CBU Admissions TJBots

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
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 TJBots are made of a few main parts. The Raspberry Pi 3, USB Microphone. Speaker, LED, Button, and Servo. The Raspberry Pi 3 is the brain of the TJBot. 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 LED, Servo and button all connect to GPIO pins that are on the Raspberry PI. The Diagram Below shows where each of the pins connect.

**<Put Diagram here>**

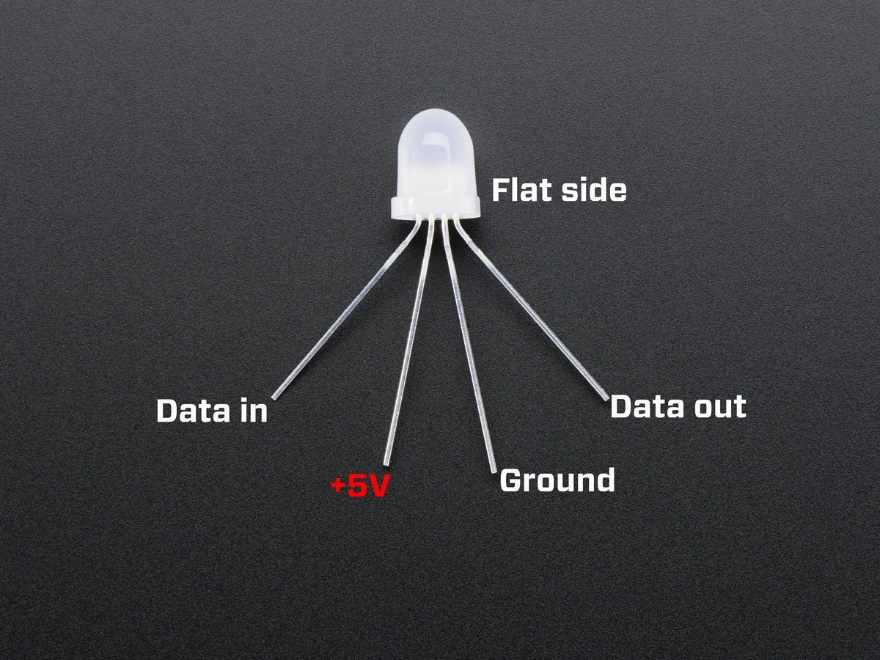
**The Raspberry Pi**

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.
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.



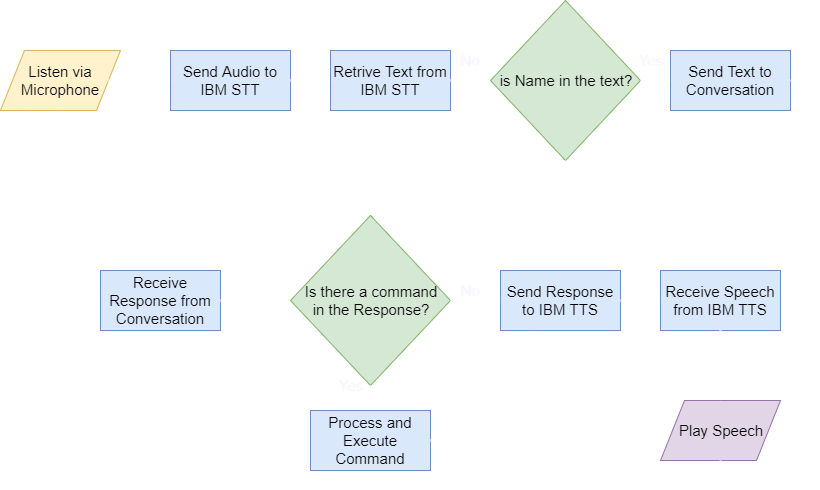
****

**FIGURE 1**

****

**FIGURE 2**

**Python Flow Chart**

****

**Python Documentation**

**Class Diagram**

**Dependencies**

The python code for the TJBots require 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 dependecnies needed to run.

|  |
| --- |
| 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>

**File Structure and Important Paths**

**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 TJBots USB ports and tell your phone to allow tethering. This option will be located in your phone settings.

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

Now the TJBot 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 TJBot is setup to connect to a network you can do it one of two ways. Via SSH or Via monitor.

Plug in monitor mouse and keyboard and check the things

SSH and go to that file

**Checking and Changing Credentials**

**Testing Audio Out**

Use APlAY and see if noise is made

Check Speaker connection and such

**Testing Audio In**

Test using Arecord.

**Known Issues**

Buffer overflow.

**Fatal Failures**

**Restoring the TJBot**

**References**