Configure the AutoPilot

This procedure will cover AutoPilot configuration of the Hitec UAVCAN Servo. This document is a continuation of the previous document and comes with the following assumptions:

* AutoPilot will be NODE 10 on the UAVCAN Bus
* Servo will be Servo 2, which is going to be configured as the Elevator servo
* The AutoPilot software in this document is ArduPlane 4.0.5
* The AutoPilot hardware in this instance is a CUAV v5 Nano and we will be connected to CAN1

All of the hardware AutoPilot devices currently supported in ArduPilot do not provide suitable power to the CAN ports. Because of this, it is strongly recommended that integrators supply appropriate current to the UAVCAN servos outside of that bus. Hitec has a collection of breakout boards to meet this demand and simplify wiring. Do not forget to connect a bus termination resistor if you are not connecting any other devices.

After connecting to ArduPilot, the following parameter changes are required:

CAN\_D1\_PROTOCOL 1

CAN\_P1\_DRIVER 1

Once CAN\_P1\_DRIVER is changed from 0 to 1, you will need to reboot the autopilot gain access to the rest of the CAN Parameters:

CAN\_D1\_PROTOCOL 1

CAN\_D1\_UC\_NODE 10

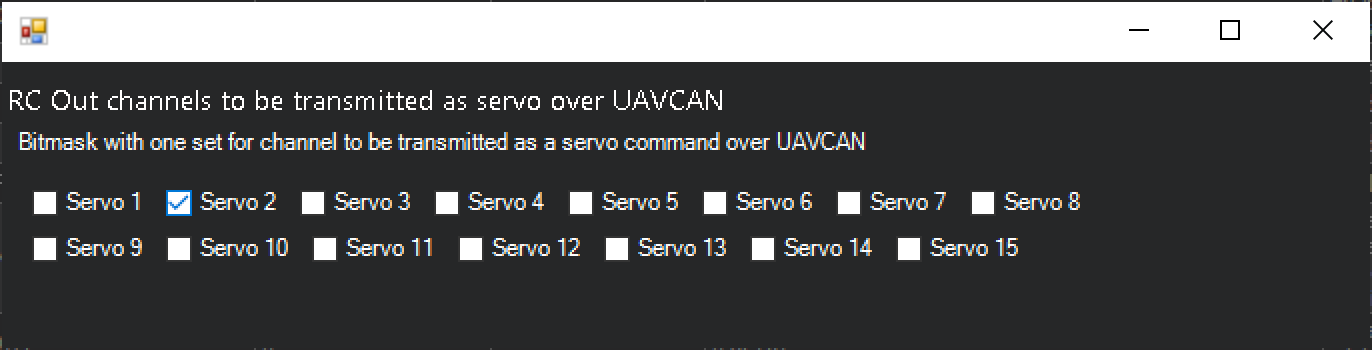
CAN\_D1\_UC\_SRV\_BM 2

CAN\_D1\_UC\_SRV\_RT 50

CAN\_P1\_BITRATE 1000000

CAN\_P1\_DRIVER 1

For **CAN\_D1\_UC\_SRV\_BM** you will need to know how to compute the bitmask for the servo you are using; This is an ArduPilot bitmask, not one for the servo itself. It is telling ArduPilot to copy any PWM-OUT for the corresponding servos to the UAVCAN bus in the appropriate format. Fortunately, Mission Planner makes this very easy with a pop-up once you click that value. Since we’re wanting to make SERVO2 be UAVCAN, we’ll select that servo in this pop-up, close it, and press [Write Params]



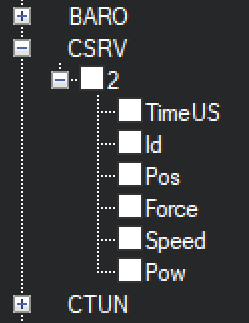
Next we’ll verify SERVO2\_FUNCTION is configured to be our Elevator servo by setting it to 19 if it isn’t already, and pressing [Write Params]

Optionally, we may want to run the servo at a rate higher than 50Hz. Depending on which servo you have, you can configure CAN\_D1\_UC\_SRV\_RT to be a number greater than 50Hz. Should you configure for greater than 50Hz, make sure SCHED\_LOOP\_RATE parameter is also at least at that rate. (For Plane, the default is 50Hz)

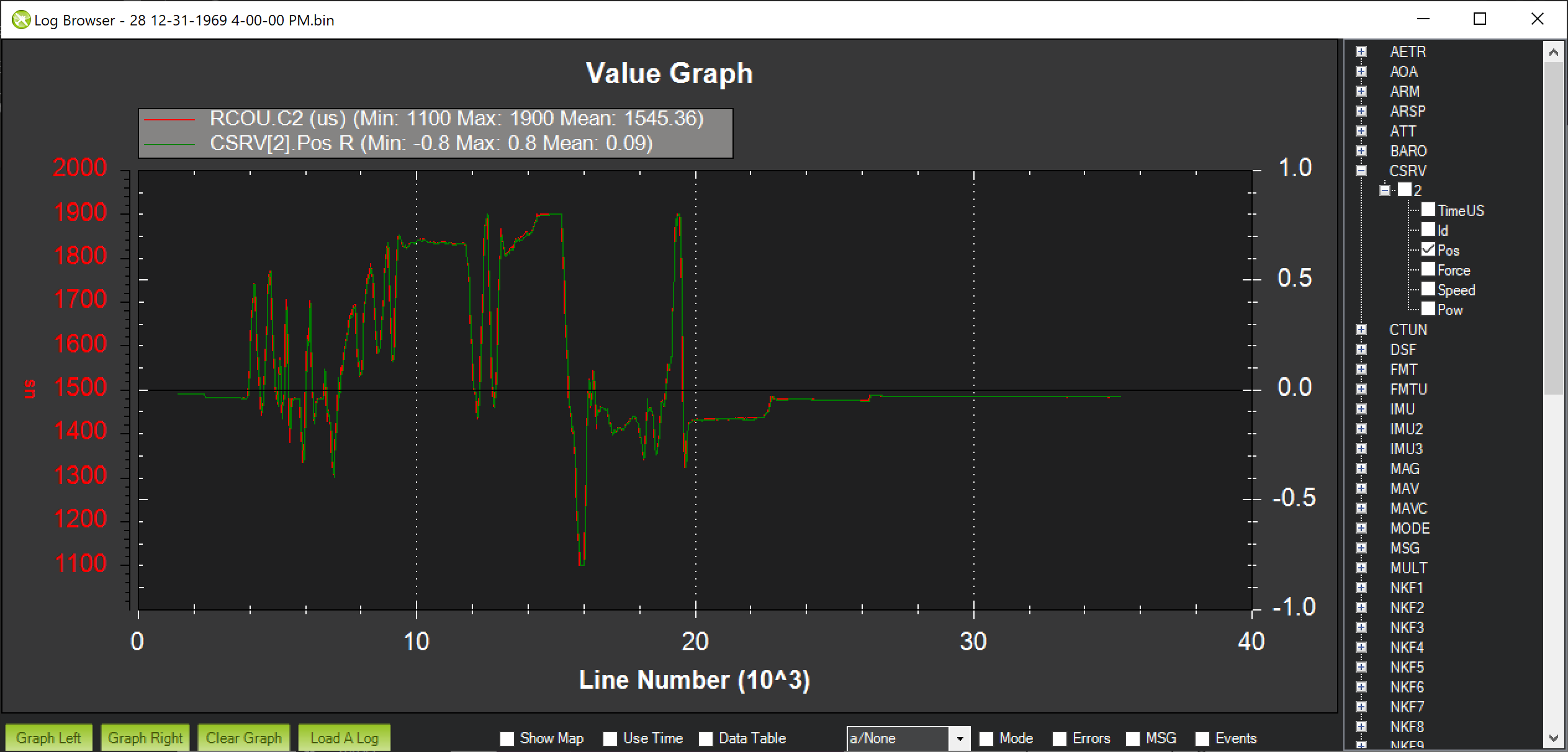
Depending on the rest of your configuration, you may be able to configure the plane for FBWA or some other attitude-stabilizing mode, move the autopilot around and watch the servo actuate.

Log Analysis

Since we configured the servo to stream data at 50Hz, ArduPilot will be able to see this data and will log it at the stream rate you’ve configured. To view this data, you can open your favorite dataflash viewer, open the log, and browse to CSRV:



Here is an example of RCOU.C2 (SERVO2’s PWM Value) mapped against CSRV[2].Pos



Here is another sample where I’ve mapped CSRV[2].Pos against CSRV[2].Force to show how I was randomly providing resistance to the servo while moving it using FBWA:

