BaBee: Agilicious Documentation for Autonomous Flying

1. Components List

Main Compute Unit: Nvidia Jetson Xavier NX
Breakout board: A203 Carrier board for Jetson Nano/Xavier NX V2
Flight Controller: SpeedyBee F7 V3 FC
Electronic Speed Controller: SpeedyBee BL32 50A 4-in-1 ESC
Radio Receiver: FS-X6B
Radio Transmitter: FS-16X
Main Plate: ?
Motors: ?
Propellers: ?
Battery: Tattu R-Line V3.0 2000mAh 4S 120C LiPo Battery

2. Firmware List

To follow our recommended process, you will need the following firmware:

a. A *Base Computer* running Ubuntu 20.04, with installed:
   i. Motive (Mocap software)
   ii. Betaflight (FC software - we recommend using a stable release)
   iii. ROS
b. A *Babee Computer* running Ubuntu 20.04, with installed:
   i. ROS

3. Mechanical Assembly

The mechanical assembly of the quadrotor’s frame (SpeedyBee Frame V2) can be done following this tutorial:
https://youtu.be/hWUoH5cyySo?si=n4RCJ04sYB27ETVS.

4. Flight Controller (FC) Electronic Assembly

The electronic assembly of the flight controller (SV-F7V3-BL32-50A) can be done following the tutorials below:
https://youtu.be/VfNIDSZoTvc?si=3v8P6FFKEZn0rWTS
https://youtu.be/lw5rZoBxtzk?si=9PThKLTMh40VmC07
https://youtu.be/UVIMAifJjvE?si=jcgfB9_tG2y0-G_0

5. (Optional) Betfalight Software General Tutorials

For those unused to Betaflight, we recommend watching the following playlist/serie of videos, which explain in detail every aspect of the software and its configurations:
https://youtube.com/playlist?list=PLwoDb7WF6c8nT4jjsE4VENEmwu9x8zDiE&si=eUgScTlpqcQ8FIik.

6. Flashing Jetson Xavier NX

In order to flash Jetson Xavier NX on an A203 Carrier Board, you may want to follow the official instructions from the following link:

https://wiki.seeedstudio.com/reComputer_A203_Flash_System/#flashing-jetpack-os-via-command-line.

Note: We **highly recommend not** to flash using Nvidia's SDK Manager. Instead, we suggest using the **command line** installation steps from the above link.

Jetson Linux Version: "35.5.0 >"
JetPack version: "5.0.2"

7. (Optional) Betaflight Software for Remote Control

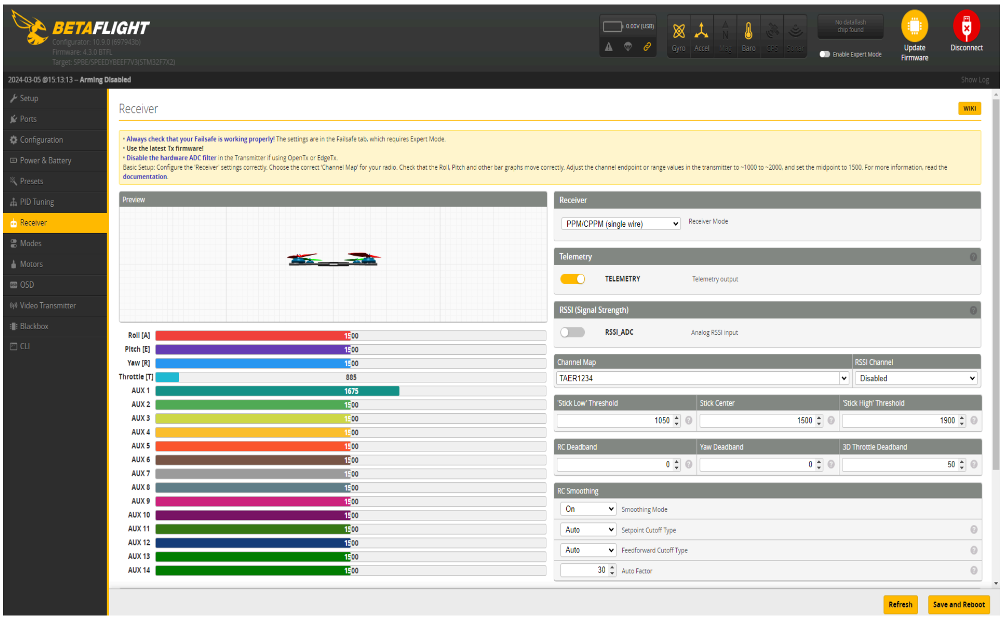
a. Reminder that we use the *Remote Controller* **FS-I6X Transmitter**, and *Receiver* **FS-X6B**, where the below schematic applies:

Remote Controller ————————>	Flight Controller
GND	*G*
5V	*4V5*
PPM	*R2*

Note that while *PPM* is working fine, *iBUS* is not recommended.

b. The RC needs to be configured separately to unable arming on RC's end. For that, you can follow: https://www.youtube.com/watch?v=IyqV5KMwK4I.

c. Then, you must allow for communication between RC and FC. To do so, setup Betaflight's receiver configuration as in the below image:



2024-03-05 @15:13:13 – Arming Disabled

Receiver

- Always check that your Failsafe is working properly! The settings are in the Failsafe tab, which requires Expert Mode.
- Use the latest Tx firmware!
- Disable the hardware ADC filter in the Transmitter if using OpenTx or EdgeTx.

Basic Setup: Configure the 'Receiver' settings correctly. Choose the correct 'Channel Map' for your radio. Check that the Roll, Pitch and other bar graphs move correctly. Adjust the channel endpoint or range values in the transmitter to ~1000 to ~2000, and set the midpoint to 1500. For more information, read the documentation.

Roll [A]	1500
Pitch [E]	1500
Yaw [R]	1500
Throttle [T]	885
AUX 1	1675
AUX 2	1500
AUX 3	1500
AUX 4	1500
AUX 5	1500
AUX 6	1500
AUX 7	1500
AUX 8	1500
AUX 9	1500
AUX 10	1500
AUX 11	1500
AUX 12	1500
AUX 13	1500
AUX 14	1500

Receiver
PPM/CPPM (single wire) Receiver Mode

Telemetry
● TELEMETRY Telemetry output

RSSI (Signal Strength)
○ RSSI_ADC Analog RSSI input

Channel Map	RSSI Channel
TAER1234	Disabled

'Stick Low' Threshold	Stick Center	'Stick High' Threshold
1050	1500	1900

RC Deadband	Yaw Deadband	3D Throttle Deadband
0	0	50

RC Smoothing
On Smoothing Mode
Auto Setpoint Cutoff Type
Auto Feedforward Cutoff Type
30 Auto Factor

d. Finally, you may want to set up the **Angle** and **Horizon** mode under the *Modes* tab of Betaflight.

8. Hardware Connections

a. SBUS has been used to connect FC (flight controller) and SBC (single board computer). Two **distinct** RX-TX pairs will be required on FC and SBC (you must not use the same X for RX/TX pairs).
b. Solder the FC R2 and connect it to SBC USB-to-TTL cable on RX (note that the USB-to-TTL cable inverts RX and TX, therefore we connect R2 to USB_RX).
c. Solder the FC T3 and connect it to SBC GPIO RX (UART_1; Pin 10, W8).
d. Connect FC GND to SBC GPIO GND (Pin 6, W8).

9. Betaflight Software for Autonomous Flight

a. Flashing (tuto - https://youtu.be/LkBWRiEGKTI?si=Y_OdhbqLRy0kyudE)

- Go to “Firmware Flasher”
- Press “Auto Detect” to choose your flight controller. If it does not auto-detect, DO NOT FLASH and do research to find out what the correct target is.
- We recommend flashing on firmware 4.3.0.
- First, click “save backup”. The FC comes with some customisation on it. If in future, all the data is erased, your FC will go to default settings, which is not what we want.
- Load firmware online
- Flash Firmware
- Now on the top right, the menu should change to DFU mode. If it does not appear automatically, watch the video above.
- After flashing, press connect on the top right.
- Press “Apply Custom Defaults”
- Press “Connect “ again
- Betaflight might give some warnings now. Just follow the steps given.

NOTE: NEVER click “Reset settings” button in the “Setup” tab. This will reset the manufacturer’s customisation. This is not what you may want.

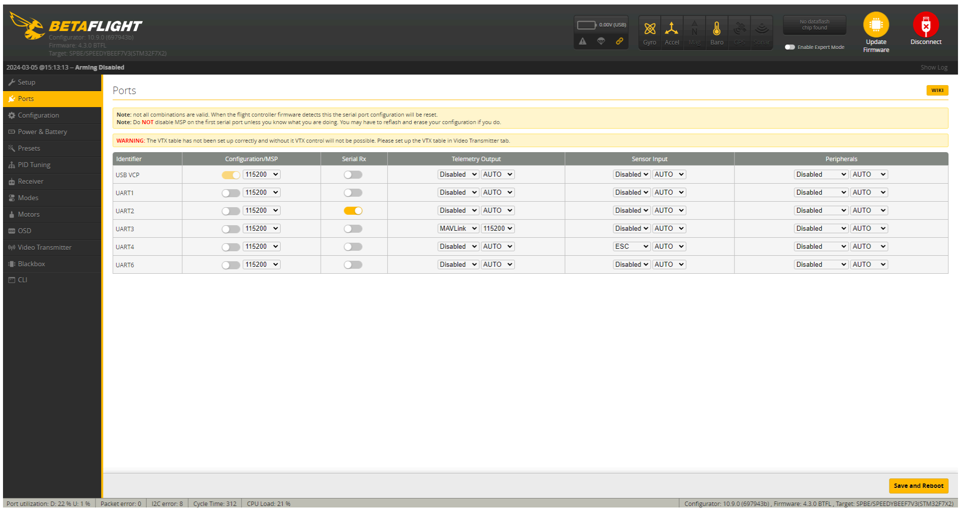
b. **Calibration**
In the *Setup* tab of Betaflight, click *‘Calibrate Accelerometer’* and follow the given instructions.

c. **Ports (tutorial - https://youtu.be/UDksru4JWnE?si=eHp38Ue3zo30OkMd)**
Note that in each row, you can toggle at most one of the available settings (Configuration/MSP; Serial RX, Telemetry Output; Sensor Input; Peripherals). We set the following:

i. *USB VCP* is by default set to **MSP** and should never be touched unless you know what you are doing. This would disable access to FC from USB-C connection and may require resetting the FC completely.
ii. UART2’s *Serial RX* to **ON** as R2 is used on the FC as a serial receiver.

iii. UART3's *Telemetry Output* to **MAVLink** (communication protocol used by Mavros) with *baud rate* as **115200**.

iv. UART4's *Sensor Input* to **ESC** to allow communication from FC to ESC.



2024-03-05 @15:13:13 – **Arming Disabled**

Ports

Note: not all combinations are valid. When the flight controller firmware detects this the serial port configuration will be reset.
Note: Do **NOT** disable MSP on the first serial port unless you know what you are doing. You may have to reflash and erase your configuration if you do.

WARNING: The VTX table has not been set up correctly and without it VTX control will not be possible. Please set up the VTX table in Video Transmitter tab.

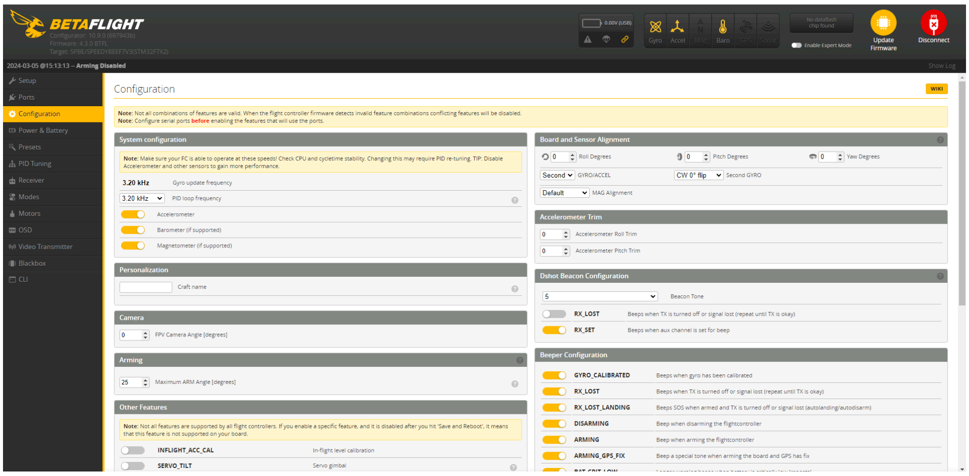
Identifier	Configuration/MSP	Serial Rx	Telemetry Output	Sensor Input	Peripherals
USB VCP	● 115200	○	Disabled AUTO	Disabled AUTO	Disabled AUTO
UART1	○ 115200	○	Disabled AUTO	Disabled AUTO	Disabled AUTO
UART2	○ 115200	●	Disabled AUTO	Disabled AUTO	Disabled AUTO
UART3	○ 115200	○	MAVLink 115200	Disabled AUTO	Disabled AUTO
UART4	○ 115200	○	Disabled AUTO	ESC AUTO	Disabled AUTO
UART6	○ 115200	○	Disabled AUTO	Disabled AUTO	Disabled AUTO

Port utilization: D: 22 % U: 1 % | Packet error: 0 | I2C error: 8 | Cycle Time: 312 | CPU Load: 21 %
Configurator: 10.9.0 (697943b) , Firmware: 4.3.0 BTFL , Target: SPBE/SPEEDYBEEF7V3(STM32F7X2)

d. Configuration (https://youtu.be/eY_H6qtpU18?si=YzRFTZ9OLYa4ZBWI)
Our configuration tab is left (nearly) as default. A few minor changes have be done, such as the following:

i. *RX_LOST* in *Dshot Beacon Configuration* may be set to **ON/OFF** based on your preference. Enabling this will enable a beeping sound when your FC does not receive any signal from a controller while armed. We have disabled this setting for convenience during testing.

ii. *Beacon Tone* simply indicates which beeping sound you wish to use and is not important.



Configuration

Note: Not all combinations of features are valid. When the flight controller firmware detects invalid feature combinations conflicting features will be disabled.
Note: Configure serial ports **before** enabling the features that will use the ports.

System configuration

Note: Make sure your FC is able to operate at these speeds! Check CPU and cycletime stability. Changing this may require PID re-tuning. TIP: Disable Accelerometer and other sensors to gain more performance.

3.20 kHz Gyro update frequency
3.20 kHz PID loop frequency
● Accelerometer
● Barometer (if supported)
● Magnetometer (if supported)

Personalization

Craft name

Camera

0 FPV Camera Angle [degrees]

Arming

25 Maximum ARM Angle [degrees]

Other Features

Note: Not all features are supported by all flight controllers. If you enable a specific feature, and it is disabled after you hit 'Save and Reboot', it means that this feature is not supported on your board.

○ INFLIGHT_ACC_CAL In-flight level calibration
○ SERVO_TILT Servo gimbal

Board and Sensor Alignment

0 Roll Degrees 0 Pitch Degrees 0 Yaw Degrees
Second GYRO/ACCEL CW 0° flip Second GYRO
Default MAG Alignment

Accelerometer Trim

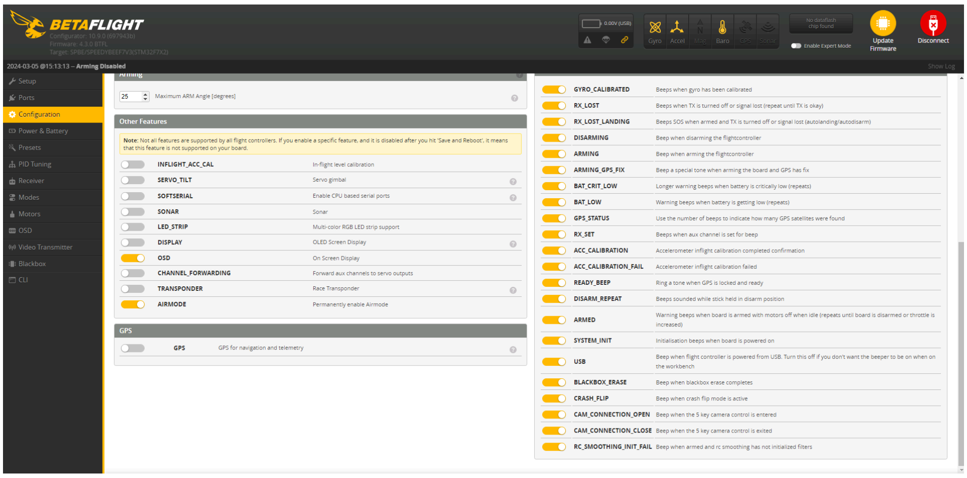
0 Accelerometer Roll Trim
0 Accelerometer Pitch Trim

Dshot Beacon Configuration

5 Beacon Tone
○ RX_LOST Beeps when TX is turned off or signal lost (repeat until TX is okay)
● RX_SET Beeps when aux channel is set for beep

Beeper Configuration

● GYRO_CALIBRATED Beeps when gyro has been calibrated
● RX_LOST Beeps when TX is turned off or signal lost (repeat until TX is okay)
● RX_LOST_LANDING Beeps SOS when armed and TX is turned off or signal lost (autolanding/autodisarm)
● DISARMING Beep when disarming the flightcontroller
● ARMING Beep when arming the flightcontroller
● ARMING_GPS_FIX Beep a special tone when arming the board and GPS has fix



Arming

25 Maximum ARM Angle [degrees]

Other Features

Note: Not all features are supported by all flight controllers. If you enable a specific feature, and it is disabled after you hit 'Save and Reboot', it means that this feature is not supported on your board.

○ INFLIGHT_ACC_CAL In-flight level calibration
○ SERVO_TILT Servo gimbal
○ SOFTSERIAL Enable CPU based serial ports
○ SONAR Sonar
○ LED_STRIP Multi-color RGB LED strip support
○ DISPLAY OLED Screen Display
● OSD On Screen Display
○ CHANNEL_FORWARDING Forward aux channels to servo outputs
○ TRANSPONDER Race Transponder
● AIRMODE Permanently enable Airmode

GPS

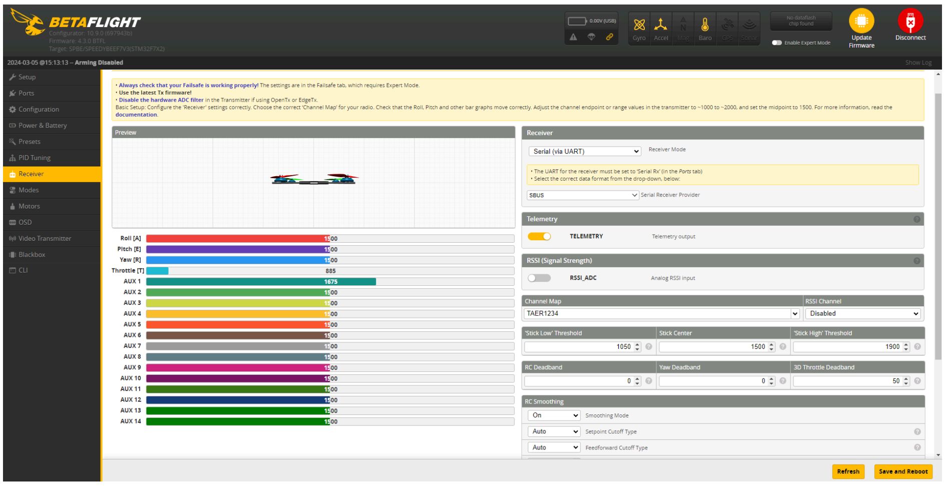
○ GPS GPS for navigation and telemetry

● GYRO_CALIBRATED Beeps when gyro has been calibrated
● RX_LOST Beeps when TX is turned off or signal lost (repeat until TX is okay)
● RX_LOST_LANDING Beeps SOS when armed and TX is turned off or signal lost (autolanding/autodisarm)
● DISARMING Beep when disarming the flightcontroller
● ARMING Beep when arming the flightcontroller
● ARMING_GPS_FIX Beep a special tone when arming the board and GPS has fix
● BAT_CRIT_LOW Longer warning beeps when battery is critically low (repeats)
● BAT_LOW Warning beeps when battery is getting low (repeats)
● GPS_STATUS Use the number of beeps to indicate how many GPS satellites were found
● RX_SET Beeps when aux channel is set for beep
● ACC_CALIBRATION Accelerometer inflight calibration completed confirmation
● ACC_CALIBRATION_FAIL Accelerometer inflight calibration failed
● READY_BEEP Ring a tone when GPS is locked and ready
● DISARM_REPEAT Beeps sounded while stick held in disarm position
● ARMED Warning beeps when board is armed with motors off when idle (repeats until board is disarmed or throttle is increased)
● SYSTEM_INIT Initialisation beeps when board is powered on
● USB Beep when flight controller is powered from USB. Turn this off if you don't want the beeper to be on when on the workbench
● BLACKBOX_ERASE Beep when blackbox erase completes
● CRASH_FLIP Beep when crash flip mode is active
● CAM_CONNECTION_OPEN Beep when the 5 key camera control is entered
● CAM_CONNECTION_CLOSE Beep when the 5 key camera control is exited
● RC_SMOOTHING_INIT_FAIL Beep when armed and rc smoothing has not initialized filters

e. **Power&Battery (https://youtu.be/Je6eFQKbPBw?si=ZXjQsn5YEWhgAprC)**
All parameters are left as default here. (Should we calibrate the battery? forgot)

f. **Receiver (https://youtu.be/vEDdk5w6YS4?si=6Z9CoixKkIVePBiz)**
Note that values in the *Preview* (left) tab might be different from one configuration to another. However, you must set the following:

i. *Receiver Mode* is set to **Serial (via UART)** to allow serial communication via UARTs.
ii. *Serial Receiver Provider* is set to **SBUS** as Agilicious uses SBUS to communicate.
iii. *Telemetry* should be set **ON** to allow telemetry transmission from FC.
iv. *Channel Map* is set to **TAER1234** as required by Agilicious.



2024-03-05 @15:13:13 – **Arming Disabled**

- **Always check that your Failsafe is working properly!** The settings are in the Failsafe tab, which requires Expert Mode.
- **Use the latest Tx firmware!**
- **Disable the hardware ADC filter** in the Transmitter if using OpenTx or EdgeTx.

Basic Setup: Configure the 'Receiver' settings correctly. Choose the correct 'Channel Map' for your radio. Check that the Roll, Pitch and other bar graphs move correctly. Adjust the channel endpoint or range values in the transmitter to ~1000 to ~2000, and set the midpoint to 1500. For more information, read the documentation.

Roll [A]	1500
Pitch [E]	1500
Yaw [R]	1500
Throttle [T]	885
AUX 1	1675
AUX 2	1500
AUX 3	1500
AUX 4	1500
AUX 5	1500
AUX 6	1500
AUX 7	1500
AUX 8	1500
AUX 9	1500
AUX 10	1500
AUX 11	1500
AUX 12	1500
AUX 13	1500
AUX 14	1500

Receiver

Serial (via UART) Receiver Mode

- The UART for the receiver must be set to 'Serial Rx' (in the *Ports* tab)
- Select the correct data format from the drop-down, below:

SBUS Serial Receiver Provider

Telemetry

TELEMETRY Telemetry output

RSSI (Signal Strength)

RSSI_ADC Analog RSSI input

Channel Map	RSSI Channel
TAER1234	Disabled

'Stick Low' Threshold	Stick Center	'Stick High' Threshold
1050	1500	1900

RC Deadband	Yaw Deadband	3D Throttle Deadband
0	0	50

RC Smoothing

On Smoothing Mode

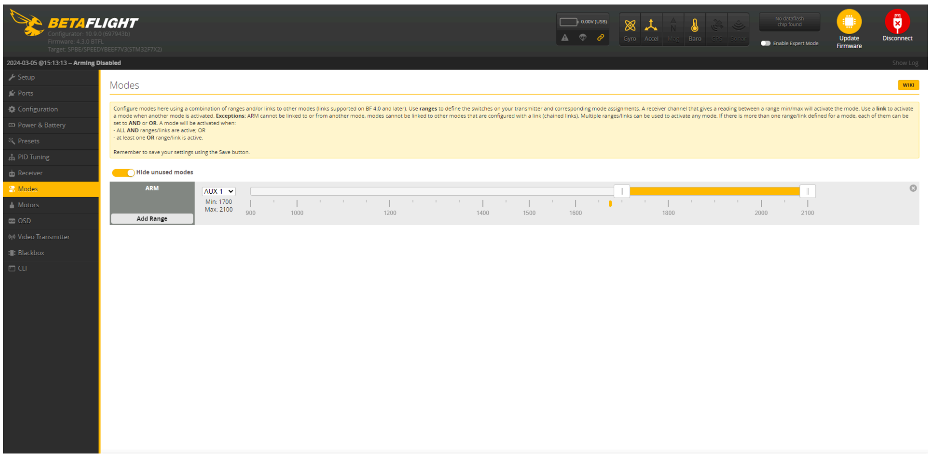
Auto Setpoint Cutoff Type

Auto Feedforward Cutoff Type

g. **Modes (https://youtu.be/kDAotpevszs?si=w6etbl4qXF82e5Yy)**

i. We bind **ARMing** to **AUX1** and set the enabling range from *1700 to 2100*. This will allow Agilicious to arm the quadrotor. When first launching autonomous flying from Agilicious, you will observe **AUX1** going to *1000* (unarmed). Then, when clicking '*Arm'* in the software, the value will rise to *2000* and rotors will start spinning (armed).

ii. Other modes can be configured here such as flying modes, beep signals…



2024-03-05 @15:13:13 – **Arming Disabled**

Modes

Configure modes here using a combination of ranges and/or links to other modes (links supported on BF 4.0 and later). Use **ranges** to define the switches on your transmitter and corresponding mode assignments. A receiver channel that gives a reading between a range min/max will activate the mode. Use a **link** to activate a mode when another mode is activated. **Exceptions:** ARM cannot be linked to or from another mode, modes cannot be linked to other modes that are configured with a link (chained links). Multiple ranges/links can be used to activate any mode. If there is more than one range/link defined for a mode, each of them can be set to **AND** or **OR**. A mode will be activated when:
- ALL **AND** ranges/links are active; OR
- at least one **OR** range/link is active.

Remember to save your settings using the Save button.

Hide unused modes

ARM — AUX 1 — Min: 1700 Max: 2100

h. **Motors (https://youtu.be/1WYDsiYJGoQ?si=QNzsBX40A-W6r9j-)**

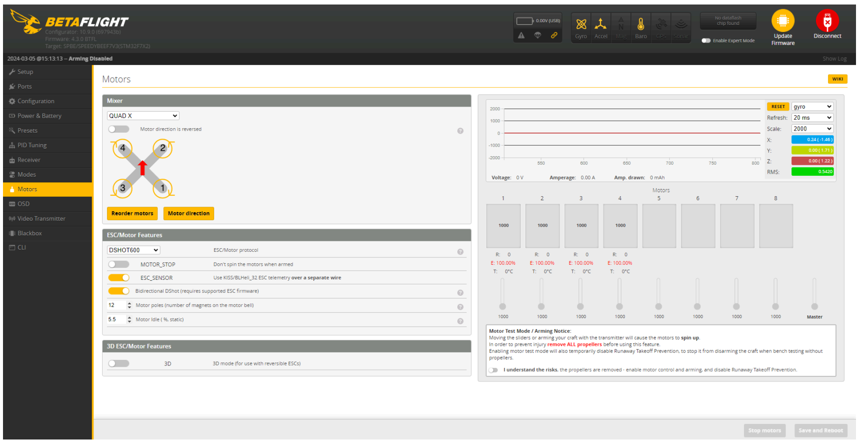
i. We recommend leaving *ESC_SENSOR* **ON**, although this may not affect autonomous flying.

ii. We set *Bidirectional DShot* to **ON** to allow bidirectional communication between the FC and ESC. That is to send commands from FC to ESC (such as Motor Commands) and information signals from ESC to FC (such as Battery Voltage).

iii. Then, you should set *ESC/Motor Protocol* to **DSHOT600**.

iv. The value of *Motor Poles* depends on the number of magnets in each motor, our motors have **12** magnets each.

v. You must configure your motors in the *Motors* section, such as setting up *spacial configuration* (here QuadX), *motor indexing* and *spinning directions*.



Motors

Mixer

QUAD X

Motor direction is reversed

ESC/Motor Features

DSHOT600 ESC/Motor protocol

MOTOR_STOP Don't spin the motors when armed

ESC_SENSOR Use KISS/BLHeli_32 ESC telemetry **over a separate wire**

Bidirectional DShot (requires supported ESC firmware)

12 Motor poles (number of magnets on the motor bell)

5.5 Motor Idle (%, static)

3D ESC/Motor Features

3D 3D mode (for use with reversible ESCs)

gyro

Refresh: 20 ms

Scale: 2000

X: 0.24 (-1.46)

Y: 0.00 (1.71)

Z: 0.00 (1.22)

RMS: 0.5420

Voltage: 0 V **Amperage:** 0.00 A **Amp. drawn:** 0 mAh

Motors

1	2	3	4	5	6	7	8	
1000	1000	1000	1000					
R: 0	R: 0	R: 0	R: 0					
E: 100.00%	E: 100.00%	E: 100.00%	E: 100.00%					
T: 0°C	T: 0°C	T: 0°C	T: 0°C					
1000	1000	1000	1000	1000	1000	1000	1000	Master

Motor Test Mode / Arming Notice:
Moving the sliders or arming your craft with the transmitter will cause the motors to **spin up**.
In order to prevent injury **remove ALL propellers** before using this feature.
Enabling motor test mode will also temporarily disable Runaway Takeoff Prevention, to stop it from disarming the craft when bench testing without propellers.

I understand the risks, the propellers are removed - enable motor control and arming, and disable Runaway Takeoff Prevention.

i. **CLI Tab**

The CLI is the command line dedicated to our FC configuration. To cope with Agilicious's setup, we must import their recommended internal configurations to our FC. To do so, we paste the below commands in our CLI. **We recommend saving your current configuration, which will be lost after importing Agilicious's parameters.**

```
set serialrx_provider = SBUS
set serialrx_inverted = ON

map TAER1234

profile 0

# profile 0
set profile_name = -
set dyn_lpf_dterm_min_hz = 70
set dyn_lpf_dterm_max_hz = 170
set dyn_lpf_dterm_curve_expo = 5
set dterm_lowpass_type = PT1
set dterm_lowpass_hz = 150
set dterm_lowpass2_type = PT1
set dterm_lowpass2_hz = 150
set dterm_notch_hz = 0
set dterm_notch_cutoff = 0
set vbat_pid_gain = OFF
set vbat_sag_compensation = 0
set pid_at_min_throttle = ON
set anti_gravity_mode = SMOOTH
set anti_gravity_threshold = 250
set anti_gravity_gain = 1000
set feedforward_transition = 0
set acc_limit_yaw = 0
set acc_limit = 0
set crash_dthreshold = 50
set crash_gthreshold = 400
set crash_setpoint_threshold = 350
set crash_time = 500
set crash_delay = 0
set crash_recovery_angle = 10
set crash_recovery_rate = 100
set crash_limit_yaw = 200
set crash_recovery = OFF
set iterm_rotation = OFF
set iterm_relax = OFF
set iterm_relax_type = SETPOINT
set iterm_relax_cutoff = 15
set iterm_windup = 100
set iterm_limit = 400
set pidsum_limit = 500
set pidsum_limit_yaw = 400
set yaw_lowpass_hz = 0
set throttle_boost = 0
set throttle_boost_cutoff = 15
set acro_trainer_angle_limit = 20
set acro_trainer_lookahead_ms = 50
set acro_trainer_debug_axis = ROLL
set acro_trainer_gain = 75
set p_pitch = 40
set i_pitch = 80
set d_pitch = 20
set f_pitch = 0
set p_roll = 40
set i_roll = 80
set d_roll = 20
set f_roll = 0
set p_yaw = 30
set i_yaw = 60
set d_yaw = 0
set f_yaw = 0
set angle_level_strength = 50
set horizon_level_strength = 50
set horizon_transition = 75
set level_limit = 55
set horizon_tilt_effect = 75
set horizon_tilt_expert_mode = OFF
set abs_control_gain = 0
set abs_control_limit = 90
set abs_control_error_limit = 20
set abs_control_cutoff = 11
set use_integrated_yaw = OFF
set integrated_yaw_relax = 200
set d_min_roll = 0
set d_min_pitch = 0
set d_min_yaw = 0
set d_min_boost_gain = 27
set d_min_advance = 20
set motor_output_limit = 100
set auto_profile_cell_count = 0
set launch_control_mode = NORMAL
set launch_trigger_allow_reset = ON
set launch_trigger_throttle_percent = 20
set launch_angle_limit = 0
set launch_control_gain = 40
set ff_interpolate_sp = AVERAGED_2
set ff_spike_limit = 60
set ff_max_rate_limit = 100
set ff_smooth_factor = 37
set ff_boost = 15
set idle_min_rpm = 0
set idle_adjustment_speed = 50
set idle_p = 50
set idle_pid_limit = 200
set idle_max_increase = 150
set level_race_mode = OFF

rateprofile 0

# rateprofile 0
set rateprofile_name = -
set thr_mid = 50
set thr_expo = 0
set rates_type = BETAFLIGHT
set roll_rc_rate = 220
set pitch_rc_rate = 220
set yaw_rc_rate = 204
set roll_expo = 0
set pitch_expo = 0
set yaw_expo = 0
set roll_srate = 0
set pitch_srate = 0
set yaw_srate = 0
set tpa_rate = 0
set tpa_breakpoint = 2000
set tpa_mode = D
set throttle_limit_type = OFF
set throttle_limit_percent = 100
set roll_rate_limit = 1998
set pitch_rate_limit = 1998
set yaw_rate_limit = 1998
save
```

10. Mocap Setup

Open Motive App
Make sure no marker are detected
Click “Start Wanding”
Do the physical calibration
When done click “Calculate”
And apply the result if “Excellent/Exceptional”
Set up the ground (initial coordinate)
Select the ground on Motive
Click “Set ground plane”
Click “Export”

Go to “View” > “Data Streaming Pane” > “Streaming” > “Show Advanced” (if not automatic)
Set “Up Axis” to “Z Up”
Enable “VRPN” transmission/streaming

In “Streaming” > “Local Interface” select an IP (whichever you want, this will be used as the *server* address later)

11. Agilicious Installation

First, we create two copies of the Agilicious repository. One should be for your local/base/host computer and the other will be on your SBC.
Use the below commands to create/clone the clean copies on each computer:

```
mkdir catkin_ws/src
cd catkin_ws/src
git clone https://github.com/catkin/catkin_simple
git clone https://github.com/ethz-asl/mav_comm.git
git clone git@github.com:uzh-rpg/agilicious.git
catkin build
```

12. Demonstration

You should now be ready to run the below demonstration on the **Tracking Arena**.
Please note that, in the demonstration, we refer to the BSC as *Babee*.

Demonstration for **Tracking Arena**:

1) Agilicious codebase configuration:

a) On Babee:

- In *onboard_pilot_betaflight.yaml* change *quadrotor* to *kingfisher*, if not by default.
- In the file responsible for sending commands from NX->FC (here, *sbus.yaml*), change the TTY address to the one of the TX pin of the SBC (here, */dev/ttyUSB0*).
- In the file responsible for receiving telemetry data from FC->NX (here, *betaflight.launch*), change the TTY address to the one of the RX pin of the SBC (here, */dev/ttyTHS0*).

b) On Base Computer:

- In *arena_basecomputer_onboard.launch* change the *server* address to the one of your Mocap system (for us, 10.206.0.244).
  By default, this will be: <param name="server" value="192.168.200.119"/>.

2) Setup before Launching:

Before starting this setup, you must decide on which, Base Computer (BC) or Babee, you wish to set as your *MASTER*. This doesn't matter and can be either. We use BC.

a) On Base Computer:

- Run the following commands in your BC terminal:

```
cd catkin_ws/src/agilicious_internal/agiros/agiros/launch/tracking_arena
source /opt/ros/noetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://<MASTER_IP>:11311
export ROS_IP=<BC_IP>
```

b) On Babee:

- Run the following commands in your Babee terminal:

```
ssh@<Babee_IP>
source /opt/ros/noetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://<MASTER_IP>:11311
export ROS_IP=<Babee_IP>
sudo systemctl stop serial-getty@ttyTHS0.service serial-getty@ttyUSB0.service
sudo chmod a+rw /dev/ttyTHS* /dev/ttyUSB*
sudo usermod -a -G dialout $USER
```

3) Launching:

a) On Base Computer:

- Launch the following file to start Autonomous flying:

```
roslaunch arena_base_computer_onboard.launch quad_name:="kingfisher"
```

- This file will run 3 nodes:
  1. A **GUI** (to Connect, Arm and Hover) - This GUI is an alternative to RC.
  2. A **VRPN Client** (vrpn_client_ros) to get odometry data from the mocap.
  3. A **Rviz** instance to visualize everything.

b) <u>On Babee:</u>

- Launch the following file to start Autonomous flying:

```
roslaunch arena_quadrotor_onboard_betaflight.launch quad_name:='kingfisher'
```

- This file will run 2 nodes:
  1. A **Mavros** (betaflight) instance.
  2. A **Controller** - which is either an *MPC* or *PID,* based on what is set in your configuration file in Agilicious codebase. By default *Tracking Arena* uses **MPC**.

c) <u>On the GUI (from Base Computer):</u>

- If everything has been set up properly, when pressing *Connect* you should now see the *Battery Voltage* (indicating that FC-to-NX communication is working) and *Odometry Data* (indicating that data was successfully read from the Mocap system). Note that the *Battery Voltage* is now coming from the real hardware, so you must make sure that this value is correct (**tips**: if it is 15.5V, the telemetry communication probably failed).
- When clicking *Arm* motors should now be rotating (indicating that NX-to-FC communication is also working).
- When clicking Start, the motors should start spinning rapidly.
  Important: Make sure that the Stop button works before using the quadrotor with propellers on.