

# Application of TFmini-i and TF02-i CAN in PixHawk (ArduPilot Firmware)



Written by: Ibrahim (FAE)

www.benewake.com

TFmini-i and TF02-i can be interfaced with PixHawk1 CAN port or any flight controller which has Ardupilot firmware flashed and having CAN interface. Support for CAN protocol has been added to Ardupilot firmwares, starting from Copter 4.2.0 for the purpose of obstacle avoidance and Altitude Hold.

## 1. TFmini-i and TF02-i Settings:

It should be noted that TF02-i and TFmini-i have two different hardware versions for 485 and CAN. So when buying LiDAR, please pay attention to buy LiDAR with CAN interface. Multiple LiDARs can be interfaced to a single CAN bus. We need to assign different CAN IDs to each LiDAR just like we do for IIC communication. The baud-rate of each LiDAR needs to be set to the same value. On LiDAR side we have two types of CAN IDs:

- Send ID: it becomes Receive ID on CAN bus side (we need to set this ID to a new value if we are connecting multiple LiDARs.)
- Receive ID: it becomes Send ID on CAN bus side

I will consider three LiDARs example but Ardupilot supports up to 10 sensors. The commands are mentioned in details in the manual of LiDAR but I will add them here for convenience. It is still advised to read the manual of LiDAR carefully there are important points.

5A 0E 51 00 08 03 00 00 00 04 00 00 02 8 [CHANGE SEND ID TO 04] 5A 0E 51 00 08 03 00 00 00 05 00 00 00 C9 [CHANGE SEND ID TO 05] 5A 0E 51 00 08 03 00 00 00 06 00 00 00 CA [CHANGE SEND ID TO 06] 5A 04 11 6F [SAVE SETTINGS]

5A 05 60 01 C0 [Enable 120Ω Terminating Resistor]

5A 05 60 00 BF [Disable (Default) 120Ω Terminating Resistor]

5A 0E 51 00 08 03 00 00 00 03 00 00 00 C7 [CHANGE RECEIVING ID BACK TO 03]

**Some details about terminating resistor on LiDAR**: Although resistor on LiDAR is disabled by default and LiDAR works without enabling resistor but adding resistor helps in reducing equivalent resistance of transmission wires, because adding more resistors in parallel will reduce the equivalent resistance. So in case you are experiencing any kind problem with data stability then you could enable resistors on LiDARs by sending command I added above. I have tested with total five LiDARs (two with resistors enabled and three without enabling resistors and I was able to get stable data).

For sending the above commands, you will either need CAN analyzer or TTL-USB board (because UART interface of TF02-i/TFmini-i can be used to configure its parameters).



www.benewake.com

Once you are done with above settings then it's time to move to physical connection and Ardupilot firmware settings.

We take three TFmini-i or TF02-i CAN as an example in this passage and set the addresses to 0x03 and 0x04 and 0x05 separately. The default sending ID of LiDAR is 0x03 so leave it for one LiDAR and configure for other two LiDARs to 0x04 and 0x05.

## 2. PixHawk Connection:

The following two diagrams show how to interface TFmini-i and TF02-i CAN with PixHawk flight controller. The wiring details of TFmini-i and TF02-i CAN is the same.



Figure 1: Schematic Diagram of Connecting TFmini-i CAN to CAN Interface of PixHawk1

#### Note:

1. Please pay attention to connect right wire to the right pin of flight controller. Look at the pinout of controller, pin configurations are **starting from left to right**:



AN						
Pin	Signal	Volt				
1 (red)	VCC	+5V				
2 (blk)	CAN_H	+12V				
3 (blk)	CAN_L	+12V				
4 (blk)	GND	GND				

Figure 2: Pin details of CAN Interface of PixHawk1

- 2. Related connectors need to be purchased by user, LiDAR connector is 7-pin JST with 1.25mm pitch.
- 3. If LiDAR faces down, please take care the distance between lens and ground, it should be larger than LiDAR's blind zone (10cm).
- 4. If more LiDARs need to be connected (10 LiDARs can be connected), the method is same.
- 5. Power source should meet the product manual current and voltage requirement: 7V to 30V, larger than 100mA\*number of LiDAR. I used 12V supply.



Figure 3: Schematic Diagram of Connecting TF02-i CAN to CAN Interface of PixHawk1



## 3. Parameters settings:

### Common settings for obstacle avoidance:

AVOID\_ENABLE= 3 [if 3 = UseFence and UseProximitySensor doesn't work in IIC then choose 2 = UseProximitySensor]

AVOID\_MARGIN=4

PRX\_TYPE=4

Settings for CAN-1 port:

 $CAN_P1_DRIVER = 1$ 

 $CAN_D1_PROTOCOL = 11$ 

CAN\_P1\_BITRATE = [Baud-rate: For TFmini-i and TF02-i it is 250000, and for TF03 the default baud-rate needs to be set to 1000000.]

In case of pixhawk1 we only have one CAN interface but if there are more than one interfaces then configure the parameters for CAN-2 interface.

## Settings for CAN-2 port:

 $CAN_P2_DRIVER = 1$ 

 $CAN_D2_PROTOCOL = 11$ 

CAN\_P2\_BITRATE = [Baud-rate: For TFmini-i and TF02-i it is 250000, and for TF03 the default baud-rate needs to be set to 1000000.]

# Settings for first TFmini-i or TF02-i:

RNGFND1\_RECV\_ID = 3 [CAN Transmit ID of #1 TFmini-i or TF02-i in decimal]

RNGFND1\_GNDCLEAR=15 [Unit: cm, depending upon mounting height of the module and should be larger LiDAR than non-detection zone. This parameter is required to be configured for altitude hold, it is the installation height of LiDAR from ground.]

RNGFND1\_MAX\_CM = 400 [It could be changed according to real demands but should be smaller than effective measure range of LiDAR, unit is cm]

RNGFND1\_MIN\_CM=30 [It could be changed according to real demands and should be larger than LiDAR non-detection zone, unit is cm]



#### RNGFND1\_ORIENT=0 [#1 TFmini-i real orientation]

RNGFND1\_TYPE = 34 [TFmini-i CAN same as TF02-i and TF03-CAN]

#### Settings for second TFmini-i or TF02-i:

RNGFND2\_RECV\_ID = 4 [CAN Transmit ID of #2 TFmini-i or TF02-i in decimal]

RNGFND2\_MAX\_CM=400

RNGFND2\_MIN\_CM=30

RNGFND2\_ORIENT = 6 [#2 TFmini-i real orientation]

RNGFND2\_TYPE = 34 [TFmini-i CAN same as TF02-i and TF03-CAN]

#### Settings for third TFmini-i or TF02-i:

RNGFND3\_RECV\_ID = 5 [CAN Transmit ID of #3 TFmini-i or TF02-i in decimal] RNGFND3\_MAX\_CM=400 RNGFND3\_MIN\_CM=30 RNGFND3\_ORIENT = 4 [#3 TFmini-i real orientation] RNGFND3\_TYPE = 34 [TFmini-i CAN same as TF02-i and TF03-CAN]

Upon setting of these parameters, click [Write Params] on the right of the software to finish.

If the error message "Bad LiDAR Health" appears, please check if the connection is correct and the power supply is normal. Please turn-off completely the flight controller after configuring the parameters, otherwise changes will not take place. If your battery is connected to your flight controller, please disconnect it as well.



How to see the target distance from the LiDAR: press Ctrl+F button in keyboard, the following window will pop out:

🖷 temp	-	×
Geo ref images Geo Refrence photos 10 CTHO		-
Warning Manager Create custom audio warnings		
Follow Me use a numea gps to follow me		
MEA outputs the may location in mmea Inject GPS MEALURE PRESSNE		
MicroDrone outputs the may location in microdrone format FFT DIFFERENTIAL PRESSURE		
Maylink mirrors the maylink stream received by mp		
Param gen regenerage the param info used inside mp		
Lang Edit translation language editor pixhawk		
OSDVideo overlay the hud into your recorded videos QNH		
Moving Base show an extra icon on the map of your current Sequence		
Shp to Poly convert shp file ot a polygon file		
output the may location into xplanes nk In vlc ANGULE EASTER CONTROL		
Swarm multi may swarm interface gstream attitute statication		
Follow the leader follow the leader swarm		
MAVSerial pass create a exclusive passthrough to the gps Data z attract control		
remove all apm drivers		
Sort TLogs sort tlogs into there type and sysid		
rip all fw download all current fw's		
Inject GE add custom imagery to mp calib mo cross		
Clear Custom Maps wipe custom imagery		
structtest struct conversion speed test		
DashWare		
arm and takeoff quad: arm and takeoff cheske		
gimbal test run the gimbal pointing algo		
map logs create map jpg's for all tlogs in a dir		
logindex tlog browser Swarm styless worke		
GST test DEM logdownload ReSort All Custom GDAL Custom DTED LOCCING		



Click button *Proximity*, the following window will appear:



The number in green color means the distance from LiDAR in obstacle avoidance mode the number refreshes when the distance changes or window opens, closes, zooms in or zooms out, and this distance will not be influenced in Mission Planner, the version used at the time writing this tutorial is v1.3.72.

## **Altitude Hold using CAN Interface:**

Let say we use fourth LiDAR for the purpose of *Altitude Hold*. Connect the flight control board to mission planar, Select [Full Parameter List] in the left from the below bar-[CONFIG/TUNING]. Find and modify the following parameters:

PRX\_TYPE = 0 [on equal to 4 also gives the value if RNGFND4\_ORIENT = 25]

RNGFND4\_RECV\_ID = 6 [CAN Transmit ID of #4 TFmini-i or TF02-i in decimal]

RNGFND4\_GNDCLEAR = 15 [Unit: cm, depending upon mounting height of the module and should

#### be larger LiDAR than non-detection zone. This parameter is required for Altitude Hold.]

RNGFND4\_MAX\_CM = 400 [It could be changed according to real demands but should be smaller than effective measure range of LiDAR, unit is cm]

RNGFND4\_MIN\_CM = 30 [It could be changed according to real scenario and should be larger than LiDAR non-detection zone, unit is cm]

RNGFND4 ORIENT = 25 [#4 TFmini-i real orientation]

RNGFND4\_TYPE = 34 [TFmini-i CAN same as TF02-i and TF03-CAN]

Upon setting of these parameters, click [Write Params] on the right of the software to finish.

If the error message "**Bad LiDAR Health**" appears, please check if the connection is correct and the power supply is normal.

Select option *sonarrange*, see following picture:

Display This								×
accel_cal_x	🖌 az3	ch11out	ch7out	🔲 gimballng	gz	my	remnoise	ter_space
accel_cal_y	AZTOMAV	dh12in	🔤 ch8in	gpsh_acc	gz2	my2	remotesnrdb	🗌 timeInAir
accel_cal_z	battery_cell1	ch12out	ch8out	gpshdg_acc	gz3	my3	remrssi	timeInAirMinSec
accelsq	battery_cell2	ch13in	ch9in	gpshdop	HomeAlt	nz	roll	timesincelastshot
accelsq2	battery_cell3	ch13out	ch9out	gpshdop2	🗖 horizondist	nz2	rpm1	toh
accelsq3	battery_cell4	ch14in	climbrate	gpsstatus	hwvoltage	mz3	rpm2	tot
airspeed	battery_cell5	ch14out	crit_ADA	gpsstatus2	i2cerrors	nav_bearing	rssi	turnrate
alt 🗌	battery_cell6	ch15in	current	gpsv_acc	KIndex	nav_pitch	rxerrors	verticalspeed
alt_error	battery_kmleft	ch15out	current2	gpsvel_acc	lat	nav_roll	rxrssi	vibex
altasl	battery_mahperkm	ch16in	DistFromMovingBas	groundcourse	lat2	noise	satcount	🗖 vibey
altasl2	battery_remaining	ch16out	DistRSSIRemain	groundcourse2	linkqualityges	opt_m_x	satcount2	vibez 🗌
altd100	battery_temp	ch1in	DistToHome	groundspeed	lng	opt_m_y	satcountB	vlen
altd1000	battery_usedmah	ch1out	distTraveled	groundspeed2	lng2	packetdropremote	servovoltage	vx
altoffsethome	battery_usedmah2	ch2in	ekfcompv	gx 📃	load	🔤 pi dachi eved	sonarrange	vy vy
ADA 📃	battery_voltage	ch2out	ekfflags	<b>g</b> x2	localsmrdb	🔤 pi dD	sonarvoltage	vz 📃
aspd_error	battery_voltage2	ch3in	ekfposhor	<b></b> gx3	<pre>mag_declination</pre>	piddesired	speedup	watts
asratio	ber_error	ch3out	ekfposvert	🗖 øv	mag_ofs_x	pidff	SSA	wind_dir
ax 📃	🔲 boardvoltage	ch3percent	🔜 ekfstatus	<b>gy</b> 2	mag_ofs_y	🔲 pidI	target_bearing	wind_vel
ax2	brklevel	ch4in	ekfteralt	<b>E</b> y3	mag_ofs_z	pidP	targetairspeed	wp_dist
<b>ax</b> 3	🔤 campointa	ch4out	ekfvelv	gyro_cal_x	magfield	pitch	targetalt	wpno
ay ay	campointb	🔤 ch5in	ELT oMAV	gyro_cal_y	🔤 magfield2	press_abs	🔤 targetaltd100	<pre>xtrack_error</pre>
ay2	campointo	ch5out	fixedp	gyro_cal_z	magfield3	press_temp	ter_alt	yaw
<b>ay</b> 3	🗖 ch10in	ch6in	freemem	gyrosq	mx	radius	ter_ouralt	
az	ch10out	ch6out	GeoFenceDist	gyrosq2	mx2	raw_press	ter_load	
az2	ch11in	ch7in	gimballat	gyrosq3	mx3	raw_temp	ter_pend	

The altitude distance from the LiDAR will be displayed in Sonar Range (meters), see the following

picture:



