PX4-DShot Configuration Documentation

1 Introduction

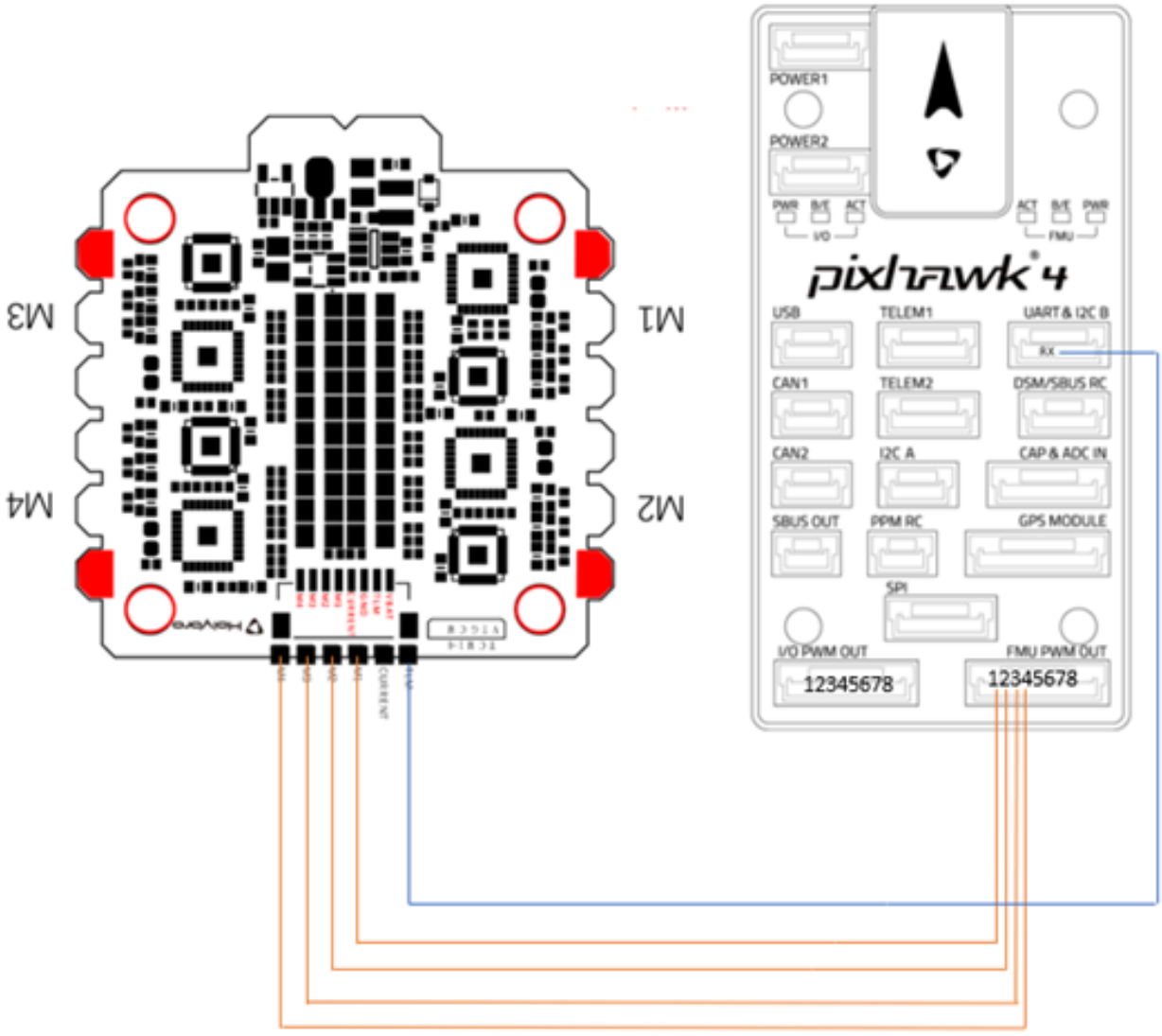
This document instructs how to setup a pixhawk 4 to run motors using the Holybro Tekko 32F3 Metal 4in1 ESC. The esc official site can be found at: http://www.holybro.com/product/tekko32f3-metal-4in1-esc65a/

Further explanation of using the DShot support module for testing developed by Fausto Tapia on top of the PX4 version 1.10. The source code can be cloned from: https://github.com/faustoTapia/px4_sem_project.git

2 Hardware

The hardware needed for setup is the following:

- pixhawk 4
- Holybro Tekko 32F3 Metal 4in1 ESC
- USB type A to USB type B micro cable (to connect computer to pixhawk)
- 10 pin connector and 6 pin connectors for Pixhawk 4
- ESC connector (Included in ESC box)



The following Connections need to be made from the ESC to the Pixhawk 4:

ESC	Pixhawk
1. VBAT (Battery Voltage measurement)	Voltage measurement pin (not crucial for esc protocol, can be left disconnected). This value can also be retrieved from the telemetry package instead.
2. TLM	Requires configuration in software, but can be either of: • UART & I2CB (RX, pin 3) • TELEM1 (RX, pin 3) • TELEM2 (RX, pin 3)
3. GND	Can be connected in any ground pin in PX4. For convenience set to the same port as TLM. So Could be either of:

	• UART & I2CB (GND, pin 6) • TELEM1 (GND, pin 6) • TELEM2 (GND, pin 6) Another convenient placement could also be along with FMU PWM OUT: • FMU PWM OUT (GND, pin 10)
4. Current	PX4 analog current measurement pin
5. M1. Motor 1 Throttle	From port FMU PWM OUT, either of: • FMU_CH1, pin 2 • FMU_CH2, pin 3 • FMU_CH3, pin 4 • FMU_CH4, pin 5
6. M2. Motor 2 Throttle	
7. M3. Motor 3 Throttle	
8. M4. Motor 4 Throttle	

A thorough schematic of pins and ports numbering for the Pixhawk 4 can be found here: http://www.holybro.com/manual/Pixhawk4-Pinouts.pdf

3 Software

NOTE: Although the development toolbox can be installed in Window, Linux or Mac OS X, it is RECOMMENDED TO use an Ubuntu LTD (I used it with Ubuntu 18.04 and it is also the recommended OS)

3.1 Install QGroundControl

Download and install QGroundControl. Detailed instructions are found here: https://docs.qgroundcontrol.com/en/getting_started/download_and_install.html

3.2 Setup development environment

The instructions in this section will walk through the **Linux** setup.

3.2.1 Installing "Gazebo, JMAVSim and NuttX (Pixhawk) Targets"

Install the "Gazebo, JMAVSim and NuttX (Pixhawk) Targets" analogous to what is described in (https://dev.px4.io/master/en/setup/dev_env_linux_ubuntu.html) for the original px4 software, but with different source code. Follow the steps below:

1. Download PX4 source code:
    a. Open a terminal window
    b. Browse to the directory where you would like to setup the environment
    c. Perform the command:

        git clone https://github.com/faustoTapia/px4_sem_project.git --recursive

        Note: It's a private repository so contact Fausto Tapia to request access.

2. Run the ubuntu.sh with no arguments (in a bash shell) to install.
    a. In the in the px4_sem_project folder run:

```
bash ./Tools/setup/ubuntu.sh
```

3. Restart the computer

If you would prefer to check other OSs, please check: https://dev.px4.io/master/en/setup/dev_env.html and select your OS. Again, Linux has more extensive support, so it is suggested.

3.3 Install and setup Visual Studio Code

IMPORTANT: when Following the PX4 standard setup instructions, instead of opening the folder "Firmware", open the folder "px4_sem_proj". The instructions you want to follow are:

1. Installation and Setup
2. Building

Please find the instructions in the site below:

https://dev.px4.io/master/en/setup/vscode.html

3.4 Building and uploading the code

1. On a terminal window, Navigate to the folder "px4_sem_project"
2. Physically plug the Pixhawk 4 via USB to the computer
3. Run the command:

```
make px4_fmu-v4_default upload
```

A successful upload will end with:

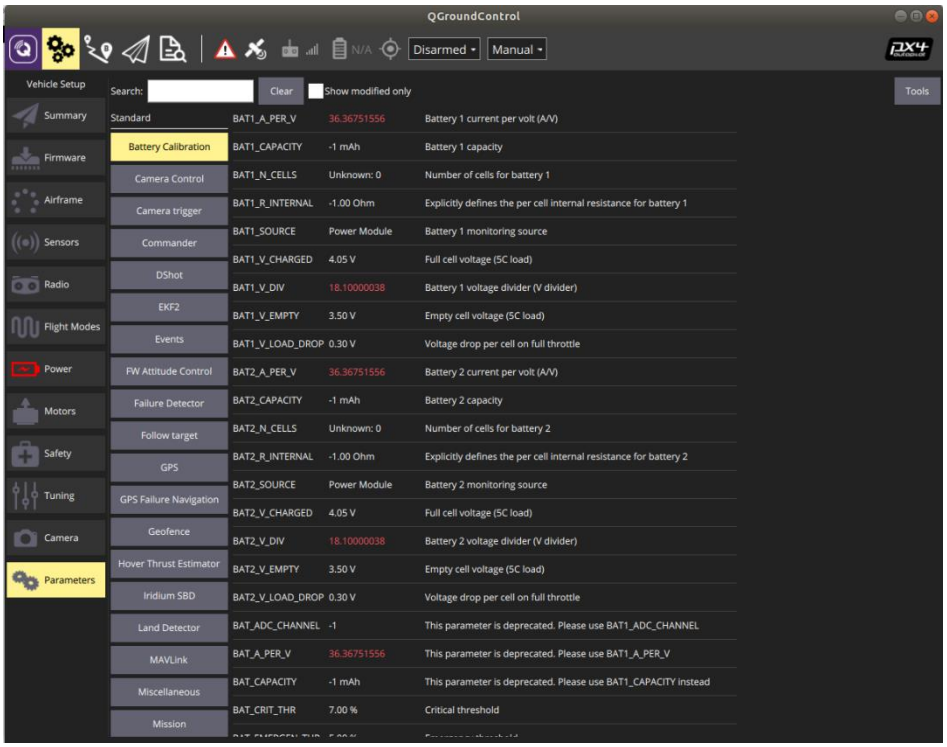
```
Erase  : [====================] 100.0%
Program: [====================] 100.0%
Verify : [====================] 100.0%
Rebooting.

[100%] Built target upload
```

3.5 Setting PX4 parameters (Necessary)

This section describes how to setup PX4 parameters to enable DShot esc communication protocol (Pixhawk 4 to ESC) and enable arming without safety checks (So that motors can be run in an experimental setup without a full UAV setup).

1. Open QGroundControl. And Navigate to the parameters setup window as shown below:

   Here you can search the parameters that will need to be changed using the "Search Window"



2. Search for the following Parameters and set them to the values as per the following tables:

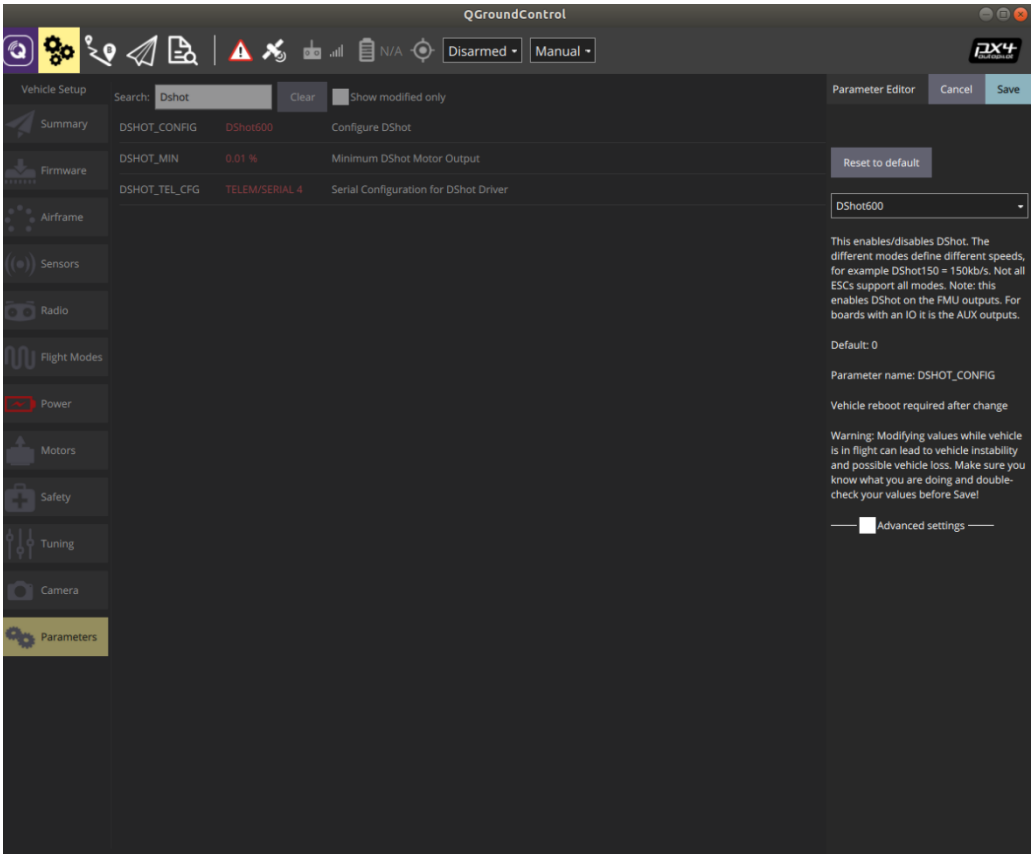
   To Enable Dshot:

Parameter	Value	Description
DSHOT_CONFIG	Dshot600 (Or other DShot options)	Enables DShot protocol. Either of 150, 300, 600 or 1200 can be used. Staying with intermediate speed as all of them can handle more than 800Hz command sending.
DSHOT_MIN	0.01	Minimum arm throttle as fraction.
DSHOT_TEL_CFG	TELEM/SERIAL 4	Directs the “UART & I2C B” pixhawk RX port to be used to receive ESC telemetry readings
MOT_POLE_COUNT	14	Used for RPM calculation. The current motors have 14 poles.
SYS_USE_IO	0	Directs output to FMU PWM OUT (As that port is directly connected to MC)

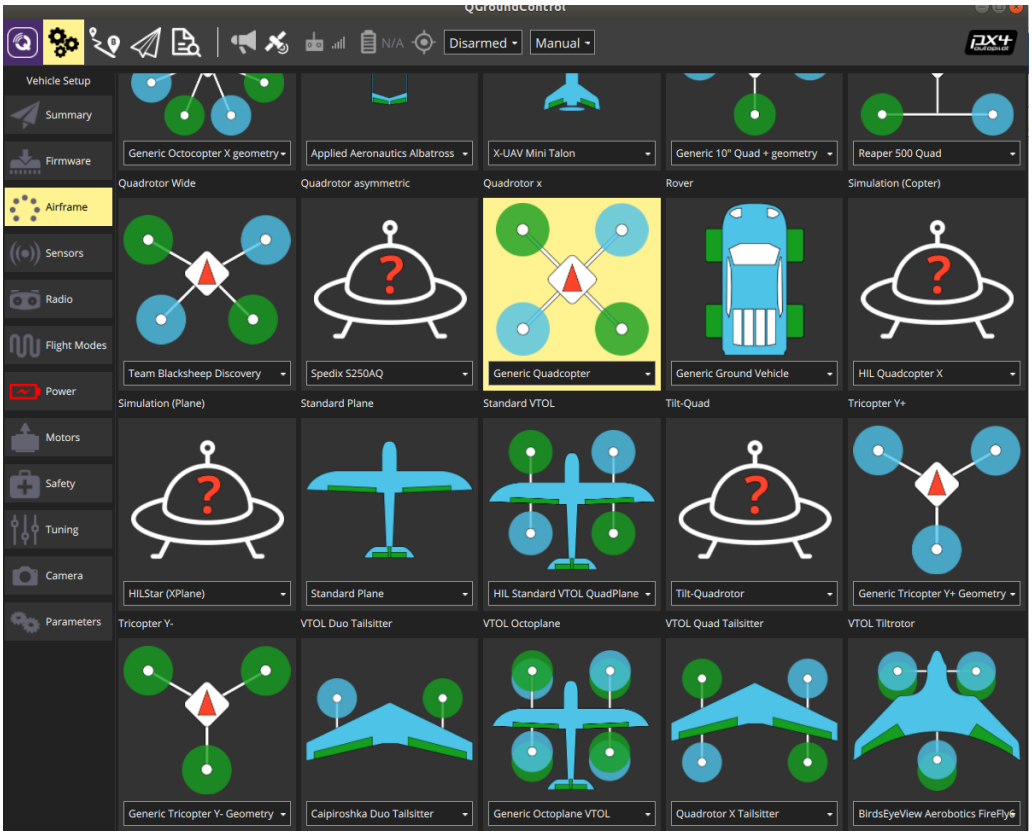
   To Disable safety checks when arming:

Parameter	Value	Description
CBRK_IO_SAFETY	22027	Disables IO safety
CBRK_USB_CHK	197848	Disables USB connected check
COM_ARM_CHK_ESCS	0	Disable ESC check for arming
COM_DISARM_PRFLT	0	Enables running motors pre-flight
COM_RC_IN_MODE	Joystick/No RC checks	Chooses to take inputs from joystick , hence enabling arming without RC

When selecting any parameter a sub-window will appear on the right where you can select the value desired. Then, need click Save.



3. Some of the mentioned parameters require System reboot. There after the setting all the parameters, reboot the pixhawk 4 by clicking: Parameters/Tools/Reboot Vehicle/OK.
4. Then check that the parameter changes have taken effect.
5. Select the Airframe to be a “Generic Quadcopter” and don’t forget to click “Apply and Restart” on the top right of the Airframe window



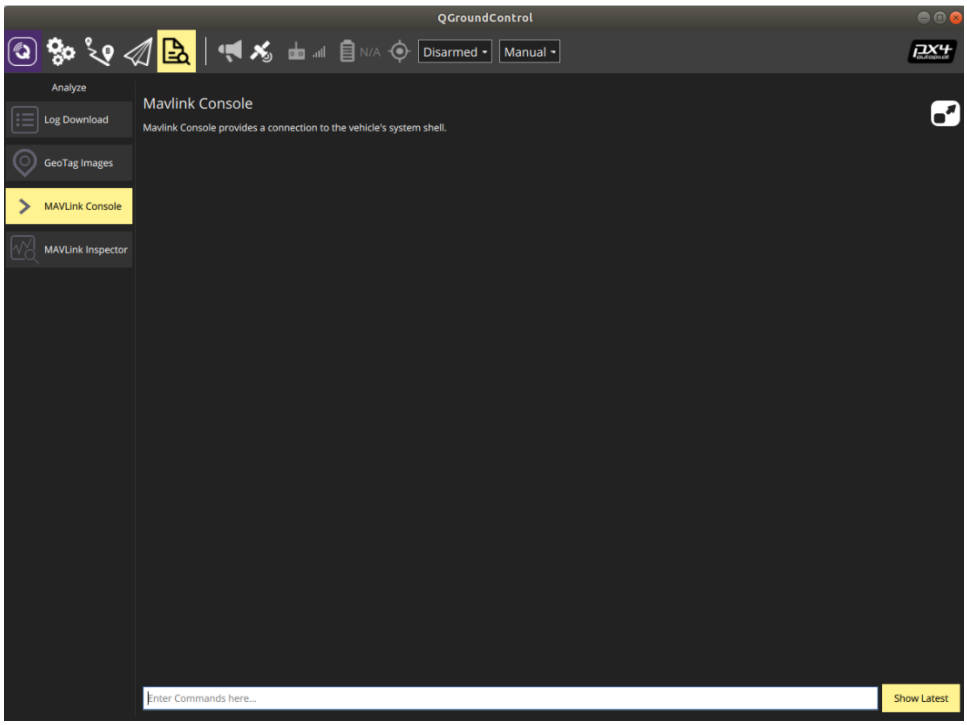
6. Wait for the Pixhawk 4 to restart and the setup is finished

3.6 DShot controller usage:

PX4 is setup in a modular structure. The module of interest is called: “dshot_controller”

First, you need to have access to the Nutshell, to interact with the pixhawk. Make sure that the pixhawk is connected to your computer via USB.

On QGroundControl, you can get access to the interactive Nutshell.



3.6.1 Startup procedure (NECESARY)

This needs to be performed to disable signal publishing conflicts with other modules. It must be done every time the pixhawk is restarted.

Perform the following commands on the Nutshell

1. mc_rate_control stop
2. dshot_controller start

3.6.2 Arming and disarming command:

This command will START SPINNING the motors at the minimum speed, so be CAREFUL:

commander arm -f

When you need to stop the motors, you can use the disarming command:

commander disarm

If everything run worked during setup, the nutshell should have given the following outputs

```
NuttShell (NSH)
nsh> mc_rate_control stop
nsh>
dshot_controller start
INFO  [dshot_controller] Instance Initialized
nsh>
commander arm -f
nsh>
```

Additionally, you can check the arming status with the command:

```
commander status
```

3.6.3 Using dshot_controller commands

If you type the module name (dshot_controller) in the nutshell it will show the available commands with a respective explanation. As below

```
NuttShell (NSH)
nsh> dshot_controller
Usage: dshot_controller <command> [arguments...]
 Commands:

   start

   stop

   status         print status info

   arm            Sends arming command to actuator_armed topic

   disarm         Sends disarming command to actuator_armed topic

   throttle       Prints current throttle being sent

   command        Sends throttle command to specified motor (others 0)
     [-m <val>]   Motor selected (1-8)
                  default: 1
     -p <val>     Throttle value (0-1)
     [-a]         Sets all motors to given power values

   listen_esc     Shows a number of esc datasets retrieved
     [-n <val>]   Number of datasets to receive (1-100)
                  default: 1

   run_sine       Outputs a Sinusoid to a selected motor
     [-m <val>]   Motor selected (1-8)
                  default: 1
     [-p <val>]   Period in ms (10-10000)
                  default: 2000
     [-d <val>]   Duration in s (2-20)
                  default: 10
     [-o <val>]   Wave Offset (0.01-0.9)
                  default: 0.5
     [-a <val>]   Amplitude (0-0.5)
                  default: 0.1
nsh>
```

The commands “arm” and “disarm” in the “dshot_controller” module will only be called instantaneously. They are only used for debugging (without commander module). Hence, you do not need to and should NOT use them.

Hence the way to call them in the nutshell is as follows:

dshot_controller <command> <option 1> <val1> <option 2> <val2> <option 3>…

Note that options with square brackets in the description are optional (I.e. if not given they will take a default value or behavior)