Table of Contents

Introduction

SDK Documentation Home

This document helps you get started with the various aspects of building SDK application and describes the protocol which can be used by software to control the Gremsy's gimbal.
The gimbal can be controlled using serial via the COM2 connector.

General Protocol Overview

The API is accessed via serial massages in the Mavlink format (see https://mavlink.io/en/guide/mavlink_version.html). Mavlink provides an open data format for interaction as well as a suite if tools to assist the programmer in developing and testing interface.

Gimbal uses Mavlink v2.0 message and communicates with the host at 115200 baudrate and 8N1.

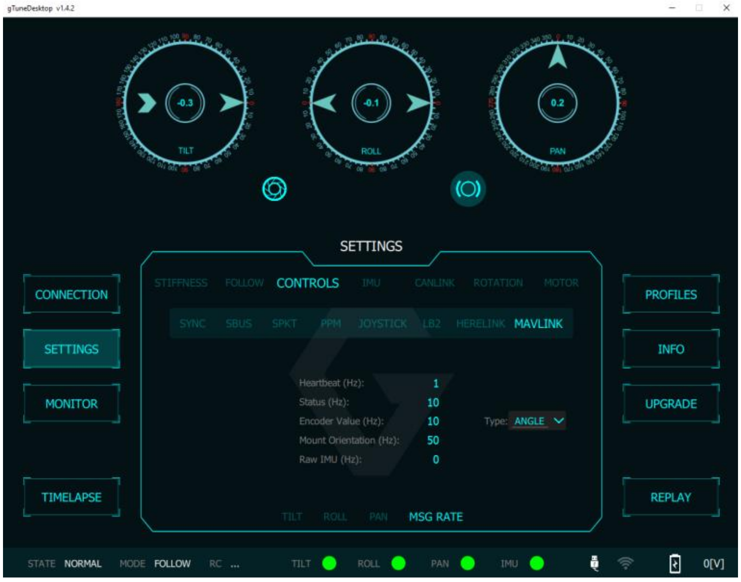
NOTE:

- SDK_V2 has just been released for the gimbal's firmware and gTuneDesktop app.
- **Firmware:**
  - T3V2: T3V2_v7.0.0 or above
  - S1V2: S1V2_v7.0.0 or above
  - Pixy: Pixy_v7.0.0 or above
  - T3V3: GremsyT3V3_v750_Official or above
  - S1V3: GremsyS1V3_v750_Official or above
- **gTuneDesktop:**
  - gTuneDesktop_v1.4.2 or above
- **Link:** https://github.com/Gremsy

Enable SDK API

Baud & Data Transmission Rates

- The APIs can set Data Transmission Rates for each message.
- Update your gimbal with the latest released firmware (from v750) and gTune software version 1.4.2 or above.
- Connect your computer to the Micro-USB port on the Gimbal and open gtune software



Heartbeat (Hz): 1
Status (Hz): 10
Encoder Value (Hz): 10 Type: ANGLE
Mount Orientation (Hz): 50
Raw IMU (Hz): 0

STATE NORMAL MODE FOLLOW RC ... TILT ● ROLL ● PAN ● IMU ● 0[V]

HEARTBEAT (Message ID: #0)

The Gimbal will send a heartbeart message approximately once per second using the component ID MAV_COMP_ID_GIMBAL. It sends through COM2 on the QR after powering on Gimbal. Taking advantage of this message to check the connection between your devices and gimbal.

SYS_STATUS (Message ID: #1)

The gimbal pushes state information back through sys_status at up to 50 Hz (default 10Hz). State information includes the working modes, sensor’s states, motor’s states.

See gimbal_status_t structure

RAW_IMU (Message ID: #27)

Gimbal provides the raw imu. Supporting for application want to combine and calculate orientation.

Acceleration (raw). For a range of 4g = 8192

Angular speed around X/Y/Z axis (raw) - degrees per second

Field Name	Type	Official Description	Gimbal Implementation
time_usec	uint64_t	Timestamp	Same as official
xacc	int16_t	X acceleration (raw)	Same as official
yacc	int16_t	Y acceleration (raw)	Same as official
zacc	int16_t	Z acceleration (raw)	Same as official

xgyro	int16_t	Angular speed around X axis	Same as official
ygyro	int16_t	Angular speed around Y axis	Same as official
zgyro	int16_t	Angular speed around Z axis	Same as official
xmag	int16_t	X Magnetic field	Ignored
ymag	int16_t	Y Magnetic field	Ignored
zmag	int16_t	Z Magnetic field	Ignored

MOUNT_STATUS (Message ID: #158)

This message can be used to provide the raw encoder values or encoder angle values of the gimbal.

The encoder resolution: 2^16 bits.

Field Name	Type	Official Description	Gimbal Implementation
pointing_a	int32_t	Pitch (deg*100)	Pitch encoder value
pointing_a	int32_t	Roll (deg*100)	Roll encoder value
pointing_a	int32_t	Yaw (deg*100)	Pan encoder value
Target System	uint8_t	System ID	Onboard system ID
Target Component	uint8_t	Component ID	Onboard component ID

MOUNT_ORIENTATION (Message ID: #265)

This message can be used to provide the orientation of the gimbal. Using this message to check as feedback from gimbal when moving gimbal in angle or speed mode.

Field Name	Type	Official Description	Gimbal Implementation
time_boot_ms	int32_t	ms (Timestamp)	Same as official
roll	float	Roll in global frame	Same as official
pitch	float	Pitch in global frame	Same as official
yaw	float	Yaw relative to vehicle	Yaw encoder angle
yaw_absolute**	float	Yaw in absolute frame	IMU's angle of yaw axis.

Gimbal Control Messages

COMMAND_LONG (Message ID: #76)

The three main commands to use mavlink gimbals are: (See gSDK)

- **MAV_CMD_DO_MOUNT_CONFIGURE.**

This message supports seting gimbal modes. The gimbal has serveral work modes that combined with axis mode.

Axis mode (see control_gimbal_axis_input_mode_t)	Pitch	Roll	Yaw
CTRL_ANGLE_BODY_FRAME	Body	N/A	Body
CTRL_ANGULAR_RATE	Ground	Ground	Ground
CTRL_ANGLE_ABSOLUTE_FRAME	Ground	Ground	Ground

NOTE: Only Gimbal Pixy and T3V3 support CTRL_ANGLE_BODY_FRAME mode with Pitch and Yaw axis.

The following is an illustration the gimbal work modes.

Custom mode.	**Description**
Yaw Follow Mode Pitch and roll are controllable. Yaw will follow the aircraft.	**Pitch and Roll set:** CTRL_ANGLE_ABSOLUTE_FRAME **Pan set:** CTRL_ANGLE_BODY_FRAME
Free Mode. Pitch, Roll and yaw are all controllable, meaning the gimbal can move independently of the aircraft heading. In this mode, even if the aircraft yaw changes, the gimbal will continue pointing in the same world direction.	**Pitch, Roll, and Pan set:** CTRL_ANGLE_ABSOLUTE_FRAME
First Person View Mode. Only Pitch is controllable. Yaw and Roll will be fixed relative to the product while pitch remains controllable.	**Pitch set:** CTRL_ANGLE_ABSOLUTE_FRAME **Roll set:** CTRL_ANGLE_ABSOLUTE_FRAME **Pan set:** CTRL_ANGLE_BODY_FRAME

The gimbal can be moved in two ways:

- Control gimbal angle
- Control gimbal speed

When using angle mode to rotate the gimbal's pitch, roll, yaw, the rotation angle of the gimbal can be defined as either **Absolute** (relative to the aircraft heading "ABSOLUTE FRAME") or **Relative** (relative to it's current angle "BODY FRAME")

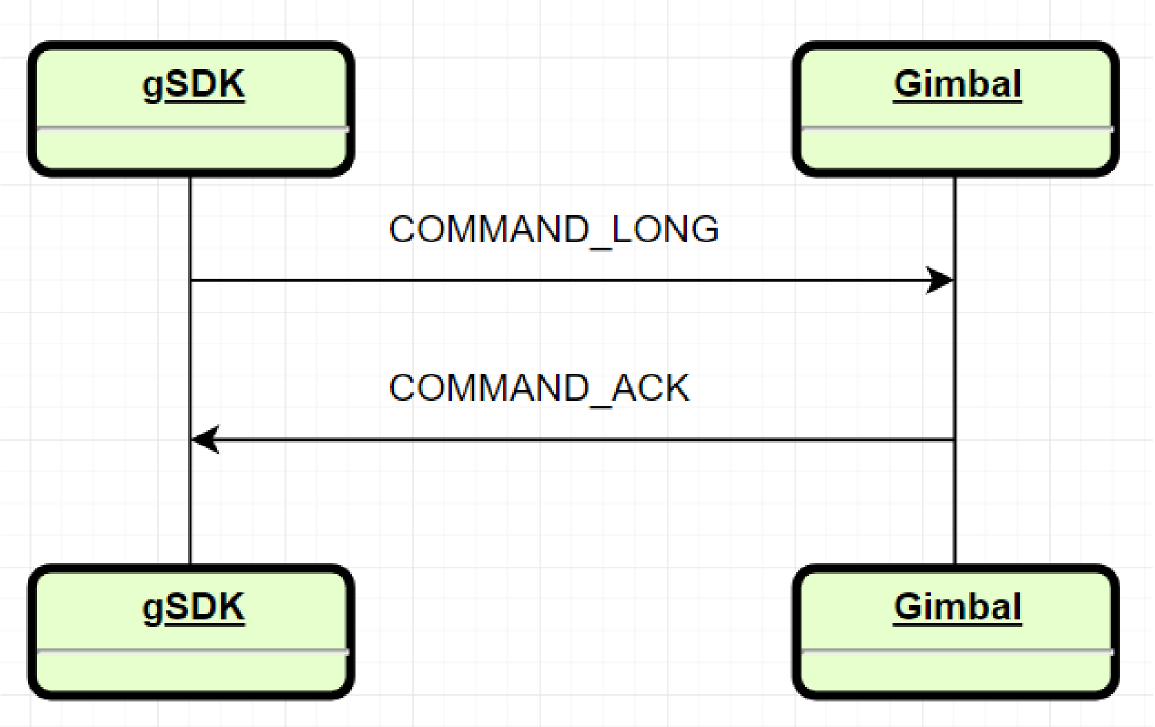
- **MAV_CMD_DO_MOUNT_CONTROL.**
  Param #1 contains desired pitch in degrees or degree/second. It depends on your setting mode above.
  Param #2 contains desired roll in degrees or degree/second. It depends on your setting mode above.
  Param #3 contains desired yaw in degrees or degree/second but note that the yaw is in relation to the front of the vehicle so "0" is straight ahead, "90" is to right. "-90" is to the left.

NOTE: All gimbals have mechanical limits (or stops) to their rotation around each axes. Input range allow control:

Gimbal	Pitch	Roll	Pan
T3V2	[-40 : +90]	[-40 : +40]	360-continuous pan
S1	[-40 : +90]	[-40 : +40]	[-180 : +180]
PixyU	[-40 : +90]	[-40 : +40]	[-300 : +300]
T3V3	[-40 : +90]	[-40 : +40]	[-300 : +300]

COMMAND_ACK (Message ID: #77)

The MAVLink command protocol allows guaranteed delivery of MAVLink commands (DO_MOUNT_CONTROL and DO_MOUNT_CONFIGURE).



The command DO_MOUNT_CONTROL may not complete immediately, in which case the gimbal can report its progress by sending COMMAND_ACK messages with COMMAND_ACK.result = MAV_RESULT_IN_PROGRESS.

- **MAV_CMD_PREFLIGHT_REBOOT_SHUTDOWN.**

  Param4: 1 to reboot gimbal

What It Does

Gimbal supports following functions.

- **Gimbal status (see gimbal_status_t structure)**. Status related to gimbal. Gimbal will send a status message approximately once per 100 ms by default.
- **The gimbal orientation (see gimbal_mount_t structure)**. Gimbal will send the orientation message approximately once per 20 ms by default.

  NOTE: the yaw_absolute field is a yaw angle in earth frame. It does not support now
- **The gimbal encoder values**. This is raw encoder values or angle after calculating. It depends on your setting the Data Transmission above. Gimbal will send encoder values approximately once per 20 ms by default.
- **The gimbal raw imu values**. Gimbal will send the raw imu values approximately once per 20 ms.
- **The HEARBEAT message** can ensure that the Gimbal is booted and ready to receive Mavlink Packets. Gimbal will send a heartbeat message approximately once per second by default.
- **The COMMAND_ACK message** the feedback from gimbal to indicate command arrival.

SDK_v2 supports following functions.

- **Control gimbal for each axis mode (see structure control_gimbal_axis_mode_t)**
- **Control gimbal motor (ON/OFF).**
- **Set gimbal move**. This function will be combined with control gimbal for each axis mode. Allow control gimbal move in angle body or angular rate.
- **Set gimbal reboot**. This function will allow reboot gimbal.

These functions are supported from the firmware version from v7.5.0 or above

- **Set gimbal motor control**: For gimbal with adjustable payload, finer level motor control is useful to compensate for payload with different masses. See **set_gimbal_motor_control() function:**

Field Name	Type	Description	Range
stiffness	uint8_t	This setting adjusts the degrees to which the gimbal tries to correct for unwanted camera movement and hold the camera stable	It depends on your payload

holdstrength	uint8_t	The power level provided to the corresponding axis	Default: 40
gyro_filter	uint8_t	Defines the strength of the filter applied to Gyro sensor output.	Default: 2
output_filter	uint8_t	Defines the strength of the filter applied to motors output.	Default: 3
gain	uint8_t	Defines how fast each axis will return to commanded position.	Defaut: 120

NOTE: See user_manual to tune some parameters for the best performance. Using these parameters for your payload profile.

- **Set gimbal config axis:** User can adjust some parameters related to control in lock or follow. (**Note**: See **gimbal_config_axis_t** structure or **set_gimbal_config_pan_axis(), set_gimbal_config_roll_axis(), set_gimbal_config_tilt_axis().**
- **Set gimbal config mavlink message rate:** Configuration the message mavink with rate. **(NOTE:** see **set_gimbal_config_mavlink_msg() function)**

Field Name	**Type**	**Description**	**Range**
emit_heatbeat	int8_t	Enable or disable emit heartbeat from gimbal to COM2 port.	0: Disable and 1Hz Default: 1Hz
status_rate	uint8_t	Providing gimbal status	[0: 100] Hz. Default 10Hz
henc_value_rate	uint8_t	Rate for sending the encoder values. The type of message it depends on the enc_type_send (Raw/Angle)	[0: 100] Hz. Default 50Hz
enc_type_send	uint8_t	0: Encoder angle 1: Encoder value (Resolution 2^16)	[0:1] Default: 0
orientation_rate	uint8_t	Rate for sending the orientation of the gimbal	[0: 100] Hz Default: 50Hz
imu_rate	uint8_t	Rate for sending the raw imu values	[0:100] Hz Default: 0 (Disable)

Development Workflow

Prerequisites

To build an SDK based application the following are required

- Programming experience C/C++
- A compatible with gimbal (T3, S1, Pixy)
- Your own Onboard Computer with an available TTL UART port
- Software tool to build SDK
- PC to run the required software tool

Hardware setup guide

This guide will help you connect your Onboard Computer with the Gimbal (T3, S1, PixyF, PixyU).

Data

The onboard computer communicates to the Gimbal through a UART interface.

Onboard Computer
UART
GIMBAL

Power

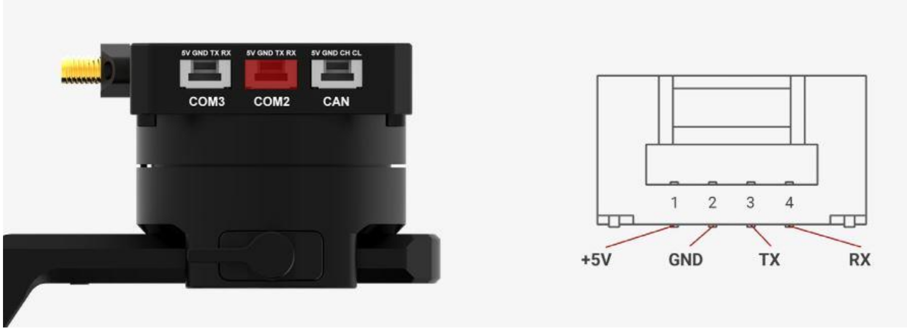
Power can be drawn directly from the COM2 port (1A max@5V).

UART

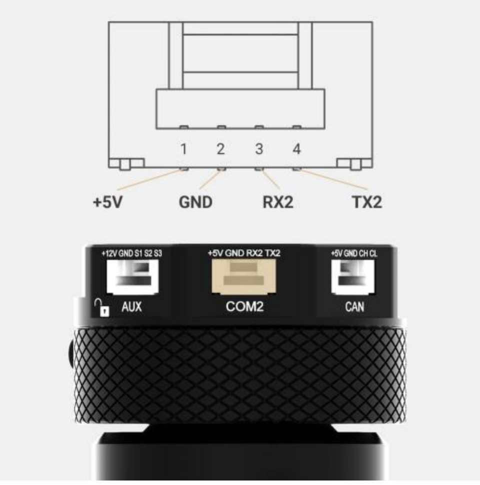
Interface Details

- The UART electrical interface for all SDK compatible Gremsy Gimbal is 3.3 volt TTL
- You must ensure that your onboard computer UART port operates at the same voltage to avoid damging the Gimbal Controller. For example, RS-232 ports will need a level-shifting circuit
- The UART interface does not require power from the onboard computer

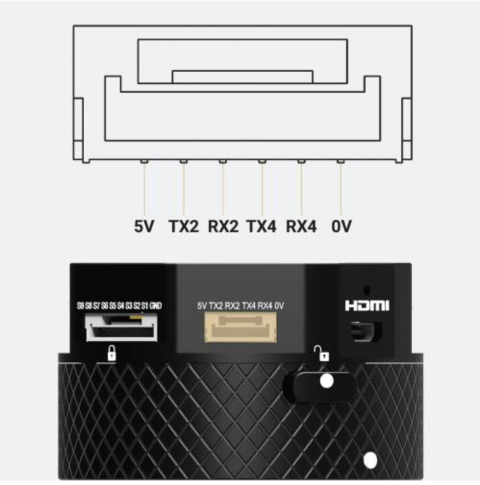
Connector Pinout S1



Connector Pinout T3

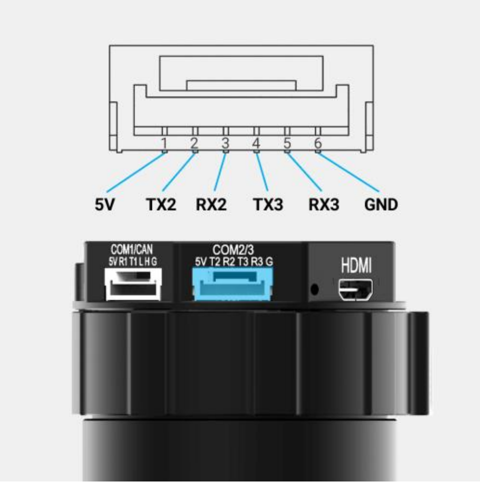


T3V2_QR



T3V3_QR

Connector Pinout PIXY



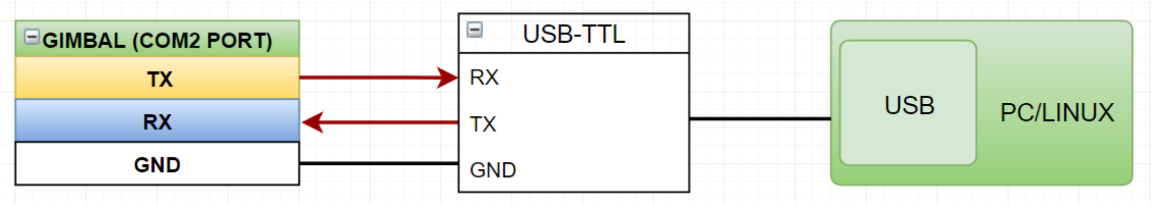
Connecting to your Onboard Computer

❖ Gimbal + PC/LINUX machine

The diagram below shows the hardware connect between an GIMBAL and a PC or LINUX machine. Note that:

- The recommended choice of USB to TTL cable is FT232 module.

- Baurate: 115200
- Bit data: 8 bits
- Parity: No Parity



Sortware Enviroment Setup Guide

This guide details the software environment needed to work with the SDK

Download the gSDK and Required tools

❖ Installing gTune app www.gremsy.com → Support → Product Support → GremsyS1 (S1, T3, Pixy)

Update Firmware

Here's what you need to update your gimbal PixyF, S1V2 and T3V2 from firmware version 6.x.x to 7.x.x.

NOTE:

- With T3V2 is slightly different. You need to remove your camera and balance without payload. But with PixyF and S1 must be conducted with camera and balanced carefully.
- Firmware V7.x.x or above must be flash with the gTuneDesktop V1.4.0 or above.

Prerequisites

- **Remove your payload on the T3V2 and balance Roll Axis**

  When the proper left-right roll balanced is achieved, the roll axis will stay level when you remove your hands.

- **Balancing Gimbal S1V2 and PixyF**

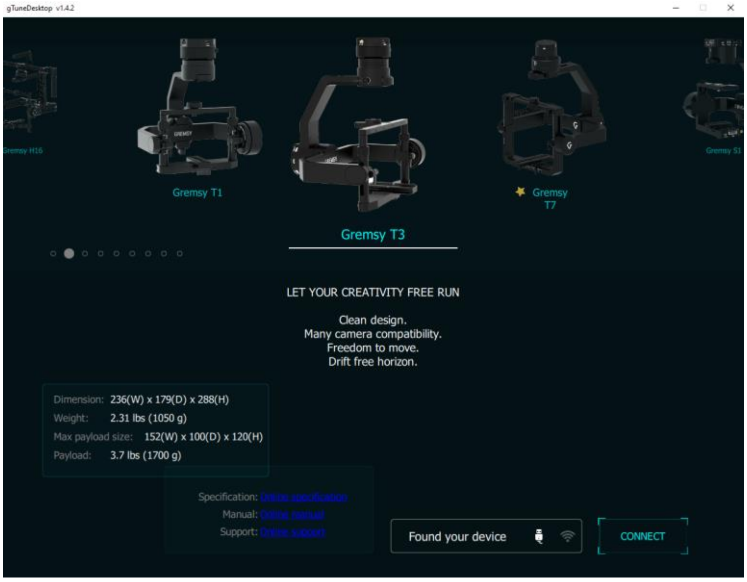
  A payload needs to be properly balanced before starting gimbal calibration. For gimbal without adjustable payload like PixyF, then move to next step.

  Refer to S1 manual → balancing: https://gremsy.com/gremsy-s1-manual/

Upgrading Firmware

- **Uninstall the old gtune on your PC/MAC and install the new gTuneDesktop v1.4.2**

  Link: https://github.com/Gremsy/gTuneDesktop/releases



LET YOUR CREATIVITY FREE RUN

Clean design.
Many camera compatibility.
Freedom to move.
Drift free horizon.

Dimension: 236(W) x 179(D) x 288(H)
Weight: 2.31 lbs (1050 g)
Max payload size: 152(W) x 100(D) x 120(H)
Payload: 3.7 lbs (1700 g)

Specification: Online specification
Manual: Online manual
Support: Online support

- **Download the firmware v7.5.0 or above**

  Link: https://github.com/Gremsy

- **Open gTuneDesktop v1.4.2 and update the firmware.**

  01 - Power on the GIMBAL.

02 - Connect USB cable from GIMBAL controller to Mac/PC.

03 - Run the gTune Desktop v1.4.2 or above software.

04 - In the software, select “Serial” option on “connection” tab.

05 - Select the port in the list.

06 - Click on the “Connect” button.

07 - Go to “Upgrade” tab.

08 - “Browse” to firmware file from your computer.

09 - Make sure the RF receiver (if available) is already removed.

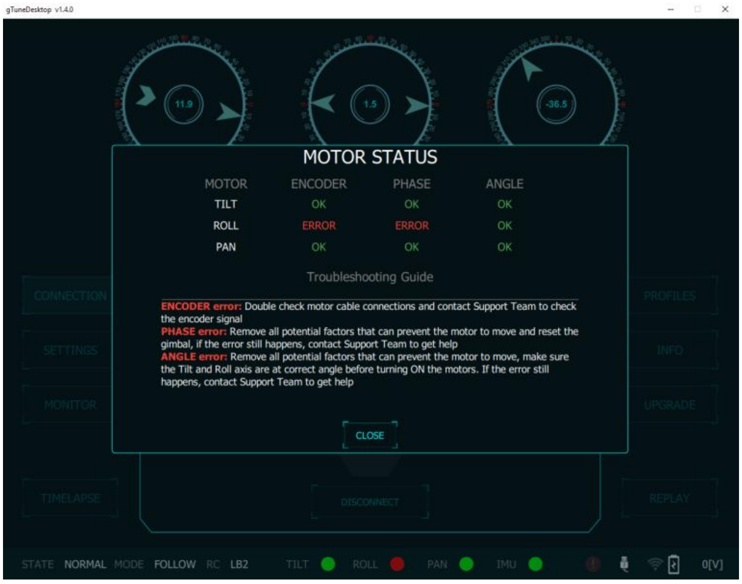
10 - Click “Upgrade” button. The process will take about 2 minutes. When the firmware is upgraded successfully, the Gimbal will be restarted automatically.

- **Calibration.**

Gimbal will be automatically calibrated on power up at the first time upgrading from the firmware version less than v7.2.0. User will see the camera rotate and shake around all three axes for serveral minutes when the gimbal is calibrating.

The gimbal should be stationary (not flying, or being held) and horizontal during calibration. For gimbal with adjustable payload, the payload should be present and balanced before doing a calibration (**NOTE: Exception for Gimbal T3 need to calibrate without Payload**).

In case, gTuneDesktop pops up motor status like below. Please, close it and go to **SETTINGS → MOTOR → CALIB MOTOR .**



MOTOR STATUS

MOTOR	ENCODER	PHASE	ANGLE
TILT	OK	OK	OK
ROLL	ERROR	ERROR	OK
PAN	OK	OK	OK

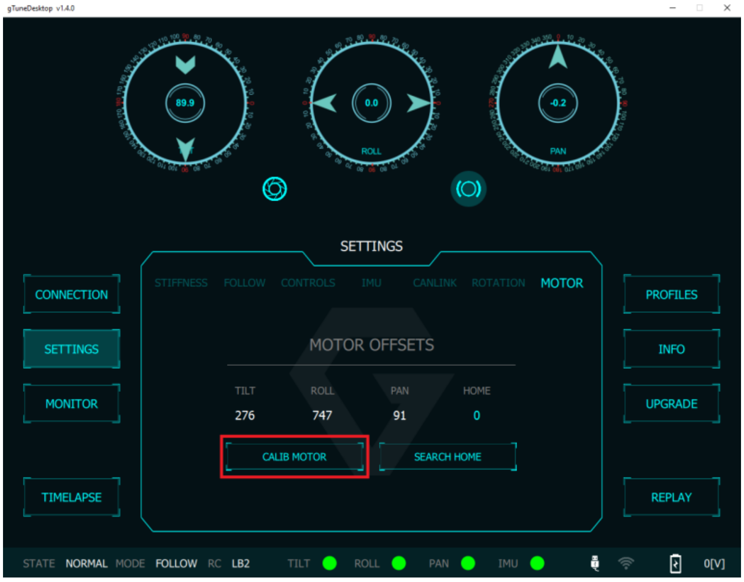
Troubleshooting Guide

ENCODER error: Double check motor cable connections and contact Support Team to check the encoder signal

PHASE error: Remove all potential factors that can prevent the motor to move and reset the gimbal, if the error still happens, contact Support Team to get help

ANGLE error: Remove all potential factors that can prevent the motor to move, make sure the Tilt and Roll axis are at correct angle before turning ON the motors. If the error still happens, contact Support Team to get help

Press CALIB MOTOR and wait a second



TILT	ROLL	PAN	HOME
276	747	91	0

STATE NORMAL MODE FOLLOW RC LB2 TILT ● ROLL ● PAN ● IMU ●

Confiure Linux Development Environment

Install Development Tools

To build standalone Linux applications based on the gSDK, you need:

- A supported C++ compiler
- A bash shell
- A modern Linux distribution

Add UART Permissions

Please follow the steps below to add UART read and write permissions for users specified in Linux:

- Use the `sudo usermod -a -G dialout $ USER` command to add the user to the `dialout` group.
- After logging in to the added account again, the account can obtain UART read and write permissions.

Setting up samples

Before you start

1. Make sure you have followed the steps in the **Hardware Setup Guide** to get your connection right.
2. Follow the steps in the Enviroment Setup guide to get your software ready to run samples.

Run The Sample On The Linux

This is a simple MAVLink to UART interface example for Linux systems that can allow communication between gimbal and an Onboard Computer.

This example will receive Mavlink messages from gimbal and send MAVLink messages for controlling and setting gimbal.

Building the gSDK and running example

1. Clone (or download as zip) the gSDK

   https://github.com/Gremsy/gSDK.git

2. Open a terminal, cd into the gSDK folder and follow these steps to build the gSDK:

   ```
   $ cd gSDK
   $ make
   ```

Execution

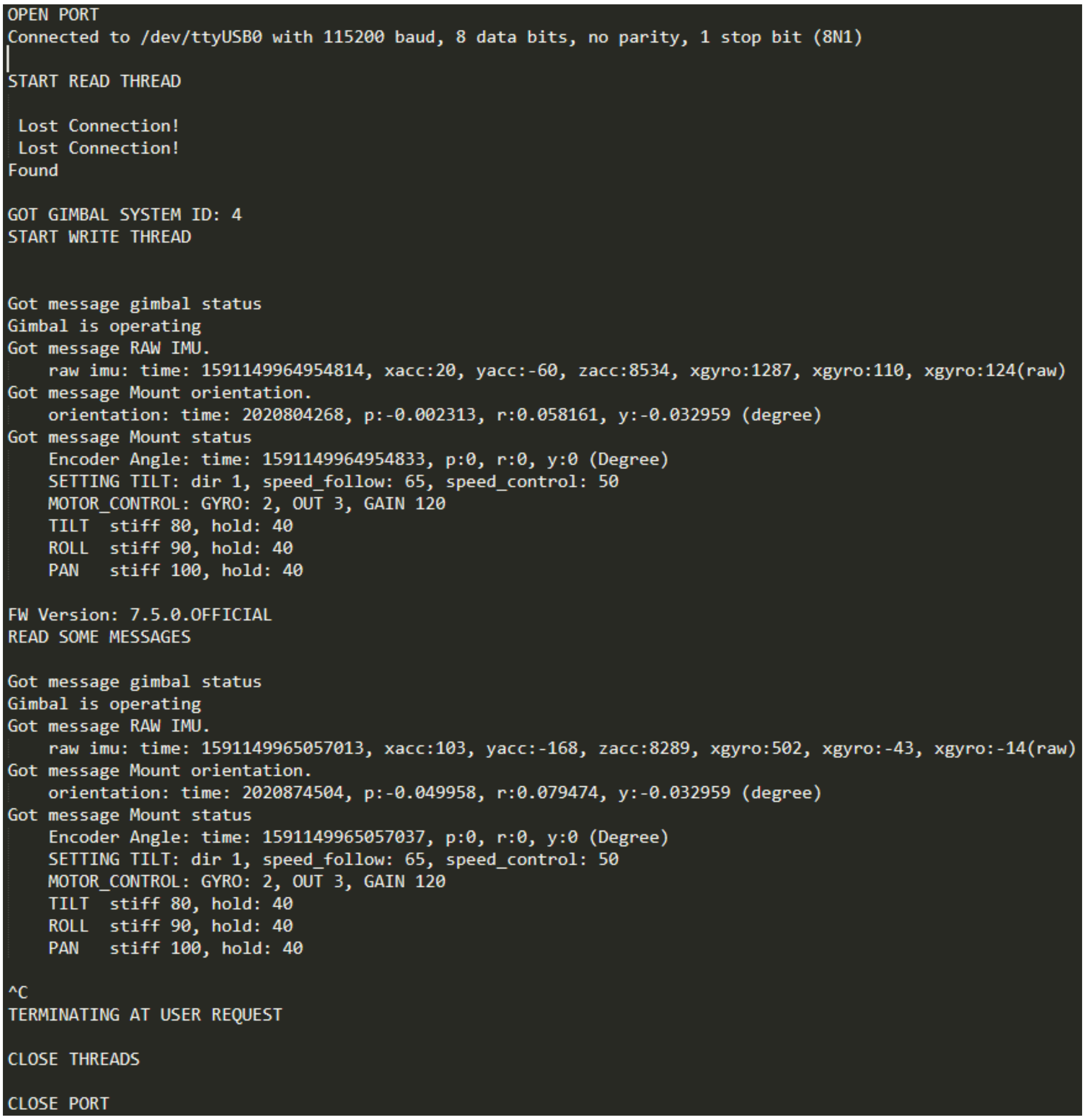
You have to pick a port name, try searching for it with

```
$ ls /dev/ttyACM*
$ ls /dev/ttyUSB*
```

Run the example executable on the host shell:

```
$ cd gSDK
$ ./gSDK –d /dev/ttyUSB0
```

To stop the program, use the key sequence Ctrl-C



```
OPEN PORT
Connected to /dev/ttyUSB0 with 115200 baud, 8 data bits, no parity, 1 stop bit (8N1)

START READ THREAD

 Lost Connection!
 Lost Connection!
Found

GOT GIMBAL SYSTEM ID: 4
START WRITE THREAD


Got message gimbal status
Gimbal is operating
Got message RAW IMU.
    raw imu: time: 1591149964954814, xacc:20, yacc:-60, zacc:8534, xgyro:1287, xgyro:110, xgyro:124(raw)
Got message Mount orientation.
    orientation: time: 2020804268, p:-0.002313, r:0.058161, y:-0.032959 (degree)
Got message Mount status
    Encoder Angle: time: 1591149964954833, p:0, r:0, y:0 (Degree)
    SETTING TILT: dir 1, speed_follow: 65, speed_control: 50
    MOTOR_CONTROL: GYRO: 2, OUT 3, GAIN 120
    TILT  stiff 80, hold: 40
    ROLL  stiff 90, hold: 40
    PAN   stiff 100, hold: 40

FW Version: 7.5.0.OFFICIAL
READ SOME MESSAGES

Got message gimbal status
Gimbal is operating
Got message RAW IMU.
    raw imu: time: 1591149965057013, xacc:103, yacc:-168, zacc:8289, xgyro:502, xgyro:-43, xgyro:-14(raw)
Got message Mount orientation.
    orientation: time: 2020874504, p:-0.049958, r:0.079474, y:-0.032959 (degree)
Got message Mount status
    Encoder Angle: time: 1591149965057037, p:0, r:0, y:0 (Degree)
    SETTING TILT: dir 1, speed_follow: 65, speed_control: 50
    MOTOR_CONTROL: GYRO: 2, OUT 3, GAIN 120
    TILT  stiff 80, hold: 40
    ROLL  stiff 90, hold: 40
    PAN   stiff 100, hold: 40

^C
TERMINATING AT USER REQUEST

CLOSE THREADS

CLOSE PORT
```