

ParaDrone

Flight Operations Manual

2022-02-06

By BASEline



Notice: You are entirely responsible for your own safety and compliance with regulations.

ParaDrone is a free, open-source, community project. No warranty is provided on software or hardware for any purpose.

Paragliding, powered paragliding, and parachuting are all inherently dangerous activities. Attempting to combine or go outside of the normal parameters of these sports will expose you to new and possibly unforeseen hazards. Do not participate without carefully considering all possible consequences.

Different countries and jurisdictions have different aviation laws. You are responsible for using ParaDrone in a manner consistent with local regulations.

Chapter 1: Setup	4
Calibration	4
Chapter 2: Autopilot	5
Planner	5
Chapter 3: Remote control	6
Remote control device	7
Chapter 4: Flight procedures	8
Pilot checklist	8
Groundcrew checklist	9
Remote control	9
Chapter 5: Assembly	10
Materials	11
Autopilot	11
Remote control	12
Tools	12
Cost	12
Motors	13
Printed Circuit Board	14
Software	17
Chapter 6: Implementation details	18
Remote control protocol	18
Appendix A: Board layout	22

Chapter 1: Setup

ParaDrone controls a glider by pulling on the left and right toggles.

How to attach ParaDrone depends on the design of the glider. For skydiving, clip it to your leg straps. For paragliding either mount it like a flight deck, or make a long ParaDrone to go under your legs.

Do not take off with ParaDrone attached to your toggles. Launch first, and then only when flying safely under a fully inflated canopy should ParaDrone be attached. It's not going to help on launch, and will just get in the way and create additional hazards.

Calibration

Differences in hardware and gear configuration means that it is necessary to calibrate ParaDrone before use. Perform calibration on the ground, with slack in the actuator lines. Calibration is initiated from the app. Each motor spins forward and then backward. The returned values are the number of encoder ticks per second at max speed. There can be some variation in the encoder ticks forward and backward, and left and right. A difference of ~10% is no problem, but if any of the numbers are 0, that means that the encoder is not sensing direction correctly, and must be fixed before flight.

Chapter 2: Autopilot

ParaDrone is designed to autonomously fly and land at a designated landing zone. It will attempt to fly a standard landing pattern including downwind, base, and final legs.

Landing zone is set using the android app. Click settings ⚙, and then "Set LZ". Drag the map to be centered on the intended landing zone, and rotate the map to align the arrow with the landing direction (into the wind). The app will attempt to fetch ground elevation from the internet, but can be set manually if needed, or if internet access is not available.

Autopilot will not make any turns below 30m. It will begin flaring to land at 6m above ground level.

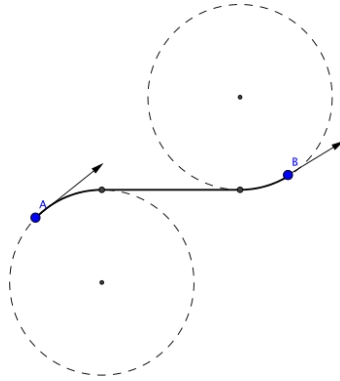
If GPS signal is lost for 60 seconds, ParaDrone will enter a slow right-hand spiral. This is to keep it from flying far away from the LZ in case of lost signal.

Planner

The planner uses waypoints and dubins paths to approximate paraglider flight paths. It can be in one of several modes, depending on location, landing zone, and user input. Modes include:

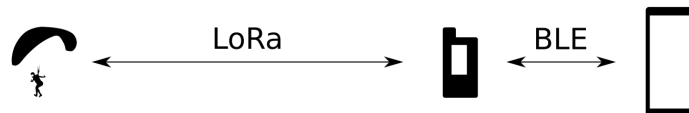
- Idle - toggles to neutral position
- RC - remote control override (lasts for 10 seconds from last control input)
- Auto - autopilot engaged but no plan
- Str - straight ahead
- NaiveL, NaiveR - naive path plan (left or right turn)
- DubinL, DubinR - dubins path plan (left or right turn)
- Waypoint - downwind, base, final waypoint plan
- Final - final approach, toggles up (30m AGL)
- Flare - toggles down! (6m AGL)

The flight planner is designed to achieve several goals. First it should be able to run on low power microcontrollers. It should fly a pattern similar to a human skydiver. If there is adequate altitude, the autopilot will set up near the landing zone, and then fly a left or right hand pattern to land in the indicated direction. At a predetermined altitude of 30m AGL there will be no more turns and toggles return to full flight position. At 6m AGL it will begin to flare for landing.



Chapter 3: Remote control

Note: You may need a Remote Pilot Certificate to operate ParaDrone remotely, or an equivalent in your jurisdiction.

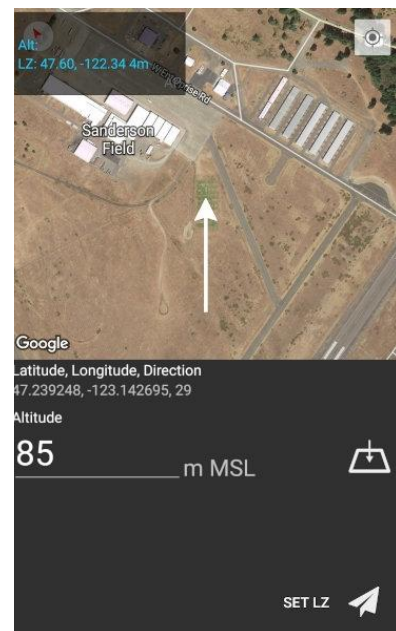


ParaDrone can use LoRa long-range radio for remote operation. LoRa is a protocol that uses unlicensed radio spectrum to do extremely long range communication with low power requirements. The Heltec ESP32 LoRa v2 board has a builtin LoRa transceiver. To receive location updates on the ground, and to send remote control commands to ParaDrone, it is necessary to use a LoRa gateway between an android phone and ParaDrone.

Ground crew can override autopilot using the android app plus the LoRa relay device. Toggle position is controlled using the touchpad at the bottom of the android app.

The touchpad operates in two modes: single-touch and multi-touch. In multi-touch mode, left and right toggles are controlled separately from the left and right sides of the screen. In single-touch mode, touch the right side to pull the right toggle, touch the center to pull both toggles (for example to flare for landing).

The app sends the desired toggle position to ParaDrone. The ParaDrone device tracks estimated toggle position, and will apply motor control to change to a new desired toggle position. If ParaDrone loses signal from the app, it will stay at the last toggle position for 10 seconds, and then revert to autopilot control.



Remote control device

The remote control device is an ESP32 microcontroller with LoRa and bluetooth radios. A 3D printed case is included in the ParaDrone project files, but is not strictly necessary.

Chapter 4: Flight procedures

Pilot checklist

- Pre-flight
 - Audio comm check
 - Battery levels
 - RC control check
 - Landing zone set
 - Satellite messenger on interval, on glider not pilot
 - Extra hook knife
 - If top launching with ballast: bring luggage scale
- Pre-jump
 - Wing configuration (trimmers)
 - Autopilot mode is idle
 - Connect toggles
 - AP control check (make sure its following path)
 - RC control check? (risk of inducing a spiral?)
 - Engine off
 - Set autopilot mode to auto
 - Audio comm: "Ready to jump"
 - Wait for all-clear from ground crew
- Jump run
 - Situational awareness (look around, think)
 - Check pilot chute! (handle available, not snagged on harness)
 - Ensure you are fully detached
 - Count number of steps to unbuckle
 - What to do if hanging below? Hook knife
 - Watch out for your speed bar. Consider removing before flight.
 - Stay altitude aware! Paraglider jumps often end up low!
 - If in doubt: can always land with glider
- In flight and under canopy
 - Stay aware of ParaDrone above you. If it went unstable on exit, it may be spiraling down at high speed

Groundcrew checklist

- Pre-flight
 - Audio comm check
 - Align relay antenna with jumprun/landing
 - Battery levels
 - Receiving LoRa data
 - Landing zone set
 - RC control check: left turn, right turn, flare
 - Wind check
 - Landing direction
 - Wind limits
- Pre-jump
 - Audio comm check
 - Confirm drop location and altitude
 - Landing zone set
 - Wind check
 - Give all-clear
- Jump run
 - Watch jumper
 - Watch for instability or entanglement of glider on exit
 - Monitor flight telemetry data
- Remote control operation
 - Stay over LZ in a slight turn
 - Ensure toggles are “up”

Remote control

Under what circumstances should ground crew take control?

Chapter 5: Assembly

ParaDrone is an open source, community driven project. I do not sell ParaDrones. The design is intended to be easy to build for any maker.

Many of the components can be substituted with no problem. The battery and motors can be changed as long as there is enough power to pull the toggles. I found experimentally that 200-300 rpm motors worked best for the forces seen in typical canopies under full weight load.

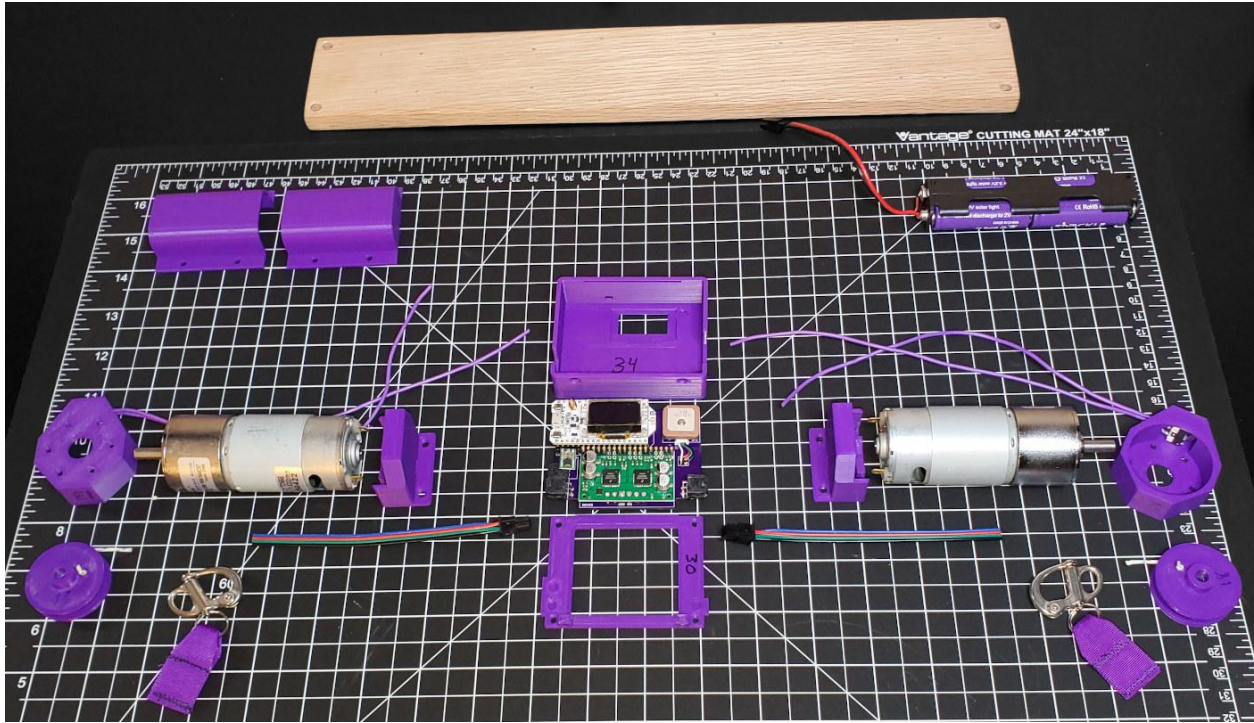
Battery considerations:

Any battery that is roughly 12v and can supply at least 4 amps of power will do. However, most LiPo batteries have fire risk if punctured or shorted. Do you want a lithium ion battery fire on your lap? No? Then use a LiFePO4 battery. We use four AA-sized LiFePO4 batteries, intended for solar lamps but work well for this purpose.

Warning: you must either: 1) Use LiFePO4 fire safe batteries; or 2) include a cutaway system for ParaDrone in case of fire!

The length of the fishing line used for the toggle attachment depends on the glider being used. Typical control stroke lengths are in the range 70-120cm. Length also depends where the device is mounted on the glider. Creativity is required for how to mount on your gear.

The software will pull the toggles all the way down until the knot is hit on the line. If this is beyond the stall point of the glider, you can tie a knot in the line to raise the bottom point. Stroke length above the knot can be configured through the app.



Materials

Autopilot

Item	Cost	Grams	Count	Total Cost	Total grams
12V DC 220RPM Gear Motor w Encoder	\$36.50		2	\$73.00	0.0
37mm metal motor bracket	\$2.50	22.5	2	\$5.00	45.0
RSL snap shackle	\$3.00	12.8	2	\$6.00	25.6
3D printed actuator	\$0.50	11.0	2	\$1.00	22.0
3D printed pulley spool	\$0.13	4.5	2	\$0.26	9.0
2mm Eyelet	\$0.10	0.1	2	\$0.20	0.2
20cm 20-gauge wire	\$0.01	1.8	4	\$0.04	7.2
Wood screws #4 x 1/2"	\$0.05	0.5	16	\$0.80	8.0
M3 8mm machine screws	\$0.15	0.4	16	\$2.40	6.4
140cm 100lbs fishing line	\$0.10	1.0	2	\$0.20	2.0
Oak Board 16" x 2.5" x 0.5"	\$1.00	250.0	1	\$1.00	250.0
Heltec ESP32 LoRa V2	\$23.00	30.0	1	\$23.00	30

Pololu Dual Motor Driver w/ feedback	\$30.00	7.2	1	\$30.00	7.2
SAM-M8Q	\$27.00	6.0	1	\$27.00	6.0
LiFePO4 3.2v 650mAh AA battery	\$4.00	18.8	4	\$16.00	75.2
4 x AA battery holder	\$8.00	9.1	1	\$8.00	9.1
Pololu S9V11F5 Voltage Regulator	\$6.00	1.0	1	\$6.00	1.0
Molex 43650-0200 Header	\$0.71	0.5	1	\$0.71	0.5
Molex 43645-0200 Housing	\$0.21	0.3	1	\$0.21	0.3
Molex 43045-0600 Header	\$1.23	1.1	2	\$2.46	2.2
Molex 43025-0600 Housing	\$0.48	0.6	2	\$0.96	1.2
Molex 43030-0007 Terminal	\$0.13	0.1	14	\$1.88	0.9
AP PCB	\$10.48	13.6	1	\$10.48	13.6
Total				\$216.60	522.5

Remote control

- 3D printed relay case (top & bottom)
- Heltec ESP32 LoRa V2
- 3.7V 680mAh LiPo battery
- Sliding switch 7 x 13mm

Tools

- Phillips screwdriver
- Drill with 1/16" and 1/4" bits
- Super glue (cyanoacrylate)
- Soldering kit
- LiFePO4 battery charger
- 3D printer

Cost

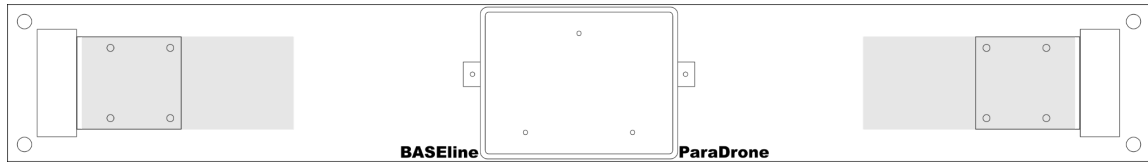
If you have all the necessary tools, it's pretty easy to assemble a ParaDrone for under \$250 USD. The majority of the cost is:

uxcell 12V DC 200RPM Gear Box Motor (x2)	\$40.00
Pololu Dual MC33926 Motor Driver Carrier	\$30.00
U-blox SAM-M8Q GPS	\$27.00
Heltec ESP32 LoRa V2	\$23.00

4x LiFePO4 3.2V 650mAh AA battery	\$16.00
Pololu D24V5F3 Voltage Regulator	\$5.00

Motors

1. Use pattern in Appendix A to drill mounting holes in the board.



2. Solder wires between battery holder and molex connector. Be quick to avoid melting the plastic.



3. Glue spool pieces together and tie fishing line to spool.

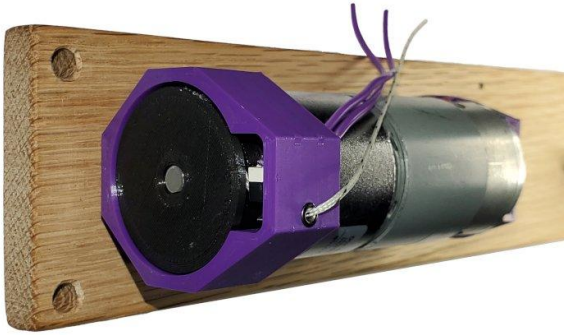


4. Screw motor bracket to board. Screw motor to motor bracket with 2 screws.



5. Screw actuator to motor using 4 remaining screws.

6. Thread fishing line out the top, and press the spool onto the motor shaft (glue if needed).



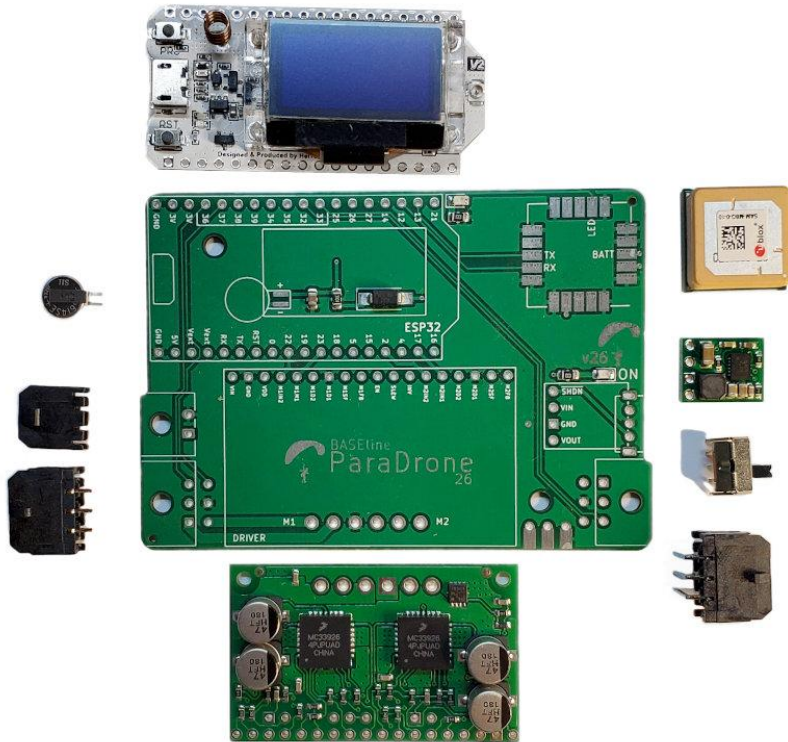
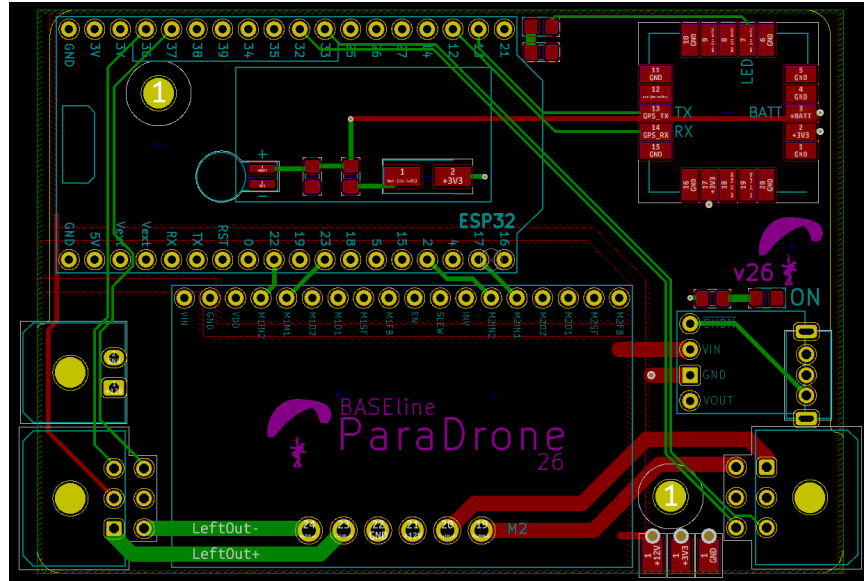
7. Tie a loop, and girth hitch the fishing line to the snap shackle.

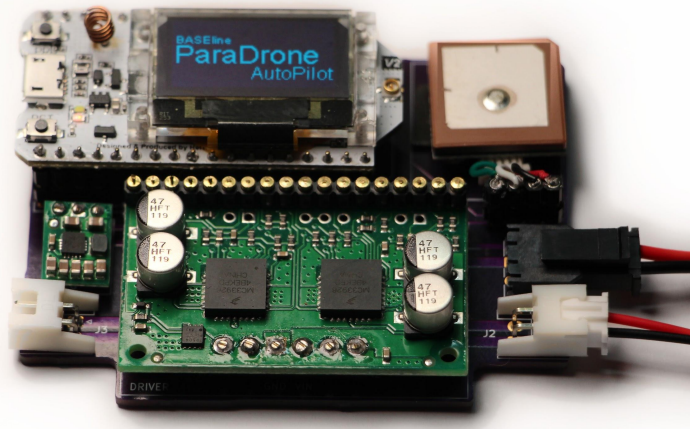


Printed Circuit Board

ParaDrone uses a custom printed circuit board (PCB) to integrate the microcontroller, gps, and motor driver. You can have small runs made by services like oshpark.com and jlcpcb.com.

The circuit takes in power up to 36V to drive the motors. Internally power is stepped down to 3.3V for the microcontroller and peripherals. It has a backup coin cell battery for faster GPS signal acquisition.





Heltec ESP32 LoRa

Pin 17	Driver M2_IN1
Pin 2	Driver M2_IN2
Pin 23	Driver M1_IN1
Pin 22	Driver M1_IN2
Pin 13	GPS TX out
Pin 12	GPS RX in
Pin 33	Right Hall Sensor A
Pin 32	Right Hall Sensor B
Pin 37	Left Hall Sensor B
Pin 36	Left Hall Sensor A
Vext	VDD
+3.3V	+3.3V
GND	Ground

Pololu dual MC33926 motor driver

M2_FB	
M2_SF	
M2_IN1	ESP32 pin 17
M2_IN2	ESP32 pin 2
M1_FB	
M1_SF	
M1_IN1	ESP32 pin 23
M1_IN2	ESP32 pin 22
VDD	ESP32 Vext
GND	Ground
VIN	+12V

U-blox SAM-M8Q GPS

RX in	ESP32 pin 13
-------	--------------

TX out	ESP32 pin 32
--------	--------------

Pololu Voltage Regulator D24V5F3

VIN	+12V
GND	Ground
VOUT	+3.3V

Software

Source code is available from <https://github.com/platypii/ParaDrone>

Use PlatformIO to install software on the ESP32 board. To program the ESP32:

Install Visual Studio Code and PlatformIO. Open the project in the arduino directory of the git repository. Connect ParaDrone to the computer by USB. Use the PlatformIO "Upload" function to program the device.

Chapter 6: Implementation details

This section contains internal technical details of how ParaDrone works. It is not necessary to understand this to operate ParaDrone.

Remote control protocol

ParaDrone uses the following commands when communicating between the autopilot and remote control. The same messages are used on bluetooth and LoRa. Each message type is identified by a letter.

(C) Motor config

Send a motor configuration message. Sent from phone to device, this sets the configuration. Sent from device in response to a “QC” query for config.

Fields

- Frequency - LoRa radio frequency to use
- Stroke - length from the top of the toggle stroke to the motor (millimeters)
- Motor direction - bitfield for left and right motor direction (counter/clockwise)

0	1	2	3	4	5	6	7	8	9	10	11	12
C	frequency				stroke mm		dir					

(I) Initial calibration

Response message from device containing calibration values. Calibration values are the number of encoder ticks measured in one second. Forward and back, left and right motors.

0	1	2	3	4	5	6	7	8	9	10	11	12
I	left1		left2		right1		right2					

(L) Location update

Location message from device

Fields

- Latitude, longitude in microdegrees

- Altitude MSL in decimeters

0	1	2	3	4	5	6	7	8	9	10	11	12
L	latitude				longitude				alt		left	right

(M) Flight mode

Whether autopilot should be engaged (“auto” = 1) or disabled and toggles should return to a neutral position (“idle” = 0)

0	1	2	3	4	5	6	7	8	9	10	11	12
M	mode											

(P) Ping

Ping the device on LoRa to start broadcasting its location.

0	1	2	3	4	5	6	7	8	9	10	11	12
P												

(Q) Query for option

Query for information from the device. Options include:

- QC - query for configuration
- QI - run motor calibration
- QZ - query for landing zone

0	1	2	3	4	5	6	7	8	9	10	11	12
Q	opt											

(S) Motor speed control

Set the left and right motor speeds. -127 = up, 127= down

0	1	2	3	4	5	6	7	8	9	10	11	12
S	left	right										

(T) Toggle position control

Set the left and right toggle positions. 0 = up, 255 = down

0	1	2	3	4	5	6	7	8	9	10	11	12
T	left	right										

(U) URL for device

After the device connects to wifi, it sends the IP address back to the phone via url message.

0	1	2	3	4	5	6	7	8	9	10	11	12
U	url											

(W) Start web server

This command tells the device to connect to a wifi network, and start a small webserver. The webserver can be used to browse log files, and for testing the device against the simulator.

0	1	2	3	4	5	6	7	8	9	10	11	12
W	ssid:password											

(Z) Landing zone

Landing zone location and direction.

Fields:

- Latitude, longitude in microdegrees
- Altitude MSL in decimeters
- Direction in milliradians

0	1	2	3	4	5	6	7	8	9	10	11	12
Z	latitude				longitude				alt		direction	

Appendix A: Board layout

Use this template to accurately drill mounting holes in your board.

