Telemetry

Telemetry allows you to know what is happening on your aircraft while you are flying it. Among other things you can receive battery voltages and GPS positions on your transmitter.

Telemetry can be either always on, or enabled when armed. If a serial port for telemetry is shared with other functionality then telemetry will only be enabled when armed on that port.

Telemetry is enabled using the 'TELEMETRY` feature.

```
feature TELEMETRY
```

Multiple telemetry providers are currently supported, FrSky, Graupner
HoTT V4, SmartPort (S.Port), LightTelemetry (LTM) and Ibus

All telemetry systems use serial ports, configure serial ports to use the telemetry system required.

FrSky telemetry

FrSky telemetry is transmit only and just requires a single connection from the TX pin of a serial port to the RX pin on an FrSky telemetry receiver.

FrSky telemetry signals are inverted. To connect a cleanflight capable board to an FrSKy receiver you have some options.

1. A hardware inverter - Built in to some flight controllers.
2. Use software serial and enable frsky_inversion.
3. Use a flight controller that has software configurable hardware inversion (e.g. STM32F30x).

For 1, just connect your inverter to a usart or software serial port.

For 2 and 3 use the CLI command as follows:

```
set tlm_inversion = ON
```

Available sensors

The following sensors are transmitted :

Vspd : vertical speed, unit is cm/s.

Hdg : heading, North is 0°, South is 180°.

AccX,Y,Z : accelerometers values.

Tmp1 : baro temp if available, gyro otherwise.

RPM : if armed : throttle value, battery capacity otherwise. (Blade number needs to be set to 12 in Taranis).

VFAS : actual vbat value (see VFAS precision section bellow).

Curr : actual current comsuption, in amp.

Fuel : if capacity set :remaining battery percentage, mah drawn otherwise.

GPS : GPS coordinates.

Alt : barometer based altitude, init level is zero.

Date : time since powered.

GSpd : current speed, calculated by GPS.

GAlt : GPS altitude, sea level is zero.

Tmp2 : number of sats. Every second, a number > 100 is sent to represent GPS signal quality.

Cels : average cell value, vbat divided by cell number.

> Cleanflight will send Cels (FLVSS Individual Cell Voltages Telemetry), disable the setting to use actual FLVSS sensor with:
>
> ```
> set telemetry_send_cells = OFF
> ```
>
> Note: cell voltage values are an assumed reputation of the cell voltage based on the packs voltage. Actual cell voltage may differ.
>
> To view individual cells or more importantly to get lowest cell (all cells are the sum of vbat, so each cell is the same in this case):
> See [OpenTX 2.1 & FrSky FLVSS Individual Cell Voltages (http://openrcforums.com/forum/viewtopic.php?t=7266)](http://openrcforums.com/forum/viewtopic.php?t=7266).
> Add a new sensor, to display the lowest cell voltage set it up like this:
>
> - Type: Calculated
> - Formula: Cell
> - Cell Sensor: Cels *(pack total voltage, sum of all cells)*
> - Cell Index: Lowest

Precision setting for VFAS

Cleanflight can send VFAS (FrSky Ampere Sensor Voltage) in two ways:

```
set frsky_vfas_precision  = 0
```

This is default setting which supports VFAS resolution of 0.2 volts and is supported on all FrSky hardware.

```
set frsky_vfas_precision  = 1
```

This is new setting which supports VFAS resolution of 0.1 volts and is only supported by OpenTX radios (this method uses custom ID 0x39).

HoTT telemetry

Only Electric Air Modules and GPS Modules are emulated.

Use the latest Graupner firmware for your transmitter and receiver.

Older HoTT transmitters required the EAM and GPS modules to be enabled in the telemetry menu of the transmitter. (e.g. on MX-20)

You can connect HoTT-Telemetry in two ways:

Old way:

Serial ports use two wires but HoTT uses a single wire so some electronics are required so that the signals don't get mixed up. The TX and RX pins of
a serial port should be connected using a diode and a single wire to the T port on a HoTT receiver.

Connect as follows:

- HoTT TX/RX T -> Serial RX (connect directly)
- HoTT TX/RX T -> Diode -(|)- > Serial TX (connect via diode)

The diode should be arranged to allow the data signals to flow the right way

```
-(  |)- == Diode, | indicates cathode marker.
```

1N4148 diodes have been tested and work with the GR-24.

When using the diode disable `tlm_halfduplex`, go to CLI and type `set tlm_halfduplex = OFF`, don't forget a `save` afterwards.

New way:

You can use a single connection, connect HoTT RX/TX only to serial TX, leave serial RX open and make sure `tlm_halfduplex` is ON.

As noticed by Skrebber the GR-12 (and probably GR-16/24, too) are based on a PIC 24FJ64GA-002, which has 5V tolerant digital pins.

Note: The SoftSerial ports may not be 5V tolerant on your board. Verify if you require a 5v/3.3v level shifters.

LightTelemetry (LTM)

LTM is a lightweight streaming telemetry protocol supported by a
number of OSDs, ground stations and antenna trackers.

The Cleanflight implementation of LTM implements the following frames:

- G-FRAME: GPS information (lat, long, ground speed, altitude, sat info)
- A-FRAME: Attitude (pitch, roll, heading)
- S-FRAME: Status (voltage, current+, RSSI, airspeed+, status). Item suffixed '+' not implemented in Cleanflight.
- O-FRAME: Origin (home position, lat, long, altitude, fix)

In addition, in the inav (navigation-rewrite) fork:

- N-FRAME: Navigation information (GPS mode, Nav mode, Nav action, Waypoint number, Nav Error, Nav Flags).

LTM is transmit only, and can work at any supported baud rate. It is designed to operate over 2400 baud (9600 in Cleanflight) and does not benefit from higher rates. It is thus usable on soft serial.

More information about the fields, encoding and enumerations may be found at
https://github.com/stronnag/mwptools/blob/master/docs/ltm-definition.txt

MAVLink telemetry

MAVLink is a very lightweight, header-only message marshalling library for micro air vehicles.
Cleanflight supports MAVLink for compatibility with ground stations, OSDs and antenna trackers built
for PX4, PIXHAWK, APM and Parrot AR.Drone platforms.

MAVLink implementation in Cleanflight is transmit-only and usable on low baud rates and can be used over soft serial.

SmartPort (S.Port)

Smartport is a telemetry system used by newer FrSky transmitters and receivers such as the Taranis/XJR and X8R, X6R and X4R(SB).

More information about the implementation can be found here:
https://github.com/frank26080115/cleanflight/wiki/Using-Smart-Port

Available sensors

The following sensors are transmitted :

A4 : average cell value. Warning : unlike FLVSS sensors, you do not get actual lowest value of a cell, but an average : (total lipo voltage) / (number of cells)

Alt : barometer based altitude, init level is zero.

Vspd : vertical speed, unit is cm/s.

Hdg : heading, North is 0°, South is 180°.

AccX,Y,Z : accelerometers values.

Tmp1 : actual flight mode, sent as 4 digits. Number is sent as (1)1234. Please ignore the leading 1, it is just there to ensure the number as always 5 digits (the 1 + 4 digits of actual data) the numbers are aditives (for example, if first digit after the leading 1 is 6, it means GPS Home and Headfree are both active) :

1. 1 is GPS Hold, 2 is GPS Home, 4 is Headfree
2. 1 is mag enabled, 2 is baro enabled, 4 is sonar enabled
3. 1 is angle, 2 is horizon, 4 is passthrough
4. 1 is ok to arm, 2 is arming is prevented, 4 is armed

Tmp2 : GPS lock status, Number is sent as 1234, the numbers are aditives :

1. 1 is GPS Fix, 2 is GPS Home fix
2. not used
3. not used
4. number of sats

VFAS : actual vbat value.

GAlt : GPS altitude, sea level is zero.

GSpd : current speed, calculated by GPS.

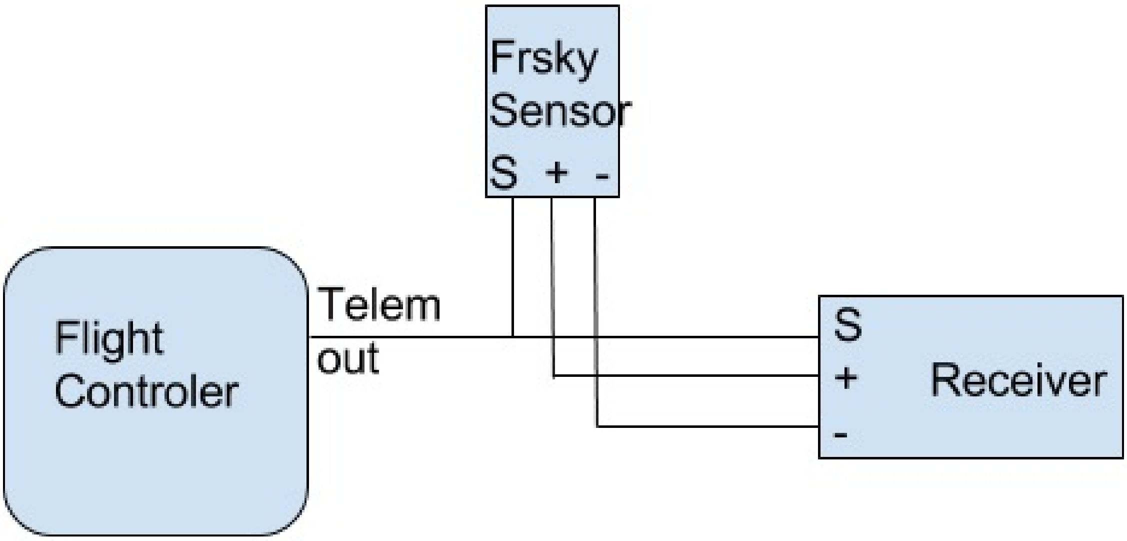
GPS : GPS coordinates.

Cels : average cell value, vbat divided by cell number.

> Cleanflight will send Cels (FLVSS Individual Cell Voltages Telemetry), disable the setting to use actual FLVSS sensor with:
>
> ```
> set telemetry_send_cells = OFF
> ```
>
> Note: cell voltage values are an assumed reputation of the cell voltage based on the packs voltage. Actual cell voltage may differ. It is recommeded that you chain the flight controllers telemetry with a real Frsky FLVSS s.port sensor.
>
> To view individual cells or more importantly to get lowest cell (all cells are the sum of vbat, so each cell is the same in this case):
> See OpenTX 2.1 & FrSky FLVSS Individual Cell Voltages (http://openrcforums.com/forum/viewtopic.php?t=7266).
> Add a new sensor, to display the lowest cell voltage set it up like this:
>
> - Type: Calculated
> - Formula: Cell
> - Cell Sensor: Cels *(pack total voltage, sum of all cells)*
> - Cell Index: Lowest

Integrate Cleanflight telemetry with FrSky Smartport sensors

While Cleanflight telemetry brings a lot of valuable data to the radio, there are additional sensors, like Lipo cells sensor FLVSS, that can be a great addition for many aircrafts. Smartport sensors are designed to be daisy chained, and CF telemetry is no exception to that. To add an external sensor, just connect the "S" port of the FC and sensor(s) together, and ensure the sensor(s) are getting connected to GND and VCC either from the controler or the receiver



SmartPort on F3 targets with hardware UART

Smartport devices can be connected directly to STM32F3 boards such as the SPRacingF3 and Sparky, with a single straight through cable without the need for any hardware modifications on the FC or the receiver. Connect the TX PIN of the UART to the Smartport signal pin.

For Smartport on F3 based boards, enable the telemetry inversion setting.

```
set tlm_inversion = ON
```

SmartPort on F1 and F3 targets with SoftSerial

Since F1 targets like Naze32 or Flip32 are not equipped with hardware inverters, SoftSerial might be simpler to use.

1. Enable SoftSerial `feature SOFTSERIAL`
2. In Configurator assign *Telemetry* > *Smartport* > *Auto* to SoftSerial port of your choice
3. Enable Telemetry `feature TELEMETRY`
4. Confirm telemetry inversion `set tlm_inversion = ON`
5. You have to bridge TX and RX lines of SoftSerial and connect them together to S.Port signal line in receiver

Notes:

- This has been tested with Flip32 and SPracingF3 boards and FrSky X8R and X4R receivers
- To discover all sensors board has to be armed, and when GPS is connected, it needs to have a proper 3D fix. When not armed, values like ***Vfas*** or GPS coordinates may not sent.

Ibus telemetry

Ibus telemetry requires a single connection from the TX pin of a bidirectional serial port to the Ibus sens pin on an FlySky telemetry receiver. (tested with fs-iA6B receiver, iA10 should work)

It shares 1 line for both TX and RX, the rx pin cannot be used for other serial port stuff.
It runs at a fixed baud rate of 115200.

```
     _______
    /       \                                             /---------\
    | STM32 |--UART TX-->[Bi-directional @ 115200 baud]<--| IBUS RX |
    |  uC   |--UART RX--x[not connected]                  \---------/
    \_______/
```

It should be possible to daisy chain multiple sensors with ibus. This is implemented but not tested because i don't have one of the sensors to test with, the FC shall always be the last sensor in the chain.

It is possible to combine serial rx and ibus telemetry on the same uart pin on the flight controller, see Rx (Rx.md).

Configuration

Ibus telemetry can be enabled in the firmware at build time using defines in target.h. It is enabled by default in those targets that have space left.

```
#define TELEMETRY
#define TELEMETRY_IBUS
```

CLI command to enable:

```
serial 1 1024 115200 57600 115200 115200
```

CLI setting to determine if the voltage reported is Vbatt or calculated average cell voltage

```
set ibus_report_cell_voltage=[ON/OFF]
```

Available sensors

The following sensors are transmitted :

Tmp1 : baro temp if available, gyro otherwise.

RPM : throttle value

Vbatt : configurable battery voltage or the average cell value, vbat divided by number of cells.

RX hardware

These receivers are reported to work with i-bus telemetry:

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

Note that the FlySky/Turnigy FS-iA4B 4-Channel Receiver (http://www.flysky-cn.com/products_detail/productId=46.html) seems to work but has a bug that might lose the binding, DO NOT FLY the FS-iA4B!