

# Project Deliverable K: **User and Product Manual**

GNG 1103 – Engineering Design

Faculty of Engineering – University of Ottawa

## **Group 1**

Marina Adoumou Foumane (300000206)

Daniel Alvarado (300059628)

Georges Barake (300176557)

Mohammad Faraz (300074523)

Hyoin (Joshua) Ahn (300175363)

April 11, 2021

# Table of Contents

---

Table of Contents	2
List of Figures.	3
Figure 1 - Final prototype design	3
List of Tables.	4
1 - Introduction	4
2 - Overview	5
2.1- Conventions	5
2.2 - Cautions and Warnings	5
3 - Getting started	5
3.1 - Set-up Considerations	6
3.2 - User Access Considerations	6
3.3 - Accessing The System	6
3.4 - Exiting The System	6
4 - Using The System	6
4.1 - <Given Function/Feature>	7
4.2 - <Given Function/Feature>	7
4.2.1 - <Given sub-function/sub-feature>	8
5 - Troubleshooting and Support	8
5.1 - Error Messages and Behaviour	8
5.2 - Special Considerations	9
5.3 - Maintenance	9
5.4 - Support	9
6 - Product Documentation	11
6.1 - <Subsystem 1 of prototype>	12
6.1.1 - Bill of Materials	12
6.1.2 - Equipment List	13
6.1.3 - Instructions	13
6.2 - Testing and Validation	15
7 - Conclusions and Recommendations for Future Work	16
8 - Bibliography	16

## List of Figures.

---

Figure 1 - Final prototype design

Figure 2 - The two different phases

Figure 3 - Speaker mounting

Figure 4 - attachment of LED lights (red)

Figure 5 - attachment of LED lights (blue,green)

Figure 6 - Main Code Used to Program the Circuit

Figure 7 - TinkerCad model of Circuit Wiring

Figure 8 - AutoCad Models of 3D printed parts

Figure 9 - First Part of the Code

Figure 10 - Setup to reference the libraries in the code

Figure 11 - Input and Output setup in the Code

## List of Tables.

---

Table 1 - All information of when and how to get an emergency assistance under troubleshooting situations

Table 2 - Bill of all materials purchased

Table 3 - List of all design deliverables with a link to their location (referenced documents)

## 1 - Introduction

This manual will provide you with information about our designed emergency system for the JAMZ drone in the case that the drone fails during flight. The module will display flashing lights and the speakers will generate a sound, and both of these together will be a warning for animals and pedestrians to know that there is a failed drone falling. This will ensure everyone gets clear of the drone and doesn't come near it for their own safety, and the safety of the drone. This manual will go over everything from materials needed to putting together the system and testing it. This manual will take you through the entire process that we went to when designing the system, but without all the challenges and setbacks, to make an easier design process for the user of this manual.

## 2 - Overview

### 2.1- Conventions

Designing a module for the JAMZ drone to provide a fail-safe case which will be triggered if the drone fails mid-flight.

This module will display different light patterns as well as speaker patterns/sounds for the 2 stages of the drone failure (level 1 - parachute engaged, level 2 - drone on the floor).



**Figure 1 - Final prototype design**

## **2.2 - Cautions and Warnings**

This product is generally a very safe product and design. There are no sharp objects/materials that are used during the design process, or during the use of the product itself. Please note to be safe when soldering the wires of the circuit and using necessary protective equipment.

## **3 - Getting started**

In the system there are two different phases. During phase 1, which resembles when the parachute is deployed, there is a constant beep noise with red flashing lights at the same time. Once the system is put into phase 2, which means that the sensors read that the drone is less than 5 metres above ground level, the speakers will immediately start playing a warning message, with flashing blue lights. In our prototype, there are buttons to mimic the two different phases, one for when the parachute deploys, one for when the sensors read that the drone is less than 5 metres above ground level.



**Figure 2 - The two different phases**

### **3.1 - Set-up Considerations**

The inputs on this system are the parachute and barometer. For the parachute it reads the high to low state of change or low to high state of change, the barometer reads the height of the drone. For output, we can include the amps, speakers, and lights. The amps and speakers play a warning sound, and for the light they flash.

### **3.2 - User Access Considerations**

To be able to access the system, you just need to know the basics of how an arduino board functions and some simple wiring techniques in case a wire is accidentally damaged.

### **3.3 - Accessing The System**

To test the system, you have to plug it into a laptop with the code software ready, and just upload it into the arduino board and run the system.

### **3.4 - Exiting The System**

To safely exit the system, all you have to do is push the reset button on the arduino board, and unplug the system, making sure no wires are touched.

## **4 - Using The System**

As said above, you will need to connect the arduino board to a power supplier. Our design was simple, you just need to use the 2 buttons to work the system. The first button starts the first stage, and the other one is for the 2nd stage. On the actual drone, there will be another code that will automatically turn on the lights, the speakers and the parachute.

#### 4.1 - <Given Function/Feature>

The first function is the speakers, located on each side of the drone.

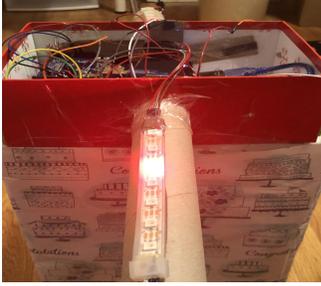


**Figure 3 - Speaker mounting**

It is connected with wires to the arduino board, making the speaker possible to function. During the first stage, the speaker makes a sound that alerts the surroundings that the drone is failing. The second stage plays a sound saying “Warning, Don’t approach”. It is to give a final warning before it reaches the ground

#### 4.2 - <Given Function/Feature>

The other function of the drone is the lights. Similar to the speakers, the wires are connected to the arduino board, but the actual wires of the lights are connected to the breadboard. The first stage makes the lights go red, warning the surroundings. The 2nd stage has blue and green lights.



**Figure 4 - attachment of LED lights (red)**



**Figure 5 - attachment of LED lights (blue,green)**

#### **4.2.1 - <Given sub-function/sub-feature>**

The lights aren't just flashing all the time, it keeps blinking. The 2nd one, has a blue on all the time, and green blinking.

## **5 - Troubleshooting and Support**

### **A. Error conditions that may be generated**

- the lights do not turn on
- The intensity of the light which is not as strong as usual
- No sound produced by the speakers
- the intensity of the sound which is not as loud as usual

### **B. Recovery procedures including**

- Debug the code to see if there are any features to replace
- We can go to the Arduino and Breadboard to check that the connections are made correctly or that there are no disconnections of the wires
- We can do an inspection of the assembly on the drone to see if the lights or speakers are not detached or slightly disconnected

### **C. Error correction procedures**

- Repair any features that would cause the code to malfunction or even search for more efficient news if necessary.
- Reconnect all incompatible wires from the Arduino
- Redo the assembly and correctly fix all the parts on the drone assembly

#### D. Corrective actions that may need to be taken

- You need to replace the equipment by buying a new one that works well
- Or replace lights and speakers with larger capacity ones
- Set up a system to activate more adequate shares from the code
- Make sure that the programming will work well in case of emergence before the departure of the Drone

### 5.1 - Error Messages and Behaviour

All the error messages or behaviors that a user may receive, the cause(s) and/or possible corrective actions for the error. Have been mentioned in the previous section (section %).

As for the list of parts that are prone to breaking in this section we can mention: The LED lights and the speakers, the jumpers wires, and the amp.

### 5.2 - Special Considerations

**(particular circumstances and identification of the problem)** The particular circumstances in which a repair action must be taken into account would be for example if it is found that a Drole had an accident during the trip without however having generated the preventive functions such as the lights and the speakers.

**(actions)** It will therefore be necessary to establish a theory and set up an analysis system in order to determine the probable cause of equipment malfunctions. This can be done by simple tests to check if everything was connected well or if there was a disconnection between the code and the physical elements.

**(warnings)** After having found the origin of the problem and in our most probable case is a bad connection of the parts, we can restore the assembly of the various elements of the system.

**(exceptions)** If necessary, the stability of the attachments can be enhanced by using bonding methods with extremely strong adhesive tapes or even a glue stick.

**(Verification, Solution Testing, and Results)** The troubleshooting would end with testing the new system to observe and note improvements to the system. This test can be a short trip not very far where the distance will be sufficient to record all the observations.

### 5.3 - Maintenance

The regular maintenance that must be performed on the prototype to avoid failure are:

- The device and in particular the necessary cameras should be wiped down with a suitable soft cloth and a non-abrasive cleaner.
- Clean the engine chamber
- Make sure the antennas are free of debris to ensure a good connection with the base.
- Make sure the software is up to date and running the latest version.
- Replace motors after prolonged flight and use.
- Verify that all motors respond by trying a drone start-up sequence.
- Pass a simple flight test where you don't have to fly too high and maneuver at low speed. If the drone responds correctly, there is less risk of failure.

### 5.4 - Support

**Table 1 - All information of when and how to get an emergency assistance under troubleshooting situations**

	<p><b>Lights failures / Speakers failures / Crash / unexpected sound</b></p>
<p>Information on how the user can get emergency assistance and system support</p>	<ul style="list-style-type: none"> <li>• Intervention of the security service of the company Jamz</li> <li>• Police intervention</li> <li>• Interventions of computer scientists specializing in aeronautical information technologies</li> <li>• The population can report the incident to the Jamz’s company by calling them</li> </ul>
<p>Names of the responsible person and email addresses of the staff who serve as points of contact for system support</p>	<p><b>Jamz office</b></p> <p>1- Security office: (613)-900-0357  <a href="mailto:Jamzsecurity@hotmail.ca">Jamzsecurity@hotmail.ca</a></p> <p>2- Technology department: (613)-000-5703  <a href="mailto:Jamztechnology@hotmail.ca">Jamztechnology@hotmail.ca</a></p> <p>3-secretariat (in case one can join the 2 above specialized services directly: (613)- 099-5307  <a href="mailto:Jamzsecretariat@hotmail.ca">Jamzsecretariat@hotmail.ca</a></p> <p><b>Police office:</b>          Police Ottawa : 911</p>

<p>Instructions for how identified problems with the system are to be reported</p>	<p>By phone call or by email:</p> <ul style="list-style-type: none"> <li>- Greetings, presentations, explains the situation (could be a crash, a noisy sound...), Give coordinates (Area) and gives facts (describe if completely broken, if the sound doesn't stop and gives further details)</li> </ul> <p>NB: If necessary or asked, pictures must be taken and sent</p>
<p>Instructions for security incident handling, as appropriate.</p>	<p><b>SAFETY FIRST / ASSESS DAMAGE / REPORT TO FAA / REFLECT</b></p> <ul style="list-style-type: none"> <li>● Install drone drop warning signs to warn residents</li> <li>● Block the road during the cleaning operation if necessary</li> <li>● Stop the drone and remove the battery and accessories</li> <li>● Clean all dirt / sand / debris from the drone using alcohol swabs</li> <li>● Check the drone for structural damage (battery, camera, etc.)</li> <li>● Inspect the area around the territory to make sure that no physical parts that could injure someone have escaped</li> <li>● If this is a major incident, report the situation to the appropriate agencies such as the police and aviation</li> </ul>

## 6 - Product Documentation

This prototype was built using 3 main components, software, electrical, and mechanical.

The software component was the coding of the circuit. This code was the means for the functionality of the speakers and lights. Without the code, our prototype would not be working.

```

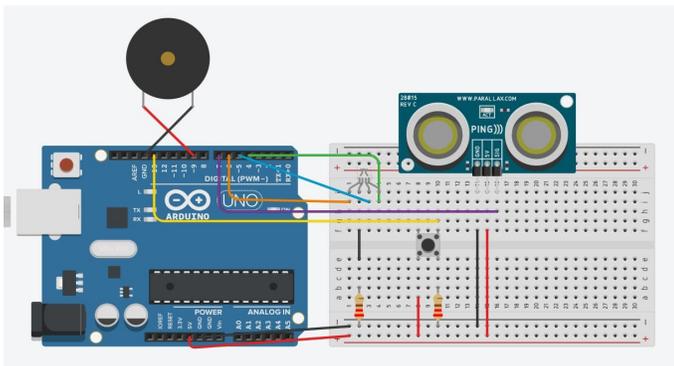
void loop()
{
  int Stage1,i;
  Stage1=digitalRead(13);
  while (Stage1==HIGH)
  {
    int Stage2;
    voice.say(sp4_TONE1);
    tone(9,520,5000);
    for( i=0;i<7;i++){
pixels.setPixelColor(i, pixels.Color(255, 0, 0));
pixels.show();
delay(500);
pixels.setPixelColor(i, pixels.Color(0, 0, 0));
pixels.show();
}

    Stage2=digitalRead(12);
    while (Stage2==HIGH)
    {
      voice.say(sp4_TONE2);
      voice.say(sp4_WARNING);
      voice.say(spa_DONT);
      voice.say(sp5_APPROACH);
      for( i=0;i<7;i++){
        int x;
pixels.setPixelColor(i, pixels.Color(0, 0, 255));
x=i+1;
pixels.setPixelColor(x, pixels.Color(0, 255, 0));
pixels.show();
delay(500);
}
    }
  }
}

```

**Figure 5 - Main Code Used to Program the Circuit**

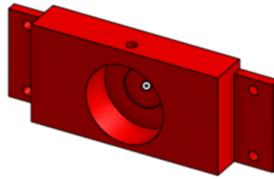
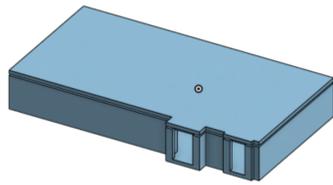
The electrical component was the circuit wiring. This included wiring the speakers and lights to the arduino board, and then using the code to program them to work .



**Figure 6 - TinkerCad model of Circuit Wiring**

The mechanical component was the 3D printed parts that were needed to put together our prototype. The material used in the system was PLA plastic, this was used for the 3D printing part because the plastic is a very firm and well built material, to ensure the highest level of quality was used. We could have used material made out of metal but, we figured that the PLA

plastic was the most cost efficient.



**Figure 7 - AutoCad Models of 3D printed parts**

Please refer to **Figure 1** and **Figure 2** for pictures of the final prototype.

## 6.1 - <Subsystem 1 of prototype>

### 6.1.1 - Bill of Materials

**Table 2 - Bill of all materials purchased**

Material Names	Sources	Quantity	Unit price (\$)
Led strips	<a href="https://www.amazon.ca/gp/product/B08CXCJ3N6/ref=ox_sc_act_title_2?smid=AUBKFGW0HQDN6&amp;psc=1">https://www.amazon.ca/gp/product/B08CXCJ3N6/ref=ox_sc_act_title_2?smid=AUBKFGW0HQDN6&amp;psc=1</a> Or use the one available at makerlab	1	12.99
speakers	<a href="https://www.amazon.ca/gp/product/B01LYHFS1P/ref=ox_sc_act_title_3?smid=A34K5WF5Z9R33P&amp;psc=1">https://www.amazon.ca/gp/product/B01LYHFS1P/ref=ox_sc_act_title_3?smid=A34K5WF5Z9R33P&amp;psc=1</a>	2,	10.98 ( the price is for both)
amp	<a href="https://www.amazon.ca/gp/product/B07PW4FGS3/ref=ox_sc_act_title_4?smid=A37V0VEET6R0X3&amp;psc=1">https://www.amazon.ca/gp/product/B07PW4FGS3/ref=ox_sc_act_title_4?smid=A37V0VEET6R0X3&amp;psc=1</a>	1	9.45\$
Jumper wires		27	Free
220 ohms resistors		2	Free
5v power supply	The led strips that were buying come with a 5v supply but if we can get them from	1	Free

	<p>makerlab then  <a href="https://www.amazon.ca/RBSC-H-5V-2A-Power-Adapter/dp/B01M7WQDGU/ref=sr_1_28?dchild=1&amp;keywords=5v+power+supply&amp;qid=1615492224&amp;s=electronics&amp;sr=1-28">https://www.amazon.ca/RBSC-H-5V-2A-Power-Adapter/dp/B01M7WQDGU/ref=sr_1_28?dchild=1&amp;keywords=5v+power+supply&amp;qid=1615492224&amp;s=electronics&amp;sr=1-28</a></p>		
--	---	--	--

### 6.1.2 - Equipment List

- 3D printed arduino uno case
- 3D printed speaker mounts
- 3D printed LED light clamps
- 3M tape
- M3 screws
- Hot glue gun

### 6.1.3 - Instructions

Step 1: Writing the code

```
#include <Adafruit_NeoPixel.h>
#include "Talkie.h"
#include "Vocab_US_Large.h"
#include "Vocab_US_Clock.h"
#include "Vocab_US_Acorn.h"
```

**Figure 8 - First Part of the Code**

This is the first part of the code. The “#include” is a function to let the system know where outside libraries(they include things like special functions for specific tasks). To use these liberties they must also be added to the arduino IDE software(this is the program we use to write code) use <https://www.arduino.cc/en/guide/libraries> to know more about that process.

The specific libraries used are Adafruit\_NeoPixel and Talkie. The first is to control our individually addressable lights(note: the lights must be individually addressable and have 3 contacts. In other words the lights must be WS2812B lights. The adafruit liberties allows us to send instructions that the lights can read and perform part of it. What it can do is assign specific values for what color to each light should be independent of the color of the other lights around it. For more info on it visit [https://github.com/adafruit/Adafruit\\_NeoPixel](https://github.com/adafruit/Adafruit_NeoPixel). The second library Talkie allows the use of the other liberties below(Vocab\_US\_Large,Vocab\_US\_Clock,Vocab\_US\_Acorn) combined these libraries come

with 1000 pre-made words that can be used for more info visit <https://www.arduino.cc/reference/en/libraries/talkie/>. Also there are ways to add your own words and even sentences the instructions for that can be found here <https://github.com/ArminJo/Talkie/blob/master/README.md>.

```
#define NUM_LEDS 7
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUM_LEDS, 6, NEO_GRB + NEO_KHZ800);
Talkie voice;
```

### Figure 9 - Setup to reference the libraries in the code

This is a setup to be able to reference the libraries later in the code. First the second line here is the setup for the adafruit library. “Pixels” is the name used in our code to reference the lights. Anyother name can be used as long as it follows the C/C+code rules. The rest of the line refers to informations like the number of LEDs in your strips(NOTE: if your strips are longer the preset value the unaccounted leds will not be turned on), the second value refers to the pin you wish to use to send the data and lastly the last part is to tell the program what lights your using(NOTE: for the light used here just copy what is in the example if you wanted to use lights strips that would be configured differently this could mean changing that last part again for more information on how to do that visit adafruit's information site. The last line on there is to reference the Talkie library again whatever name you decide to use should work.

```
void setup()
{
  pixels.begin();
  pinMode(13, INPUT);
  pinMode(12, INPUT);
}
```

### Figure 10 - Input and Output setup in the Code

This part of the code is to set up inputs and outputs that will be used later in the project. The first part to let the program know you want to start writing to the lights. The two other parts are so set which pins are going to be our inputs(NOTE: one input is for the parachute signal, where we are trying to read a low to high or a high to low stage change. The second input to read the signal from the barometer.

Please refer to **Figure 5** for the main part of the code.

This is the last part and the main part of the program, here we are going to set instructions for when and how our inputs and outputs interact. In this code we are using push buttons as our inputs, while this can work fine to replace the input from the parachute the code for the barometer will differ quite a bit. To know more about how to code for the barometer you must look into how the signal is being sent by the barometer and figure out max and min values to

know what value input corresponds to what height. The main takeaway from this part of the code is the while loop in the while loop setup all the other parts are just instructions sent to the outputs on how they should react. Firstly the exterior while loop is face 1 or in other terms when the drone is known to be falling. Face 2 is the inner loop, when we have established the drone is on the ground. So for face 1 we use the parachute input, once in the first loop it will go through the instructions for face one and at the end it will check whether it should go to face 2 or not. If it does not move to face 2 it will repeat the set of instructions and check again until the system is reset or the desired input is received to move to face 2. Once in face 2 the code will repeat the instructions determined in the inner loop until the system is reset.

**NOTE:** the number of faces can be increased using this loop in a loop system. Also the conditions to enter a loop should be made by first copying the input then comparing the copied value to see if it matches our requirements to move into the while loop. This will ensure that when the loop restarts and it checks the requirements again it will always read true. This is due to the fact that while the true input might fluctuate the copied input will remain the same unless we instruct the code to recopy the true input. So as long as while in the loop we don't change the copied value the value won't change. Lastly the instructions for the lights and speakers can be customized to whatever the user can code. As mentioned before you don't even need to use the premade works from the libraries if you can follow the instructions on how to make your own. Also the lights are highly customisable from the exact color you wish to display as well when each light should light up.

## 6.2 - Testing and Validation

Test 1 - Testing the speakers

- Making sure the code worked when connected to the arduino
- Connecting the speakers to the circuit
- Having the speakers work in such a way that they could show the 2 different stages of the drone failure
- Stage 1 - parachute engaged, speaker would output a constant pattern
- Stage 2 - drone on the ground, speaker would output a different pattern along with the phrase "Warning, don't approach"

Test 2 - testing the lights

- Making sure the code worked when connected to the arduino
- Connecting the LED lights to the circuit
- Having the lights work in such a way that they could show the 2 different stages of the drone failure

- Stage 1 - parachute engaged, a flashing red light on the LED strip
- Stage 2 - drone on the ground, flashing blue light on the LED strip

Since we did not have access to the JAMZ drone itself while building our prototype, we had to use push buttons to engage the 2 stages of the drone failure, and engage the stages manually. If we had the JAMZ drone, we would connect our circuit to a barometer and adjust the wiring to be able to engage the stages of the speakers and lights automatically.

## 7 - Conclusions and Recommendations for Future Work

In conclusion, this project was a very difficult one because of major time restrictions, yet was fun because of how challenging it was. We learnt how to properly time manage as a group and always stay on top of everything despite everything else going on like other classes and work schedules. Despite a lot of issues because of different situations our group successfully came together and immediately fixed any issue, just like we were taught in class. For example, having a new member join our group, we used all the lessons and strategies taught in class and implied it in the real world, and without a doubt we found a solution. Something we would have done differently if we would've had more time would be a more in depth light pattern that can also be much brighter and noticeable. Another major reason that prevented us from doing that was budget restrictions, because a brighter light would have cost significantly more than the budget given to us. Something else that we abandoned due to lack of time was a more detailed and realistic drone configuration for our prototype, that would have made the entire prototype look more professional and realistic to the actual JAMZ drone.

## 8 - Bibliography

“5W Audio Amplifier Module, DROK 5W+5W Digital Amplifier Board 2.0 Dual-Channel Immersion Gold Stereo Amp Mini Class D DC5V Amplify DIY Circuit Board for Bookshelf Speakers Floor Speakers with Knob.” *Amazon.ca: Electronics*, [www.amazon.ca/Amplifier-DROK-Dual-channel-Immersion-Bookshelf/dp/B077MKQJW2/ref=pd\\_bxgy\\_2/136-7093863-8818208?\\_encoding=UTF8&pd\\_rd\\_i=B077MKQJW2&pd\\_rd\\_r=a6ea4dd4-67a9-404e-bd03-5f75f3abf7af&pd\\_rd\\_w=X92Q3&pd\\_rd\\_wg=h6Ywt&pf\\_rd\\_p=42339929-297e-4141-b7b2-fe55db70f4b7&pf\\_rd\\_r=R5R7DRP0TET3VQ2BT4B1&psc=1&refRID=R5R7DRP0TET3VQ2BT4B1](http://www.amazon.ca/Amplifier-DROK-Dual-channel-Immersion-Bookshelf/dp/B077MKQJW2/ref=pd_bxgy_2/136-7093863-8818208?_encoding=UTF8&pd_rd_i=B077MKQJW2&pd_rd_r=a6ea4dd4-67a9-404e-bd03-5f75f3abf7af&pd_rd_w=X92Q3&pd_rd_wg=h6Ywt&pf_rd_p=42339929-297e-4141-b7b2-fe55db70f4b7&pf_rd_r=R5R7DRP0TET3VQ2BT4B1&psc=1&refRID=R5R7DRP0TET3VQ2BT4B1).

“Gikfun 2 Inch 4Ohm 3W Full Range Audio Speaker Stereo Woofer Loudspeaker for Arduino (Pack of 2pcs) EK1725.” *Amazon.ca: Electronics*,

[www.amazon.ca/Gikfun-Speaker-Stereo-Loudspeaker-Arduino/dp/B01N74TGFM/ref=sr\\_1\\_6?dchild=1&keywords=speaker%2Bdriver&qid=1615134057&sr=8-6](http://www.amazon.ca/Gikfun-Speaker-Stereo-Loudspeaker-Arduino/dp/B01N74TGFM/ref=sr_1_6?dchild=1&keywords=speaker%2Bdriver&qid=1615134057&sr=8-6).

“Led Strip Lights 32.8ft TECKIN Remote Control Waterproof RGB Led Light Strips for Bedroom Color Changing Lights 5050 RGB 300 LEDs Light Strips Kit for Home, Bedroom, Kitchen,DIY Decoration.” *Amazon.ca: Tools & Home Improvement*, [www.amazon.ca/TECKIN-Changing-Lighting-Flexible-Decoration/dp/B07YSM82NW/ref=sr\\_1\\_9?dchild=1&keywords=rgb%2Bled%2Bstrips&qid=1614819778&s=hi&sr=1-9&th=1](http://www.amazon.ca/TECKIN-Changing-Lighting-Flexible-Decoration/dp/B07YSM82NW/ref=sr_1_9?dchild=1&keywords=rgb%2Bled%2Bstrips&qid=1614819778&s=hi&sr=1-9&th=1).

## 9 - Appendix I: Design Files

**Table 3 - List of all design deliverables with a link to their location (referenced documents)**

Document Name	Document Location and/or URL	Issuance Date
Project Deliverable B	<a href="https://docs.google.com/document/d/1uiXptzeRMrMZYqU54E2Fc1PExgWltOb2qB8-ch_NhjQ/edit">https://docs.google.com/document/d/1uiXptzeRMrMZYqU54E2Fc1PExgWltOb2qB8-ch_NhjQ/edit</a>	February 13, 2021
Bill of Materials table	<a href="https://docs.google.com/document/d/1i5BUD3JHGcxO8aSckEMCicVWdwfinphvIWJr8HA8MnM/edit">https://docs.google.com/document/d/1i5BUD3JHGcxO8aSckEMCicVWdwfinphvIWJr8HA8MnM/edit</a>	February 28, 2021
Project Deliverable E	<a href="https://docs.google.com/document/d/1kR-m8DmCwJDq1Dto-YMkIW6-XiSHOBnrNU01j8xC3Xc/edit">https://docs.google.com/document/d/1kR-m8DmCwJDq1Dto-YMkIW6-XiSHOBnrNU01j8xC3Xc/edit</a>	February 28, 2021
Project Deliverable H	<a href="https://docs.google.com/document/d/1x2OLo8bUSZWobzh85ODi9n0oUgsAC-f-cxu9O89vmp0/edit">https://docs.google.com/document/d/1x2OLo8bUSZWobzh85ODi9n0oUgsAC-f-cxu9O89vmp0/edit</a>	March 28, 2021

