Receivers (RX)

A receiver is used to receive radio control signals from your transmitter and convert them into signals that the flight controller can understand.

There are 3 basic types of receivers:

1. Parallel PWM Receivers
2. PPM Receivers
3. Serial Receivers

As of 2016 the recommendation for new purchases is a Serial or PPM based receiver. Avoid Parallel PWM recievers (1 wire per channel). This is due to the amount of IO pins parallel PWM based receivers use. Some new FC's do not support parallel PWM.

Parallel PWM Receivers

8 channel support, 1 channel per input pin. On some platforms using parallel input will disable the use of serial ports
and SoftSerial making it hard to use telemetry or GPS features.

PPM Receivers

PPM is sometimes known as PPM SUM or CPPM.

12 channels via a single input pin, not as accurate or jitter free as methods that use serial communications, but readily available.

These receivers are reported working:

- FrSky D4R-II (http://www.frsky-rc.com/product/pro.php?pro_id=24)
- Graupner GR24 (http://www.graupner.de/en/products/33512/product.aspx)
- R615X Spektrum/JR DSM2/DSMX Compatible 6Ch 2.4GHz Receiver w/CPPM (http://www.hobbyking.com/hobbyking/store/__46632__OrangeRx_R615X_DSM2_DSMX_C
- FrSky D8R-XP 8ch telemetry receiver, or CPPM and RSSI enabled receiver (http://www.frsky-rc.com/product/pro.php?pro_id=21)
- FrSky X4R and FrSky X4RSB (http://www.frsky-rc.com/download/view.php?sort=&down=158&file=X4R-X4RSB) when flashed with CPPM firmware and bound with jumper between signal pins 2 and 3
- All FrSky S.Bus enabled devices when connected with S.Bus CPPM converter cable (http://www.frsky-rc.com/product/pro.php?pro_id=112). Without jumper this converter cable uses 21ms frame size (Channels 1-8). When jumper is in place, it uses 28ms frame and channels 1-12 are available
- FlySky/Turnigy FS-iA4B, FS-iA6B, FS-iA10 receivers all provide 8channels if the tx is sending them. (FS-i6 and FS-i10 transmitters). Use setting rx-setup/ppm to enable.

Serial Receivers

Spektrum

8 channels via serial currently supported.

These receivers are reported working:

Lemon Rx DSMX Compatible PPM 8-Channel Receiver + Lemon DSMX Compatible Satellite with Failsafe
http://www.lemon-rx.com/index.php?route=product/product&product_id=118

S.BUS

16 channels via serial currently supported. See below how to set up your transmitter.

- You probably need an inverter between the receiver output and the flight controller. However, some flight controllers have this built in (the main port on CC3D, for example), and doesn't need one.
- Some OpenLRS receivers produce a non-inverted SBUS signal. It is possible to switch SBUS inversion off using CLI command `set sbus_inversion = OFF` when using an F3 based flight controller.
- Softserial ports cannot be used with SBUS because it runs at too high of a bitrate (1Mbps). Refer to the chapter specific to your board to determine which port(s) may be used.
- You will need to configure the channel mapping in the GUI (Receiver tab) or CLI (`map` command). Note that channels above 8 are mapped "straight", with no remapping.

These receivers are reported working:

FrSky X4RSB 3/16ch Telemetry Receiver
http://www.frsky-rc.com/product/pro.php?pro_id=135

FrSky X8R 8/16ch Telemetry Receiver
http://www.frsky-rc.com/product/pro.php?pro_id=105

Futaba R2008SB 2.4GHz S-FHSS
http://www.futaba-rc.com/systems/futk8100-8j/

OpenTX S.BUS configuration

If using OpenTX set the transmitter module to D16 mode and ALSO select CH1-16 on the transmitter before binding to allow reception
of all 16 channels.

OpenTX 2.09, which is shipped on some Taranis X9D Plus transmitters, has a bug - [issue:1701 (https://github.com/opentx/opentx/issues/1701)](https://github.com/opentx/opentx/issues/1701).
The bug prevents use of all 16 channels. Upgrade to the latest OpenTX version to allow correct reception of all 16 channels,
without the fix you are limited to 8 channels regardless of the CH1-16/D16 settings.

SRXL (formerly XBUS)

(Serial Receiver Link Protocol)
SRXL is an open data transfer protocol which allows to transport control data from a rc receiver to another device like a flybarless system
by only using one single line. This protocol has been established by SRXL.org based on the idea to create a freely available and unified protocol

that manufacturers can easily implement to their receivers and devices that process receiver data. The protocol does not describe an exact definition of
how the data must be processed. It only describes a framework in which receiver data can be packed. Each manufacturer can have his own ID, which must be
attached to the beginning of each data set, so that the device using this data can correctly identify and process the payload of the dataset.

Supported receivers:

Multiplex:

All receivers with SRXL (also FLEXX receivers)

####Gaupner / SJ HOTT:
All receiver with SUMD support

Spektrum:

AR7700 / AR9020 receiver

JR:

JR X-BUS
Make sure to set your TX to use "MODE B" for XBUS in the TX menus!
See here for info on JR's XBUS protocol: http://www.jrpropo.com/english/propo/XBus/
These receivers are reported working:

XG14 14ch DMSS System w/RG731BX XBus Receiver
http://www.jramericas.com/233794/JRP00631/

Jeti:

Receivers with UDI output

XBUS MODE B RJ01

There exist a remote receiver made for small BNF-models like the Align T-Rex 150 helicopter. The code also supports using the Align DMSS RJ01 receiver directly with the cleanflight software.
To use this receiver you must power it with 3V from the hardware, and then connect the serial line as other serial RX receivers.
In order for this receiver to work, you need to specify the XBUS_MODE_B_RJ01 for serialrx_provider. Note that you need to set your radio mode for XBUS "MODE B" also for this receiver to work.
Receiver name: Align DMSS RJ01 (HER15001)

SUMD

16 channels via serial currently supported.

These receivers are reported working:

GR-24 receiver HoTT
http://www.graupner.de/en/products/33512/product.aspx

Graupner receiver GR-12SH+ HoTT
http://www.graupner.de/en/products/870ade17-ace8-427f-943b-657040579906/33565/product.aspx

SUMH

8 channels via serial currently supported.

SUMH is a legacy Graupner protocol. Graupner have issued a firmware updates for many recivers that lets them use SUMD instead.

IBUS

10 channels via serial currently supported.

IBUS is the FlySky digital serial protocol and is available with the FS-IA4B, FS-IA6B and FS-IA10 receivers. The Turnigy TGY-IA6B and TGY-IA10 are the same devices with a different label, therefore they also work.

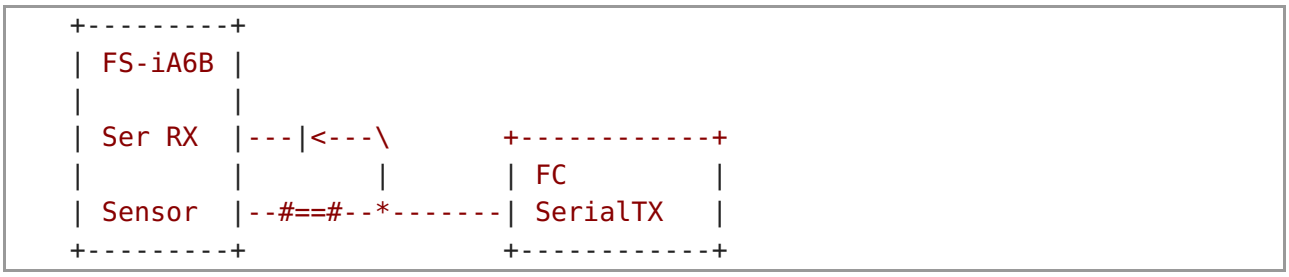
If you are using a 6ch tx such as the FS-I6 or TGY-I6 then you must flash a 10ch firmware on the tx to make use of these extra channels.

These receivers are reported working (all gives 10 channels serial):

- FlySky/Turnigy FS-iA4B 4-Channel Receiver (http://www.flysky-cn.com/products_detail/productId=46.html)
- FlySky/Turnigy FS-iA6B 6-Channel Receiver (http://www.flysky-cn.com/products_detail/&productId=51.html)
- FlySky/Turnigy FS-iA10 10-Channel Receiver (http://www.flysky-cn.com/products_detail/productId=53.html)
- FlySky/Turnigy FS-iA10B 10-Channel Receiver (http://www.flysky-cn.com/products_detail/productId=52.html)

Combine flysky ibus telemetry and serial rx on the same FC serial port

Connect Flysky FS-iA6B receiver like this:



Use a diode with cathode to receiver serial rx output (for example 1N4148), the anode is connected to the FC serial *TX* pin, and also via a resistor (10KOhm) to the receiver ibus sensor port.

Note (2018-07-27): In some cases, the value of the series resistor may be too large, and going down to 1K[ohm] may provide a good result.

Enable with cli:

```
serial 1 1088 115200 57600 115200 115200
feature RX_SERIAL
set serialrx_provider = IBUS
save
```

MultiWii serial protocol (MSP)

Allows you to use MSP commands as the RC input. Only 8 channel support to maintain compatibility with MSP.

Configuration

There are 3 features that control receiver mode:

```
RX_PPM
RX_SERIAL
RX_PARALLEL_PWM
RX_MSP
```

Only one receiver feature can be enabled at a time.

RX signal-loss detection

The software has signal loss detection which is always enabled. Signal loss detection is used for safety and failsafe reasons.

The `rx_min_usec` and `rx_max_usec` settings helps detect when your RX stops sending any data, enters failsafe mode or when the RX looses signal.

By default, when the signal loss is detected the FC will set pitch/roll/yaw to the value configured for `mid_rc`. The throttle will be set to the value configured for `rx_min_usec` or `mid_rc` if using 3D feature.

Signal loss can be detected when:

1. no rx data is received (due to radio reception, recevier configuration or cabling issues).
2. using Serial RX and receiver indicates failsafe condition.
3. using any of the first 4 stick channels do not have a value in the range specified by `rx_min_usec` and `rx_max_usec`.

RX loss configuration

The `rxfail` cli command is used to configure per-channel rx-loss behaviour.
You can use the `rxfail` command to change this behaviour.
A flight channel can either be AUTOMATIC or HOLD, an AUX channel can either be SET or HOLD.

- AUTOMATIC - Flight channels are set to safe values (low throttle, mid position for yaw/pitch/roll).
- HOLD - Channel holds the last value.
- SET - Channel is set to a specific configured value.

The default mode is AUTOMATIC for flight channels and HOLD for AUX channels.

The rxfail command can be used in conjunction with mode ranges to trigger various actions.

The `rxfail` command takes 2 or 3 arguments.

- Index of channel (See below)
- Mode ('a' = AUTOMATIC, 'h' = HOLD, 's' = SET)
- A value to use when in SET mode.

Channels are always specified in the same order, regardless of your channel mapping.

- Roll is 0
- Pitch is 1
- Yaw is 2
- Throttle is 3.
- Aux channels are 4 onwards.

Examples:

To make Throttle channel have an automatic value when RX loss is detected:

```
rxfail 3 a
```

To make AUX4 have a value of 2000 when RX loss is detected:

```
rxfail 7 s 2000
```

To make AUX8 hold it's value when RX loss is detected:

```
rxfail 11 h
```

WARNING: Always make sure you test the behavior is as expected after configuring rxfail settings!

`rx_min_usec`

The lowest channel value considered valid. e.g. PWM/PPM pulse length

`rx_max_usec`

The highest channel value considered valid. e.g. PWM/PPM pulse length

Serial RX

See the Serial chapter for some some RX configuration examples.

To setup spectrum on the Naze32 or clones in the GUI:

1. Start on the "Ports" tab make sure that UART2 has serial RX. If not set the checkbox, save and reboot.
2. Move to the "Configuration" page and in the upper lefthand corner choose Serial RX as the receiver type.
3. Below that choose the type of serial receiver that you are using. Save and reboot.

Using CLI:
For Serial RX enable RX_SERIAL and set the `serialrx_provider` CLI setting as follows.

Serial RX Provider	Value
SPEKTRUM1024	0

SPEKTRUM2048	1
SBUS	2
SUMD	3
SUMH	4
XBUS_MODE_B	5
XBUS_MODE_B_RJ01	6
IBUS	7

PPM/PWM input filtering.

Hardware input filtering can be enabled if you are experiencing interference on the signal sent via your PWM/PPM RX.

Use the `input_filtering_mode` CLI setting to select a mode.

Value	Meaning
OFF	Disabled
ON	Enabled

Receiver configuration.

FrSky D4R-II

Set the RX for 'No Pulses'. Turn OFF TX and RX, Turn ON RX. Press and release F/S button on RX. Turn off RX.

Graupner GR-24 PWM

Set failsafe on the throttle channel in the receiver settings (via transmitter menu) to a value below `rx_min_usec` using channel mode FAILSAFE.
This is the prefered way, since this is *much faster* detected by the FC then a channel that sends no pulses (OFF).

NOTE:
One or more control channels may be set to OFF to signal a failsafe condition to the FC, all other channels *must* be set to either HOLD or OFF.
Do **NOT USE** the mode indicated with FAILSAFE instead, as this combination is NOT handled correctly by the FC.

Receiver Channel Range Configuration.

The channels defined in CleanFlight are as follows:

Channel number	Channel name
0	Roll
1	Pitch
2	Yaw
3	Throttle

If you have a transmitter/receiver, that output a non-standard pulse range (i.e. 1070-1930 as some Spektrum receivers)
you could use rx channel range configuration to map actual range of your transmitter to 1000-2000 as expected by Cleanflight.

The low and high value of a channel range are often referred to as 'End-points'. e.g. 'End-point adjustments / EPA'.

All attempts should be made to configure your transmitter/receiver to use the range 1000-2000 *before* using this feature
as you will have less preceise control if it is used.

To do this you should figure out what range your transmitter outputs and use these values for rx range configuration.
You can do this in a few simple steps:

If you have used rc range configuration previously you should reset it to prevent it from altering rc input. Do so
by entering the following command in CLI:

```
rxrange reset
save
```

Now reboot your FC, connect the configurator, go to the `Receiver` tab move sticks on your transmitter and note min and
max values of first 4 channels. Take caution as you can accidentally arm your craft. Best way is to move one channel at
a time.

Go to CLI and set the min and max values with the following command:

```
rxrange <channel_number> <min> <max>
```

For example, if you have the range 1070-1930 for the first channel you should use `rxrange 0 1070 1930` in
the CLI. Be sure to enter the `save` command to save the settings.

After configuring channel ranges use the sub-trim on your transmitter to set the middle point of pitch, roll, yaw and throttle.

You can also use rxrange to reverse the direction of an input channel, e.g. `rxrange 0 2000 1000`.