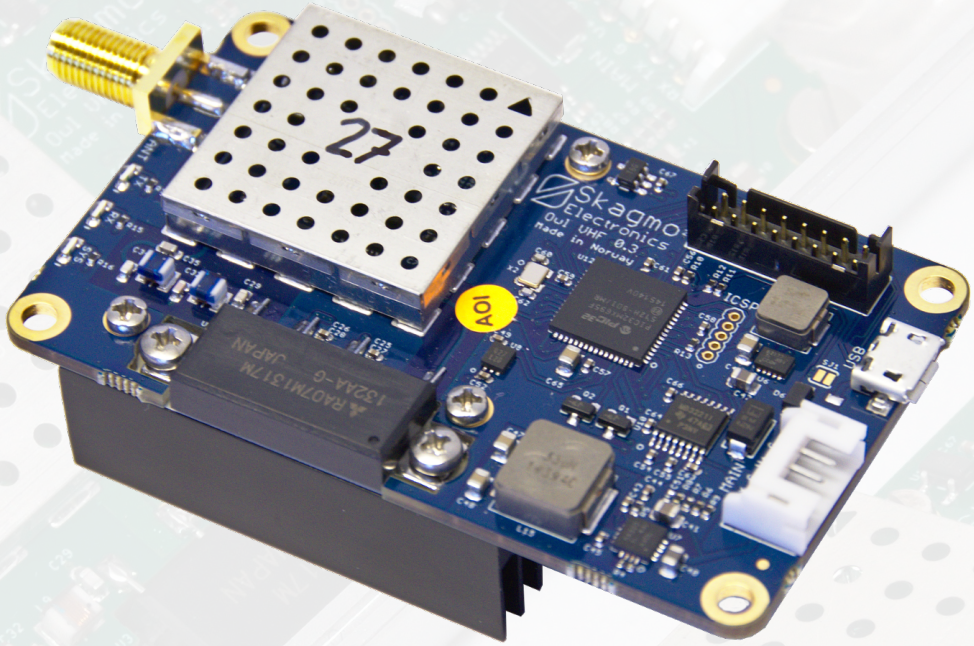
# Owl VHF

High performance radio transceiver

Manual for firmware 0.3.11

Generated November 2, 2015




Skagmo Electronics

# Contents

# 1 Getting started

The board layout is shown below. The minimum connections are a 50 ohm antenna to the ANT connector, 9-16 VDC power supply between V+ and G on the MAIN connector, as well as RS-232 communication (38400 baud 8N1 as default) on MAIN connector (Port 0). See chapter 6.3 on page 19 for details on how to build a basic cable.

Two additional serial ports are available on the EXPANSION connector, but these operate at 3.3 V and not +/- 12 V as the RS-232-level port 0.

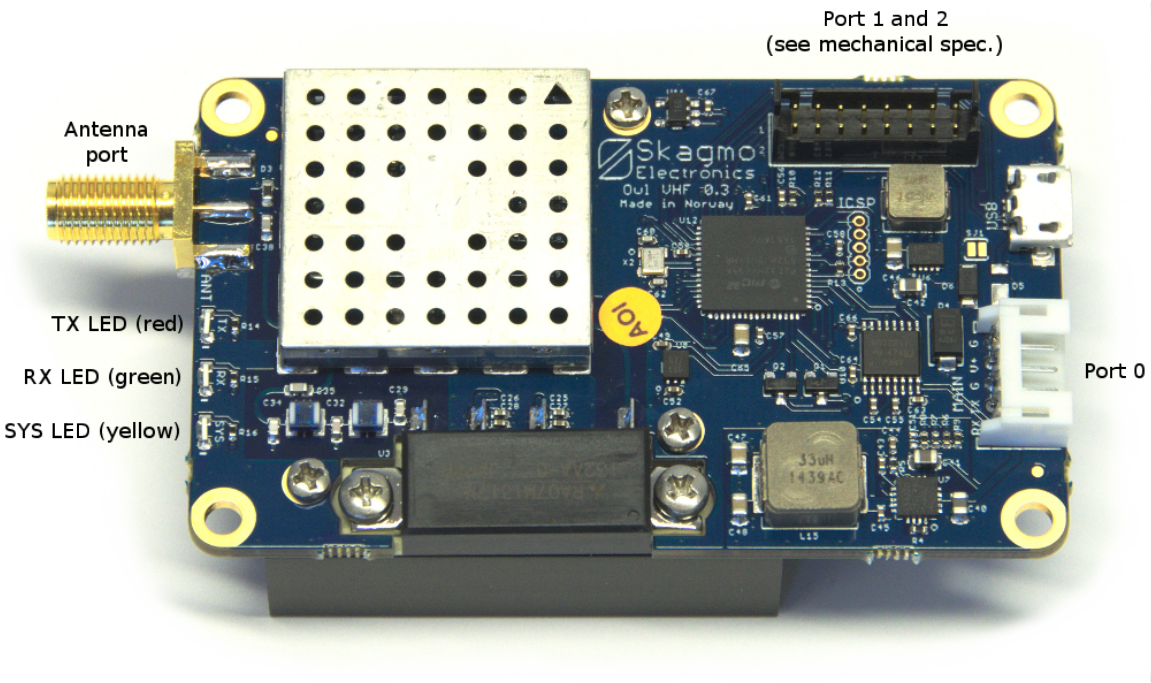



Figure 1: Board layout

When you start the Owl, its SYS LED will light solid for 3 seconds then typically start blinking patterns, indicating a successful startup.

Table 1: SYS LED pattern description (yellow LED)

| Blinking pattern | Description |
|---|---|
| No light | Standby (receiver off, but radio is ready to transmit) |
| One 0.1s blink | Receiver on |
| Two 0.1s blinks | Local time valid (TDMA can be used) |
| Three 0.1s blinks | Connected GPS/GNSS has valid time fix |
| Continously on | Bootloader waiting for connection |
| 0.5s on, 0.5s off | Bootloader failed to start application |

## 1.1 Configuration example

After startup, a command line interface is available on port 0 as default. You will need a computer running a serial terminal at this point. In Windows, try PuTTY. For full details, see the separate chapter regarding the menu system, but for now let's try some commands to send some FM-modulated voice samples:

| Please note the following conventions: | |
|---|---|
| Text | Command sent to radio |
| Text | Command from radio |
| [enter] | Carriage return, line feed or both |
| [crlf] | Carriage return and line feed |
| [tab] | Tabulator character |
| [space] | Space character |
| [ctrl]+C | Hold Control and press C |

| | |
|---|---|
| freq 145500000[enter] | Set frequency to 145.5 MHz |
| freq 145550000 ok[crlf] | Radio confirms new frequency |
| mode voice[enter] | Set radio mode to voice |
| mode voice ok[crlf] | |
| access csma[enter] | Make the radio consider the channel as free when RSSI is low |
| access csma ok[crlf] | |
| pac0 text[enter] | Set serial port packet protocol on port 0 to text |
| pac0 text ok[crlf] | |
| set[enter] | Apply port settings (serial port will now accept text data) |
| 123[enter] | FM voice is now transmitted at 145.5 MHz (TX LED is lighting up) |
| [ctrl]+C | Exit text packer and go back to command input |

# 2 Bootloader (firmware upgrade and factory reset)

The Owl comes with a built in bootloader for upgrading the firmware through port 0. Software can be downloaded from skagmoelectronics.com to load new firmware. This comes in three variants as described in table 2.

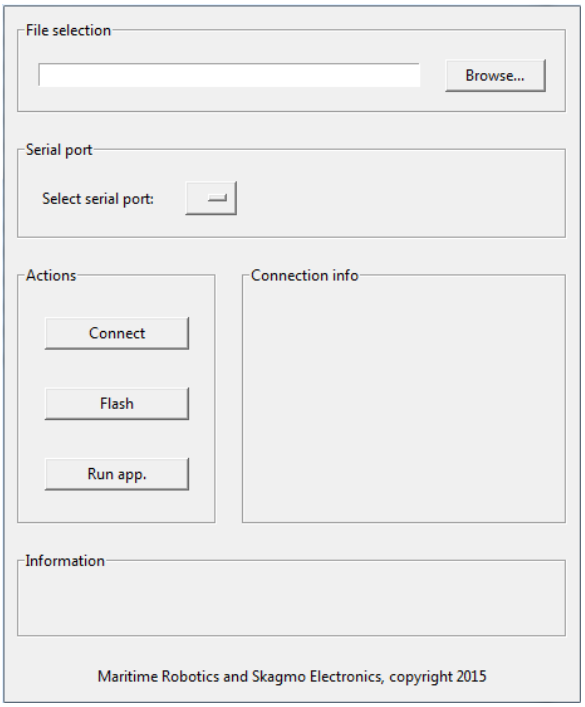



Figure 2: Bootloader

After connecting power to the Owl, the yellow SYS LED is constantly lit. This means the Owl is ready to connect to the bootloader on port 0. If no bootloader connects within three seconds, the Owl will do a self check and boot the radio firmware. To do a firmware upgrade, start the GUI-uploader, select a firmware file (.bin-extension) and serial port. Click Connect, then apply power to the Owl. The program should immediately connect and make the Owl ready to receive an update within the three second timeout. Click Flash to start the upgrade process, and finally click Run app. to start the new application. If you somehow lock yourself out from the command system on the Owl, you can restore the factory settings by sending a simple command. You will not use the bootloader application for this. This is done by sending exactly three plus-characters ("+++") to the bootloader at 115200 baud within the three second timeout after powering on the Owl. A message will be sent back to the terminal acknowledging the reset.

Table 2: Bootloader uploader variants

| Variant | Description |
|---|---|
| uploader.py | Command line utility for firmware upgrade. Install dependecies with "sudo apt-get install python-serial python-crcmod". |
| gui-uploader.py | Easy to use GUI version of the uploader. |
| gui-uploader.exe (found in gui_uploader_executable.zip) | GUI version for Windows (binary executable) with dependencies included. |

# 3 Command line menu system

The Owl features a comprehensive CLI (command line interface) for setting and reading parameters. It supports autocompleting in a known fashion. A single press on the tabulator button autocompletes the command as much as possible, and a double press lists all possible options. The special "help"-command will list all possible input if run alone, and will list all possible input for a given command if you type "help" followed by a command name. To see all possible commands, "[tab][tab]" and "help[enter]" does the same:

[tab][tab]
[crlf]access, baud0, baud1, baud2, ...[crlf]

To see possible parameters for a given command, autocompletion and help behaves a little different. First let's see what autocompletion will do:

mo[tab]
mode
[space][tab][tab]
[crlf]ais, ax25-1k2, ax25-9k6, cw, ngham, test, voice[crlf]
>mode[space]
n[tab]
>mode ngham
[enter]
[crlf]mode ngham ok[crlf]
>

Then using help and no autocompletion:

help mode[enter]
[crlf]help mode enum (ais, ax25-1k2, ax25-9k6, cw, ngham, test, voice)[crlf]
mode ngham[enter]
[crlf]mode ngham ok[crlf]

## 3.1 Serial port settings (menu items "pacN", "baudN" and "set")

To set the type of serial port protocol, used the command "pacN", where N is the port number. See the table below for a list of options for the "pacN"-command. "baudN" is used to set the serial port baud rate. All changes to "pacN" and "baudN" must be followed by the command "set" to apply. This to make it possible to change the configuration and save it before the command line interface becomes unaccessible.

| Name | Description | Syntax |
|---|---|---|
| none | Disable this port. Always set an unused port to this mode to prevent noise from being interpreted as commands or data. | |
| aivdm | Used with Automatic Identification System. Encoded VDM NMEA-message. | $AIVDM...[crlf] |
| cmd | Use port for command line menu system. | freq 144800000[enter]<br>freq 144800000 ok[crlf] |
| kiss | Widely used transparent amateur radio host protocol. | `http://www.ax25.net/kiss.aspx` |
| ngham-spp | Structured protocol to allow both commands and data without escaping/states. See separate NGHam protocol description document on `https://github.com/skagmo/ngham/tree/master/documentation`. | [start][crc0][crc1][type][length]Hello |
| nmea | Use this port as an input for a GPS capable of sending GPRMC or GPGGA NMEA messages. Remember to set correct baud rate as well. | $GPGGA...[crlf] |
| piccolo | For use with Cloud Cap Piccolo autopilot. Aligns Piccolo packets with NGHam RF packets and will reconfigure autopilot for 1/3 Hz low-res telemetry on the connected serial port if the Owl TX buffer is filled over 50%. | |
| tnc2 | The well known TNC2 style formatting of AX25-packets. | LA3JPA>LA1K:Hello[crlf] |
| text | Simply type in any data and terminate the string with "[enter]". Data received is terminated with "[crlf]". Somewhat like TNC2, but without the AX25 header. The text-packer can also parse certain escape sequences. "\t" = temperature, "\v" = voltage, "\h" = GPS HDOP. | Hello[enter]<br>Hi there[crlf] |
| transp | Transparent mode: Data received on the serial port of the transmitting radio is buffered until a timeout (specified with "transp-time") or maximum packet size is reached, then it is packed and sent. The exact same data is sent out on the serial port of the receiving radio. | |

## 3.2 Radio operating modes (menu item "mode")

The menu item "mode" defines the RF physical layer (modulation, forward error correction etc.) to the layer just below the serial port protocol (with some exceptions). By typing "help mode[enter]" the radio will return a list of possible modes. A quick description of the modes are shown below.

| Name | Description | Technical details |
|---|---|---|
| ais | Automatic Identification System, RX only. Use with packer "AIVDM" and on frequencies 161.975 MHz and 162.025 MHz. | 9600 sym/s 2-GMSK, bt=0.5, HDLC-coding. |
| ax25-1k2 | 1200 baud AX.25. A popular amateur radio mode used for APRS (144.800 MHz in Europe, 144.390 MHz in USA) and much more. | 1200 sym/s 2-AFSK (FM-modulated FSK), HDLC-coding. |
| ax25-9k6 | 9600 baud AX.25. Not so widely used – mostly for satellite operation and some ground based packet nodes. | 9600 sym/s 2-GFSK, 3 KHz deviation, K9NG scrambler polynomial, HDLC-coding. |
| cw | Morse code generator, TX only. Input ASCII text, for example with the text-packer. | |
| ngham | High performance FEC-protocol, suggested for amateur radio as well as other use. See separate NGHam protocol description document on `https://github.com/skagmo/ngham/tree/master/documentation`. | 9600 sym/s (default), GMSK, bt=0.5, short preamble, sync word correlator, lexicode size tag, Reed Solomon FEC code block. Max. 220 byte payload. Sensitivity below -120 dBm. See details in appendix A. |
| test | Generates a PN9 test sequence (pseuro-random number generator), and measures bit error rate in reception. Not finished. | |
| voice | Some pre-recorded voice samples played back in narrow band FM. TX only. Input ASCII text, for example with the text-packer. | |

## 3.3 Channel access (menu items "access", "tdma-*")

The channel access method determines how the radio will enter transmission when it has data to send. In other words, it is the method used to determine if the channel is busy or free. A proper channel access method is crucial to avoid collisions between multiple radio nodes, and the Owl has three different methods which can be selected with the "access"-command.

| Name | Description | Technical details |
|---|---|---|
| none | Never allow access / receive only | |
| always | Always allow immedate access | |
| csma | Carrier Sense Multiple Access | The channel is considered free when receiver is not busy decoding a packet. |
| tdma | Time Division Multiple Access | Transmission is purely based on time. Each node will transmit in it's own dedicated time slot, which theoretically eliminates collissions and gives a completely predictable throughput for each node. For many nodes and short packets, this will allow much better channel utilization than CSMA. |

The TDMA method requires a common syncronized clock between all radios. This can be done by connecting a GNSS-/GPS-receiver with a PPS (pulse-per-second) output to all radios, or radios can inherit time from each other. For example, one radio can be connected to a GPS, and set up to transmit time-of-hour packets ("nei-toh") at regular intervals. Alternatively, no GPS is needed at all if the radio is set up to use local uptime as time-of-hour ("tdma-tohloc 1") rather than actual time of hour from GNSS. If "tdma-tohloc" is used, it must only be enabled on a single radio. If multiple radios tries to use their own local uptime as timing reference the timing will fail completely. The TDMA method has quite a few dedicated commands:

| Name | Description | Technical details |
|---|---|---|
| tdma-frame | Time in milliseconds for the whole TDMA frame. | |
| tdma-guard | Guard time after transmission, in milliseconds. | Some "dead-time" is needed between time slots to deal with non-perfect time synchronization, signal propagation time, and to allow a node to ramp down it's transmitter before the next node will access the channel. |
| tdma-offset | Selects time slot relative to the start of the period. | An offset of 0 means the node is using the first time slot in the frame. An offset of tdma-slots – 1 means the node is using the last slot. |
| tdma-slots | Number of slots in a whole frame. | |
| tdma-strtch | Allows "stretching" transmission over multiple time slots. | A stretch value of 2 means the radio will occupy two successive time slots. |
| tdma-tohloc | Time reference for TDMA will be based on local node uptime rather than GPS-referenced time of hour. | In the case where no radios are equipped with a GNSS-receiver, this command allows one node to provide a common time reference based on local uptime rather than absolute time of hour. If a node with a GNSS-receiver (and actual time of hour) enters the same network, all nodes will choose to adjust their timeframe to actual time of hour, including the one originally configured to use its local uptime as timing reference. |

See the following figure for some examples on how to configure the TDMA system:

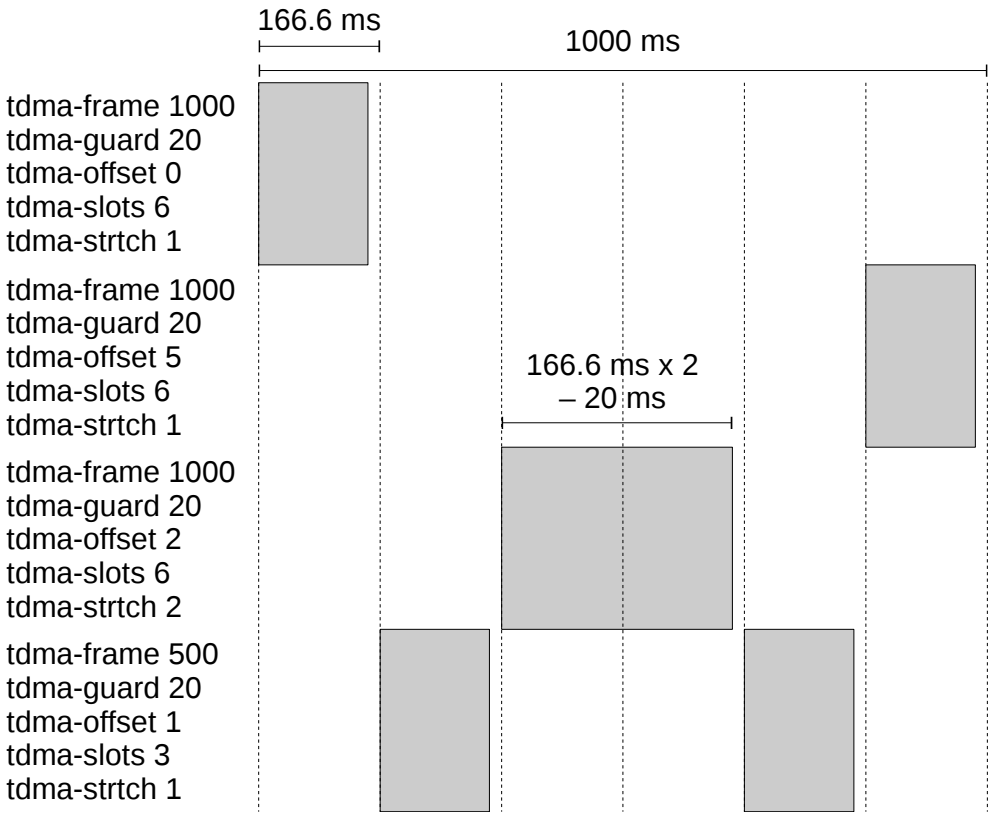



Figure 3: TDMA configuration examples

# 4 Advanced configuration examples

## 4.1 APRS

APRS is a popular positioning system used by amateur radio operators. A typical configuration will be like shown below. All commands should be succeeded by [enter].

```
freq 144800000
mode ax25-1k2
access csma
pac0 tnc2
set
```

Packets should now be printed on the TNC2-format as they are received.

## 4.2 End-to-end NGHam text messaging

This is a simple example of sending text messages end-to-end with two Owl radios. Do this configuration on both radios. In case the radios are not configured similarly in regards of modulation

etc., run "cfg-def" before doing further configuration to reset configuration. All commands should be succeeded by [enter].

| | |
|---|---|
| freq 144900000 | Feel free to choose another frequency |
| mode ngham | |
| access csma | |
| pac0 text | Assuming you are connected to serial port 0 |
| set | |

Write some text and press enter. The text should now show up on the other Owl.

## 4.3 NGHam with TDMA and packer "transp"

This is an advanced example for a robust and completely transparent link with injected statistics packets (contains a time stamp used for TDMA synchronization between nodes), but with a delay up to one second due to the use of TDMA. All commands should be succeeded by [enter].

| | |
|---|---|
| mode ngham | |
| access tdma | |
| pac0 transp | |
| nei-toh 4 | Inject time of hour packet to sent data every fourth TDMA frame |
| tdma-frame 1000 | TDMA frame time 1000 ms |
| tdma-guard 20 | Guard time between time slots is 20 ms |
| tdma-offset 0 | Select first slot in frame |
| tdma-slots 2 | |
| tdma-strtch 1 | |
| tdma-tohloc 1 | Select this as the timing reference in the network – must send stat packets |
| set | |

Before the "set"-command, "cfg-save" could have been used to make the settings permanent and load immediately after a power cycle. Be careful though, as the transparent packer has no escape sequence. This means you are effectively locked out from the menu system unless you connect to one of the other ports on the Owl or do a factory reset on port 0. This procedure is described in the bootloader-chapter.

The next node in the same system should be configured similarly, but with "tdma-offset 1", to use slot 2/2 in this TDMA-network, and not "tdma-tohloc 1", as this will cause confusion when two nodes try to be the time reference in the network.

## 4.4 Using with Pixhawk

The Owl can be used as a telemetry radio for eg. Pixhawk, but it is important to ensure the telemetry rates are low enough to prevent TX buffer overflow.

A few different ground control applications have been tested, and APM Planner is the recommended software to use as of today. Mission Planner and QGroundControl can not be used as they tend to increase telemetry rates under various conditions, whereas APM planner is consistent and will hold its settings. Go to "CONFIG/TUNING", "APM Planner 2.0 Config", "Advanced", and set "Attitude", "Position" and "Status/Mode" to 1 Hz (all else should be 0).

For this example, connect port 2 (3.3 V UART) on the Owl's black expansion connector to the Pixhawk telemetry port 2, and configure Pixhawk with BRD_SER2_RTSCTS as 0 and SERIAL2_BAUD as 38. It is now assumed that the command line interface is available on port 0 ("pac0 cmd"). Use port 0 (RS-232) on the Owl's white main connector to connect a terminal and set the following settings:

Common settings for both radios:

| | |
|---|---|
| freq 144925000 | For use on amateur radio frequency 144.925 MHz |
| power mid | |
| modulation 4-9600 | For increased throughput, but 2-9600 is the most robust |
| transp-time 50 | |
| nei-stat 8 | Inject statistics packet to sent data |
| nei-call 8 | Inject ID packet (containing callsign) to sent data |
| mycall CALL | Change CALL to your callsign |
| access tdma | |
| tdma-frame 1000 | TDMA frame time 1000 ms |
| tdma-guard 20 | Guard time between time slots is 20 ms |
| tdma-slots 3 | Three slots in total |

Enter this after setting the common settings on the plane-side:

| | |
|---|---|
| pac2 transp | |
| baud2 38400 | |
| tdma-offset 0 | Select first slot in frame |
| tdma-strtch 2 | Use two consecutive slots as more data goes from plane to ground than the other way |
| tdma-tohloc 0 | Don't use local uptime as time reference! |
| cfg-save | |
| set | Set/apply port settings (done automatically on reboot) |

On the ground side, the RS-232 port (port 0) is typically used for GCS communication. A separate serial port can be used for the command line interface as done on the airplane side, but it is also possible to configure the port and then switch to transparent mode.

Enter the common settings first, then the settings shown below:

| | |
|---|---|
| pac2 none | |
| tdma-offset 2 | Select third slot in frame |
| tdma-strtch 1 | |
| tdma-tohloc 1 | Use local uptime as timing reference in this network |
| nei-toh 4 | Inject time of hour packet to sent data to allow other radios to synchronize |
| pac0 transp | Use transparent packer |
| cfg-save | Saving at this point makes the port unavailable for commands after reboot (must do a configuration reset!) |
| set | Applying the port settings makes the port unavailable for commands immediately |

To get access to the ground radio command interface on port 0 (if not one of the other ports are configured for command interface), a configuration reset will be necessary as explained in the separate bootloader chapter.

In APM Planner, go to "Tool Widgets", "MAVLink Inspector", and check that 10 messages are coming in at 1 hz. Transmitting parameters will take a while (typically a minute), and they should be sent at about 5 Hz.

## 4.5 Using with Cloud Cap Piccolo

The Owl is very well suited for the Cloud Cap Piccolo. The Owl has a custom packer for Piccolo which aligns RF packets with the Piccolo packets, and will automatically reconfigure the connected port on the Piccolo to 1/3 Hz low-resolution telemetry if the TX-buffer is filling up uncontrollably. Of course this can be used with more than one plane by configuring more TDMA-slots and a longer TDMA-frame.

Make sure the Piccolo does not send out unnecessary data, for example by sampling IO state often. In Piccolo Command Center, go to "Window", "Preflight", "Payload IO Settings". Set "Input Sample Time" to 5 seconds or more if you can. Setting this to eg. 1 second injects a lot of data to radio stream.

Common settings:

| | |
|---|---|
| freq 144925000 | For use on amateur radio frequency 144.925 MHz |
| power mid | |
| access tdma | |
| modulation 2-9600 | |
| nei-stat 8 | Inject statistics packet to sent data |
| nei-call 8 | Inject callsign packet to sent data |
| mycall CALL | Change CALL to desired callsign |
| tdma-frame 1000 | TDMA frame time 1000 ms |
| tdma-guard 20 | Guard time between time slots is 20 ms |
| tdma-slots 3 | Three slots in total |
| pac0 piccolo | |

Plane specific:

```
tdma-strtch 2
tdma-offset 1
tdma-tohloc 0
cfg-save
set
```

Ground station specific:

```
tdma-strtch 1
tdma-offset 0
tdma-tohloc 1
nei-toh 4
cfg-save
set
```

# 5 Full command list

| Name | Description | Example |
|---|---|---|
| access | Select channel access method (how the radio goes into transmission on a shared channel). See separate chapter. | access csma[enter] |
| baudN (0-2) | Set baud rate of serial port N. | baud0 9600[enter] |
| bias-rx | Keep the power supply for the PA drain active during RX to get slightly faster RX to TX turnaround time on expense of higher RX power consumption. (PA gate will not be biased in RX, so the difference is small). | bias-rx 1[enter] |
| cfg-def | Load default "factory configuration" immediately. Will not overwrite saved configuration in flash and will only last until next power cycle. | cfg-def[enter] |
| cfg-save | Save current configuration to flash to make it load after power loss or a forced reboot. | cfg-save[enter] |
| cw-wpm | Set the keying speed for mode cw in words per minute. | cw-wpm 15[enter] |
| dest | Destination field for protocols TNC2 and MIC-E. Was "tnc2-dest" in previous firmware. | dest APRS[enter] |
| echo | Enable local echo and auto completion on the command line interface. | echo 1[enter] |
| fm-emphasis | Enable pre-emphasis on FM-modulated modes (AX25_1K2 and VOICE). | fm-emphasis 1[enter] |
| freq | Set operating frequency in hertz. | freq 144800000[enter] |
| help | Display available parameters and parameter type for one of the other command items. | help freq[enter] |
| info | Show a large list of software/hardware parameters. | info[enter] |
| kiss-smack | If set to "1", packer KISS will add a checksum according to the SMACK standard (Stuttgart Modified Amateurradio-CRC-KISS). Received KISS data will need a valid CRC to be accepted. | kiss-smack 1[enter] |
| mice-cmt | Comment field for the MIC-E protocol. Escape sequences such as used with the text-packer are valid. | mice-cmt Testing Owl VHF. Voltage=\v.[enter] |
| mice-cmtint | Comment interval in number of MIC-E-packets. For example, a value of 3 will send a comment with each third MIC-E-packet. | mice-cmtint 3[enter] |
| mice-int | Transmission interval in seconds for MIC-E protocol. | mice-int 120[enter] |
| mice-msg | Specifies the MIC-E message, from 0-7. These are Off duty, En route, In service, Returning, Committed, Special, Priority, Emergency. | mice-msg 0[enter] |

| | | |
|---|---|---|
| mode | Radio operating mode / physical layer. See separate chapter. | mode ngham[enter] |
| modulation | NGHam modulation format. On the form "x-y", where "x" is bits per symbol and "y" is the symbol rate | modulation 2-9600[enter] |
| mycall | Source field for protocols TNC2, MIC-E and NGHam extensions. Was "tnc2-src" in previous firmware. | mycall LA3JPA-9[enter] |
| nei-call | Transmit NGHam extension callsign packet at an interval in units of the value specified in "tdma-period". If "tdma-period" is 1000 ms, a value of 5 will inject a packet each 5000 ms. Interval 0 is off. | nei-stat 5[enter] |
| nei-pos | Same as above, but for the NGHam extension position packet. | nei-pos 5[enter] |
| nei-stat | Same as above, but for the NGHam extension statistics packet. | nei-stat 5[enter] |
| nei-toh | Same as above, but for the NGHam extension time of hour packet. | nei-toh 5[enter] |
| pacN (0-2) | Set packer / serial port protocol. See separate chapter. | pac0 text[enter] |
| path | Path field for protocols TNC2 and MIC-E. Was "tnc2-path" in previous firmware. | path WIDE2-2[enter] |
| power | Set power level in one of three steps. "lo" (0.2 W), "mid" (1 W) or "hi" (5 W). | power mid[enter] |
| preamb | Set length of preamble in bytes. Only valid for AX.25-modes. | preamb 50[enter] |
| remote-cmd | Enable/disable remote commands (configuration through radio) with NGHam extension. | remote-cmd open[enter] |
| restart | Make the radio do a restart immediately. | restart[enter] |
| set | Apply serial port settings (pacX and baudX). Not running this command after setting packer will allow configuration to be saved. Also serves the same purpose as the "CONVERSE" command in TNC2-modems. See separate chapter. | set[enter] |
| tdma-* | TDMA-related settings. See channel access chapter. | |
| transp-time | Timeout before data will be sent using the transparent packer. | transp-time 100[enter] |
| verbose | Verbose level for messages printed on ports configured for cmd packer. | verbose 5[enter] |

# 6 Connectors and pinout

## 6.1 Pinout

Table 4: Expansion connector (mates with Hirose DF11-16DS-2C)

| | | | |
|---|---|---|---|
| GND | 2 | 1 | Port 1 RX, 3.3 V |
| (GPIO) | 4 | 3 | Port 1 TX, 3.3 V |
| (GPIO) | 6 | 5 | (reserved UART RX or SDA) |
| Port 2 RX, 5 V tolerant | 8 | 7 | (reserved UART TX or SCL) |
| Port 2 TX, 3.3 V | 10 | 9 | (GPIO) |
| 3.3 V out | 12 | 11 | (GPIO) |
| DC in | 14 | 13 | DC in |
| GND | 16 | 15 | GND |

Table 5: Main connector (mates with JST PHR-5)

| | |
|---|---|
| 1 | Port 0 RX, RS-232 level |
| 2 | Port 0 TX, RS-232 level |
| 3 | GND |
| 4 | DC in |
| 5 | GND |

Table 6: Professional version connector (mates with male DE-9 connector)

| | | | |
|---|---|---|---|
| Port 2 TX | 1 | 6 | DC in |
| Port 0 TX, RS-232 level | 2 | 7 | (reserved) |
| Port 0 RX, RS-232 level | 3 | 8 | (reserved) |
| Port 2 RX, 5 V tolerant | 4 | 9 | GND |
| GND | 5 | | |

## 6.2 Connector part list

The connectors used on the different versions of Owl is widely available:

Table 7: Main connector

| Component | Vendor | Vendor PN | Mfg. | Mfg. PN |
|---|---|---|---|---|
| Connector housing | Digi-Key | 455-1163-ND | JST | PHR-5 |
| JST-PH crimp pins | Digi-Key | 455-1127-1-ND | JST | SPH-002T-P0.5S |
| Crimp tool | Digi-Key | 455-1128-ND | JST | WC-240 |
| Pre-terminated wires | Farnell | 2320543 | JST | 01SPHSPH-26001L300 |

Table 8: Expansion connector

| Component | Vendor | Vendor PN | Mfg. | Mfg. PN |
|---|---|---|---|---|
| Connector housing | Digi-Key | H2025-ND | Hirose | DF11-16DS-2C |
| Crimp pins (gold) | Digi-Key | H1505-ND | Hirose | DF11-2428SCA |
| Crimp pins (tin) | Digi-Key | H1504-ND | Hirose | DF11-2428SC |
| Crimp tool | Digi-Key | H9995-ND | Hirose | DF11-TA2428HC |
| Pre-terminated wires | Digi-Key | Search for "H3AXG" | - | - |

Table 9: Pro version D-sub

| Component | Vendor | Vendor PN | Mfg. | Mfg. PN |
|---|---|---|---|---|
| Solder cup connector | Digi-Key | AE10972-ND | Assmann | A-DS-09-LL/Z |
| Backshell | Digi-Key | 609-1424-ND | FCI | 86303637BLF |

The antenna connector uses a standard SMA connector. As for antennas, radio amateurs probably know a lot of alternatives for 2-meter antennas. For professional use, the AC Marine CX4-5 can be recommended. This is a very robust and well performing antenna with a wide useable frequency range.

## 6.3 Simple cable example

This example is the most basic cable for direct connection to the main port of the Owl. It will power the Owl and allow and RS-232 communication on port 0. For the 5-pin JST PH connector you can buy the pins and use a crimp tool as seen in the previous chapter, or simply buy some terminated wires.

Table 10: Bill of materials with vendor example

| Component | Qty. | Vendor | Vendor PN | Mfg. | Mfg. PN |
|---|---|---|---|---|---|
| 9-pin female DE9 | 1 | Farnell | 1084678 | Multicomp | 5501-09SA-02-F1 |
| Wires w/ JST-PH pins | 5 | Farnell | 2320543 | JST | 01SPHSPH-26001L300 |
| 5-pin JST-PH housing | 1 | Farnell | 3616216 | JST | PHR-5 |

Table 11: Connections

| From | To | Function |
|---|---|---|
| JST-PH pin 1 | D-sub pin 3 | RS-232 data to Owl |
| JST-PH pin 2 | D-sub pin 2 | RS-232 data from Owl |
| JST-PH pin 3 | D-sub pin 5 | Ground |
| JST-PH pin 4 | | Power V+ |
| JST-PH pin 5 | | Power ground |

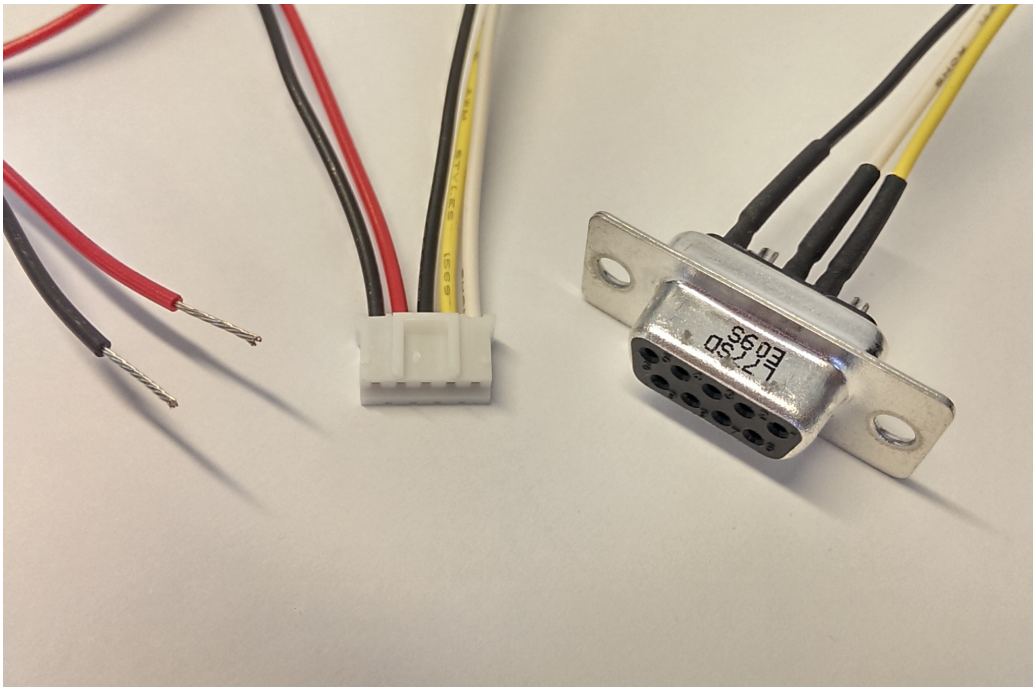


Figure 4: Simple RS-232 and power cable

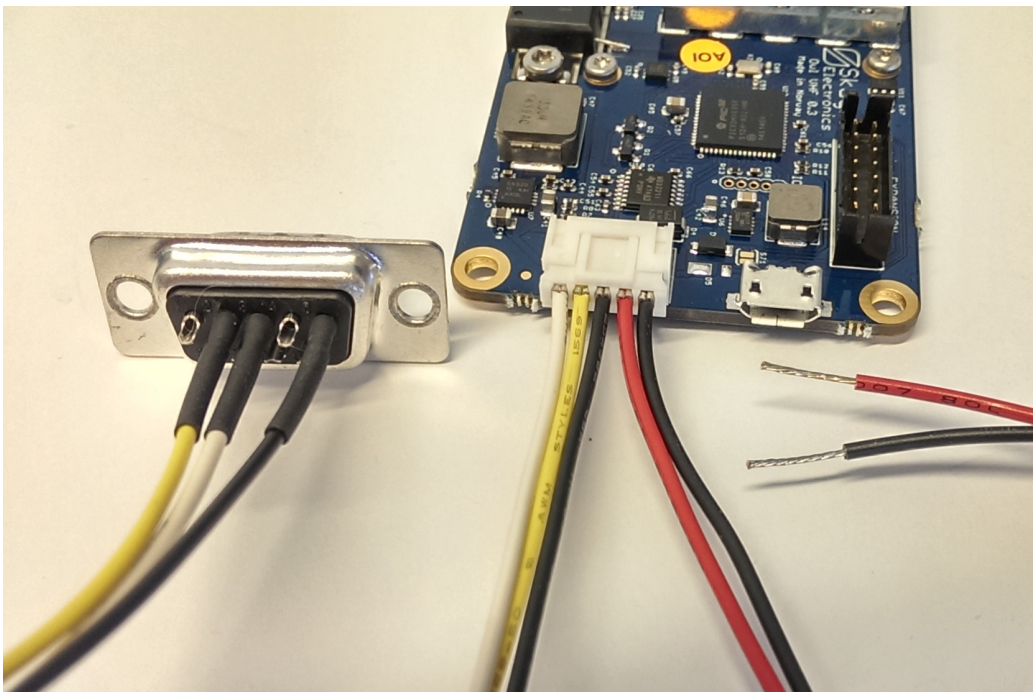


Figure 5: Simple RS-232 and power cable

## 6.4 USB cable for Owl professional

This is a cable example for the Owl professional, with two FTDI USB to serial adapters. One with RS-232 levels connected to port 0 on the Owl, and one with 3.3 V logic levels connected to port 2 on the Owl. This allows, for example, one port to be used for CLI / configuration, while the other is used for transparent data connection.

Table 12: Bill of materials with vendor example

| Component | Qty. | Vendor | Vendor PN | Mfg. | Mfg. PN |
|---|---|---|---|---|---|
| 9-pin male DE9 D-sub | 1 | Digi-Key | AE10972-ND | Assmann | A-DS-09-LL/Z |
| Backshell | 1 | Digi-Key | 609-1424-ND | FCI | 86303637BLF |
| USB-to-serial converter, 3.3 V level | 1 | Digi-Key | 768-1016-ND | FTDI | TTL-232R-3V3-WE |
| USB-to-serial converter, RS-232 | 1 | Digi-Key | 768-1065-ND | FTDI | USB-RS232-WE-1800-BT_0.0 |
| (Power wires) | | | | | |

Table 13: Connections

| From | To | Function |
|---|---|---|
| D-sub pin 1 | USB-TTL yellow | Port 2 TX |
| D-sub pin 2 | USB-RS232 yellow | Port 0 TX |
| D-sub pin 3 | USB-RS232 orange | Port 0 RX |
| D-sub pin 4 | USB-TTL orange | Port 2 RX |
| D-sub pin 5 | USB-TTL and USB-RS232 black | Ground |
| D-sub pin 6 | | Power V+ |
| D-sub pin 9 | | Power ground |

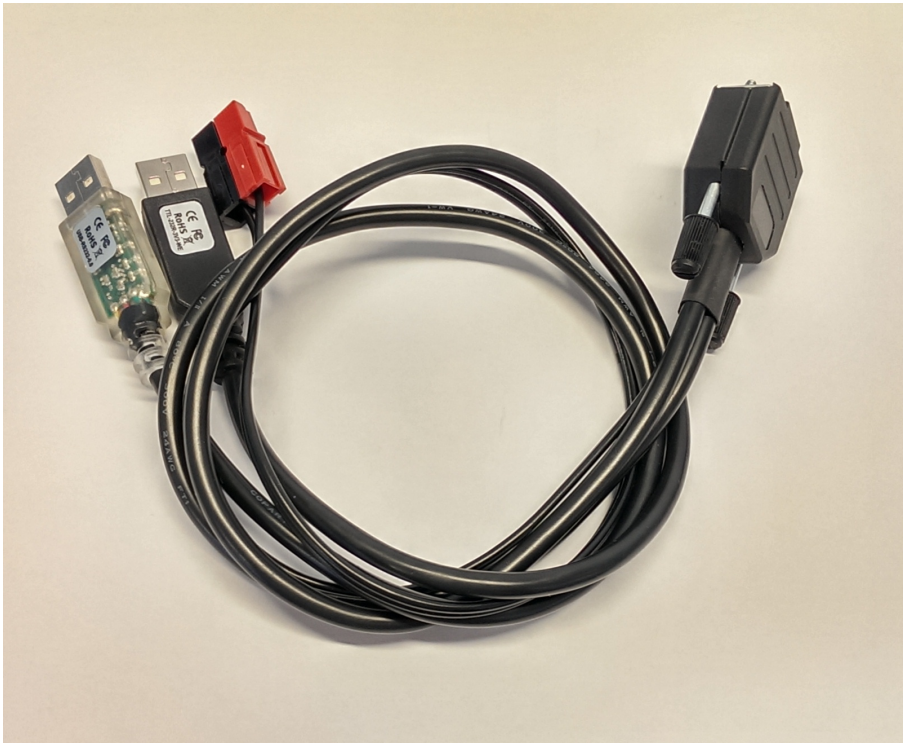


Figure 6: Dual USB and power cable

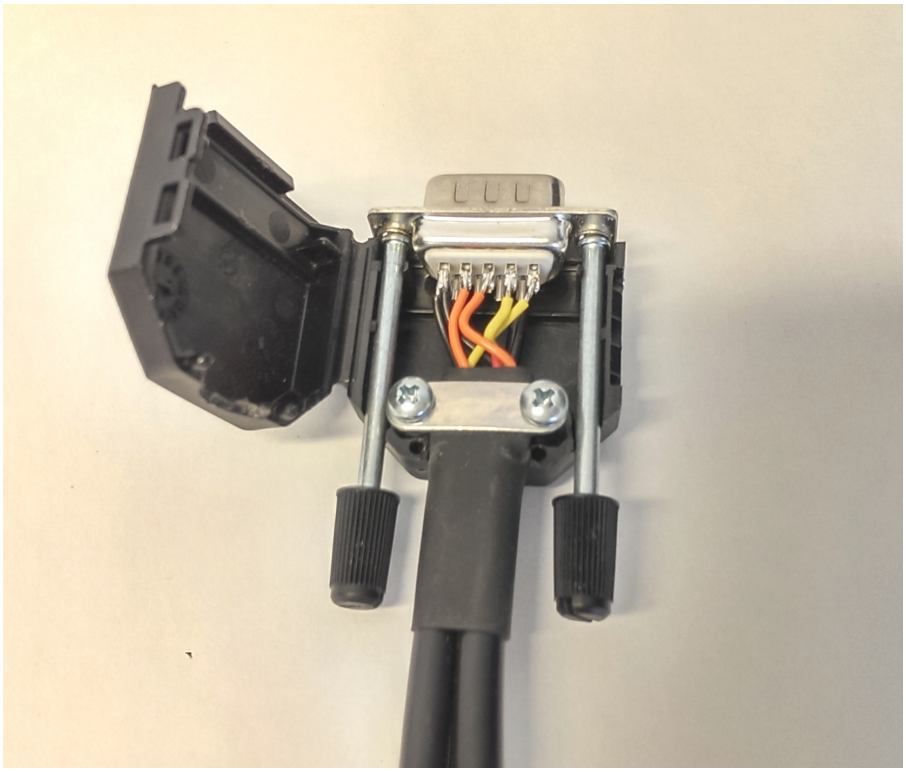


Figure 7: Dual USB and power cable

# 7 GPS connection

In many cases it is necessary to connect a GPS with a PPS output to the Owl:

- The Owl will be used as a standalone position tracker
- Multiple nodes are used in a large TDMA network where many nodes does not have a direct signal path
- Nodes are often out of range for longer periods and drifts out of sync
- Best timing accuracy is needed

The GPS must output NMEA 0183 data. TX and RX can be 3.3 V TTL, 5 V TTL or RS-232 levels, depending on which port you connect it to. The PPS output must be 3.3 V or 5 V level, and connected to expansion connector pin 9. Power to the GPS can be taken from the expansion header - either the 3.3 V output or the raw supply voltage. Up to 100 mA current on 3.3 V is fine.

Table 14: Example GPS connection on expansion connector

| | | | |
|---|---|---|---|
| GND | 2 | 1 | Port 1 RX, 3.3 V (GPS TX) |
| | 4 | 3 | Port 1 TX, 3.3 V (GPS RX) |
| | 6 | 5 | |
| | 8 | 7 | |
| | 10 | 9 | GPIO (GPS PPS out) |
| 3.3 V out (GPS power) | 12 | 11 | |
| DC in | 14 | 13 | DC in |
| GND | 16 | 15 | GND |

Enable GPS input on port 1 with this command: pac1 nmea[enter]

# 8 F.A.Q.

**Is there any known birdies in the receiver stage?** The noise floor on the Owl is generally very low, but the internal 40 MHz oscillator is present in the received signal at multiples of 40 MHz. In other words there exists a birdie at 160 MHz in the reception range of the Owl VHF which has a power level of approximately -105 dBm.

**Does it support flow control on the serial ports?** The OWL does not have any flow control on the serial ports. Traditionally, hardware flow control is used when buffers are small and overflow quickly. With large buffers however, it will result in a lot of data being in the buffer and delay communication as a lot of data will be queued the whole time. The Owl has large buffers (4 KB TX buffer), and it can still be increased if needed. It will transmit for several seconds to empty the whole buffer. If TDMA is used, I would recommend sending data chunks less than a time slot at a time to keep delay down. To see how many bytes can be transferred in a single time slot, run "info" and check the field "slot_b". Remember that this includes preamble etc. A 220 byte payload (maximum size) uses 262 bytes in the TX buffer.

When sending large amounts of data, I would generally suggest implementing a higher level protocol which sends a reasonable amount of data at a time (way less than the buffer size), and waits for an ACK before transmitting the next block of data. This will effectively keep the data transfer at "maximum speed". Small data blocks will make sure other real-time data can pass with minimal latency, and large data blocks will increase throughput because less time is used for acking.