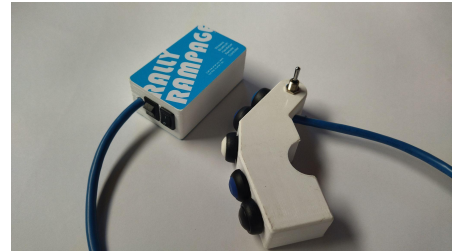


Welcome to our



Rampage Navigator Project!

Version 1.5

Updated: 20210927

This document is the explanation and instructions to assemble your own device. In the design and making of the unit, we have kept in mind the use of over-the-counter products. This will play a role in helping you source any exact components or similar components for your project.

What's included in these instructions:

- Components list
- 3D print model files
- Stickers for your little unit
- Some instructions on assembly
- Arduino IDE code for your Adafruit 32u4 board

So let's get started!

So let's get started!	1
What do you need to build this unit?	3
Components List	3
Tools	4
Adafruit Board Default Layout	5
The Rampage Nav Circuit Design	6
Notes on circuit assembly	7
Instructions	8
Arduino IDE Sketch Explanation	10
Final testing	11
Suggestions	11
Final thoughts	12
Open Source License - Conditions	13
GITHUB	13

What do you need to build this unit?

You will need to do and get the following:

- some electronic components to build your bluetooth board with buttons.
- A 3D printer to print the controller and housing.
- Soldering and electronic assembly with wires.
- Upload code onto your board using the Arduino IDE.
- Common sense...lol.. No seriously, you will need to think practically.

Components List

For this project you will need the following components:

#	Component	Qty	Online link of product and example
1	Adafruit 32u4 Bluefruit Board	1	https://www.robotics.org.za/AF2829
2	Mini Switch 12mm Waterproof	6	https://www.robotics.org.za/PBS-33B
3	Rocker Switch	2	https://www.robotics.org.za/KCD11
4	Toggle switch - center return	1	https://www.communica.co.za/products/m2019ss1w01
5	UTP Cable	1	Any 8 core UTP cable to use between your switches and your bluetooth box
6	Wires	1	Additional wire to connect between switches and common joints
7	2 Colour LED	1	https://www.robotics.org.za/RG5-10
8	M2 Screws	10	Miniature screws to attached lids
9	1000mA battery (FOR OPTION 1 DESIGN)	1	https://www.robotics.org.za/DTP603450?search=1000ma Battery to use for power if not permanently powered
10	5v Step down regulator (For OPTION 2 design)	1	https://www.robotics.org.za/3792 Used for permanent power on your unit to use your vehicle's 12v perm source.
11	Micro USB Charging Cable (For OPTION 2 design)	1	To be used for the regulator and cut up for perm power OPTION 2

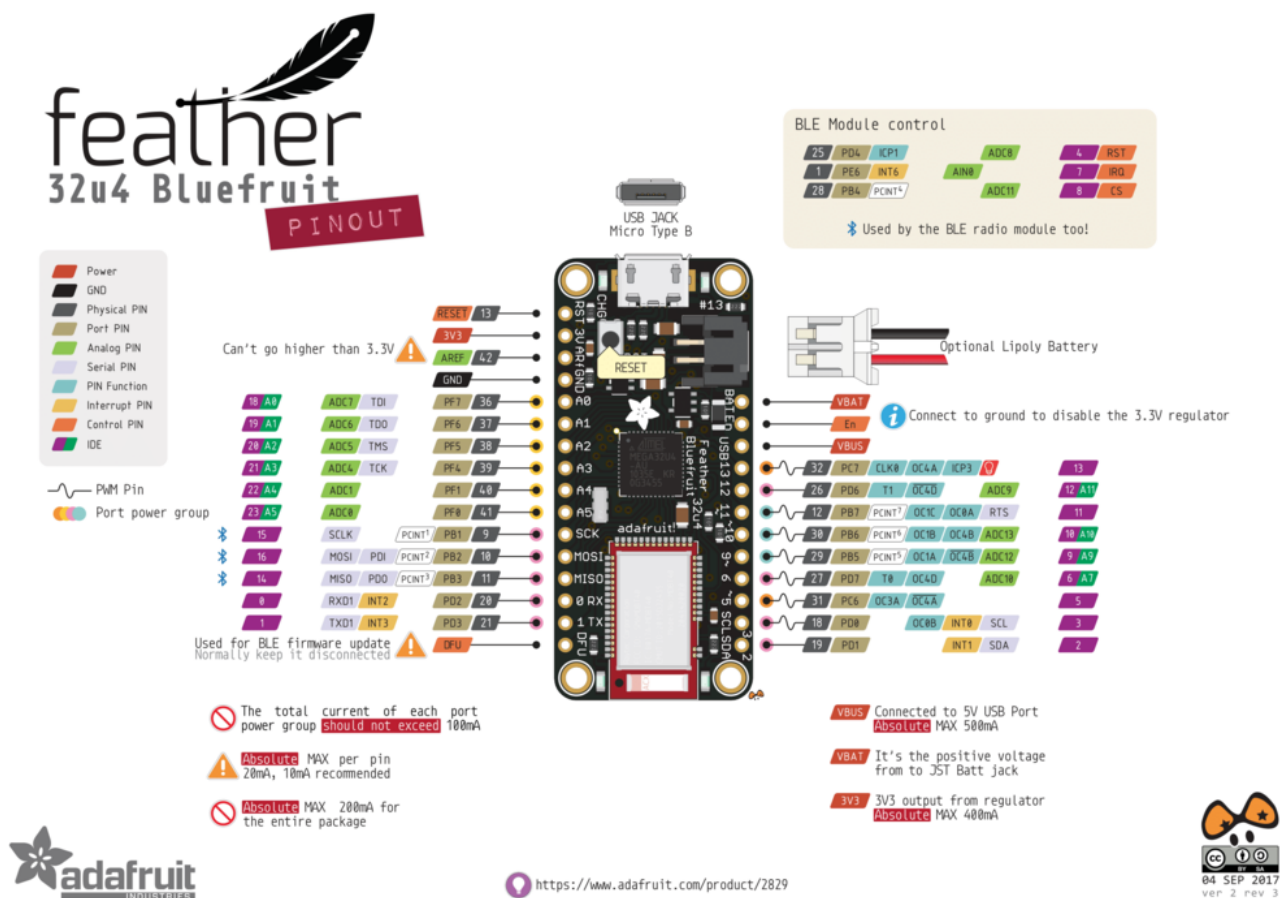
Tools

You will need the following minimum tools to build this device:

- Soldering iron and solder
- Wire cutters
- Wire stripper
- Computer with Arduino IDE installed

Adafruit Board Default Layout

The default PIN layout for this Adafruit board is as follows:



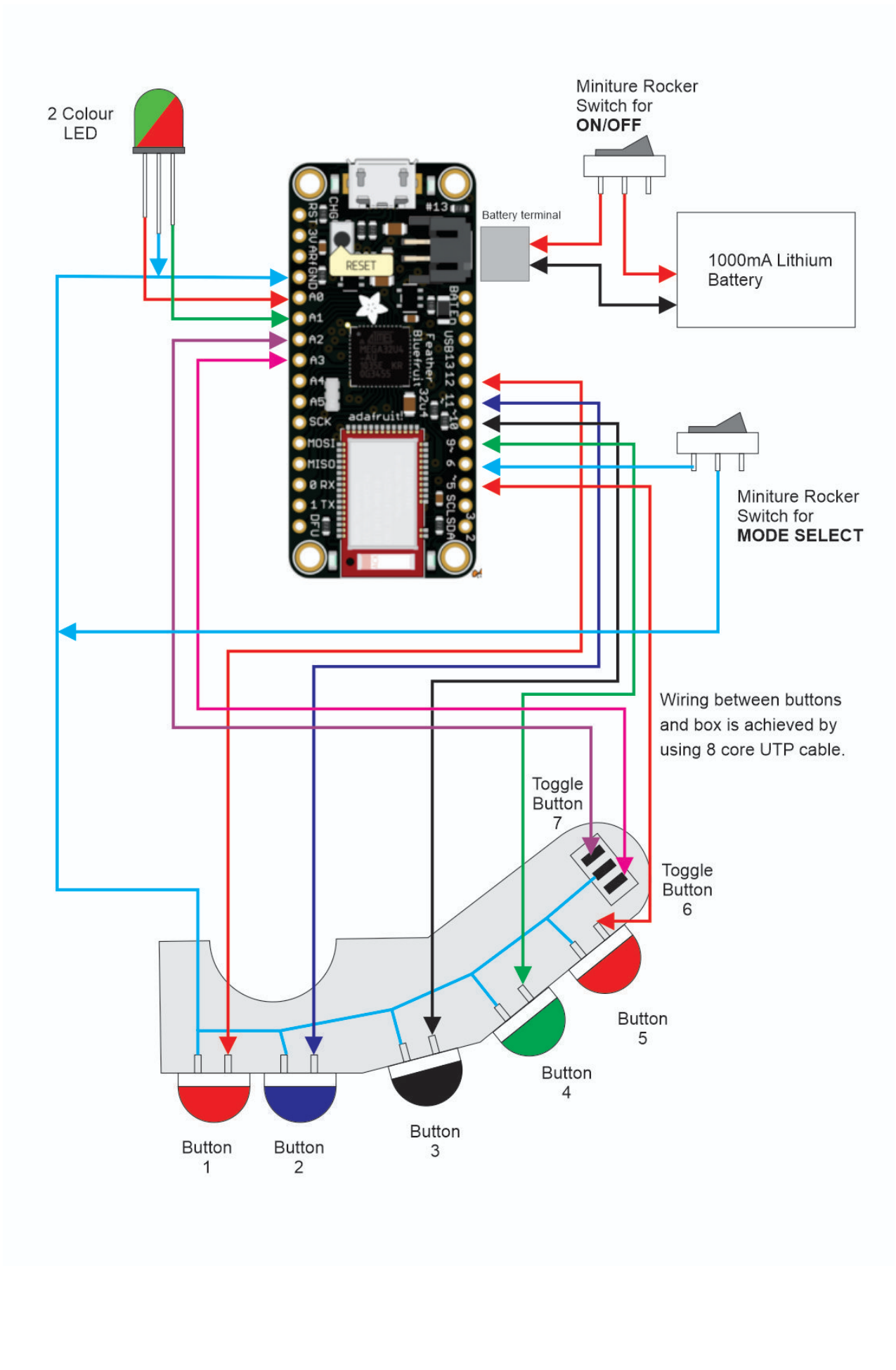
For more information about this board and how to set it up click this link:

<https://learn.adafruit.com/adafruit-feather-32u4-bluefruit-le?view=all>

DESIGNS OPTION1:

The Rampage Nav Circuit Design with Li Battery

The circuit design and buttons we are using are the following:



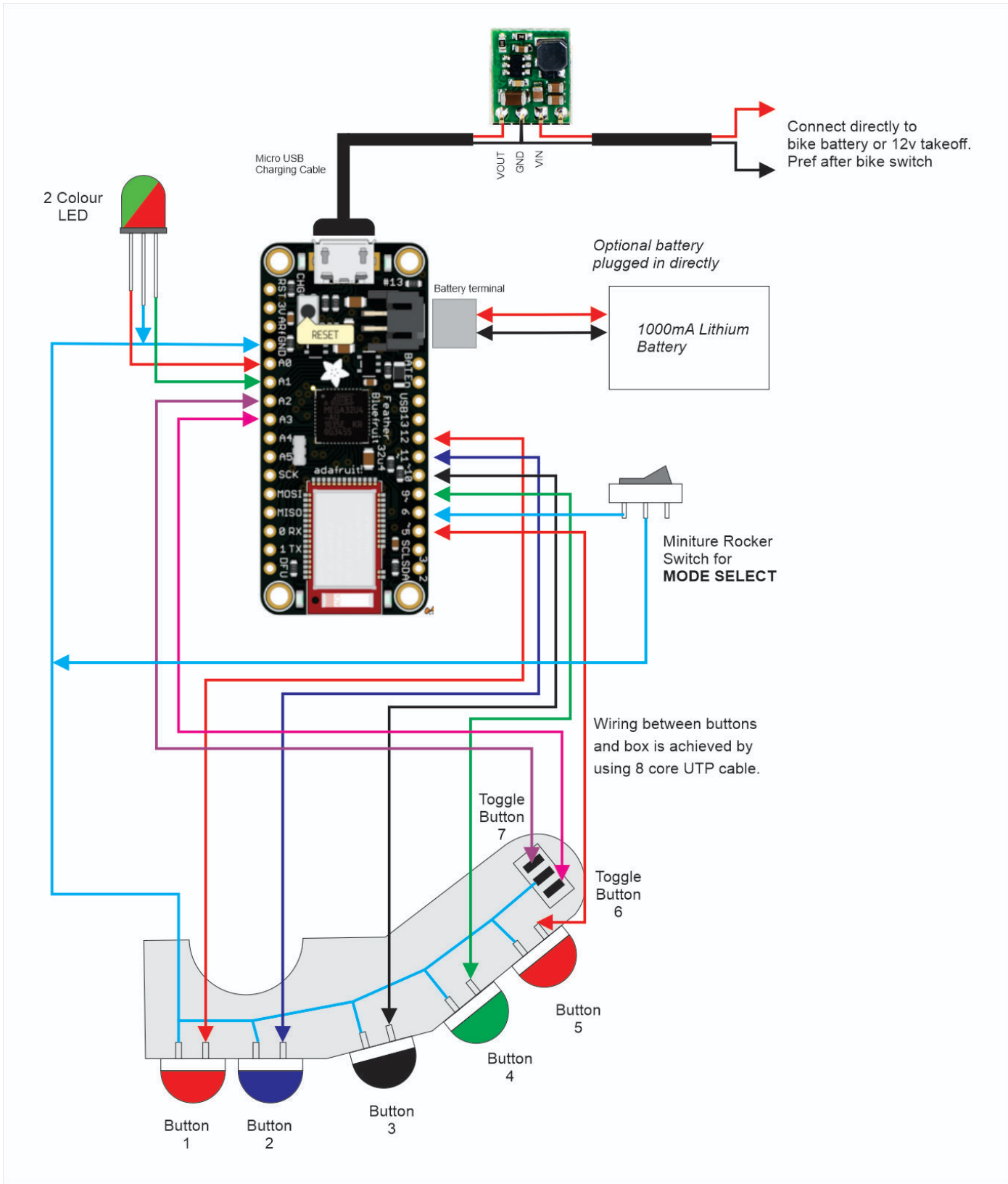
Notes on circuit assembly for Option 1

- The ground (GND) will need to go to all the buttons, one rocker switch and the central LED PIN.
- The LED PINs are close enough to just bend and solder directly into the board.
- The main mode LED must stick through the larger hole in the box.
- The battery lies at the bottom of the bluetooth box.
- Wiring between buttons and box is achieved by using 8 core UTP cable.

DESIGNS OPTION 2:

The Rampage Nav Circuit Design with permanent power from 12v vehicle

The circuit design and buttons we are using are the following:



Notes on circuit assembly for Option 2:

- The ground (GND) will need to go to all the buttons, one rocker switch and the central LED PIN.
- The LED PINs are close enough to just bend and solder directly into the board.
- The main mode LED must stick through the larger hole in the box.
- The battery lies at the bottom of the bluetooth box.
- Wiring between buttons and box is achieved by using 8 core UTP cable.
- For the perm power use an existing USB charging cable and simply solder the 5v regulator in the middle of the cable. Use the micro USB in your unit and cut off the larger USB as you will need positive and negative directly.

Instructions

Please remember this is a DIY project. You will need to have some understanding of electronics, soldering, assembly and Arduino IDE coding.

As far as I could I have made note of anything that would help you in building your buttons.

1. Preparation

- a. Buy all components.
- b. Prepare soldering and electronics tools for assembly.
- c. Install and set up Arduino IDE on your computer to flash the Adafruit board with the code.
- d. Cut the UTP cable in approx 30cm length. The one side will go into the button controller and the other will go into the bluetooth housing.

2. Boxes

- a. Print the 3D files to create your button controller and the bluetooth housing box.
- b. You must end up with the button controller housing as well as the bluetooth board housing.

3. Buttons

- a. Install the 5 push buttons into the 3D printed housing.
- b. Insert the toggle switch into the button housing.
- c. Insert the rocker switches into the bluetooth housing.

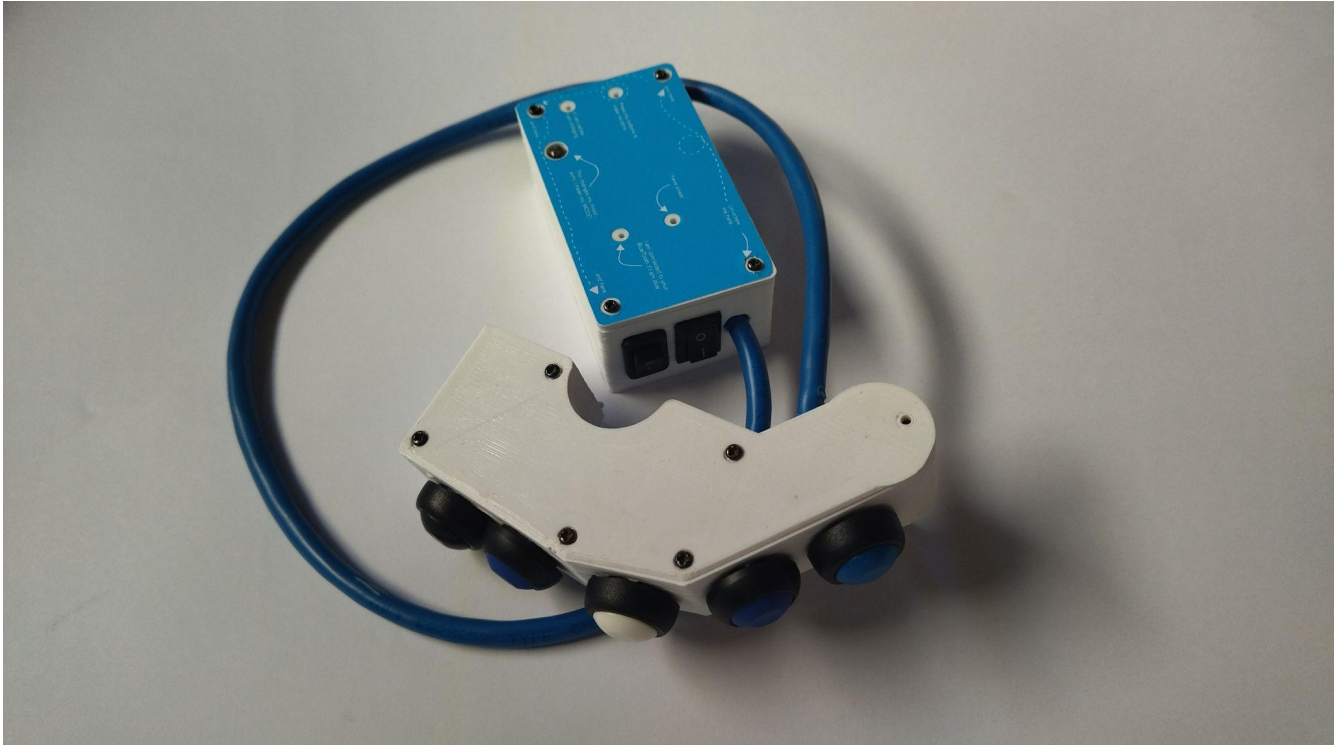
4. Soldering

- a. Solder the UTP cable between the button controller and the bluetooth housing. Making sure you solder the correct colour code of the UTP cable wire to the buttons and the pin on the Adafruit board as indicated in the above drawing of the board and its PINs.

5. Coding your board

- a. Setup your Arduino IDE interface as per these instruction from Adafruit themselves
<https://learn.adafruit.com/adafruit-feather-32u4-bluefruit-le/setup>
- b. Once setup, you can open the Rampage Navigator sketch on your computer
- c. I would suggest just Compiling the script to make it all fine.

- d. You can upload the script to your board.
- e. Start testing buttons using your serial monitor from the IDE



Arduino IDE Sketch Explanation

- The current sketch creates two states using PIN 6. These states are referred to as the ‘Modes’
- The button command for the various apps are set by enabling the correct app for the mode you want at line 23-31
- On the **first mode**, using a While statement (line 112) you can set any of the 7 pins applicable and send bluetooth commands as the buttons are pressed.
- The **second mode**, also using a While statement that starts at line 308. This mode then sends different bluetooth commands for the same buttons pressed.
- You will notice in this mode you simply assign different keyboard commands to be sent via bluetooth to your board.
- The following table has standard keyboard commands that can be used for different apps on your phone.

Button	Code Assign						
First push button	Button 1						
Second push button	Button 2						
Third push button	Button 3						
Fourth push button	Button 4						
Fifth push button	Button 5						
Toggle forward	Button 6						
Toggle back	Button 7						
Function	APP Locus Maps		Function	APP Rally Navigator		Function	APP Media player
Zoom in	Volume +		Trip Up	Volume +		Volume +	Volume +
Zoom out	Volume -		Trip Down	Volume -		Volume -	Volume -
Map up	r		Not assigned	Not assigned		Play/pause	Play/Pause
Center map	c		Scroll Up	Media Next		Next	Media Next
Display on/off	d		Scroll Down	Media Previous		Back	Media Previous
Zoom in	Volume +		Scroll Up	Media Next		Mute	Mute
Zoom out	Volume -		Scroll Down	Media Previous		Home	Android Home

- You can also open the serial monitor on your Aduino IDE and use this while pressing the buttons to make sure they are active and sending the correct code.

Final testing

- Once all your buttons are active and sending commands you are ready to test on your phone.
- Turn your phone's bluetooth on.
- Select the Rally Navigator bluetooth device that should be visible now.
- Once it pairs between the devices, you should hear what I call the 'happy sound' to show the device is seen as a keyboard for the phone.
- To test this pairing, select your phone's HOME button. Press the top controller button which is supposed to increase the phone's volume. If you see the phones volume goes up, it works.
- Make sure which mode your phone is on by looking at the two color LEDs. One colour will represent one mode, the other colour will represent the other mode.
- If you have used my normal sketch for your buttons, Mode 1 will have the functions for Locus Maps and the other Mode should work with media.

Suggestions

1. You can assemble the electronics first using a non-soldering breadboard (<https://www.robotics.org.za/NE00352>).
 - a. This works great to learn and understand the workings, code and layout.

Final thoughts

This is an open-source project to allow other parties to contribute to the code, design and project.

If you make improvements, please forward them back to us so we can eventually launch it publicly on our website.

Thanks for your involvement.

Eugene Beetge

info@rallyrampage.com

South Africa

GITHUB

This project is also available on GitHub at:

https://github.com/eugeneworldwide/rampage_navigation_controller

Open Source License - Conditions

This project is offered under the **GNU Affero General Public License v3.0**

The content of the license text is available at: <https://www.gnu.org/licenses/>