 subscribe to workout on user 2
23
24 int status = WL_IDLE_STATUS;
25 int msgReceived=0;
26 char payload[512];
27 char reportpayload[512];
28 char rcvdPayload[512];
29
30 int points = 0;
31 int count = 0;
32 int workout_sub = 0; //User 2's workout status (if 0 -> not working
33 out, if 1 -> working out)
34 int workout_pub = 0; //User 1's (this user) workout status (if 0 ->
35 not working out, if 1 -> working out)
36
37 const int situp_pin = 23;
38 const int boxing_pin = 22;
39 const int led = 2;
40
41 void mySubCallBackHandler (char *topicName, int payloadLen, char *
42 payLoad)
43 {
44     strncpy(rcvdPayload,payLoad,payloadLen);
45     rcvdPayload[payloadLen] = 0;
46     msgReceived = 1;
47 }
48
49 void setup() {
50     if(SERIAL){
51         Serial.begin(115200);
52     }
53
54     pinMode(situp_pin, INPUT_PULLUP);
55     pinMode(boxing_pin, INPUT_PULLUP);
56     pinMode(led, OUTPUT);
57
58     //connect to wifi
59     wifi_init(status, WIFI_SSID, WIFI_PASSWORD);
60
61     //connect to AWS and subscribe to topic
62     if(connection.connect(HOST_ADDRESS,CLIENT_ID)== 0) //Connect to
63     AWS IoT C0re
64     {

```

```

57   if(SERIAL) Serial.println("Connected to AWS");
58   delay(1000);
59
60   if(0==connection.subscribe(SUB_TOPIC, mySubCallBackHandler)) // 
61     Subscribe to user 2's workout status
62   {
63     if(SERIAL) Serial.println("Subscribe Successfull");
64   }
65   else
66   {
67     if(SERIAL) Serial.println("Subscribe Failed, Check the Thing
68       Name and Certificates");
69     while(1);
70   }
71
72 else
73 {
74   if(SERIAL) Serial.println("AWS connection failed, Check the
75     HOST Address");
76   while(1);
77 }
78
79 void loop() {
80   bool inWorkoutSitUps = false;
81   bool inWorkoutBoxing = false;
82
83   //get user 2's workout status
84   if(msgReceived == 1) {
85     msgReceived = 0;
86     workout_sub = aws_sub(rcvdPayload);
87     digitalWrite(led, workout_sub);
88   }
89
90   if(!digitalRead(situp_pin)) {
91     if(SERIAL) Serial.println("Starting situps");
92     //publish user 1 is working out to user 2 device
93     workout_pub = 1;
94     aws_pubWorkout(workout_pub, payload, connection, PUB_TOPIC1);
95     //publish workout status of this user (user 1)
96
97     inWorkoutSitUps = true;
98     while(inWorkoutSitUps) {
99       if(msgReceived == 1) {
100         msgReceived = 0;
101         workout_sub = aws_sub(rcvdPayload);
102         digitalWrite(led, workout_sub);
103       }
104       if(!digitalRead(situp_pin)) {
105         //publish user 1 is no longer working out to user 2 device
106         workout_pub = 0;
107         aws_pubWorkout(workout_pub, payload, connection, PUB_TOPIC1
108 ); //publish workout status of this user (user 1)
109
110         aws_pubPoints(points, payload, connection, PUB_TOPIC2);

```

```

109
110     inWorkoutSitUps = false; //leave while loop
111 } else {
112     //integrate Hall Effect Sensor
113     int initialCount = count;
114     count = getCount(count);
115
116     //required to ensure points dont keep increasing while
117     //count % 10 = 0 for more than 1 cycle of loop
118     if(initialCount % 10 != 0) {
119         points = pointsReturn(count, points, workout_sub);
120     }
121     char output[512];
122     sprintf(output,"Points: %d Count: %d\n", points, count);
123     if(SERIAL) Serial.print(output);
124 }
125
126
127 if(!digitalRead(boxing_pin)){
128     if(SERIAL) Serial.println("Starting boxing");
129     //publish user 1 is working out to user 2 device
130     workout_pub = 1;
131     aws_pubWorkout(workout_pub, payload, connection, PUB_TOPIC1);
132     //publish workout status of this user (user 1)
133
134     inWorkoutBoxing = true;
135     //Accelerometer code goes here
136     while(inWorkoutBoxing){
137         if(msgReceived == 1) {
138             msgReceived = 0;
139             workout_sub = aws_sub(rcvdPayload);
140             digitalWrite(led, workout_sub);
141         }
142         if(!digitalRead(boxing_pin)) {
143             //publish user 1 is no longer working out to user 2 device
144             workout_pub = 0;
145             aws_pubWorkout(workout_pub, payload, connection, PUB_TOPIC1);
146             //publish workout status of this user (user 1)
147             aws_pubPoints(points, payload, connection, PUB_TOPIC2);
148
149             inWorkoutBoxing = false; //leave while loop
150         } else{
151             points = acc.readAcc(points, workout_sub);
152             // if(SERIAL) Serial.println(points);
153         }
154     }
155
156     count = 0; //reset count on SD
157     points = 0; //reset points on SD
158 }
```

Listing 2: User 1 Secondary device server Aduino file

U2SD.ino

```

1 #include <AWS_IOT.h>
2 #include <WiFi.h>
3 #include <ArduinoJson.h>
4 #include "wifiConn.h"
5 #include "awsPubSub.h"
6 #include "hall.h"
7
8 AWS_IOT connection;
9
10 char WIFI_SSID[] = "VM7251536"; //Replace with your WiFi Name
11 char WIFI_PASSWORD[] = "dmc9hsnvPyKv"; // Replace with your WiFi
12     Password
13 char HOST_ADDRESS []= "a2uci4q0y80ahc-ats.iot.eu-west-2.amazonaws.
14     com"; //AWS endpoint
15 char CLIENT_ID []= "User2SD";
16 char PUB_TOPIC1 []= "User2SD/user2/workout"; //topic used to
17     publish to user 2's workout status
18 char PUB_TOPIC2 []= "User2PD/user2/points"; //topic used to
19     publish to user 2's points
20 char PUB_TOPIC3 []= "User2PD/user2/level"; //topic used to
21     publish to user 2's level
22 char SUB_TOPIC []= "User1SD/user1/workout"; //topic used to
23     subscribe to workout status on user 1
24
25 int status = WL_IDLE_STATUS;
26 int msgReceived=0;
27 char payload[512];
28 char reportpayload[512];
29 char rcvdPayload[512];
30
31 int points = 0;
32 int count = 0;
33 int level = 0;
34 int workout_sub = 0; //User 1's workout status (if 0 -> not working
35     out, if 1 -> working out)
36 int workout_pub = 0; //User 2's (this user) workout status (if 0 ->
37     not working out, if 1 -> working out)
38
39 const int PUSHBUTTONPIN = 23;
40
41 void mySubCallBackHandler (char *topicName, int payloadLen, char *
42     payLoad)
43 {
44     strncpy(rcvdPayload,payLoad,payloadLen);
45     rcvdPayload[payloadLen] = 0;
46     msgReceived = 1;
47 }
48
49 void setup() {
50     Serial.begin(115200);
51     pinMode(PUSHBUTTONPIN, INPUT);
52
53     //connect to wifi
54     wifi_init(status, WIFI_SSID, WIFI_PASSWORD);
55
56     //connect to AWS and subscribe to topic
57     if(connection.connect(HOST_ADDRESS,CLIENT_ID)== 0) //Connect to

```

```

AWS IoT C0re
{
    Serial.println("Connected to AWS");
    delay(1000);

    if(0==connection.subscribe(SUB_TOPIC, mySubCallBackHandler)) // 
    Subscribe to Accepted GET Shadow Service
    {
        Serial.println("Subscribe Successfull");
    }
    else
    {
        Serial.println("Subscribe Failed, Check the Thing Name and
Certificates");
        while(1);
    }
}
else
{
    Serial.println("AWS connection failed, Check the HOST Address")
;
    while(1);
}

delay(300);

aws_pubLevel(level, payload, connection, PUB_TOPIC3);
}

void loop() {
    bool inWorkoutSitUps = false;
    int init_level = level;

    //get user 1's workout status
    if(msgReceived == 1) {
        msgReceived = 0;
        workout_sub = aws_sub(rcvdPayload);
    }

    if(digitalRead(PUSHBUTTONPIN) == HIGH) {
        //publish user 1 is working out to user 2 device
        workout_pub = 1;
        aws_pubWorkout(workout_pub, payload, connection, PUB_TOPIC1);
        //publish workout status of this user (user 1)

        inWorkoutSitUps = true;
        while(inWorkoutSitUps) {
            //get user 1's workout status
            if(msgReceived == 1) {
                msgReceived = 0;
                workout_sub = aws_sub(rcvdPayload);
            }

            if(digitalRead(PUSHBUTTONPIN) == HIGH) {
                //publish user 1 is no longer working out to user 2 device
                workout_pub = 0;
                aws_pubWorkout(workout_pub, payload, connection, PUB_TOPIC1

```

```

101 ); //publish workout status of this user (user 1)
102 aws_pubPoints(points, payload, connection, PUB_TOPIC2);
103 //publish level here
104
105 (points < 100) ? level = 0 :
106   ((points >= 100 && points < 300) ? level = 1 :
107     ((points >= 300 && points < 600) ? level = 2 :
108       ((points >= 600 && points < 1000) ? level = 3 :
109         ((points >= 1000 && points < 1500) ? level = 4 :
110           ((points >= 1500 && points < 2100) ? level = 5 :
111             level = 6))))));
112
113   Serial.print("init_level: ");
114   Serial.println(init_level);
115   Serial.print("level: ");
116   Serial.println(level);
117
118   if(init_level != level)
119     aws_pubLevel(level, payload, connection, PUB_TOPIC3);
120
121   inWorkoutSitUps = false; //leave while loop
122 } else {
123   //integrate Hall Effect Sensor
124   int initialCount = count;
125   count = getCount(count);
126
127   //required to ensure points dont keep increasing while
128   //count % 10 = 0 for more than 1 cycle of loop
129   if(initialCount % 10 != 0) {
130     points = pointsReturn(count, points);
131   }
132   char output[512];
133   sprintf(output,"Points: %d Count: %d\n", points, count);
134   Serial.print(output);
135 }
136
137 }

```

Listing 3: User 2 Secondary device server Aduino file

example_credentials.h

```

1 #define SSID "YOUR_SSID"
2 #define PASSWORD "YOUR_PW"

```

Listing 4: Header file for credentials

pedometer.h

```

1
2 #include "arduino.h"
3
4
5 #ifndef threshold2_H
6 #define threshold2_H
7

```

```

8 #include <stdio.h>
9 #include <stdbool.h>
10
11
12 #define ELEMENT_COUNT(X) (sizeof(X) / sizeof((X)[0]))
13
14 #define filter_len 22
15
16 // Pedometer structure
17 typedef struct {
18     unsigned long steps ;
19     int total_points;
20     bool step_detected;
21 } pedometer_struct;
22
23
24
25 extern volatile bool step_counter_interrupt_flag; // global
26     variable
27
28 // Function definitions
29 pedometer_struct detect_step_and_count(pedometer_struct, int x_pin,
30                                         int y_pin, int z_pin);
31 void IRAM_ATTR set_pedometer_flag();
32 void shift_left_and_update(float buffer[], int Length, float
33                             new_value);
34 float get_filtered_value(float buffer[], int buffer_len);
35 pedometer_struct init_pedometer_timer();
36
37
38
39 #endif

```

Listing 5: Pedometer header file

pedometer.cpp

```

1
2 #include "Pedometer.h"
3
4
5
6 const int led = 2;
7
8 volatile bool step_counter_interrupt_flag;
9 // filter coefficient values
10 float filter[] = {
11     -2.057870e-03,
12     -3.375331e-03,
13     -5.446070e-03,
14     -6.347726e-03,
15     -2.294223e-03,
16     1.082595e-02,
17     3.518455e-02,
18     6.913013e-02,
19     1.067513e-01,
20     1.393380e-01,
21     1.582914e-01,
22

```

```

23   1.582914e-01,
24   1.393380e-01,
25   1.067513e-01,
26   6.913013e-02,
27   3.518455e-02,
28   1.082595e-02,
29   -2.294223e-03,
30   -6.347726e-03,
31   -5.446070e-03,
32   -3.375331e-03,
33   -2.057870e-03
34 };
35
36
37
38 void IRAM_ATTR set_pedometer_flag() {
39
40     step_counter_interrupt_flag = true;
41 }
42
43
44
45 pedometer_struct init_pedometer_timer()
46 {
47     hw_timer_t * timer = NULL;
48
49     pedometer_struct pedometer;
50
51     pedometer.steps =0;
52     pedometer.total_points = 0;
53     pedometer.step_detected = false;
54
55     float clock_freq = 240;//in Mhz
56     float clock_divider = 80;
57     float timer_freq = 25;
58     float ratio = (float) (clock_freq / (clock_divider * timer_freq))
59     ;
60
61     unsigned long int max_tick = 40000; // for 25 Hz
62
63     // Serial.println(max_tick);
64
65     timer = timerBegin(1, clock_divider, true);
66
67     /* Attach onTimer function to our timer */
68     timerAttachInterrupt(timer, &set_pedometer_flag, true);
69
70     /* Set alarm to call onTimer function every second 1 tick is 1us
71      => 1 second is 1000000us */
72     /* Repeat the alarm (third parameter) */
73
74     timerAlarmWrite(timer, max_tick, true);
75
76     /* Start an alarm */
77     timerAlarmEnable(timer);
78
79
80     return pedometer;

```

```

79 }
80 }
81
82
83 void shift_left_and_update(float buffer[], int Length, float
84 new_value)
84 { // adds a new value to a rolling buffer and shifts the content
85 left
85 for (int i = 0; i < Length - 1; i++)
86 {
87     buffer[i] = buffer[i + 1];
88 }
89 buffer[Length - 1] = new_value;
90 }
91
92 float get_filtered_value(float buffer[], int buffer_len)
93 {
94     //buffer len should be equal to filter len.
95     float out_sum = 0;
96     for (int i = 0; i < buffer_len; i++)
97     {
98         out_sum += buffer[buffer_len - i] * filter[i];
99     }
100    return out_sum; // returns the filtered magnitude value.
101 }
102
103 // main function to detect step and increment step value.
104 pedometer_struct detect_step_and_count(pedometer_struct
104 Old_pedometer, int x_pin, int y_pin, int z_pin)
105 {
106
107 pedometer_struct pedometer;
108 pedometer = Old_pedometer;
109
110 if (!step_counter_interrupt_flag)
111 {
112     pedometer.step_detected = false;
113     return pedometer;
114 }
115 else
116     step_counter_interrupt_flag = false;
117     //digitalWrite(led, HIGH);
118     float time = 0;
119     static float old_time = 0;
120     static int counter = 0;
121     counter++;
122     bool basic_serial_debug = false;
123     bool advanced_serial_debug = false;
124     //Hysteresis algorithm thresholds
125     float lower_threshold = 1.85; // 0.338 volts value for without
125     analog filtering
126     float upper_threshold = 1.94;
127     // 0.358 volts value for without analog filtering
128     enum state {Lower, Middle, Upper};// Hysteresis algorithm
states
129
130     static enum state signalState = Lower;

```

```

131
132     bool step_incremented = false;
133
134     float x_value, y_value, z_value;
135     float current_voltage = 3.0;
136     float offset = 0; // // 1.52  volts value for without analog
filtering
137
138     float filtered_voltge;
139     const int buffer_len = filter_len;
140     static float buffer[filter_len];
141     // Converts raw values to voltages.
142     x_value = 3.3 * (float)analogRead(x_pin) / 4096.0;
143     y_value = 3.3 * (float)analogRead(y_pin) / 4096.0;
144     z_value = 3.3 * (float)analogRead(z_pin) / 4096.0;
145
146     /*
147     Serial.print("x value = ");
148     Serial.print(x_value);
149     Serial.print(",y value = ");
150     Serial.print(y_value);
151     Serial.print(",z value = ");
152     Serial.println(z_value);
153     */
154
155
156     // Calculates magnitude after removing offset from the X,Y and Z
values.
157     current_voltage = sqrt((x_value - offset) * (x_value - offset)
+ (y_value - offset) * (y_value - offset) + (z_value - offset)
* (z_value - offset));
158
159     shift_left_and_update(buffer, buffer_len, current_voltage);
160     filtered_voltge = get_filtered_value(buffer, buffer_len);
//printf("%f,\n",filtered_voltge);
161
162     /*
163     Serial.print(counter);
164     Serial.print(",");
165     // Serial.print(filtered_voltge);
166     // Serial.println(",");
167
168
169
170
171     // Beginning of Hysteresis state machine.
172     if (signalState == Lower)
173     {
174         if (filtered_voltge > lower_threshold)
175         {
176             signalState = Middle;
177             pedometer.step_detected = false;
178         }
179     }
180     else if (signalState == Middle)
181     {
182         if (filtered_voltge > upper_threshold)

```

```

184 {
185     signalState = Upper;
186
187
188     time = (float) counter / 25.0;
189     // Serial.println("step incremented");
190     old_time = time;
191     pedometer.step_detected = true;
192     // Serial.println("Step detected");
193     pedometer.steps++;
194     if(pedometer.steps % 5 == 0)
195     {
196         pedometer.total_points++;
197         // Serial.println("points incremented");
198     }
199     // step_incremented = true;
200 }
201
202 else if (filtered_volatile < lower_threshold)
203 {
204     signalState = Lower;
205     pedometer.step_detected = false;
206 }
207 }
208
209 else if (signalState == Upper)
210 {
211     if (filtered_volatile < upper_threshold)
212     {
213         signalState = Middle;
214     }
215 }
216
217 }
218 // End of Hysteresis state machine.
219
220
221
222
223 return pedometer;
224
225
226
227 }
```

Listing 6: Pedometer C file

hall.h

```

1 int getCount(int count)
2 {
3     int val = hallRead();
4     Serial.println(val);
5     delay(1000);
6
7     if(val > 40 || val < 0) {
8         count++;
9     }
```

```

10     return count;
11 }
12
13 int pointsReturn(int count, int points)
14 {
15     if((count % 10 == 0) && count != 0) {
16         points += 10;
17     }
18
19     return points;
20 }
21

```

Listing 7: Hall effect header file

localTimer.h

```

1 #include <Arduino.h>
2
3 extern volatile bool Timer_interrupt_flag;
4 void IRAM_ATTR onTimer0();
5 void toggleLed(bool ledToggleStatus);
6 void init_timer();
7 void reset_timer0_method1();
8 void check_steps(bool steps_detected);
9 bool inactivity();

```

Listing 8: Inactivity timer header file

localTimer.cpp

```

1 #include "localTimer.h"
2
3
4
5 #define LED_ALARM    2
6
7
8 hw_timer_t *timer1 = NULL;
9
10
11
12 volatile bool Timer_interrupt_flag = false;
13
14 bool inactivity(){
15     bool active = true;
16     static bool ledToggleMode = false;
17     toggleLed(ledToggleMode);
18
19     static double lastTime = 0.0;
20     if(Timer_interrupt_flag)
21     {
22         Serial.println("User has been inactive for 30 seconds");
23         active = false;
24         Timer_interrupt_flag = false;
25         // Serial.println("Timer_interrupt_flag is set to false");
26         //digitalWrite(LED_ALARM, HIGH);
27         ledToggleMode = true;
28     }

```

```

29     lastTime = timerReadSeconds(timer1);
30
31 }
32 else
33 {
34     if(timerReadSeconds(timer1) - lastTime >= 1)// toggles led
35     for one second if user has been inactive for 5 seconds
36     {
37         ledToggleMode = false;
38         digitalWrite(LED_ALARM,LOW);
39         lastTime = timerReadSeconds(timer1);
40
41     }
42     if(timerReadSeconds(timer1) <= 0.2)
43     {
44         lastTime = timerReadSeconds(timer1);
45     }
46     //digitalWrite(LED_ALARM, LOW);
47 }
48 return active;
49 }

50 void check_steps(bool steps_detected){
51     if(steps_detected)
52     {
53         reset_timer0_method1();
54     }
55 }
56 }
57 }

58 void IRAM_ATTR onTimer0(){
59     // Increment the counter and set the time of ISR
60
61
62
63
64     Timer_interrupt_flag = true;
65     //Serial.println("Timer_interrupt_flag is true");
66
67 }
68 }

69

70 void toggleLed(bool ledToggleStatus)
71 {
72     //Serial.printf("\ntime now = %lf",timerReadSeconds(timer1));
73     static double localTimeSeconds = 0.0;
74     static bool ledState = false;
75     double ledDelay = 0.2;
76     // Serial.println(ledToggleStatus);
77
78     if(timerReadSeconds(timer1) - localTimeSeconds >= 0.1)
79     {
80
81         if(ledToggleStatus)
82         {
83             // Serial.println("led state toggled");

```

```

85     ledState = !ledState;
86     digitalWrite(LED_ALARM, ledState);
87
88 }
89
90     localTimeSeconds = (double)timerReadSeconds(timer1);
91 }
92
93 if(timerReadSeconds(timer1) <= 0.2)
94 {
95     localTimeSeconds = (double)timerReadSeconds(timer1);
96 }
97
98
99
100 }
101
102
103 void init_timer()
104 {
105     pinMode(LED_ALARM, OUTPUT);
106     timer1 = timerBegin(0, 80, true);
107     timerAttachInterrupt(timer1, &onTimer0, true); // 1 tick = 1 usec
108     timerAlarmWrite(timer1, 10000000, true); // set this values to 30
109         min - currently 10 sec
110     timerAlarmEnable(timer1);
111     // Serial.println("local timer initialized");
112 }
113
114 void reset_timer0_method1()
115 {
116     // Serial.println("restarting timer using method1");
117     timerWrite(timer1, 0); // restarting timer
118     init_timer();
119 }
120
121 void reset_timer0_method2() // this method needs to reinitialise the
122     timer again after restarting
123 {
124     // Serial.println("restarting timer");
125     timerAlarmDisable(timer1); // stop alarm
126     timerDetachInterrupt(timer1); // detach interrupt
127     timerEnd(timer1); // end timer
128     init_timer();
129 }
```

Listing 9: Inactivity timer C file

GlobalTime.h

```

1
2 #include <WiFi.h>
3 #include <time.h>
4 #include "arduino.h"
5
6 String getLocalTime();
```

```
7 String get_time(const char* ssid, const char* password);
```

Listing 10: Global Time header file

GlobalTime.cpp

```
1 #include "Globaltime.h"
2
3 const char* ntpServer = "uk.pool.ntp.org";
4
5 String getLocalTime()
6 {
7     String time_str;
8     struct tm timeinfo;
9     if(!getLocalTime(&timeinfo)){
10         //Serial.println("Failed to obtain time");
11         return("");
12     }
13     char output[80];
14     strftime(output,80,"%D/%H:%M:%S",&timeinfo);
15     time_str = String(output);
16     return(time_str);
17 }
18
19 String get_time(const char* ssid, const char* password)
20 {
21     String time_str = "";
22     volatile bool time_tries = false;
23     //Serial.printf("Connecting to %s ", ssid);
24     //init and get the time
25     while(time_str==""){
26         configTime(0, 0, ntpServer);
27         time_str = getLocalTime();
28     }
29
30     return(time_str);
31 }
```

Listing 11: Global Time C file

heartrate.h

```
1 #include <Wire.h>
2 #include "MAX30105.h"
3
4 const int DETECTION_THRESHOLD = 50000;
5 const int BUFFER_SIZE = 400;
6 const int LED_BRIGHTNESS = 70;
7 const int SAMPLE_AVERAGE = 1;
8 const int LED_MODE = 2;
9 const int SAMPLE_RATE = 100;
10 const int PULSE_WIDTH = 69;
11 const int ADC_RANGE = 16384;
12 const int AVERAGING = 3;
13
14 int find_period (int buffer[BUFFER_SIZE]);
15 void mean_center (int buffer[BUFFER_SIZE]);
16 long int autocorrelation(int lag ,int size, int *function );
17 int calculate_bpm(int period);
```

```

18 void init_sensor(MAX30105 *sensor);
19 int read_bpm(MAX30105 *sensor, int *bpm);
20 float calculate_factor(int restBPM, int finalBPM);

```

Listing 12: Heart rate header file

heartrate.cpp

```

1 #include "heartrate.h"
2
3
4 //return period of input signal based on autocorrelation function
5 int find_period (int buffer[BUFFER_SIZE])
6 {
7     int period = 0;
8
9     //find first local minimum of autocorrelation function
10    while((autocorrelation(period+1, BUFFER_SIZE, buffer)<
11           autocorrelation(period, BUFFER_SIZE, buffer)) ||
12          (autocorrelation(period+2, BUFFER_SIZE, buffer)<
13           autocorrelation(period, BUFFER_SIZE, buffer)))
14    {
15        period++;
16    }
17
18    //find first local maximum of autocorrelation function
19    while((autocorrelation(period+1, BUFFER_SIZE, buffer)>
20           autocorrelation(period, BUFFER_SIZE, buffer)) ||
21          (autocorrelation(period+2, BUFFER_SIZE, buffer)>
22           autocorrelation(period, BUFFER_SIZE, buffer)))
23    {
24        period++;
25    }
26
27    return period;
28 }
29
30
31 //subtracts mean of the signal from every record in the buffer
32 void mean_center (int buffer[BUFFER_SIZE])
33 {
34     long int sum = 0;
35     int mean;
36
37     //calculate mean
38     for(int i=0; i<=(BUFFER_SIZE-1); i++)
39     {
40         sum += buffer[i];
41     }
42     mean = sum/BUFFER_SIZE;
43
44     //subtract mean
45     for(int i=0; i<=(BUFFER_SIZE-1); i++)
46     {
47         buffer[i]-= mean;
48     }
49 }
50
51 //returns the value of normalised autocorrelation function for a
52 //given lag

```

```

47 long int autocorrelation(int lag ,int size , int *function )
48 {
49     long int sum = 0;
50     int iter = size - lag;
51     for(int i=0; i<iter; i++)
52     {
53         sum += function[i]*function[i+lag];
54     }
55     return sum/iter;
56 }
57
58 int calculate_bpm(int period)
59 {
60     return(60*SAMPLE_RATE/period);
61 }
62
63 void init_sensor(MAX30105 *sensor)
64 {
65     sensor->begin(Wire, I2C_SPEED_FAST);
66     sensor->setup(LED_BRIGHTNESS, SAMPLE_AVERAGE, LED_MODE,
67                     SAMPLE_RATE, PULSE_WIDTH, ADC_RANGE);
68 }
69
70 int read_bpm(MAX30105 *sensor, int *bpm)
71 {
72     static int buffer[BUFFER_SIZE] = {0};
73     static int bufferIndex = 0;
74     static int counter = 0;
75     int temp;
76     static int avg = 0;
77     int rvalue = 0;
78
79     //check status of the sensor
80     sensor->check();
81     if(sensor->available())
82     {
83         //discard the result if less than a given threshold (no finger)
84         if(sensor->getFIFORed() < DETECTION_THRESHOLD)
85         {
86             sensor->nextSample();
87             return 0;
88         }
89
90         //read and store the data in a buffer
91         buffer[bufferIndex] = sensor->getFIFORed();
92         bufferIndex++;
93         sensor->nextSample();
94
95         //once buffer is full, process the data
96         if(bufferIndex == BUFFER_SIZE)
97         {
98             //apply signal processing algorithms
99             mean_center(buffer);
100            temp = calculate_bpm(find_period(buffer));
101
102

```

```

103     //discard faulty data
104     if(temp > 40 && temp <200)
105     {
106
107         Serial.println(temp);
108         rvalue = 2;
109         avg += temp;
110         counter++;
111     }
112     bufferIndex = 0;
113 }
114
115 //once a given amount of records has been registered, calculate
116 //the average
117 if(counter == AVERAGING)
118 {
119     *bpm = avg/AVERAGING;
120     Serial.println("average: ");
121     Serial.println(*bpm);
122     counter = 0;
123     avg = 0;
124     rvalue = 1;
125 }
126 return rvalue;
127
128 }
129
130 float calculate_factor(int restBPM, int finalBPM)
131 {
132     return finalBPM/restBPM;
133 }
```

Listing 13: Heart rate C file

```

accel.h

1 #ifndef ACCEL_H
2 #define ACCEL_H
3
4 #define ADC1_4 32
5 #define ADC1_6 34
6 #define ADC1_7 35
7
8 #define X_STABLE 2400
9 #define Y_STABLE 1100
10 #define Z_STABLE 1500
11
12 #define X_BUFFER 1000
13 #define Y_BUFFER 750
14 #define Z_BUFFER 750
15
16 class Accelerometer{
17     private:
18         bool xmov, ymov, zmov;
19         int count;
20     public:
21         Accelerometer();
```

```

22     int readAcc(int points, int group_factor);
23
24 };
25
26 Accelerometer::Accelerometer(){
27     xmov = false;
28     ymov = false;
29     zmov = false;
30     count = 0;
31 }
32
33 int Accelerometer::readAcc(int points, int group_factor){
34     int x = analogRead(ADC1_4);
35     int y = analogRead(ADC1_7);
36     int z = analogRead(ADC1_6);
37     // Serial.printf("%d %d %d\n", x, y, z);
38
39     bool stopped = (x >= X_STABLE-100 && x <= X_STABLE+100) && (y >=
40                     Y_STABLE-100 && y <= Y_STABLE+100) && (z >= Z_STABLE-50 && z <=
41                     Z_STABLE+50);
42
43     if ((xmov||ymov||zmov) && stopped){
44         xmov = false;
45         ymov = false;
46         zmov = false;
47
48         count++;
49         //Prints number of punches
50         if(count%2 == 0)
51             Serial.println(count/2);
52
53         //user gets 10 points for every 5 punches
54         if (count%10 == 0){
55             points += 10*(group_factor+1);
56             Serial.printf("Plus 10 points! Current total: %u\n", points);
57         }
58
59         //Checks if device is moving in each direction
60         if(x < X_STABLE - X_BUFFER || x > X_STABLE + X_BUFFER){
61             // Serial.printf("%u: x moving\n", x);
62             xmov = true;
63         }
64         if(y < Y_STABLE - Y_BUFFER || y > Y_STABLE + Y_BUFFER){
65             // Serial.printf("%u: y moving\n", y);
66             ymov = true;
67         }
68         if(z < Z_STABLE - Z_BUFFER || z > Z_STABLE + Z_BUFFER){
69             // Serial.printf("%u: z moving\n", z);
70             zmov = true;
71         }
72
73     return points;
74 }
```

```
75 #endif
```

Listing 14: Accelerometer for secondary device header file

WifiConn.h

```
1 void wifi_init(int status, char* ssid, char* pass) {
2     Serial.print("Connecting to WIFI");
3     while (status != WL_CONNECTED) {
4         Serial.print(".");
5         status = WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network
6         . Change this line if using open or WEP network:
7         delay(1000); // wait 5 seconds for connection:
8     }
9     Serial.println("Connected to wifi");
10 }
```

Listing 15: WiFi connection header file

awsPubSub.h

```
1 void aws_pub(int level, char* payload, AWS_IOT connecting, char*
2     topic) {
3     sprintf(payload, "{\"state\": {\"user1\": {\"level\": \"%d\"}}}", 
4             level);
5
6     //increase QoS by sending data 3 times
7     for(int i = 0; i < 3; i++) {
8         if(connecting.publish(topic,payload) == 0) { // Publish the
9             message
10            Serial.print("Publish update:");
11            Serial.println(payload);
12        }
13        else {
14            Serial.println("Publish failed");
15        }
16        delay(500);
17    }
18
19    int aws_sub(char* rcvdPayload) {
20        Serial.print("Received Message:");
21        Serial.println(rcvdPayload);
22        StaticJsonDocument<256> doc;
23        deserializeJson(doc, rcvdPayload);
24
25        int points = doc["state"]["user1"]["points"];
26        Serial.print("Points: ");
27        Serial.println(points);
28
29        return points;
30    }
31
32    void aws_pubPoints(int points, char* payload, AWS_IOT connecting,
33                        char* topic) {
34        sprintf(payload, "{\"state\": {\"user1\": {\"points\": \"%d\"}}}", 
35                points);
36    }
37}
```

```

33
34     if(connecting.publish(topic,payload) == 0) { // Publish the
35         message
36         Serial.print("Publish update:");
37         Serial.println(payload);
38     }
39     else {
40         Serial.println("Publish failed");
41     }
42     delay(300);

```

Listing 16: AWS header file

FormattedFinal.json

```

1 [
2   {
3     "id": "cdb94dd.c03f6b",
4     "type": "tab",
5     "label": "Working Website",
6     "disabled": false,
7     "info": ""
8   },
9   {
10    "id": "ce9913f3.e8ed6",
11    "type": "http in",
12    "z": "cdb94dd.c03f6b",
13    "name": "",
14    "url": "/user1/mytree",
15    "method": "get",
16    "upload": false,
17    "swaggerDoc": "",
18    "x": 350,
19    "y": 280,
20    "wires": [
21      [
22        "2fc0d2ac.eaba4e"
23      ]
24    ],
25   {
26     "id": "88ac596.87bc8a8",
27     "type": "template",
28     "z": "cdb94dd.c03f6b",
29     "name": "mytree.html",
30     "field": "payload",
31     "fieldType": "msg",
32     "format": "handlebars",
33   }

```

```

34     "syntax": "mustache",
35     "template": "<!DOCTYPE html>\n<html lang=\"en\"
dir=\"ltr\">\n  <head>\n    <meta charset=\"utf-8
\">\n    <title>My Tree</title>\n    <style>\n      body {\n        margin: 0px;\n      }\n\n      /*adding the background slideshow*/\n      my_tree_Container {\n        width: 100%;\n        height: 100vh;\n        background-position:
center center;\n        background-size: cover;\n        background-repeat: no-repeat;\n        background-visibility: hidden;\n        animation: animate 20s ease-in-out infinite;\n        background-image: url('https://cdn.pixabay.com/
photo/2016/11/21/03/56/landscape-1844226_960_720.
png'); /*morning*/\n      }\n\n      @keyframes animate{\n        0%, 100%{\n          background-image: url('https://cdn.pixabay.com/
photo/2016/11/21/03/56/landscape-1844227_960_720.
png'); /*Morning*/\n        }\n        20% {\n          background-image: url('https://cdn.
pixabay.com/photo/2016/11/21/03/56/landscape-184422
9_960_720.png'); /*midday*/\n        }\n        40% {\n          background-image: url('https://cdn.pixabay.com/photo/2016/11/21/03/56/landscape-18
44230_960_720.png'); /*Sunset*/\n        }\n        60% {\n          background-image: url(
https://cdn.pixabay.com/photo/2016/11/21/03/56/
landscape-1844231_960_720.png'); /*nightime*/\n        }\n        80% {\n          background-
image: url('https://cdn.pixabay.com/photo/2016/11/2
1/03/56/landscape-1844226_960_720.png'); /*early
morning*/\n        }\n      }\n\n      .navbar {\n        display: flex;\n        justify-content: space-between;\n        padding:
20px;\n      }\n\n      .left_Side {\n        display: flex;\n        .navbar > .
left_Side > div {\n          margin-right: 20px;\n          font-size: 0.9em;\n          text-
transform: uppercase;\n        }\n\n        .right_Side > div {\n          margin-bottom: 5
px;\n          font-size: 0.9em;\n          text-
transform: uppercase;\n          color: white;\n        }\n\n        .hyperlink {\n          height: 2
2px;\n          border-bottom: 1px solid
transparent;\n          transition: border-bottom 0
.5s;\n        }\n      }\n\n      .hyperlink a { /*take

```

```

the hyperlink itself*/\n          color: white;\n
    text-decoration: none;\n
transition: color 0.5s;*\n            }\\n\n        .\nhyperlink:hover { /*when hovering over the\nhyperlink*/\\n            border-bottom: 1px solid\nblack;\\n                .hyperlink a:hover {\n/*when hovering over the hyperlink*/\\n\ncolor: black;\\n            }\\n            /*finished\nstyling the navigation bar*/\\n\\n        .center {\n            display: block;\\n            margin-left:\nauto;\\n            margin-right: auto;\nobject-fit: cover;\\n            width: auto;\n            height: 85%;\\n        }\\n            /*centres the\nimage in the middle of the page*/\\n\\n        </style>\n</head>\\n <body>\\n\\n        <div class=\"\nmy_tree_Container\">\\n            <!--beginning of navbar\n-->\\n            <div class=\"navbar\">\\n                <!--left\nside of the navbar-->\\n                <div class=\"\nleft_Side\">\\n                    <!--hyperlink to my tree\n-->\\n                    <div class=\"hyperlink\">\\n                        <a href=\"https://18.133.254.24:1880/user1/\nmytree\">my tree</a>\\n                    </div>\\n\\n                    <!--hyperlink to our garden-->\\n                    <div\nclass=\"hyperlink\">\\n                        <a href=\"https:\n//18.133.254.24:1880/ourgarden\">our garden</a>\\n                    </div>\\n\\n                    <!--hyperlink to\ncompany donations-->\\n                    <div class=\"\nhyperlink\">\\n                        <a href=\"https://18.133.\n254.24:1880/companydonations\">company donations</a>\\n                    </div>\\n                    </div>\\n                <!--\nright side of the navbar-->\\n                <div class=\"\nright_Side\">\\n                    <div class=\"identity\">\\n\n                <!--Needs to be edited-->\\n                    <div>\nCompanyID: A Company</div>\\n\n                <!--Needs to be\nedited-->\\n                    </div>\\n                    <div class\n= \"identity\">\\n\n                <!--Needs to be edited-->\\n                    <div>UserID: James Smith</div>\\n\n                <!--Needs\nto be edited-->\\n                    </div>\\n                    </div>\\n\n                </div>\\n            <!--end of navbar-->\\n\\n            <!--adding the tree image-->\\n                <img id\n= \"myImg\" src=\"https://i.ibb.co/6vgp5V0/\ndrawisland-3.png\" class=\"center\" width=\"502\"\\n

```

```

height="580">\n\n      </div>\n      <script type="text/javascript">\n          function updateTree() {\n              if ({{payload}} == 6){\n                  document.\n                  getElementById(\"myImg\").src = String.raw`https://i.ibb.co/6B78s51/drawisland-8.png`;\n              }else if\n                  ({{payload}} == 5){\n                      document.\n                      getElementById(\"myImg\").src = String.raw`https://i.ibb.co/ZGLQyrH/drawisland-7.png`;\n                  } else\n                  if ({{payload}} == 4){\n                      document.\n                      getElementById(\"myImg\").src = String.raw`https://i.ibb.co/TLrLzx2/drawisland-6.png`;\n                  } else\n                  if ({{payload}} == 3){\n                      document.\n                      getElementById(\"myImg\").src = String.raw`https://i.ibb.co/Dg4rwBL/drawisland-4.png`;\n                  } else\n                  if ({{payload}} == 2){\n                      document.\n                      getElementById(\"myImg\").src = String.raw`https://i.ibb.co/M1RsZBJ/drawisland-5.png`;\n                  } else\n                  if ({{payload}} == 1){\n                      document.\n                      getElementById(\"myImg\").src = String.raw`https://i.ibb.co/jWKhnDm/drawisland-2.png`;\n                  } else {\n                      document.getElementById(\"myImg\").src\n                      = String.raw`https://i.ibb.co/6vgp5V0/drawisland-3.png`;\n                  }\n                  \n                  }\n                  \n                  updateTree();\n              </script>\n      </body>\n  </html>\n",
      "output": "str",
      "x": 830,
      "y": 280,
      "wires": [
        [
          "d89f163d.40a788"
        ]
      ]
    },
    {
      "id": "d89f163d.40a788",
      "type": "http response",
      "z": "cdb94dd.c03f6b",
      "name": "",
      "statusCode": "",
      "headers": {
        }
      ,
      "x": 1010,
      "y": 280,
      "wires": [

```

```

58     ]
59 },
60 {
61     "id": "8be0fcb8.5ead4",
62     "type": "http in",
63     "z": "cdb94dd.c03f6b",
64     "name": "",
65     "url": "/ourgarden",
66     "method": "get",
67     "upload": false,
68     "swaggerDoc": "",
69     "x": 360,
70     "y": 660,
71     "wires": [
72         [
73             "1df27ab2.508075"
74         ]
75     ]
76 },
77 {
78     "id": "d7147d09.0067b",
79     "type": "http in",
80     "z": "cdb94dd.c03f6b",
81     "name": "",
82     "url": "/companydonations",
83     "method": "get",
84     "upload": false,
85     "swaggerDoc": "",
86     "x": 330,
87     "y": 1060,
88     "wires": [
89         [
90             "49e3ff02.48e2"
91         ]
92     ]
93 },
94 {
95     "id": "ff9ea546.d476b8",
96     "type": "template",
97     "z": "cdb94dd.c03f6b",
98     "name": "ourgarden.html",
99     "field": "payload",
100    "fieldType": "msg",
101    "format": "handlebars",
102    "syntax": "mustache",
103    "template": "<!DOCTYPE html>\n<html lang=\"en\""

```

```

dir=\"ltr\">\n    <head>\n        <meta charset=\"utf-8\n        \">>\n        <title>Our Garden</title>\n        <style\n            media=\"screen\">\n            body {\n                margin: 0\n                px;\n            }\n            /*adding the background\n            slideshow*/\n            .my_tree_Container, .\n            our_garden_Condainer {\n                width: 100%;\n                height: 100vh;\n                background-position:\n                center center;\n                background-size: cover;\n                background-repeat: no-repeat;\n                background-visibility: hidden;\n                animation: animate 20s ease-in-out infinite;\n                background-image: url('https://cdn.pixabay.com/\n                photo/2016/11/21/03/56/landscape-1844226_960_720.\n                png');\n            }\n            @keyframes animate {\n                0%, 100% {\n                    background-image: url(\n                        https://cdn.pixabay.com/photo/2016/11/21/03/56/\n                        landscape-1844227_960_720.png); /*Morning*/\n                }\n                20% {\n                    background-\n                    image: url('https://cdn.pixabay.com/photo/2016/11/2\n                    1/03/56/landscape-1844229_960_720.png'); /*midday\n                    */\n                }\n                40% {\n                    background-\n                    image: url('https://cdn.pixabay.com/\n                    photo/2016/11/21/03/56/landscape-1844230_960_720.\n                    png'); /*Sunset*/\n                }\n                60% {\n                    background-\n                    image: url('https://cdn.\n                    pixabay.com/photo/2016/11/21/03/56/landscape-184423\n                    1_960_720.png'); /*nightime*/\n                }\n                80% {\n                    background-image: url('https\n                    ://cdn.pixabay.com/photo/2016/11/21/03/56/landscape\n                    -1844226_960_720.png'); /*early morning*/\n                }\n            }/*finished adding the\n            background slideshow*/\n            \n            /*styling the\n            navigation bar*/\n            .navbar {\n                display\n                : flex;\n                justify-content: space-between;\n                padding: 20px;\n            }\n            .left_Side {\n                display: flex;\n            }\n            .navbar > .left_Side > div {\n                margin-right: 20px;\n                font-size: 0.9em;\n                text-transform: uppercase;\n            }\n            .navbar > .right_Side > div {\n                margin-\n                bottom: 5px;\n                font-size: 0.9em;\n                text-transform: uppercase;\n                color: white;\n            }\n            .hyperlink {\n                height\n                : 22px;\n                border-bottom: 1px solid\n                transparent;\n                transition: border-bottom 0.5\n                s;\n            }\n            .hyperlink a { /*take\n            */

```

```
the hyperlink itself*/\n          color: white;\n  text-decoration: none;\n          transition:\n  color 0.5s;\n      }\n      \n      .hyperlink:\n  hover { /*when hovering over the hyperlink*/\n    border-bottom: 1px solid black;\n  }\n\n      .hyperlink a:hover { /*when hovering over the hyperlink*/\n        color: black;\n      }\n\n    /*finished styling the navigation bar*/\n\n      .center {\n        display: block;\n        margin-left: auto;\n        margin-right: auto;\n        object-fit: cover;\n        width: auto;\n        height: 85%;\n      }\n\n      /*centres the image in the middle of the page*/\n\n      .rightImage {\n        width: 50%;\n        position: absolute;\n        right: 0px;\n        height: 85%;\n        /*positions an image on the right side of the page*/\n      }\n\n      .leftImage {\n        width: 50%;\n        position: absolute;\n        left: 0px;\n        height: 85%;\n      }\n\n  </style>\n</head>\n<body>\n  <div class=\"our_garden_Container\"\>\n    <!--beginning of navbar-->\n    <div class=\"navbar\"\>\n      <!--left side of the navbar-->\n      <div class=\"left_Side\"\>\n        <!--hyperlink to my tree-->\n        <div class=\"hyperlink\"\>\n          <a href=\"https://18.133.254.24:1880/user1/mytree\"\>my tree</a>\n        </div>\n        <!--hyperlink to our garden-->\n        <div class=\"hyperlink\"\>\n          <a href=\"https://18.133.254.24:1880/ourgarden\"\>our garden</a>\n        </div>\n      <!--hyperlink to company donations-->\n      <div class=\"hyperlink\"\>\n        <a href=\"https://18.133.254.24:1880/companydonations\"\>\n          company donations</a>\n        </div>\n    <!--right side of the navbar-->\n    <div class=\"right_Side\"\>\n      <div class=\"identity\"\>\n        <!--Needs to be edited-->\n        <div>CompanyID : A COMPANY</div>\n\n        <!--Needs to be edited-->\n        <div class=\"identity\"\>\n          <!--Needs to be edited-->\n          <div>UserID : JAMES SMITH</div>\n        </div>\n      </div>\n    </div>\n  </div>\n</body>
```

```

        <!--Needs to be edited-->\n            </div>
>\n            </div>\n\n            </div>\n        <!--end of
    navbar-->\n\n            <!--start inserting avatars-->\n        <img id=\"u1Img\" src=\"https://i.ibb.co/6B7
8s51/drawisland-8.png\" alt=\"Tree stage 6\" class
=\"rightImage\">\n            <img id=\"u2Img\" src=\"
https://i.ibb.co/Dg4rwbL/drawisland-4.png\" alt=\"
Tree stage 4\" class=\"leftImage\">\n            <!--<p>
Level1: <span id=\"level1\"></span></p>\n            <p>
Level2: <span id=\"level2\"></span></p>-->\n\n        </div>\n\n        <script type=\"text/javascript\">\n            function updateTrees() {\n                var level1 = {{
payload.level1}};\n                var level2 = {{payload.
level2}};\n                \n                \n                //document.
getElementById(\"level1\").innerHTML = level1;\n                //document.getElementById(\"level2\").innerHTML
= level2;\n                \n                if (level1 === 6){\n                    document.getElementById(\"u1Img\").src = String
.raw 'https://i.ibb.co/6B78s51/drawisland-8.png';\n                } else if (level1 === 5){\n                    document.
getElementById(\"u1Img\").src = String.raw 'https://
i.ibb.co/ZGLQyrH/drawisland-7.png';\n                } else
if (level1 === 4){\n                    document.getElementById(\"u1Img\").src = String.raw 'https://i.ibb.co/
TLrLzx2/drawisland-6.png';\n                } else if (level1
=== 3){\n                    document.getElementById(\"u1Img
\").src = String.raw 'https://i.ibb.co/Dg4rwbL/
drawisland-4.png';\n                } else if (level1 === 2){
\n                    document.getElementById(\"u1Img\").src =
String.raw 'https://i.ibb.co/M1RsZBJ/drawisland-5.
png';\n                } else if (level1 === 1){\n                    document.
getElementById(\"u1Img\").src = String.raw
'https://i.ibb.co/jWKhnDm/drawisland-2.png';\n                } else {\n                    document.getElementById(\"u1
Img\").src = String.raw 'https://i.ibb.co/6vgp5V0/
drawisland-3.png';\n                }\n                \n                if (level2 ===
6){\n                    document.getElementById(\"u2Img\").
src = String.raw 'https://i.ibb.co/6B78s51/
drawisland-8.png';\n                } else if (level2 === 5){\n
                    document.getElementById(\"u2Img\").src =
String.raw 'https://i.ibb.co/ZGLQyrH/drawisland-7.
png';\n                } else if (level2 === 4){\n                    document.
getElementById(\"u2Img\").src = String.raw
'https://i.ibb.co/TLrLzx2/drawisland-6.png';\n                } else if (level2 === 3){\n                    document.
getElementById(\"u2Img\").src = String.raw 'https://

```

```

    i.ibb.co/Dg4rwbl/drawisland-4.png';\n        } else\n    if (level2 === 2){\n        document.getElementById\n        ('\"u2Img\"').src = String.raw`https://i.ibb.co/M1\n        RsZBJ/drawisland-5.png';\n    } else if (level2\n    === 1){\n        document.getElementById(\"u2Img\")\n        .src = String.raw`https://i.ibb.co/jWKhnDm/\n        drawisland-2.png';\n    } else {\n        document.getElementById(\"u2Img\").src = String.raw\n        `https://i.ibb.co/6vgp5V0/drawisland-3.png';\n    }\n}\n    }\n    updateTrees();\n</script>\n</body>\n</html>\n",
    "output": "str",
    "x": 860,
    "y": 580,
    "wires": [
        [
            "a25ea116.067ba"
        ]
    ]
},
{
    "id": "a25ea116.067ba",
    "type": "http response",
    "z": "cdb94dd.c03f6b",
    "name": "",
    "statusCode": "",
    "headers": {
        }
,
        "x": 1040,
        "y": 580,
        "wires": [
            [
                "3cfcc3258.18d1ee"
            ]
        ]
},
{
    "id": "3cfcc3258.18d1ee",
    "type": "http response",
    "z": "cdb94dd.c03f6b",
    "name": "",
    "statusCode": "",
    "headers": {
        }
,
        "x": 1110,
        "y": 940,

```

```

139     "wires": [
140       ]
141   },
142   {
143     "id": "4531e19a.469ff",
144     "type": "mqtt in",
145     "z": "cdb94dd.c03f6b",
146     "name": "",
147     "topic": "User1PD/user1/level",
148     "qos": "0",
149     "datatype": "auto",
150     "broker": "9293df66.3b8c9",
151     "x": 350,
152     "y": 200,
153     "wires": [
154       [
155         "2fc0d2ac.eaba4e"
156       ]
157     ]
158   },
159   {
160     "id": "2fc0d2ac.eaba4e",
161     "type": "function",
162     "z": "cdb94dd.c03f6b",
163     "name": "",
164     "func": "//turn the string to an int\nif(msg.
topic == \"User1PD/user1/level\") {\n    p = JSON.
parse(msg.payload);\n    g = p.state.user1.level;\n    msg2 = parseInt(g, 10);\n    msg.payload = msg2
;\n    context.value = msg2;\n    return msg;\n}\nelse {\n    msg.payload = context.value;\nreturn msg;\n}",
165     "outputs": 1,
166     "noerr": 0,
167     "initialize": "",
168     "finalize": "",
169     "x": 620,
170     "y": 200,
171     "wires": [
172       [
173         "88ac596.87bc8a8",
174         "9318d5.c3aa0728"
175       ]
176     ]
177   },
178 }

```

```

179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
{
    "id": "9318d5.c3aa0728",
    "type": "debug",
    "z": "cdb94dd.c03f6b",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "false",
    "statusVal": "",
    "statusType": "auto",
    "x": 830,
    "y": 140,
    "wires": [
        ]
},
{
    "id": "ebb42989.e43488",
    "type": "inject",
    "z": "cdb94dd.c03f6b",
    "name": "inject level",
    "props": [
        {
            "p": "payload"
        },
        {
            "p": "topic",
            "vt": "str"
        }
    ],
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "topic": "User1PD/user1/level",
    "payload": "{\"state\": {\"user1\": {\"level\": \"4\"}}}",
    "payloadType": "str",
    "x": 380,
    "y": 120,
    "wires": [
        [
            "2fc0d2ac.eaba4e"
        ]
    ]
}

```

```

224     ]
225 },
226 {
227     "id": "e473d61e.c4b1f8",
228     "type": "comment",
229     "z": "cdb94dd.c03f6b",
230     "name": "inject node used for debugging",
231     "info": "",
232     "x": 370,
233     "y": 80,
234     "wires": [
235         ]
236     },
237     {
238         "id": "a136b75e.416768",
239         "type": "comment",
240         "z": "cdb94dd.c03f6b",
241         "name": "MQTT gets live data",
242         "info": "",
243         "x": 150,
244         "y": 200,
245         "wires": [
246             ]
247         },
248     },
249     {
250         "id": "1df27ab2.508075",
251         "type": "function",
252         "z": "cdb94dd.c03f6b",
253         "name": "",
254         "func": "//turn the string to an int\nif(msg.\ntopic == \"User1PD/user1/level\") {\n    p = JSON.\n    parse(msg.payload);\n    if(p == null) {\n        level1=context.get('level1') || 0;\n    } else\n        level1=p.state.user1.level;\n}\n\nmsg.payload.level1=level1;\ncontext.set(\n    'level1', level1);\nreturn msg;\n}\nelse if\n(\n    msg.topic == \"User2PD/user2/level\") {\n    p =\n        JSON.parse(msg.payload);\n    if(p == null) {\n        level2=context.get('level2') || 0;\n    } else\n        level2=p.state.user2.level;\n}\n\nmsg.payload.level2=level2;\ncontext.\nset('level2', level2);\nreturn msg;\n}\nelse {\n    level1=context.get('level1') || 0;\n    context.\n    set('level1', level1);\n    msg.payload.

```

```

level1 = level1;\n      \n      level2=context.get('
level2') || 0;\n      context.set('level2', level2);\n      msg.payload.level2 = level2;\n      return msg;\n    },
    "outputs":1,
    "noerr":0,
    "initialize":"",
    "finalize":"",
    "x":640,
    "y":500,
    "wires":[
      [
        "ff9ea546.d476b8",
        "34ae8d61.fe6a72"
      ]
    ],
  },
{
  "id":"d06ec8d9.8447d8",
  "type":"mqtt in",
  "z":"cdb94dd.c03f6b",
  "name":"",
  "topic":"User1PD/user1/level",
  "qos":0,
  "datatype":"auto",
  "broker":"9293df66.3b8c9",
  "x":350,
  "y":580,
  "wires":[
    [
      "1df27ab2.508075"
    ]
  ],
},
{
  "id":"54598b34.3665b4",
  "type":"mqtt in",
  "z":"cdb94dd.c03f6b",
  "name":"",
  "topic":"User2PD/user2/level",
  "qos":0,
  "datatype":"auto",
  "broker":"9293df66.3b8c9",
  "x":350,
  "y":500,
  "wires":[

```

```

298      [
299          "1df27ab2.508075"
300      ]
301  ],
302 {
303     "id": "34ae8d61.fe6a72",
304     "type": "debug",
305     "z": "cdb94dd.c03f6b",
306     "name": "",
307     "active": true,
308     "tosidebar": true,
309     "console": false,
310     "tostatus": false,
311     "complete": "payload",
312     "targetType": "msg",
313     "statusVal": "",
314     "statusType": "auto",
315     "x": 850,
316     "y": 420,
317     "wires": [
318
319         ]
320     },
321 },
322 {
323     "id": "b79f7daf.18f79",
324     "type": "inject",
325     "z": "cdb94dd.c03f6b",
326     "name": "Inject User 1 level",
327     "props": [
328         {
329             "p": "payload"
330         },
331         {
332             "p": "topic",
333             "vt": "str"
334         }
335     ],
336     "repeat": "",
337     "crontab": "",
338     "once": false,
339     "onceDelay": 0.1,
340     "topic": "User1PD/user1/level",
341     "payload": "{\"state\": {\"user1\": {\"level\": \"3\"}}}",
342     "payloadType": "str",

```

```

343     "x":350,
344     "y":420,
345     "wires": [
346       [
347         "1df27ab2.508075"
348       ]
349     ]
350   },
351   {
352     "id":"38fe4778.66edc8",
353     "type":"inject",
354     "z":"cdb94dd.c03f6b",
355     "name":"Inject User 2 level",
356     "props": [
357       {
358         "p":"payload"
359       },
360       {
361         "p":"topic",
362         "vt":"str"
363       }
364     ],
365     "repeat":"",
366     "crontab":"",
367     "once":false,
368     "onceDelay":0.1,
369     "topic":"User2PD/user2/level",
370     "payload":{"\\"state\\": {"\\"user2\\": {\\"level\\": \"4\"}}},
371     "payloadType":"str",
372     "x":350,
373     "y":340,
374     "wires": [
375       [
376         "1df27ab2.508075"
377       ]
378     ]
379   },
380   {
381     "id":"e819992b.50bbe8",
382     "type":"comment",
383     "z":"cdb94dd.c03f6b",
384     "name":"inject node used for debugging",
385     "info":"",
386     "x":210,
387     "y":380,

```

```

388     "wires": [
389
390   ],
391 },
392 {
393   "id": "c734b85e.5c8ad8",
394   "type": "comment",
395   "z": "cdb94dd.c03f6b",
396   "name": "MQTT gets live data",
397   "info": "",
398   "x": 150,
399   "y": 540,
400   "wires": [
401
402   ],
403 },
404 {
405   "id": "d101ce79.43922",
406   "type": "template",
407   "z": "cdb94dd.c03f6b",
408   "name": "companydonations.html",
409   "field": "payload",
410   "fieldType": "msg",
411   "format": "handlebars",
412   "syntax": "mustache",
413   "template": "<!DOCTYPE html>\n<html lang=\"en\" dir=\"ltr\">\n  <head>\n    <meta charset=\"utf-8\">\n    <title>Company Donations</title>\n  <style>\n    h1 {text-align: center;}\n    p {\n      text-align: center;}\n    body {\n      margin: 0px;}\n  </style>\n  <div style=\"width: 100%; height: 100vh; background-image: url('https://cdn.pixabay.com/photo/2015/12/01/15/43/black-1072366_1280.jpg'); background-size: cover; position: relative; overflow: hidden;\">\n    <div style=\"display: flex; justify-content: space-between; padding: 20px;\">\n      <div style=\"display: flex; align-items: center; gap: 10px;\">\n        <div style=\"flex-grow: 1; display: flex; justify-content: center; gap: 10px; font-size: 0.9em; text-transform: uppercase; margin-right: 20px; font-weight: bold;\">\n          <div style=\"margin-bottom: 10px;\">\n            .navbar > .left_Side > div {
              margin-right: 20px;
              font-size: 0.9em;
              text-transform: uppercase;
            }
          </div>
        <div style=\"flex-grow: 1; display: flex; justify-content: center; gap: 10px; font-size: 0.9em; text-transform: uppercase; margin-bottom: 10px; margin-top: 10px;\">
          .navbar > .right_Side > div {
            margin-bottom: 10px;
          }
        </div>
      </div>
    </div>
  </div>
</body>\n</html>"

```

```

: 5px;\n        font-size: 0.9em;\n        text-\n        transform: uppercase;\n        color: white;\n    }\n\n    .hyperlink {\n        height: 22px;\n        border-bottom: 1px solid transparent;\n        transition: border-bottom 0.5s;\n    }\n\n    .hyperlink a { /*take the hyperlink itself*/\n        color: white;\n        text-decoration: none;\n        transition: color 0.5s;\n    }\n\n    .hyperlink: hover { /*when hovering over the\n        hyperlink*/\n        border-bottom: 1px solid\n        yellow;\n    }\n\n    .hyperlink a: hover { /*\n        when hovering over the hyperlink*/\n        color:\n        yellow;\n    }\n\n    /*finished styling the\n        navigation bar*/\n    .center {\n        display:\n        block;\n        margin-left: auto;\n        margin\n        -right: auto;\n        object-fit: cover;\n        width: auto;\n        height: 85%;\n    }\n\n    .leftImage {\n        width: 50%;\n    }\n\n    position: absolute;\n    left: 0px;\n    height: 85%;\n}\n\n    .bottombar {\n        background-color: #fff;\n        overflow: hidden;\n        position: fixed;\n        bottom: 0;\n        width: 100%;\n        max-height: 25%;\n    }\n\n    .table {\n        display: flex;\n        align-items: center;\n        justify-content:\n        center;\n        border-spacing: 5px; /* need some\n        space */\n        table-layout: auto;\n        height: 20%;\n        width: 100%;\n    }\n\n    .table div {\n        display: inline;\n        max-width: 30%;\n        padding: 0px;\n    }\n\n    img {\n        width: 100%;\n    }\n\n    p {\n        display: block;\n        margin-top:\n        0;\n        margin-bottom: 0;\n        margin-left:\n        0;\n        margin-right: 0;\n    }\n\n    rightTree {\n        position: absolute;\n        right: 100px;\n        height: 50%;\n        width:\n        auto;\n    }\n\n    h1 {\n        color: white;\n        font-size: 300%;\n    }\n\n</head>\n<body>\n    <div class=\"\n        company_donations_Container\"\n        >\n        <!--beginning\n        of navbar-->\n        <div class=\"navbar\"\n            >\n            <!--left side of the navbar-->\n            <div\n                class=\"left_Side\"\n                >\n                    <!--hyperlink to my\n                    tree-->\n                    <div class=\"hyperlink\"\n                        >\n                            <a href=\"https://18.133.254.24:1880/user1/\n                                mytree\"\n                            >my tree</a>\n                        </div>\n                </div>\n            </div>\n        </div>\n    </div>\n</body>\n
```

```

<!--hyperlink to our garden-->\n          <div\n  class=\"hyperlink\"\>\n          <a href=\"https:\n//18.133.254.24:1880/ourgarden\">our garden</a>\n        </div>\n      <!--hyperlink to\n  company donations-->\n      <div class=\"\n  hyperlink\"\>\n      <a href=\"https://18.133.\n254.24:1880/companydonations\">company donations</a\n>\n    </div>\n    </div>\n    <!--\n  right side of the navbar-->\n    <div class=\"\n  right_Side\"\>\n    <div class=\"identity\"\>\n
      <!--Needs to be edited-->\n          <div>\n    CompanyID: A Company</div>\n          <!--Needs to be\n  edited-->\n          </div>\n          <div class\n  =\"identity\"\>\n            <!--Needs to be edited-->\n            <div>UserID: James Smith</div>\n            <!--Needs to\n  be edited-->\n            </div>\n            </div>\n            <!--end of navbar-->\n            <\n  img src=\"https://cdn.freelogovectors.net/wp-\ncontent/uploads/2017/04/tree_001.svg\" class=\"\n  rightTree\"\>\n            <br><br><br>\n            <!--adds\n  headings that change based on variables -->\n            <\n  h1 class=leftImage>Your company has donated: <\n  span id=\"money\"\></span> <\n  h1>\n            <br>\n            <\n  h1 class = leftImage>Total Company Points: <\n  span id\n  =\"points\"\></span><\n  h1>\n            <br>\n            <\n  h1 class = leftImage>Trees Donated By Your Company: <\n  span id=\"trees\"\></span><\n  h1>\n            <br>\n            <\n  div class=\"bottombar\"\>\n            <p>Below are\n            our linked charities, who you can donate money to\n            plant real life trees for you.</p>\n            <\n  div\n            <table\"\>\n            <\n  div>\n            <a href=\"https://www.worldlandtrust.org/appeals/plant-\n-a-tree/\">\n            <\n  img src=\"data:image/png;\n  base64,iVBORw0KGgoAAAANSUhEUgAAUOAAACXCAMAAACm/\n  PkLAAAAk1BMVEX///8LaWgAZ2YAXVwAYWD//v8AZGMAW1\n  oLaWkAX1z1+voAWFcLamcAZ2UAX10AZWW0sbCow8Ll8PDG2dm2\n  zc3A1NI4fHoqdXSdubjz+PgfchBfkpKAp6dNiIfY5uaQsrIAT05\n  0n5/U4uJ7o6NJhYTg6uq3zs5Yjo1pmJesxsZLhoWivb0/f38AU1\n  EwengASkmYvrwpyC7kAAAgAE1EQVR4n01dCXuiytKG7k43\n  OwoqgqgBRSBi5v7/X/dVdb06JCYz852Z85y6zz2TRGR5qa69\n  qjXt5+11/PPL9Pf/6D/6B8kK8109nh00VZUdZuv5KSysf/qe/\n  kayisWsilwqh0kxzrnPOW0mEII1ZVrn8U9f4COZ8fCzx1+6/

```

eT1T5FK+bpMPMo4cQFDlzi6Itfj8F/Cmefpx1n427j0Qxji8
FTXp93nB/4RYAb1htmMEN3RHU/fpPtsRYkEk7wvUoQTPuGEMJtu
58W3L108N1tFzQr0Mmve+19DdcSskUe8NbPRzc2iy4/XC/
yPbsLR2TZN//3tdpPNzstgAme9Oh7hk7KE8zer0+R0QvXZ9Nu/
hFPiuvQmb3nREXu1oBdcwsk3mpYx+NCINomBf6De8Rx870
pWURuMA+8nSxTFQZ7ZIE84W4VFK0WC3ZxwX2zC/grWgQm7PId5
OK8oFU3en60owtKEk3H7Ynv2RVDBN7vxexW7GQVx5XrRPC+
mYiourMpry0uNqX26wWEW5SF2k9SXnnUN3RDYmk4HpwwXi7h2
oGoCIUfkI0eaZpTCFOPFou8bfvWFBC2+M9ne9p3A+b8LtoeDV
6DeXOW0HbzXRLmbjgxs4HTnBq7HyAyMOF+vpe2Z45DV/VtZwrX
5VqqDeFeXHifcbnb3j32KrHolue5MacwpgajzeADPtg1ggVwAxN
68fWbvrAV2K0sNnYto/T211DLD0Q0DreCVsQmr5zbztzvyi1YLQg
/pR0pkaIhvhwBwjpp15p40wt9eTq+w24+Az3XvvMTCHV/e/
FpXQHVecv/VUQNZap77LxGpeWEGFj0e+w0V8Yjg+h883
gLBrIwvYVCzh1vS04I177uw7e06o4bB9/+sCxhRdjg+Y2wNr1
sJBUSNJATKjBs2Gg2vT0d3+1LiGRm8KKLANQ0xF5vD4CVHM0p19
IYhuiP39oz8i+LK1Zx7RyWrWrrMIuQSUUU21+KzbJy20
pQVPpYsCbl0fk8FE+g08G9dwm+GRu06gNBblurLR70RAKR2dXl1i
5upi+jB5NbcaUoB8oFrouHknDaISmpJwBBnb94PCP6Jx4iJoPq2
yXHuEMa7gVelK3BDcFDFLArRJLqxb4DAwWKZ+gCWqeka+/yTOc1
R7WdsV1Nx19HPtJr1th6ZrT1QsI4yvedUt3iuYOGYGNj47pV9
DU1rj6xJd1bBi1FhCIQ21mc2ppa8/A5dyiSY6ofQxc+it42
PASaVrpT9EEsUqo82AdPaRYgK7bttyv3BfgPwb2tzto+dD+mshL5
DfeDpPWP3S9jNF80C5/JHuu0r/Emmi+67pdfkDKdg2C2
1WhJgjcUQJYmbXiHd3BvxeeXddSUW7hEqsruNWct8svGqAl6L2o
/81KHGkv9J/2Cz3maolcEb7thQApb2pohtgTKwnf1RfQxGvqIr9
//A1JhqgJc7iXJB6+SpCHWuPn2jI9Uv1YsFwc9
RhLivoitpFBdgctvo01FArsejV+fAdSHE+00tH7xR3TTkXL13r9
tBqaG1I4qqU+RVNiQcf/c/FU0tQPTJy/3M4o3
tuHTCkDcvYohiYqnFRz6uNjlagalYGKNICuUJ0zAx6DYvQuOfT
4FVETA3qD4tmhqBq0+t100/W9daWEvCYLb402v0zRRJ1KmvHBX0
ZTnoLcPfwehQYzzERdIAALTxs86CLD074RD9gsTmx6
gWcqh0OARDtR3aCFvJCDIgBddmpiWISQThtxvnj6
BqTiIVHHjXsKbtwoJtehkpG43deU0IGSTtFE006MVHJX0Yzdp9
e6vD09zYxdNABQZpZK0eVRbm+xSvpcfZOG7kkPAUIKty1Z6
GccvgW/rqmnJ082mb7c13X5322TZjhD0mS0vYXVkd004
PkSzgsAd1vl2dh944Lrmis3Pn6livliTRB8ywcg20nx34ZTe100
0eMJKuEq6PtHPuMgoI5USU1gL+RPQBNbZCcJlw1Np5
XMHbMvCOuj87T6Ny4HdXXGC4ydG9Jn7a1fQdKTzw+Fwc0cUCiSK
/u7Z7x/GeaN5zCivUTq2kGK05t3XDxe1DF19HE/jbYE/ZfUGE1o
9BAmQ0Bxy6Wi71Gdnk9aQklU71bL301hMDSnB7GPxw5GOVSvw+
1ZvDepGPbtk6NVtaoVzPn0UTTZ/WLEzBFWx8x29ZqhG9oKxvhGD
/bbBiWtGAaDr4Q3G0GPfY70rYr6N5wHd1uHf4hIAPuC+
dQobuwspDgHChAl9o6RFW+FxiMehyxji5VtzJRicgKTHsQgVr0
nDg0nArXDiPj9bqU3jCInaEVdxWQgPQfuBfS87eiYEZ5/QBb+7
ZYDkhmiSKdC7Af17XN1L262imz0jf1Ue09DqtgcIxX8H9543t01

xZq9bBG1mRCJsk4krmBCiNQYnrhFGnGmT1wkf2dMxbe+Y+
gXBSeIR2g16C7qjH21+GNfaQN/cjI2ZBDd05ZG1CHMe9E5/8
DppP8WZt663G60w75K51QwnyRZgJJj813KOeHh3LKgOqyuNKZ9
TjvuHoUyLcXvWGjbWR5pP9pKMJXqxD8PKZ9IpBVhHp3iQjl1
PKzfc7X8bnZS05iUbJztMNfseH+TqaFQYOPn0MANOXzGlgvAJv5
qBL1bV492b7SDCAi3AqkmodTk0rcRCes4iKm9WvE3Hs+XMtxYS4
Flz36EWarihgNE/a2gvTkUHO3WXEi0qn3+E31EKd1u21EJj6Dj3
fCJqvo3kkhiM+tPdeMLal82bReefSojrZYot25J4xBv4IYzSpFg
/N8HiZJZSRKw7ldto97wnsE+M50EHbE0kOL20Z8cFAEt2
jShrhgeFH5569ebxrb4K/6ogbW/+LaMLFc3an/jKsGwIGg+no5
AKWC6hiqELngq0cFgzrye6+h7laSKu4ki6F3WXPoEyMZ5c7
CgUgS03toqZgMmJrns0iRsjauYd0xqdca/9eUATPSw3uj72y7
yJYai7Vu6IQuP2+jHc2pgDglWhBdwxvIRJFquWq4RQDg/3Yn3
gJlphaTNjAqfPu7tFycySXfiEYsdIGZ1buq5kCgYy7SK3J6b3
zFO3ef1VMQqGj0xN9AnZtfr4Mm/0RK5ARUo8tAC1/e+
qxhTP1vtoblpCPDaQ4F+ogP2J60XJvjA1CkOfIKn44KJoI5
0PQqPmSwffXVEGNY8LnsPCvvutp9NA7UFBuRuNcuSPoghzfCDK4
13F02rXT53eBPj8Qb7cKFbhgyl+RtpzKtxUEwVqocLE7
sqlBxViRpD3E9+Ii/goM9Xu/G8HDRWpqQT4ToUGPrq/5BUU9HkT
0NZspMys1CgffA37pfJp7liqsI0ujW76K5aMPzd9A8g8XFPnaSj
1yaRuj81JejPHc8AwEXeVI73DNu52aTiTjJg9TFC94s0FpjhU8
8Ubi3yL8nia+othwnVGwJkef5jpMG4CRT6+1GoqnQdtM0Aw4
cYZAsrrMPTsddvHcoo1X9PmHd592f0ftQUDZC00eHSSeC+/
BqeXPHTawsoU9HX2YfwZ8AwTbwznCL+Y61ehh7uU8
YmAwmDDjWFyBi8y/d6YFd5IVcffoPjzYkWjNHrukLz3EZArRs0
Y8yNOUdiVpPBWVxTyjBtbnuAn27kwwLjFnhiWPH7Qwfh2
eBVkTcuXy8WgHPVIzMRD/3foKA0YnEn+hDc/ZaewX6
eCQmwWRUWeaNd0WwqAjvw9sA8cffq1AooOZMCx1
jKiIqRzPaXvGSc5SHgzPTT62NQPPgaWneA/OG2uFy0GFH0HE+0x
+M0j6DIWPhRM1vrXOHYjr+s6il7cb6myQnUNIsMC399ELbgnW+
chWD2znTtZC25iG0dawoLjHYKrU4HCDVuQ6jkyKna2o09TA0
tGW7wsPsmnazVzDWZ8f0Nv4ELvctGtY21nILKg5B2mLMXC4Mi7
LuNoeMTh5kcIN9GyRM0pNR41700tqbge7jjpeR1F4H0
nZkozNnHgVq55zxZibZdW/bn2nm9
PzJctmfIeXpywgeHOHjyZsSyCGCgo8n1J3Q59aVKlrZYL5ivq80
v1UUobVhB1o1hmoZHAdd53UrR9xyG6R47BN+cN5wx1QA2
qNaNScGY2oE7sh3mfmZDuu0HLAbxQZulY9Vk8aUcnuh+cNeveN8
JIUC8NgzjIZ0SIa0AdgCejLiG+v4zj7vpnm7i+HZ13sWbF8
ZwSIFfYHzmcyMg3fHnJfYJ1al7SbFc+ViM16SfGSGFjagUEQive
9tJhk6teMcYaTALCuVzzWzb5WDmyScFJLhVYhPHe+okx1
bfBDhlyhKX/qdW5moTbAvtRGGyxpavtrBfx8V2LNTK9yR615
REUf/BBv8WJc5Gm0dR8r6SB6xW61/8NcHFHeJn7V8ioGaTzj/
zBFHXoHugd+YhpmMOHAMfy1COMfTjvR2qhFI02Ruu23MtcMFPW6
GioP52dtAN1+8ecVooQFIPfmkPmCvmdz9d64tpJn72OBkTh/
PZYXZe/nw57vfpBe0/Bnjw6z7ZiDU5uBLbKMmSo5miIkk6ZTk6u
4THWmNpDTuFbrVERI6uQ9yHwjnnfDt5zLUnedPwPvbPbm/2

TyfLdbEqZdt7LuxSxonbWXapkAWZizAscJnbYN3nJgHTGGRxyV4
 zFNB5pftY0yHKR4yRCy70zbCu0dN1XM0/jcR8hV7+gBLsA9Ux4
 XLsbHMaByvfd9Qyj1dU2YvUfv2BknmZ+IW2MMHYzYA9G8DC0ie9
 D8npoxIZ0GOEC7Nqq7hBkhAanTJG3h58oaV/Hp8vEdrR+
 EgHFVVH690CznRkyWiRKJwcRrdrJYCtNZYkgIG0tQIMxtTshy2
 APPDYMOr3SLWDfwCfc2HI2twNp/ocnTSDPhP++GsIg9RYfBCI1
 nifyxoqWV5aMGWyu1xWDQa7PM/lyrT2WiCZKq7
 KepnvdvnyX0qUgCn045GeWAtPxHA8bBrfIV6yu5mdf5W2slyS4
 xShy4G19EEKcFEx7hA4LZgrgJwdzauQJNMkFLK0zUKAgvD8
 HFRSOvWOiXcMSvMo0eLdZqur3CVpbSoe4ht7yWfL7SAXPt0
 fzVVMss0I4RBqnvRWVZZuNIbPrbikM/Cyjc72ehwc4Xme6dw4n3
 0yhIZxbEacdDijDOTMsao2UyUPAYQpEtEVQ/CMnoNtZT795
 JkfyeB5yuN0lE9achbWGQtungvo10sTpFr9/z0EmMu0
 ycmLPzVaV9quc3DjHkmY6ZL7HIceWrvtpa4/AHIVtw2J8w4f8
 WOii0c03agX5LnTjkCtSvs2Wdde+adi/sFra7CbXQ071MW7
 wSMIWSOH190xfg0mHE3h1Dde5DXyG4AekrlowBUP5iYekfS4GP7
 rd0o7y+60tdHICDRtqi6LPpGJIvGcf/EyX0feaNcq44rs6
 qfQAmKPiMFQdbP/+RB4Jbs6jQgn3zhmFtBxFj+xyMWsQufx0o5
 XIEFivWfqZj5A8irNACI3GN6pt4wqRySSv2kJVJ099
 NdEBTheuMIQT1KV7IapsAj4gvnVMzAi6LbyDWwbkY7
 mkVdaXkjPOjCQ5wtxgFOzCLY+bahrEUfkndIo+yQr8
 NQSIuULJthV6MzJ0pkKISxn1caNdaxRpwxJ+rvFHa726yPjbm48
 17Ym19Jgr3fpd/rrQQubQtky/KZtW6jYDnHvquKUwCGZvt0J4
 Wio9T/8V7c05bbAEvZeXri9AOUP42VzaRw7BEBVrhube4uBRP4J
 /Y1jejL03q7UWoDEOpromWogjMbIdFKC6DrezJ1NnI33FcrP4
 sMiyotSlVJtiTpZB/0GXs0jfzdF1TrQpqW9LAu9YZ4Hn5sZXLP9
 9Qpgq+wEoyLoe+Ha/WXZ1oWMM1EzJhZB0i6m0JMdeNrtwT+84
 oclFixaXZthrqn7mXfwVzk47BtryNqMSL1Kg6zaQPlC9PErb8
 zfbRPYSHX/5YaIedlu+3CTN5k66TJBer5N1XWeKw0xlh7
 HhYnz0d6a0Q78rX+UqGSvQho/lxcvrvoOUk+1ApY81T2VS5001
 JMqnYCNnvgv2ksSywAG3DCAW9XtRbysr/HTyGTCExO+T53NvBi
 4Uukl7xkZ9ffFipzLNZRaVf36L57cbFP4pGIZqf9
 DEdVeWDBTtYoBavN+XxbYVO+4y2ERDMBs2oG4B5jxYAabLqHINu
 4skBhcRSvGk7mzBgwLBhpmc0KzLuzzFQmArhuaNk038iG/yH08
 tUk8qW6K6Cd0dJ3V5TD6ygC/wtWPXRZBvMSch2mJ5Vigo0sd+
 xbfAVDjxdQFQ0xi8sbQ5yAV0DVx8xt2BYmpwPm5XBKKXc91vN95
 ff3q4pd7L1GFC1uFV3ooy902/UBZxmSxk875
 Y1WocHBkaNyhiDyQkGUZJu3ws0x2MXrNU5
 JcDimAwHPUMwEmcMVYExvDVZzxoh+WJ90BrC9vRHLbh/D+0pt8
 dlc5h27mrIT6ZC6m0hdVJuD466H2ifYvJtj3rK4NkyD90E1
 ZayxosfNZaoAqQBjpB15iW6yYhsJ2wJBDSbDQ1EnCLB7qXJ/y5
 qi0xx7wmLtrrKW1nc7DPgyviEDuhI/c7QsgJnUKpknmpzMJ0s7
 AdRmdEYZHePYmV64+uGK7KdFeczg40a6WSKmhJjrtXp+YTb+j0
 gzv/vs/96Avbz1PoreHWKZUVftyCxzJBHAFN6WWNor+i+
 awVbwgrorVWuj64nuMTQn64QS9d25UeGM1NLmhMpMB07
 nSMWoBdWQvUP8qC3BkkVtHnr6MyVzt5T/MWY/m0tdQ33GuBDb88

Yn265nleoEtHau0wSTEE5fi4m5YvCBZrLSTycYyeRqpka3clt3
YFdXxxkm4FybZoV2uAwJuEy9HGHF9ga/uReGTgdj7009
iVaSoiSKPnQ6t+zq41nSfnea7ttG8s9R815qVX96ScdNNjvtftq
1ndNOgVuSzTh4LX01zoI6YAMuTRfTZtGoSVoE6CaZoYaTffgyq+
t08wFbLX/IqmTAsh9Cc0sMnDoYm5P+51G23w+hpjguNmA/
ESJmuzj+aGxUQa99Ui5OKxurM6b5EX1K7MKNY/w5FVjdQW1YPi6
92Cb1ksN0+6mUXuU7rdF8Er3x05mFJ4C63AgDe7
CGTLuKiGClKbCVjqVUlxiBOeNw4BsbFSr9+nhMNu3ZQL5
uzDBGxebZKixlG/PaBNytQrqu1OL8504n3nvl2ha3
MsqhTgqwrtRaxEfji2xRKDtYNzNt5QT9nF10afUkK5HOetT6X1
dHmoKsASZEcse4J7ADsQYDmJzZ20/xJkIkAR7TmX4U5WGcI9
xQNaKRHVehfNzI4g76t1H/eeojmCrEo4MXF1VeB247nifLT/Ld0
524uayG3+cPEWmy0ZzadBvNoePw5UJC814tn//DqG305Y0vXfmz
1AGifrdIBlLwrBwVXCMrVAw5NBQ6P2oy4IgUiWvBhq9
QJEbidkuegSQ8vEH3eWzBi8CmSF0BgiV5Nqa7TWd00T01Pn3
tyEdrbIdQvibIQvmixi/LJSqhvkcSXvp1Swrk0+6DW2118vAF7
zZRfacty+D0Y2BpfQEckMb1l+u0JGW0ZrgaUZdWFmxes21
tNpYQzrbGQjgTUr+3NnYjh0Wtw79z5tIZHy3Jh0QWgpVVptpvJ8
Nkr1n67RBFcZ+8q+0V6stfrk9frmGPETrxXhzR6JRgk2
tAoxONKi6eAQCOk1tULASnzAu+7PB4uYY8NUwGdW4w73
Wni0myhntCwJqavNjenQ2bu0VL1MN2pUE6Z4
YjjMRUeEzTNazRlfpp/OzI49zreDHvB6A3vpvut8oIW7iQpC217
mHYHNcpXsdJuOE3GdZ0xyVDRuHtBxbNbunQv5kLCIEwIoUA4uPI
6bBHedGdM+Z0DzU94sxRwk/r2S2tSuwXdwpxN5H/OmhkaKYX/
XwcVYuxIUiw6w8Y0duF+1TNPGkr7HjeWAEoWe7Qac9
YMmYMVJxxY82SP0ZQVtfUPkKaib4k5+o5+oYwkTVml+/X8tL7
yLcF8+wzNQpgWzr24U3oDaMaoF+k4wzdG80Zuwo1hEaT/3
cIThMWUr6JfmWTU+LEHXQM3Mx+zJtiEgVQx20HAxyl2A8d2
xQamjEHfo7kwIgfhzEtvpDFAXJiTutKckknXyRN0zuIg3
berdhWk1J7LASPGuFPrMzRVEPKbigh9GWws92i02ePs0dhrP8
EFvL+D0sGggEs+yrZzLByGdY76yFxMNZfeliifbKOfApWyq/su6
BS8z9GMPTQSUK3ffioimtrcNibofIym1mqE5/ppb+iNdMHMU/fs
4wZEHD9B40FBGQY0goiVne+YyxHXGoxl01nv3colWb44
qWyAhrygyfsmjDNDdyou8nBzn2eHaovzjsZ39zmaZ1kmHnPntm9
SoakdTMCetBN1v0UTdkC+c21jp1MKv6W92i0XgxADO5J2puDhu
5cQlz/UgmDhgo6s5JzVaLVJz3pejzoUJIBafNMq/oOIxgaEHfVq
2iHHx0QEd747j5HMxKycRPukLjXRpjCU1rKf183/ymas1Woad8
iGeuvJnPRVm5OemYATba9RizG8Wbe2oWPwvFNee+sia31Nh2ZUG
6xRpXK7VpfaHwRXzqtDmaVVwP+jt5dw0lnQuN6FM1lG5TCGj963
fHcomkRF07bXfNTNMHpc0aa8kuUkG7GRmx0Rs7
orgBNbydzFbLqknlyEFzWAQ+21RK3sPxD6rG30ZAuHLqwu5Kc5F
1bE63qfC2wVuHVrcd9g9On+xTNo93eLLzgm1EULZqyQM3pXKBp0
ZSNKN+sHJe8qZZ6p2rGPVsn6hjWpqssFpVyC2PddzqhIMdkgWLZ
2QQLDsahwVdBrQncxf8WC466zgXeIZ6AvmX0CzsLhZ3Gk+
haqlDEzhY732SJ9HcfZs3dU0+irZQ7gpN0sSsqxZybmbLqA0
vbeYZal0zZWG6pHV05kfM8dADi5NHcng74gD/pm7wzzhDNbFB/X

0czE2n31KBprqvsejS1tTB0X7l8n6Pjn5gv8YDaHBnqw2
UbkRuHwGrqNik2DRGBWAbnaq1VB9VBi+wJbleg5xMA5HIq23pAU
8ryM6feCFA/AiW07J4aYyabRTL6ytfQjN1hHMXac65niw5
oapWHOhBP/YzchL9+Ly7Xun/oPFutBh7XCMypQZB1WZRjmTv3
wBBOMLPYik70myrmkBV4MRttdx6jaSggivNRCN7pGVhsGe2
agEPghHVW1GMG/gqaZ7ENDi2Ft2biCE0Lu/t18/ZTOv3zwNV9
WqkHgQX3om1a9Ty6pzVVM83QoJ970BsU3Di5DiI10kGRBpjk9
FAulZX0+zsy8bWsa3hRpw1rUfbmc9pfYEEdaetiYLGjr+j0
iNPXji7McODdjqXdCE2tdzGfsjevWr0fpq6oHQFql/p48
IeyHrGUdYehHpzBVbzpR7GWKwDwYVM5vJAEBs0dCo
GcSzXTSttrzpYWpCrsxquF2Y8zme0YERn8rHlnJvDx3X2MZmgn+
1lP4IM5dFLjM0YTZLRj4IjID+0bSDgyvxmCU+nrzHT9
hKRbtRrRxnHzM0a5bgs2MbMD1Exq9ZMx/kDWmWntufJTmCd9ME1
b63VP+YFNXzP08RaIcf8kXJp9mpOnsIVyAhW6skZG04kB/wxmkd
7YmGCCpj6MBM0Zf2+I3bP+Ok0+2aEs1PkciC1svzIKDeDdjq0y8
hUKwFxLcnAjnmQsSuUjl4hR2W5R/AZbSvu5SY2YTLAAkdm6
gxXdyrnRnLauwfSfjLcNy3P22qxq80A7qK5bBlwJ8YDFF60M5
jdk0HEUzS11AR1GmX0Y96cP4rkf04
vvSpooBNtyZbfCtQuUSPxDm1jIPyMzGw40MOhtRNVRv5
xIBXITDAHv3YjwPKRbrEoB1wBYOCOejFqIec2FYt/kxbX9
BiCIF7p1b0XTSjNoRxmNTSv6hJ0ONq7ys05ZRG4rOP0cTYo/3d4
PvZG/PS2UTUhiH00093wf2ddd1MGlqiE4pxNzebgf1YUTcBuQs+
fe/Z2L10UJou4MRpb1UgEsw+JypbDx0Mmx9sWy7AA590a7pBE2
euGB1416UMchzmaDzTNZqqU419GC2uUPp+ZVL1hMJBa2
BqVOBmhwQgsB8s0S7s6ziumkkugV33TgyKgoaxAC3PuI8fG6
rcTuRgDisxwQJVT9cVFuL0RwJqudBW51wWJKXmmzwJLLsb3
kxavNfiuiUm98Y5rvs0tQBnP3yI5h4H9n1ToWtyTntPrFTgDuED
0NbC6sMhq02H9ATTRPYnF5S3FdGN5eMeIW0YBISCAvXYyu0+8
GUAMDIa1eUyQgrWwaHuRZ1VvF11D+9vc5bVpfUf3dt2
rRVxiDGgk9nXqllmwUYr/Tov10JYn/L79Qnx0ALpVdIqH4
RwhinGpm8GFEUftpNV760XrhoKXUYM9KMwpjylt2+Txnaodf1
bACNC3dkxe2/wu+JwvzEEN6arF9fuxMGJS9qiu6f0ZrYHvqt2
xhF+UrbjkpZHnUWLppp87E3QzI9Ud+3Pp5N+
QKtxKgKYHETbsLgybu6ULGOF5JwOsblDwX3iaF7
WUtuhfYzbMKiYnDfXVDsHBjU7uet0BfoVDGgLXsLg2
XacMpkUsSfuHR4KtIsW1x608u6Irx91T1TccQJnrK+oo+zWzq/
Hb6b0RFvYpyH1ar6KFiUlHM5WeVnKmfGUR7He8v1kU7LGF14
nQFJt7Ga496u760aZi/1bQk6B3c18dbFQglHeioUhgbD/
LBNUazaXHPnmWdJeB4knYPqJHBGATHkfoc90XqSDDuy91k+d7
jnCfz8djuIE/1D1WcpvDnIFIuGKFVHoxZG3A6UrVJUxDkG0F819
gXQZKEy62qVc/u6HVmo2WOijY5ujqY/06sx/VbB5Uwx0
wkRZTuZ1IgRslgbwKbYMfYtuq5N49p7BrdzWtfFR/4
HTxGRrHoREAAEmSURBVMxtMns6RnZk6ZVNMs94
OraMjbisWtTuyvY91zqrfyC2aa0f1bfjnynaM28x7Njr929tbWc
3mikl/qlT8KtsgalnP0n5Y006suq5aQENSgyCasJAZTtTnMqIgR
8HT1Rogu0jNBWD0kHGn7b3o2S/1YB6eI5OPihPu7HXqyn4nW00

1gRNmmWAfzfVV/2ecwEFgwsZq3oC+mAN0Pb4fp0i7WEY7
 TrRSYkDZhNm3e0u80Rf8m2HZTP3qLAYQihhzDatSp/UTq9
 QHvvDxz00+n93aloaMoLcEZc8xW68aqNUrl/JplfMmHjYPIEY4
 kKysuk4x1ju1cwMdeCroY1nyHppoonJft5Pz2UGkeSHsonq8
 oazdx/EUP9Y+hKX1Lg7Gm8Ri00sSkb+/fqSCdgzx0
 vGihSGT1kSQw6x0XHPeoJwaWyhvwttnF0kn/U+tQZSNbzaqEVb1
 gz1UUBpMwQJJXm5u5gYNfRVzc/+vNIwmNWAcYddugfoBDrtHre6
 mf0HRswpkEEdvM37VjOrAeN49j1iBlOR1syGlebPyCaPpZ/0
 mSOFX1dcNSbE8YpdYo/ePFkcvOuL+7YQhzlu0wABSty0YR4027
 qdSMgi48EsNs05noWTpolRuGfj2TNdgn4MLZRRdRcVdaXWxcAq5
 gyTu3VbmleAJ2uShn8Y2I4i8
 HuwKGYYhgjhNnPgyNqXIwJaGvDQAAyDSMmWHYoXKHQDOn0il1
 nDuzYa25y726qxMDhzTLQhyFmwa0y7jI21vH8QXdrpVpkmoPx19
 PNTXgEZc2BcdkZqMBC5qmj4+vKZo0rusQa4GlaItYilJzp4
 SAYbDbcioNEzHsGLS8CZZ95MkJvXZcYmwdi0FgjLrM1DcnhW5
 xfuPOd8t//iSKGfImeGU00WIpL4GPhiBnIhboUZM3Nbe6VoKTn5
 X8dNjxf116b1hd6FnwHYSq2XEYa8ryfjmg+GUqgK5OEzfcV+Vb
 5em/VFUcpCJ9Svp9v48Ybiinwv3noVjy4FvkNX6eLDauIGIh+Gr
 3eVQqw0aI4wiRy8Src8q0qXkBq0mtkc1KVBmRyGfHffkL+
 0lgIscauNb+UpAz7B3EjLc8ut2lshX1LpMA7lh4b02AftJ6
 eLcpxAUsDrcUvwfdxqrdAH1eZStY+skd1YP8GMojZbmV3
 bgQDwzsA/uuEWLBBU8fxLIyPOJddF8LAybPZhyH/9cVoXaEGeM9
 H32eWqlpRtMp4pBZdfmDKjbd/qlXnj6GZ6
 iZdlqbngjfhXLHqXbMfZxJb9K1cUaFv2q3qiRest3fhMSuaG9
 KOKmKdAKipJUDLa2L5TJxXKhJckePfDuJ/adrQQ03Tx255wrYj7
 ibasbUSD6tG8lp4AIhbKd2cbSb8JmeREwoGbpZ9/kPNsU/r7
 EDuasfp0RZkce7xv5tVHLHU2rFx+mQYCDGrotTI5H1VJ4W08
 wyuHt0iZk8uyV1LWQld340DVUahvEx2RbjM5NiCh2codXcwp1j/
 gXGpiJl02JlwlsA1pBBQ8y00iqtMGtHPOo0d11FDF3Ep6
 nYUmYQEoZt/CNTRSw+Pa7DcK620CDUzEqCSY2/3
 RFqaSUHQBGHywYHcpnJYW+qVKAvLRK1tkuYPfta33Ge+
 eAYNIbazOSAvpdDSGdBtFNnOXeNX/5Q/xjh4iPuysWNhcJXmf8P
 0NrGhdl11sAondvi+I1g625xeIs8FJaA5p7qvtFrbyxe68rD/j0
 E0jAKYs/Byg0ZdNzJmnUckG+1USN+2NptUuU7tNtz9AFC29
 DRWjiXqwoVFPh6eIvfJh/nLAEPdDWrr8oHklVv7gx7DmPzbzJ1
 4c6jqAPEmbWEZHb4W4p9710p0SU8vh41H8D1T6GiFdqtsGy3
 RHIMQw71+1/Mgj3syMd4+ptyXTDxb011PezweIs/bqY6/+
 RfpPeK2zMx8jdKePNaLcqwnetL3n5+blk1vrddUjZt2dieCnGF3
 f5Z23N11+06swz1L5Y56GuWpcW6A47gF361W3P79ELLnViLcK+
 jYMttYs4FZ2zw2yHb8FF//4eEXB2c1vv2CFe4P2axWDeX7j69
 qXX2K5xtf8sorBmWDBUJ5g0/a9Q45noPbJq6EaH42WoZggnPqlR
 4f+goAzUA4dzeSn5HkLpGZVugJ09tfDdxU5VCPLY1
 twtvvwoqTAUrUt/RWdKFqtaGfUESzTnumphidv+
 qiYNwWXrSHKPo2CTGbt4kVftgWplsruX+0
 tbpMgXGdHzcrQVnIRA298HUTsL42+MEBsoJjqBdaJU3MD6s9JPP
 3NmJEjtnXDkoWZKv/vFwkIkSfdk18AkckLy/J4Iyj5YDni+

aNTP18afJCg4qRlhv4cUN9aJzvgv3kaB+0m6hFNiEU8rgirq8
 JngeBDe1JrIcKsBdUyP4j/WWA17dVI3QvjNzuWK6h71
 BrNxhF1HohOQa9jq4N8Vm36AFJTrxUMmVffkJkG25E4
 ztCbHLaw7ANfnCYcaF7hsg7CibQ4LRS/P1
 LiYAzbZiHanTIXfkxFVY2M8ddfplqcV8b6admH4hvR5X3Anr9
 VSXtVScTS8bHGG0+Nb2KFNF6HGDJda+N4V+23f76DZ9gZ2sZylr
 +MZSlkHU2k/qf9mJk5AU+LirTW56HuMzW/eXcbfXDpBjf1xfSX6
 utupyR31AuEIzzdR02Ed2d/y2GiT+aNypjjhLQ10+
 GwWfMoKglCasArau2PchZUd14cU/sazRe1KawqHyiyNI7f4M1
 bpaiwei19w9QcUYDSmPg7LV7hGzpKOPFeC0YeqLem10wTNLvtC8
 I7GXWPoPfrJ6PXLfMmN/HStG8/LoQzfDeCSCftpNXwtf/rCM0X+
 Hv/rjo012mokq1lfosm0gzgxIlFhc88Q4Yij5THxTv4K8T7icj4
 ArfacJMCwGRGIJeZLkNVou7Yg5L9pDeapmgugvV8EzR1beJ4Q0
 TM15IbbT80ZKw9MpmdhZq9Z2afg20eVPThg0Ce95MtQSrp4NE0
 9xFU9sy0eQFfGk5ATtecV1EJG15uS/GdygoBdEdrlwEre7
 IoTOxhJNvKOE3e0W31E9YuEIzaHtgpmaiRzVupfgMTQwzjj+2
 WhkwH4php2iWfSXzwJtajVMcD/UjNOPEEx3nEO10+KtzxDu+4
 HZDg0/RruQb5kq09AzPBFZXuVchB7E201FaTjm/GFQ00SMEpmt0
 HUzQx0z2eCfIZmlu0zz1SBWr04frtNEs+g9GaFq6YVmJ9QhN0
 GkGbp4+x3MXui8Lh7WyLYExmD6UOT9JdeKBqUWjUFtT7jq6B+
 IQi1gLlvPy9hN7jVyh2dIUzzP4GprAm2TCn0cbB2KK5kAjNLX9
 pT4je95F80VNS1CFV3KsnW6CcXVpc77oEZH1F/jTqhNc5D6XD4
 GtmWjfLKxEJA7xTOm1xWa7bucoonu2rhZ6DM05yagP9as4
 dfRPgc95goLDx7xJu6ZqSYDhhRTG/J9L314k3a6RZVPqJ5+Lj/
 1E2M9Eu71IDdqDeSORbj1LLARe7NkZSN9aq/AZ3hzS2RHWE+
 foRmgYT3uxNJuDr6LJjxY1Er6EEExsQe7n+hhNTI2
 ZmdxATYWIvzNnaS9zkNlpp2k6YPlnx82gpV1MkbCywkbKXA1
 ibWwHOgobIXDn2uA+8JNHHQ2Hh84mdovqjpYw59f+ww3+dNa0
 eFLFGxZg2AEOb41nh1qr8mJFKqm5s2fWGmuWar4tjoE/trlU4B
 5CV6w84tKg33PTb3a7BJMpZ00gON8cBYXmWrTnekynfz9EMw08
 gk6mRn/Fm55ERu5Jq8A5f3EdzuV+v97AIRP/t1//D0mjQNTP40/
 8e3j54am1ZtWvEWk4v+NjxpmsFYIgt92iy0Yc3DxmH8yry6DDLE
 1bTC0bvEQ81jg5HbmZUp8+Wvn6K5tI1Op9udf05mlrVrj5sUL5
 Hj+TmlykVbUg0va+ZnJc67wHi+xWVHQGEe7Zwm006W5/r+rzep1
 WjU5v2o7dx+uFWBLhJT0uwSu4GNCuqP9+A9xBNnsZBEK5XNve8
 anrAE2hqNVOPxNhd+SXoantu6kUcJ0VsFLxJrpIhQtm/WLozKB
 2CY4TNU3P8xiaQJPxXPbsLNryZV19oHqJfPasxf0e/0M0dW7bF8
 p8vzxff/wMMuBW2KqKX6zui9fhyY09JAjmnpNwg1D7KFHC0
 FRnZMoPt9qLWD6fXKIV+c202RAWDdkI7/c18khuvjCft7mzc1
 iUXP9Xj/5U2hiE7cqcfbt2/mdvxBNbY1FH4
 bskchXFDufHWabVEhPUMY+uR2tUYLX1LmPpfssYm5gYFDt4Iptg
 4FC8yk7s7+vH3ITH/aspv1f0X0097d7wM65ZAZD3
 BhrvxJNrUhwc1YRwfabr96y0Mg3K51NTKn5NB13b07aPpwfxa4
 kpg01mVcjo/AYmMwZ19KA32ihTbA+fTawHiSN4Hig6m8susRpr8
 UTVTi0ngx7bgix1MRGiwybCjfyJ1KXtqr3hIRRwb7
 xdpRDLkvPDkMuthw3WuCL3mnn6BpuahX6fjsVma36CZ3g17746

```

4ush1r9KvRRMjsy5p26KUbMreN5g6usxGNoVFx3
qnNQVs uSPbVnKtTD3ENW7DrKWMky/3LH9mIaGSu95wcGkaundj+
WT9XNN48tkMBwZf97v/ajS1XcT6DuAQDSdyKVCTjIcEh6/
jLmLDAMuPb+TWjKv2y3YnkmqXgiH75T1VP7U3Z/BCxTTmjOU+4
iY8te2X83qK/sxzht7kln45mrg22k5AjORHLu4VEAt9EuENK25
TLtNhnFHb3dQqv h91Y2MdL9qH+WkWCZfYm68XEH/uC+Gw52k+
pDAVA0wpsTtLaD4uh3iRDTfx1bi/AU0tj9TscZyHltoYQnt7
snttc1Cdbt6aZlvtF/2mEGwkTonc38rVm f6dadVPeJZwMT6ZhRS
7xm1DsSX6sfDz1+lbxQHNvx9NeG1rzuRch3
ieurRkkZdj eDcRss13rINT6FdkcPPwrU1Snoh6oAqnE50xIbdm6
HLYuHj+OuWHE3X4FXQrVTH1qymocLlzXS0cvHHH4VdZ2ymky4
TdgMntt9338p7PROTQK5iUep+pgVOGJnQcpp7
NxTSDcmaGfSUYfsdK1wAU2A1N7AZr/o+t9fn4QuHRvjGawGL6
dqkRNtLeohlN0MSeMHciOsn4c1l0lNIB3rk5nbx0JDfdShFRG0/
8Bso3Nnd8msy64a+OfT8yGNDhCYUlclLd/ETZFqal6A1fk8
kWhKDEHSWN0gptwxEjyWlVY1Q7Nqe00YIKOFnu4zMm9/uzDj+
nIgP/h7B+fpKbyMqczbwYqqR289Kj13xJGNv8VAlcyZ3
bfTBkJGVsn5+xQoGNzK85JYbZ7SEV7DkV9fhDzsUqVPhZe5uI66
y+DNDc0gC/jIL1u+CDPJRMUiAo1ZvqcDhkm4YIyvUrN5
OLRDZZfbNQpMjDUnDu+zTN+wkzVp6vGY6XYVG9633KEufQ4E6R3
WFhQgmnqypNs8YUdjReTPNLtfHtS3RY1+tMpx6ZWMF4Ae7jBVaL
3U+VZXxIeaa2ZJKY4uuUI5GInJzDrxe47nLKqi8b6xOabbflR1K
5PXanKo7dHzfwcb8TXHvcIFSsecOEULN/j1MdjilKa5mRy+
v1YgtnP9VXwwXK7fa37goZZpHpqZQwDUYzaG70jmcm2ekf2RJ9
tA6sXbhYLMKb/es6jsOPw390M2drty4ZuD8umEnvd0Ie4BNRm73
t898E5UOhcfXBy/Xvv/p6v452dXZM7FR4DISaKq/0fY5ReJYcs/
r3CZt/LV1BuFinGTqVSNTNlq4XefCPr05/Nf1Levz+o//oX0D/B
5iMaTD6w8GWAAAAAE1FTkSuQmCC\" alt=\"Worldland Trust
\" style=\"width:50%; height:100%; margin-left:auto
; margin-right:auto\">\n                </a>\n            </
div>\n                <div>\n                    <a href=\"https://
www.nature.org/en-us/get-involved/how-to-help/plant
-a-billion/\">\n                        <img src=\"https://
initiative20x20.org/sites/default/files/2019-03/
TNCLogoPrimary_RGB.jpg\" alt=\"Nature Conservancy\"/>
style= \"width:50%; height:100%; margin-left:auto
; margin-right:auto\">\n                        </a>\n                    </
div>\n                <div>\n                    <a href=\"https://
onetreeplanted.org/\">\n                        <img src=\"https:
//media.pagefly.io/file/get/otp-logo-long-greenpng-
1519919839601.png\" alt=\"Worldland Trust\" style
=\"width:50%; height:100%; margin-left:auto; margin
-right:auto\">\n                        </a>\n                    </div>\n                <div>\n                    <a href=\"https://treecouncil.org
.uk/\">\n                        <img src=\"https://pbs.twimg.
com/profile_images/1310870707811872769/vqXwpHfx_400

```

```

x400.jpg\" alt=\"Worldland Trust\" style=\"width:50
%; height:100%; margin-left:auto; margin-right:auto
\">\n          </a>\n          </div>\n      </div>\n      </div>\n      <script type=\"text/javascript
\">\n          //variables to get points, find how much
money each point gets and round to 2dp\n          var
totalPointsVal = {{payload}}; //change with payload
in node-red\n          var totalMoney = totalPointsVal *
0.0001;\n          var treesDonated = totalMoney * 10;\n
          treesDonated = Math.floor(treesDonated);\n
          totalMoney = totalMoney.toFixed(2); \n
          function updateInfo() {\n              document.
getElementById(\"money\").innerHTML = totalMoney;\n
              document.getElementById(\"points\").
innerHTML = totalPointsVal;\n              document.
getElementById(\"trees\").innerHTML = treesDonated
;\n          }\n          updateInfo();\n      </script>\n</body>\n</html>",
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
    "wires": [
        [
            "3cfcc3258.18d1ee"
        ]
    ],
},
{
    "id": "47b350d0.710ab",
    "type": "debug",
    "z": "cdb94dd.c03f6b",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "false",
    "statusVal": "",
    "statusType": "auto",
    "x": 870,
    "y": 820,
    "wires": [
        [
        ]
    ],
}

```

```

442     "id": "49e3ff02.48e2",
443     "type": "function",
444     "z": "cdb94dd.c03f6b",
445     "name": "",
446     "func": "var totPoints=context.get('totPoints')\n|| 0;\nif(msg.topic == \"User1PD/user1/points\") {\n    p = JSON.parse(msg.payload);\n    points = p.state.user1.points;\n    pointsInt = parseInt(points);\n    totPoints+=pointsInt;\n    msg.payload=totPoints;\n    context.set('totPoints', totPoints);\n    context.value=totPoints;\n}\nelse if(msg.topic == \"User2PD/user2/points\") {\n    p = JSON.parse(msg.payload);\n    points = p.state.user2.points;\n    pointsInt = parseInt(points);\n    totPoints+=pointsInt;\n    msg.payload=totPoints;\n    context.set('totPoints', totPoints);\n    context.value=totPoints;\n}\nreturn msg;\n}\nelse if(msg.topic == \"User1SD/user1/points\") {\n    p = JSON.parse(msg.payload);\n    points = p.state.user1.points;\n    pointsInt = parseInt(points);\n    totPoints+=pointsInt;\n    msg.payload=totPoints;\n    context.set('totPoints', totPoints);\n    context.value=totPoints;\n}\nreturn msg;\n}\nelse {\n    msg.payload = context.value;\n    return msg;\n}",
447     "outputs": 1,
448     "noerr": 0,
449     "initialize": "",
450     "finalize": "",
451     "x": 660,
452     "y": 880,
453     "wires": [
454         [
455             "47b350d0.710ab",
456             "d101ce79.43922"
457         ]
458     ],
459 },
460 {
461     "id": "7685613.6a956a",
462     "type": "inject",
463     "z": "cdb94dd.c03f6b",
464     "name": "User2SD (300)",
465     "props": [
466         {
467             "p": "payload"

```

```

468     },
469     {
470         "p":"topic",
471         "vt":"str"
472     }
473 ],
474 "repeat":"",
475 "crontab":"",
476 "once":false,
477 "onceDelay":0.1,
478 "topic":"User2PD/user2/points",
479 "payload":"{\\"state\": {\\"user2\": {\\"points\": \"300\"}}}",
480     "payloadType":"str",
481     "x":360,
482     "y":760,
483     "wires":[
484         [
485             "49e3ff02.48e2"
486         ]
487     ]
488 },
489 {
490     "id":"93c0c90.65c9a38",
491     "type":"inject",
492     "z":"cdb94dd.c03f6b",
493     "name":"User1PD (800)",
494     "props":[
495         {
496             "p":"payload"
497         },
498         {
499             "p":"topic",
500             "vt":"str"
501         }
502     ],
503     "repeat":"",
504     "crontab":"",
505     "once":false,
506     "onceDelay":0.1,
507     "topic":"User1PD/user1/points",
508     "payload":"{\\"state\": {\\"user1\": {\\"points\": \"700\"}}}",
509     "payloadType":"str",
510     "x":360,
511     "y":820,

```

```

512     "wires": [
513         [
514             "49e3ff02.48e2"
515         ]
516     ]
517 },
518 {
519     "id": "5634e15.a7b282",
520     "type": "mqtt in",
521     "z": "cdb94dd.c03f6b",
522     "name": "",
523     "topic": "User2PD/user2/points",
524     "qos": "0",
525     "datatype": "auto",
526     "broker": "9293df66.3b8c9",
527     "x": 340,
528     "y": 880,
529     "wires": [
530         [
531             "49e3ff02.48e2"
532         ]
533     ]
534 },
535 {
536     "id": "a8c4bb38.c4d8e8",
537     "type": "mqtt in",
538     "z": "cdb94dd.c03f6b",
539     "name": "",
540     "topic": "User1PD/user1/points",
541     "qos": "0",
542     "datatype": "auto",
543     "broker": "9293df66.3b8c9",
544     "x": 340,
545     "y": 940,
546     "wires": [
547         [
548             "49e3ff02.48e2"
549         ]
550     ]
551 },
552 {
553     "id": "c14d8976.4f80a8",
554     "type": "http in",
555     "z": "cdb94dd.c03f6b",
556     "name": "",
557     "url": "/user2/mytree",

```

```

558     "method": "get",
559     "upload": false,
560     "swaggerDoc": "",
561     "x": 1370,
562     "y": 300,
563     "wires": [
564       [
565         "cc8e62e3.5ff29"
566       ]
567     ]
568   },
569   {
570     "id": "c47e09ab.78c1d8",
571     "type": "template",
572     "z": "cdb94dd.c03f6b",
573     "name": "mytree.html",
574     "field": "payload",
575     "fieldType": "msg",
576     "format": "handlebars",
577     "syntax": "mustache",
578     "template": "<!DOCTYPE html>\n<html lang=\"en\">\n  <head>\n    <meta charset=\"utf-8\">\n    <title>My Tree</title>\n    <style>\n      body {\n        margin: 0px;\n      }\n\n      /*adding the background slideshow*/\n      my_tree_Container {\n        width: 100%;\n        height: 100vh;\n        background-position: center center;\n        background-size: cover;\n        background-repeat: no-repeat;\n        backface-visibility: hidden;\n        animation: animate 20s ease-in-out infinite;\n        background-image: url('https://cdn.pixabay.com/photo/2016/11/21/03/56/landscape-1844226_960_720.png'); /*morning*/\n      }\n\n      @keyframes animate{\n        0%, 100%{\n          background-image: url('https://cdn.pixabay.com/photo/2016/11/21/03/56/landscape-1844227_960_720.png'); /*Morning*/\n        }\n        20% {\n          background-image: url('https://cdn.pixabay.com/photo/2016/11/21/03/56/landscape-1844229_960_720.png'); /*midday*/\n        }\n        40% {\n          background-image: url('https://cdn.pixabay.com/photo/2016/11/21/03/56/landscape-1844230_960_720.png'); /*Sunset*/\n        }\n        60% {\n          background-image: url('https://cdn.pixabay.com/photo/2016/11/21/03/56/\n
```

```

landscape-1844231_960_720.png'); /*nightime*/\n
} \n          80% {\n            background-\nimage: url('https://cdn.pixabay.com/photo/2016/11/2\n1/03/56/landscape-1844226_960_720.png'); /*early\nmorning*/\n          }\n          }\n        .\n      navbar {\n        display: flex;\n        justify-content: space-between;\n        padding:\n          20px;\n        }\n        .left_Side {\n          display: flex;\n          }\n          .navbar > .\n        left_Side > div {\n          margin-right: 20px;\n          font-size: 0.9em;\n          text-\n          transform: uppercase;\n          }\n          .navbar\n        > .right_Side > div {\n          margin-bottom: 5\n          px;\n          font-size: 0.9em;\n          text-\n          transform: uppercase;\n          color: white;\n          }\n          .hyperlink {\n            height: 2\n            2px;\n            border-bottom: 1px solid\n            transparent;\n            transition: border-bottom 0\n            .5s;\n            }\n            .hyperlink a { /*take\nthe hyperlink itself*/\n            color: white;\n            text-decoration: none;\n            transition: color 0.5s;\n            }\n            .\n            hyperlink:hover { /*when hovering over the\nhyperlink*/\n            border-bottom: 1px solid\n            black;\n            }\n            .hyperlink a:hover {\n/*when hovering over the hyperlink*/\n            color: black;\n            } /*finished\nstyling the navigation bar*/\n            .center {\n              display: block;\n              margin-left:\n              auto;\n              margin-right: auto;\n              }\n              object-fit: cover;\n              width: auto;\n              height: 85%;\n              } /*centres the\nimage in the middle of the page*/\n            </style>\n            </head>\n            <body>\n            <div class=\"\n            my_tree_Container\">\n            <!--beginning of navbar-->\n            <div class=\"navbar\">\n            <!--left\n            side of the navbar-->\n            <div class=\"\n            left_Side\">\n            <!--hyperlink to my tree-->\n            <div class=\"hyperlink\">\n            <a href=\"https://18.133.254.24:1880/user2/\n            mytree\">my tree</a>\n            </div>\n            <!--hyperlink to our garden-->\n            <div class=\"hyperlink\">\n            <a href=\"https://18.133.254.24:1880/ourgarden\">our garden</a>\n            </div>\n            <!--hyperlink to\n            company donations-->\n            <div class=\"\"

```

```

hyperlink\">\n          <a href=\"https://18.133.
254.24:1880/companydonations\">company donations</a
>\n          </div>\n          </div>\n          <!--
right side of the navbar-->\n          <div class=\"
right_Side\">\n          <div class=\"identity\">\n
          <!--Needs to be edited-->\n          <div>
CompanyID: A Company</div>\n          <!--Needs to be
edited-->\n          </div>\n          <div class
= \"identity\">\n          <!--Needs to be edited-->\n
          <div>UserID: James Smith</div>\n
          <!--Needs
to be edited-->\n          </div>\n          </div>\\
n\n          </div>\n          <!--end of navbar-->\n
\n          <!--adding the tree image-->\n          <img id
= \"myImg\" src=\"https://i.ibb.co/6vgp5V0/
drawisland-3.png\" class=\"center\" width=\"502\"
height=\"580\">\n\n          </div>\n          <script type=\"
text/javascript\">\n          function updateTree() {\n
            if ({{payload}} == 6){\n                document.
getElementById(\"myImg\").src = String.raw`https://
i.ibb.co/6B78s51/drawisland-8.png`;\n            } else if
({{payload}} == 5){\n                document.
getElementById(\"myImg\").src = String.raw`https://
i.ibb.co/ZGLQyrH/drawisland-7.png`;\n            } else
if ({{payload}} == 4){\n                document.
getElementById(\"myImg\").src = String.raw`https://
i.ibb.co/TLrLzx2/drawisland-6.png`;\n            } else
if ({{payload}} == 3){\n                document.
getElementById(\"myImg\").src = String.raw`https://
i.ibb.co/Dg4rwBL/drawisland-4.png`;\n            } else
if ({{payload}} == 2){\n                document.
getElementById(\"myImg\").src = String.raw`https://
i.ibb.co/M1RsZBJ/drawisland-5.png`;\n            } else
if ({{payload}} == 1){\n                document.
getElementById(\"myImg\").src = String.raw`https://
i.ibb.co/jWKhnDm/drawisland-2.png`;\n            } else {
\n                document.getElementById(\"myImg\").src
= String.raw`https://i.ibb.co/6vgp5V0/drawisland-3.
png`;\n            }\n            }\n            updateTree();\n
          </script>\n      </body>\n    </html>\n",
        "output": "str",
        "x": 1850,
        "y": 300,

```

```

582     "wires": [
583         [
584             "6d0b66d1.04bdd8"
585         ]
586     ]
587 },
588 {
589     "id": "6d0b66d1.04bdd8",
590     "type": "http response",
591     "z": "cdb94dd.c03f6b",
592     "name": "",
593     "statusCode": "",
594     "headers": {
595         ...
596     },
597     "x": 2030,
598     "y": 300,
599     "wires": [
600         ...
601     ]
602 },
603 {
604     "id": "d0131af2.fc2f38",
605     "type": "mqtt in",
606     "z": "cdb94dd.c03f6b",
607     "name": "",
608     "topic": "User2PD/user2/level",
609     "qos": "0",
610     "datatype": "auto",
611     "broker": "9293df66.3b8c9",
612     "x": 1370,
613     "y": 220,
614     "wires": [
615         [
616             "cc8e62e3.5ff29"
617         ]
618     ]
619 },
620 {
621     "id": "cc8e62e3.5ff29",
622     "type": "function",
623     "z": "cdb94dd.c03f6b",
624     "name": "",
625     "func": "//turn the string to an int\nif(msg.topic == \"User2PD/user2/level\") {\n    p = JSON.parse(msg.payload);\n    g = p.state.user2.level;\n}\n\nreturn g;\n"

```

```

    msg2 = parseInt(g, 10);\\n      msg.payload = msg2
;\\n      context.value = msg2;\\n      return msg;\\n}
else {\\n      msg.payload = context.value;\\n
return msg;\\n}",
"outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"x":1640,
"y":220,
"wires":[
[
  "c47e09ab.78c1d8",
  "23738080.ba78a"
]
],
{
  "id":"23738080.ba78a",
  "type":"debug",
  "z":"cdb94dd.c03f6b",
  "name":"",
  "active":true,
  "tosidebar":true,
  "console":false,
  "tostatus":false,
  "complete":false,
  "statusVal":"",
  "statusType":"auto",
  "x":1850,
  "y":160,
  "wires":[
]
},
{
  "id":"348449b7.bf3566",
  "type":"inject",
  "z":"cdb94dd.c03f6b",
  "name":"inject level",
  "props":[
    {
      "p":"payload"
    },
    {
      "p":"topic",
    }
  ]
}

```

```

668         "vt ":"str"
669     }
670   ],
671   "repeat":"",
672   "crontab":"",
673   "once":false,
674   "onceDelay":0.1,
675   "topic":"User2PD/user2/level",
676   "payload":"{\\"state\": {\"user2\": {\"level\": \"3\"}}}",
677   "payloadType":"str",
678   "x":1400,
679   "y":140,
680   "wires":[
681     [
682       "cc8e62e3.5ff29"
683     ]
684   ]
685 },
686 {
687   "id":"97c6e3e.0ddb92",
688   "type":"comment",
689   "z":"cdb94dd.c03f6b",
690   "name":"inject node used for debugging",
691   "info":"",
692   "x":1390,
693   "y":100,
694   "wires":[
695     []
696   ],
697 },
698 {
699   "id":"bf641905.78dcc8",
700   "type":"comment",
701   "z":"cdb94dd.c03f6b",
702   "name":"MQTT gets live data",
703   "info":"",
704   "x":1170,
705   "y":200,
706   "wires":[
707     []
708   ],
709 },
710 {
711   "id":"5a4a903b.7e731",
712   "type":"comment",

```

```

713     "z": "cdb94dd.c03f6b",
714     "name": "MQTT gets live data",
715     "info": "",
716     "x": 130,
717     "y": 900,
718     "wires": [
719       []
720     ],
721   },
722   {
723     "id": "7226aef5.5434f",
724     "type": "mqtt in",
725     "z": "cdb94dd.c03f6b",
726     "name": "",
727     "topic": "User1SD/user1/points",
728     "qos": "0",
729     "datatype": "auto",
730     "broker": "9293df66.3b8c9",
731     "x": 340,
732     "y": 1000,
733     "wires": [
734       [
735         "49e3ff02.48e2"
736       ]
737     ],
738   },
739   {
740     "id": "9293df66.3b8c9",
741     "type": "mqtt-broker",
742     "name": "AWS",
743     "broker": "a2uci4q0y80ahc-ats.iot.eu-west-2.
amazonaws.com",
744     "port": "8883",
745     "tls": "f8007a67.e3a5a8",
746     "clientid": "",
747     "useTLS": true,
748     "compatmode": false,
749     "keepalive": "60",
750     "cleansession": true,
751     "birthTopic": "",
752     "birthQos": "0",
753     "birthPayload": "",
754     "closeTopic": "",
755     "closeQos": "0",
756     "closePayload": "",
757     "willTopic": ""

```

```
758     "willQos": "0",
759     "willPayload": ""
760   },
761   {
762     "id": "f8007a67.e3a5a8",
763     "type": "tls-config",
764     "name": "",
765     "cert": "",
766     "key": "",
767     "ca": "",
768     "certname": "af117cc32c-certificate.pem.crt",
769     "keyname": "af117cc32c-private.pem.key",
770     "caname": "AmazonRootCA1.pem",
771     "servername": "xyz",
772     "verifyservercert": false
773   }
774 ]
```

Listing 17: Final json file

Appendix C:

Date	Length	Sub-team	Agenda
10/02/21	0:19:56	All	Welcome meeting, getting to know each other, initial thoughts
11/02/21	0:40:00	All	Discussing initial thoughts and ideas, getting a more concrete idea of what our solution will be
			Think on ideas more: What exercises do you want? Which sensors would you need for this?
12/02/21	1:27:08	All	What ideas do you have about existing/new ideas?
			How should the points work based on the sensors and the data? How are we going to get data from the hardware to the software? Draw out the design we want and see what people want to work on.
13/02/21	0:55:49	All	Worked on: initial abstract first draft and thoughts on schematic overview
			Technical specs Splitting people into sub groups
14/02/21	0:51:29	All	Completing a first draft of the Initial design form
15/02/21	1:00:23	Website team	Talking through sub-projects, basic design ideas, functionality of which IDE to use
15/02/21	0:30:34	All	Discussing and finalising initial design form, talking about funds and biscope vs box distribution, deciding team leader
16/02/21	1:33:00	Hardware team	Discussing which sensors and microcontrollers to use
16/02/21	0:27:33	All	Further discussion on sensors
16/02/21	1:16:21	Software/Website team	Discussing how the Database/server can be designed, doing initial designs of webpages
17/02/21	0:36:45	Hardware team	Deciding what microprocessor to use and allocating people to what sensor to work on
18/02/21	1:21:49	Hardware team	Found devices and components to fill out purchase approval request form
18/02/21	1:50:56	Hardware team	Finalisation of purchase approval request form
19/02/21	0:21:27	Hardware team	Discussion of power and what form we need to fill out next
			Testing out remote raspberry pi WAN website access, finding graphics and deciding on how the tree avatar growth will work
19/02/21	1:32:21	Website team	

20/02/21	1:54:58	All	Filling out Project Proposal Form (Form 2) and Risk Assessment Form (Form 6)
20/02/21	0:58:34	Website team	Setting up git for sharing code, checking access to files and planning work for coding the tree and garden images
21/02/21	1:13:08	All	Filling out Form 6, continuing Form 2 and starting to plan for Form 3
21/02/21	0:45:39	James, Harry, Kesavaram	Discussing plan for accelerometer and distributing tasks
22/02/21	0:28:04	All	Starting to look at Form 3, catch up on progress with other forms
22/02/21	1:15:03	Website	Filling out website parts of forms, talking about server integration plans, planning next parts of website design
23/02/21	0:47:19	All	Talk about progress with pseudocode/circuit diagrams/gpio pins for esp sections
23/02/21	0:40:28	Server (Harry, Benjamin, George, Milly)	Talked over server ideas and using AWS, talked about integration and website hosting on the server
24/02/21	1:00:42	All	Adding in more detail to forms 2 and 3
24/02/21	0:27:10	Website	Talking through new designs of CompanyDonation.html, talking through data handling ideas
24/02/21	0:31:32	Timer	Discussed about the division of work for timer, pseudocodes and having ESP32 library installed on Arduino IDE.
25/02/21	1:03:00	All, then integration team	Reviewing forms for hand in, the integration team integrate the circuit diagrams and pseudocodes
26/02/21	0:49:40	All	Reviewing form 3 after meeting with Steve Gunn and adding more detail to the milestones
27/02/21	1:13:36	All, then integration	Discussing plans for development to begin
28/02/21	0:13:10	All	Last full group meeting until after individual development is completed. Decided to have sub-team meetings between now and Wednesday
28/02/21	0:11:37	Website	Reviewing how server will interact with data, deciding to create individual html tree pages, deciding to not include login details
29/02/21	0:09:45	Website	Server progress update
02/03/21	1:51:50	Server/Website	Making progress with the server sending data to the website and implementing HTML code within the node red application
03/03/21	0:12:47	All	Updates on progress, checking everybody has completed individual hardware, looking to plans to integrate into two devices.
3/3/2021	0:30:00	Harry, Przemek	Starting heart rate monitor integration

3/3/21	2:13:33	Server & Website	Continuing integrating website to the server
4/3/21	45min	Harry, Kesavaram, Ben	Pedometer and Hall effect integration
4/3/21	1h	Harry, Ben, kesavaram, James, Przemek	Integration of code and defining interfaces
5/3/21	0:11:06	All	Talking about milestones and plans to get them checked off
06/03/21	0:01:07	Kesavaram,Harry	Discussed about the pedometer and the working of the analog circuitry for it.
6/3/21	2:14:00	Harry, Ben	Testing side by side use of pub and sub to AWS_IOT
7/3/21	0:21:22	Harry, Ben, James	Integration of entire Secondary Device and server
7/3/21	2:50:00	Harry, Ben	Milestone video of full integration
8/3/21	0:56:26	All	Splitting into teams for website and trade fair slides, initial discussions as to what we want to include
9/3/21	0:57:00	All	Discussing plans for video and branding and trade fair slides
09/03/21	1:36:00	Video team	Making script for video, shot by shot, deciding who is doing what, planning for filming tomorrow
10/3/21	1:30:00	Harry, Ben, George	Filming joint shots for video
10/3/21	1:13:00	Video team	Discussing filming shots and who is editing what
14/03/21	1:59:09	All	Filling out Form 5 (Project Completion Form)
14/03/21	0:30:00	Kesavaram,Yevin and Hock.	Discussing the distribution of effort among the timer group and Kesavaram.
15/03/21	0:19:24	All	Distributing sections for the team report