Mirrorage

DIY Guide

Build your own Mirrorage

Table of Contents

Build your own Mirrorage 1

Introduction 2

How to Follow the Guide 3

The Mirror 4

The Printed Circuit Boards for Audio & Power 5

- Disclaimer 5
- Step 1: Creating the Schematic Document 5
- Step 2: Creating The Printed Circuit Board 6
- Step 3: Fabrication Outputs 8
- Step 4: Switching Buck Regulator Creation 12
- Step 5: Soldering the PCBs 13

The Software 15

- Step 1: Updating Your Raspberry Pi 15
- Step 2: Wifi Configuration 15
- Step 3: Static IP Setup 17
- Step 4: Hosting The Webpage 19
- Step 5: Setting up Pi to Kiosk mode: 21

Enclosure 24

- Step 1: Safety First 24
- Step 2: The Right wood for the job 24
- Step 3: Rough stock 25
- Step 4: Rabbits? 25
- Step 5: Making the external frame 25
- Step 6: Let there be light 26
- Step 7: Moving inside 26
- Step 8: More Rabbits! 26
- Step 9: The backbone of the project 27
- Step 10: Support 27
- Step 11: Connecting it all together 27
- Step 12: Show it off! 28

Putting it All Together 29

Introduction

Congratulations on obtaining an issue of a DIY guide for your very own Mirrorage Smart Mirror. By the end of this build guide you are going to have a complete smart mirror which can serve you your daily weather, time and date, Bluetooth audio and finally LED lighting for day and night. In this DIY guide we provide you with links to the code used along with general prices that we expect each component to run you.

The following guide features the steps taken by Team Mirrorage of Camosun College to completely design a build a smart mirror. It is important to note that some experience in electronics and woodworking is strongly recommended. As the steps following were catered around a product built for a client, there is flexibility to make Mirrorage your own.

How to Follow the Guide

This "do-it-yourself" guide features 6 sections:

- The Mirror
- Printed Circuit Boards
- The Software
- The Enclosure
- Putting it Together

In each section you will find detailed steps to follow to reach your end goal and the parts required to do so. Within the appendices you will find the source Team Mirrorage used in obtain the parts, as certain parts were donated you will need to do some research of your own in finding them.

In sections The Software, The Audio and The Power you will note that certain files can be created yourself using the steps as a guideline but the files Team Mirrorage has created are also available. Directions to achieve the files will be noted in each section necessary.

The Mirror

If you choose to follow this build guide to the same parameters as done by Team Mirrorage you will need to obtain a piece of tempered glass that measures 22” x 30” and 3/16” thick. Of course, any other size can be obtained if desired you will just have to adjust as you move through the guide to fit your glass. See Appendix E for the glass company used by Team Mirrorage in their build.

With the glass obtained you will need to apply films to both sides of the glass. This is best done by professionals to ensure the films required is properly applied to the glass. On one side of the glass you will need a 2” border of a misted film for light diffusion along with the interior being coated with a 60-70% reflective film that will allow light to pass through from one side, but appear as a mirror when no light is present. In Figure 1 you will see a graphical representation of the layout used with numerical values measured in inches. See Appendix E for the film company used by Team Mirrorage.

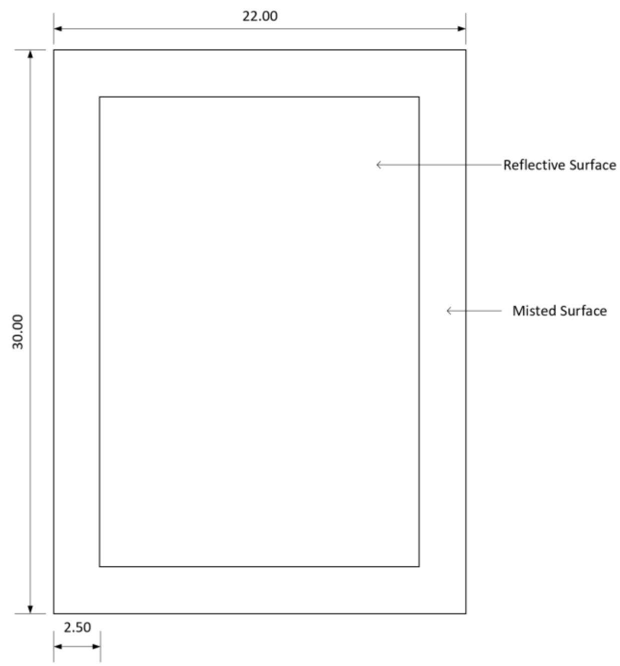


Figure 1 The films

On the other side of the mirror you can choose to apply a layer of anti-fog if your glass film supplier has it available. Keep in mind, the anti-fog film used by Team Mirrorage proved to be very sensitive to scratches and is only necessary if you deem it to be.

Lastly, for optimal reflection, on the inside of the mirror an additional adhesive should be applied around the placement of your monitor to ensure the only light passing through will be the monitor. But we won’t to this until the last section of this guide.

The Printed Circuit Boards for Audio & Power

Disclaimer

For this project, we will be using *Altium Designer 16* to design the PCBs for the project. If you do not want to make your own board we have provided files on *Github* that can be sent to a manufacturer to have the board made for you. I repeat these steps are only if you want to make the board from scratch. With that out of the way let us proceed in showing you how to make your own version of the original board.

This next part of the guide will show you how to design the two PCBs needed for this project. The libraries for both designs are located on *Github*. The reason for making two PCBs instead of just one is because the buck regulator needed in this design is a switching regulator. This switching produces harmonic frequencies that could interfere with the Bluetooth portion of this design. This way we will make sure that our boards aren't interfering with one another. We will start by making the breakout board for the amplifier, and *Bluetooth* module. When designing the switching buck regulator, the steps will be the same but schematics documents and libraries will be different.

Step 1: Creating the Schematic Document

Start by opening *Altium Designer* up. Next you are to create a **New PCB Project**. Select **File → New → Project**. Make sure you select a PCB project.

Your next process is to add in the correct libraries. I have already gone through the time to create custom footprints and schematic components for everything. You can download them directly and add them into the project.

To add the libraries used in this design, **Design → Add/Remove Libraries → Install**, then search for the library needed and click **OK**. Select the **Library** tab on the right side of *Altium* and Place all components from the newly added library onto the schematic and route the correct pins together like shown in Figure 2.

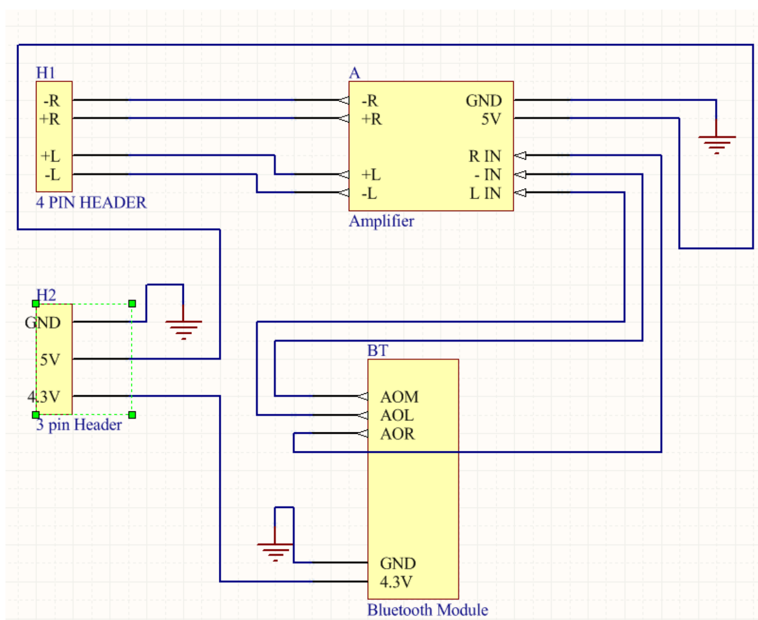


Figure 2 The Power Schematic

We now want to compile the schematic document click on **Design → Compile** (should be at the top of the list). If any errors are present in the design you will be notified, but if not then nothing will pop up and your design will be error free. Now save the newly compiled project if no errors are present. If there are errors fix them and recompile then save.

We now want to create a schematic library for this project. **Go to Design → Make Schematic Library**. You can now move onto creating the PCB.

Step 2: Creating The Printed Circuit Board

Create a new PCB, **File → New → PCB**. To Transfer the components into the PCB select **Design → Update PCB Document *"name of PCB project".***

Before you start placing components around the PCB make sure the Bluetooth module is on the bottom layer. Right click on the Bluetooth component and select **Properties**. The only thing that needs to be changed is circled blue in Figure 3.

Now orientate the components around like shown in Figure 4 Then Interactively trace each pad together ignoring the ground pads.

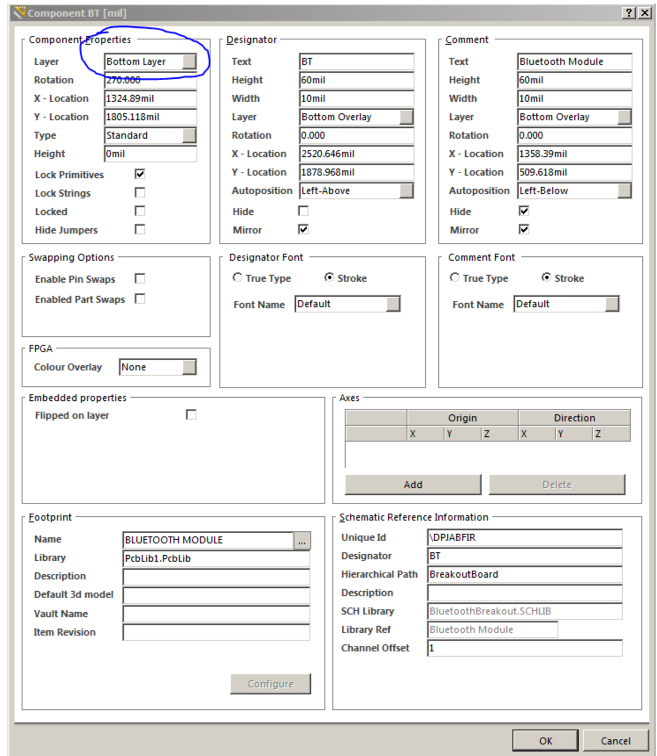


Figure 3 Starting the PCB

NOTE: You will have lines telling you that the grounds need to be routed together. Ignore it because a ground plane will take care of that. Ground planes are used to reduce noise on the ground trace and eliminate inaccurate references at different points along the ground. After you are done routing the PCB and the components are laid out correctly. It is now time to put a border around the edge so we can define the keep out layer so the milling machine knows where the edges of your PCB is.

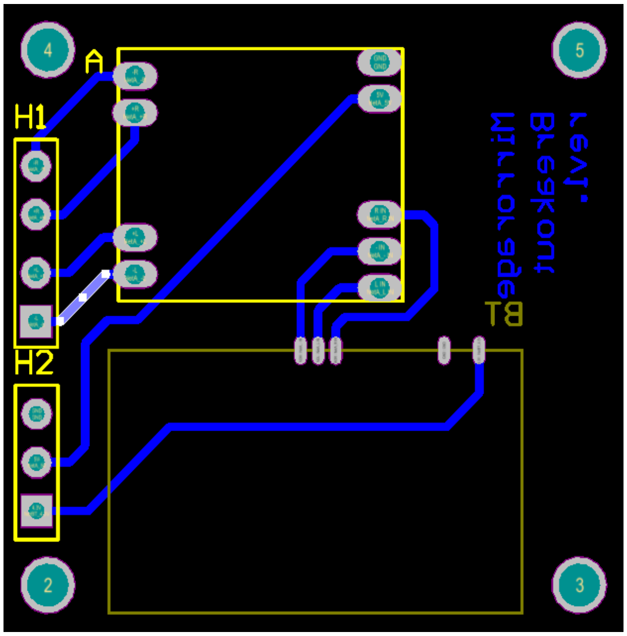


Figure 4 Bluetooth board layout

Click on the routing tool and draw a square border around the perimeter of your components
Select all of the 4 sides of the Perimeter using the shift key and then select **Design → Board Shape → Define from selected objects**

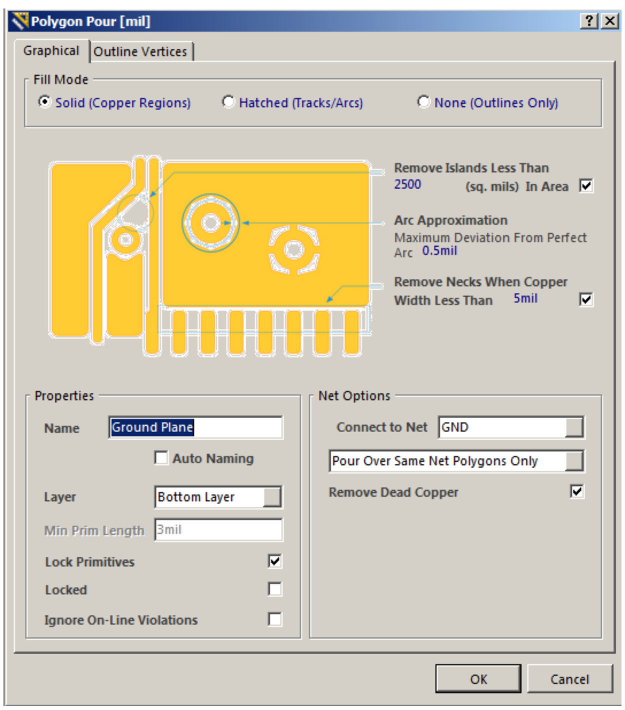


Figure 5 Polygon pour

After you define the board shape you can now do a polygon pour. This will create a ground plane like done for the previous PCB.

Place → Polygon Pour and set it up like Figure 5
Once connecting all four corners of the PCB press the ESC key to pour the ground plane.

The ground plane should look the same as Figure 6. If you ever need to fix something you can shelve the ground plane by right clicking the **Ground plane → Polygon Actions → Shelve All**. After the problem has been fixed to re-pour the polygon you go **Tools → Polygon Pour → Restore Shelved Polygon(s)**.

It is now time to create your PCB library. To do this, **Tools → Make PCB Library**. After that is done you should save your new library and your entire project.

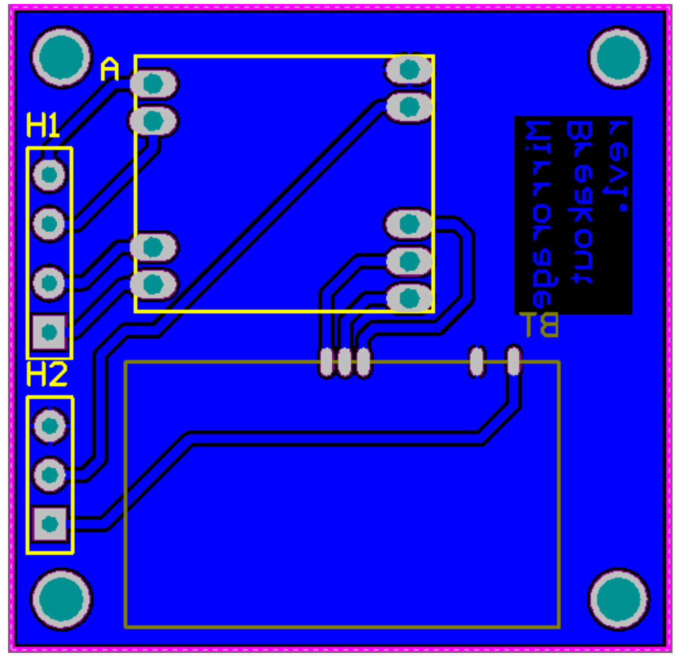


Figure 6 Ground plane

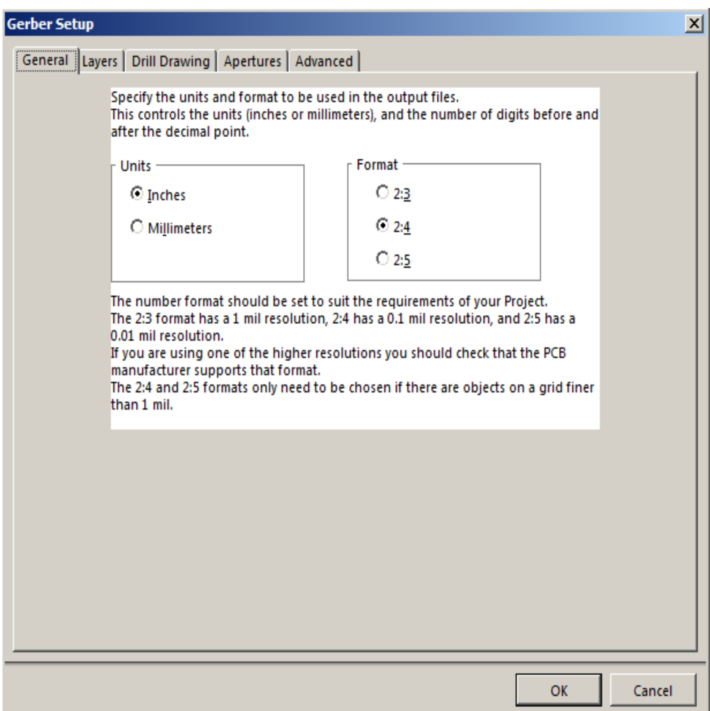


Figure 7 Gerber setup

Step 3: Fabrication Outputs

The next step in this process is to create your Gerber Files and NC Drill Files. This enables the company that fabricates the PCBs to have design specifications.

We'll create the Gerber File first, **File → Fabrication Outputs → Gerber Files**.

Set up the first tab of "Gerber Setup" like shown in Figure 7.

Setup the second tab of "**Gerber Setup**" like shown in Figure 8.

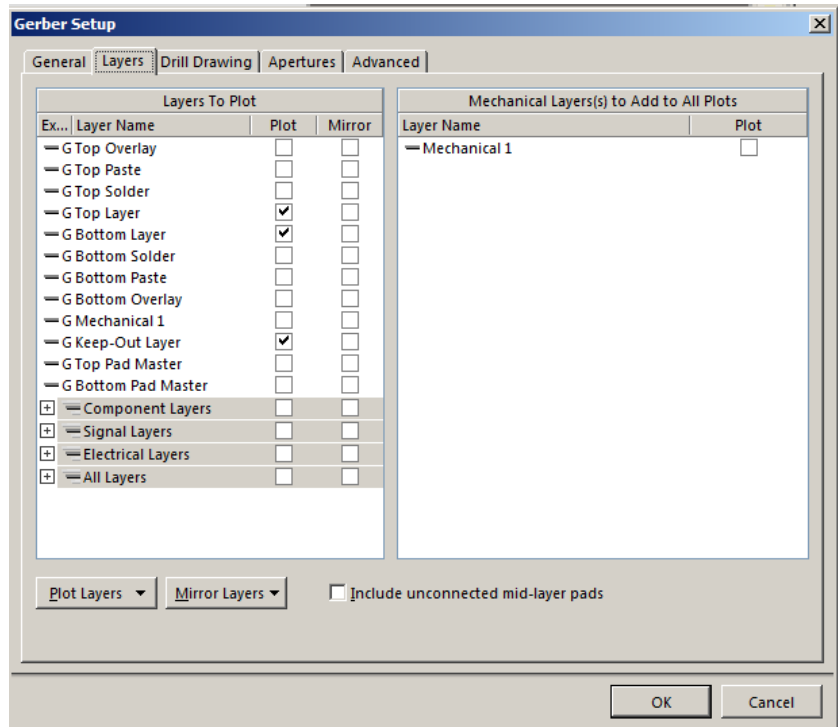


Figure 8 Gerber setup

When you click on "**Drill Drawing**" tab be sure to click on "**Configure Drill Symbols**" Change the symbol size to 50mil. This is shown in Figure 9.

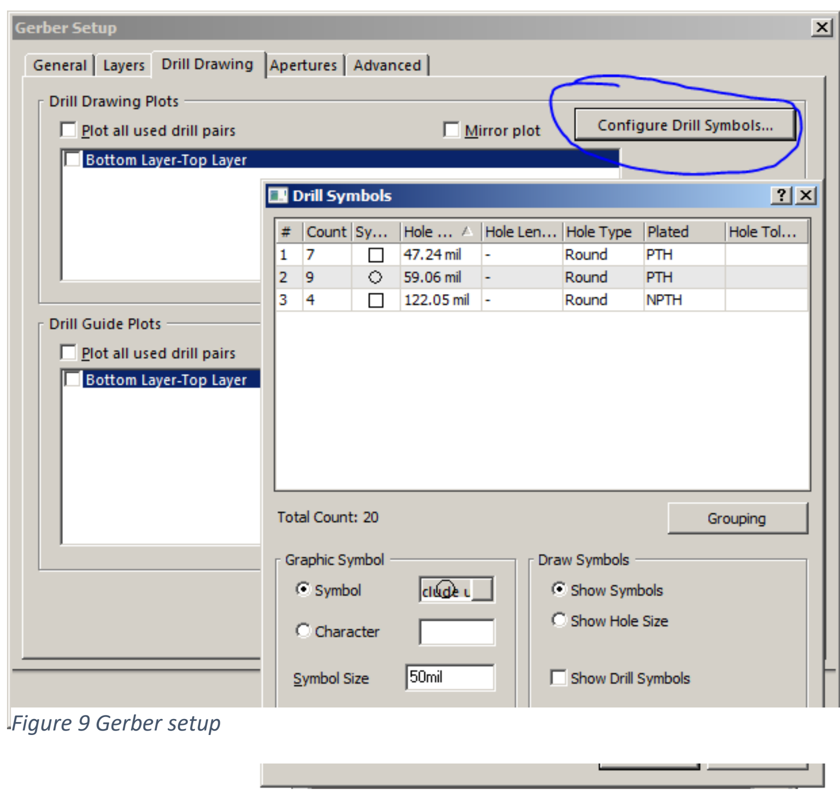


Figure 9 Gerber setup

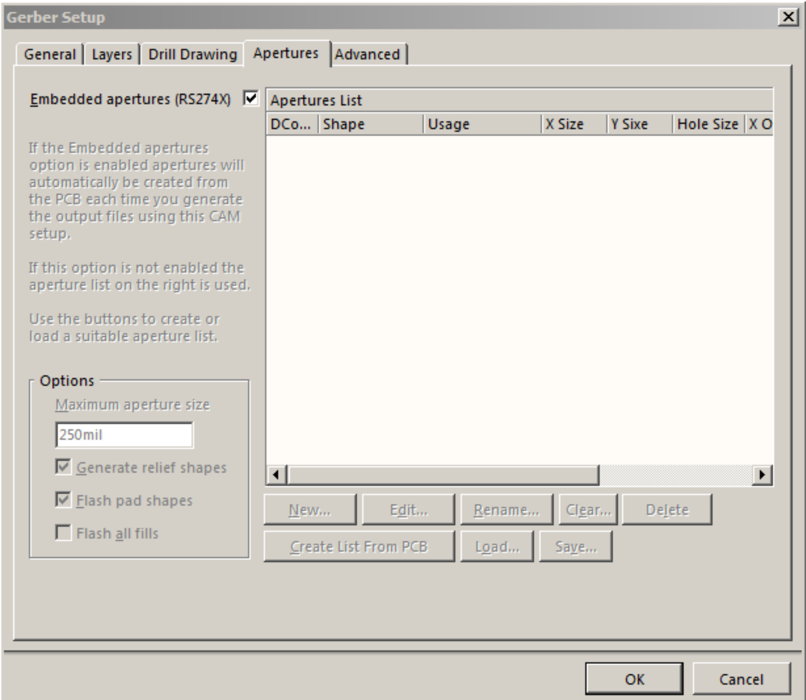


Figure 10 Gerber setup

If you click on the next tab "**Apertures**" nothing needs to be set up but just incase match it to Figure 10.

The last tab of the Gerber Setup should look like Figure 11

Then finally click the button "**OK**", and it should generate the correct files.

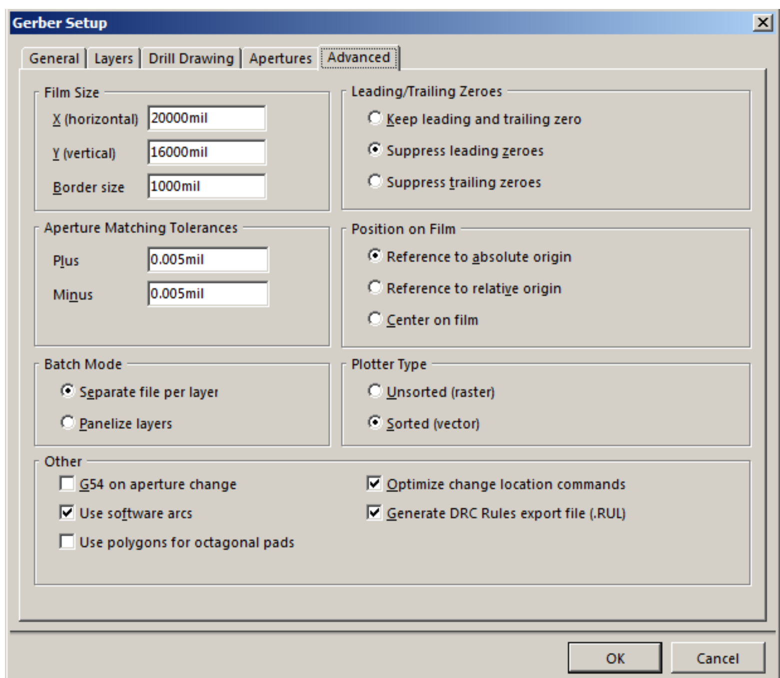


Figure 11 Gerber setup

Now that the Gerber file is created we can setup the drill file. Get back into your PCB document. Then click **File → Fabrication Outputs → NC Drill Files**. Setup the settings according to Figure 12. Select **OK** and the drill file will be created.

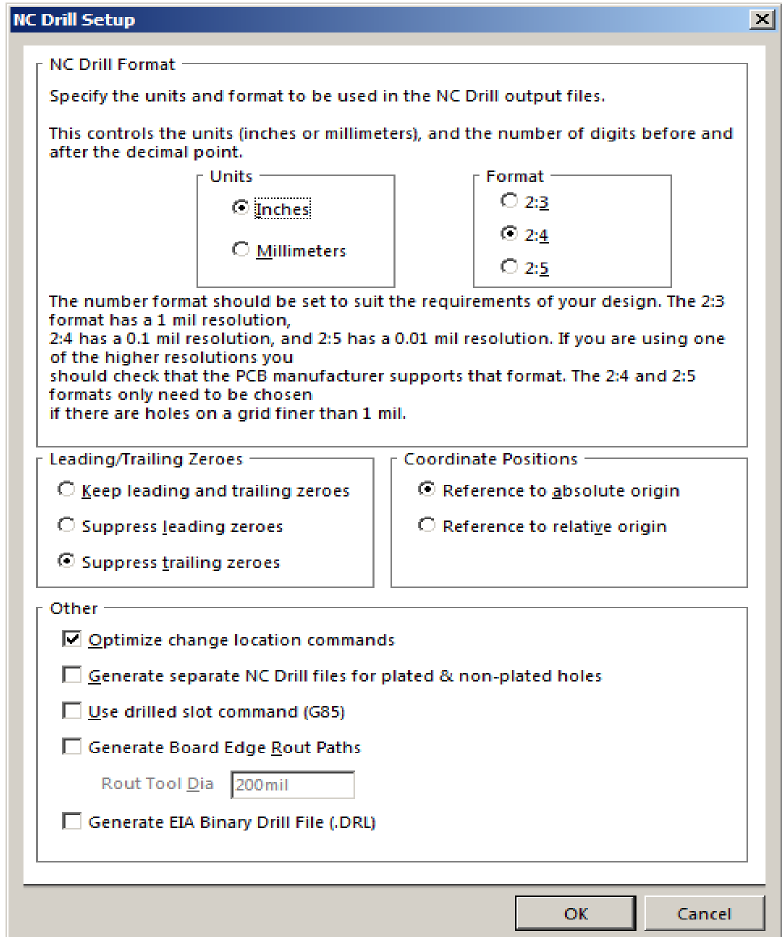


Figure 12 NC Drill Setup

Step 4: Switching Buck Regulator Creation

As stated in the disclaimer for this project we will be walking you through how to use *Altium* to design the board that we built to take our 90W power supply and step it down into multiple different rails. These include a 12V at 5A, with a 5V and 4.3V rail sharing 5A. These are used to power all the devices on board.

Follow Step 1: "Creating the Schematic Document" and refer to Figure 13.

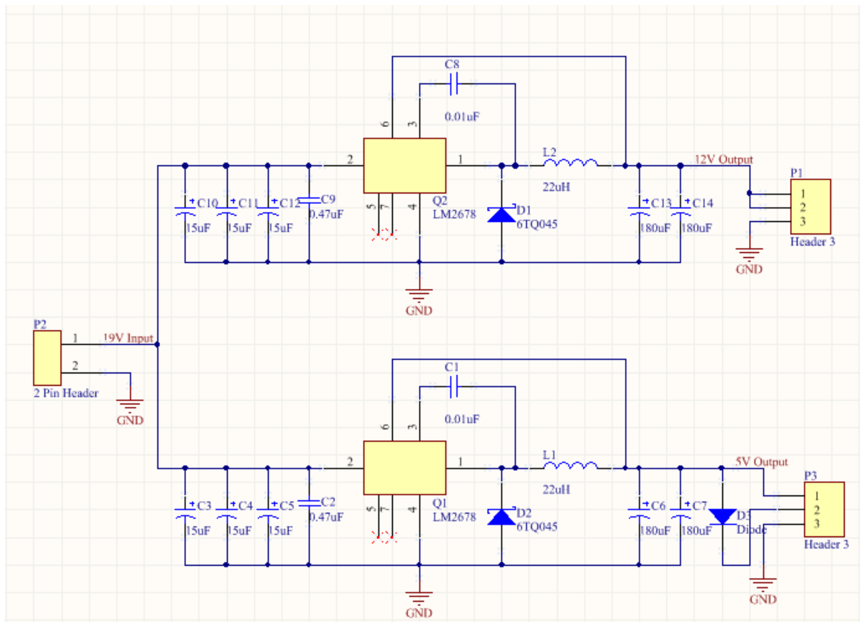


Figure 13 Power Schematic

Follow Step 2: “Creating The Printed Circuit Board”, and refer to Figure 14.

Follow Step 3: “Fabrication Outputs”

NOTE: Making sure there is enough room for heatsinks on the LM2678s and the Schottky diodes. You can measure distance between objects by pressing “ctrl + m” keys on your keyboard or going **Reports → Measure Distance**. Before you finish your PCB make sure the trace widths are able to handle ample amounts of current. In this case a rule of thumb should be that your traces can handle at a minimum 6A. There are many trace width calculators on the internet that could be used for this step.

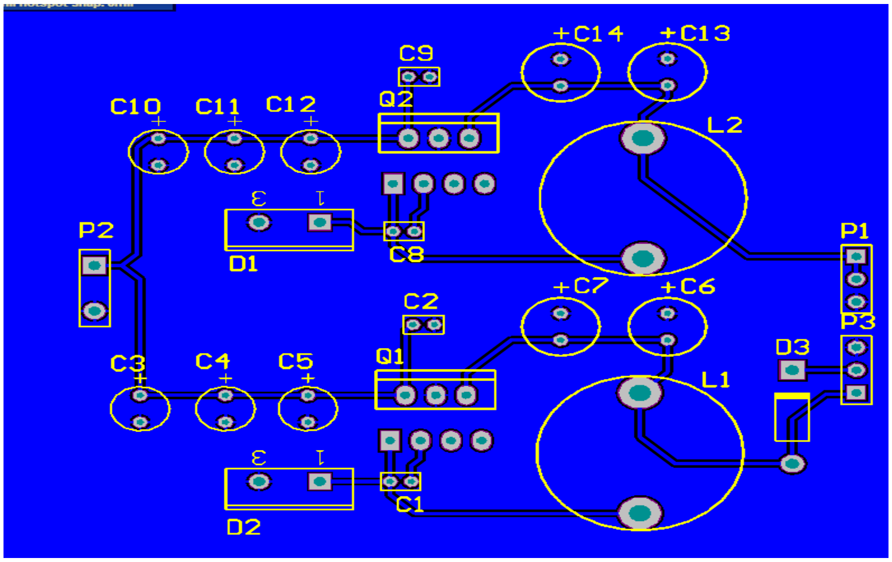


Figure 14 Power PCB

Step 5: Soldering the PCBs

In this step of the guide you will complete the process of assembly of the PCBs designed in *Altium Designer* from the previous 4 steps. It is important that you take your time being careful that the devices are oriented properly. To complete the following procedure make sure you have all the required equipment available at hand:

- Soldering Iron, knife tip
- Solder (lead is easiest to work with, lead free is an option)
- Solder sucker
- Copper braid
- Isopropyl alcohol
- Liquid flux
- Electrician's tape
- Single inline package header pins

Please refer to Appendix E for the following electronic devices so they can be orientated correctly on your PCB:

- LM2678 (Switching Regulator)
- 6TQ045PBF (Schottky Diode)
- *Bluetooth* module

Start with the regulator board and start by dropping a single component from the top (non-ground plane side) through so the tips of the leads are sticking through the board. Apply flux to the pads and then solder the component in. Be careful you do not apply an extended time of heat to the pad as it could come unglued and you will have to try to rework it or get another PCB made for you. Continue to drop a single component in at a time until your entire board is finished. **DO NOT** solder wires into the inputs and outputs of the board until the last procedure of this guide.

Next grab the breakout board. It is easiest to start by soldering the *Bluetooth* module in first. Before you begin, grab a roll of electricians tape, and cover the bottom of all of the exposed pads besides the 5 used with tape. This will ensure the unused pads do not short to the ground plane. Be very conscious that the solder flows from the module to the pads for all 5 connections. Next grab the amplifier and your header pins and break off the correct amount for all of the holes in the board. poke the square copper through to the writing side of the amplifier module and solder the header pins in. After soldering all header pins in drop the board into the breakout board (drop from top side through to ground plane side) and solder the pins in.

This completes the construction of PCBs. Be sure to be careful when soldering the components into the board and if in doubt when orientating a component refer to the datasheet of Appendix E for the pinout. And make sure to give all solder connections a good scrubbing with isopropyl alcohol to remove any flux residue.

The Software

For this part of the guide we will be working with the software behind the Mirrorage. By the end of this section you will have a *Raspberry Pi 2* that is fetching data serving up your own web page which will allow you to display your local weather along with the current date and time. To achieve this you will need the following:

- *Raspberry Pi 2* running Jessie OS
- Internet connection
- Wifi USB Dongle
- 7" HDMI compatible display
- Ethernet cable

This can also be done using a *Raspberry Pi 3* if you wish, that way you will not require a wifi dongle as it will be built in. For more information on the components listed, it can be found in Appendix E. Also in this section, any text highlighted in off-yellow direct text that will be typed in your *Raspberry Pi's* command line.

Step 1: Updating Your *Raspberry Pi*

To start this project we need to make sure that our chosen *Raspberry Pi* is up to date and has all the supporting software to backup the steps we are about to run through! To start this project we will require a reliable internet connection. With your *Raspberry Pi* connected via an ethernet cable you will be required to update and install the latest packages for your *Raspberry Pi* and their versions with the following commands in your *Raspberry Pi'* terminal.

Sudo apt-get update

Sudo apt-get upgrade

Using "Sudo apt-get update" you will be downloading the latest packages. Once you have completed this step using the "Sudo apt-get upgrade" command you will start the installation process finishing the process of updating your Pi.

Step 2: Wifi Configuration

Now that we have a Pi that is fully up to date we can now configure the Wi-Fi so that we do not need to hardwire the final project. This guide uses a *Raspberry Pi 2*, therefor you will need to plug your Wi-Fi dongle into the USB port.

With this out of the way we will need to edit the Pi's networking files, this can be done by entering the following command into the Pi's terminal window:

Sudo nano /etc/network/interfaces

Now that you have access to the file you will be required to replace the text currently present in the document with the following:

auto lo

iface lo inet loopback
iface eth0 inet dhcp

auto wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.con

Once the text is replaced we will need to finalize and save the file. This can be achieved by pressing the key combination “Ctrl-o”, “Ctrl-x” followed by “enter”.

Now moving on to edit the Pi’s configuration file wpa_supplicant.conf. This is done by entering the following into the Pi’s command line:

sudo nano/etc/wpa_supplicant/wpa_supplicant.conf

Now that we have the file open we can replace the text in the file with the following text making sure to fill in our own network information in the appropriate areas that are bolded.

ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1

network={
    ssid="**YOUR WIFI NETWORK NAME**"
    psk="**YOUR WIFI PASSWORD**"
    proto=WPA
    key_mgmt=WPA-PSK
    pairwise=TKIP
    group=TKIP
    id_str="**YOUR WIFI NETWORK NAME**"
}

Once this has been completed we have to finalize, complete, and exit our file by using the same commands as above (“Ctrl-o” to write out and save the file, then type “Ctrl-x” and “enter” to exit).

Once you have exited make sure to reboot your *Raspberry Pi* for the changes to take effect by entering the following command:

Sudo reboot

Step 3: Static IP Setup

Now that we have the wifi setup we need to set up a static IP address so that we will always know how to reach our Pi reliably.

To do this we have to discover our default gateway IP address by typing in the following code into our command line:

route -ne

This command will spit out some informations back into your command line which will include the IP address we are looking for. Write down the IP in the same row as wlan, an example is shown below

```
pi@raspberrypi:~ $ route -ne
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG        0 0          0 wlan0
0.0.0.0         192.168.0.1     0.0.0.0         UG        0 0          0 wlan0
192.168.0.0     0.0.0.0         255.255.255.0   U         0 0          0 wlan0
```

Figure 15 IP

Now that we have gathered our Gateway IP we will need to find the IP of the domain name servers by typing in the following:

cat /etc/resolv.conf

Write down the IP next to nameserver, example shown below.

```
pi@raspberrypi:~ $ cat /etc/resolv.conf
# Generated by resolvconf
nameserver 192.168.0.1
nameserver 2001:558:feed::1
nameserver 2001:558:feed::2
```

Figure 16 IP

With the necessary information acquired, we can apply this IP's to the "dhcpcd.conf" file. This is done by typing in the following command:

sudo nano /etc/dhcpcd.conf

Once that you can see that the file has come up we are going to replace the current text with the following, be sure to fill your own information into the bold text that match your network along with the static IP you choose to use:

interface eth0
static ip_address=**YOUR DESIRED STATIC IP**

interface wlan0
static ip_address=**YOUR DESIRED STATIC IP**

static routers=**DEFAULT GATEWAY IP**
static domain_name_servers=**NAMESERVER IP NAMESERVER IP** 2001:558:feed::1 2001:558:feed::2

If done correctly it should mimic the following image in format with your found IPs from the previous steps, Team Mirrorage use a static IP of 192.168.0.50.

```
interface eth0
static ip_address=192.168.0.50

interface wlan0
static ip_address=192.168.0.50

static routers=192.168.0.1
static domain_name_servers=192.168.0.1 192.168.0.1 2001:558:feed::1 2001:558:feed::2
```

Figure 17 Static

Once this has been completed we have to finalize, complete, and exit our file ("Ctrl-o" to write out and save the file, then type "Ctrl-x" and "enter" to exit) follow by:

Sudo reboot

You will now be able to use your *Raspberry Pi's* Static IP. Keep in mind you still may receive a dynamic IP when typing "ifconfig" into the terminal but once the process is complete but to verify you have executed the steps correctly by typing the following command:

ip addr show

Referencing the image below you can see highlighted in red you can see a dynamic IP on top, with the static outlined in red below.

```
pi@raspberrypi:~ $ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group d
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast stat
default qlen 1000
    link/ether b8:27:eb:be:e4:28 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::7c02:8d56:26b6:1020/64 scope link tentative
       valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP grou
n 1000
    link/ether 00:0f:60:05:b9:a8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.12/24 brd 192.168.0.255 scope global wlan0
       valid_lft forever preferred_lft forever
    inet 192.168.0.50/24 brd 192.168.0.255 scope global secondary wlan0
       valid_lft forever preferred_lft forever
```

Figure 18 inet

Step 4: Hosting The Webpage

Now that we have our *Raspberry Pi* running wirelessly using a static IP we can begin to get things ready for our Mirrorage display. First, we must install *Apache Web Server* application onto your Pi that will serve as a host your web page. This begins by entering the following commands onto your Pi:

```
sudo apt-get install apache2 -y
```

Once *Apache* is installed we will need to download and install *Chromium* browser onto our Pi by entering the following command:

```
sudo apt-get install chromium
```

Followed by a reboot:

```
Sudo reboot
```

Once *Chromium* is installed, it's time to test the *Apache Web Server*. To do so we need to open the Chromium Browser on your Pi and type in your static IP address into the URL. You should then be greeted with the Apache Test Page. This page shown an HTML file that is stored in your Pi's "/var/www/html" directory under the name "index.html".

The next step will be to create the files that are going to replace the *Apache* test "index.html" file. The files used in this guide are available in Appendices B-D or are available on *github* at https://github.com/marshall9570/mirrorage. This guide will show the method by obtaining the files through *github* as it requires less work.

To acquire the files we will have to navigate to https://github.com/marshall9570/mirrorage and download the ZIP project file as shown in the caption below.

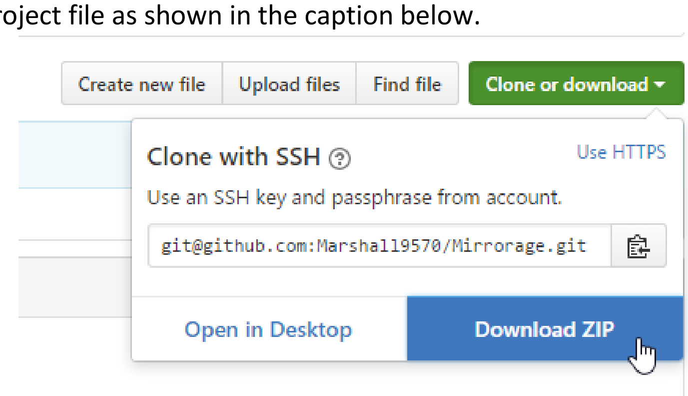


Figure 19 SSH

With the project downloaded, unzip the file to a convenient location on your local PC that will be easy to find (eg. desktop).

With the files saved it's time to sign up for a free account with *Open Weather Map* at https://home.openweathermap.org/users/sign_up. This is account will be used to get the weather information to your webpage.

Once signed up, obtain your API key from the account information page, it will be a long string of alphanumeric characters. Write this down as you will need it shortly.

Next you will need to obtain your city ID that you require the weather data for at http://openweathermap.org/help/city_list.txt. Write this down as well.

With this information to make the webpage you own from *Open Weather Map*, open the Functions.js file from the project and place your API key (leave in quotes) as well as your City ID in the placeholders (no quotes) at the top of the document similar to the following capture below and then save the file.

```
//API Key Specific to user account with openweathermap.org
var API_KEY = "YOURAPIKEY";

//city ID provided by openweathermap.org
var cityid = 6049429;
```

Figure 20 Your API key

Next you will need to download *Filezilla* from [*https://Filezilla-project.org/*](https://Filezilla-project.org/) and select the download button as shown in the capture below. *Filezilla* will be the application we use in transferring the project files over to the *Raspberry Pi.*

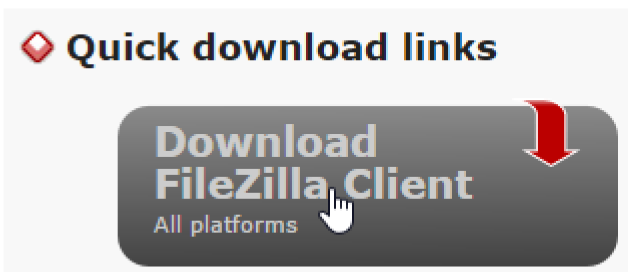


Figure 21 FileZilla

When *Filezilla* is successfully downloaded, open it and fill in the remote information on the top portion of the window. The Host text box will contain your *Raspberry Pi's* static IP address, the username and password will be the default ones set by the Pi itself (username being "pi" and password being "raspberry") and the port will be set to 22 (default). Once this is filled in press connect as shown in the capture below and you will be shown your remote computers files on the left-hand side of the window and your pi's file directory on the right-hand side of the screen.

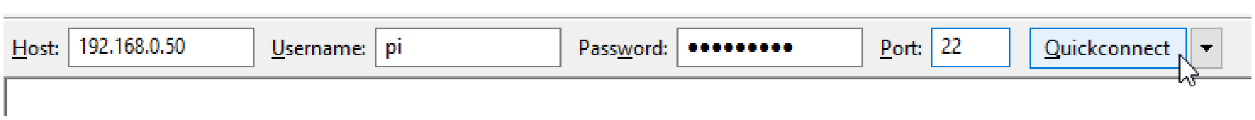


Figure 22 Static IP

Locate the project folder download from *github* on your local machine and drag it across the screen and into the *Raspberry Pi's* desktop directory as shown below, no need to bring over the PCB files.

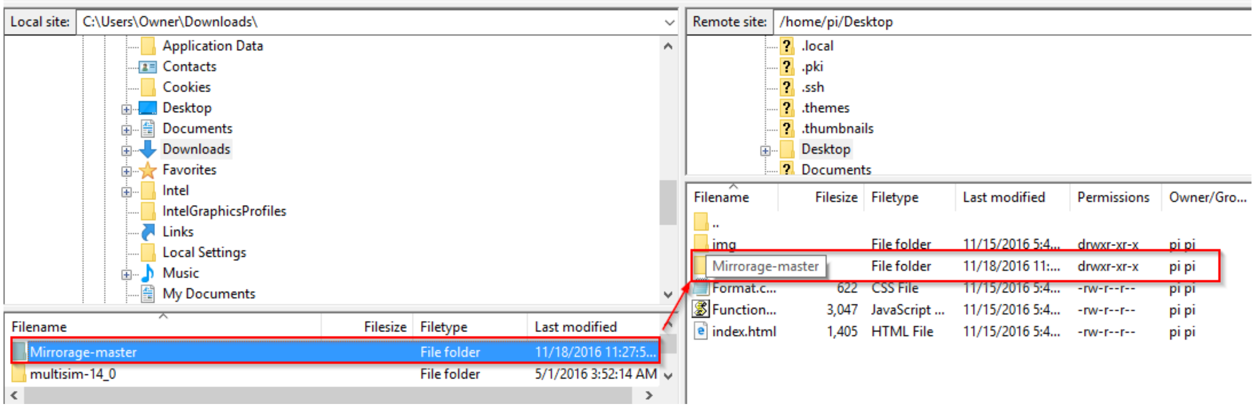


Figure 23 The files

When the files are correctly transferred, you can close *Filezilla* and log back into the *Raspberry Pi* via SSH or manually and move the contents from the Mirrorage-master folder into the the "var/www/html" directory followed by a reboot.

When the *Raspberry Pi* turns back on it will be hosting the webpage over its static IP address. This can be tested by opening a browser on any device that lies on the same network and typing the static IP address into the url!

Step 5: Setting up Pi to Kiosk mode:

Install Chromium and unclutter onto your *Raspberry Pi* by entering the the following command into its command line:

sudo apt-get install chromium x11-xserver-utils unclutter

sudo apt-get install unclutter

We will now need to alter the *Pi's* autostart file to open up chromium on boot at full screen without any error dialogs, sleep disabled and without a cursor. This can be done by by entering the following command:

sudo nano ~/.config/lxsession/LXDE-pi/autostart

In this file we will comment out the screensaver line with "#", run unclutter to remove the cursor, disable any power management settings using xset, disable any error dialogs with sed and begin chromium in kiosk mode at startup to host the web page (use your static IP where indicated). The ideal text is shown below:

@lxpanel --profile LXDE
@pcmanFM -- desktop --profile LXDE
#@screensaver -no-splash

@unclutter -idle 0.1 - root
@unclutter -idle 0

@xset s off
@xset -dpms
@xset s noblank

@sed -i 's/"exited_cleanly": false/"exited_cleanly": true/' ~/.config/chromium/Default/Preferences

@chromium --noerrdialogs --kiosk STATIC IP ADDR --incognito

When complete the file should appear same as the capture below.

```
pi@raspberrypi: ~
  GNU nano 2.2.6            File: /home/pi/.config/lxsession/LXDE-pi/autostart

@lxpanel --profile LXDE
@pcmanfm --desktop --profile LXDE
#@xscreensaver -no-splash

@unclutter -idle 0.1 -root
@unclutter -idle 0

@xset s off
@xset -dpms
@xset s noblank

@sed -i 's/"exited_cleanly": false/:"exited_cleanly": true/' ~/.config/chromium/Default/Preferences

@chromium-browser --noerrdialogs --kiosk 192.168.0.50 --incognito
```

Figure 24 Kiosk Mode

Next we will save and exit the file using the key combinations “Ctrl-o”, “Ctrl-x” followed by “enter”.

now with a reboot your PI should correctly boot into the webpage at full screen and ready to be mounted in the enclosure. It will appear similar to the capture below but with different information as far as location and weather information.

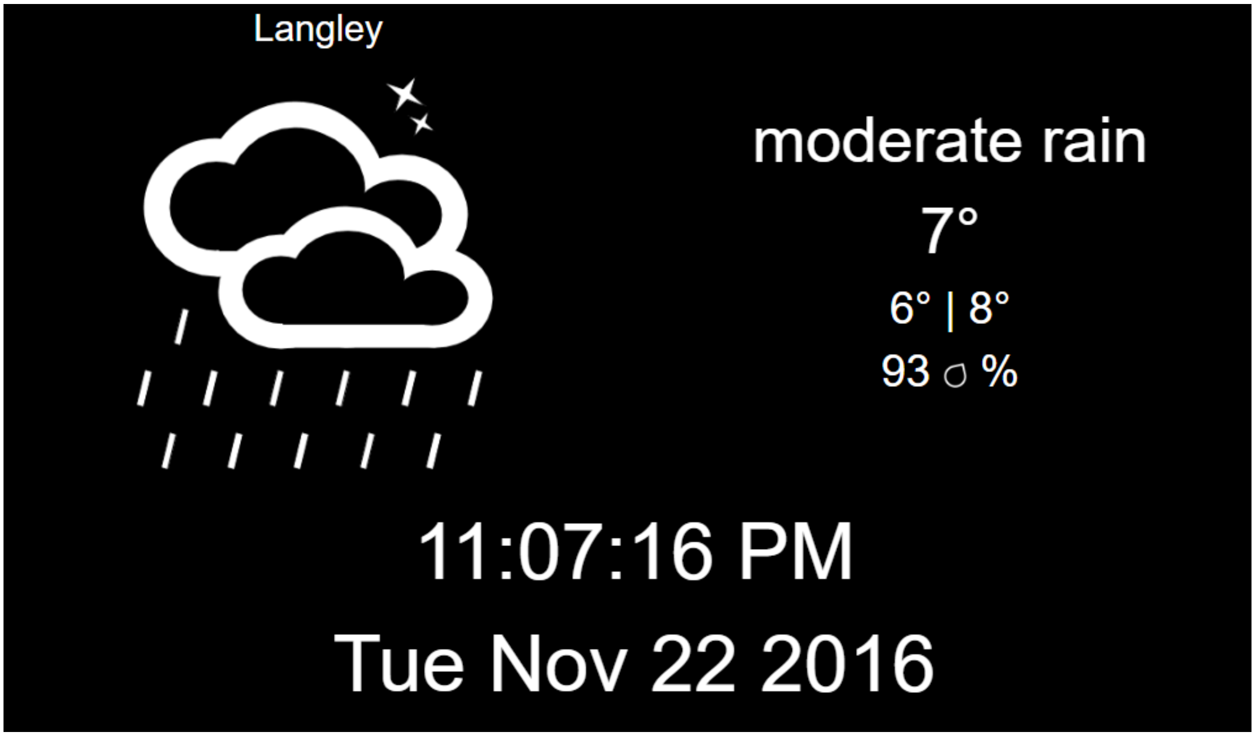


Figure 25 Weather Display

With all this completed, your *Raspberry Pi* has been successfully set up to be used in your Mirrorage.

Enclosure

For this part of the project we have break it down into stages so that the process becomes multiple small steps that end in a finished project. For the woodworking part of the project you will need access to:

- A router
- A table saw
- A chop saw (optional)
- Ear protection
- Eye protection
- Finishing nails
- Wood glue
- ¾" Woodstock
- Plainer
- Chalk Paint
- Mirror (would recommend getting a mirror that has a thickness in inches like 1/8" it will simplify the process greatly)
- Sand paper
- Primer
- Wood filler
- Paint/Stain

Step 1: Safety First

While working with the tools that will be required to finish this project it is important that the age old saying of safety first is taken to heart. Proper hearing and eye protection should be used at all time during the process (as seen in Figure 26) with sleeves pulled up and all jewelry tucked away so that it cannot be pulled into any machinery.

Figure 26 Safety gear

Step 2: The Right wood for the job

When choosing what wood you want to use for this project you have to keep in mind what you want to achieve with this project. A necessary requirement for the wood we choose is that it must be a hard wood. Choosing a hardwood is a necessary requirement since most high-end furniture or construction that is needed to last is made using hardwood due to its resilience. As well as lasting longer, hardwood is less susceptible to warping due to humidity and marring from cleaning or moving the mirror. With this choice out of the way we have narrowed our options. Some examples of hardwood include alder, balsa, beech, hickory, mahogany, maple, oak, teak, and walnut. With those options available the next thing to keep in mind is how you want the enclosure to be finished. This is obviously a personal choice that will be decided due to design language that you have in your home. In this guide we will be finishing the enclosure in gloss white so we are not paying as much attention to the wood grain as the quality of the wood itself so that we do not have to do hours of sanding and filling before we paint. For those who are staining you will want to take a test piece and test different stains before proceeding along with the project. The stock we settled upon was maple since it is what we had available that also met our requirements that are listed above.

Step 3: Rough stock

We will start by cutting the stock for the outside enclosure so that we end with rough stock with the dimensions of ¾” by 1 ¼”. This rough stock is the starting point for the frame of our enclosure and will shortened to the correct size depending on the size of mirror you desire. With different sizes of mirror you will need a minimum of two pieces of wood with a length of $Mirror\ Width + 1$” and two pieces of wood with a length of $Mirror\ Height + 1$”. Though these equations will give you the bare minimum length required for your desired mirror enclosure it is important to leave a margin of error while woodworking to allow for any errors. With this kept in mind to allow for a margin of error for a mirror with the dimensions of 22” x 30” we cut our rough stock to lengths of 2 x 24” and 2 x 32” . Now that we have cut the stock for the outside of the enclosure we will proceed to cut the stock for the electronic enclosure that is situated inside the external frame. We will continue to use ¾” stock for the inside of the box making it so that we can make the most out of the supplies that we purchased.

Step 4: Rabbits?

For the next step of the enclosure build we will be cutting the rabbit that the glass slots into. During our testing process with the mirror we discovered that a ¼” rabbit was optimal for the project. After setting up the router to get our ¼” rabbit we ran a test piece through to double check our measurements. As you can see from Figure 27 the rabbit was ¼”(6.35mm) wide and 0.19685”(5mm) deep. The depth of your rabbit will vary depending on the thickness of your glass, the requirement of our rabbit is that the depth of the rabbit must be the same as the thickness of the glass so that the glass sits flush with the wood. After verifying the setup of the router we proceeded to rabbit the stock that we had marked for use for the external enclosure. Once you have reached this part of the DIY guide you should be left with four pieces of external stock that has been rabbeted and the four pieces of rough stock that we put aside intended for the inside electronic enclosure.

Figure 27 The rabbit

Step 5: Making the external frame

Now that we are done with rabbiting the external enclosure we are going to have to cut a 45-degree angle on each end of all our outside pieces so that they fit together to make a frame (see Figure 28). The cut must be made so that the width of your mirror is equivalent to the width of the highest portion of the rabbit. Once you mark that point you must take a triangle, making sure you have it pressed firmly against the wood and mark a 45-degree angle into the stock. Now that you have your line you may cut the piece, cutting into the waste wood leaving a clean cut and some room to play if we slightly miss measured. Proceed to do this for the other pieces of the enclosure until you have four pieces of wood that fit snuggly together. Finally before we move onto to the inside enclosure we

Figure 28 Making the frame

need run a dado into the bottom of the wood so that we can run the power cord out of the enclosure. This will be a personal choice depending on the location you want the plug to come out of and the size of the power cable that you end up getting with your supply. When working on the internal frame we must make another dado that lines up with our external dado so that the wire can go through both the inner and outer enclosure, we will get back to that later.

Step 6: Let there be light

We now have something that looks vaguely like an outside frame however we are not done just yet. With the lighting being built into the mirror we need to build in channels so that the LEDs sit flush in the wood. This is to eliminate any bright points that could appear in the diffuser due to the distribution of our LEDs. When testing, we found that the optimal depth for the dado was ¼" (6.35mm) deep by still emitting light but not seeing any hot points when looking through the mirror. The thickness of the dado however will depend widely on how thick the LED strip that we decide to place in the enclosure is. To stay on the safe size, We ended up making the dado ½" wide and ¼" deep to allow for the strip to continue to fit after sealing the pieces.

Step 7: Moving inside

Remember that wood that we had cut for the inside box that I told you to put aside? Well it's time to pull it back out and get ready for some more fun. Now that we have our rough saw we are going to have to head back to our table saw. This stock well still ¾" thick must be the height of the outside piece of wood minus whatever your rabbit ended up being. For us with a rabbit of 5 mm we needed to take our inside stock down 5 mm from our original height. Before we move onto rabbits we must put two more dados into our enclosure. The first one is so that we can run our power cable out of our enclosure you will need to measure it and cut another dado so that it lines up with the dado you cut in the outside of the enclosure. About an inch up from your first dado you will need to cut another one of the same size whose purpose will be to be a channel for the LED power cables and speaker wires.

Step 8: More Rabbits!

Now that we have everything all set up it's time to add more rabbits! You know why? More LEDs! You can't have a smart mirror without lights that just wouldn't be fair! As we did with our first dado on the outer case we are going to set the LEDs a quarter of an inch into the wood on the outside of the inner case so that there will be no hotspots to show through. Now for those who are not familiar with the difference between a rabbit and a dado the difference is basically the position of the trench or inset in the wood. A rabbit is a cut at the edge of the wood where as a dado is just a trench in the wood itself. Now after that quick explanation I know it does not make much sense as to how a rabbit is going to help hide the LEDs but I have the answer for you! At the start of this DIY guide when we worked on the glass part of our build we applied a black vinyl coating to the back of the mirror finish removing any leakage light through the mirror finish itself. If we rabbit the outside of the inner casing, we can hide the LEDs behind the black vinyl that we applied while still maintaining the correct spacing in our light box without intruding too much into our inner enclosure.

Step 9: The backbone of the project

Now that we have our inner and outer enclosures we need to put a back piece on the enclosure. This is when the 1/8" piece of ply that we had talked about comes into play. Now using the table saw we are going to cut the piece of ply so that it will be the width of the outside of the frame by the height of the outside of the frame. In other words, this means that the back piece in our case must be 23" by 31" accounting for the extra half inch on all sides of the glass. If you did not decide to build a 22" by 30" mirror the dimensions of the back piece will just be the dimensions of your mirror with a half inch added all around. With this done the enclosure is starting to take shape and look like a semi complete project. (see Figure 26)

Figure 29 The back piece

Step 10: Support

Now that we have a back piece, inner enclosure, and outer enclosure we just need some support pieces to tie it all together. These will just be small pieces that run between the inner and outer squares as well as a support between the two. These will be tall enough so that they are reaching just to the height of the lowest LED dado and 2.5" long and will be placed between the inner and outer frames.

Step 11: Connecting it all together

Now this is when everything comes together. We will start with cutting some corner supports from the same stalk as we did the side supports. With these they will use the same method that we did on the outer enclosure we will cut 45 degree angles into the wood so that both angles originate on the same side. This will allow us to snug them in so that they span from one length of the enclosure to another length on the enclosure. A visual representation of the supports can be seen in Figure 27. Once these are done we are done with power tools and wood for this project we will finish in step 12 with finish nails, glue and some filler.

Figure 30 The supports

Step 12: Show it off!

Now that the enclosure has been cut and measured we are onto the good stuff. For this we will need to grab a few things including;

- Sand paper
- Primer
- Wood filler
- Finishing nails
- Wood Glue
- Paint

This part is when you will have the chance to show your artistic side. We decided to go with a semi gloss white cause that is what suited us however if you want to put a stain you would just have to take the appropriate steps to finish the mirror off in the finish you desire. Now that we are onto the last stretch you may have the desire to finish and put it together however the most important part of any paint job is the preparation. We will need to sand down all our pieces to a level in which we feel is good. A good rule of thumb is to get it to the point where the wood is smooth enough were you are comfortable with running your hands on the wood with no fear of a splinter. Once our prep is done we will put a coat of primer on sealing the enclosure from any moisture that it may experience. Now that all the individual pieces are sanded we are going to glue all the pieces into place and secure them with finishing nails giving us two methods of securing so if one lets go we always have a backup. We will now be mounting our electronics, we will require a package of 4/40 bolts and two packages of 4/40 nuts. Placing your electronics into the enclosure mark out there the mounting holes line up. Using the correctly sized bit make holes for the bolts to come up through the back of the enclosure. Once the holes have been made you can run the bolts up through the back of the enclosure and place a nut on the inside of the enclosure securing the mounting posts in place. You can now place components onto your posts and are done with your mounting hardware. Now take the wood filler and fill any imperfections you can see in the wood including the holes created by the finishing nails we just put in. Another quick sand and we should be ready for paint! Using whatever paint you have chosen you can finish off your enclosure and put it aside to dry as of now your enclosure is complete.

Putting it All Together

With all previous parts of the guide done we can begin to put everything together to get a final product. This section will require some creativity as there are many ways everything can be positioned in the housing. Since there is no direct order to get everything in place, the capture below shows the inside of the original Mirrorage product. With this capture, there will be numerations regarding each main step that needs to be completed. 20 gauge stranded wire should be used to wire everything up in the finish product as it can handle supple amounts of current. As well as you will not have to worry about the wires breaking if you move it to many times.

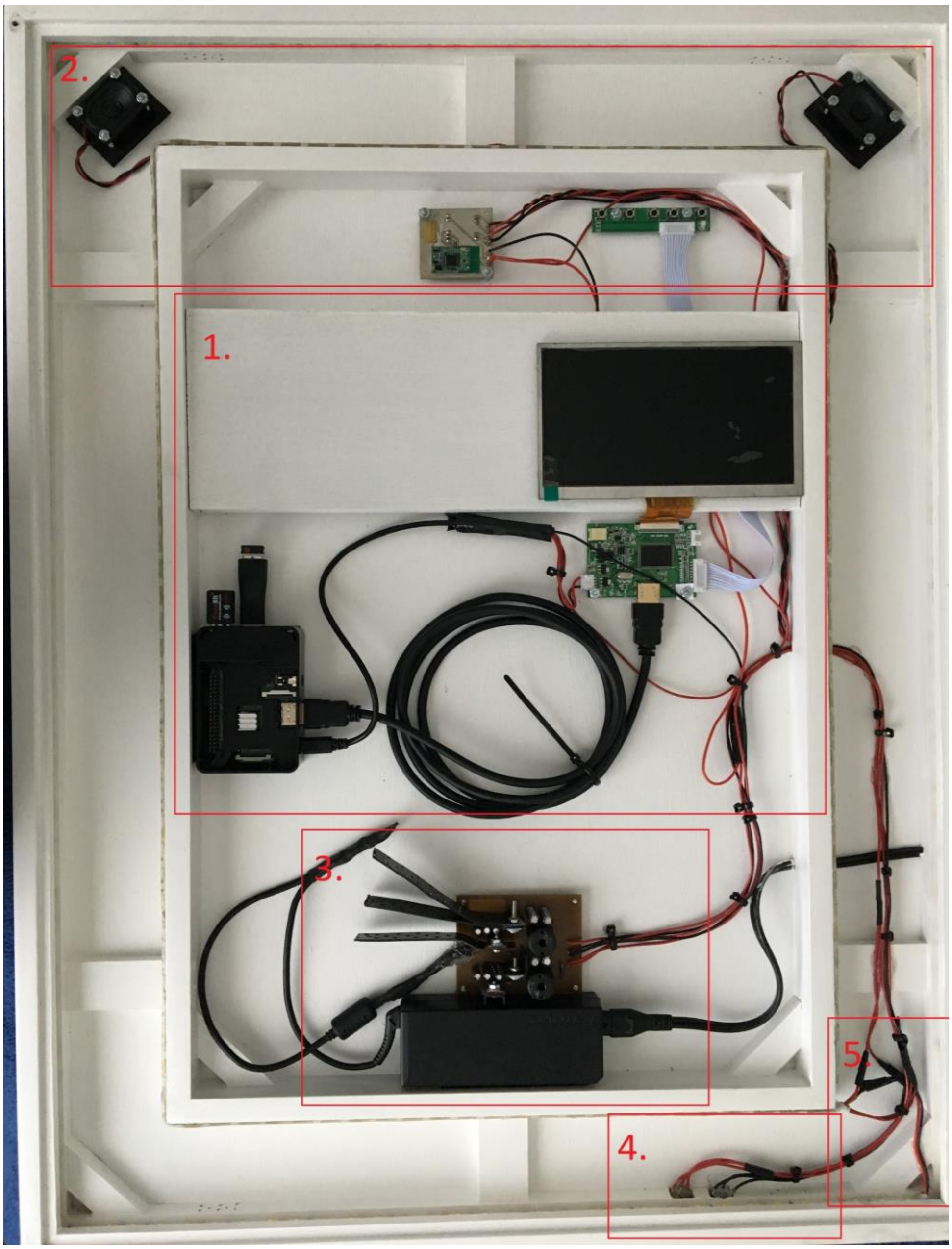


Figure 31 Inside the enclosure

1. The *Raspberry Pi 2* and display monitor being powered by the 5V rail of your power PCB but are ran through a switch to allow the user to control when the Pi is running or not. Mount the components through the backplane of the enclosure or by any sort of adhesive. When mounting the monitor it is important to use something with some spring to allow a flush sit against the mirror. Secure the HDMI Cable with adhesive to prevent it was weighing down the ports.

2. When mounting the breakout PCB, make sure that the *Bluetooth* module is facing out of the glass. This will ensure that the antenna doesn't have the ground plane interrupting its signal. Use screws through the backplane of the enclosure to secure the board. Run the wires for the speakers in a twisted fashion ensuring noise doesn't become an issue.

3. The output of the laptop power supply is wired directly into the regulator PCB. Make sure to fashion some sort of heatsink onto the back of all diodes and regulators shown in Figure 28. For the addition shown two pieces of aluminum have lots of holes drilled out to allow for lots of surface area to be gained. You should use a very little amount of thermal paste between the heatsink and the back of the components. Since the 5V regulator isn't sourcing as much power smaller heat sinks may be used. Drill ⅛" holes through the back of the enclosure then screw in 4/40 machine screws and secure the PCB with a nut on the top. The 12V output wire from the PCB needs to go directly to the LEDs (common source). One ground wire goes into a single switch. while the other needs to head up to the breakout board PCB, as well as the 4.3V output goes to the breakout board.

4. For the switches, double pole, double throw switches are required. We wired the LEDs and the power source for the *Raspberry Pi* through the switches. To make the LEDs work we wired them in a low side configuration tying the positive leads of both the white and the red LED strips to positive twelve volts. We then wired the negative of the white strips to one pole of the switch, the negative of the red LED strips to another pole and the neutral of the switch went to ground. This allowed us to make or break the ground connection to the LED strip that we desired and making it so that only one strip could ever run at a time. For our power supply switch, we just wired it as you would a normal light switch in your home. When the connection is made, the power can flow through the switch to our unit however when the switch is turned off no power reaches the compute unit.

5. When testing different LED strips, we analyzed and studied the hue, brightness, and power draw of each unit. With these requirements kept in mind we decided to go with a single strip of red 5050 LEDs and two strips of 5630 LEDs to supply our white lighting. We then used an adhesive to secure the LED strips into the dados we ran in the wood that can be read about in "The Enclosure" section of the DIY guide "Step 6: Let There Be Light" for more information. The LEDs are powered by the twelve-volt rail of our power supply and run through the switch to allow independent operation of the LEDs when the *Raspberry Pi* is not powered.

With all the steps above complete you can begin to prep your glass for mounting into your enclosure. As mentioned earlier in this guide we are going to apply an additional film to the backside of the glass to ensure the only light that passes through the reflective film is from the display. There are many ways this can be achieved, the original Mirrorage product was coated with an adhesive chalk film. The important thing before applying anything is to check that light will not pass through it. Apply the film around the spot for the monitor and to the edges of the reflective film. An example of this is shown to the right in Figure 32.

Figure 32 Black vinyl

Once the film is applied you can set the glass into the into the enclosure and turn everything on to see your Mirrorage running in full effect. Now we will need to secure the mirror in place. To do this, you will need to use four 1”x1” pieces of brushed aluminum sanded to your specifications. These pieces of aluminum will be secured into place using ½” screws in each corner of the enclosure. Remember to drill the holes our carefully before setting the screw into place. It should appear like the capture Figure 33.

Figure 33 aluminum clips

With that, the process is complete and your Mirrorage is finally done. It is recommended to use a French Cleat design to hang the Mirrorage as it was evenly distribute the weight.

Congratulations!

Appendices

Appendix A

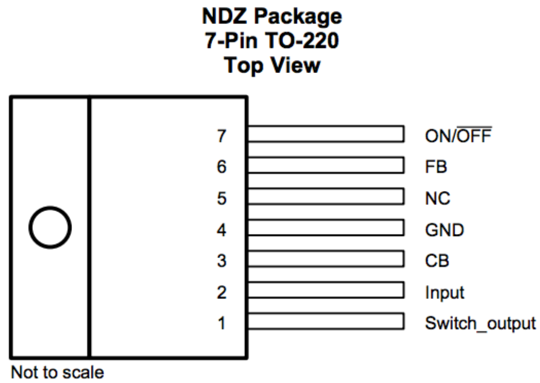


Figure 35 Buck regultor

LM2678 Pinout

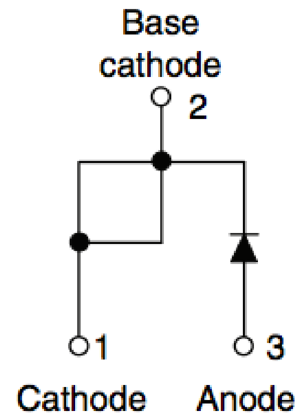
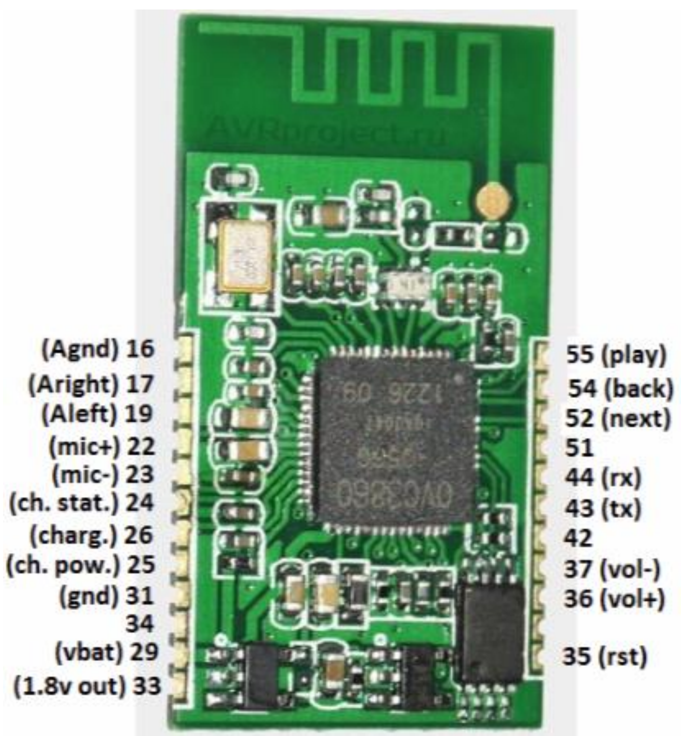


Figure 34 Pinout

6TQ045PBF Pinout



Bluetooth Module (16, 17, 19, 31, 29 are only pins used)

Figure 36 Bluetooth Module

Appendix B: HTML

```
<!DOCTYPE html>
<html>
<head>
	<title>Mirror</title>
	<link rel="stylesheet" type="text/css" href="Format.css" />
	<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
	<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
	<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
	<script type="text/javascript" src="Functions.js" ></script>
</head>
<body>
<div class="mirror">
	<div class="leftside">
		<div class="location"><span id="location">Unknown</span></div>
		<div class="top"><img id="icon" width="350px" height="350px" src="img/01d.png" /></div>
	</div>
	<div class="rightside"><br>
		<div class="highest">
			<div class="description"><span id="description">Unknown</span></div>
		</div>
		<div class="high">
			<div class="temperature"><span id="temperature">1</span>&deg;</div>
		</div>
		<div class="middle">
			<div class="temperatureminmax"><span id="temperaturemin">0</span>&deg; | <span
id="temperaturemax">2</span>&deg;</div>
		</div>
		<div class="low">
			<div class="humidity"><span id="humidity">0</span> <img src="img/Humidity.png"
height="20px" width="20px" /> %</div>
		</div>
	</div>
</div>
<div class="timedate">
	<div id="time"></div>
	<div id="date"></div>
</div>
</body>
</html>
```

Appendix C: JS

```
//API Key Specific to user account with openweathermap.org
var API_KEY = "747dab7d28711d5d4cad6f2021e38541";
//variables used to obtain data
var temp;
var loc;
var desc;
var tempmin;
var tempmax;
var humidity;
var icon;
//when the page loads, the variables will initialize corresponding to those on the html page
window.onload = function() {
	loc = document.getElementById("location");
	temp = document.getElementById("temperature");
	desc = document.getElementById("description");
	tempmin = document.getElementById("temperaturemin");
	tempmax = document.getElementById("temperaturemax");
	humidity = document.getElementById("humidity");
	icon = document.getElementById("icon");
	//building our object with some default values/strings incase something is offline these will remain
	var weather = {};
	weather.humidity = 35;
	weather.loc = "Location";
	weather.temp = "Actual Temp";
	weather.tempmin = "Min Temp";
	weather.tempmax = "Max Temp";
	weather.desc = "Description";
	weather.icon = "01d";
	//calls start time to begin
	startTime();
	//calls updatebyid to begin the data pulling from openweathermap
	UpdateById(6049429);
}
function update(weather){
	//removing trailing decimals from numerical values
	weather.temp = weather.temp.toFixed(0);
	weather.tempmin = weather.tempmin.toFixed(0);
	weather.tempmax = weather.tempmax.toFixed(0);
	//sending weather variables to html file to be displayed
	humidity.innerHTML = weather.humidity;
	temp.innerHTML = weather.temp;
	desc.innerHTML = weather.desc;
	tempmin.innerHTML = weather.tempmin;
	tempmax.innerHTML = weather.tempmax;
	loc.innerHTML = weather.loc;
	//updating soure location per icon response
```

```
        icon.src = "img/" + weather.icon + ".png";
}
//builds the request with the city ID and the user API key
function UpdateById(id){
        var url = "http://api.openweathermap.org/data/2.5/weather?" + "id=" + id + "&appid=" + API_KEY;
        SendRequest(url);
}
//creates an xmlhttp request which is a built-in js function that allows
function SendRequest(url){
        var xmlhttp = new XMLHttpRequest();
        //when the request gets a response back from openweathermap; callback
        xmlhttp.onreadystatechange = function(){
                //if the ready state is 4 it means we have recieved an object back
                //if the status is 200, it means the request was successful
                if (xmlhttp.readyState == 4 && xmlhttp.status == 200){
                        //using built in json functionality to allow use to sift through the data
                        var data = JSON.parse(xmlhttp.responseText);
                        //construct weather object
                        var weather = {};
                        //assigning the json we got back to the various fields used in display
                        weather.humidity = data.main.humidity;
                        weather.desc = data.weather[0].description;
                        weather.loc = data.name;
                        weather.temp = data.main.temp - 273.15;
                        weather.tempmin = data.main.temp_min - 273.15;
                        weather.tempmax = data.main.temp_max - 273.15;
                        weather.icon = data.weather[0].icon;
                        //call update weather function to send values over to html page
                        update(weather);
                }
        };
        //open with a get request with the url
        xmlhttp.open("GET", url, true);
        //once opened, send is called which will push our request to openweathermap
        xmlhttp.send();
}
//interval functions used to make sure everything remains up to date
//whole page will reload once a day
setInterval(function() {
        window.location.reload();
}, 86400000);
//weather will update every 10 mins
setInterval(function(){
        UpdateById(6049429);
}, 600000);
//function used to show real time using the system time
function startTime() {
  var today = new Date();
  var h = today.getHours();
```

```
   var m = today.getMinutes();
   var s = today.getSeconds();
   m = checkTime(m);
   s = checkTime(s);
          //check what hours is at to obtain am or pm
   var ampm = (h >= 12) ? "PM" : "AM";
          //brings weather to 12 hours time
          if(h>12){
                 h=h-12;
          }
          //send time data over to html page
   document.getElementById('time').innerHTML = h + ":" + m + ":" + s + " " + ampm;
   var t = setTimeout(startTime, 500);
          //send date over to html page
          document.getElementById("date").innerHTML = today.toDateString();
}
function checkTime(i) {
   if (i < 10) {i = "0" + i};  // add zero in front of numbers < 10
   return i;
}
```

Appendix D: CSS

```
.mirror {
            margin: auto;
             width: 1024px;
              height: 400px;
              text-align:center;
            font-size:52px;
            background-color: Black;
}
.leftside {
            font-size:30px;
            float: left;
width:512px;
            height:400px;
            color:white;
}
.rightside {
            float: right;
            width:512px;
            height:400px;
            color:white;
}
.temperatureminmax {
            font-size:36px;
}
.humidity {
            font-size:36px;
}
.body {
            background-color: Black;
}
.timedate
            {
            margin: auto;
            text-align: center;
            background-color: Black;
            color: White;
            height: 200px;
            width: 1024px;
            font-size:64px;
}
```

Appendix E: Suppliers & Sources

Product	Source
Glass	http://alliedglass.ca/
Film	http://sunshadewindowfilms.com/
Display	https://www.adafruit.com/product/2300
Raspberry Pi 2	https://www.amazon.ca/Raspberry-Pi-Model-Desktop-Linux/dp/B00T2U7R7I/ref=sr_1_2?ie=UTF8&qid=1480821357&sr=8-2&keywords=raspberry+pi+2
Wifi Dongle	https://www.amazon.ca/CanaKit-Raspberry-Wireless-Adapter-Dongle/dp/B0131YG3LG/ref=sr_1_2?ie=UTF8&qid=1480821396&sr=8-2&keywords=raspberry+pi+2+wifi
Bluetooth Module	https://www.amazon.com/gp/product/B00K85FW4G/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1
Amplifier Board	https://www.amazon.com/gp/product/B00EZI0RGA/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1
Custom PCB Boards	Camosun College
Enclosure	Donated
Paint	EVAN
Finishing Hardware	Home Depot & Camosun College
Speakers	http://www.digikey.com/product-detail/en/pui-audio-inc/AS04008PS-4W-WR-R/668-1116-ND/1464855
Switches	https://www.amazon.ca/gp/product/B01CI0R1NI/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1
Red LEDs	https://www.amazon.ca/gp/product/B00MWKYWDM/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1

Soft White LEDs	https://www.amazon.ca/gp/product/B00KCLZA48/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1
Lenovo Power Supply	Donated
MOSFETs	http://www.digikey.com/product-search/en?keywords=IRLB8721PBF
12 Volt Switching Regulator	http://www.digikey.com/product-detail/en/texas-instruments/LM2678T-12-NOPB/LM2678T-12-NOPB-ND/366923
5 Volt Switching Regulator	http://www.digikey.com/product-detail/en/texas-instruments/LM2678T-5.0-NOPB/LM2678T-5.0-NOPB-ND/363828
Capacitors	http://www.digikey.com/product-search/en?keywords=493-10516-1-nd http://www.digikey.com/product-search/en?keywords=493-5017-1-nd http://www.digikey.ca/product-detail/en/vishay-bc-components/K103K15X7RF5TL2/BC1078CT-ND/286700 http://www.digikey.ca/product-detail/en/tdk-corporation/FG28X7R1H474KRT06/445-173600-1-ND/5812205
Inductors	http://www.digikey.com/product-search/en/inductors-coils-chokes/fixed-inductors/196627?k=m8341-nd
Diodes	http://www.digikey.com/product-search/en?keywords=1N4448CT-ND https://www.digikey.ca/product-detail/en/vishay-semiconductor-diodes-division/VS-6TQ045PBF/6TQ045PBF-ND/812032
20 AWG Wire	Camosun College

Table 1: Sources