ELEV8 Flight Controller (ELEV8-FC)

Communications Bus Concept #2

Concept Overview

Essentially a multidrop configuration. All devices plug into FC via standard 3-pin servo cable.

FC reads pulses from receiver and encodes into fixed format 8 byte flight protocol:

byte 1: throttle 0-255
byte 2: yaw 0-255
byte 3: roll
byte 4: pitch
byte 5: receiver ch. 5 (could be switch, VR, whatever)
byte 6: receiver ch. 6
byte 7: receiver ch. 7
byte 8: receiver ch. 8

FC transmits these 8 bytes on the serial bus, and waits for 20mS. After 20mS, FC processes flight data (does it's flight thing)

ALL modules receive the 8 bytes at once. Some will want to reply, others not. Does not matter. Each sensor (and we limit to 8), can reply at Twait*module_number and transmit back the core 8 bytes, plus optionally up to 16 bytes (actually, we need to make the last byte a chksum, and perhaps a single bit flag to set if the data contains FC-data or not)

So module1 waits until the IO line is low for Twait1 microseconds, then can transmit. Module 2 does the same for Twait2, and so on. If a module doesn't need to reply, obviously it'll not use its time slot, so the next module will reply more quickly. That's OK. FC does not care- it just waits up to 20mS.

IF a module writes back FC-data, then FC just updates it's buffer with that new data. Multiple modules might write back data, so the order the modules are plugged in could be important, but that's up to the user.

All modules can hear each other at the same time, so they can share data. FC will disregard the extra 16 byte optional data for now... Later it might transmit to base via xbee, but for now, that is only for the other modules to perhaps listen to.. Example, a logging module or the beginner module might be interested in the GPS data, before it makes a final edit to the FC-data.

With the bus running at 115200, we can complete the loop (assuming all 8 modules reply with 16 bytes+8 core bytes), in 18mS.

Super simple or what !