Table of Contents

Introduction........3
SDK Documentation Home........3
General Protocol Overview........3
Messages........3
HEARTBEAT (#0)........3
SYS_STATUS (#1)........4
RAW_IMU (#27)........4
MOUNT_STATUS (#158)........4
MOUNT_ORIENTATION (#265)........5
COMMAND_LONG (#76)........5
What’s It Does........8
Development Workflow........9
Prerequisites........9
Hardware setup guide........9
Data........10
Power........10
UART........10
Sortware Enviroment Setup Guide........12
Download the gSDK_STM32 and Required tools........12
Update Firmware........13
STM32........16
Ubuntu Linux........17
Setting up samples........17
Before you start........17

Linux Onboard Computer 18

Building the gSDK and running example 18

Execution 18

STM32 Onboard Computer 20

Installing and setting up the gSDK_STM32 20

Building and Running the sample 20

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 maessages 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 have just been released for the gimbal's firmware and gTuneDesktop app.
- Firmware version_v7.0.0 or above must be flashed with gTuneDesktop_v1.4.0 or above.
  - T3V2: T3V2_v7.0.0 or above
  - S1V2: S1V2_v7.0.0 or above
  - Pixy: Pixy_v7.0.0 or above
  - gTuneDesktop: gTuneDesktop_v1.4.0.0 or above
    (see https://github.com/Gremsy/gTuneDesktop/releases)
- gSDK_Linux: see https://github.com/gChuNguyen/gSDK_Linux

Messages

HEARTBEAT (#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 (#1)

The gimbal pushes state information back through sys_status at up to 10 Hz. State information includes the work mode, sensor state, motor state.

See gimbal_status_t structure

RAW_IMU (#27)

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

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 (#158)

This message can be used to provide the encoder 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 (#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	[deg] Roll in global frame	Same as official
pitch	float	[deg] Pitch in global frame	Same as official
yaw	float	[deg] Yaw relative to vehicle	Yaw encoder angle
yaw_absolute**	float	[deg] Yaw in absolute frame	IMU's angle of yaw axis.

COMMAND_LONG (#76)

The two 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:

- **T3V2 only supports the CTRL_ANGLE_BODY_FRAME mode for 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:

- Move to an angle
- Move at a 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.

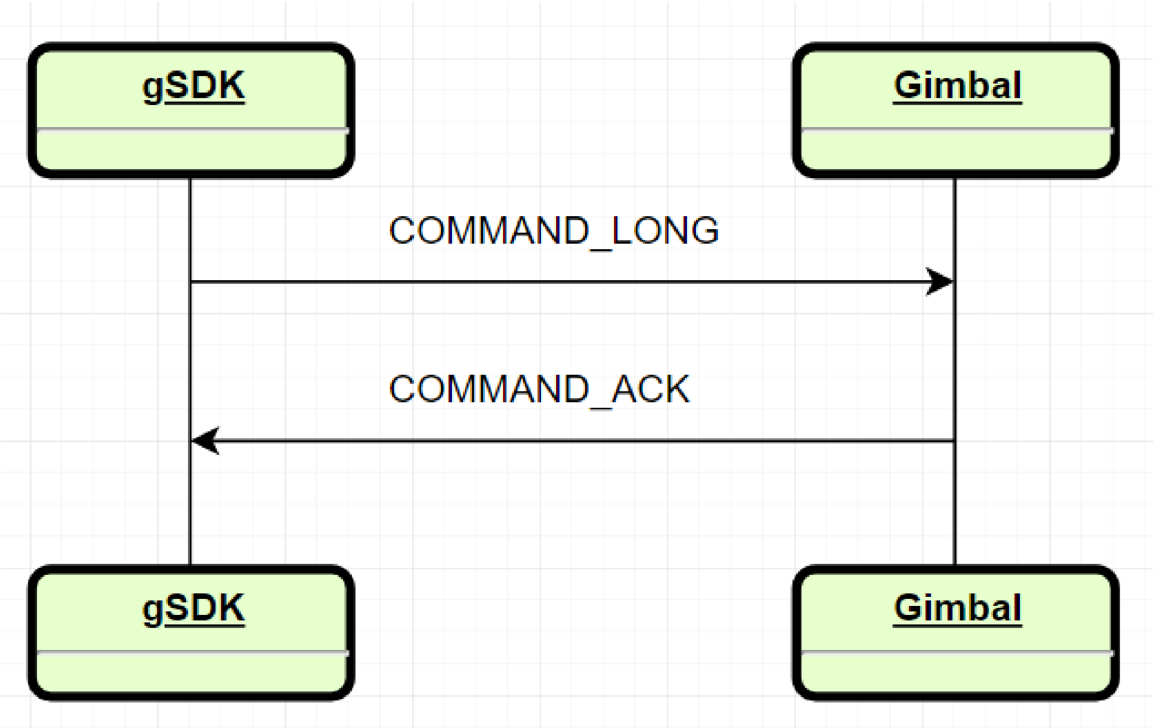
Param #3 contains desired yaw in degrees or degree/second but note that the yaw is in relation to the front of the vehicle so “0” is straight ahead, “90” is to right. “-90” is to the left.

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 (#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.

What's 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.
- **The gimbal orientation (see gimbal_mount_t structure)**. Gimbal will send the orientation message approximately once per 20 ms.

  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. Gimbal will send encoder values approximately once per 20 ms.
- **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.

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.

Development Workflow

Prerequisites

To build an SDK based application the following are required

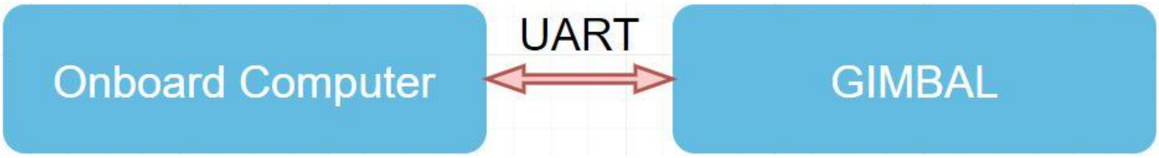
- Programming experience C/C++
- A compatible with gimbal (T3, S1, Pixy)
- Your own Onboard Computer with an available TTL UART port
  - We recommend the STM32F4 Discovery
- Software tool to build SDK
- Windown PC to run the required software tool

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.



Power

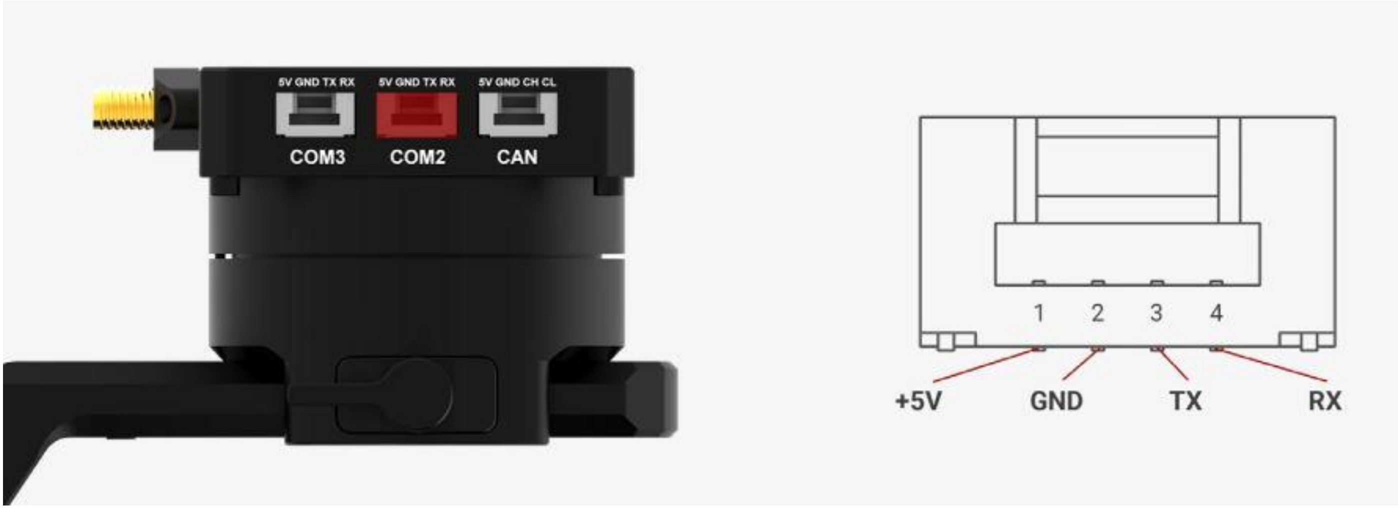
Power can be drawn directly from the COM2 port (1A max@5V).

UART

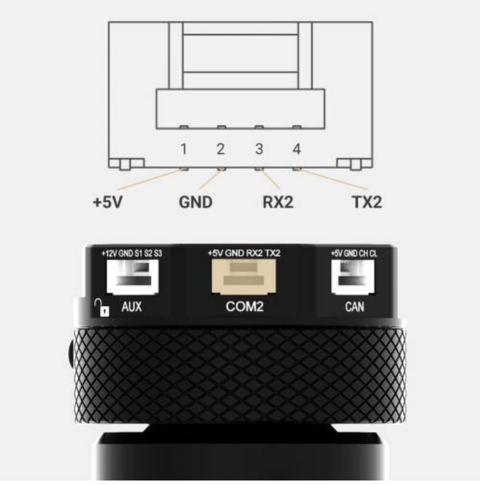
Interface Details

- 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

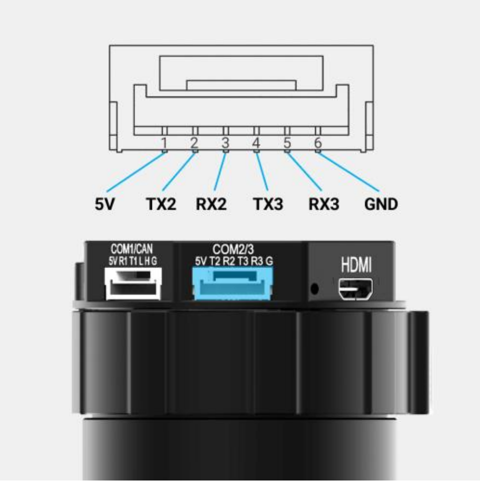
Connector Pinout S1



Connector Pinout T3



Connector Pinout PIXY

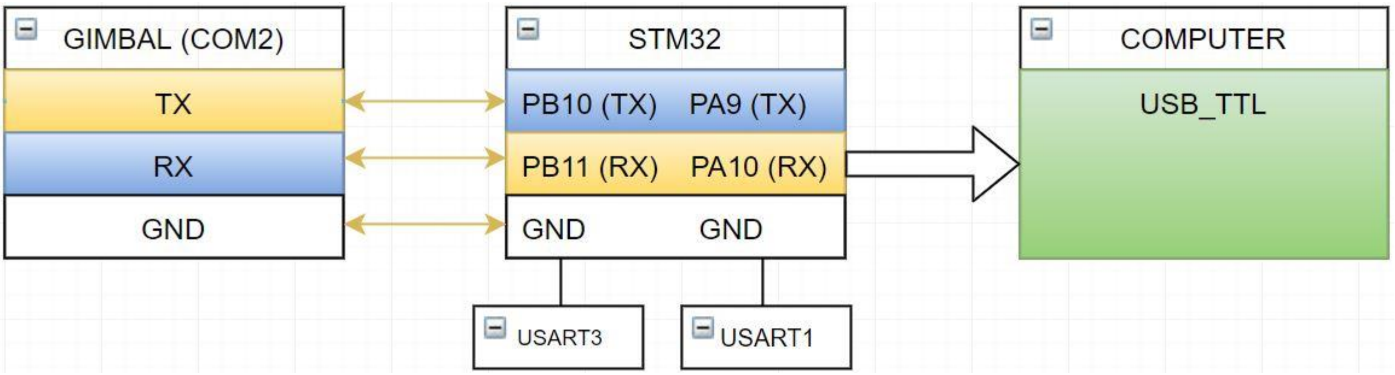


Connecting to your Onboard Computer

❖ Gimbal + STM32

The diagram below shows the hardware connection between a Gimbal and STM32. Note that

- USB_TTL cable connects to USART1 connector on STM32 (Can use the ST-Link)
- PC is used for STM32 development
- In the STM32 sample App, users can receive feedback on the PC



NOTE: USART interface with gimbal can be changed in gProtocol_init function.

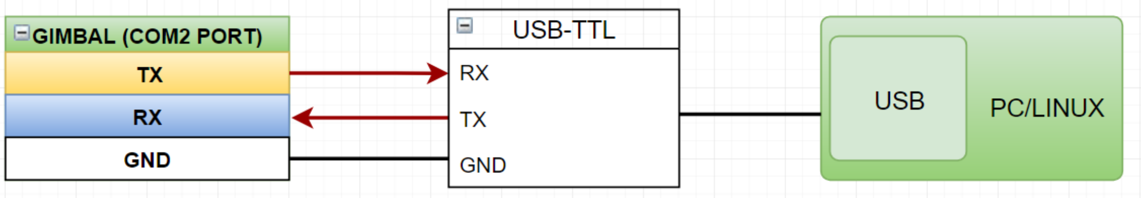
Setting up STM32 to communicate with gimbal and Computer:

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

❖ **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_STM32 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**

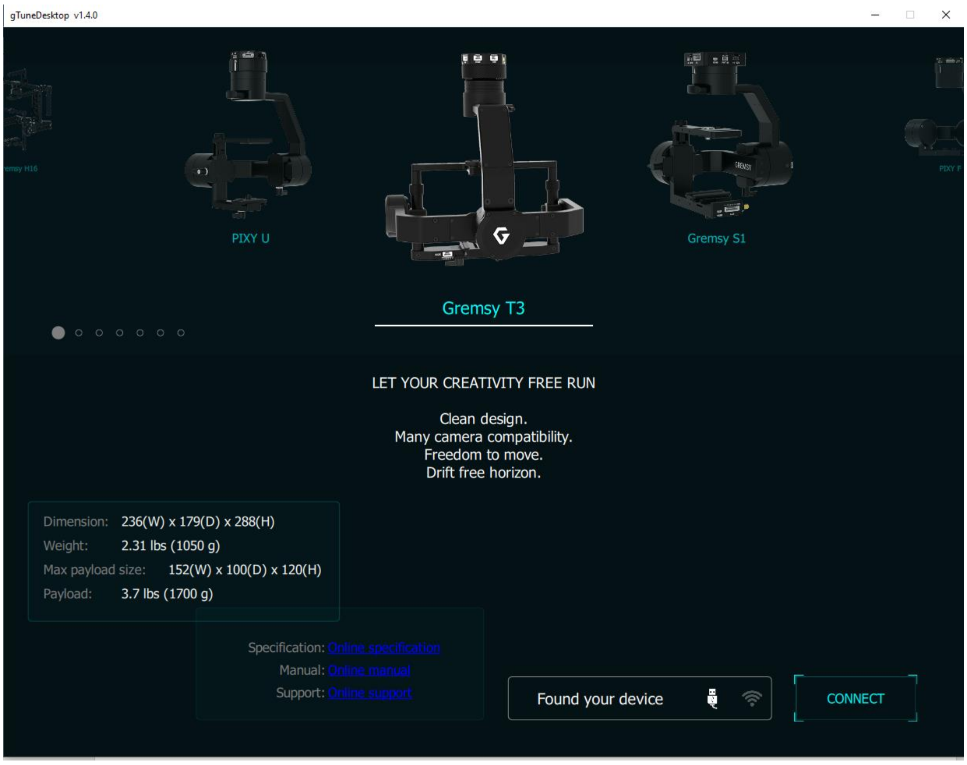
  A payload needs to be properly balanced before starting gimbal calibration. For gimbal without adjustable payload like PixyF, then move to next step.

  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 v1.4.0**

  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 v7.2.0 or above**

  Link: https://github.com/Gremsy

- **Open gTuneDesktop v1.4.0 and update the firmware.**

  01 - Power on the GIMBAL.

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

  03 - Run the gTune Desktop v1.4.0 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.

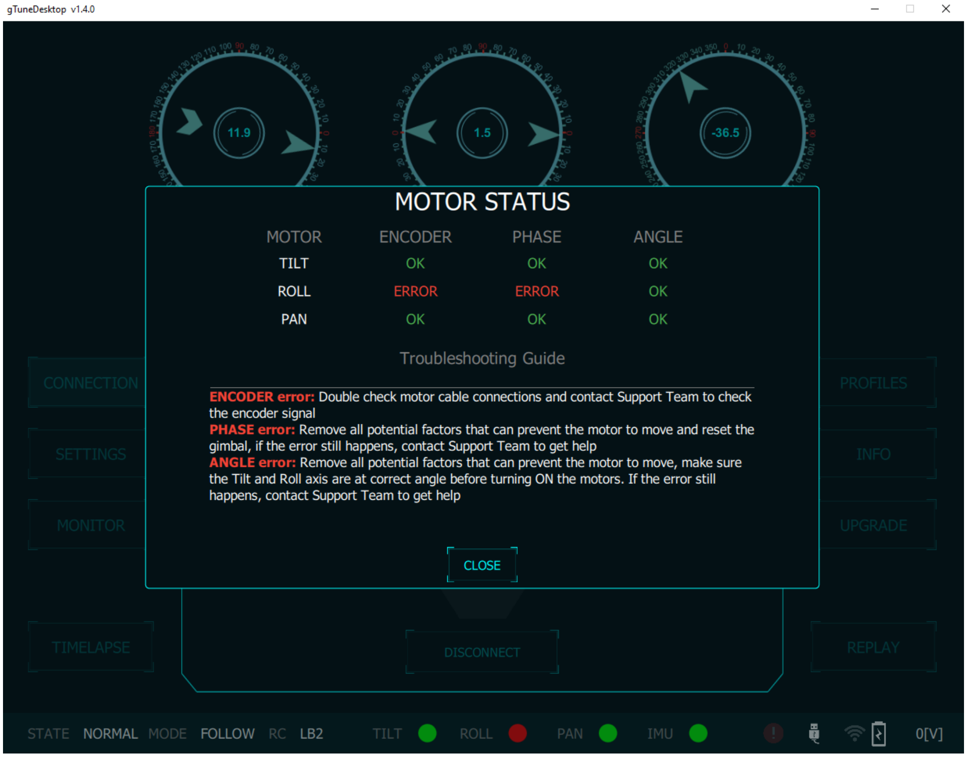
  10 - Click “Upgrade” button. The process will take about 2 minutes. When the firmware is upgraded successfully, the Gimbal will be restarted automatically.

- **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 .**



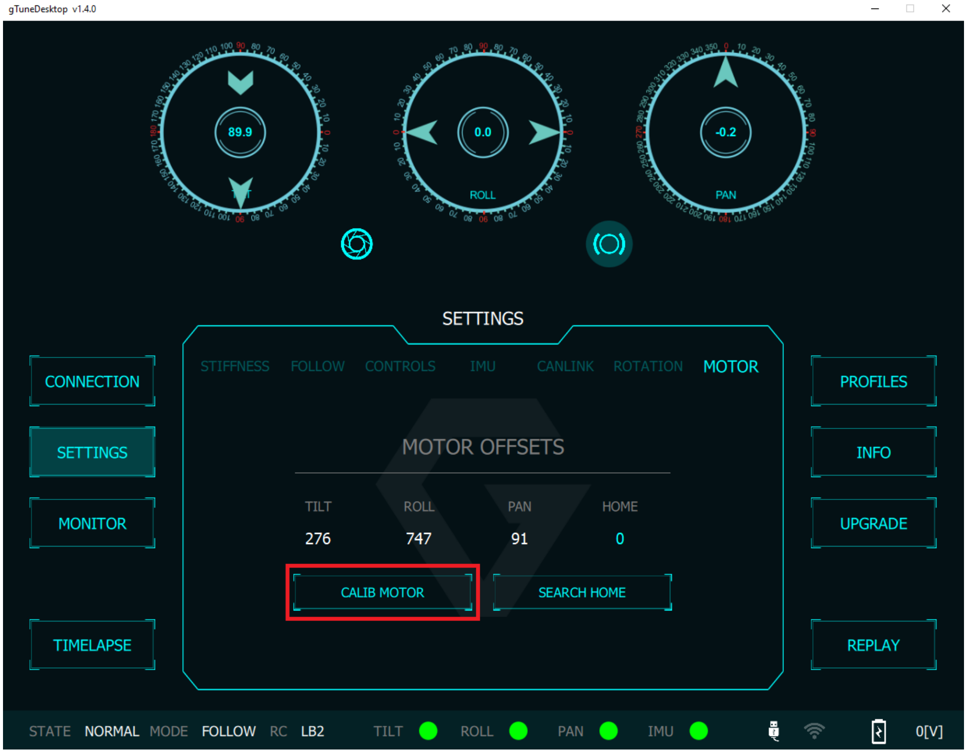
MOTOR STATUS

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



MOTOR OFFSETS

TILT	ROLL	PAN	HOME
276	747	91	0

STATE NORMAL MODE FOLLOW RC LB2 TILT ● ROLL ● PAN ● IMU ● 0[V]

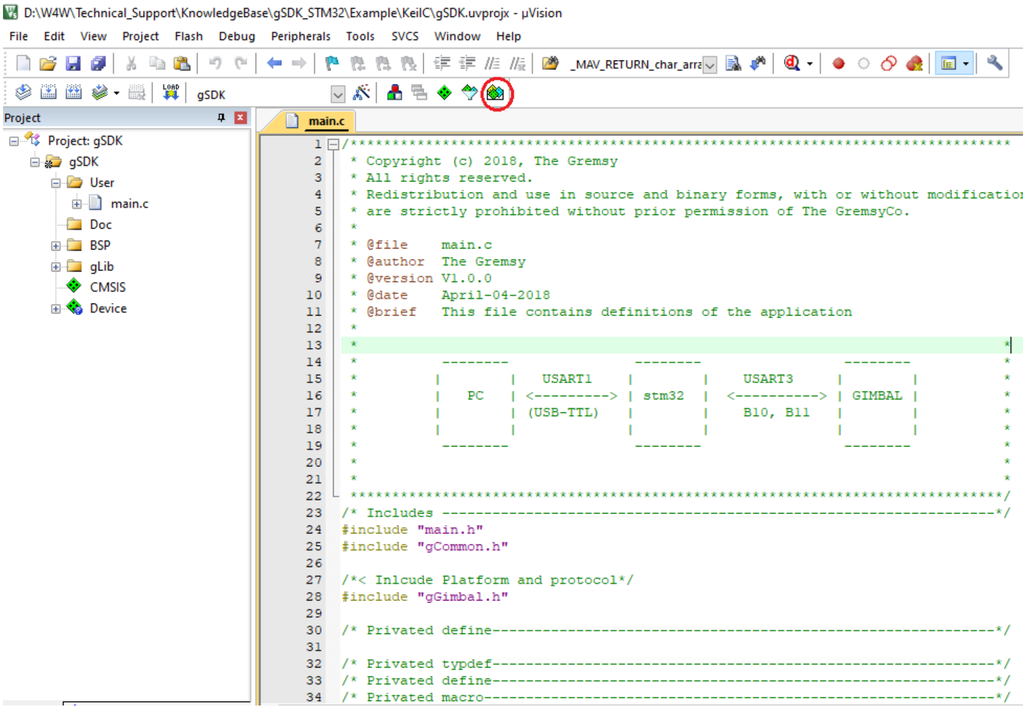
STM32

Toolchain Requirement

- Keil > MDK-ARM Version 5.22 and license it
- Window PC to run Keil

Toolchain Setup

- Configure the USART1 port to a baudrate of 115200 in your sample app
- To download (flash) the App binany to the STM32 board, connect the PC to the STM32 mini-USB” port.
- In order for Keil to build code for the target board, you need to use Keil’s “Pack installer” to install the lastest STM32F4xx_DFP as shown below

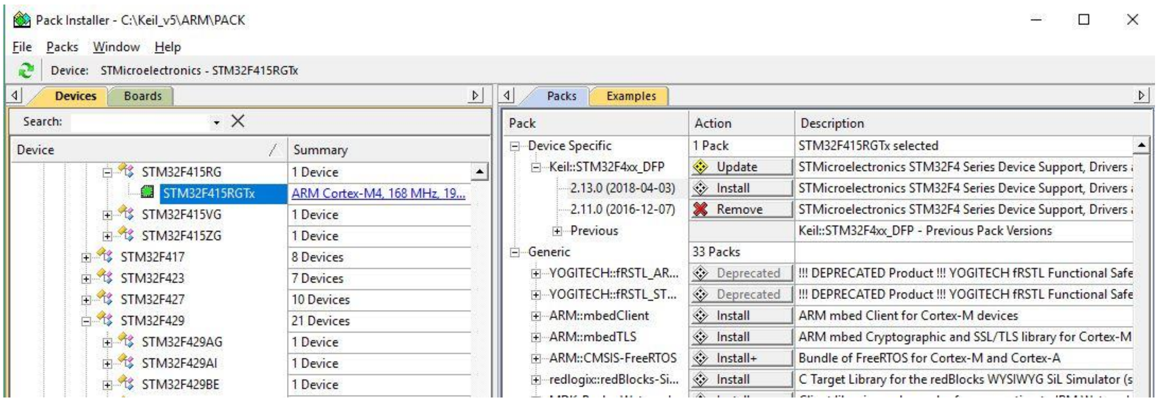

D:\W4W\Technical_Support\KnowledgeBase\gSDK_STM32\Example\KeilC\gSDK.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
_MAV_RETURN_char_arra
gSDK
Project
Project: gSDK
gSDK
User
main.c
Doc
BSP
gLib
CMSIS
Device
main.c

```
/*****************************************************************************
 * Copyright (c) 2018, The Gremsy
 * All rights reserved.
 * Redistribution and use in source and binary forms, with or without modificatio
 * are strictly prohibited without prior permission of The GremsyCo.
 *
 * @file    main.c
 * @author  The Gremsy
 * @version V1.0.0
 * @date    April-04-2018
 * @brief   This file contains definitions of the application
 *
 *                                                                           *
 *          --------            --------            --------                 *
 *         |        |  USART1  |        |  USART3  |        |                *
 *         |   PC   | <--------> | stm32 | <----------> | GIMBAL |          *
 *         |        | (USB-TTL) |        |  B10, B11  |        |            *
 *         |        |          |        |          |        |                *
 *          --------            --------            --------                 *
 *                                                                           *
 *                                                                           *
 ****************************************************************************/
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "gCommon.h"

/*< Inlcude Platform and protocol*/
#include "gGimbal.h"

/* Privated define-----------------------------------------------------------*/

/* Privated typdef-----------------------------------------------------------*/
/* Privated define-----------------------------------------------------------*/
/* Privated macro------------------------------------------------------------*/
```



Device: STMicroelectronics - STM32F415RGTx

Device	Summary
STM32F415RG	1 Device
STM32F415RGTx	ARM Cortex-M4, 168 MHz, 19...
STM32F415VG	1 Device
STM32F415ZG	1 Device
STM32F417	8 Devices
STM32F423	7 Devices
STM32F427	10 Devices
STM32F429	21 Devices
STM32F429AG	1 Device
STM32F429AI	1 Device
STM32F429BE	1 Device

Pack	Action	Description
Device Specific	1 Pack	STM32F415RGTx selected
Keil::STM32F4xx_DFP	Update	STMicroelectronics STM32F4 Series Device Support, Drivers
2.13.0 (2018-04-03)	Install	STMicroelectronics STM32F4 Series Device Support, Drivers
2.11.0 (2016-12-07)	Remove	STMicroelectronics STM32F4 Series Device Support, Drivers
Previous		Keil::STM32F4xx_DFP - Previous Pack Versions
Generic	33 Packs	
YOGITECH::fRSTL_AR...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional Safe
YOGITECH::fRSTL_ST...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional Safe
ARM::mbedClient	Install	ARM mbed Client for Cortex-M devices
ARM::mbedTLS	Install	ARM mbed Cryptographic and SSL/TLS library for Cortex-M
ARM::CMSIS-FreeRTOS	Install+	Bundle of FreeRTOS for Cortex-M and Cortex-A
redlogix::redBlocks-Si...	Install	C Target Library for the redBlocks WYSIWYG SiL Simulator (s

Ubuntu Linux

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

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

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.

Linux Onboard Computer

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

This example will receive Mavlink messages from gimbal and send MAVLink messages to control gimbal.

Building the gSDK and running example

1. Clone (or download as zip) the gSDK_Linux.
2. Open a terminal, cd into the gSDK folder and follow these steps to build the gSDK_Linux:

```
$ cd gSDK_Linux
$ 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

Here's an example output:

```
START READ THREAD

Found

GOT GIMBAL SYSTEM ID: 20
START WRITE THREAD

READ SOME MESSAGES

Got message gimbal status
Gimbal is running in FOLLOW mode!
Got message ATT
   att: time: 715054317 p: -0.072667 r: -0.127920 y: -0.623558 (degree)
Got message Mount orientation
   orientation:  time: 715084382  p :0.108494 r: -0.135546 y: 0.087891 (degree)
Got message Mount status
   mnt:  time: 1566010215715390 p:5072 r:19408 y:16 (cnt)

READ SOME MESSAGES

Got message gimbal status
Gimbal is running in FOLLOW mode!
Got message ATT
   att: time: 715054317 p: -0.072667 r: -0.127920 y:-0.623558 (degree)
Got message Mount orientation
   orientation:  time: 715084382  p: 0.108494 r:-0.135546 y:0.087891 (degree)
Got message Mount status
   mnt:  time: 1566010215715390 p:5072 r:19408 y:16 (cnt)
```

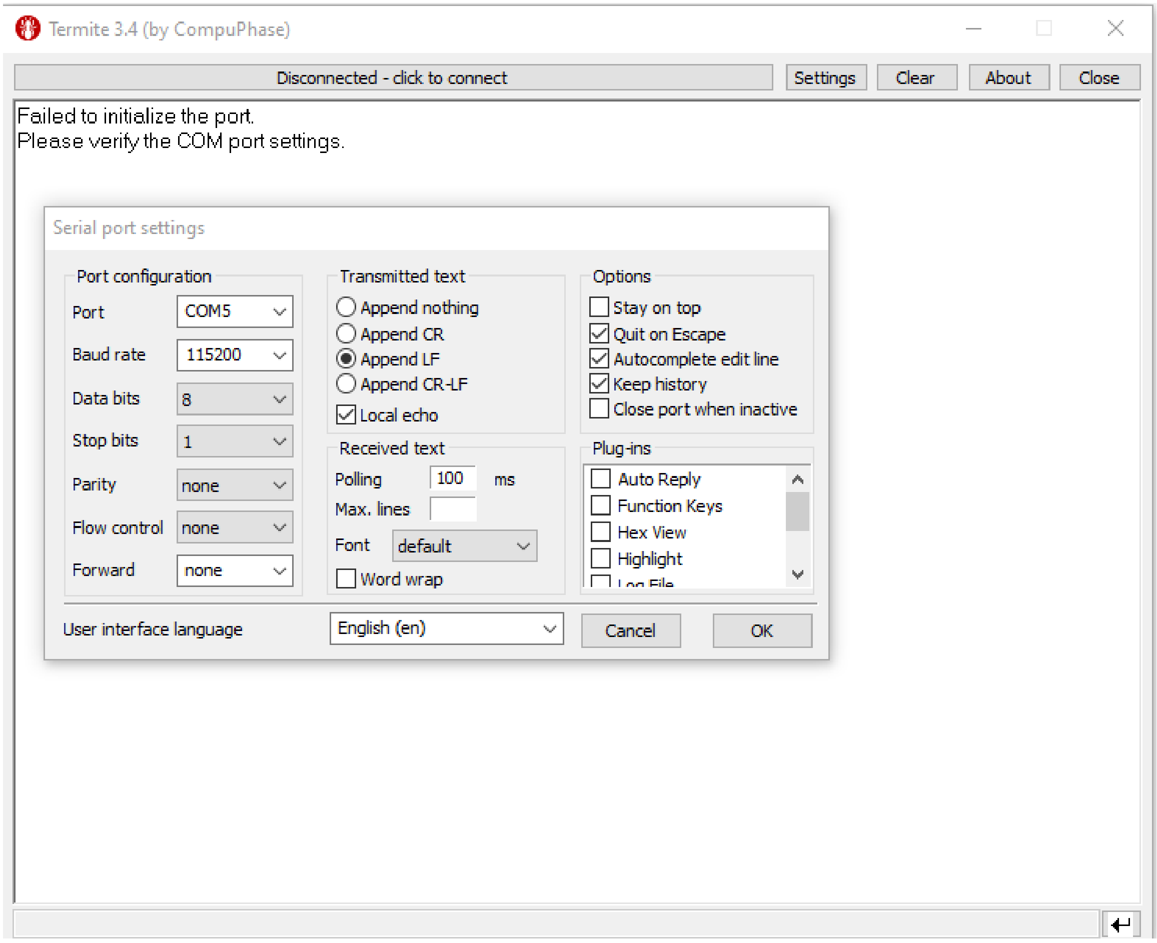
STM32 Onboard Computer

Installing and setting up the gSDK_STM32

1. Clone (or download as zip) the gSDK_STM32.
2. Open the project located in gSDK\Example\KeilC in Keil uVision IDE

Building and Running the sample

3. Use the menu item Project → Build Target and Flash->Download to build the project and flash to the STM32 board.
4. Set the baud rate of your serial terminal software (here we use the open-souce Termite 3.4) to be 115200. Configure the serial terminal to display the received information in Ascii mode.



Failed to initialize the port.
Please verify the COM port settings.

Port configuration
Port: COM5
Baud rate: 115200
Data bits: 8
Stop bits: 1
Parity: none
Flow control: none
Forward: none

Transmitted text
○ Append nothing
○ Append CR
● Append LF
○ Append CR-LF
☑ Local echo

Received text
Polling: 100 ms
Max. lines:
Font: default
□ Word wrap

Options
□ Stay on top
☑ Quit on Escape
☑ Autocomplete edit line
☑ Keep history
□ Close port when inactive

Plug-ins
□ Auto Reply
□ Function Keys
□ Hex View
□ Highlight
□ Log File

User interface language: English (en)

RoadMap for 2019/2020.

This roadmap shows the direction of the Gremsy R&D Team in 2019 and beyond.

The purpose of this roadmap is not to guarantee exactly when features will be added but instead to help the team, Partners and independent developers to spot areas for cooperation.

1. **Support gimbal setting**
   - **Motor Control**

     For gimbal with adjustable payloads, finer level motor control is useful to compenstate for payloads with difference messes. The controller is a proportional-integral-derivative (PID) type, and its coefficients can all be adjusted.
   - **Motor calibration**
   - **Set home position**
2. **Support fully compatible with the** https://mavlink.io/en/services/gimbal.html.