1. General notes:

When Minla is powered up, it begin to poll on USB port checking if USB modem is connected.
As soon as it finds modem it will connect to internet.
When connected, first thing it does is register itself with unique ID code (hard coded in MCU) on server and waits for user application to connect to server with the same unique ID.
All communication with receiver from user application perspective goes via **websocket** protocol.

So when user app connects to server (default server address: "**minlarc.com:9000**") it should indicate as a protocol number this unique receiver ID.

As javascript example:
var WS=new WebSocket("ws://minlarc.com:9000/",12345678); //where 12345678 - is unique ID

for details refer to: https://developer.mozilla.org/en-US/docs/Web/API/WebSocket

---

2. User API:

This chapter is the description of API used to send/receive data to/from Minla receiver (all is plain ASCII text received/sent over websocket):
For all following APIs:
outgoing - means direction from user app to Minla receiver
incoming - means direction from Minla receiver to user app

2.1. Outgoing API:

OUTGOING MESSAGE	DESCRIPTION
2d_CH1234:CH1#,CH2#,CH3#,CH4#;	Where: CH1# is a Pitch control value in a range of [0..1000] CH2# is a Roll control value in a range of [0..1000] CH3# is a Throttle control value in a range of [0..1000] CH4# is a Yaw control value in a range of [0..1000] **Notes:** In case of Naza flight controller: Middle position = 500 and initial values for all CH1#, CH2#, CH3#, CH4# should be 500. To run/stop motors do following: send **2d_CH1234:0,0,0,0;** and after 700 msec send **2d_CH1234:500,500,500,500;** When you want to take off, increase CH3# above 500, if you want to land then reduce CH3# below 500. If you want to fly forward then increase CH1# above 500, if you want to fly backwards reduce CH1# below 500.

	Same applies for Roll and Yaw controls. If you want to hover then set all values to 500 again.
2d_CH5VAL:CH5#;	Where: CH5# is a 3-position-switch control value in a range of [0..1000] **Notes:** In case of Naza flight controller: CH5# = 835 - GPS mode CH5# = 500 - ATTI mode CH5# = 330 - FAILSAFE mode
2d_CH6VAL:CH6#;	Where: CH6# is a auxiliary servo control value in a range of [0..1000]
2d_CH7VAL:CH7#;	Where: CH7# is a auxiliary servo control value in a range of [0..1000]
2d_CH8VAL:CH8#;	Where: CH8# is a auxiliary servo control value in a range of [0..1000]
2d_CH9VAL:CH9#;	Where: CH9# is a auxiliary servo control value in a range of [0..1000]
2d_HB	Heartbeat message. Needs to be sent out to receiver every 250 msec if no **2d_CH1234** message is sent to receiver during this period.
2d_CH1234CALIBRATION	Start Naza Pitch, Roll, Yaw, Throttle calibration procedure.
2d_CAMERAON	Turn on camera integrated in receiver. Camera is enabled by default as soon as first message arrives to receiver from user app.
2d_CAMERAOFF	Turn off camera integrated in receiver
2d_JPGRES:VAL#;	Where: VAL# is a video resolution, can be one of following [160, 320, 352] **Notes:** VAL# = 160 - 160x120 pixels VAL# = 320 - 320x240 pixels VAL# = 352 - 352x288 pixels
2d_JPGQTY:VAL#;	Where: VAL# is a video quality, can be one of following [10, 20, 30] **Notes:** VAL# = 10 - good quality VAL# = 20 - average quality VAL# = 30 - poor quality

2.2. Incoming API:

Incoming data can be binary and ASCII text.
Binary data is JPEG video frames that delivered from server over the same websocket (websocket protocol has built in features to differentiate between binary and text data).
User app should buffer binary data until **2i_JPEGFRAME** text message is received, then it can render JPEG frame and delete buffer.

INCOMING MESSAGE	DESCRIPTION
2i_CH1234:CH1#,CH2#,CH3#,CH4#;	Where: CH1# is a Pitch value in a range of [0..1000] CH2# is a Roll value in a range of [0..1000] CH3# is a Throttle value in a range of [0..1000] CH4# is a Yaw value in a range of [0..1000] **Notes:** Confirmation message, indicates that new Pitch, Roll, Throttle and Yaw are accepted by Minla receiver.
2i_JPEGFRAME	Message that separates JPEG frames. Data stored in binary buffer can now be rendered in UI.
2i_HB	Heartbeat signal from Minla that is normally received every 1 sec.
2i_LIPO3S:VAL#;	Where: VAL# is a voltage level of LiPo 1S up to 3S battery. Value type – FLOAT.
2i_LIPO4S:VAL#;	Where: VAL# is a voltage level of LiPo 1S up to 4S battery. Value type – FLOAT.
2i_NGPS:LON#,LAT#,ALT#,SPD#,FIX#,SVS#;	GPS data from Naza GPS module Where: LON# GPS longitude value in decimal degrees LAT# GPS lattitude value in decimal degrees ALT# GPS altitude value in meters above sea level SPD# GPS speed over ground value in meters/sec FIX# GPS raw fix status. NO_FIX = 0, FIX_2D = 2, FIX_3D = 3, FIX_DGPS = 4 SVS# GPS number of satellites used for navigation
2i_GPS:$GPGGA,,LAT#,,LON#,,FIX#,SVS#,,ALT#;	GPS GGA data from external U-BLOX GPS module: Where: LON# GPS longitude value in minutes and seconds LAT# GPS lattitude value in minutes and seconds ALT# GPS altitude value in meters above sea level FIX# GPS raw fix status. NO_FIX = 0, FIX_2D/FIX_3D = 1, FIX_DFPS = 2 SVS# GPS number of satellites used for navigation

2i_GPS:$GPVTG,,,,,,,SPD#;	GPS VTG data from external U-BLOX GPS module: Where: SPD# GPS speed over ground value in km/h

Other messages that start from **2i_** are info messages and can be printed in UI as a plain text.

For other details please refer to source code of official Minla web control panel UI (in Google Chrome open control panel html file and press F12, then select “Source”).