



Guides

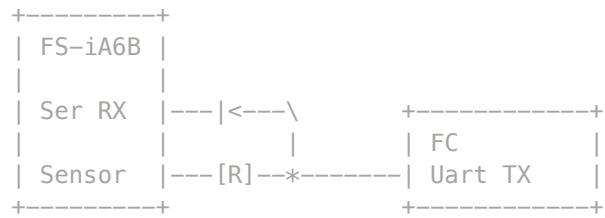
Current

Single Wire FlySky IBUS Telemetry

## Single Wire FlySky IBUS Telemetry

It's now possible to combine the serial rx and ibus telemetry on the same fc uart on a single pin.

I'm combining the ibus ports of the FS-iA6B receiver with a diode and a resistor.



$R = 10\text{Kohm}$ , Diode 1N4148 or similar.

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



Both uart tx and rx channels are used so it's not possible to use the spare pin for rx of something else.

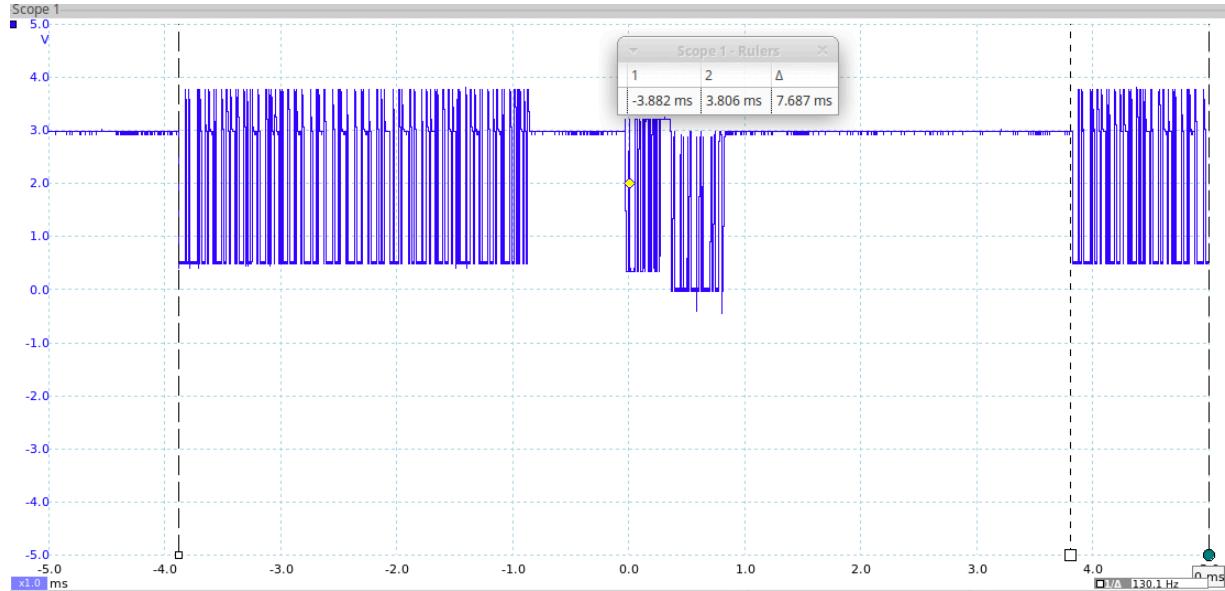
Configure the serial port like this to get both serial rx and ibus telemetry on the same port:

```
serial 1 4160 115200 57600 115200 115200
```

It is still possible to run the serial rx and ibus telemetry on two uarts like before, an example:

```
serial 1 64 115200 57600 0 115200
serial 2 4096 115200 57600 115200 115200
```

This is how the signals look:



## Technical details:

iBus Telemetry is a half-duplex serial protocol. It shares 1 line for both TX and RX. It runs at a fixed baud rate of 115200. Queries are sent every 7ms by the iBus receiver. Multiple sensors can be daisy chained with ibus but this is implemented but not tested because i don't have one of the sensors to test with

/ \ / -----

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

The protocol is driven by the iBus receiver, currently either an IA6B or IA10. All iBus traffic is little endian. It begins with the iBus rx querying for a sensor on the iBus:

/-----

| IBUS RX | > Hello sensor at address 1, are you there? -----/ [ 0x04, 0x81, 0x7A,  
0xFF ]

0x04 - Packet Length 0x81 - bits 7-4 Command (1000 = discover sensor) bits 3-0  
Address (0001 = address 1) 0x7A, 0xFF - Checksum, 0xFFFF - (0x04 + 0x81)

Due to the daisy-chaining, this hello also serves to inform the sensor of its address (position in the chain). There are 16 possible addresses in iBus, however address 0 is reserved for the rx's internally measured voltage leaving 0x1 to 0xF remaining.

Having learned it's address, the sensor simply echos the message back:

/-----

Yes, i'm here, hello! < | Sensor | [ 0x04, 0x81, 0x7A, 0xFF ] -----/

0x04, 0x81, 0x7A, 0xFF - Echo back received packet

On receipt of a response, the iBus rx next requests the sensor's type:

/-----

| IBUS RX | > Sensor at address 1, what type are you? -----/ [ 0x04, 0x91, 0x6A, 0xFF ]

0x04 - Packet Length 0x91 - bits 7-4 Command (1001 = request sensor type) bits 3-0 Address (0001 = address 1) 0x6A, 0xFF - Checksum, 0xFFFF - (0x04 + 0x91)

To which the sensor responds with its details:

/-----

Yes, i'm here, hello! < | Sensor | [ 0x06 0x91 0x00 0x02 0x66 0xFF ] -----/

0x06 - Packet Length 0x91 - bits 7-4 Command (1001 = request sensor type) bits 3-0 Address (0001 = address 1) 0x00 - Measurement type (0 = internal voltage) 0x02 - Unknown, always 0x02 0x66, 0xFF - Checksum, 0xFFFF - (0x06 + 0x91 + 0x00 + 0x02)

The iBus rx continues the discovery process by querying the next address.

Discovery stops at the first address which does not respond.

The iBus rx then begins a continual loop, requesting measurements from each sensor discovered:

/-----

| IBUS RX | > Sensor at address 1, please send your measurement -----/ [ 0x04, 0xA1, 0x5A, 0xFF ]

0x04 - Packet Length 0xA1 - bits 7-4 Command (1010 = request measurement) bits 3-0 Address (0001 = address 1) 0x5A, 0xFF - Checksum, 0xFFFF - (0x04 + 0xA1)

/-----

I'm reading 0 volts < | Sensor | [ 0x06 0xA1 0x00 0x00 0x5E 0xFF ] -----/

0x06 - Packet Length 0xA1 - bits 7-4 Command (1010 = request measurement) bits 3-0 Address (0001 = address 1) 0x00, 0x00 - The measurement 0x58, 0xFF - Checksum, 0xFFFF - (0x06 + 0xA1 + 0x00 + 0x00)

Due to the limited telemetry data types possible with ibus, we simply send everything which can be represented. Currently this is voltage and temperature and throttle value.

### Youtube

- ▶ Joshua Bardwell
- ▶ Ivan Efimov

### Links

- 🌐 Oscar Liang
- 🌐 VitroidFPV

### Community

- discord
- intoFPV

### Feeds

- RSS
- Atom
- JSON



Copyright © 2025 All rights reserved Team Betaflight

Built with Docusaurus

made with ❤️ by **VitroidFPV** and **un!t**

