SINGLE AXIS BRUSHLESS MOTOR CONTROLLER
Dynamic Docking Module Interface Specification:

(rev. F 02/24/16)

For rev E and later firmware

1.0 At Bootup:

At bootupt the board reads its address. If the switch is set to anything but #1 or #2 the program will stop and go to rest mode. You will get this message:

BOARD ADDRESS is 4. MUST BE #1 OR #2
SELECT ADDRESS AND REBOOT SYSTEM
ENTERING RESET MODE

You must select a valid address #1 for latch and #2 for whiskers and power cycle the board.

2.0 Command Motors:

Baud rate of 115 200. No time out between commands.

#Mm,C,pp,cc (CR)

m = Board address. 1: Latch board. 2: Whiskers board. 0: all boards

C = R= Retract Whiskers or Open Latch
X= eXtend Whiskers or Close Latch
S= Stop.

pp = 0x00 < pp < 0xFF. PWM limit. Adjust the motor speed [clip the PWM signal]
Recommended value: 0xFF for the whiskers and 0x50 for the latch

cc = 0x05 < cc < 0x32. Current limit. Adjust the maximum torque. If the torque on the shaft is to high the motor controller will go in current control mode and set the fault bit (see below for a description of the fault bit).
0x32 ~ 500ma
0x10 ~ 150ma
0x20 ~ 250ma
Recommended value: 0x60

- ✓ **Request for firmware version : #Vx**
  Reply : !Vx,vvvvvv-yyy with x=Board number, vvvvv the version and yyy "latch" or "whiskers". The firmware for whisker and latch is the same. The board will reply with yyy=latch or whiskers depending on its address (x=1->"latch", x=2->"whiskers").
  x is the address of the board (from 0 to 7). Address 0: all boards reply.

- ✓ **Status Request : #Fx**
  Where x is the address of the board (from 0 to 7). Address 0: all boards reply. x is optional, All boards will reply if #F(CR) is sent.

- ✓ **Bootloader : #Bx**
  Where x is the address of the board (from 0 to 7). See chapter 4.0 “Bootloader”

- ✓ **Request for spare port value : #Sx,y**
  Where x is the address of the board and y is an HEX character to parameter the digital ports. The other boards which receive this command will ignore it and will consider the last command received (#M,Sttt,Sf1f1f1,….) and will reply to it. See chapter 5.0 “Spare port”. Address 0: all boards reply.

- ✓ **Reset board : #R**
  Reset the board.

2.0 Reply:

Reply format: **!Fx,s2s1,cc,m,ff,syyy,a**

Where:

x = Number of the board. Board #2 replies after 200ms to avoid collision with board#1 messages when both boards replies at the same time.

s2s1 = Switch2 and switch1 status. 1 means the switch is ON.

Board 1 (Latch)
01: Latch in closed position.
10: Latch in open position.
00: Latch in the middle.
11: impossible (error, check wiring)

Board 2 (Whiskers)
01: Whiskers opened.
10: Whiskers closed.
00: Whiskers in the middle.
11: impossible (error, check wiring,…)

cc = Hex number between 0x05 and 0x32 representing of the current limit desired. The software clips the value of the minimum (0x05) and the maximum current allowed (0x32), therefore the feedback might be different than the desired current command.

m =
Mode: R (Retracting Whiskers/Closing Latch) or X (eXtracting Whiskers/Opening Latch). Motor servo the position and disable itself when position is reached. If manually moved the motor re-enable itself and adjust position.

Mode S (motor stopped): Motor disabled and will not servo its position.

ff = fault byte code (hex): same thing as the fin motor controller. See chapter 3.0

s = sign shaft position

yyy = shaft position from -511 to +511
a = The 2 I/O ports status byte. See chapter 5.Spare Ports for the definition of 'a'.

3.0 Faults :

- If there is an error in the serial message received: reply = !**n?**

- If there is an internal error (water leak, over current…) a fault bit is set.
- No fault if this fault byte is null.

➢ Fault Byte **CC** : b7 b6 b5 b4 b3 b2 b1 b0

Bit0 = Over-current (peak)
Bit1 = Pressure fault on thruster board
Bit2 = Motor over-heat
Bit3 = Motor controller over-heat
Bit4 = Motor Controller Fault: under voltage, short circuit or invalid Hall sensor codes
Bit5 = Water leak
Bit6 = Over-current (average) for fin motor
Torque control ON for the thruster. To avoid an over-current fault a current-limit ramp has been generated. Based on the RPM a current limit will be defined and the PWM signal monitored to limit the torque.
Bit7 = General fault (thruster only). Means a fin motor has a problem.

- **If the fault byte is not null :**

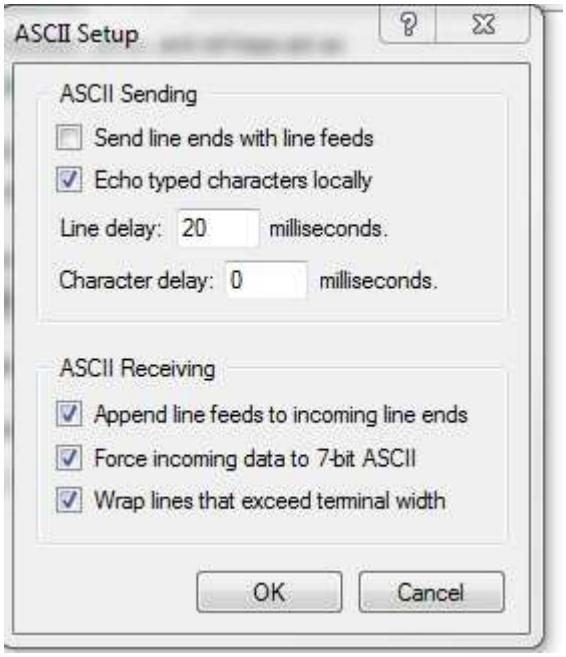
The motors are disabled for 1 second. Except if it's an over-current (average) fault. In that case the board is in current control mode and reduces the torque.
All faults are cleared when the motor is successfully re-enabled with the CR of the command message.

If the fault is still present one of the fault bit will be set again, the motor will be disabled again for 1 second and a error message will be sent.

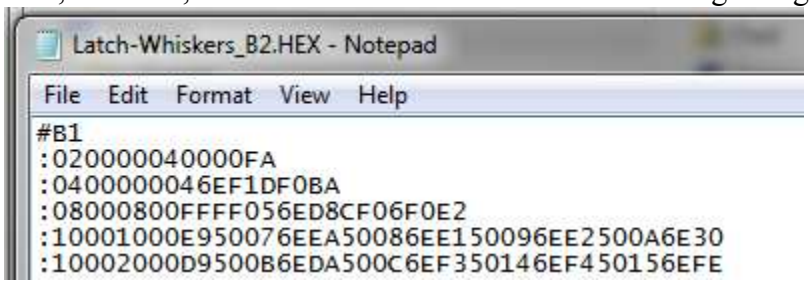
As soon as the general fault bit is set the thruster is turned off until the reset of this bit by a serial message and after 1 second delay.

4.0 Bootloader :

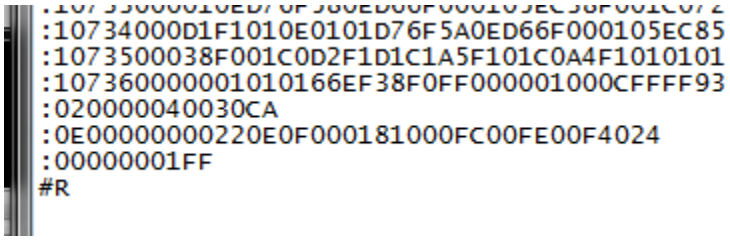
Setup the right baudrate (115200 bps). No flux control. Parity none. Stop bit 1. Make sure you leave 20ms after each line and do not send line ends with line feed:



Edit, or check, the bootloader file and make sure the beginning of the file starts like that:



And ends like that:



#B1 is the bootloading command telling the board#1 that a program is being bootload and to all the other board to ignore this command. So edit this line to #B2 if you need to reprogram board#2.

5.0 Spare ports :

Where x is the address of the board and y is an HEX character (b0 to b7) to parameter the digital ports.

The bits 2 and 3 assigned respectively the digital port1 and digital port2 as an INPUT (1)or an OUTPUT (0).

The bits 0 and 1 are the value desired for the digital ports outputs (respectively 1 and 2).

'y' can have the value :

0000 : 0	Digital ports 1 and 2 as outputs. D1=0 D2=0
0001 : 1	Digital ports 1 and 2 as outputs. D1=1 D2=0
0010 : 2	Digital ports 1 and 2 as outputs. D1=0 D2=1
0011 : 3	Digital ports 1 and 2 as outputs. D1=1 D2=1
010x : 4 or 5	D1 as an input and D2 as an output. D2=0
011x : 6 or 7	D1 as an input and D2 as an output. D2=1
10x0 : 8 or A	D1 as an output and D2 as an input. D1=0
10x1 : 9 or B	D1 as an output and D2 as an input. D1=1
11xx : C,D,E or F	Digital ports 1 and 2 as inputs.

Reply:

The reply will have this format: **!Sx,a,b**

Where:

'x' is the address of the board ($0 < x < 8$)

'b' is the value in hex of the analog spare port ($000 <$ value $<$ FFF)

'a' is a value in hex. The 2 lower bits (b0 and b1) tell us what are the logic level on, respectively, D1 and D2. And the 2 higher bits (b2 and b3) tell us if D1 and D2 are configured as an input or an output.

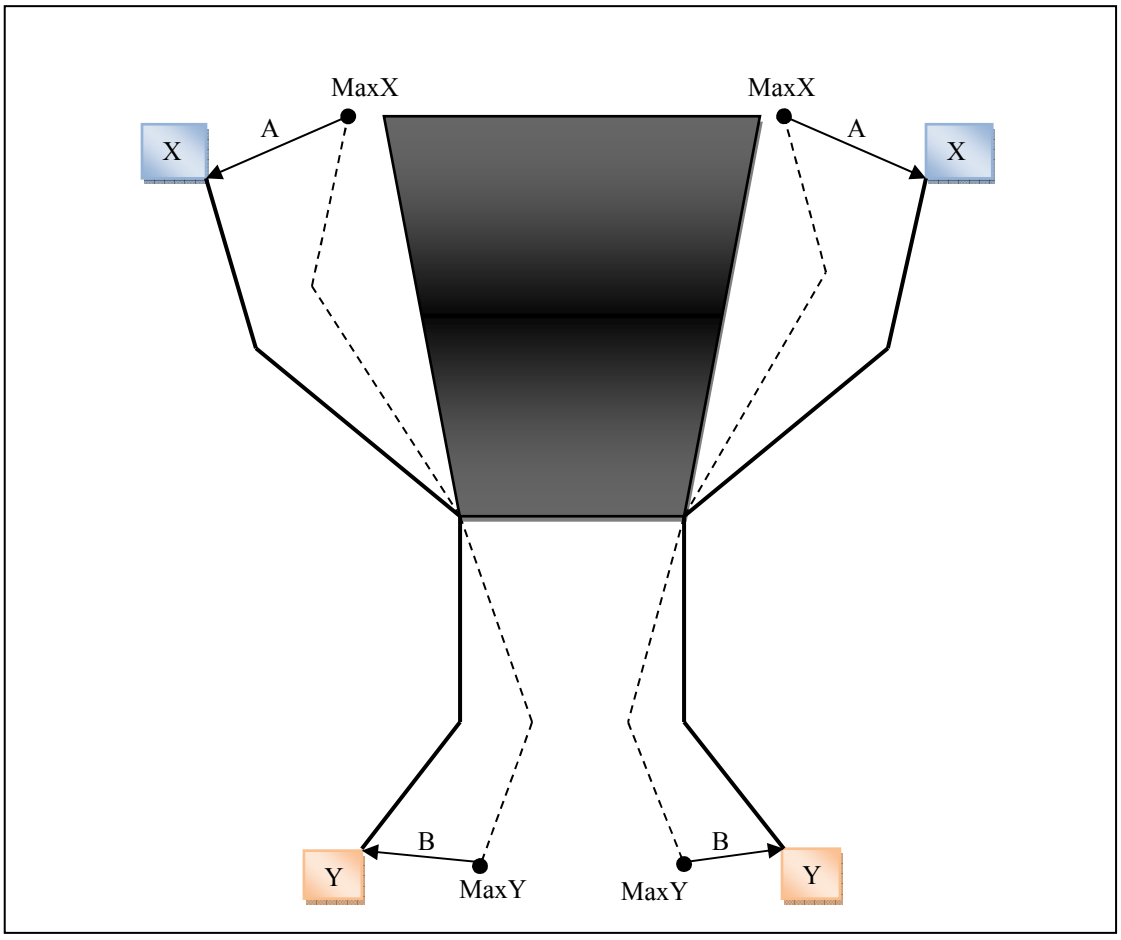
Example :

'a' = 0010 : 0 D1 = output and this output is low. D2 = output and this output is high.
'a' = 1011 : B D1 = output and this output is high. D2 = input and this input is high.

6.0 Calibration :

The calibration consists in finding the 2 end points where the arms can be fully open (X) and fully close (Y). From these 2 end points we substract to predefined numbers (A) and (B) in order to center the window in which we want the sytem to operate.

Here is a figure representing the maximum position (dashed lines) and desired position (plain lines) for the whiskers. The calibration of **both** latch and whiskers happens at the same time.



The board has 4 numbers stored in its eeprom: the low point (X), high point (Y), and the 2 correction numbers (A) and (B). These numbers are remembered after power cycle.

These numbers can be set manually or through an automatic sequence. Hard coded default values are loaded if the board address changes, in other words, if it is used for the whiskers and was previously used as the latch (and vice versa). This avoids the latch, for example, to damage the gear by opening and trying to reach the whiskers end points.

Typically the value of (X) and (Y) will drift overtime depending on how much play builds up on the potentiometer shaft and on the gears. (A) and (B) are constant and shouldn't need to be modified, although that option is available (see below).

If the address of the board is modified via the rotary switch the board will load the default values of the latch or whiskers (depending of that new address). A first-time calibration should then be performed.

Five commands are available (x is the board number):

- **#Ax,V(CR)** Display the calibration numbers currently loaded.

- **#Ax,D(CR)** Loads the default values.

- **#Ax,S,sXXX,sYYY(CR)** Set the 2 endpoints (X) and (Y)

The common reply to these 4 last commands is:**!Ax,sXXX,sYYY,AAA,BBB(CR)**
With:
s = + or –
XXX = 3 decimal digits (X) closed position potentiometer value
YYY = 3 decimal digits (Y) opened position potentiometer value
AAA = 3 decimal digits (A) offset value substracted from closed position value
BBB = 3 decimal digits (B) offset value substracted from opened position value

Auto-Calibration

- **#Ax,C(CR)**

This command starts the calibration sequence. This sequence consists in driving the shaft at low speed to the lowest and highest position (MaxX and MaxY), reports these positions and burns them in the eeprom.

Reply:
!Ax,MaxLow,'MaxX'(CR)
!Ax,MaxHigh,'MaxY'(CR)
!Ax,sXXX,sYYY,AAA,BBB(CR)

Note:
This command, not recommended and to be used for testing purpose only, is also available to setup the A and B constants: #Ax,S,sXXX,,sYYY,AAA,BBB(CR). Reply: !Ax,sXXX,sYYY,AAA,BBB(CR)