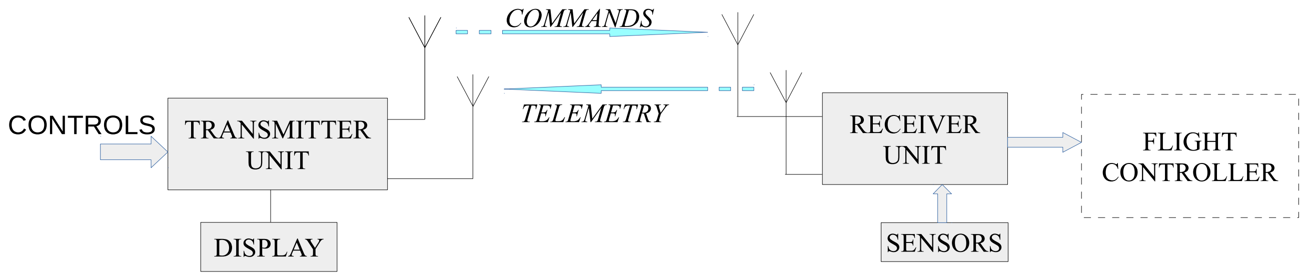
Notes on the FlySky FS-i6

These notes describe aspects of the FlySky FS-i6 transmitter along with the FS-iA6B receiver as they related to their use for control of multirotors. The Turnigy TGY-6 system is similar.

Overview

At its most basic level, this radio control system consists of two units, a transmitter unit and a receiver unit. These names are confusing however since, in order to provide for telemetry, each unit contains both a transmitter and a receiver.



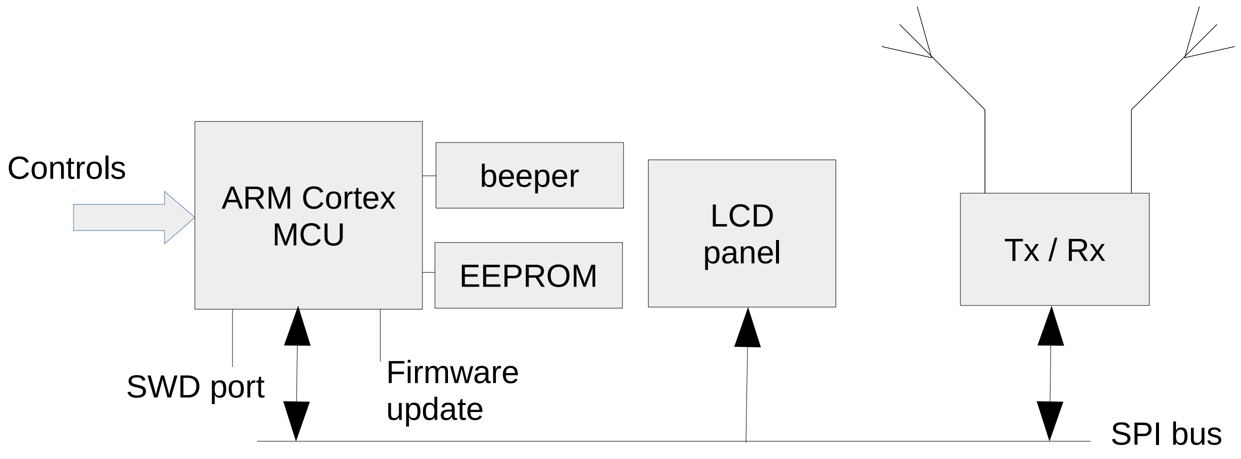
The transmitter unit takes as inputs the various controls (sticks, switches and rotary controls) and transmits their settings to the receiver unit which, in turns, passes this information on the flight controller. The receiver unit, which interfaces with one or more sensors (for instance, for measuring battery voltages), transmits their values back to the transmitter unit which, in turn, can display them to the user.

The radio signals between the two units are transmitted in the 2.4 GHz ISM band (intended for unlicensed use by miscellaneous Industrial, Scientific and Medical users). The system is designed to minimise the effects of interference from other devices (everything from microwave ovens to WiFi LANs) operating in this band, as well as from other radio control units in the vicinity. It achieves this by using frequency hopping so as to lessen the chance of the signal being lost and an elaborate error detection and correction encoding to lessen the chance of spurious signals being treated as valid.

The Transmitter unit

The Transmitter unit consists of a combined transmitter/receiver module operating within the 2.4 GHz band and a microcontroller to which are connected the various controls (sticks, switches, etc). Intercommunication between the major units (the LCD panel and the Tx/Rx module) takes place over an SPI bus ("Serial Peripheral Interconnect", a widely used industrial standard).

The unit is battery powered (nominal 6V, max 6.5V, and with low-voltage alarms at 4.2V and 4.0V). With the LCD panel backlight on, the overall unit draws about 135mA, with it off this falls to about 110mA.



The Microcontroller

The heart of the unit is an ARM Cortex M0+ microcontroller (type MKL16Z64VLH4). This 64-pin surface-mount chip handles 32-bit words at a clock rate of 48 MHz. It incorporates 8 MB of RAM and 64 kB of flash memory, a 16-bit ADC and various interfaces including:

- SPI, a 3-wire serial bus used to communicate with the LCD panel and the Tx/Rx module.
- I2C (Inter Integrated Circuit), a 2-wire bus for communicating with the 16 kB EEPROM chip (type 24C128) that provides persistent memory for holding the user's settings (for the system parameters and the model-specific parameters for up to 20 models).
- Asynchronous serial (used for the Firmware update port).
- SWD (Serial Wire Debug, a 2-wire ARM proprietary bus) that allows R/W access to the internal state of the microcontroller.

The roles played the microcontroller include:

- Digitising the analog inputs from the control sticks and rotary controls, and then scaling and combining their inputs in accordance with the transfer functions defined by the user.
- Handling user interaction and controlling the LCD panel in its various modes.
- Controlling the frequency-hopping tx/rx (including handling the binding process).
- Handling the protocol (but not error detection and correction) associated with the bit-streams to and from the tx/rx module.

- Processing and displaying the returned telemetry data
- Handling firmware updates (the bootloader section of the memory is protected: it cannot itself be corrupted by firmware updates).

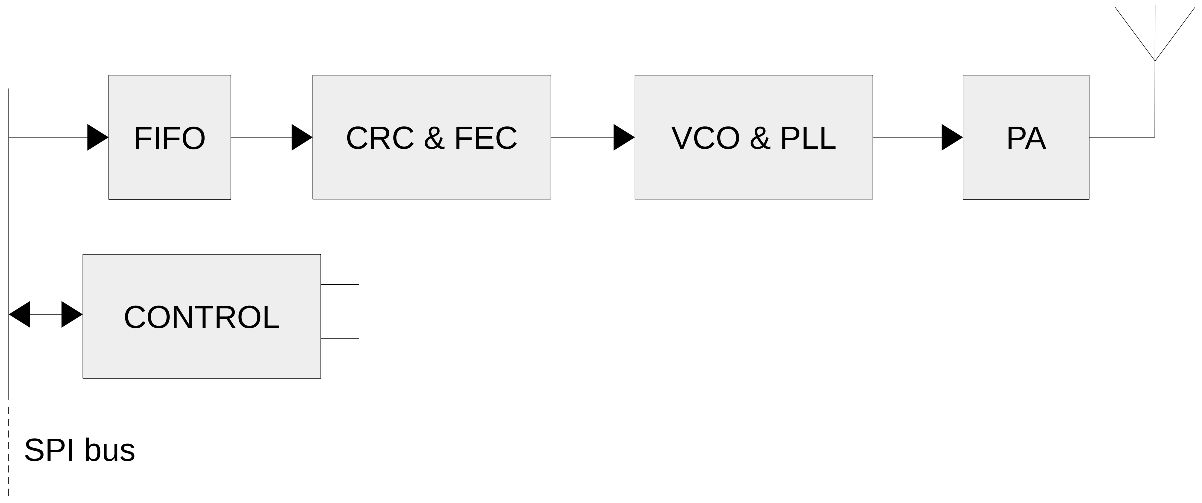
The Transceiver module

The 2.4 GHz transceiver in the "Transmitter" unit is implemented using the type A7105 module manufactured by AMIC Communication Corp. This highly sophisticated module is interfaced to the SPI bus which handles both data and control communication with the microcontroller.

The transceiver module with its coaxial feeds to the two antennae. The A7105 integrated circuit is under this metal cover (which shields it from electrical noise from the rest of the system).

The transmitter

The major elements of the transmitter portion of the A7105 transceiver are:



The data stream to be transmitted is drawn from the SPI bus and buffered with a 64-byte FIFO. It is formed into packets, with an error detecting/correcting checksum included. The resultant bit stream is then used to frequency-shift modulate a 2.4 GHz carrier. This carrier is synthesised by a Voltage Controlled Oscillator with Phase Locked Loop feedback. The centre frequency of this oscillator for each "hop" within the frequency band is specified by data from the SPI bus.

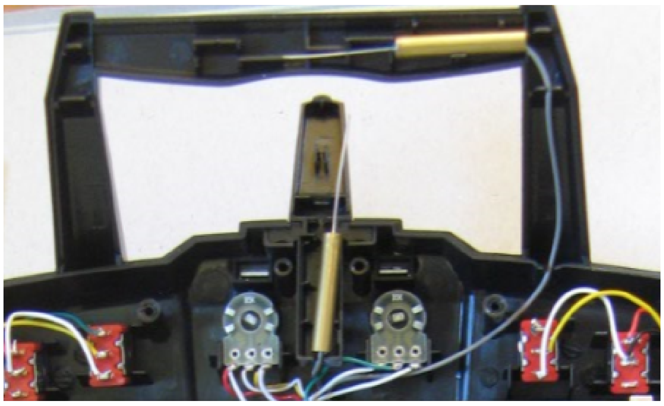
The FlySky specification quotes a transmitter frequency range of 2.405 to 2.475 GHz, that is, a total bandwidth of 70 MHz.

The modulation scheme used is a Frequency Shift Keying (with a frequency shift of 186kHz). The modulating signal is smoothed to give a Gaussian step response in order to limit the splatter outside the frequency band that would be caused by abrupt changes in the carrier frequency.

The resultant modulated UHF signal is fed into a Power Amplifier and thence, via a fine gauge coaxial cable, to the transmitting antenna. The FlySky manual states that the output power is less than 20dBm. That is, it is a factor of 100 greater than the 1mW reference level or, equivalently stated, is less than 0.1W.

The antenna

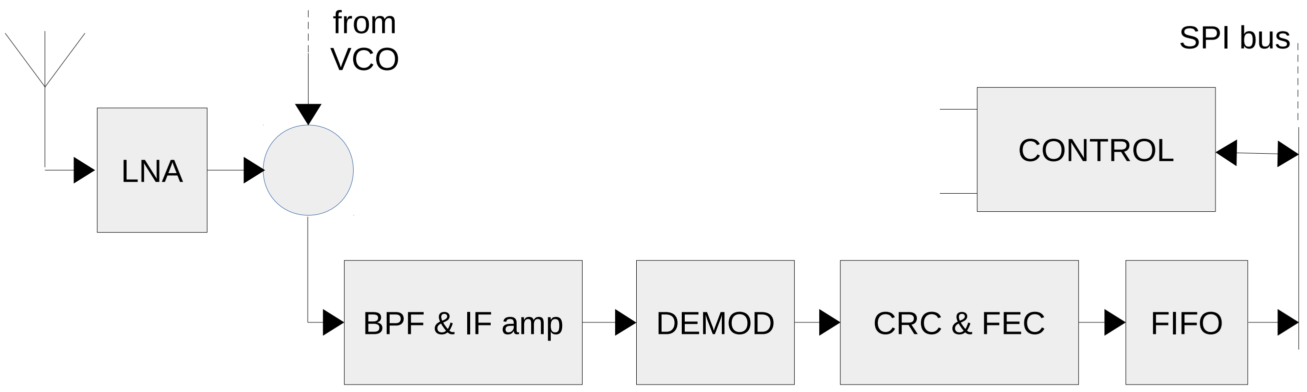
The wavelength of a 2.4 GHz signal is 12.5cm and so an ideal dipole antenna would be about 6.2 cm in length. The FlySky unit uses a sleeve dipole arrangement (see photo) in which one half of the dipole takes the form of a tube through which the coaxial feed is threaded, with the other half of the dipole being a conventional wire "whip".



The transmitting and receiving dipole antennae in the FlySky transmitter unit: one is inside the protruding stub, the other (at right angles to it) inside the handle.

The receiver

The major elements of the receiver portion of the A7105 transceiver are:



The UHF signal from the receiving antenna is fed into a Low Noise Amplifier and from there it is mixed with a local oscillator signal (generated by the frequency-hopping VCO described earlier).

The resultant intermediate frequency signal is fed through a Band Pass Filter and into the Intermediate Frequency amplifier. An FSK demodulator recovers the digital bit stream which is then error checked / corrected and, if found to be error free, is fed into the FIFO buffer from where the telemetry information it contains can be accessed via the SPI bus.

The specification for the A7105 module states that the sensitivity of its receiver depends upon the bandwidth to which it is set. It quotes a figure of -107 dBm for a 2 kbps signal and -104 dBm for a 250 kbps one. The FlySky specification quotes a receiver sensitivity of – 105 dBm.

The A7105 module generates an 8-bit RSSI (Received Signal Strength Indicator) but, sadly, this is not displayed to the user.

Control aspects

The A7105 module is highly parametrisable: it contains some 52 8-bit registers that are accessible via the SPI bus.

The transceiver can be switched between half a dozen different states, including principally a Transmit mode and a Receive mode (it cannot simultaneously both transmit and receive).

Data is transmitted in packet form. Each packet consists of:

- a 4-byte preamble (for synchronisation)
- a 4-byte identifier
- the payload
- a 2-byte CRC checksum (CCIT-16)

The Forward Error Correction mode causes each 4-bit nibble of the payload to be encoded as a 7-bit word using a (7, 4) Hamming FEC scheme. This enables single-bit errors to be corrected (at the cost of doubling the size of the payload).

Each FlySky transmitter unit has been given a unique 4 byte identifier. This identifier is included in every packet transmitted and it is also (during the binding process) conveyed to the bound receiver. The identifier in each packet that is received is checked against this reference identifier and, only if the two match, is the packet passed to the FIFO buffer.

Frequency hopping

The FlySky user manual states that the frequency-hopping scheme that is used defines some 142 channels within the 70 MHz band available of which it actually uses 16 of these. It also states that it has a repertoire of some 160 different hopping sequences. The choice of hopping sequence used by any particular transmitter unit is computed from its unique 4-byte identifier (thus ensuring that the bound receiver selects the same sequence).

The payload

The payload contains the information about the positions of the various controls; it is conveyed over the SPI bus from the microcontroller to the A7105 transmitter FIFO. It contains a sequence of some 14 byte pairs, one pair (msb first) for each control channel. (The FS-i6 system utilises only the first 6 of these control channels; the remaining 8 carry dummy information. Firmware hacks allow these extra control channels to be used constructively. . .). The payload also includes a byte that specifies the index of the (radio frequency) channel (out of the 16 channels that are used) that the frequency is to be hopped to for transmission of that packet.

Packets derived from the payload are transmitted at a rate of one every 3.8ms.

User interface

The FlySky FS i6 system is intended for use with a wide variety of model types (boats, cars, variable pitch helicopters, etc) and accordingly its user interface allows the controls on the transmitter unit to be configured for each of these. Here we deal only with those aspects that are relevant to its use with a multirotor (such as a quadcopter).

The initial menu presented to the user leads to two sub-menus: one named **System Setup**, the other **Functions Setup**. The first allow for storage of the configurations for up to 20 different models (possibly of differing types) and includes settings for the following:

- **Type**. The setting appropriate to a quadcopter is "Helicopter, Fixed pitch".
- **Sticks mode.** This allows a choice of which sticks to associate with which channel. The default is Mode 2, defined as:

Channel 1	Right stick, horiz	(roll)
Channel 2	Right stick, vert	(pitch)
Channel 3	Left stick, vert	(throttle)
Channel 4	Left stick, horiz	(yaw)

- **PPM Output**. This allows a choice of whether to the receiver should employ PPM, a multiplexed channel output (suitable for feeding directly into a flight controller) on a single pin or whether it should itself demultiplex these channels and generate PWM signals for each one on a separate pin (suitable for legacy flight controllers).
- **RX Battery**. This allows for alarms to be set on limits for the supply voltage to the receiver unit. In practice, this may be irrelevant since the receiver unit will usually be fed from a regulated power supply on the flight controller.
- **Fail Safe**. This defines the way that each channel will respond in the event of loss of signal, and allows a choice between retaining the most recent value or setting a specific value to be adopted.
- **Choose Sensors**. This allows the user to specify up to three sensors whose values are to be shown on the main screen. Assuming that no extra sensors have been attached to the receiver unit, it defaults to showing:

Int V	The (internal) supply voltage to the receiver unit
TX V	The battery voltage of the transmitter unit
Error	The percentage error rate of received packets

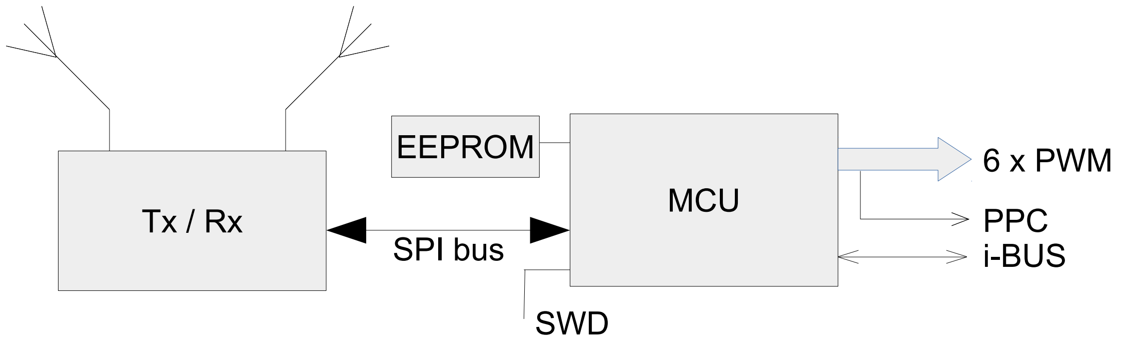
- **i-BUS Setup.** For specifying the association between i-BUS sensors (if any) and channels.
- **Servo frequency**: This defaults to 50 Hz, but may be varied upwards.

The **Function Setup** menu includes an extensive collection of setting for customising the transfer functions between the stick positions and the resultant channel outputs. In practice, many of these functions are more conveniently defined within the flight controller with the advantage that a more sophisticated (computer based) graphical user interface can be employed.
This menu also includes:

- **Auxiliary channels**. This allows channels 5 and 6 to be configured to be driven by any of VrA and VrB (the two rotary controls) or any of the four toggle switches A, B, C, or D.

The Receiver unit

Like the transmitter unit (described above), the FS-iA6B receiver unit consists of a transceiver module and a microcontroller, interconnected using an SPI bus. The FlySky specification states that it requires a power supply in the range 4.0 to 6.5V. It draws a current of about 40mA.



The Microcontroller

Again like the Transmitter unit, the heart of the unit is an ARM Cortex M0+ microcontroller but, this time, of type STM32F031C6T6. It incorporates 16kB [or 32kB ?] of flash memory, 4 kB of static RAM, a 16-bit ADC, a 12-bit ADC and the following interfaces:

- SPI, used to communicate with the transceiver module
- I2C, used to communicate with an EEPROM that provides persistent memory for holding parameters such as the unique identifier of the transmitter unit to which the receiver is bound and the default settings to be adopted by the servo outputs if the signal is lost.
- Asynchronous serial, used for the i-BUS.
- SWD, that allows debug access to the internal state of the microcontroller.

The data sheet for the chip states that it provides *"an advanced control timer for 6 channels PWM output"* (a handy basis for implementing a 6-channel RC system).

The roles played by the microcontroller include:

- Controlling the frequency-hopping transceiver (including handling the binding process)
- Handling the protocol associated with the byte streams to and from the tx/rx module.
- Generating the various (PWM, PPM, i-BUS) servo outputs.
- Generating and returning telemetry data.

The transceiver module

The 2.4GHz transceiver is implemented using a type A7106 module. This is a more recent, backwards compatible, version of the A7105 module used in the transmitter unit. The operation of the module is essentially identical to that described earlier, so is not repeated here.

As with the transmitter unit, there are separate, sleeve-dipole antennae for the receiver and transmitter . The FlySky user manual states that these should be placed at right angles to each other. [**Aside**: It is not clear why this is required: it is not for diversity (since there is only a single receiver). Nor is it to prevent cross-channel rx-tx interference, since only one of these can operate at a time. Possibly it is to prevent one antenna resonating with the other, and so distorting its omni-directional polar diagram.]

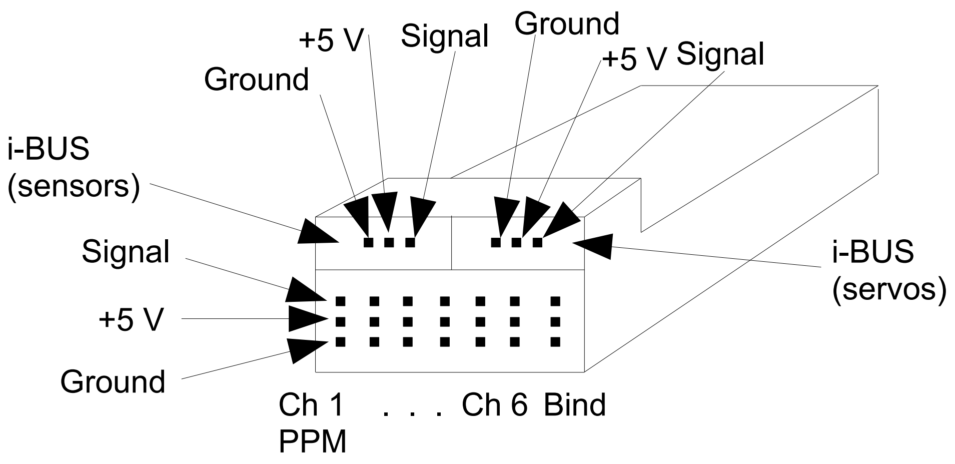
The Servo outputs

The unit provides three different interfaces for the servo outputs:

- A set of 6 PWM (Pulse Width Modulation) channels
- A PPM (Pulse Position Modulation) channel (an alternative to the PWM channels)
- An i-BUS channel (proprietary to FlySky)

The first two are analog in nature, the third is digital. The amplitude of all signals is about 3.36V.

The pin connections are:



The PWM channels

There are 6 PWM outputs, one associated with each channel. The first 4 are associated with the control sticks, the last 2 are optionally associated with the switches or rotary controls of the transmitter unit.

The states of the switches are represented by analog signals: 0% represents OFF, 100% represents ON and (for the 3-position toggle switch), 50% represents the mid position.

The value of each channel is represented in the long-established way, using the width of a positive-going pulse of duration between 1ms to represent 0%, and 2ms duration to represent 100%. The pulses are repeated at a rate set by the "Servo Frequency" setting on the transmitter unit. This can range from 50Hz (so that the pulses occur every 20ms) up to 380Hz (so they occur about every 2.64ms, thus reducing the response time).

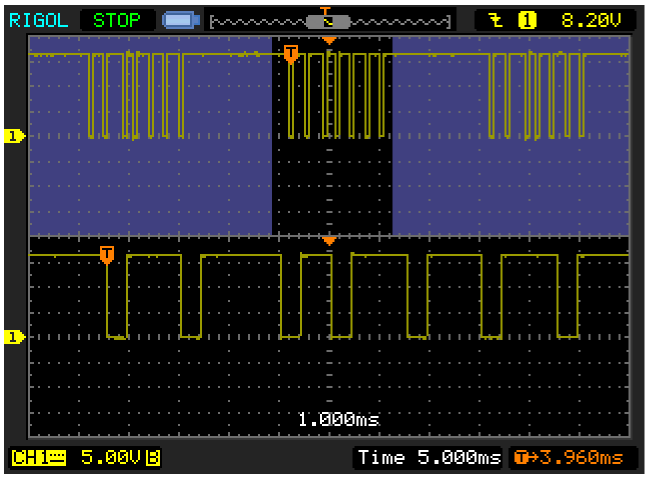
There are no pulses on these lines until a signal is first received from the transmitter unit but, once received, the pulses remain present even if the received signal disappears.

The PPM channel

The receiver unit may (by setting the "PPM" option in the transmitter unit menu) instead be set to multiplex the 6 channels onto a single output (the leftmost pin). This simplifies the connection between the receiver unit and the flight controller. In this case, the values of the six channels are represented by the duration of a sequence of 6 positive-going pulses, each one of duration between 0.6ms and 1.6ms, separated by intervals of 0.4ms. This sequence is repeated every 20ms (this rate is *not* affected by the setting of the "Servo Frequency" parameter).

As an example, this pictures shows the PPC waveform yielded when the 6 channels are set to the following values:

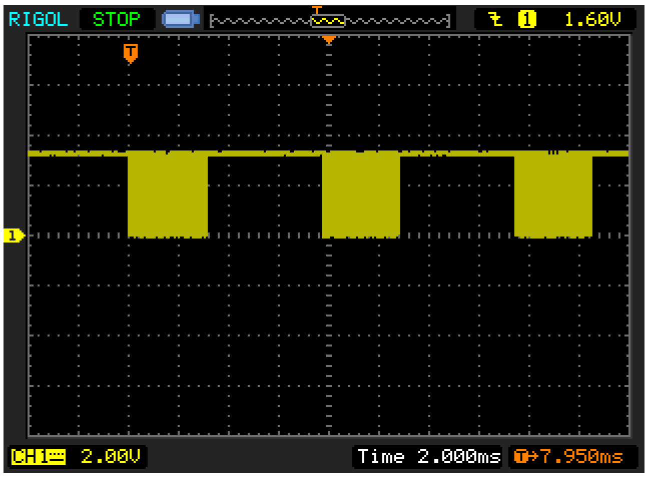
Channel	Value
1	Mid range
2	Maximum
3	Minimum
4	Mid range
5	Mid range
6	Mid range

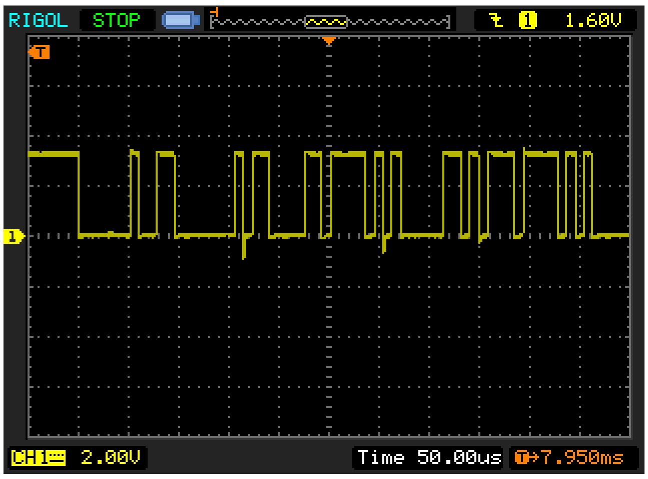


The i-BUS channels

There are two i-BUS ports, one to receive telemetry data from an optional set of sensors (voltage, temperature, etc), the other to send control data to the flight controller. Each port is a single-wire serial interface running at 115,200 baud.

The signal on the "Servo" channel consists of a series of bursts of data, repeated at intervals of about 7.5ms, with each one lasting about 3ms.





Unlike the PWM and PPM modes of operation, this signal conveys the values for some 14 output channels, with each value represented by a 16-bit word (since the transmitter unit only populates some 6 channels, the remaining 8 are presently unused).

The "Sensor" port, by default, carries the data for only a single sensor, the onboard one that measures the supply voltage to the receiver unit. However, extra sensors, such as the FlySky FS-CVT01 voltage measuring sensor, can be chained together and onto the Sensor port

Acknowledgements

These notes are based on material in the FlySky FS-i6 instruction manual, on inspection of the innards and waveforms of an FS-i6 system, on the manufacturers' data sheets for the main components and on snippets of information gleaned from the Internet, especially from the RC Groups forums.

Where no information was easily available, I have made plausible assumptions. Not all of these will be correct, so do treat these notes, and especially the details, with caution! I would be very happy to receive additions/corrections from anyone who is familiar with current radio-control technology or who has detailed knowledge of FlySky products.