

Configuring TF Series (mini-S, Plus, Pro, Luna) LiDAR on Ardupilot Firmware using PixHawk 6C and 6X with IIC Interface



Written by: Ibrahim (FAE)

www.benewake.com
Benewake (Beijing) Co., Ltd.

Benewake TF Series (mini-S, mini-Plus, 02-Pro, Luna) LiDAR can be connected with the IIC port of PixHawk 6C and 6X Flight from HolyBro. There are three IIC ports available on PixHawk 6C:

1. **On GPS-1** > pin-4: SCL1, pin-5: SDA1;
2. **On GPS-2** > pin-4: SCL2, pin-5: SDA2;
3. **On I2C** > pin-2: I2C2_SCL, pin-3: I2C2_SDA;

but by default, data can only be read through **I2C port**. In order to use other ports some settings will be required like compiling the firmware from source code and directing the data flow to other ports etc., because there are no direct settings available in Ardupilot firmware. In this tutorial we will use I2C port (I2C2_SCL, I2C2_SDA). LiDAR can be interfaced with flight controller for the purpose of **Altitude Holding, Obstacle Avoidance** or **Terrain Following** (first two will be explained in this document). At the time of writing this document, the controller used was PixHawk 6C from HolyBro flashed with ArduCopter V4.3.3. However, this document can also be used with PixHawk 6X and other flight controllers running with different ArduCopter firmware versions with slight modification in parameter names and choosing the right port on flight controller. For choosing right port, refer to the port mapping of flight controller in its documentation. **Please note that supported firmware of Ardupilot for PixHawk 6C and 6X is 4.2.3 stable release and later.**

1. TF Series LiDAR Settings for IIC Interface:

Note: IIC interface is like interactive mode, you need to send command and receive the response from LiDAR. So, in order to process the command to obtain data-packet, LiDAR needs some processing time. The recommended relation is:

$$f_{\text{external}} = \frac{f_{\text{lidar}}}{2.5}$$

So, if LiDAR frame-rate is 100Hz then external frame-rate (the rate at which you send the command to the LiDAR) should 20Hz. If you need higher external frame-rate then you could increase LiDAR internal frame-rate by sending commands and following the above relation. Please refer to the manual of respective LiDAR. However, this is required if there are fluctuations in readings, otherwise don't need to do so. Please see the details of "frame rate" and changing the communication interface commands in the manual.

Standard output mode of LiDAR needs to be used instead of PIX mode in the latest firmwares. PIX mode was only required for the firmware versions older than Arducopter V3.6.2.

The default communication of TF Series LiDAR is TTL (UART). Both interfaces use the same cable, so please set the LiDAR to IIC communication first, see detailed commands in product manual.



We take three LiDARs as an example for (obstacle avoidance and altitude hold) in this tutorial and set the addresses to 0x10, 0x11 and 0x12 (16, 17, 18 in decimal respectively).

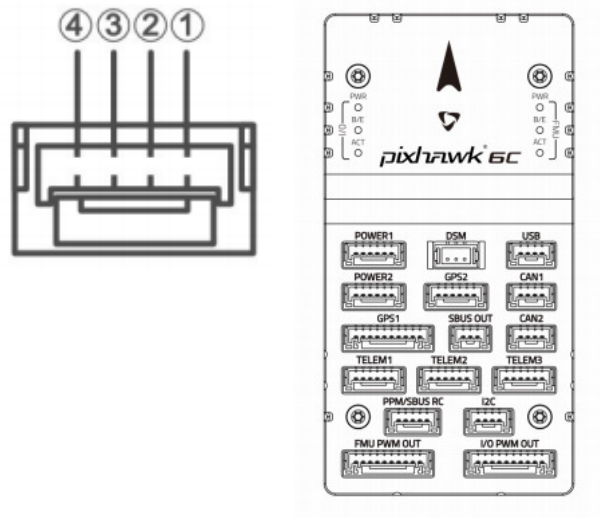


Figure-1: Pinout sequence of available ports on PixHawk 6C

Please pay attention that **Pin 1** starts from the flight controllers "right side" like in the diagram shown above.

Note:

1. Default cable sequence of LiDAR and PixHawk (6C and 6X) is different, please pay attention to the wiring sequence. LiDAR connector is **4-pin JST** with 1.25mm pitch and controller needs **JST GH 1.25mm 4-pin** connector to interface LiDAR with I2C port. Or you can also make an intermediate cable for connecting TF series LiDAR with flight controller. Looking at the pinout of controller (6C), pin configurations are:

I2C Port

Pin	Signal	Voltage
1(red)	VCC	+5V
2(black)	I2C2_SCL*	+3.3V
3(black)	I2C2_SDA*	+3.3V
4(black)	GND	GND

Figure-2: Pinout description of I2C port on PixHawk 6C

2. If LiDAR faces down, please take care of the distance between lens and ground, it should be larger than LiDAR blind zone (10cm or 20cm depending upon which LiDAR you are using).
3. If more LiDARs need to be connected (10 LiDARs can be connected), the method is same.

4. Power source should meet the product manual requirements; **Voltage:** $5V \pm 0.5V$, **Current:** larger than (**peak-current of LiDAR*number of LiDARs connected**). For current and voltage requirements, please refer to the data-sheet of respective LiDAR.
5. The communication interface of TFmini-S, TFmini-Plus and TF02-Pro can be switched by sending commands. However, in case of TF-Luna, it can be switched by connecting its 5th pin to ground. Please see TF-Luna IIC communication pin details as below:

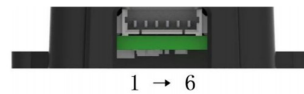


Figure 4 TF-Luna's pin numbers

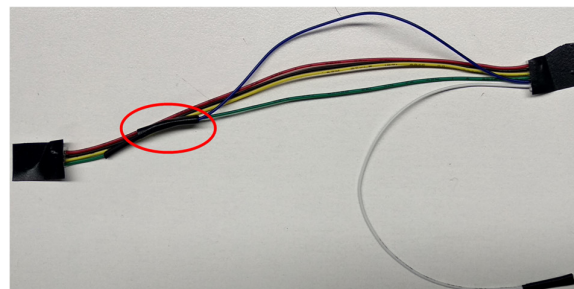
Table 6: The Function and Connection Description of each pin

No.	Function	Description
1	+5V	Power supply
2	RXD/SDA	Receiving/Data
3	TXD/SCL	Transmitting/Clock
4	GND	Ground
5	Configuration Input	Ground: I2C mode /3.3V: Serial port Communications mode
6	Multiplexing output	Default: on/off mode output I2C mode: Data availability signal on but not switching value mode

Figure 3: Pin sequence TF-Luna

If we look at the pin configuration of TF-Luna, IIC can be set by grounding **pin-5 in addition to the other four pins**. For this purpose, a customized cable is needed because in IIC mode we need to connect both pin-4 and pin-5 to the ground.

The modified cable is shown below. I have connected green wire (**pin-4**) and blue wire (**pin-5**) to single pin which will go to the GND pin of the source. Leave pin-6 connected. **Please ignore the color standard in this case as black wire represents SDA while yellow wires represent SCL**, just follow the pin numbering according to the product-manual.



So, based on the above discussion, you will need to add an extra wire (for TF-Luna) in the diagram (Figure 3) in order to use TF-Luna in IIC mode. For more understanding, I have added a

separate connection diagram for connecting multiple TF-Luna using IIC interface.

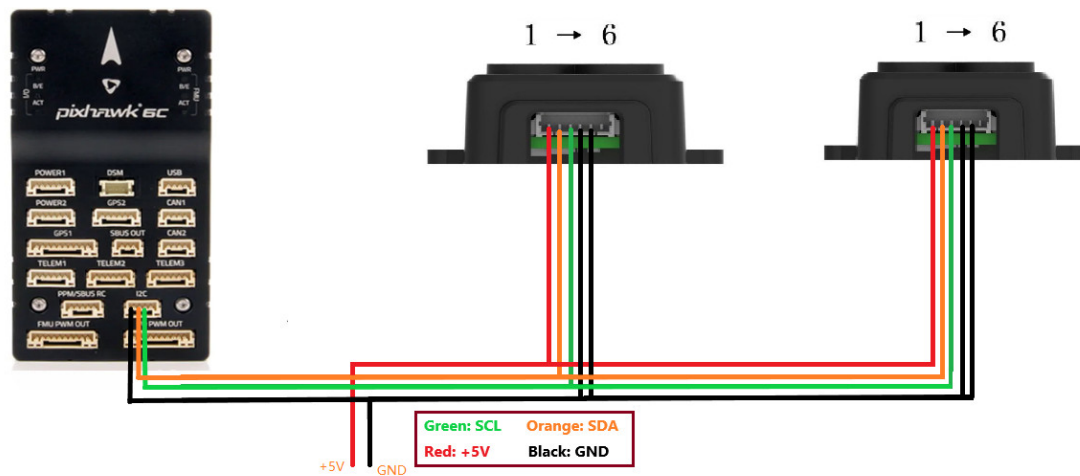


Figure 4: Schematic Diagram of Connecting TF-Luna to I2C Interface

2. PixHawk 6C (from HolyBro) Connection:

Please refer to the product manual for wiring sequence of LiDAR:

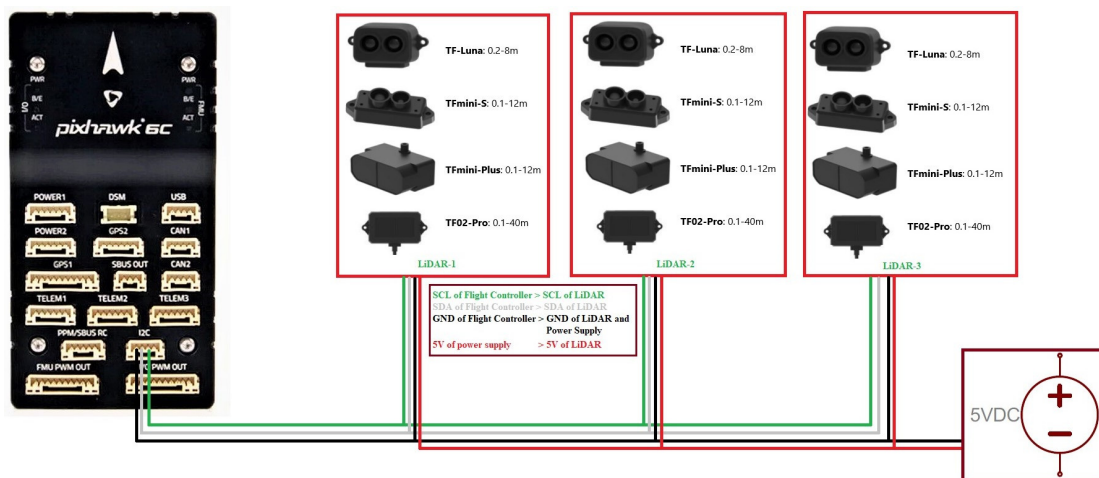
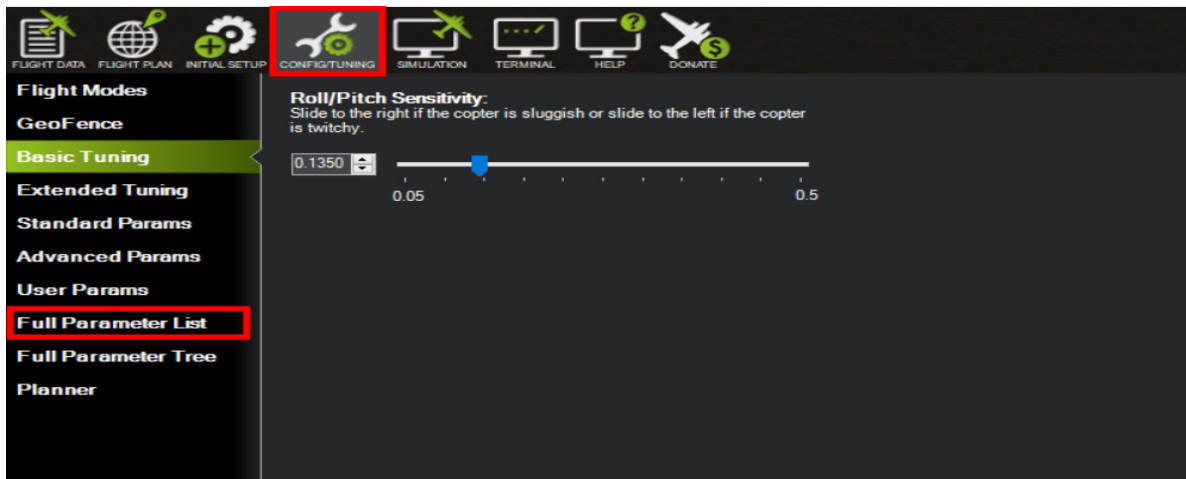


Figure 5: Schematic Diagram of Connecting three TF-LiDARs to I2C Interface of Flight Controller

3. Parameters settings (Obstacle Avoidance and Altitude Hold):

Connect the flight control board to Mission Planar. Select [CONFIG/TUNING] and then click on [Full Parameter List] in the left from the below bar. Find and modify the following parameters:



Attention: distance between UAV margin and LiDAR should be larger than LiDAR non-detection zone.

Common settings:

AVOID_ENABLE = 2 [if 3 = UseFence and UseProximitySensor doesn't work in IIC then choose **2 = UseProximitySensor**]

AVOID_MARGIN = 4 [can be set based on user requirements, can have different value depending upon which LiDAR you are using]

PRX1_TYPE = 4

AVOID_BEHAVE = 0 [This parameter will define what drone will do upon the encounter of obstacle (stop or slide to avoid the object) **0: Slide; 1: Stop**]

Settings for First LiDAR:

RNGFND1_ADDR = 16 [Address of #1 sensor in decimal]

RNGFND1_MAX_CM = 400 [It could be changed according to real application requirement but should be smaller than effective measure range of LiDAR, depends on which LiDAR you are using, unit is cm]

RNGFND1_MIN_CM = 30 [It could be changed according to real application requirement and should be larger than LiDAR non-detection zone, depends on which LiDAR you are using, unit is cm]

RNGFND1_ORIENT = 0 [#1 sensor real orientation; 0~7, 24 = Up and 25 = Down (total ten are supported up till now), see details in MP]

RNGFND1_TYPE = 25 [same for TFmini-S, TFmini-Plus, TF-Luna, TF02-Pro IIC]

Settings for Second LiDAR:

RNGFND2_ADDR = 17 [Address of #2 sensor in decimal]



RNGFND2_MAX_CM = 400

RNGFND2_MIN_CM = 30

RNGFND2_ORIENT = 4 [#2 sensor real orientation; 0~7, 24 = Up and 25 = Down (total ten are supported up till now), see details in MP]

RNGFND2_TYPE = 25 [same for TFmini-S, TFmini-Plus, TF-Luna, TF02-Pro IIC]

Settings for Third LiDAR (Altitude Hold):

RNGFND3_ADDR = 18 [Address of #3 sensor in decimal]

RNGFND3_MAX_CM = 400

RNGFND3_MIN_CM = 30

RNGFND3_ORIENT = 25 [#3 sensor real orientation; 0~7, 24 = Up and 25 = Down (total ten are supported up till now), see details in MP]

RNGFND3_TYPE = 25 [same for TFmini-S, TFmini-Plus, TF-Luna, TF02-Pro IIC]

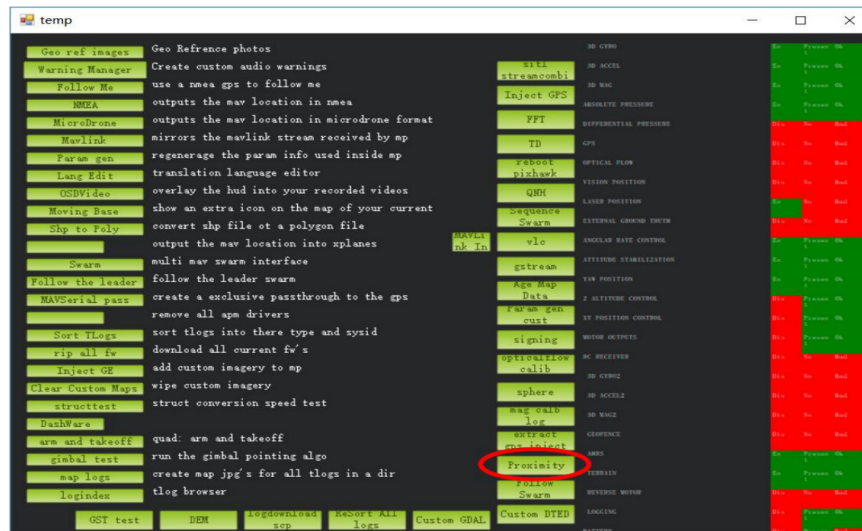
RNGFND3_GNDCLEAR = 15 [Unit: cm, depending upon mounting height of the module and should be larger LiDAR than non-detection zone. This parameter is required for Altitude Hold.]

Upon setting of these parameters, click [**Write Params**] on the right of mission planner to finish. After writing the parameters you need to power off the controller and then turn it on to apply the setting changes.

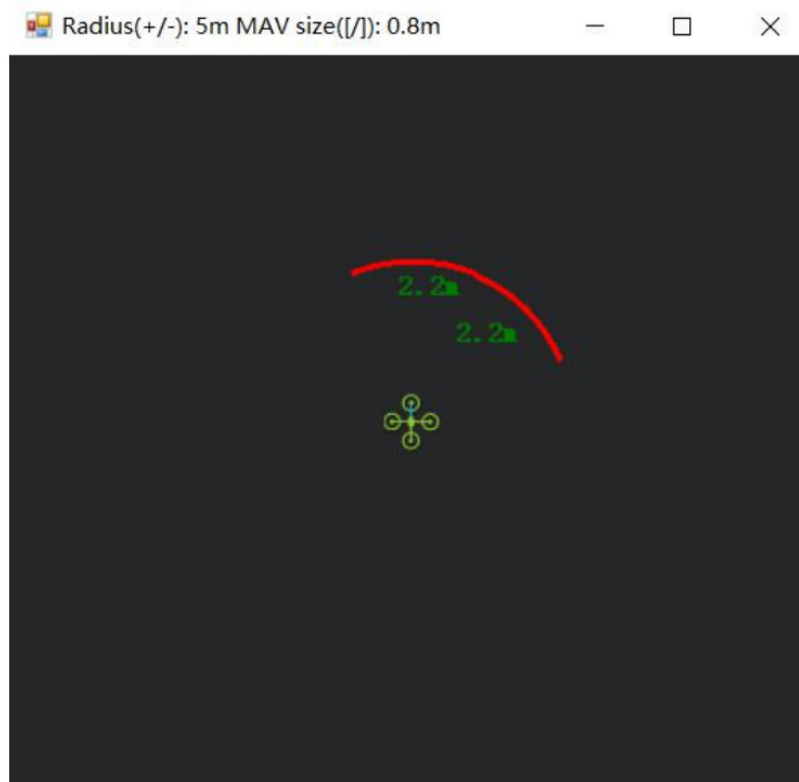


If the error message “**Bad LiDAR Health**” or “**PreArm: check the proximity sensor**” appear, please check if the connection is correct, the power supply is normal and you have restarted the controller. Also, check it whether you have changed the mode from **Standard mode** to **Pix mode** if yes then the same error will encounter.

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



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



The number in green color means the distance from LiDAR in obstacle avoidance mode (it doesn't mean the real time distance from LiDAR and will not be influenced in Mission Planner. The mission planner version at the time of writing this tutorial was v1.3.79).

Display This									
serial_cal_x	☒ ar3	ch1out	ch1out	gimbaling	gz	gy	remote	ter_space	
serial_cal_y	☒ ATX_MAV	ch2in	ch2in	gimb_acc	gz2	gy2	remote_fb	ter_tair	
serial_cal_z	☒ battery_cell1	ch2out	ch2out	gimb_acc	gz3	gy3	remote	timeAirMif	
serial_q	☒ battery_cell2	ch3in	ch3in	gimbdop	HomeAlt	az	roll	timesinceLast	
serial_q2	☒ battery_cell3	ch3out	ch3out	gimbdop2	horizontdist	az2	roll1	toh	
serial_q3	☒ battery_cell4	ch4in	ch4in	guststat	hovelatage	az3	roll2	tot	
airspeed	☒ battery_cell5	ch4out	ch4out	crit_AOA	guststat2	nav_bearing	rssi	turnrate	
alt	☒ battery_cell6	current	current	gsv_acc	klindex	nav_pitch	rsensors	verticalspeed	
alt_error	☒ battery_halfalt	ch5out	current2	gsvel_acc	lat	nav_roll	rsrssi	vibex	
altalt	☒ battery_subsuper	ch6in	DistTfrcColoring&st	gromdcourse	latc	noise	satcount	vibey	
altalt2	☒ battery_remaining	ch6out	DistHSC&Remain	gromdcourse2		opt_n	satcount2	vibez	
altid00	☒ battery_temp	ch1in	DistTfrcome	gromdspeed	linz	opt_n_v	satcount3	vlen	
altid000	☒ battery_usdmsh	ch2in	disttraveled	gromdspeedc2	linz	packetdropremote	servervoltage	vx	
altidofftheshore	☒ battery_usdmsh2	ch2in	ekCompv	gz	load	pidachieved	sonarump	vy	
AOA	☒ battery_voltage	ch2out	ekflights	gz2	local_nu&b	pidD	sonarvoltage	vz	
aspd_error	☒ battery_voltage2	ch3in	ekfposhor	gz3	mag_declination	piddesired	speedup	vatts	
asratio	☒ bar_error	ch3out	ekfposvert	gz	mag_ofs_x	piddiff	SSA	wind_dir	
az	☒ bar_voltage	ch3percent	ekfbitatus	gz2	mag_ofs_y	pidI	target_bearing	wind_val	
az2	☒ bar_volt	ch4in	ekfvelalt	gz3	mag_ofs_z	pidD	target&speed	wp_dist	
az3	☒ compoata	ch4out	ekfvelv	gz4	gro_cal_x	magfield	targetalt	wpno	
ay	☒ compoatb	ch5in	ELF_MAV	gro_cal_y	magfield2	press_abs	target&id00	track_error	
ay2	☒ compoate	ch5out	findsp	gro_cal_z	magfield3	press_temp	ter_alt	yaw	
az	☒ ch10in	ch6in	freemem	grostat	az	radius	ter_curlat		
az2	☒ ch10out	ch6out	GeoFence&dist	grostat2	az2	raw_press	ter_load		
az2	☒ ch11in	ch7in	gimballat	grostat3	az3	raw_temp	ter_sand		

The screenshot displays the 'Telemetry Logs' interface with a grid of data points. The top navigation bar includes 'Quick', 'Actions', 'Preflight', 'Gauges', 'Status', and 'Servo'. The data points are as follows:

Sonar Range (meters)	GroundSpeed (m/s)
1.80	0.04
Servo Rail Voltage	Sonar Voltage (Volt)
0.00	0.00
Vertical Speed (m/s)	Horizon Dist (m)
0.29	3534.12

Data from Three LiDAR sensors:

You can see in the following image that **RangeFinder1 (cm)** does not display any data. The reason is, this LiDAR is used as Altitude Hold sensor, so its data is shown in **Sonar Range (m)**. While the data of other two sensors is shown in **Proximity window** (right hand side).

