CBUBioloid

Technical Documentation

Table of Contents

1. Introduction
2. Wiring Diagram
3. Hardware Documentation
4. Python Flow Chart
5. Python Documentation
   1. Class Diagram
   2. Dependencies
   3. Classes
      1. Methods and Functions
   4. Libraries to Learn
   5. Notes about Pypot
6. File Structure and Important Paths
7. Phone Tethering
8. Checking Connected Networks
9. Checking and Changing Credentials
10. Testing Audio Out
11. Testing Audio In
12. Known Issues
13. Fatal Failures
14. Restoring the TJBot
15. References

**Introduction**

This Document contains detailed information on the CBU Admissions TJBots. With this document you should be able to troubleshoot most every error that will occur with one of the TJBots and get it back to working condition. It is noted that it would be recommended to have a basic knowledge of programming, and electronics before you continue with this guide.

**Wiring Diagram**

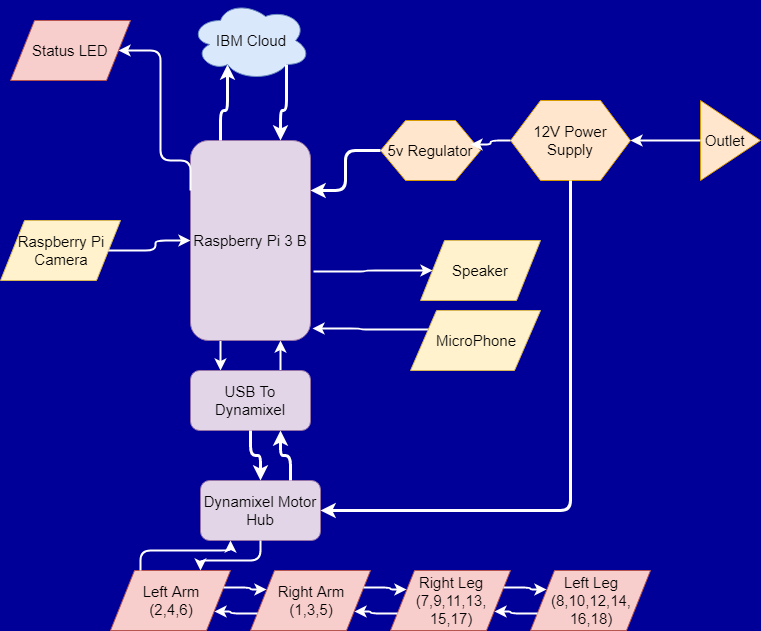
If any part of the TJBot may come disconnected during use please refer to this wiring diagram. The colors should coordinate to the colors of wires connected to the TJBot.

**<Insert Picture of TJBot Wirign Diagram here>**

**Hardware Documentation**

The Bioloid is made of a few main parts. The Raspberry Pi 3, USB Microphone. Speaker, LED, Button, Motor Power hub, USB2AX adapter, PiCamera, and lastly the body of the Bioloid and is 16 motors. The Raspberry Pi 3 is the brain of the Bioloid. All of the other components connect to the PI. The Microphone connects to one of the USB ports on the Raspberry pi and acts as an Input device. The Speaker connects to the 3.5mm audio port on the Pi but receives power from the USB Port. Like the name implies, the USB2AX adapter connects to the USB port of the pi. The Camera connects directly to the camera port of the Raspberry pi. The LED connects to the GPIO pins on the Raspberry pi to power and control it. The Motors all connect to the motor hub which is connect to the USB2AX adapter to allow commination between the motors and the Pi.

A simplified connection diagram can be seen below.

****

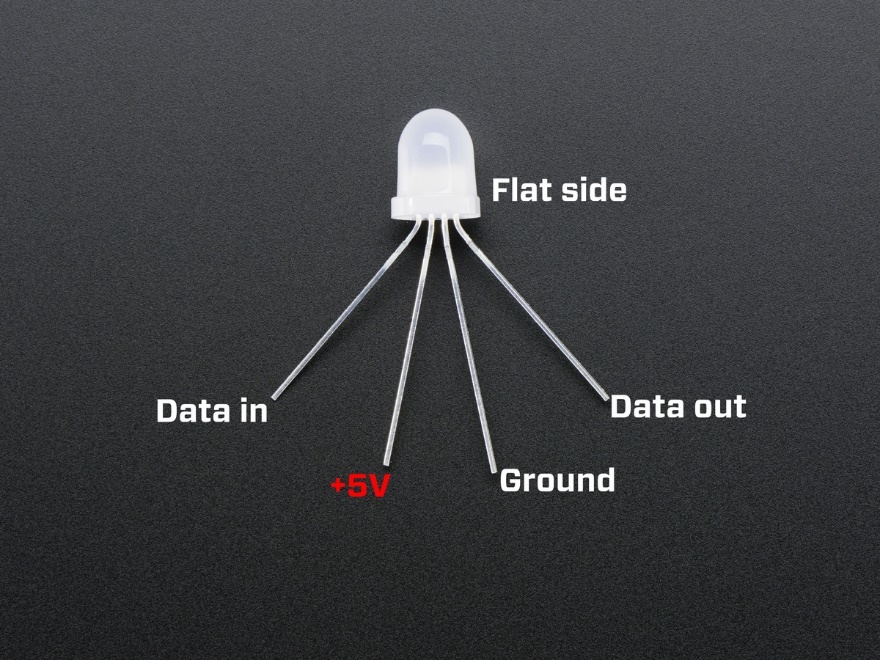
**The Raspberry Pi**

The Raspberry Pi is the brain of the operation. Every thing connects to it and it connects to the IBM Cloud. It has a several ports on it that can be used.

1. **Power Input**
   1. The Raspberry Pi 3 Is powered from a Micro USB port on the side of the head. It is recommended to run the Raspberry Pi off a **5v 2A** power source to sufficiently power it and avoid damaging the device.
   2. If you need a battery powered TJBot you can use a phone charging battery bank with a Micro USB cable to power the TJBot. Just ensure that it gives off enough power.
2. **HDMI** 
   1. There is an HDMI port on the Raspberry Pi 3 aswell. When a monitor is connected to this port the Pi will boot up and display its Images on this screen. This can be helpful if you need to sit down and ensure credentials are correct. Or if you need to check and see if the PI is on the right network.
3. **Ethernet**
   1. In the Event that the WIFI is down or you need a steady connection to the PI, you can connect and Ethernet cable to this port and to your local network jack to give the PI access to the internet.
4. **Audio**
   1. The Speaker connects to this standard 3.5mm input/output device. It is possible to configure the Pi to use this port as a Component out for audio and video.
5. **USB**
   1. There are 4 USB ports on the underside of the Pi. One is used to use the microphone. The remining 3 are un-used and can be connected to if you need to Tether a phone for data usage.
6. **SD Card**
   1. At the top of the PI is a slot for a Micro SD Card. This SD card has the operating system and programs for the PI to operate. Do not damage or lose this. Avoid removing it unless needed.
7. **GPIO Pins**
   1. The pins running along the side of the Pi are the GPIO pins, these have the LED, Servo, and other components connected to them. These allow the Pi to access other devices. When the PI is on do not unplug or plug anything into these pins to avoid damaging the Pi.
8. **Camera** **Port**
   1. There is a slot for a ribbon cable to connect to labeled Camera. In the event you wish to connect a Raspberry Pi Camera you connect it to this port. This is the recommended way to connect a camera. These can be accessed with Python simply using the piCamera library. The Bioloid has a camera connected to it for visual recognition here
9. **Display Port**
   1. Like the Camera port there is a ribbon for a display port. I have never bothered to use this.

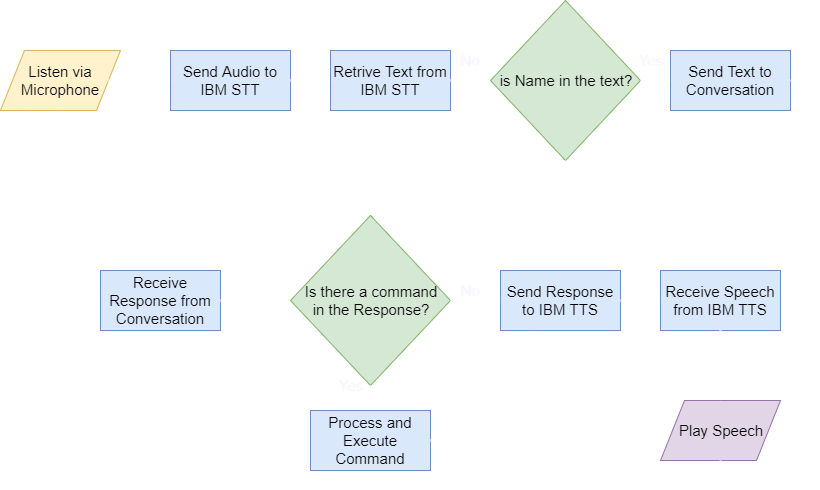
**LED**

The Led that acts as the status indicator for the software sits on top of the RPI and connects to the GPIO pins. This LED is a Adafruit Neo Pixel LED. It is different from other RGB LEDS as it reads data to check and see if it needs to change. Not PWM. The PI communicated with the LED via the Neopixel Library for Python. The pinout of the LED is denoted by a flat edge on the LED as seen below.



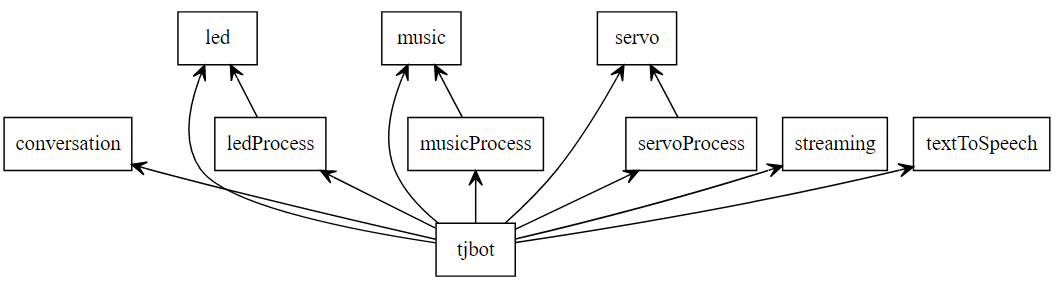
**FIGURE 1**

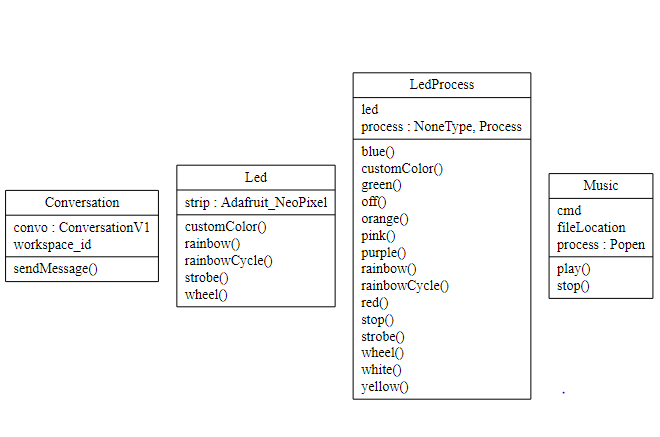
**FIGURE 2**

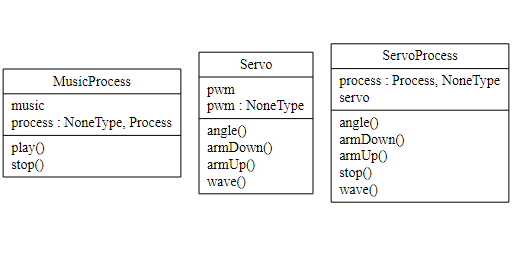
**Python Flow Chart**

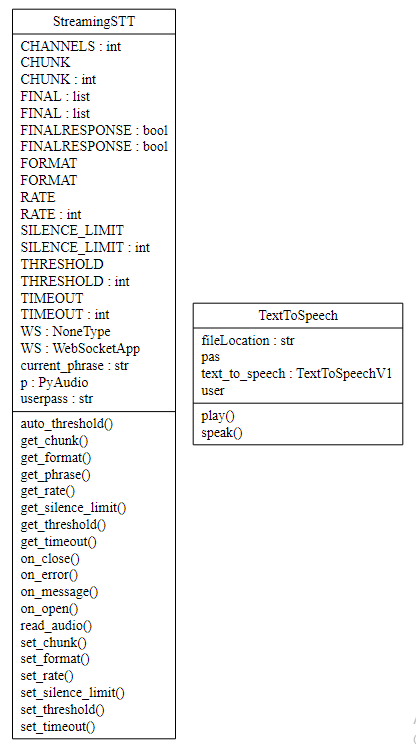
**Python Documentation**

**Class Diagram**









**Dependencies**

The python code for the Bioloid requires a lot of dependencies and setup prior to use. As there are so many a script called **setup.sh** has been included to help ease the install process. This script should be run with the command **“sudo ./setup.sh”** to ensure the files are properly installed. The script is as follows and contains the dependencies needed to run most of the code.

The Bioloid also depends on the pypot library, piCamera, and more. They are listed below the setup script. Pypot requires many of its own dependencies.

|  |
| --- |
| echo Update apt-get lists and upgrade existing |
|  | sudo apt-get -y update |
|  | sudo apt-get -y upgrade |
|  | sudo apt-get -y dist-upgrade |
|  |  |
|  | echo Install vim and screen |
|  | sudo apt-get -y install vim screen |
|  |  |
|  |  |
|  | echo Installing Port Audio Stuff. |
|  | sudo apt-get -y install portaudio19-dev |
|  | sudo apt-get -y install python-all-dev |
|  | sudo apt-get -y install python3-dev |
|  | sudo apt-get -y install python3-pyaudio python-pyaudio |
|  |  |
|  | echo Installing stuff for the watson cloud connectors. |
|  | echo This one takes a while and seems like it stalls, but just let it run! |
|  | read -n 1 -p "Press any key to continue.... BUT REMEMBER IT MAY TAKE A WHILE TO RUN!" mainmenuinput |
|  | sudo apt-get -y install libssl-dev libffi-dev build-essential scons swig |
|  | sudo easy\_install3 --upgrade watson-developer-cloud |
|  |  |
|  |  |
|  | echo Installing the RPi GPIO stuff |
|  | sudo pip3 install RPi.GPIO |
|  |  |
|  |  |
|  | echo Copying the confil files for asound |
|  | cp asound.conf /etc/asound.conf |
|  | cp .asoundrc ~/.asoundrc |
|  |  |
|  |  |
|  | echo Install VLC |
|  | sudo apt-get -y install vlc |
|  |  |
|  | echo Install the required Python3 Libraries |
|  | sudo easy\_install3 websocket-client |
|  | sudo pip3 install colour |
|  | sudo pip3 install python-vlc |
|  |  |
|  | echo install other stuff |
|  | sudo pip install requests |
|  | sudo pip install codecs |
|  | sudo pip install configparser |
|  |  |
|  |  |
|  | echo Install the Adafruit Led software |
|  | cd ~/ |
|  | git clone https://github.com/jgarff/rpi\_ws281x.git |
|  | cd rpi\_ws281x |
|  | scons |
|  | cd python |
|  | sudo python3 setup.py install |
|  |  |
|  |  |
|  | cd ~/ |
|  | git clone http://people.csail.mit.edu/hubert/git/pyaudio.git |
|  | cd pyaudio |
|  | sudo python3 setup.py install |

**Classes Methods and Functions**

The following are the classes along with their methods

<Use Pydoc ot Auto generate this>

**Libraries to Know**

**Notes about Pypot**

Pypot is the library used to control the motors using the USB2AX adapter. It communicates via UART. It is highly recommended to read the documentation for Pypot before attempting to work with motor movements. Though below some of the more important things to know will be listed.

**A modified version of Pypot is being used-** The orginal implementation of Pypot has the program stop when an error occurs and sends it to an error handler. However, the most common error when sending data to the motors was an overflow error that was suppressed. The modified version of this file will be included in the GitHub repository. Make sure you used this else the motors may lock up and stop functioning until a reboot.

**Wait to stop for primitives-** This is mentioned in the Pypot documentation and I will mention it here too as some of the primitives that have been built for the Bioloid have the option to not wait and stop. The wait to stop option freezes the python code until a movment has been completed. If the option is set to be false the robot will start to move and then begin the next segment of code. This works well until another movement is started. If this occurs the primitive threads go out of sync with one another and the main thread and cause the motors to freeze in what ever position they are in and the program to lock up. In some instances the program will recover but the motors will be in odd positions, and many other instances the program will crash.

So make sure your program has finished one set of movements before starting another. Though wait to stop is useful for moving motors while the robot talks, make sure it finishes its movements until the idle position is called again.

**Troubleshooting and Definitions**

The following sections are mostly informational and trouble shooting based. Pleaselet it be known that when “connecting to the Raspberry PI” is mentioned, and not specified to use a monitor, this can be done either from using SSH to connect to the Pi, or by connecting a Monitor, Mouse and Keyboard, to the Pi and opening a terminal session.

**File Structure and Important Paths**

The location of most of the programs can be found in

/home/pi/SeniorProjectBioloid/

**Phone Tethering**

In the event that you are taking the TJBot off campus. To perhaps demonstrate it at another school and you are unable to get internet access there it is possible to tether your phones data connection with the TJBot. However, your phone must support tethering.

First connect your phone to one of the Bioloid USB ports and tell your phone to allow tethering. This option will be located in your phone settings.

Next power up the Bioloid. Your phone may indicate that it is now sharing its data with another device or something of the sort.

Now the Bioloid should be able to access the Internet through your phone and function properly. It is recommended that you attempt this prior to a demonstration to ensure it will work.

In some cases you may have to have the TJBot powered before tethering is started. Just power up the TJBot then enable tethering from your phone options.

**Checking and Adding Connected Networks**

To check and see if the Bioloid is setup to connect to a network you must connect to the Pi.

When connected to the pi navigate to **/etc/wpa\_supplicant/** and open the file **wpa\_cupplicant.conf**

This file contains the networks that the PI has the credentials to connect to.

Add the network that you wish to have the Pi connect to in this file to allow connections to it.

More information can be found at

<https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md>

You can connect to the Pi using a monitor and once booted in the top right hand corner you click on the Wireless symbol. You can connect to the network of your choice this way.

**NOTE: T**here is a chance that in the future ITS may change the wireless network and its credentials. If this occurs the Bioloid will no longer be able to connect to the WiFi automatically. You will have to reconnect to the network using the above steps, and register the MAC address which can be found by connecting to the Bioloid and using **ifconfig**

**Checking and Changing Credentials**

In the event that the Credentials for the IBM cloud services seem to stop working you have to do a few things.

First open the config.cfg file in tj-python-master. Check and see if the Credentials have been placed in there.

Next loginto bluemix.net with the TJBot email address and password.

From here you can go to the dashboard which will show all of the created services for the Account.

Click on each of the services. There should be a services for Text to Speech, Speech to Text, and Conversation.

When you open this up there should be an option on the left side of the screen to look at service credentials. Click on this.

You should then have some view to click on view credentials or what ever and then you can click on this and make sure the UserID and Password for the resprective service is correct in the config file.

If forsome reason it is correct but this service is failing, Try creating new credentials and seeing if that works. If it does not work delete the service. Make a new one and create new credentials for that service and enter them in.

That should fix any issues you have with connecting to the IBM cloud services unless there is no internet avaible for the TJBot to connect to.

**Testing Audio Out**

If no audio is coming out of the speaker and it is turned on, charged, and the volume is turned up follow these steps to determine what is wrong.

1. Test the speaker on another device to see if it is works there
2. If the speaker does not work try another set of speakers on the Pi see if they work
3. If the new speakers work on the Pi then get a new speaker for the Pi
4. If the Speaker works on another device then check the default audio device.
5. This can be found by booting up the Pi, and connecting a monitor to it.
   1. Once booted in the top right corner is a speaker symbol. Right click on it to view the default output device. It should be set to the analog output. If it is not change it and reboot the Pi.

**Testing Audio In**

Microphone is not working. So it seems that you need to test the audio input.

1. First check and see if the device si recognized using **arecord -l** This will list all of the devices connected to the pi that are capable of recording audio
2. If you see the microphone is there then lets try recording audio. Go ahead and use **arecord output.wav** this will go on until you terminate the capture sequence with ctrl+c.
3. Once this file has been made use aplay to play the newly created file.
4. If there is no audio then run **alsamixer** to check and see if the audio capture volume is turned up and try again.
5. If audio is being recorded, there is a chance that the streaming.py software is not working properly. Go ahead and run the TJBot code from terminal and see if there are any errors from the TJBot.

**Known Issues and Causes**

**Input over flow** – The fact the RPIs seems to require different audio drivers and firmware on each board. This causes issues when trying to record audio as there is no one catch for all problems. Changing the microphone can cause this error to occur.

This error seems to occur no matter what changes and can often times occur when the program is running. This can cause a fatal crash, but other times it simply recovers from this and freezes for a few seconds. Currently streaming.py should supporess this error and keep doing its work.

**Segmentation Fault** – There is an issue that as of writing has not been fixed. It seems like it began to occur after, fixing the TSSLV Error below. Where there is a segmentation fault that occurs in streaming.py. When the error occurs is not truly known. Though a stack trace seems to follow it to malloc.c which is used from libcrypto0.1.1 and research shows it may be an error in OpenSSL. There is another idea gained that it could be the fact we are connecting to websockets in a multi threaded program and Open SSL has troubles with this. One recommendation was to force the pi to use only one version of SSL. To get a back trace of the stack launch the program using gdb, and then run Main.py and wait for a segmentation fault.

Signal handlers have been added to attempt a reboot when this error occurs, but thus far has proved fruitless and a full reboot is required. Most attempts at signaling this error has occurred have failed. Other than printing to the screen.

**HDMI-** If you want a PI to be connected to a monitor you must have the monitor connected to the Pi before boot. Else the Pi will try and connect to another device.

**TSSLV Error**- There was an error that when the TJBot tried to record audio and send it off to the IBM cloud the WebSocket kept closing. This was due to the TSSL version that was used in Streaming.py was dropped by IBM. If you get this error check IBM release notes and the TSSL version in streaming.py to make sure it is no longer outdated. As of writing streaming.py uses TSSLV1.2 and IBM no longer supports TSSLV1 or TSSLV1.1 and this error should not occur

**No Default Input/Output device** – This error occurs if one of the two audio devices are not connected. Or it will occur if the program was killed unexpectedly and there is now a zombie process of streaming.py running in the background. Try killing all python processes and starting again. Else you may need to replace some devices.

**Not connecting to wifi on boot**- There is an error where the Pi does not connect to a wireless network on boot. You should be able to connect to the network via the GUI, or you can use connmanctl to connect from terminal. Currently there is not fix that has been found other than writing a script to force connection on boot. This should be fixed soon then as it severly impacts the robot.

**Dynamixel Motors out of Sync-** See **Wait to stop for Primitives** in **Pypot Notes** You have to unpowered the whole robot and wait 10 seconds before powering it back on.

**Request Timed out can not reach Motors [x , y, z]** – This error occurs when Pypot cannot communicate with the motors listed. It often times occurs after an out of sync error. If this occurs You have to unpowered the whole robot and wait 10 seconds before powering it back on.

**Microphone Sensetivity -** Different microphones may not be sensitive enough or too sensitive. The Microphone for the Bioloid is more sensitive.

**Expired Credentials -**Every so often it seems that credentials for the IBM services may expire and need to be renewed. The cause of this is unknown. You just have to create a new set of credentials.

**Zombie Processes-** There is an issue where if the program ends forcefully it will leave zombie processes this is fixed in the start program killing all python scrips be fore it starts the main python code. Just let it be know if you are attempting to manually run the programs you need to make sure all of the python processes have been killed before starting else there will be conflicts.

**Fatal Failures**

Fatal failures are those that will commonly end with a piece of equipment being replaced or code may have to be rewritten or some thing has come disconnected internally, and it occurred unpredictably.

1. Raspberry Pi powers on but will not boot (Power status LED is red)
   1. If this issue occurs then, you know the TJBot is getting power, however it is unable to boot.
   2. The most commonly encountered cause of this is the hardware is physically damaged.
   3. First check and see if the SD card is still working. Take it out of the Pi and place it in another Pi and see if it works. If it does then your Pi is damaged and must be replaced.
   4. If the SD card does not work attempt to access it from a computer. If it is un recognized, you may have to replace the SD card. Please see **Restoring the TJBots**
2. Servo Motor will not move
   1. If the servo motor will not move first check the connections from the servo motor to the Pi.
   2. There is a chance these connections have come undone. Reconnect them if they have.
   3. If the motor is connected and still will not move. Or makes a grinding sound when it attempts to move, more than likely the gears in the servo have broken. If this occurs replace the Servo motor with a new one.
3. Status LED Stays Blue and does not Change to Orange
   1. If this occurs, the PI is getting power, and providing power to the LED though no software is able to access the LED
   2. There is a chance that the data line from the PI to the LED has come disconnected check and reconnect it if needed.
   3. If the data line is properly connected, then check and see if the PI is starting botButton.py on boot. To do this you will need to either connect a monitor to the PI or use SSH to communicate with it.
   4. Check by running botButton.py manually and see if it works.
   5. If it does check cron to make sure it is in there and working properly.
   6. You may also need to check the output of cron to see if there are any errors.
4. Status LED does not Turn on
   1. Check and make sure the LED is properly connected at both the LED and the Pi.
   2. There is a chance that the connection between the cables and the LED have come disconnected.
   3. If all of the cables are properly connected and you are unable to turn on the LED then find a replacement Neopixel LED.
5. Speaker has no status light and TJBot is plugged in
   1. Check and make sure the charging cables are connected to the Pi.
   2. The speaker will not have any lights on if it is not charging from the PI, and it is dead, or turned off. In the event that the Light is off when the TJBot is connected the speaker is no longer connected properly.

**Restoring the TJBot**

Alright, so the unthinkable has occurred. The SDCard is bad. You deleted some files you should not have, and the TJBot is dead and will no longer function. Lucky for you this is not the case.

All of the Files for the TJBot code can be found at <https://github.com/bmcginn1/SeniorProjectTJBot> Simply redownload the files you need and every thing is all set.

Or if you have really messed up and you do have to reinstall the whole entire SD card. You can contact a few different people to get a back-up image of the TJBot. We planned on this happening.

1. Get access to the CBUAdmissionsTJBot.img file
   1. If you cannot find it from any faulty make a issue request on the Github link above to get a copy. It is a large file.
2. Get a new SD card. 16GB Class 10 or better are required for this.
3. Downlaod Etcher from etcher.io and connect your new SD card
4. Follow the instructions for etcher to write the image to the SD card
5. Connect the SD card to TJBot
6. Run apt-get update and upgrade
7. Test and make sure the program is working properly
8. You are all done!

**References and Links**

**Github Repository :** <https://github.com/bmcginn1/SeniorProjectTJBot>

**NoBox tj-python:** <https://github.com/noboxio/tj-python>

**NeoPixel Documentation :**

**Arecord Help :** <http://xmodulo.com/how-to-capture-microphone-input-to-wav-format-file.html>

**Wireless Connection :** <https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md>

**Raspberry Pi Documentation:**

<https://www.raspberrypi.org/documentation/>

**NeoPixel Python Documentation:** [**https://github.com/jgarff/rpi\_ws281x**](https://github.com/jgarff/rpi_ws281x)

**Etcher.io:** <https://etcher.io/>