Tutorial based off of:

https://gideonwolfe.com/posts/security/p4wnp1/

case used:

<https://www.amazon.com/dp/B07BK2BR6C/ref=sspa_dk_detail_1?psc=1&pd_rd_i=B07BK2BR6C&pd_rd_w=L2iby&pf_rd_p=48d372c1-f7e1-4b8b-9d02-4bd86f5158c5&pd_rd_wg=aatuP&pf_rd_r=TFTNW363F3PNEZG0MZ10&pd_rd_r=ee91a8ef-704c-4d4a-884f-ce9368197958&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExRUwzOElPTVZXVUQmZW5jcnlwdGVkSWQ9QTA2NzU4MjQxVFNGS1lDRDczRFdFJmVuY3J5cHRlZEFkSWQ9QTA5MjA4NzQzNTBBTkJPSkxDM0VUJndpZGdldE5hbWU9c3BfZGV0YWlsJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==>

screen used: 1.3inch waveshare OLED  
  
pi used: Pi0WH

recommended sd card: <https://www.amazon.com/Sandisk-Ultra-Micro-UHS-I-Adapter/dp/B073K14CVB/ref=asc_df_B073K14CVB/?tag=hyprod-20&linkCode=df0&hvadid=309779531175&hvpos=&hvnetw=g&hvrand=508201239040490640&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9028142&hvtargid=pla-374265929034&psc=1&tag=&ref=&adgrpid=62412137260&hvpone=&hvptwo=&hvadid=309779531175&hvpos=&hvnetw=g&hvrand=508201239040490640&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9028142&hvtargid=pla-374265929034>

The first thing to do is download the pwnpi image from here

RoganDawes github pages under releases

https://github.com/RoganDawes/P4wnP1\_aloa/releases

The easiest way to flash it would be to using etcher

flash it onto a micro sd of your choosing after this slap it into the pi

on gnu/linux you can use dd to copy the image over

and connect to it with the usb over ethernet ip

followed by :8000

172.16.0.1:8000

Or you could power it and connect to it’s network over wifi and enter in the following after connecting (password = MaMe82-P4wnP1)

172.24.0.1:8000

Alternatively you can make it connect to a network by going to:

Web into pwnpi > wifi > working mode > change to client failover with ap or station > wifi client settings > entering the network information

After this lets proceed to connect the hardware together so that way we can get ready to setup the menu:

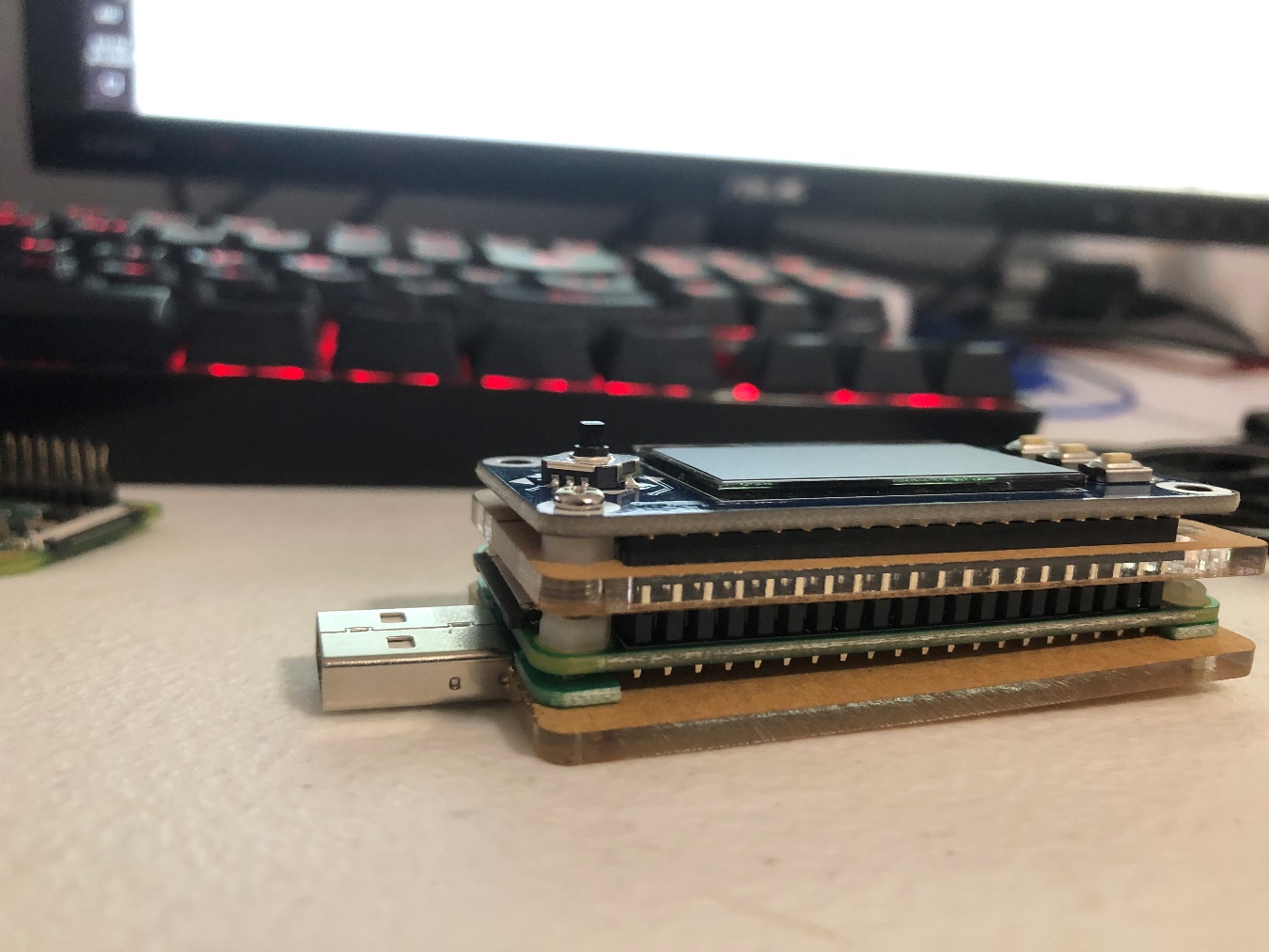
The first thing you should do if you’re using the same case I bought is connect the base acrylic that looks like the usb is going to mounted on and the RPi 0wh or w all together with the white screws that came with the OLED screen. Please mount the screws on the bottom right and top left completed by keeping down with the nuts that came with them.   
 (image has been flipped to mirror correct screw placement)



(here is a side view of the correct placement)  


After this get two metal screws that came with the case, mount the oled screen on the pi and put them through the bottom left and top right make sure to double up on the nuts here are some photos of what the final result should look like:





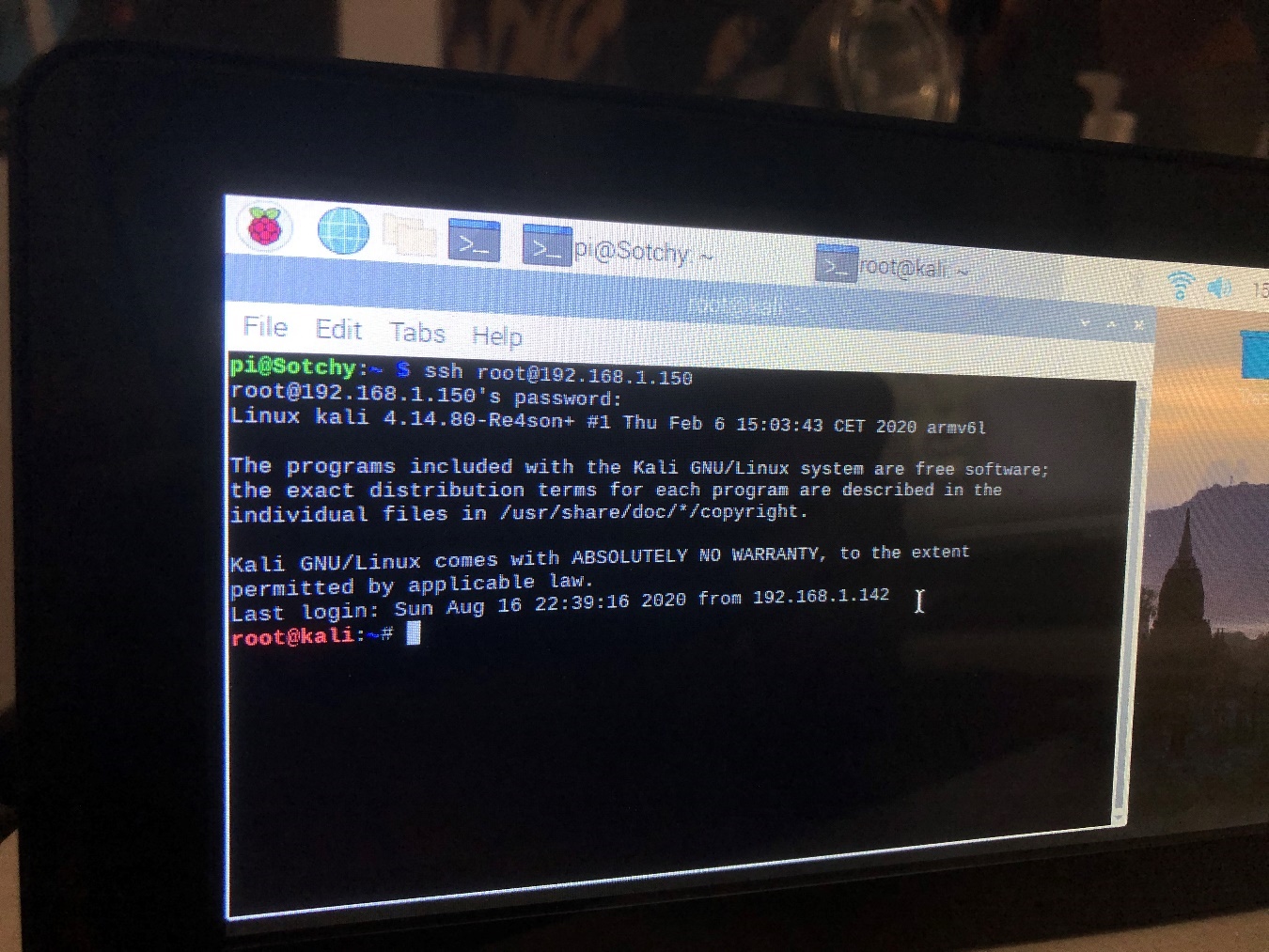
Now that this is done I will continue to follow the rest of gideons tutorial on configuring the OLED screen:

For the OLED screen we’re going to be configuring the menu using a repository made by Beboxos. But let’s start by sshing into the pi and editing config.txt in the boot directory

You can get into your pi a variety of way.  
 1. Connect it to your network and ssh into the IP assigned to it via the dhcp server

2. connect to it’s network and ssh into it

3. hook it up to your GNU/Linux PC and ssh into it

The way I’m doing assumes you connected it to your network and ssh’d into from your already existing gnu/linux machine in this case I’m going to be using my Raspberry Pi 3B+ with Raspberry Pi OS to get into it. The password to get into root if your classical root toor type scenario.

first let’s see if git is installed:  
 which git

(output should be /usr/bin/git if it’s installed)

now we need to clone beboxos repository:

git clone <https://github.com/beboxos/P4wnP1_ALOA_OLED_MENU_V2.git>

The script itself is a little outdated and will need a fair bit of tweaking to install properly make sure it looks like the following if you aren’t a time traveler who can go back to when python 2.7 was supported(before Jan 1st 2020)  
  
Please run apt-get update -y && apt-get upgrade -y before running this

reboot

(script = install.sh)  
#!/bin/sh

echo "Install Luma.core drivers"

apt-get install python3-dev python3-pip libfreetype6-dev libjpeg-dev build-essential libopenjp2-7 libtiff5

pip3 install luma.oled

pip3 install luma.core

echo "Create directory"

mkdir /root/BeBoXGui/

echo "Copying files"

cp \*.py /root/BeBoXGui/

mkdir /root/BeBoxGui/images

cd images

cp \* /root/BeBoXGui/images/

echo "Copying run script in local P4wnP1 script"

cp /root/P4wnP1\_ALOA\_OLED\_MENU\_V2/scripts/runmenu.sh /usr/local/P4wnP1/scripts/

echo "All files are ready"

echo "to run with P4wnP1 boot"

echo "Go thru web interface"

echo "Go in trigger section"

echo "Create new trigger"

echo "on service start :"

echo "run script sh and choose "

echo "runmenu.sh"

echo "Enjoy"

echo "by default gui.py use SPI interface"

echo "if you use I2C oled edit gui.py"

echo "and set I2C\_USER = 1"

after going through and making sure yours matches up to it let’s run it and then move onto actually making the menu run at boot

cd P4wnP1\_ + tab complete ( we need to be in the pwnpi directory to ensure the \*.py files are moved correctly )

./install.sh

navigate to the pwnpi scripts and chmod the runmenu script   
  
After all of this change the python call in runmenu.sh to python3 then go to runmenu.py located in the BeBoXos Gui folder in your home directory and change those python calls to 3 as well

After this we need to install a very old version of raspi-config to enable spi so this thing can finally work

Since you already did an update and upgrade the next thing to run is:

apt-get install lua5.1 alsa-utils triggerhappy curl libcurl4 -y

wget <http://archive.raspberrypi.org/debian/pool/main/r/raspi-config/raspi-config_20160322_all.deb>

dpkg -i raspi-config\_20160322\_all.deb(-i means install)(you can see this by doing man dpkg)

from here as root we run:

raspi-config > advanced options > SPI > yes > ok > yes > Do the same for i2c

reboot

ssh back in

nano /boot/config.txt

add in on a new line:   
dtparam=i2c1=on

reboot

ssh back in

If it installed correctly you can web into your pwnpi and go to > trigger actions > add one > make sure enabled is on > trigger > service started > action > run a bash script > select runmenu.sh > update

Store this triggeraction as GUI

Generic settings > load stored > startup > change trigger actions to GUI > store as GuiStartUp > Startup Settings > select GuiStartUp

reboot from ssh session

The result should look like this:

