

PSP Consolizer Build Guide

Built by Stephen Cooper

<https://github.com/ste2425/PSP-Consolizer>



This is a personal project *NOT* a commercial product. As such do your own due diligence to be sure this will not damage your PSP or Arduino.

Please don't just take my word for it that it actually. Check everything yourself before spending any money.

Building a PSP Consolizer is split into three steps

- 1.1 Having the PCB manufactured
 - 1.2 Having the PCB Assembled
 - 1.3 PCB Parts Layout
- 2 Assembling the PCB
- 3 Programming the Arduino
- 4 Connecting the Arduino and PSP
 - 4.1 Soldering the Analog Stick

The last step (Arduino programming) can be done at any point, before or after it is installed.

The files included within this release package contain everything you need to complete this project. You do not need any extra software to program the Arduino. Everything is bundled within the installer (thats why its quite large!).

⚠ Important There is fine soldering involved with the PSP-2000 which is non reversible. This project currently does not support the PSP-3000. Please review the entire process to see if your up to the challenge before spending any money.

1.1 - Having the PCB manufactured

Board configuration

I chose [JLCPCB](#) to manufacture my boards as they were recommended to me. I'm pleased with that have done so far.

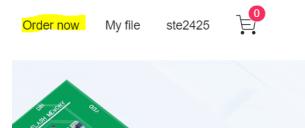
I also have JLCPCB fit the difficult to solder surface mounted parts so only the through hole parts remain.

i Files have been generated for JLCPCB to manufacture and fit the surface mount parts, however any manufacturer can be used if desired.

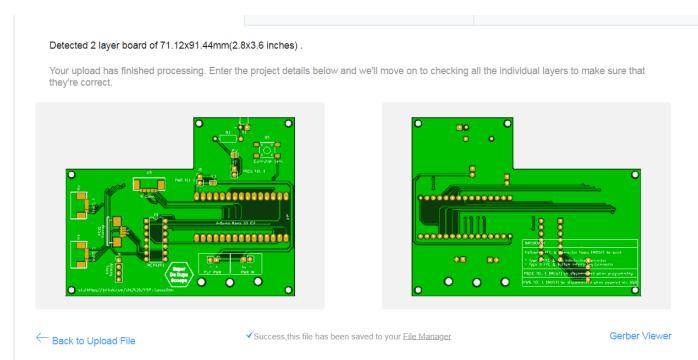
1. Firstly [create an account then Log in](#).

2. On the home page begin the process by clicking the **Order Now** link in the top right.

Click the **Add Gerber** link and select **PSP-Consolizer-v1.0.zip** from the downloaded release artifacts.



Once it loads you should see a preview of the board like below.



3. Next scroll down and ensure the various settings match below

Note about certain Settings Changing some settings such as colours may increase the predicted cost. JLCPCB works by batching orders onto single boards. If you choose an unpopular colour it will take longer for enough orders in that colour to fill a board resulting in a greater cost.

name	value
Base Material	FR-4
Layers	2
Dimensions	91.44mm * 71.12
PCB Qty	5
Product Type	Industrial/Consumer electronics
Different Design	1
Delivery Format	Single PCB
PCB Thickness	1.6
PCB Color	Pick which you like
Silkscreen	Will change depending on PCB colour
Outer Copper Weight	1 oz
Surface Finish	HASL (with lead)
Gold Fingers	No
Confirm Production File	No
Flying Probe Test	Fully Test
Castellated Holes	No
Remove Order Number	No

4. Enable PCB Assembly by clicking the toggle.



1.2 - Having the PCB Assembled

1. Select **Assemble top side** and verify the settings match below.

name	value
PCBA Type	Economic
Assembly Side	Top Side
PCBA Qty	5
Tooling holes	Added by JLCPCB
Confirm Parts Placement	No

2. Click **Confirm**.

3. Next you need to upload the BOM and CPL files.

These tell JLCPCB the parts to assemble (Bill Of Materials) and where to place those parts (Component Placement List)

For the BOM select **psp-bluetooth_bom_jlc.csv** downloaded from the release artifacts.

For the CPL select **psp-bluetooth_cpl_jlc.csv**

4. Select from the dropdown what the board is to be used for, i just selected **DIY** for both options.

5. Click **next** and you will be presented with a page listing the parts that will be used. If it looks like below click **Next**.

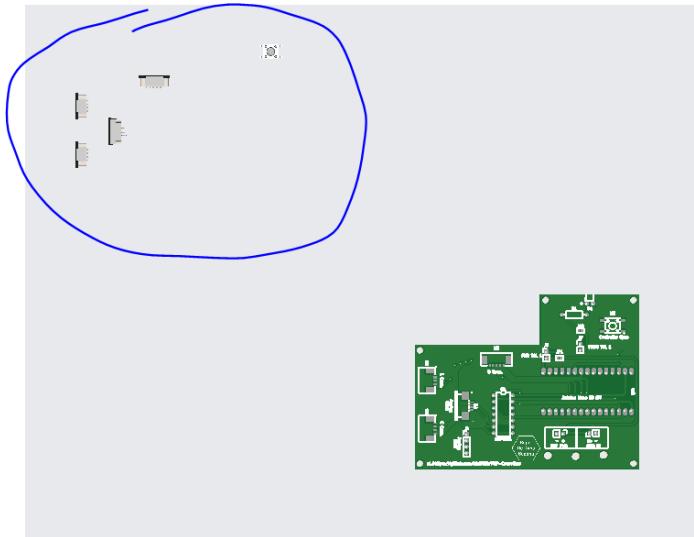
Total 4 parts detected 4 Parts confirmed 0 parts not selected

Uploaded BOM Data			Review Matched Parts						
Top Designator	Comment	Footprint	Matched Part Detail		Qty	Source	Lib Type	Total Cost	Select <input checked="" type="checkbox"/>
P1	FPC_4P_Pull_typ...	FPC-SMD_AFA07-S...	AFA07-S04FC-00 Slide Lock 4 Bottom Contact Su...	C13750	<input type="button" value="Q"/>	5	JLCPCB	Extended	£0.3828 <input checked="" type="checkbox"/>
U2,U4	AFC07-S10ECA-00	FPC-SMD_P0.50-1...	AFC07-S10ECA-00 Slide Lock 10 Contact,Top Surf...	C262583	<input type="button" value="Q"/>	10	JLCPCB	Extended	£0.9126 <input checked="" type="checkbox"/>
U3	AFC07-S14ECA-00	FPC-SMD_P0.50-1...	AFC07-S14ECA-00 Slide Lock 14 Contact,Top Surf...	C262587	<input type="button" value="Q"/>	5	JLCPCB	Extended	£0.4378 <input checked="" type="checkbox"/>
U5	TS-1102SD-A-F-R	SW-SMD_4P-L6.0...	TS-1102SD-A-F-R SMD Tactile Switches ROHS	C3010209	<input type="button" value="Q"/>	5	JLCPCB	Extended	£0.1637 <input checked="" type="checkbox"/>

1.3 - PCB parts layout

Now you will be given a preview of the board layout with the parts on it.

I don't know why but for me when it first loads the parts to assemble will appear in the wrong place.

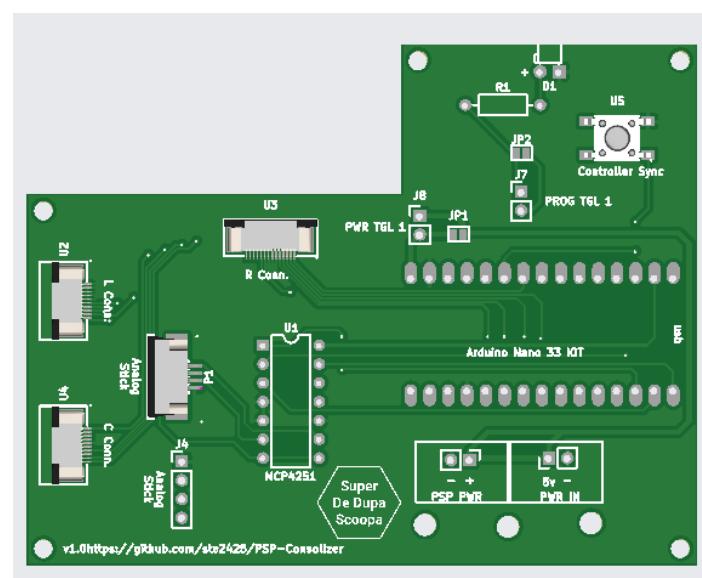


To fix this click the **Align** button in the tool bar.



It is possible to zoom in the preview using the middle mouse wheel and move about by holding the right mouse button

This should result in a preview which looks like this.



Once done click **Save to Cart** from there you can go through the purchasing process.



2 - Assembling the PCB

Next is the job of assembling the board. The following components will be needed.

- Arduino Nano 33 iot
- 2x 10pin 0.5mm pitch FFC cables Type A (contacts on both sides)
- 1x 14pin 0.5mm pitch FFC cable Type A (contacts on both sides)
- MCP 4251 digital potentiometer 14 pin DIP package
- LED colour of choice (See LED below)
 - Suitable resistor for LED
- Mounting hardware (see Mounting Arduino & MCP4251 below)
- Jumpers and pins (see Jumpers below)

Mounting Arduino & MCP4251

The Arduino and MCP4251 can be mounted permanently or be removable. The MCP4251 can be soldered directly to the PCB or using a DIP socket.

The Arduino can be mounted directly to the PCB using its castellated pins or via Male/Female header pins. The first option gives it the lowest profile but header pins allows you to use the Arduino for another project in the future.

Jumpers

The Arduino is powered via its **vin** pin. I've heard conflicting reports that powering it via this pin and USB at the same time (when programming) will cause damage. Others also say if you only USB it will back feed the **vin** pin, in our case powering up the PSP.

As a result there is a jumper **PWR TGL 1** and solder jumper **JP1**. If you know what your doing, unlike me, you can permanently solder **JP1**. You can also use a jumper pin instead (like me) and simply remove the jumper when programming the Arduino, disconnecting the **vin** pin from the Arduino.

LED

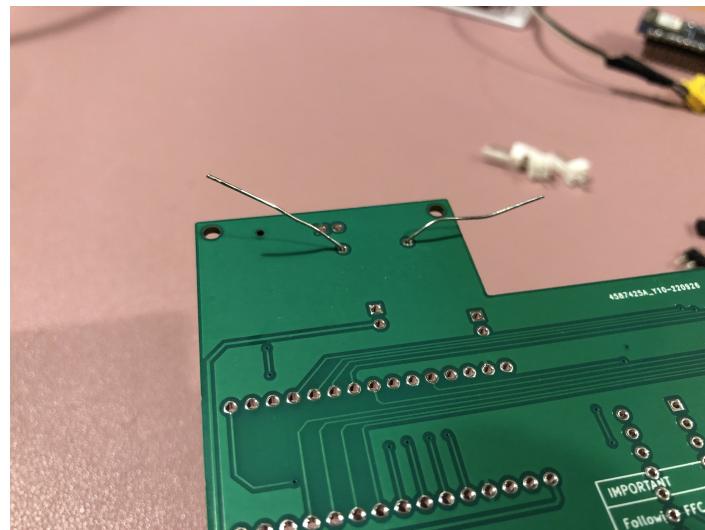
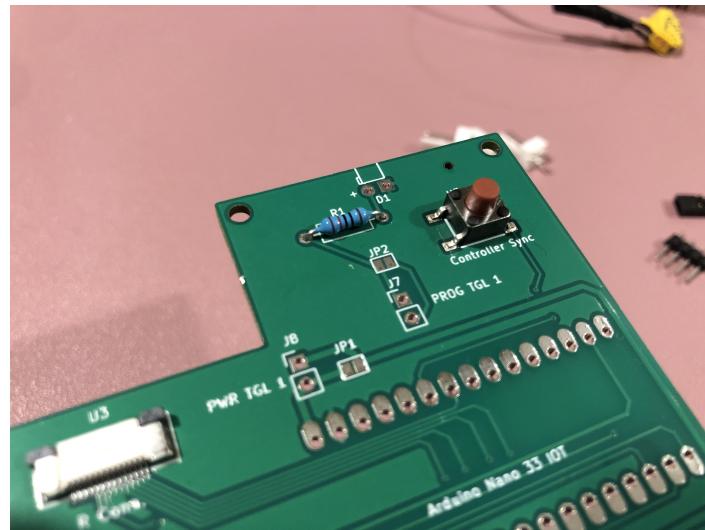
The LED is powered directly from the Arduino, using one of its serial pins. As a result choose an LED and resistor which do not overpower the Arduino pin.

As it's powered from a Serial pin I've read conflicting reports that having a load on those pins can cause programming errors.

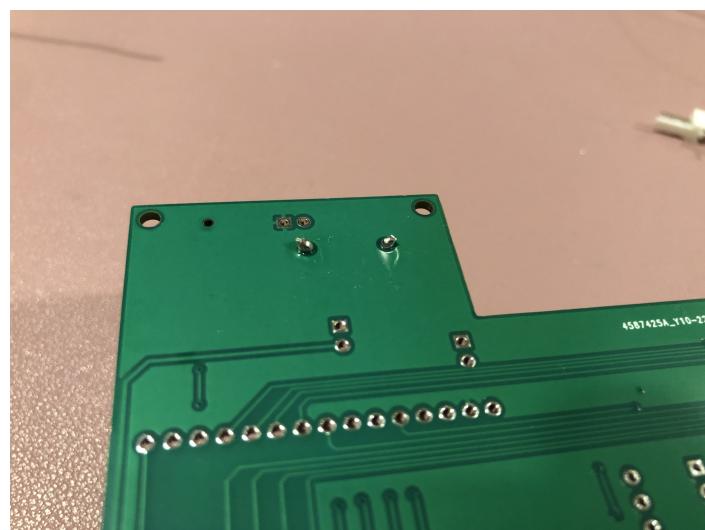
As a result there is a jumper **PROG TGL 1** and a solder jumper **JP2**. If you are confident it will not cause issues (it didn't in my case) you can permanently solder it. Otherwise, like the USB/**vin** comment you can remove the jumper when programming.

2.1 - Fitting the LED

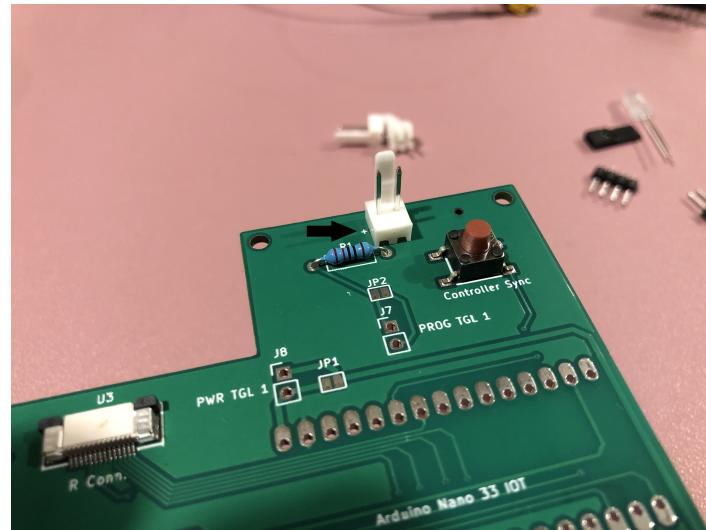
Begin by fitting the resistor so it is flush with the PCB and bend its legs outwards to hold it in place.



Solder it in place and snip off the excess legs.



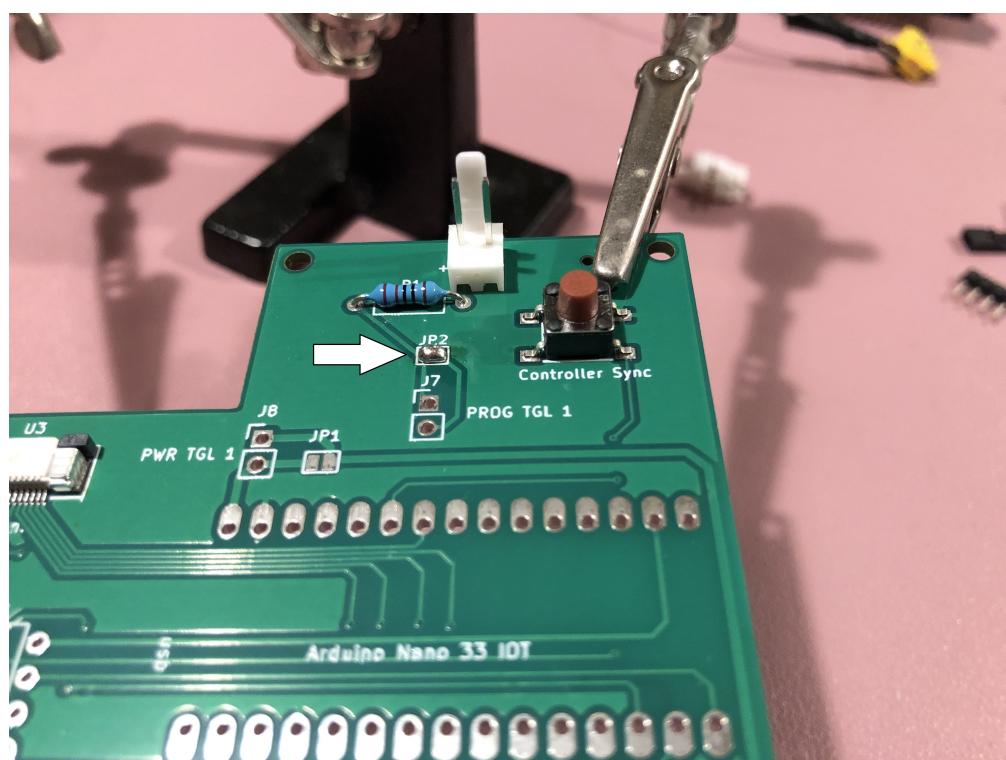
Next is the LED itself. Solder it directly following the same approach as the resistor or use a connector like I did.



Take note of polarity LED's are polarised. Take note of the + marked on the PCB.

Lastly decide if you wish to fit a removable jumper or use the solder bridge.

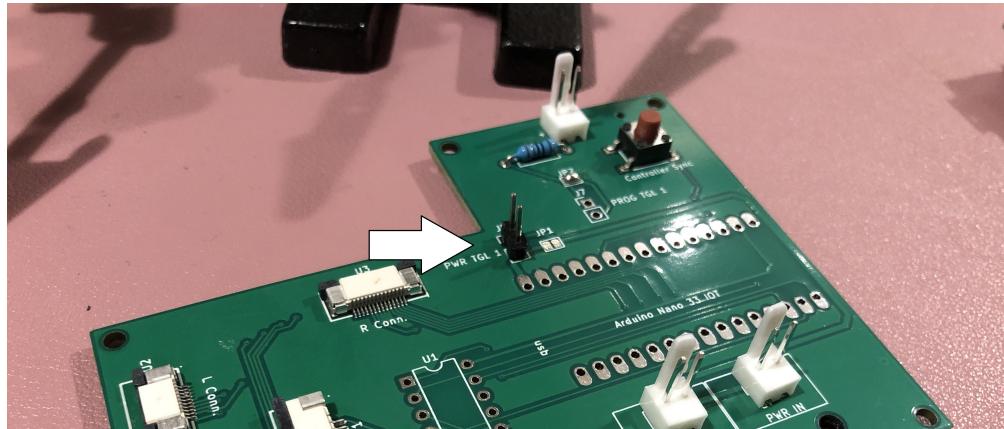
Below you can see the solder bridge being used.



2.2 - Powering the Arduino

As with the LED a jumper or solder bridge can be used. Now is the time to decide which.

Below you can see the jumper being used.



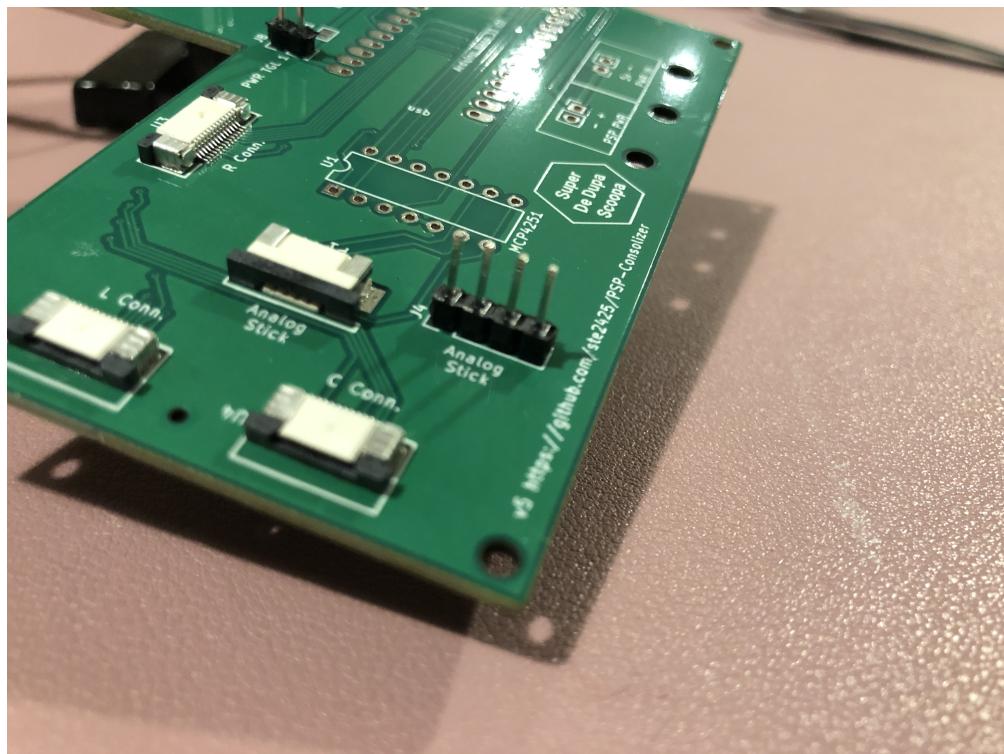
2.3 - PSP Analog Stick

The PSP's analog stick is connected to the PCB via header pins. Wires can be directly soldered if desired but that would make assembly difficult.

I chose to use header pins.

i Soldering to the PSP

Soldering to the PSP's analog stick points will be described later in this guide.



2.4 - Providing power

It is expected you will implement your own power solution. The PCB provides pin headers for Power in and Power out to the PSP.

If you choose to solder cables directly to the PCB there are mounting holes you can route cable ties through to help secure those cables.

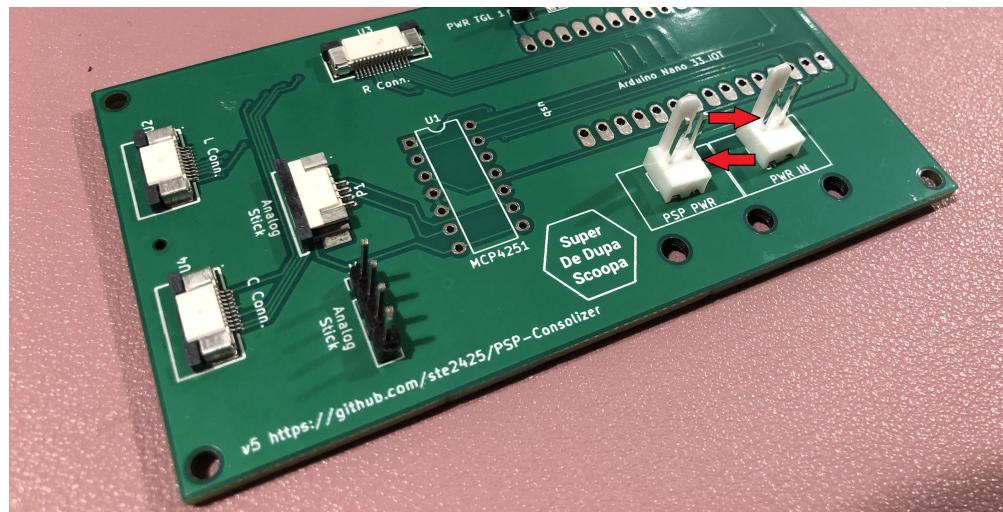
⚠ There is no power regulation

The PCB does not regulate the power input. You are expected to provide a **stable 5v**. Any more or less will damage the PSP and the Arduino.

⚠ Take note of polarity

Positive pins are both on the inside with negative on the outside.

I chose to use the same PCB connectors as the LED. If choosing connectors be sure they can handle the load the PSP will put on them.



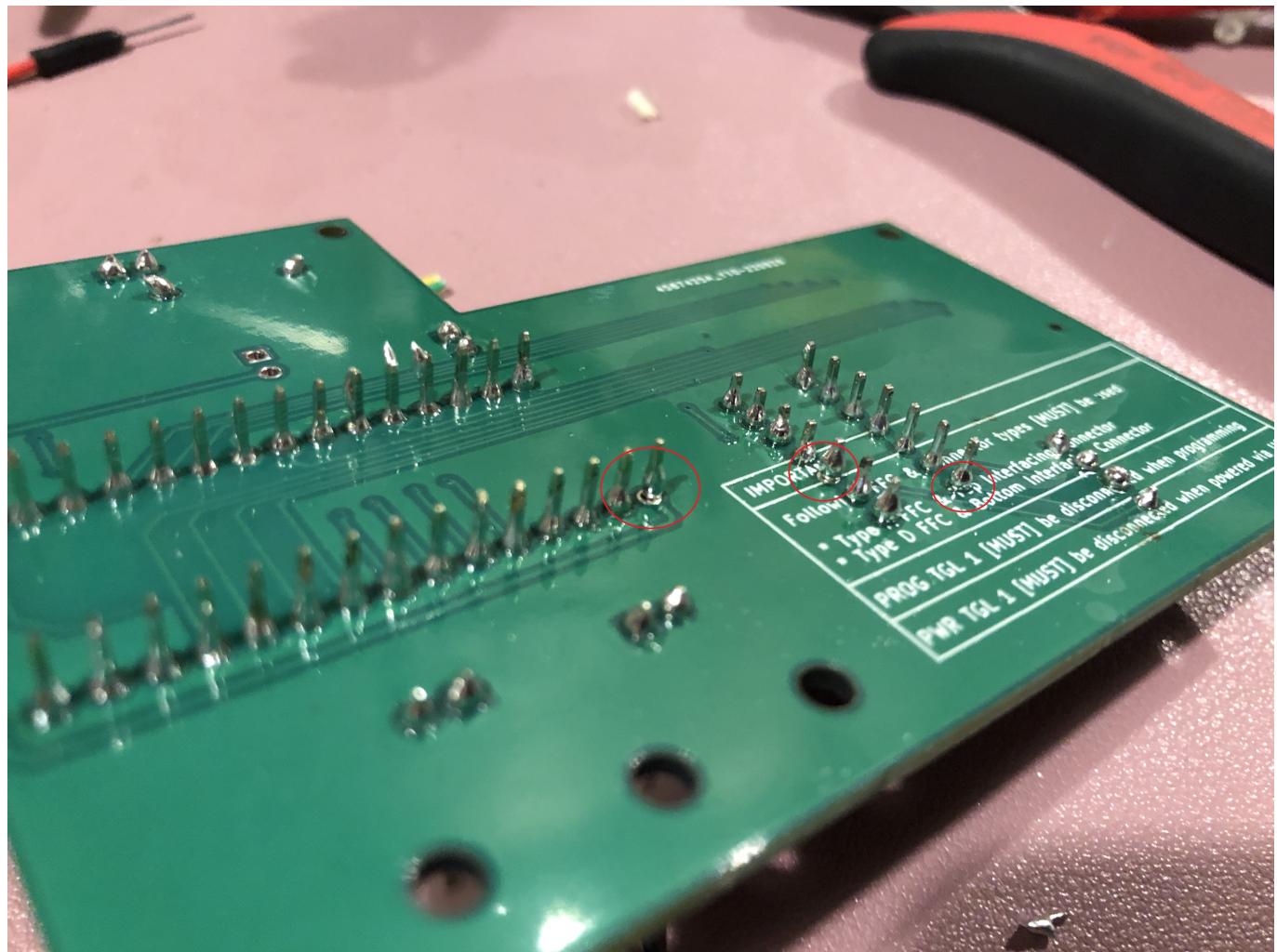
2.5 - Fitting the Arduino

The Arduino is fitted with its USB facing the edge of the board.

i In the images there is **USB** written on the board, this was a design error. **The USB should face to the outside edge of the board**

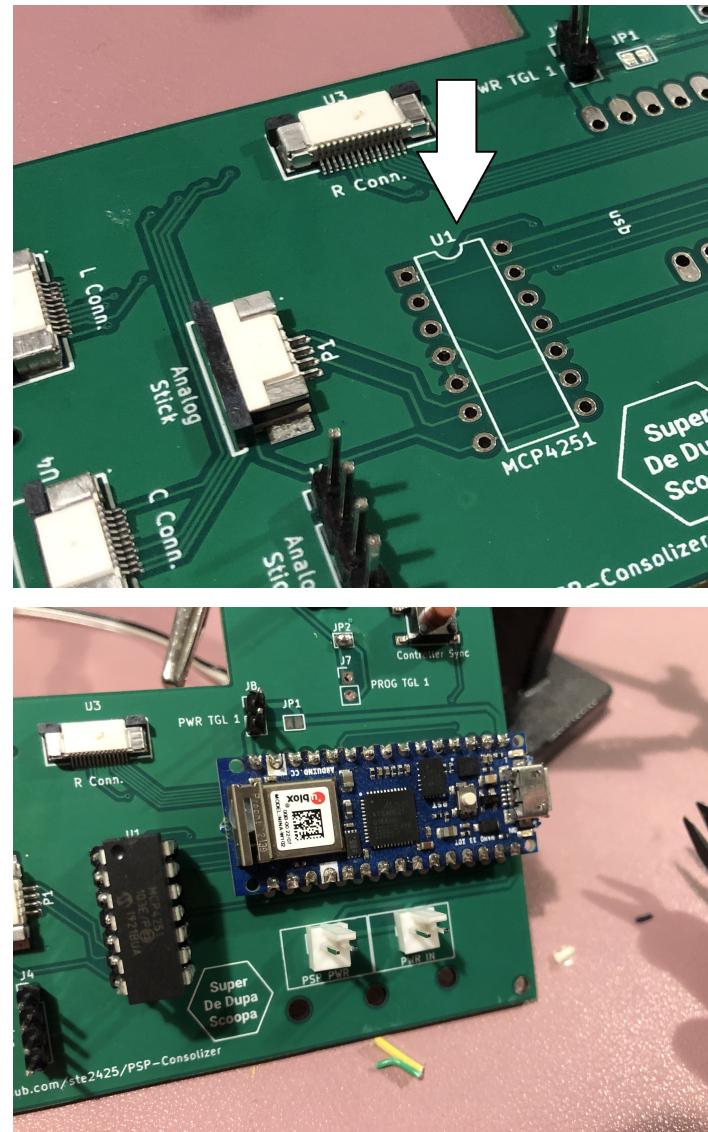
However you choose to solder the Arduino double check all soldering points as its easy to miss some.

In the following image there are multiple pins which did not solder properly both for the Arduino and the MCP4251.



2.6 - Fitting the MCP4251

When fitting the MCP4251 be sure to orientate it correctly. There is a little dot on the top of the chip, this should face the the same side as the notch and marking **u1** on the board.



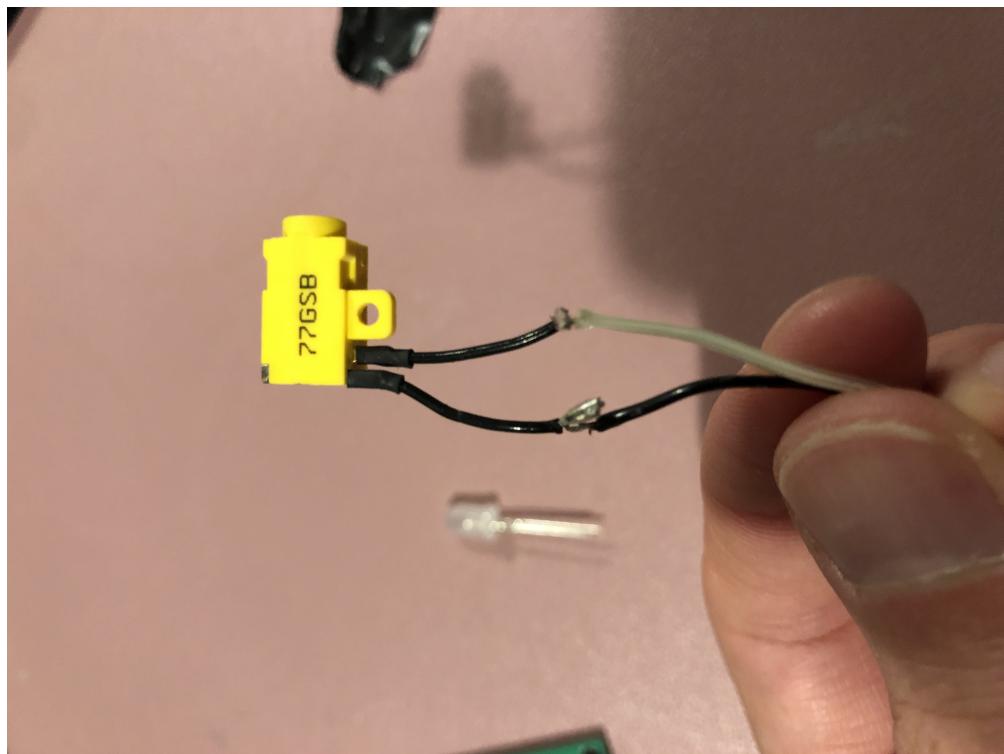
2.7 - Providing power to the PSP.

I chose to provide power to the PSP by buying a cheap replacement power cable for the PSP and then chopping it in halve.

The end with the yellow barrel jack i attached to the power in connector, the other half to the PSP power out.

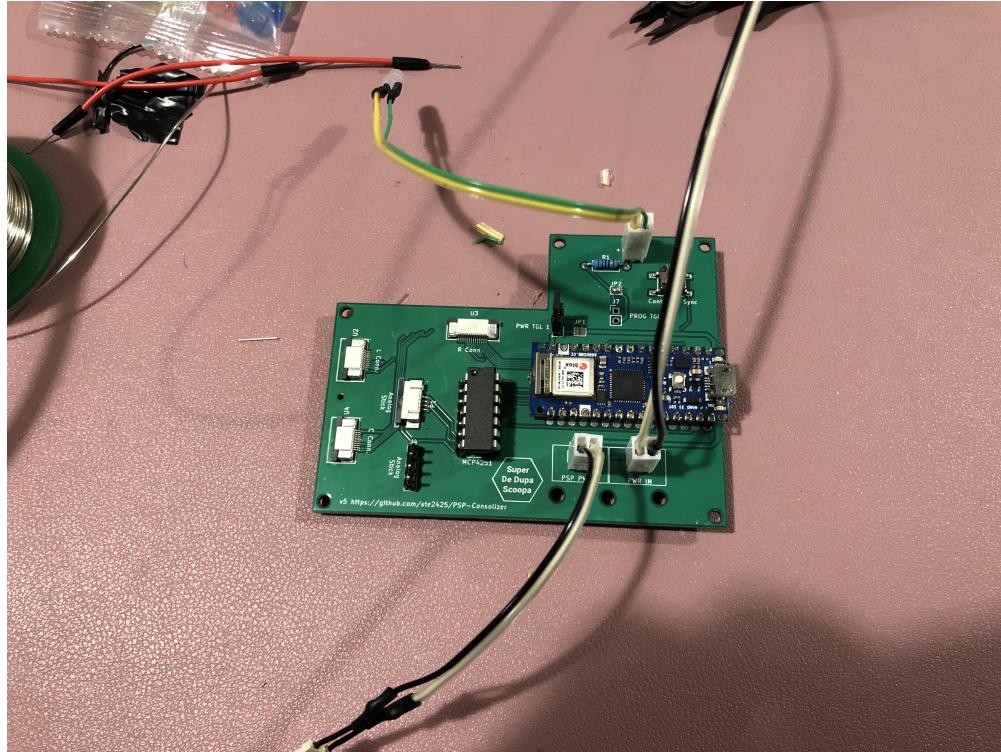
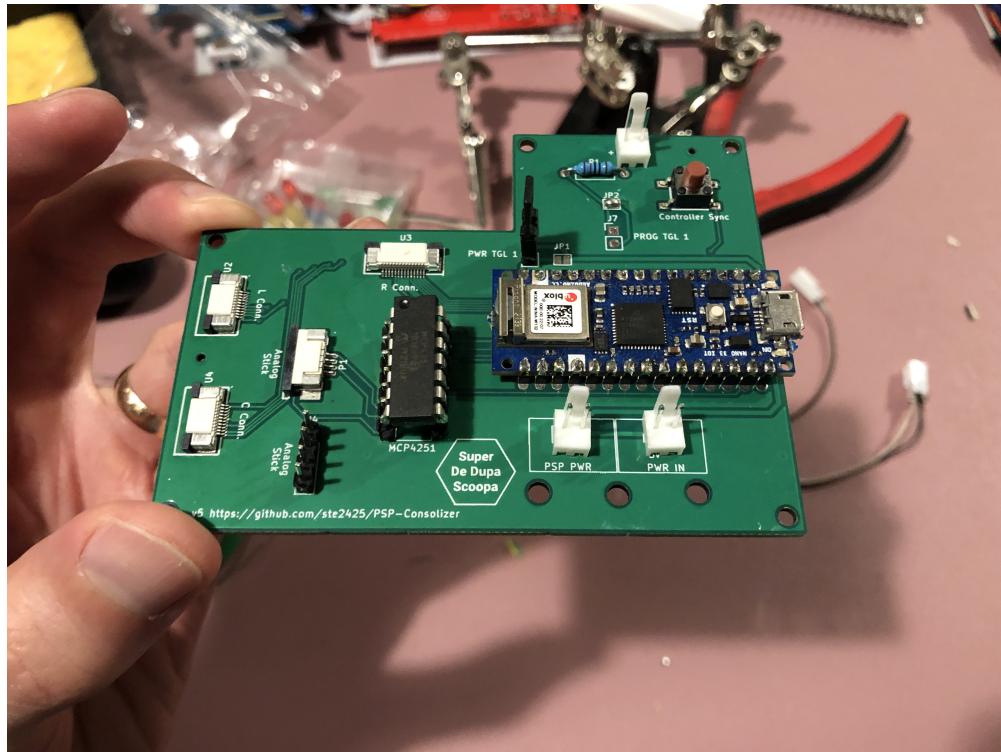
Be sure to check polarity with a multi meter, however i found the white wire in the below image to be positive.

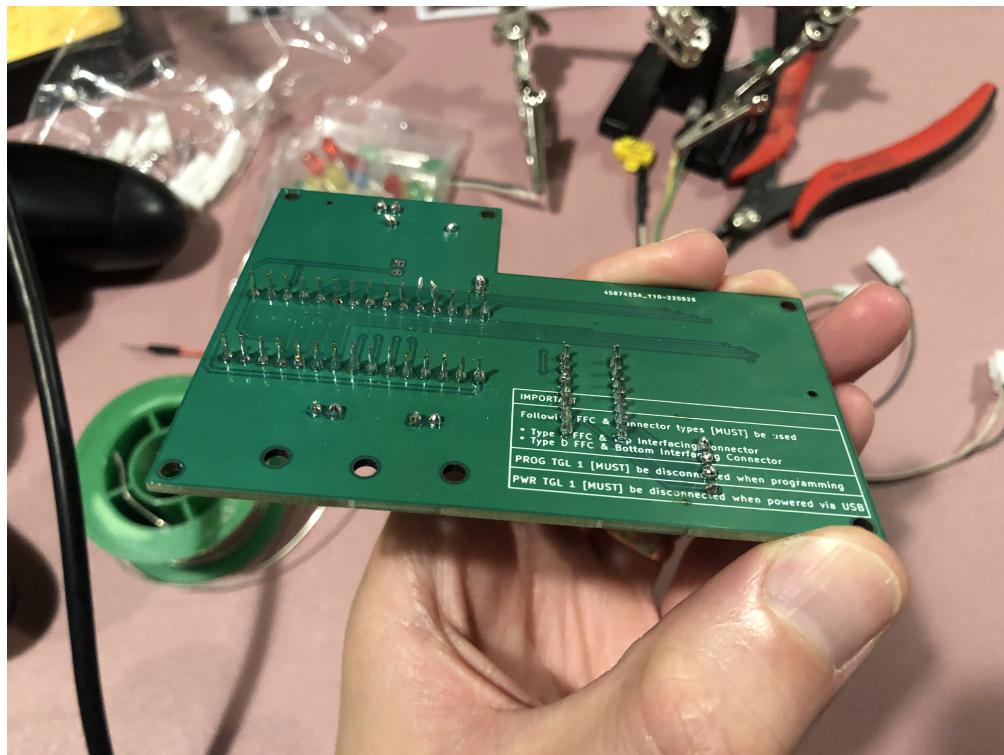
i Don't forget to put heat shrink on cables **before** soldering. Unlike me.



Assembly Finished

Your PSP Consolizer should now be fully assembled and looking something like this.





3 - Programming the Arduino

The arduino can be programmed before or after installing it into the PSP Consolizer. Launching the **psp-consoliser-programmer.exe** application downloaded from the release package.

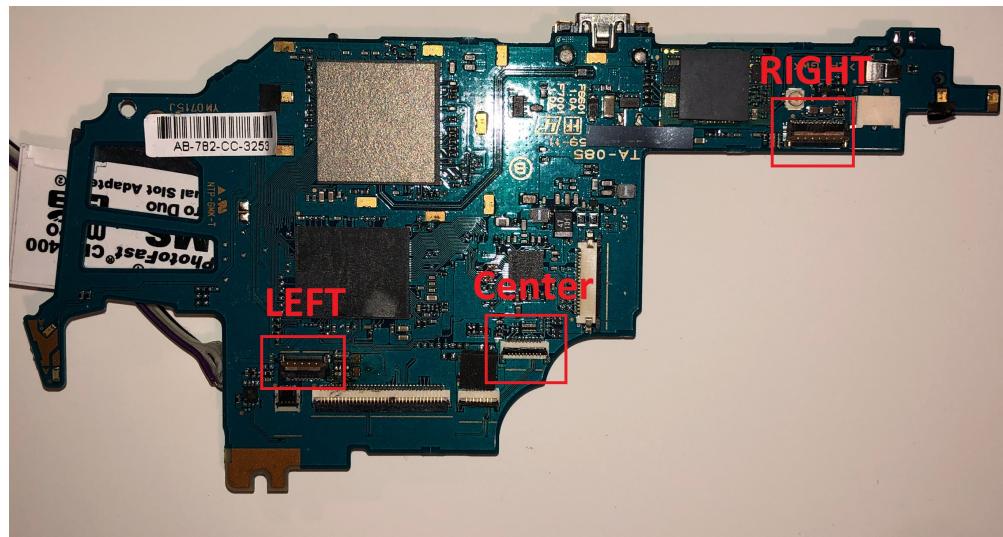
It will tell you what steps need to be followed next.

NOTE The exe is not signed. Your web browser of choice will complaining downloading the exe as will Windows when it launches it.

4 - Connecting the Arduino and PSP

The Arduino is connected to the PSP via 3 FFC cables and soldering four wires for the Analog stick.

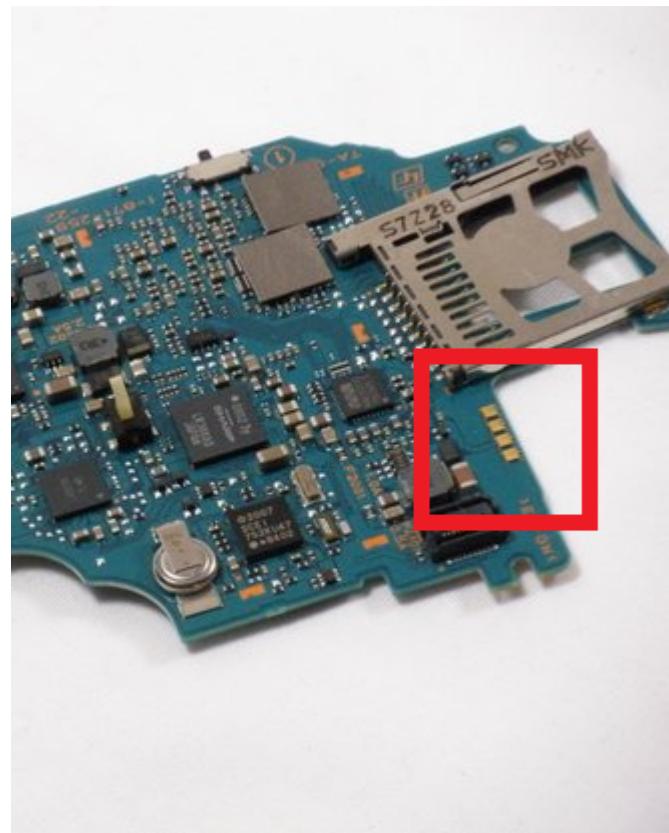
Connect the FFC's to the connectors on the PSP below. **Left** goes to **L Conn.**, **Right** to **R Conn.**, and **Center** to **C Conn..**



Don't forget to connect the power cable.

4.1 - Soldering the Analog Stick.

Next you need to solder to the following pins on the back of the PCB. These look like nice and easy large traces on the PCB. **They are not**



They are small metal tabs tack soldered to two small pads underneath on either side of them.

When you go anywhere near them the gold pads will lift.

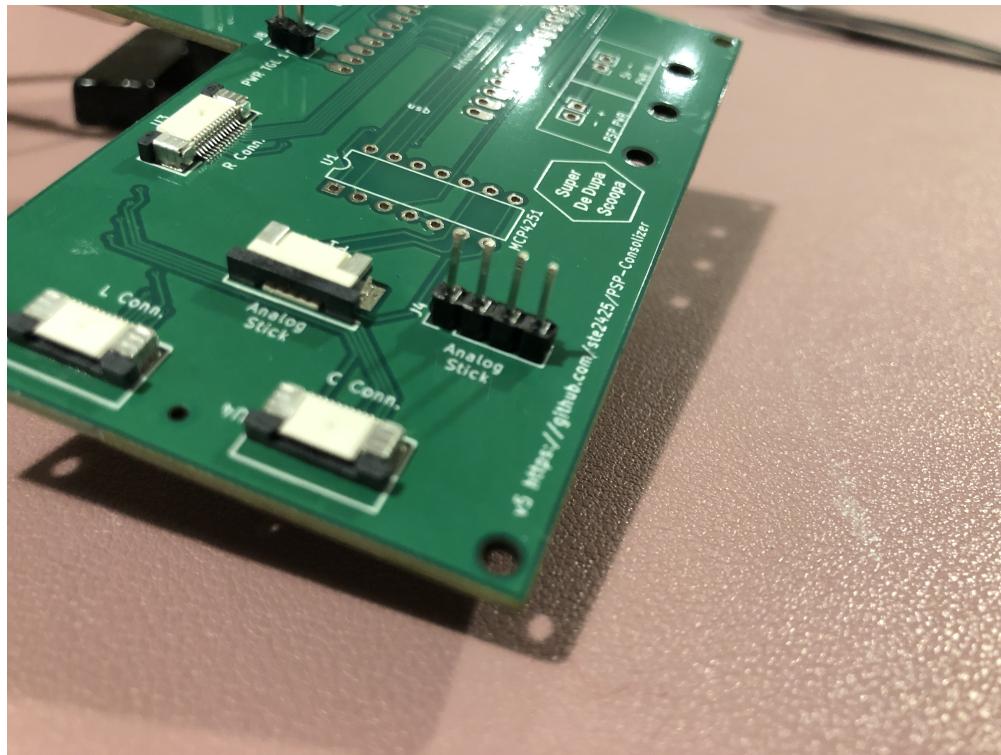
Lots of flux will help here and surface tension will help to stop solder bridges. If you do get bridges use solder wick to clean up it all up and try again. Do not put too much heat into the board as you can easily melt the small traces underneath. Lots of flux will help.

Once finished it should look something like this (my soldering is not great).



This is then connected on the headers soldered to the PCB earlier.

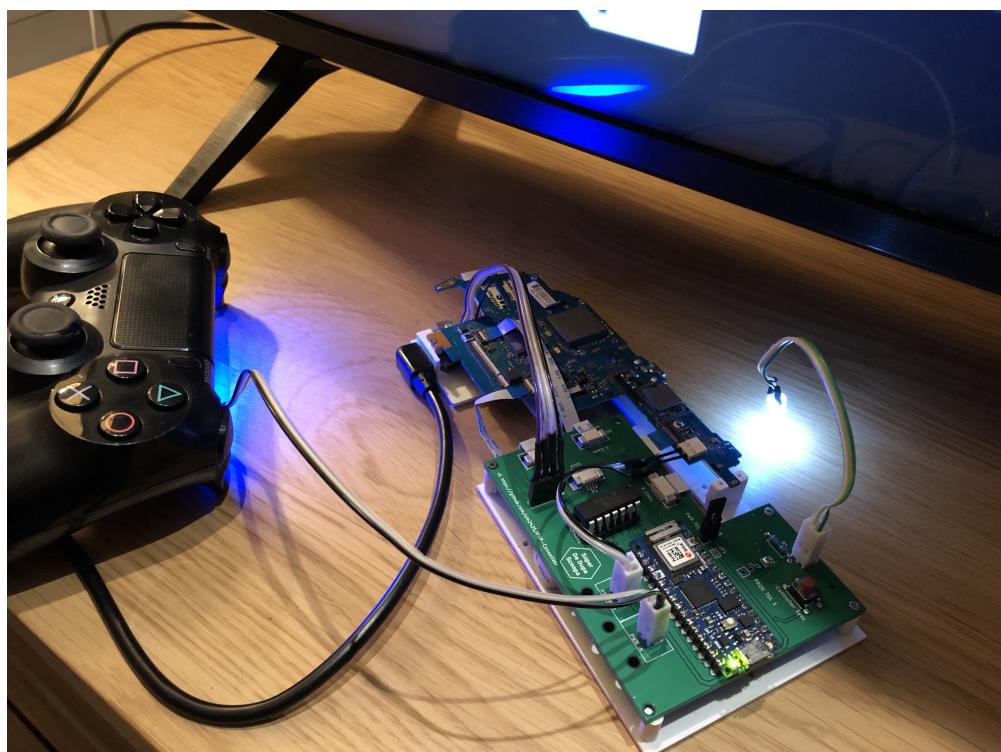
They are connected in order, with the black wire in the top image being connected to the left most pin on the connector below and the purple wire connected to the right pin.



The PSP is now fully connected and you should be complete.

The first time you turn Consolizer on you will need to press the sync button to sync a controller. Pressing this button will cause the Consolizer to forget any previously connected controller.

If a controller is connected it will not disconnect.



END OF DOCUMENT