Student name: Ben Marriner

Student ID: 220253518

SIT123: Data Capture Technologies

Lab Report Week 1: Arduino Blink

Welcome to Arduino!

Arduino is an electronic prototyping platform. Different types of sensors & actuators can be attached to Arduino boards to create our own sensing-thinking-acting systems.

Throughout this unit, we will use Arduino to create different sensing devices, and to retrieve the collected sensor data.

In Week 1, we will try out an introductory exercise, to learn the basic concepts of Arduino.

Hardware Required

Arduino Board with in-built LED

USB cable

Software Required

Arduino programming environment

Pre-requisites: You must do the following before this task

Class (Lecture)
The labs are built on concepts we discuss in class (lecture). To be able to carry out the lab tasks, you need to know the ideas introduced in the lecture. In addition, the lab tasks are also explained in the lecture. The Lectures are available as PDF and video files on unit website. If you come to the lab without watching that week's lecture, you will be in a difficult situation. You must watch that week's lecture before coming to the studio.

Reading/Videos

Some labs will have required reading material and/or videos, which you must read/view **BEFORE** you start the lab.

Why should you read/watch pre-lab materials?

These materials will help you understand the background which the lab tasks require. Students come to university from diverse backgrounds. Some of you may be familiar with the background information, some of you may not. When you come to the lab prepared, you're already equipped with confidence and will be able to participate in activities better. Ultimately, class time will be much more productive, dynamic, and fun for everyone.

Here are the pre-lab materials for our first task:

1. Watch TED Talk: https://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination#t-1114 (~15 minutes)
2. Watch https://www.lynda.com/Arduino-tutorials/Creating-your-first-sketch/783858/5015739-4.html (~3 minutes)
3. Read this task sheet from beginning to end.

Task Objective

- "We have an Arduino board with an in-built LED light. We need the LED light to be turned on and off continuously, every one second."

Task Submission Details

There are six questions in this task. Answer all of them in this word document itself and submit to unit site as part of **Lab Report 1** in Week 3.

Q1: The TED talk given under the Pre-Lab materials, shows how Arduino is being used for interesting projects to capture data from the environment, process it, and use it carry out useful actions.

Fill the given table below to answer the following:

What are three projects that use captured data as given in the TED talk? What data do they capture? What sensors do you think they could use to capture this data?

Project name	Data captured	Sensors to capture the data
txtBomber	• Type of wall surface • Speed/direction of movement	• Motion sensor
PS3 Controller	• Movement coming from a	• Motion sensor

	person's limbs	
Botanicalls	• The moistness of the soil	• Soil moisture sensor

Q2: Consider the given Task Objective. Think about how this simple system can be decomposed to 'Sense-Think-Act' as discussed in class (lecture).

a) What is the 'sensing' requirement in this system, if any?
   i) Counting the number of milliseconds passed between each blink
b) What is the 'thinking' requirement in this system, if any?
   i) Wait until a second has passed
c) What is the 'acting' requirement in this system, if any?
   i) Turn the LED on or off

Q3: Please refer to the provided 'Arduino Blink Activity Sheet' and follow the steps.

a) In Arduino-speak, what is a "sketch"?
   i) A sketch a document that contains all the code for the program to executed on the Arduino
b) setup() and loop() are key Arduino constructs. These are required in every Arduino sketch.
   i) Which of the above two, runs once at the very beginning of your program and never again (unless you reset or upload new code)?
      1) Setup()
   ii) Which of the above two, is used to continuously run code over and over again?
      1) Loop()
c) What does **pinMode()** do?
   i) You can set a pin to either input or output signals with this function

   Hint: http://arduino.cc/en/Reference/HomePage
d) What is a comment?
   i) A comment is a line that is not meant to be read by the compiler but only by programmers. It is intended to be used as a means of internal documentation of the code
e) What does the following line of code do:

```
delay(x);
```

   i) This function will make the Arduino wait an 'x' number of milliseconds before executing the next instruction

   Hint: http://arduino.cc/en/Reference/HomePage

f) There is something you need to check before uploading your sketch. What is this?
   i) You need to make sure your code does not have any errors which can be done by hitting the verify button. Also make sure the Arduino IDE is pointed to the correct COM port that the board is connected to

Q4: How can you test the Blink program to make sure it is working as given in the Task Objective?

Check to see that the LED located at pin 13 is blinking on and off every one second.

Q5: Now that you have built and tested your Blink program, it is time to deliver it (hand it over). Take a five second video of your Arduino board with the LED blinking (use your phone to record) and upload it to youtube. Include the link here. Alternatively, if you are on campus, show your working project to your tutor in the lab and get it marked.

https://youtu.be/7dk-d4Tm6ww

Q6: The Morse code is a method of transmitting text information as a series of on-off lights, or clicks.

a) Create a new Arduino project named ‘BlinkSOS’. Copy and paste your code from the Blink example to the newly created project. Modify the code in the new project, to send an SOS signal in Morse Code via turning the LED on and off.
   Upload the ‘BlinkSOS.ino’ file with this document to cloud Deakin.

[Some helpful hints have been provided for you at the end of this document]

b) How did you test your code to make sure it is working correctly?
   i) If the LED blinks the same sequence as the SOS morse code indicates, then it is working correctly

c) Take a video of your Arduino board running ‘BlinkSOS’ program and upload it to youtube. Include the link here. Alternatively, if you are on campus, show your working project to your lecturer/tutor in the lab and get it marked.

   i) https://youtu.be/4tcXm_K1FSI

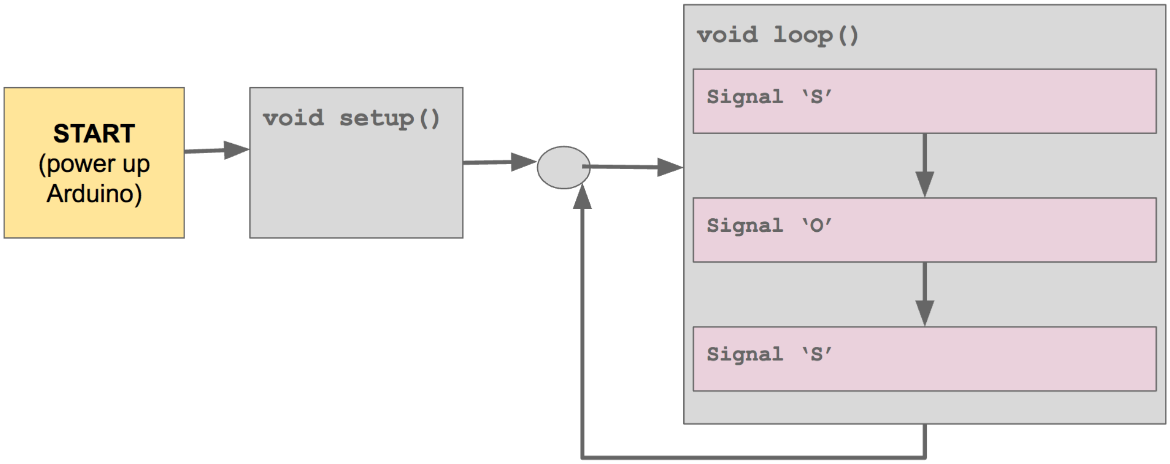
Remember to submit this to cloud Deakin under the correct Assignment folder.

Hints for Q6:

SOS signal in Morse Code: https://www.youtube.com/watch?v=GnHv7h_5P9M
Use the International Morse code given here:
https://en.wikipedia.org/wiki/Morse_code#/media/File:International_Morse_Code.svg
More information about Morse Code: https://en.wikipedia.org/wiki/Morse_code
Here is a sample code snippet signalling the letter ‘S’ below:

```
/**
   * First signal 'S'
   * Morse code for S is - - -
   * that is, three short blinks.
   */
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for half a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for half a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for half a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                 // wait for a second
```

Use your knowledge from the first task, and lecture to decide where the above should go in your code. You now have ‘S’. Next you must write for letters ‘O’ and again ‘S’.



SIT123: Data Capture Technologies

Lab Report Week 2: Getting started with Sensors

In this task, we will attach sensors to the Arduino board, and collect sensor data. Similar to last week, we will use software engineering methods to plan and organise our exercise.

Hardware Required

Arduino Board

USB cable

HCSR505 PIR (Passive Infra Red) Motion Detector

DHT22 Temperature and Humidity Sensor

DFrobot Soil Moisture Sensor

Male to Female Dupont Jumper Wires

Male to Male Dupont Jumper Wires

Software Required

Arduino programming environment

Pre-requisites: You must do the following before this task

1. **Watched the lecture video**
2. **Read this sheet from top to bottom**

Task 1 - Objective

For this task, your tutor/lecturer will be your client. Here are your client's requirements:

- **"We have an Arduino board and some sensors. We need to be able to measure air temperature, humidity, motion, and soil moisture, and see the collected data in real-time on the computer screen."**

Task 1 - Submission Details

There are 4 questions in this task. Answer all of them in this word document itself.

Q1: Consider the given Task Objective. Think about how this simple system can be decomposed to 'Sense-Think-Act' as discussed in class (lecture).

a) What is the 'sensing' requirement in this system, if any?

Sense motion, temperature, humidity and soil moisture changes

b) What is the 'thinking' requirement in this system, if any?

The Arduino board processes the data and outputs it onto the screen

c) What is the 'acting' requirement in this system, if any?

N/A

Q2: PIR Motion Detector

Please refer to the provided '**Sensing Motion Activity Sheet**' and follow the steps.

a) Refer to the given code in HCSR505motion.ino. What does the following line mean?

```
Serial.begin(9600);
```

This procedure sets the baud rate of the data transmission. In this case, we are setting the baud rate to 9600bps.

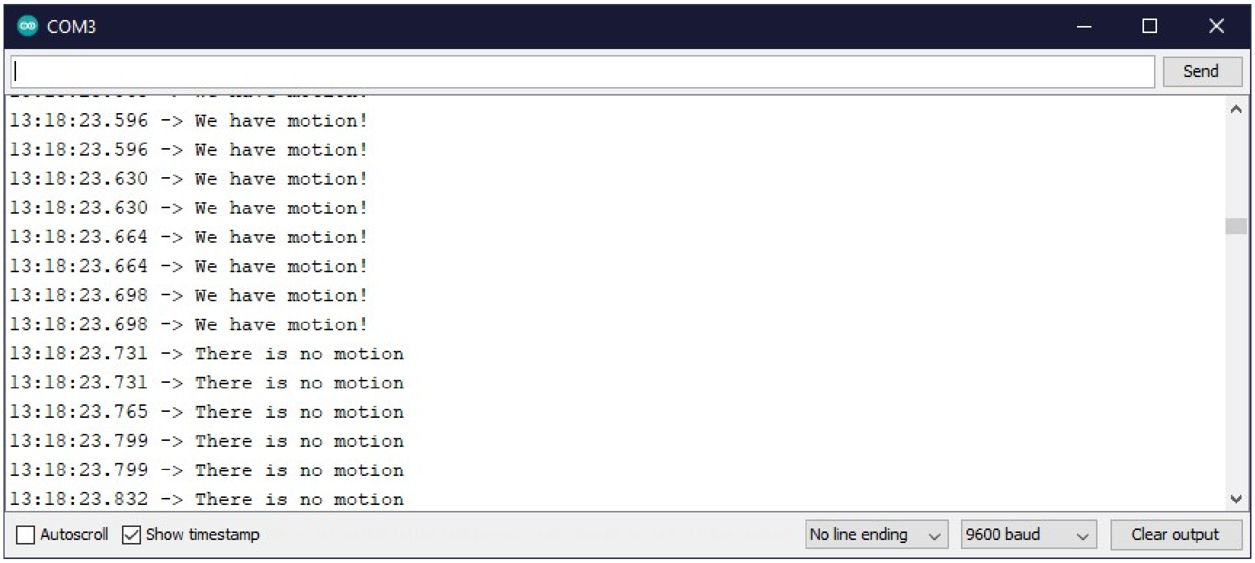
b) If the Arduino transfers data at 4800 bits per second and you're sending 12 bytes of data, how long does it take to send over this information?

It would take 20 milliseconds to send over the information

c) What is a simple strategy to test this program to make sure it is working as given in the requirements?

Print messages in the serial monitor to determine if the sensors are working correctly

d) Take a screenshot of your Serial Monitor displaying motion data logs. Add the image here.



```
13:18:23.596 -> We have motion!
13:18:23.596 -> We have motion!
13:18:23.630 -> We have motion!
13:18:23.630 -> We have motion!
13:18:23.664 -> We have motion!
13:18:23.664 -> We have motion!
13:18:23.698 -> We have motion!
13:18:23.698 -> We have motion!
13:18:23.731 -> There is no motion
13:18:23.731 -> There is no motion
13:18:23.765 -> There is no motion
13:18:23.799 -> There is no motion
13:18:23.799 -> There is no motion
13:18:23.832 -> There is no motion
```

☐ Autoscroll ☑ Show timestamp No line ending 9600 baud

e) Run your program for three minutes. In that time, make sure the sensor can detect 'Active' as well as 'Inactive' data by creating some movement for it to detect. Retrieve the collected data as text file and save it your computer's hard drive, naming it 'lab2_motionData.txt'. Upload 'lab2_motionData.txt' with this lab report.

Q3: Temperature and Humidity Sensor

a) Please refer to the provided '**Sensing Temperature and Humidity Activity Sheet**' and follow the steps. Consider the given code in the activity sheet and fill the table below. The first row is completed for you.

term	explanation	example usage from code
variable	A variable is a place to store a piece of data. It has a name, a value, and a type.	float temp;

library	A library contains code pre-made code that the programmer can use in their projects. Libraries have code that is specifically meant for working with other components such as SD cards and sensors.	#include <DHT.h>
		Dht.begin();

comment	Comments are used as internal documentation for the programmer within a sketch. These are ignore by the compiler	// I am a comment

	when verifying and uploading sketches	

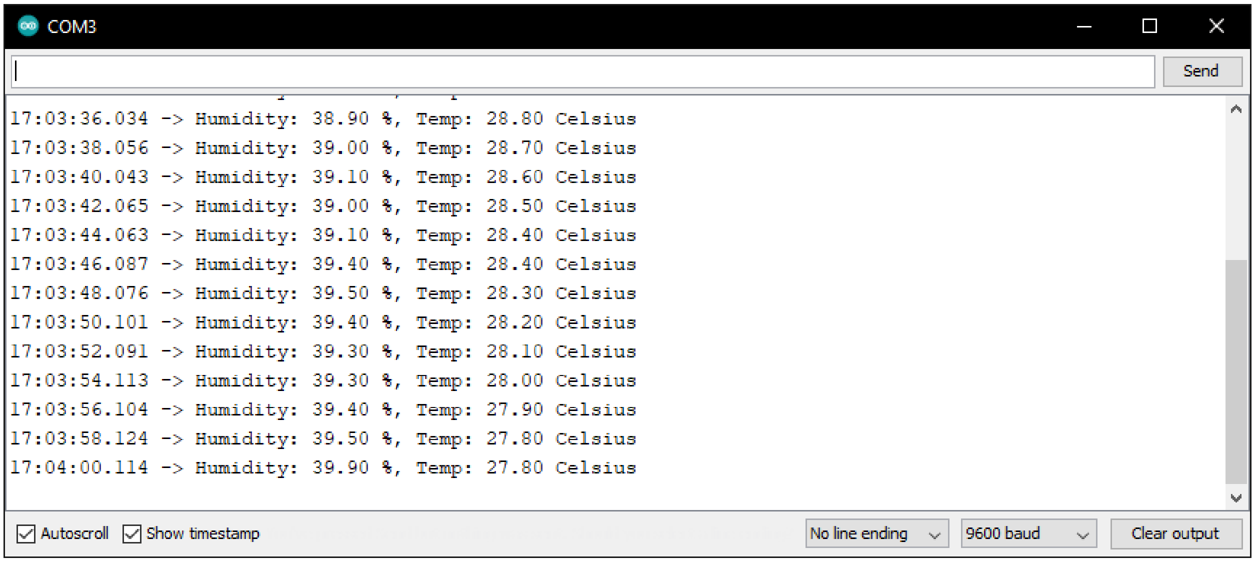
b) A spec of the DHT22 sensor is given in the link below. It mentions that the sampling rate is 0.5 Hz.
https://tronixlabs.com.au/sensors/humidity/dht22-temperature-and-humidity-sensor-australia/

i) What does the sampling rate mean?
The rate at which the sensor samples data

ii) Where is this used in the Arduino code?
Line 17: Serial.begin(9600);

c) What is a simple strategy to test this program to make sure it is working as given in the requirements?
Use the serial monitor to check that the sensor is reading humidity and temperature data

d) Take a screenshot of your Serial Monitor displaying temperature & humidity sensor data logs. Add the image here.



```
17:03:36.034 -> Humidity: 38.90 %, Temp: 28.80 Celsius
17:03:38.056 -> Humidity: 39.00 %, Temp: 28.70 Celsius
17:03:40.043 -> Humidity: 39.10 %, Temp: 28.60 Celsius
17:03:42.065 -> Humidity: 39.00 %, Temp: 28.50 Celsius
17:03:44.063 -> Humidity: 39.10 %, Temp: 28.40 Celsius
17:03:46.087 -> Humidity: 39.40 %, Temp: 28.40 Celsius
17:03:48.076 -> Humidity: 39.50 %, Temp: 28.30 Celsius
17:03:50.101 -> Humidity: 39.40 %, Temp: 28.20 Celsius
17:03:52.091 -> Humidity: 39.30 %, Temp: 28.10 Celsius
17:03:54.113 -> Humidity: 39.30 %, Temp: 28.00 Celsius
17:03:56.104 -> Humidity: 39.40 %, Temp: 27.90 Celsius
17:03:58.124 -> Humidity: 39.50 %, Temp: 27.80 Celsius
17:04:00.114 -> Humidity: 39.90 %, Temp: 27.80 Celsius
```

☑ Autoscroll ☑ Show timestamp No line ending 9600 baud

e) Run your program for five minutes. Retrieve the collected data as text file and save it your computer's hard drive, naming it 'lab2_temperatureData.txt'. Upload 'lab2_temperatureData.txt' with this lab report.

Q4: Soil Moisture Sensor

Please refer to the provided '**Sensing Soil Moisture Activity Sheet**' and follow the steps.

a) Refer to the given code in DFRobotSoilMoisture.ino. What does the following line do?

```
val = analogRead(0);
```

This line gets the board to read signals coming in from an analog pin.

b) How is `analogRead` different than `digitalRead`?

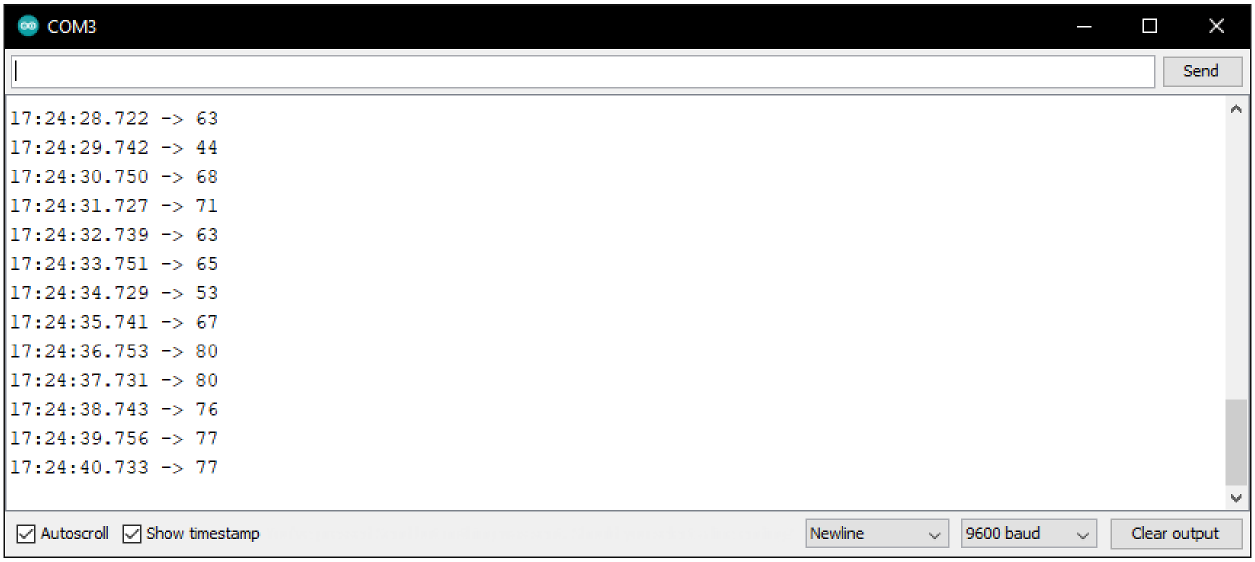
[Hint: we used `digitalRead` in the code for HCSR505 PIR Motion detector]

analogRead can read values between 0 and 1023 whereas digitalRead can only read values 1 or 0.

c) What is a simple strategy to test this program to make sure it is working as given in the requirements?

Use the serial monitor to check that the sensor is reading moisture data

d) Take a screenshot of your Serial Monitor displaying soil moisture sensor data logs. Add the image here.



```
17:24:28.722 -> 63
17:24:29.742 -> 44
17:24:30.750 -> 68
17:24:31.727 -> 71
17:24:32.739 -> 63
17:24:33.751 -> 65
17:24:34.729 -> 53
17:24:35.741 -> 67
17:24:36.753 -> 80
17:24:37.731 -> 80
17:24:38.743 -> 76
17:24:39.756 -> 77
17:24:40.733 -> 77
```

☑ Autoscroll ☑ Show timestamp Newline 9600 baud

e) Run your program for three minutes. Experiment testing the sensor in the air, in water & wet tissue. Retrieve the collected data as text file and save it your computer's hard drive, naming it '**lab2_soilMoistureData.txt**'. Upload '**lab2_soilMoistureData.txt**' with this lab report.

Important: When you are finished, gently unplug the jumper cables from the Arduino pins and the sensor pins.

Task 2 - Objective

1. Use the provided dataset HumidityDataset.CSV on collection of Humidity values for location X. The file is available here: https://drive.google.com/open?id=0B6VXEzM81LStczdRUjlnWjNEOEU
2. Investigate the data for inconsistencies. These inconsistencies could include:

   a. Missing data, rows and column values
   b. Mismatched data fields
   c. Mismatched date formats
3. Propose ways to fix consistencies, these fixes could include:
   a. Propose and use default values
   b. Remove missing rows
   c. Fix data format mismatches
4. Fix the data

Task 2- Submission Details

There are 2 questions in this task. Answer all of them in this document itself.

Q1: Once you have cleaned your data, submit the cleaned data file to unit site with this document.

Q2: Submit brief details on which inconsistencies you have found, what was your approach for fixing them and discuss the Pros. and Cons of your approach, using the given table below:

Inconsistencies found	Approach for fixing	Pros. and Cons of your approach
Some timestamps were not formatted correctly	Converted all the timestamps using text to format in Excel. Converted to dates to date format and times to time format. Split times into its own column	**Pros:** Timestamps and dates are now properly formatted and readable **Cons:** The dates format is slightly different but still the same nonetheless.
Some humidity levels were represented incorrectly or had an invalid value	Rows were removed	**Pros:** Rows of invalid data removed Incorrect representations fixed **Cons:** Removed data rows will no longer contribute to analysis of data.

One row was left completely blank	Row was removed	**Pros:** Row removed without affecting other data **Cons:** N/A

(You may add more rows to the table as required)

Student name:

Student ID:

SIT123: Data Capture Technologies

Lab Report Week 3: Using the Data Logger Shield in Arduino

In this task, we will learn about using an SD card to save sensor data.

Pre-requisites: You must do the following before this task

1. **Watched the Lecture**
2. **Already done with Lab report of Week 2**
3. **Read https://learn.adafruit.com/adafruit-data-logger-shield/using-the-real-time-clock**
4. **Read this sheet from top to bottom**

Task 1 - Objective

In this task, you will write a program to detect motion and save the detected motion readings to an SD card, using a data logger shield.

Hardware Required

Arduino Board

USB cable

SD Card (SanDisk 16Gb Ultra SDHC Memory Card)

Data Logging Shield for Arduino

CR1220 Coin Cell Battery

HCSR505 PIR Passive Infra Red Motion Detector

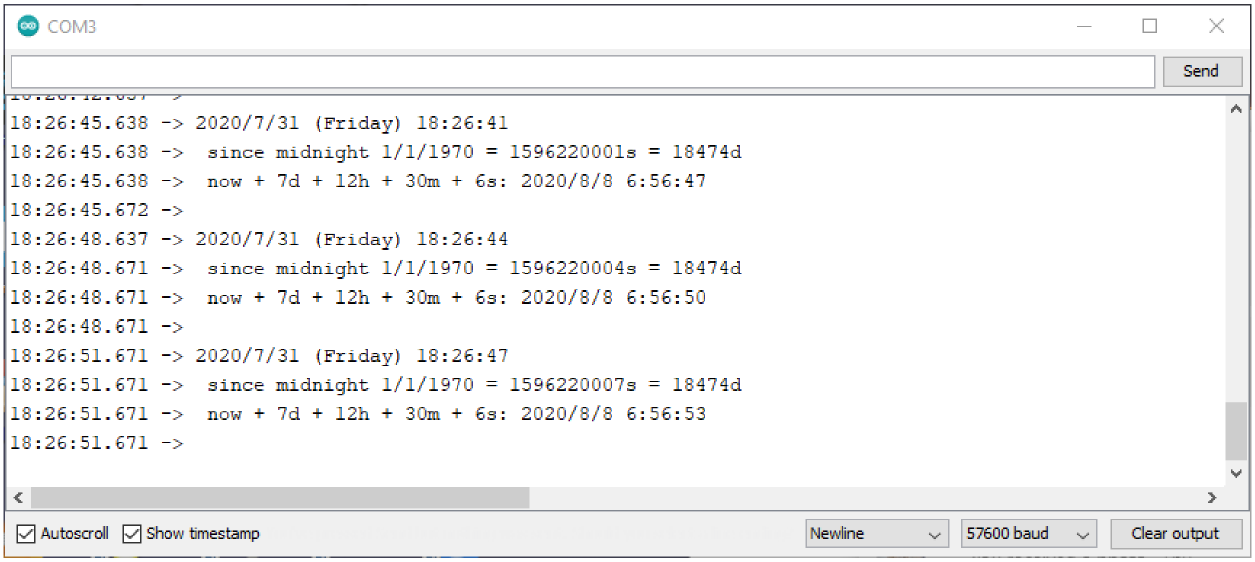
Task 1 - Submission Details

There are 3 questions in this task. Answer all of them in this word document itself and submit to unit site.

Q1. Follow the steps in “Setting Up The SD Card Activity Sheet”. At the end of activity, take a screenshot of the Serial Monitor and include here.

Q2. Follow the steps in “Using the Real Time Clock Activity Sheet”.

a. At the end of activity, take a screenshot of the Serial Monitor and include here.



```
18:26:45.638 -> 2020/7/31 (Friday) 18:26:41
18:26:45.638 ->  since midnight 1/1/1970 = 1596220001s = 18474d
18:26:45.638 ->  now + 7d + 12h + 30m + 6s: 2020/8/8 6:56:47
18:26:45.672 -> 
18:26:48.637 -> 2020/7/31 (Friday) 18:26:44
18:26:48.671 ->  since midnight 1/1/1970 = 1596220004s = 18474d
18:26:48.671 ->  now + 7d + 12h + 30m + 6s: 2020/8/8 6:56:50
18:26:48.671 -> 
18:26:51.671 -> 2020/7/31 (Friday) 18:26:47
18:26:51.671 ->  since midnight 1/1/1970 = 1596220007s = 18474d
18:26:51.671 ->  now + 7d + 12h + 30m + 6s: 2020/8/8 6:56:53
18:26:51.671 -> 
```

☑ Autoscroll ☑ Show timestamp Newline 57600 baud

b. Examine the code. What does the following line of code do?

`DateTime now = rtc.now();`

(Hint: refer to https://learn.adafruit.com/adafruit-data-logger-shield/using-the-real-time-clock)

Returns the current date and time of day being reported by the logger

Q3. Now you are ready to start logging data to file! Follow the steps in “Saving Motion Data Activity Sheet”.

a. At the end of activity, take a screenshot of the Serial Monitor and include here.

b. Run your program. Wave your hand in front if the motion sensor and observe the ‘Active’ state, then stop and wait until you see an ‘Inactive’ state on the Serial Monitor. Keep doing this for three minutes so that you get both ‘Active’ and ‘Inactive’ data. At the end of three minutes, unplug the USB. This will switch off the Arduino board. Next, retrieve the .csv file containing motion sensor data from the SD card. Upload the .csv file with this report to unit site.

References

https://learn.adafruit.com/adafruit-data-logger-shield/using-the-sd-card

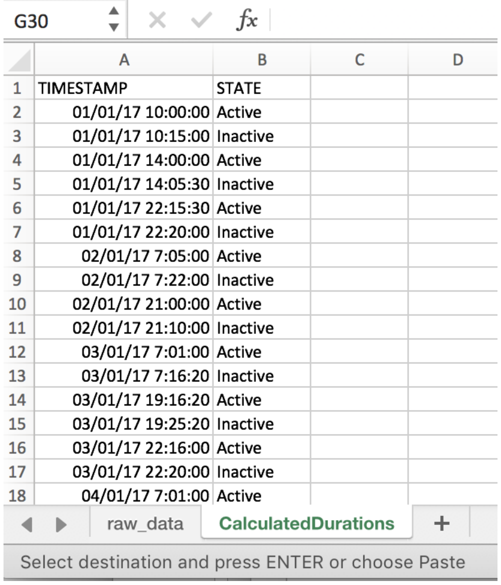
Task 2 - Overview

Raw data is given in the Excel file. Explanation about the data in the Excel file is given in the data document file “Data document.pdf”.
We will be using the following key steps to execute this task.

Step 1 - Read the provided Data document and inspect the provided raw data in the Excel file
Step 2 - Calculate the time durations for each bathroom visit
Step 3 - Calculate descriptive statistics for the data set using Excel’s built-in functions, to find out some useful information about John’s bathroom usage, such as the average time he spends in the bathroom per visit.

Step-by-step Instructions: Calculate the time durations for each bathroom visit

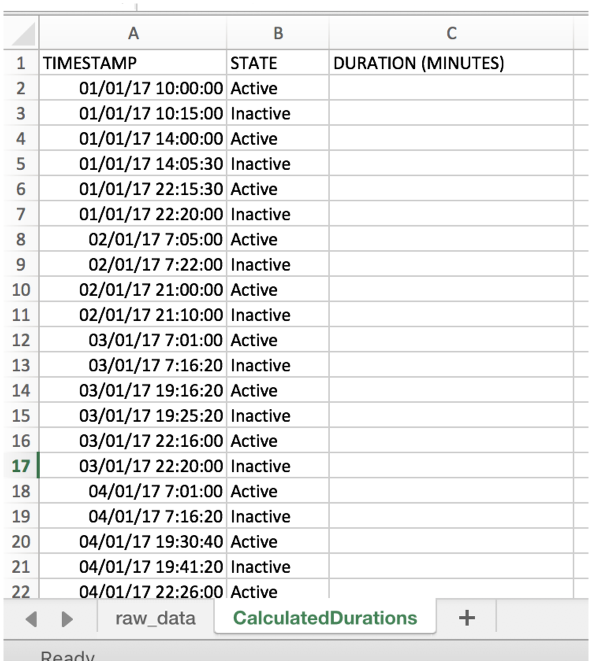
1. Open the provided Excel file. You are now going to use the given data to calculate time durations for each bathroom visit. To do this, first insert new sheet. Name it ‘CalculatedDurations’
2. Copy the two columns TIMESTAMP & STATE from the first sheet to the newly created sheet



	A	B	C	D
1	TIMESTAMP	STATE		
2	01/01/17 10:00:00	Active		
3	01/01/17 10:15:00	Inactive		
4	01/01/17 14:00:00	Active		
5	01/01/17 14:05:30	Inactive		
6	01/01/17 22:15:30	Active		
7	01/01/17 22:20:00	Inactive		
8	02/01/17 7:05:00	Active		
9	02/01/17 7:22:00	Inactive		
10	02/01/17 21:00:00	Active		
11	02/01/17 21:10:00	Inactive		
12	03/01/17 7:01:00	Active		
13	03/01/17 7:16:20	Inactive		
14	03/01/17 19:16:20	Active		
15	03/01/17 19:25:20	Inactive		
16	03/01/17 22:16:00	Active		
17	03/01/17 22:20:00	Inactive		
18	04/01/17 7:01:00	Active		

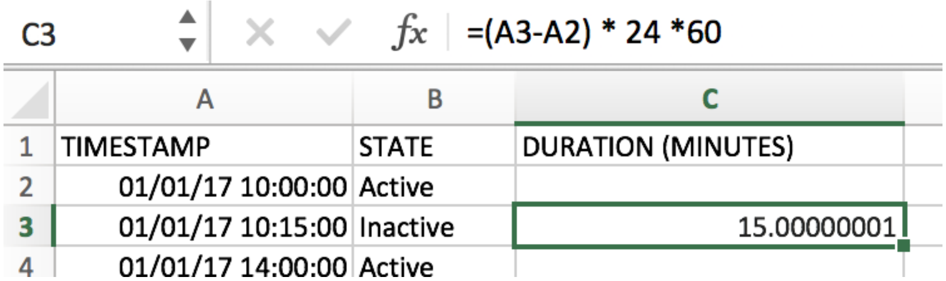
Select destination and press ENTER or choose Paste

3. Add a new column with header ‘DURATION (MINUTES)’ to the right.



	A	B	C
1	TIMESTAMP	STATE	DURATION (MINUTES)
2	01/01/17 10:00:00	Active	
3	01/01/17 10:15:00	Inactive	
4	01/01/17 14:00:00	Active	
5	01/01/17 14:05:30	Inactive	
6	01/01/17 22:15:30	Active	
7	01/01/17 22:20:00	Inactive	
8	02/01/17 7:05:00	Active	
9	02/01/17 7:22:00	Inactive	
10	02/01/17 21:00:00	Active	
11	02/01/17 21:10:00	Inactive	
12	03/01/17 7:01:00	Active	
13	03/01/17 7:16:20	Inactive	
14	03/01/17 19:16:20	Active	
15	03/01/17 19:25:20	Inactive	
16	03/01/17 22:16:00	Active	
17	03/01/17 22:20:00	Inactive	
18	04/01/17 7:01:00	Active	
19	04/01/17 7:16:20	Inactive	
20	04/01/17 19:30:40	Active	
21	04/01/17 19:41:20	Inactive	
22	04/01/17 22:26:00	Active	

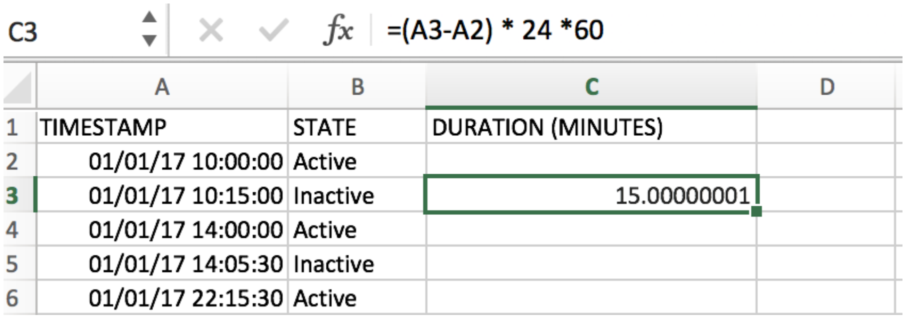
4. Now, what we want to do is to calculate how much time John spent in the bathroom per each visit. To do this, we need to find out the time difference between each 'Active' state and the next 'Inactive' state.
   The first visit in the given data starts at 10 am and ends at 10:15 am. Let's calculate the duration of this first visit.
   The duration of the first visit is the time difference between 10:15 am and 10:00 am. To calculate this in Excel, we can write a simple formula.
   Click on cell C3 and type the following:
   =(A3-A2)
   What is displayed is the time difference, but you'll notice that it is not in minutes! To get the answer in minutes, you need to multiply this by 24 * 60. This is because there are 24 hours in a day, and 60 minutes to each hour. Modify the formula in C3 to be:
   =(A3-A2) * 24 *60
   Press Enter.
   You should now see 15 as the answer. As you can see, the time difference between 10:15 am and 10:00 is indeed 15 minutes, so our formula is correct!



C3 =(A3-A2) * 24 *60

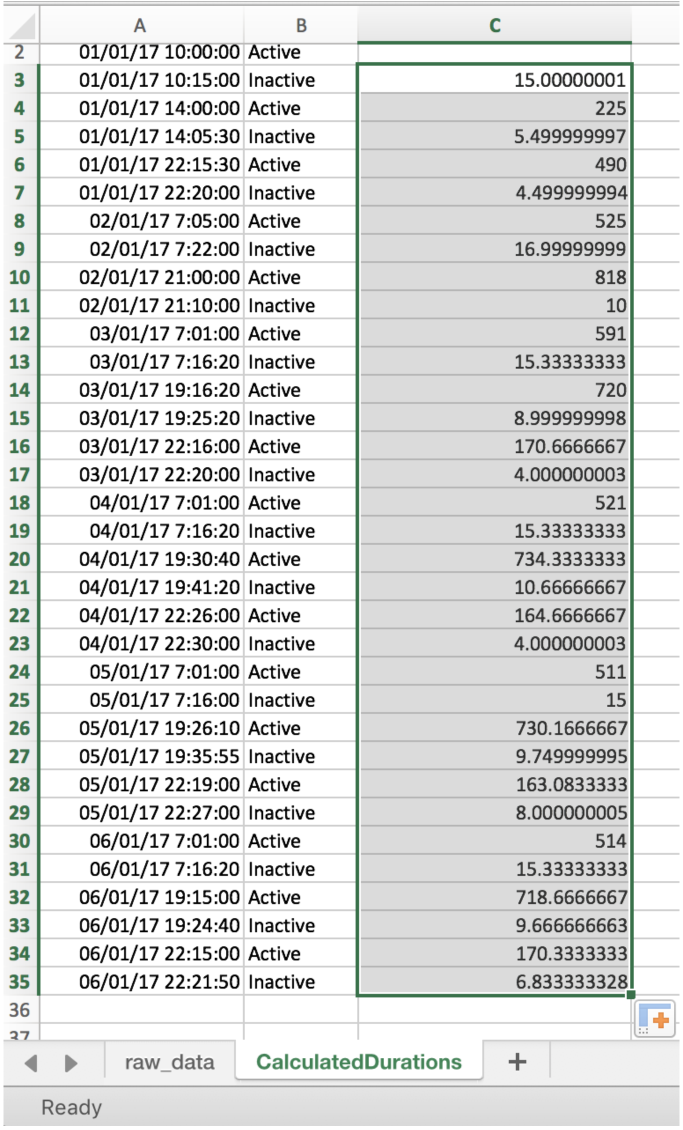
	A	B	C
1	TIMESTAMP	STATE	DURATION (MINUTES)
2	01/01/17 10:00:00	Active	
3	01/01/17 10:15:00	Inactive	15.00000001
4	01/01/17 14:00:00	Active	

5. Now we need to apply this formula to all of the remaining rows in Column C. Select C3 and hover your pointer to the edge of the C3 cell until you see a cross hair. Then click and drag the pointer to the last cell in the range (C35)



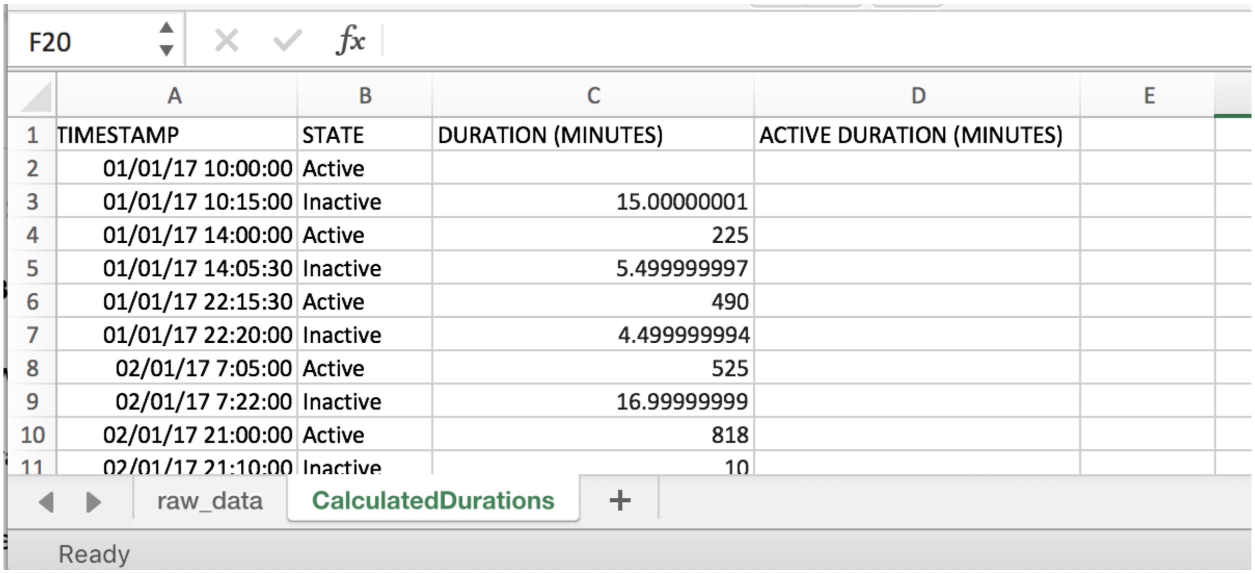
=(A3-A2) * 24 *60

	A	B	C	D
1	TIMESTAMP	STATE	DURATION (MINUTES)	
2	01/01/17 10:00:00	Active		
3	01/01/17 10:15:00	Inactive	15.00000001	
4	01/01/17 14:00:00	Active		
5	01/01/17 14:05:30	Inactive		
6	01/01/17 22:15:30	Active		



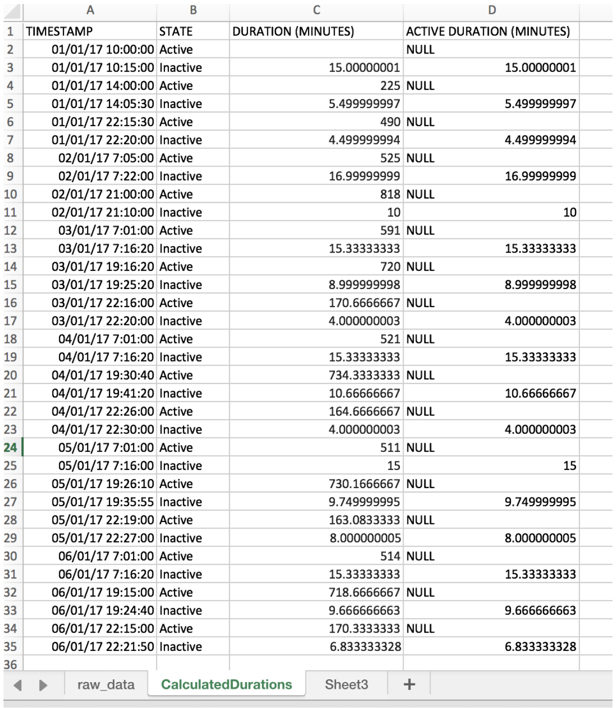
	A	B	C
2	01/01/17 10:00:00	Active	
3	01/01/17 10:15:00	Inactive	15.00000001
4	01/01/17 14:00:00	Active	225
5	01/01/17 14:05:30	Inactive	5.499999997
6	01/01/17 22:15:30	Active	490
7	01/01/17 22:20:00	Inactive	4.499999994
8	02/01/17 7:05:00	Active	525
9	02/01/17 7:22:00	Inactive	16.99999999
10	02/01/17 21:00:00	Active	818
11	02/01/17 21:10:00	Inactive	10
12	03/01/17 7:01:00	Active	591
13	03/01/17 7:16:20	Inactive	15.33333333
14	03/01/17 19:16:20	Active	720
15	03/01/17 19:25:20	Inactive	8.999999998
16	03/01/17 22:16:00	Active	170.6666667
17	03/01/17 22:20:00	Inactive	4.000000003
18	04/01/17 7:01:00	Active	521
19	04/01/17 7:16:20	Inactive	15.33333333
20	04/01/17 19:30:40	Active	734.3333333
21	04/01/17 19:41:20	Inactive	10.66666667
22	04/01/17 22:26:00	Active	164.6666667
23	04/01/17 22:30:00	Inactive	4.000000003
24	05/01/17 7:01:00	Active	511
25	05/01/17 7:16:00	Inactive	15
26	05/01/17 19:26:10	Active	730.1666667
27	05/01/17 19:35:55	Inactive	9.749999995
28	05/01/17 22:19:00	Active	163.0833333
29	05/01/17 22:27:00	Inactive	8.000000005
30	06/01/17 7:01:00	Active	514
31	06/01/17 7:16:20	Inactive	15.33333333
32	06/01/17 19:15:00	Active	718.6666667
33	06/01/17 19:24:40	Inactive	9.666666663
34	06/01/17 22:15:00	Active	170.3333333
35	06/01/17 22:21:50	Inactive	6.833333328
36			
37			

6. Now as you can see we have the active durations in Column C. But, the active durations should be, [Time motion stopped - Time motion started], or in other words [Time motion went inactive - Time motion went active]. But our formula in step 5 applied it to ALL rows. For example, look at the result in C4. That is [Time new motion started - Time previous motion stopped]. Now we need to filter these out, and only keep durations from [Time motion went inactive - Time motion went active].
7. Add a new column header named 'ACTIVE DURATION (MINUTES)' to the right.



	A	B	C	D	E
1	TIMESTAMP	STATE	DURATION (MINUTES)	ACTIVE DURATION (MINUTES)	
2	01/01/17 10:00:00	Active			
3	01/01/17 10:15:00	Inactive	15.00000001		
4	01/01/17 14:00:00	Active	225		
5	01/01/17 14:05:30	Inactive	5.499999997		
6	01/01/17 22:15:30	Active	490		
7	01/01/17 22:20:00	Inactive	4.499999994		
8	02/01/17 7:05:00	Active	525		
9	02/01/17 7:22:00	Inactive	16.99999999		
10	02/01/17 21:00:00	Active	818		
11	02/01/17 21:10:00	Inactive	10		

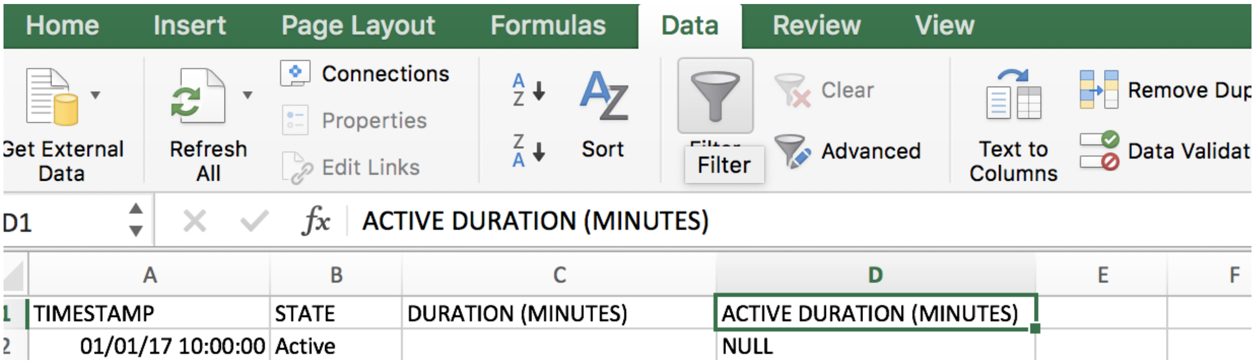
8. Of course, we can manually select which cells contain the Active Durations (time diff between Inactive - Active). But imagine if we had hundreds of rows! So we're going to automate this by making use of Excel.
9. What we want Excel to do is to select the DURATION value from Column C if the corresponding row in Column B contains 'Inactive'.
   To do this, we write this as a formula. Type the following into D2:
   =IF(ISNUMBER(FIND("Inactive",B2)),C2,"NULL")
   Here the formula is saying, IF B2 contains 'Inactive', then use the value in C2, else, insert NULL
10. Apply the above formula to cells D2:D35



	A	B	C	D
1	TIMESTAMP	STATE	DURATION (MINUTES)	ACTIVE DURATION (MINUTES)
2	01/01/17 10:00:00	Active		NULL
3	01/01/17 10:15:00	Inactive	15.00000001	15.00000001
4	01/01/17 14:00:00	Active	225	NULL
5	01/01/17 14:05:30	Inactive	5.499999997	5.499999997
6	01/01/17 22:15:30	Active	490	NULL
7	01/01/17 22:20:00	Inactive	4.499999994	4.499999994
8	02/01/17 7:05:00	Active	525	NULL
9	02/01/17 7:22:00	Inactive	16.99999999	16.99999999
10	02/01/17 21:00:00	Active	818	NULL
11	02/01/17 21:10:00	Inactive	10	10
12	03/01/17 7:01:00	Active	591	NULL
13	03/01/17 7:16:20	Inactive	15.33333333	15.33333333
14	03/01/17 19:16:20	Active	720	NULL
15	03/01/17 19:25:20	Inactive	8.999999998	8.999999998
16	03/01/17 22:16:00	Active	170.6666667	NULL
17	03/01/17 22:20:00	Inactive	4.000000003	4.000000003
18	04/01/17 7:01:00	Active	521	NULL
19	04/01/17 7:16:20	Inactive	15.33333333	15.33333333
20	04/01/17 19:30:40	Active	734.3333333	NULL
21	04/01/17 19:41:20	Inactive	10.66666667	10.66666667
22	04/01/17 22:26:00	Active	164.6666667	NULL
23	04/01/17 22:30:00	Inactive	4.000000003	4.000000003
24	05/01/17 7:01:00	Active	511	NULL
25	05/01/17 7:16:00	Inactive	15	15
26	05/01/17 19:26:10	Active	730.1666667	NULL
27	05/01/17 19:35:55	Inactive	9.749999995	9.749999995
28	05/01/17 22:19:00	Active	163.0833333	NULL
29	05/01/17 22:27:00	Inactive	8.000000005	8.000000005
30	06/01/17 7:01:00	Active	514	NULL
31	06/01/17 7:16:20	Inactive	15.33333333	15.33333333
32	06/01/17 19:15:00	Active	718.6666667	NULL
33	06/01/17 19:24:40	Inactive	9.666666663	9.666666663
34	06/01/17 22:15:00	Active	170.3333333	NULL
35	06/01/17 22:21:50	Inactive	6.833333328	6.833333328
36				

11. OK - now we have column D with only the active values, but now we've got NULL values which we need to filter out. Remember that our goal is to have a list of active duration values. We are going to filter these in the next step.

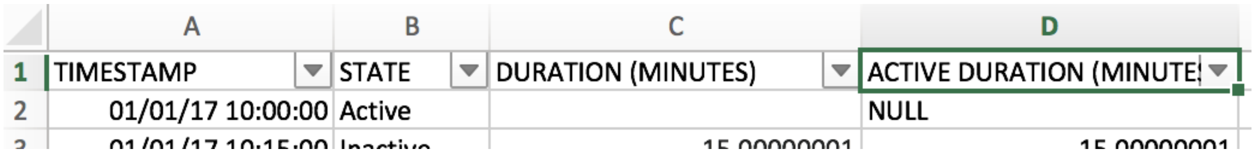
12. Click on D1, and on the Excel Ribbon, under 'Data', click on 'Filter'



D1 fx ACTIVE DURATION (MINUTES)

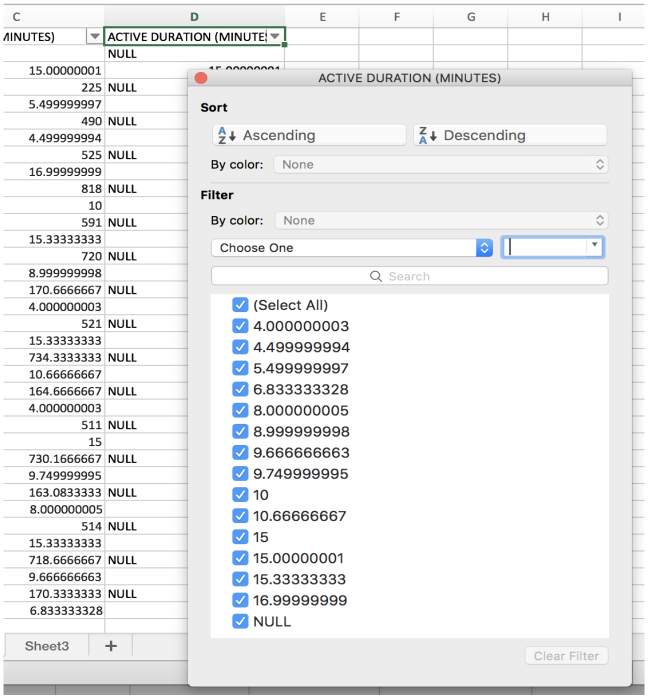
	A	B	C	D	E	F
1	TIMESTAMP	STATE	DURATION (MINUTES)	ACTIVE DURATION (MINUTES)		
2	01/01/17 10:00:00	Active		NULL		

13. Now your column headers should show drop down arrows:



	A	B	C	D
1	TIMESTAMP	STATE	DURATION (MINUTES)	ACTIVE DURATION (MINUTE!
2	01/01/17 10:00:00	Active		NULL
3	01/01/17 10:15:00	Inactive	15.00000001	15.00000001

14. Click on the drop down arrow on D1. It will show a filter dialog box



C	D
/INUTES)	ACTIVE DURATION (MINUTE!
	NULL
15.00000001	
225	NULL
5.499999997	
490	NULL
4.499999994	
525	NULL
16.99999999	
818	NULL
10	
591	NULL
15.33333333	
720	NULL
8.999999998	
170.6666667	NULL
4.000000003	
521	NULL
15.33333333	
734.3333333	NULL
10.66666667	
164.6666667	NULL
4.000000003	
511	NULL
15	
730.1666667	NULL
9.749999995	
163.0833333	NULL
8.000000005	
514	NULL
15.33333333	
718.6666667	NULL
9.666666663	
170.3333333	NULL
6.833333328	

ACTIVE DURATION (MINUTES)

Sort

Ascending Descending

By color: None

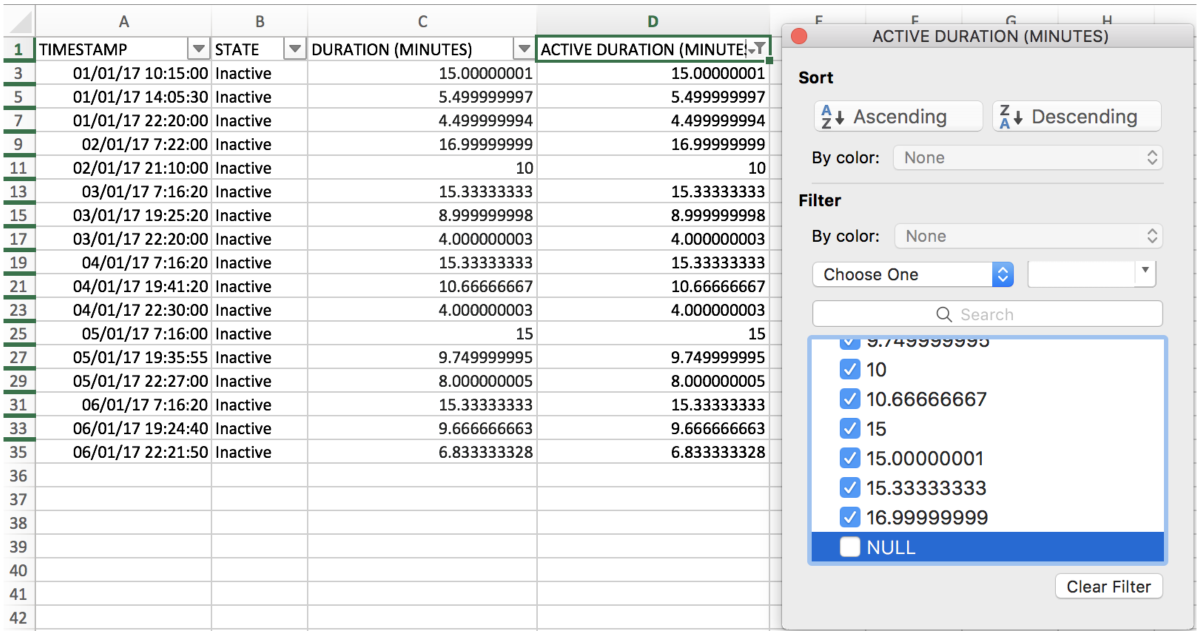
Filter

By color: None

Choose One

☑ (Select All)
☑ 4.000000003
☑ 4.499999994
☑ 5.499999997
☑ 6.833333328
☑ 8.000000005
☑ 8.999999998
☑ 9.666666663
☑ 9.749999995
☑ 10
☑ 10.66666667
☑ 15
☑ 15.00000001
☑ 15.33333333
☑ 16.99999999
☑ NULL

15. Untick the checkbox for 'NULL'. This will immediately filter out the NULL values in Column D.



	A	B	C	D
1	TIMESTAMP	STATE	DURATION (MINUTES)	ACTIVE DURATION (MINUTE
3	01/01/17 10:15:00	Inactive	15.00000001	15.00000001
5	01/01/17 14:05:30	Inactive	5.499999997	5.499999997
7	01/01/17 22:20:00	Inactive	4.499999994	4.499999994
9	02/01/17 7:22:00	Inactive	16.99999999	16.99999999
11	02/01/17 21:10:00	Inactive	10	10
13	03/01/17 7:16:20	Inactive	15.33333333	15.33333333
15	03/01/17 19:25:20	Inactive	8.999999998	8.999999998
17	03/01/17 22:20:00	Inactive	4.000000003	4.000000003
19	04/01/17 7:16:20	Inactive	15.33333333	15.33333333
21	04/01/17 19:41:20	Inactive	10.66666667	10.66666667
23	04/01/17 22:30:00	Inactive	4.000000003	4.000000003
25	05/01/17 7:16:00	Inactive	15	15
27	05/01/17 19:35:55	Inactive	9.749999995	9.749999995
29	05/01/17 22:27:00	Inactive	8.000000005	8.000000005
31	06/01/17 7:16:20	Inactive	15.33333333	15.33333333
33	06/01/17 19:24:40	Inactive	9.666666663	9.666666663
35	06/01/17 22:21:50	Inactive	6.833333328	6.833333328
36				
37				
38				
39				
40				
41				
42				

ACTIVE DURATION (MINUTES)

Sort

Ascending Descending

By color: None

Filter

By color: None

Choose One

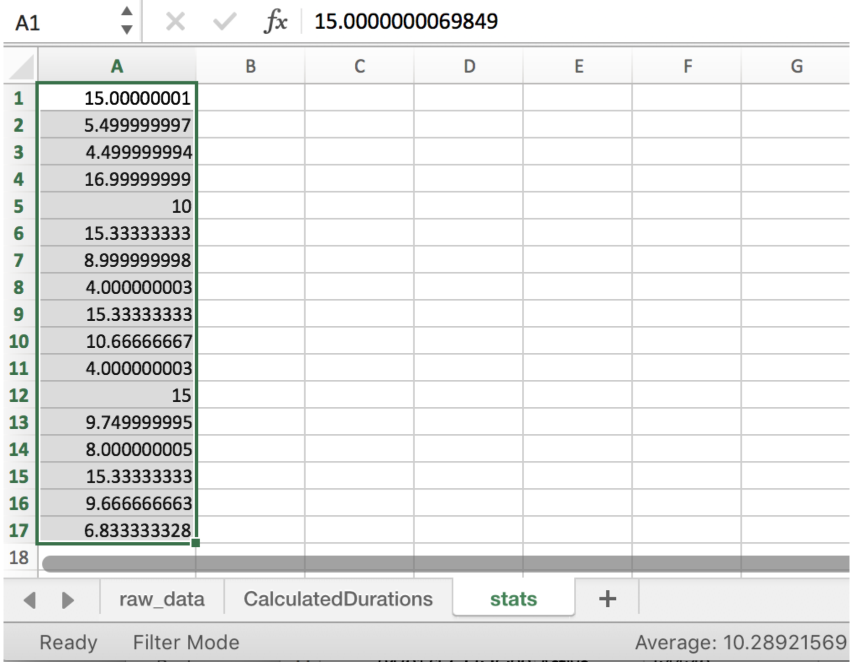
Search

☑ 9.749999995
☑ 10
☑ 10.66666667
☑ 15
☑ 15.00000001
☑ 15.33333333
☑ 16.99999999
□ NULL

16. Have a look at column D. Now all the values that are displayed are the active durations for each bathroom visit!

Step-by-step Instructions: Calculate descriptive statistics for the data set

1. First, create a new sheet named 'Stats', and copy-paste the filtered Active Duration values in column D, to the new sheet.

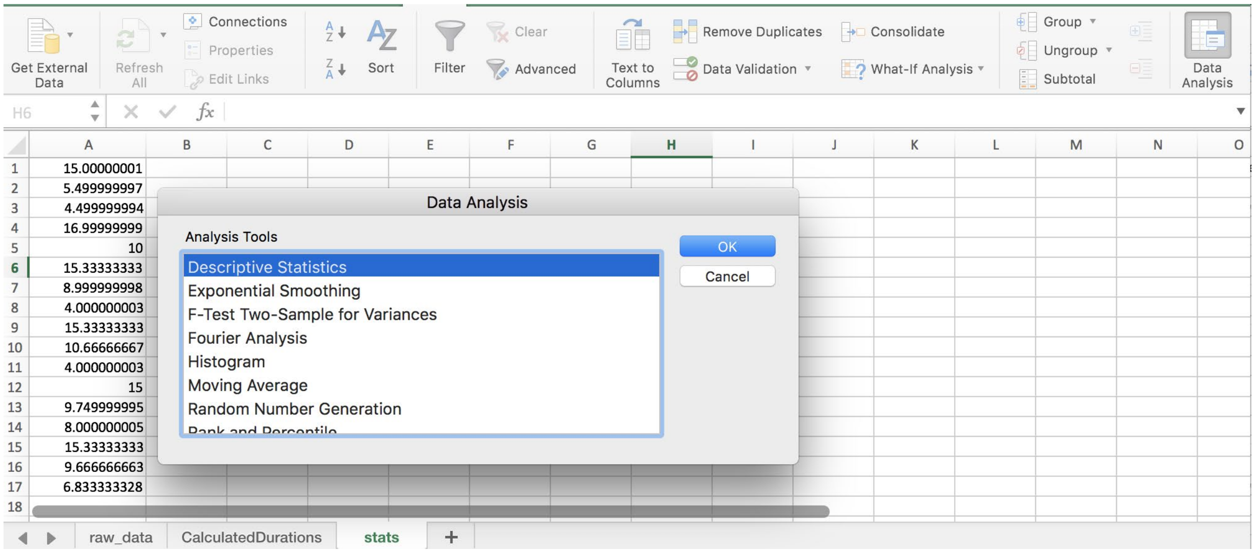


A1 *fx* 15.0000000069849

	A	B	C	D	E	F	G
1	15.00000001						
2	5.499999997						
3	4.499999994						
4	16.99999999						
5	10						
6	15.33333333						
7	8.999999998						
8	4.000000003						
9	15.33333333						
10	10.66666667						
11	4.000000003						
12	15						
13	9.749999995						
14	8.000000005						
15	15.33333333						
16	9.666666663						
17	6.833333328						
18							

Ready Filter Mode Average: 10.28921569

2. Next you need to turn on Excel's Data Analysis Add-On. Follow the instructions here: https://www.youtube.com/watch?v=mIoS7IRo36c
3. Now, click on 'Data Analysis'



	A
1	15.00000001
2	5.499999997
3	4.499999994
4	16.99999999
5	10
6	15.33333333
7	8.999999998
8	4.000000003
9	15.33333333
10	10.66666667
11	4.000000003
12	15
13	9.749999995
14	8.000000005
15	15.33333333
16	9.666666663
17	6.833333328
18	

Data Analysis

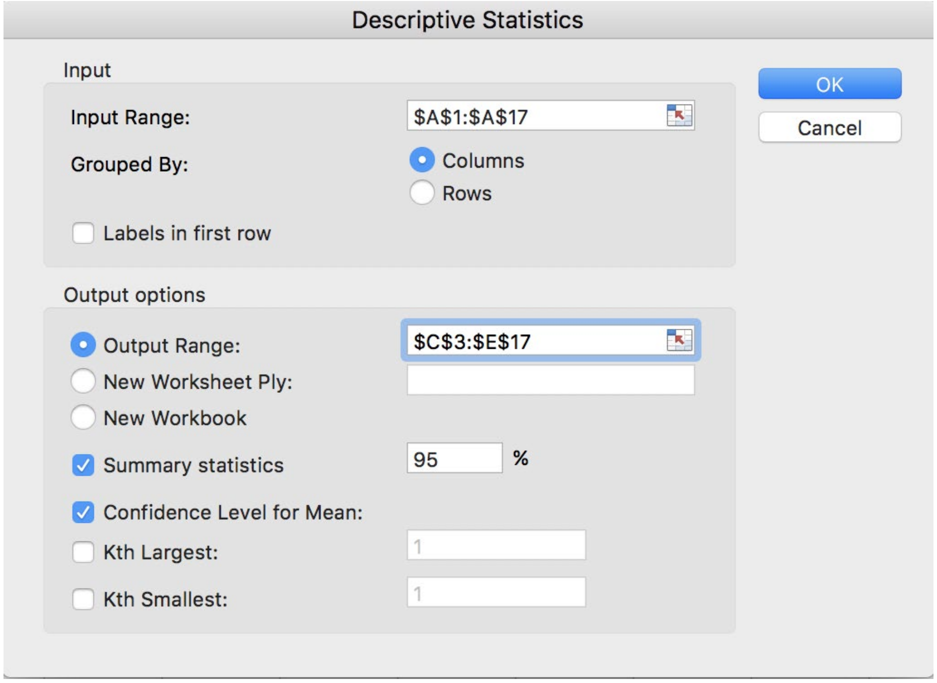
Analysis Tools

Descriptive Statistics
Exponential Smoothing
F-Test Two-Sample for Variances
Fourier Analysis
Histogram
Moving Average
Random Number Generation
Rank and Percentile

OK
Cancel

4. Select Descriptive Statistics and click OK.
5. Now, you will be shown the dialog box for Descriptive Statistics. Here, you need to select the following:
   a. Input range: this is the data range in column A
   b. Output range: this is cell range to add the descriptive statistics values
   c. Summary statistics

   Type in the following values:



Descriptive Statistics

Input

Input Range: $A$1:$A$17

Grouped By: ● Columns ○ Rows

□ Labels in first row

Output options

● Output Range: $C$3:$E$17

○ New Worksheet Ply:

○ New Workbook

☑ Summary statistics 95 %

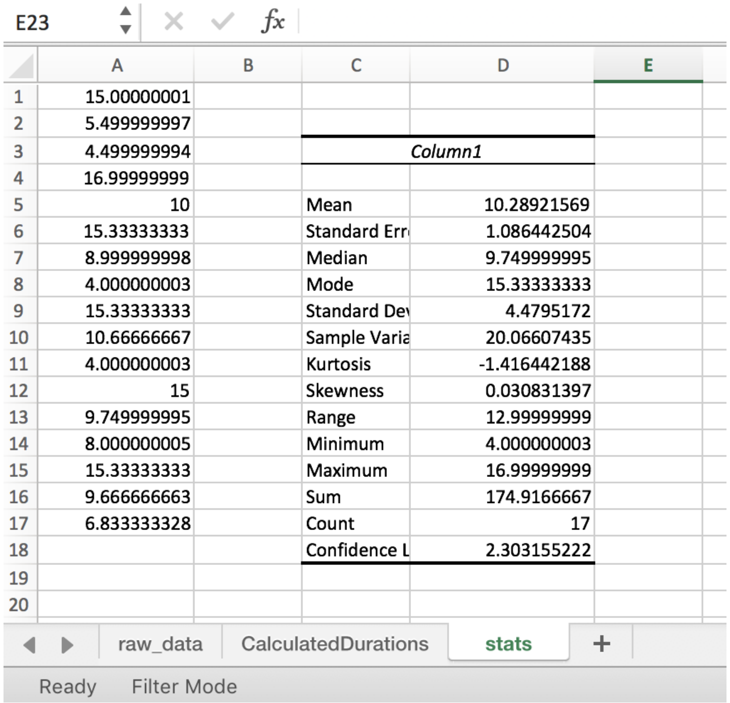
☑ Confidence Level for Mean:

□ Kth Largest: 1

□ Kth Smallest: 1

OK
Cancel

6. Click OK. Descriptive statistic values for the given sample should displayed as in the following screenshot:



	A	B	C	D	E
1	15.00000001				
2	5.499999997				
3	4.499999994		*Column1*		
4	16.99999999				
5	10		Mean	10.28921569	
6	15.33333333		Standard Err	1.086442504	
7	8.999999998		Median	9.749999995	
8	4.000000003		Mode	15.33333333	
9	15.33333333		Standard Dev	4.4795172	
10	10.66666667		Sample Varia	20.06607435	
11	4.000000003		Kurtosis	-1.416442188	
12	15		Skewness	0.030831397	
13	9.749999995		Range	12.99999999	
14	8.000000005		Minimum	4.000000003	
15	15.33333333		Maximum	16.99999999	
16	9.666666663		Sum	174.9166667	
17	6.833333328		Count	17	
18			Confidence L	2.303155222	
19					
20					

All done! Now we have some useful information about John's bathroom usage.

Task 2 - Submission Details

There are 2 questions in this task. Answer all of them in this word document itself.

Q1: Upload your completed excel workbook to unit site.

Q2: Look at the calculated descriptive statistics. Fill in the below table explaining what each term means in terms of John's bathroom usage. The first row has been done for you.

Mean	10.289 minutes	The average time John uses the bathroom per visit is 10.289 minutes.
Median	9.75 minutes	The median time John uses the bathroom is 9.75 minutes
Mode	15.33 minutes	The most common time John uses the bathroom for is 15.33 minutes
Standard deviation	4.48 minutes	The amount of time John uses the bathroom for deviates by 4.48 minutes

Minimum	4 minutes	The minimum time John uses the bathroom for is 4 minutes
Maximum	17 minutes	The maximum time John uses the bathroom for is 17 minutes
Count	17 times	John has used the bathroom 17 times
Sum	164.92 minutes	John has spent a total of 164.92 minutes in the bathroom

References

https://www.khanacademy.org/math/statistics-probability/displaying-describing-data#mean-median-basics
https://www.khanacademy.org/math/statistics-probability/displaying-describing-data/pop-variance-standard-deviation/v/range-variance-and-standard-deviation-as-measures-of-dispersion
https://www.lynda.com/Business-Skills-tutorials/Descriptive-statistics/550747/611825-4.html?org=deakin.edu.au

Task 3

Software Required

A web browser
One of the following apps installed:

- Android: Geo Tracker https://play.google.com/store/apps/details?id=com.ilyabogdanovich.geotracker&hl=en
- iOs: myTracks https://itunes.apple.com/au/app/mytracks-the-gps-logger/id358697908?mt=8
- Microsoft: GPS Tracker free https://www.microsoft.com/en-au/store/p/gps-tracker-free/9nblgggz2w34

You may install and try out any other app well as, as long as they can track GPS & export tracks to GPX format.

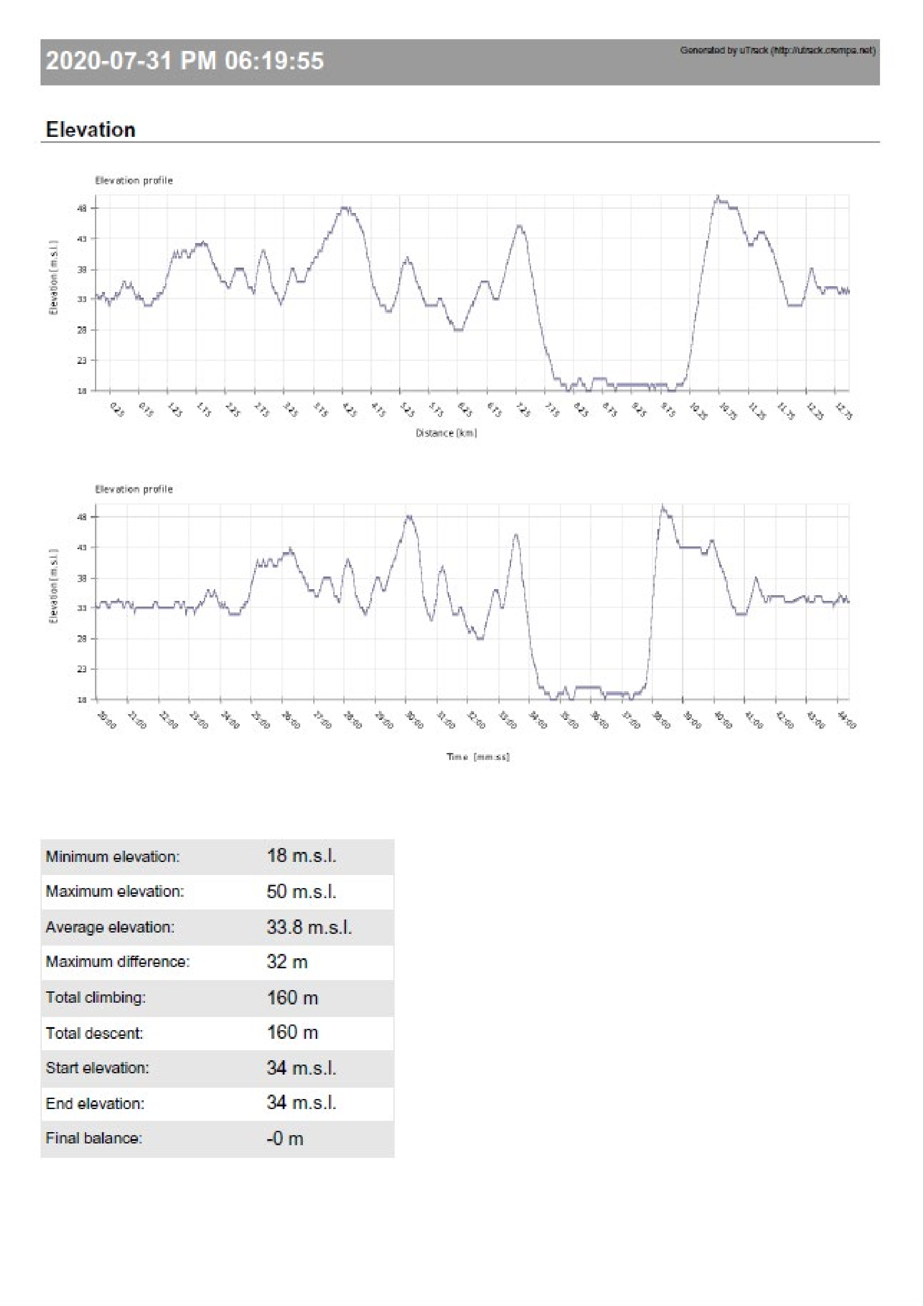
Task Submission Details

There are 3 questions in this task. Answer all of them in this word document itself and submit to unit site.

Q1: Track a journey using one of the installed mobile apps. It is best if your tracked journey spans at least 5 kilometers. Export your track to GPX format and save the file to your computer. Upload your .GPX file to unit site.

(You can email the .gpx file from your phone and download it to your computer)

Q2: Open a browser on your computer and go to http://utrack.crempa.net/ . Upload your .GPX file to the site and click



2020-07-31 PM 06:19:55

Generated by uTrack (http://utrack.crempa.net)

Elevation

Minimum elevation:	18 m.s.l.
Maximum elevation:	50 m.s.l.
Average elevation:	33.8 m.s.l.
Maximum difference:	32 m
Total climbing:	160 m
Total descent:	160 m
Start elevation:	34 m.s.l.
End elevation:	34 m.s.l.
Final balance:	-0 m

Q3: What information can you see from the generated report?

Note: depending on the app you used to record the GPS data, timestamps could be either in local time (that is AEST if you are in Melbourne) or sometimes it could be in UTC.

The information provided in the report is of the variations in elevation, speed, time and distance travelled while being tracked.