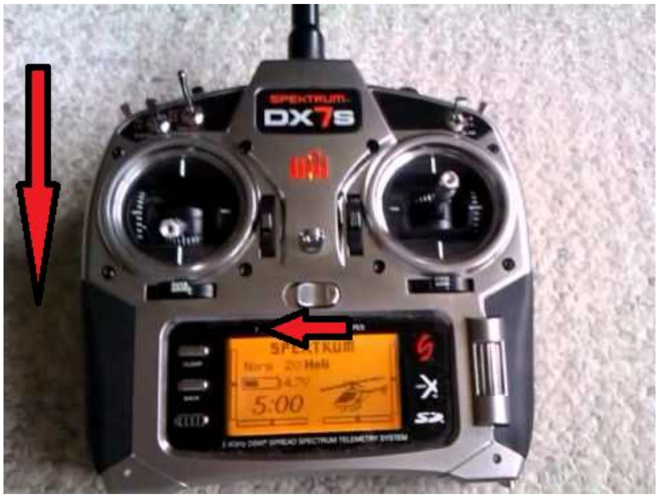
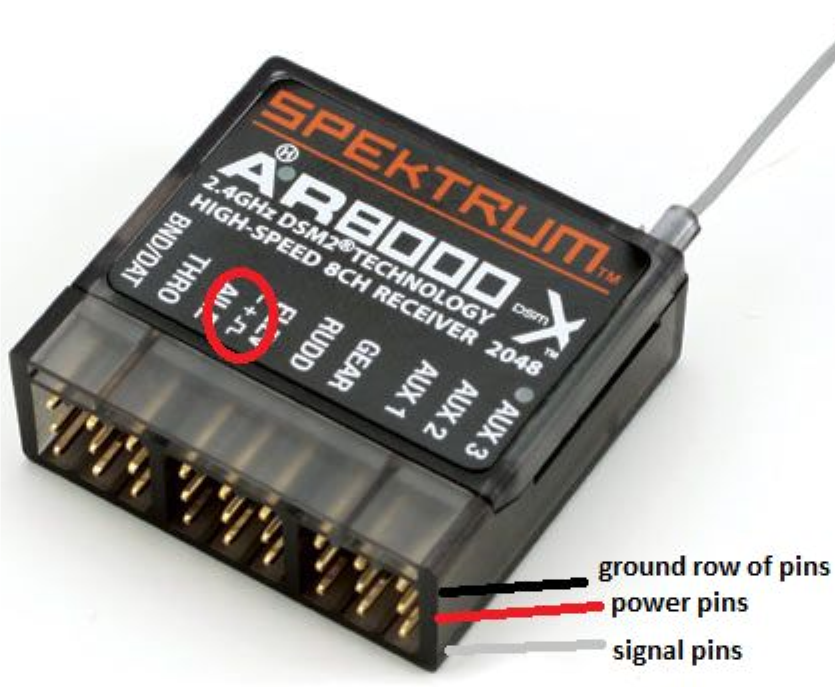
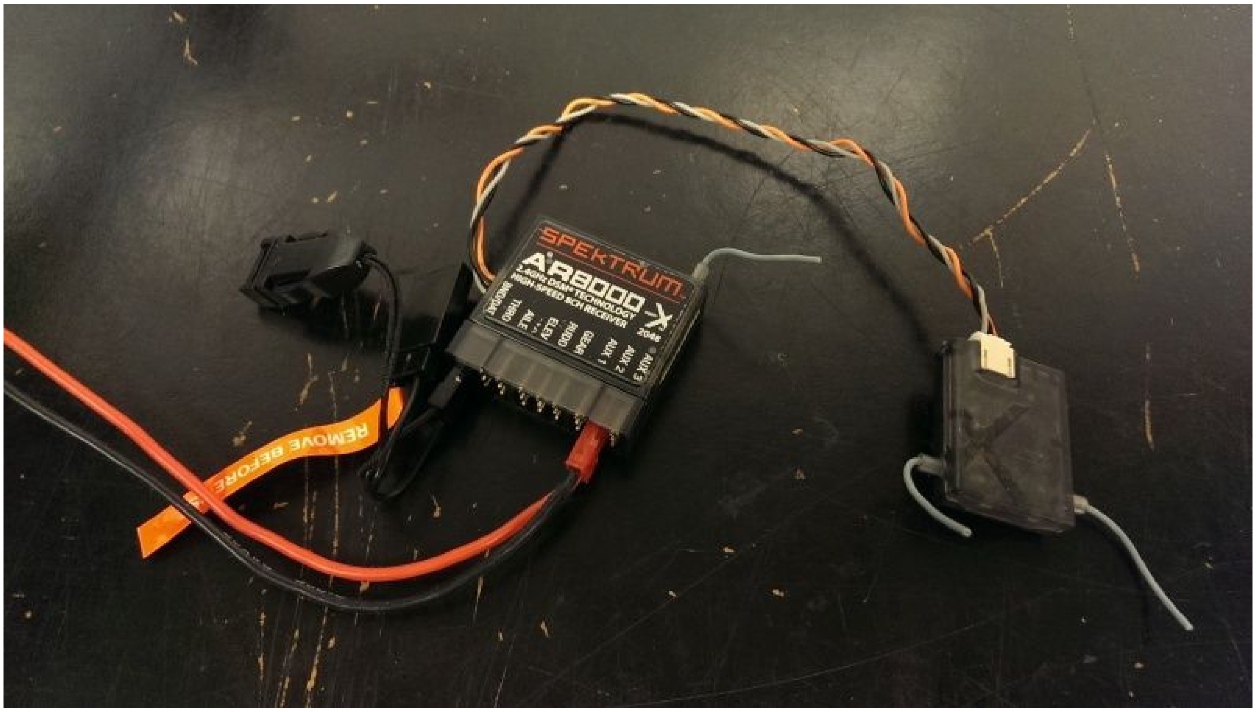
Binding the AR8000 Receiver:

1>Make sure the throttle stick is in the down position
2>turn off controller



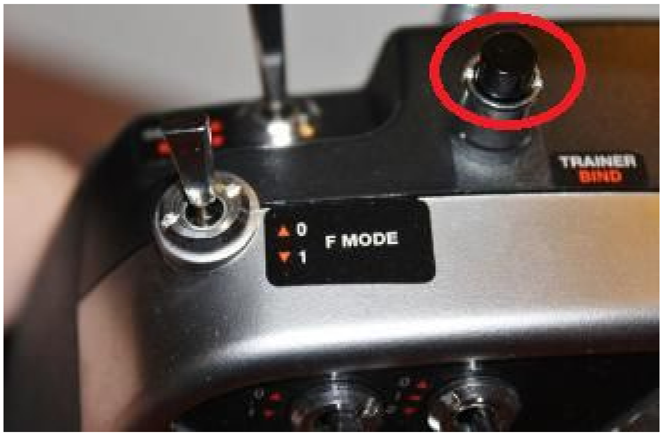
3>Insert the bind plug into the bind pins on the main receiver
4>Power the main receiver using a 5v power source, I just used the 4.8v battery pack from another dx7s receiver(The operating voltage for the main receiver is 3.5v-9.6v)





5>The unbound receivers should have a blinking orange light now

6> Hold the trainer/bind button down and power on the controller. Release the trainer bind button when the orange light stops blinking and you get a solid orange light.

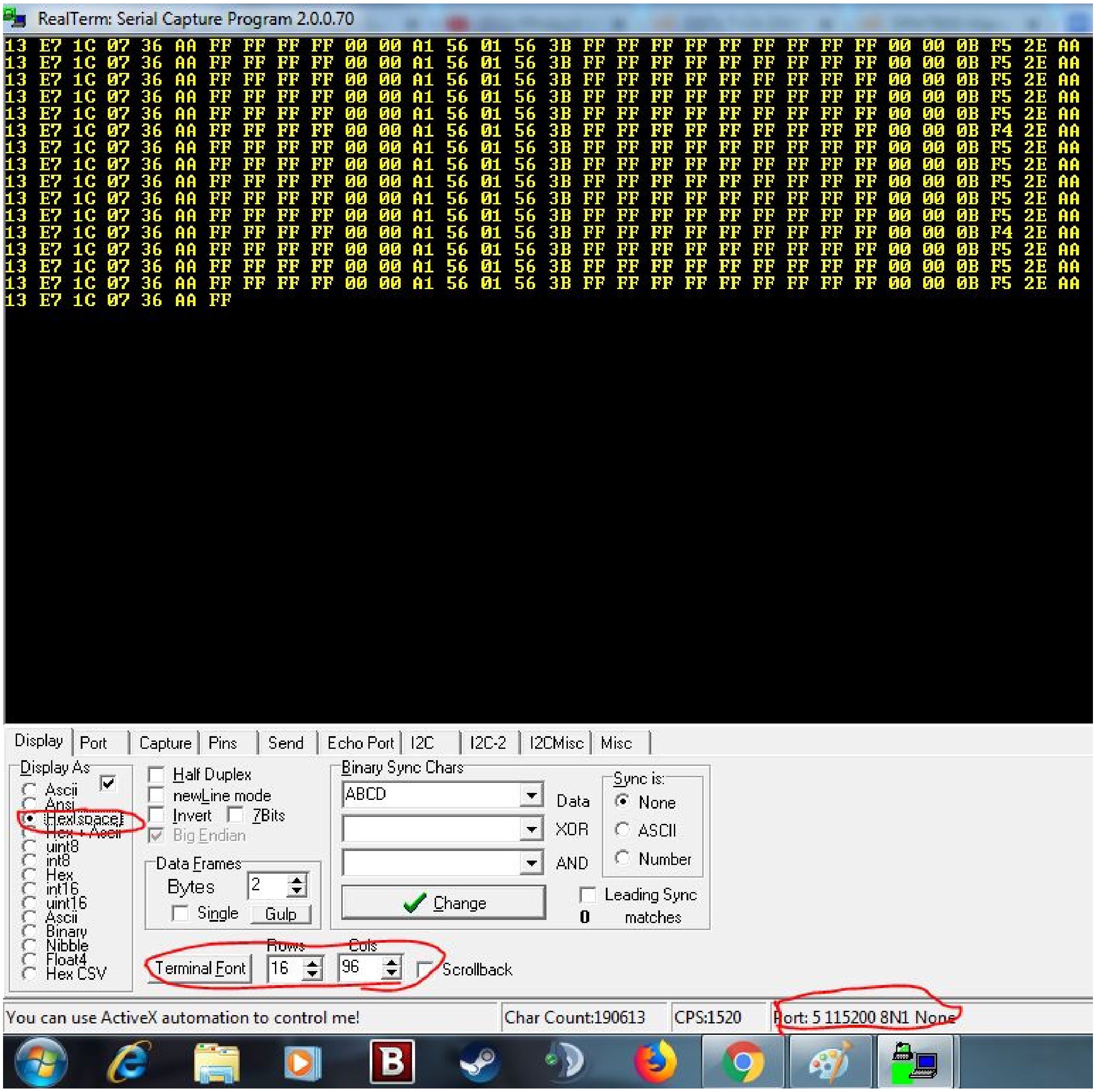


The receivers are now binded to that controller. You will have to repeat this process if you want to bind the receiver to a different controller or vise versa.

7>You can now connect the secondary receiver directly to a UART port on a microcontroller or connect to the FTDI board or UART to usb adapter.Orange wire to 3.3v, Black wire to Ground, Grey wire to RX on a UART port.

Use a serial monitor like RealTerm that will alway to read the data stream from the USB port and view it as Hex(space) format.

8>Data is sent in the format: Baud rate 115200, Data bits 8, 1 Stop bit, No Parity, No Flow Control



RealTerm: Serial Capture Program 2.0.0.70

```
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F4 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F4 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B F5 2E AA
13 E7 1C 07 36 AA FF
```

Display As: ○ Ascii ○ Ansi ● Hex[space] ○ Hex + Ascii ○ uint8 ○ int8 ○ Hex ○ int16 ○ uint16 ○ Ascii ○ Binary ○ Nibble ○ Float4 ○ Hex CSV ☑

□ Half Duplex
□ newLine mode
□ Invert □ 7Bits
☑ Big Endian

Data Frames: Bytes 2 □ Single Gulp

Binary Sync Chars: Data ABCD; XOR; AND

✓ Change

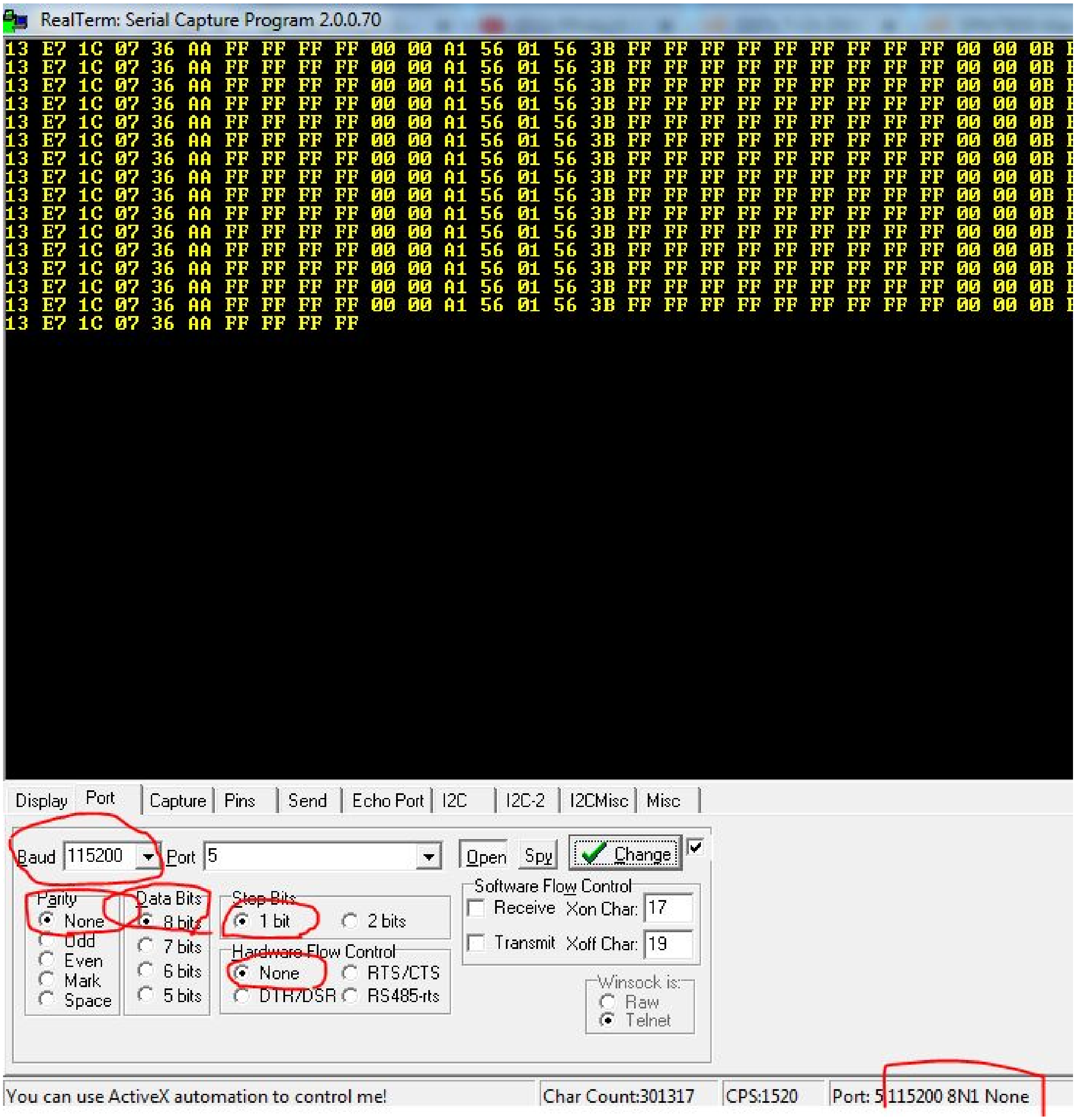
Sync is: ● None ○ ASCII ○ Number

□ Leading Sync

0 matches

Terminal Font Rows 16 Cols 96 □ Scrollback

You can use ActiveX automation to control me! | Char Count:190613 | CPS:1520 | Port: 5 115200 8N1 None



RealTerm: Serial Capture Program 2.0.0.70

```
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF 00 00 A1 56 01 56 3B FF FF FF FF FF FF FF FF FF 00 00 0B
13 E7 1C 07 36 AA FF FF FF FF
```

Baud 115200 Port 5

Parity: ● None ○ Odd ○ Even ○ Mark ○ Space

Data Bits: ● 8 bits ○ 7 bits ○ 6 bits ○ 5 bits

Stop Bits: ● 1 bit ○ 2 bits

Hardware Flow Control: ● None ○ RTS/CTS ○ DTR/DSR ○ RS485-rts

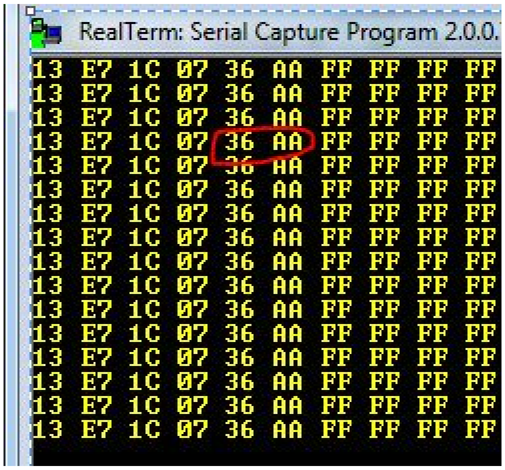
Software Flow Control: □ Receive Xon Char: 17 □ Transmit Xoff Char: 19

Winsock is: ○ Raw ● Telnet

You can use ActiveX automation to control me! | Char Count:301317 | CPS:1520 | Port: 5 115200 8N1 None

Decimal value of each 2 byte channel:

Example :



```
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
```

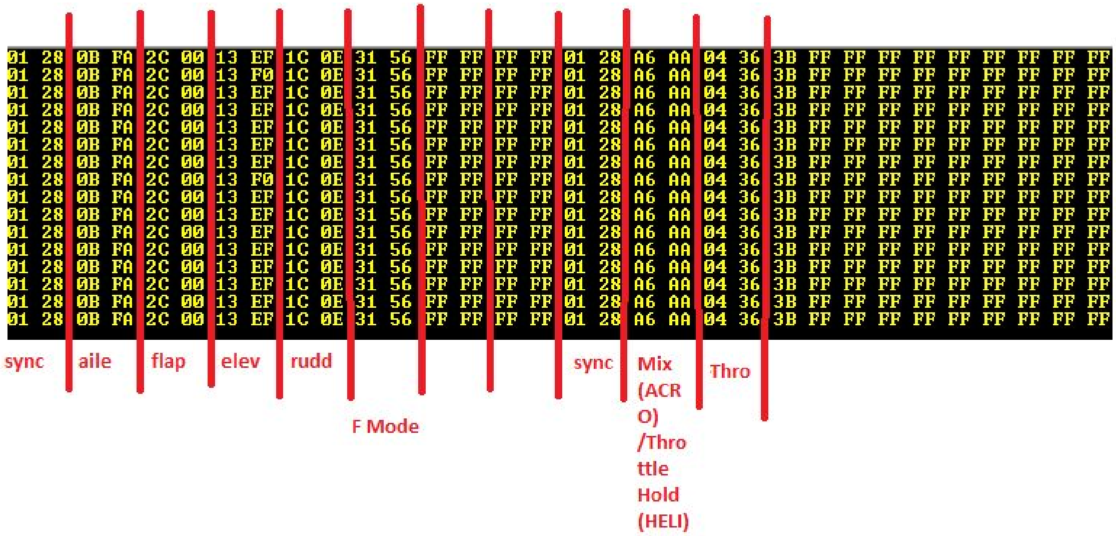
The 3rd channel of data: 0x36AA = 13994

Aile = Left = 3753 Right = 2391
Flap = pos0: 11964
pos1: 11264
pos2: 10582
Elev: Center: 5100 Down = 4439 Upper = 5801
Rudd: Left =7849 Right = 6487
Flight Mode: pos0 = **13994**
pos1 = **12630**
Mix/Hold: pos0 = **42666**
pos1 = **41302**
Thro: Down: 342-1706

Note: The 4 sliders adjust the the range of values by +/- 25%. These values are taken from a controller that was calibrated to the center of each slider. Make sure every slider on your controller is centered. You'll hear a beep when it is.

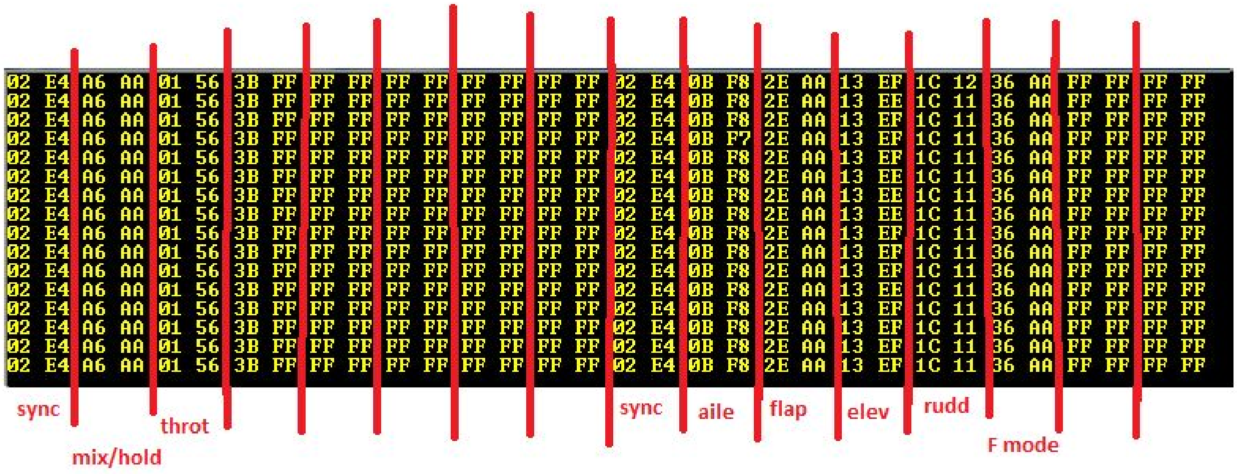
32 Byte Packets are sent each containing 16 channels(2 bytes per channel)

Flight Mode 1*



```
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 F0 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 F0 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
01 28 0B FA 2C 00 13 EF 1C 0E 31 56 FF FF FF FF 01 28 A6 AA 04 36 3B FF FF FF FF FF FF FF FF FF
```

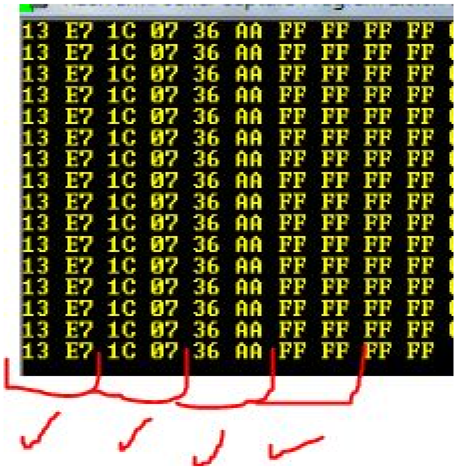
********Flight Mode 0*******



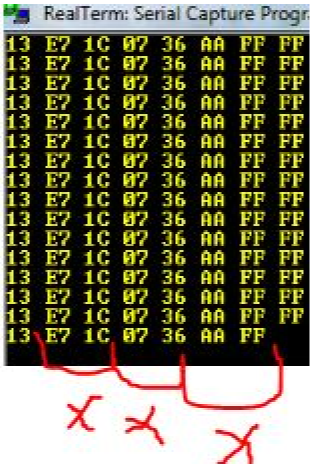
```
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 12 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EE 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F7 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EE 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EE 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EE 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EE 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
02 E4 A6 AA 01 56 3B FF FF FF FF FF FF FF FF FF 02 E4 0B F8 2E AA 13 EF 1C 11 36 AA FF FF FF FF
```

Data is send every 11ms or 22ms depending on the frame rate you set on the controller(hold the clear and back button then power on the controller to access the System Setup menu the go to Frame Rate and choose when frame rate and mode you want to use. I'd recommend staying on 22ms and DSMX mode.

You want to write code to read the serial data in a way that it is aligned properly so you can use the correct data. So you have to know when a new packet starts and stops



```
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
13 E7 1C 07 36 AA FF FF FF FF
```



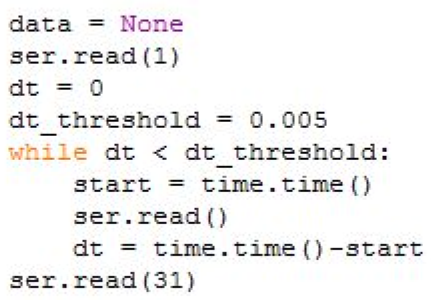
```
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF FF
13 E7 1C 07 36 AA FF
```

Note that the sync bytes that tell you when a new packet starts repeats in the same packet twice and changed every time you turn the controller on and off. Using the sync bytes to determine the start of a new packet isn't consistent and won't work in my case. You have 2 options:

Detect when you see empty channels in my case : (3B FF FF FF FF FF FF FF FF FF)
I know that following this sequence is the sync bytes of a new packet and I can use that to read in the correct data.

Read every 22ms in my case:

Some simple code can align the packets in the data stream:



```
data = None
ser.read(1)
dt = 0
dt_threshold = 0.005
while dt < dt_threshold:
    start = time.time()
    ser.read()
    dt = time.time()-start
ser.read(31)
```

(Python 2.7)

Full code:

https://github.com/YawFrempong/AR8000-Serial-Data-Read

Helpful links:
https://github.com/samfok/remote_receiver_tutorial

http://darknrgy.typepad.com/darknrgys-blog/2015/02/ar8000-satellite-as-standalone-receiver.html

http://www.dogfight.no/2011/01/spectrum-receiver-satellite-to-arduino.html