Supplemental Instructions for SAFFIR Arm

# Switching between position sensors

The default position sensor value reported by the hdt\_adroit\_driver is the “mechanical position”, an estimate based on the initial reading from the absolute position sensor, and the aggregation of subsequent changes in motor position (multiplied by the total gear reduction to convert to the output angle). This estimate is also used to close the position control loop.

In order to measure the absolute position sensor during runtime, a separate telemetry message must be enabled on each drive (using the Adroit driver console), using the following sequence of commands (where # represents the drive’s CANbus identifier):

1. Change the drive state to INIT (necessary for committing parameter changes)
   * cs # 0
2. Enable MS Telemetry
   * w # 20 7
3. *Select absolute position sensor telemetry index (see MS telem index)*
   * w # 21 0
4. Commit changes
   * c #

Now that MS telemetry has been enabled, subscribe to the /hdt\_adroit\_coms/joint#/hdt\_ms\_telem topic to read the value. Note that MS telemetry is reported as a signed 16-bit integer, and requires the use of a scaling factor to convert to engineering units. See Telemetry scaling factors for the full scale conversion values.

# Re-biasing load cell

The strain gauge load cell is read from the ADC peripheral as a 12-bit unsigned integer, and converted into engineering units (N or Nm) using a linear calibration. While the slope should remain consistent, the offset may need to be updated periodically to account for small mechanical changes to the load cell.

In order to adjust the load cell’s bias, first measure the current value (reported in the effort field of the /hdt\_adroit\_coms/joint\_telem message) with no load on the actuator. This value is should be subtracted from the existing “Effort Sensor Offset” to re-bias the load cell, and can be done with the following sequence of commands:

1. Change the drive state to INIT
   * cs # 0
2. Read existing offset
   * r # 70
3. Subtract the value observed earlier from output of the previous command, and write
   * w # 70 (new offset)
4. Commit changes
   * c #

Significant zero bias may require additional software or hardware troubleshooting; please contact HDT if the above instructions are insufficient.

# Changing control modes

The modes available on each drive are shown in the appendix section Control modes. In order to change modes, first read the current mode, and then use the bitmask parameter (see Parameter list for definition) to write the new value back.

1. Change the drive state to INIT
   * cs # 0
2. Read existing offset
   * r # 4
3. Interpret the bitmask to derive current control mode. Convert back to integer and write
   * w # 4 (new param)
4. Commit changes
   * c #

# Adroit driver console

In order to use the Adroit console, a special console\_enabled argument must be passed to the hdt\_adroit\_coms.launch file on launch.

* roslaunch hdt\_adroit\_driver hdt\_adroit\_coms.launch console\_enabled:=true

Once the driver has launched, the following output should appear in the terminal window:

[ INFO] [1537745926.490663691]: starting console

s - status

cs - change state

a - appload

r - read

ra - read all

w - write

wa - write all

c - commit

q - quit

>>

# Appendix

## Control modes

typedef enum CtlModeType {

POSITION\_MODE = (uint8\_t)0,

POS\_IMPEDANCE\_MODE = (uint8\_t)1,

VELOCITY\_MODE = (uint8\_t)2,

VEL\_IMPEDANCE\_MODE = (uint8\_t)3,

EFFORT\_MODE = (uint8\_t)4,

CURRENT\_MODE = (uint8\_t)5,

OPEN\_LOOP\_MODE = (uint8\_t)6

} CtlModeType;

## Parameter list

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **Hex** | **Sub** | | **Parameter Name** | | **Unit** | | **Data Type** | |
| 4 | 2004 | | 0 | | End Effector | |  | | Boolean | |
|  |  | | 1 | | Linear Actuator | |  | | Boolean | |
|  |  | | 2 | | Continuous Rotation | |  | | Boolean | |
|  |  | | 3 | | Temperature Current Limiting | |  | | Boolean | |
|  |  | | 4 | | Spare | |  | | Boolean | |
|  |  | | 5 | | Spare | |  | | Boolean | |
|  |  | | 6 | | Regen Algorithm Enable | |  | | Boolean | |
|  |  | | 7 | | Spare | |  | | Boolean | |
|  |  | | 8 | | Control Mode | |  | | Unsigned8 | |
|  |  | | 16 | | Spare | |  | | Unsigned8 | |
|  |  | | 24 | | Spare | |  | | Unsigned8 | |
| 20 | 2014 | | 0 | | HS Telem Enable | |  | | Boolean | |
|  |  | | 1 | | MS Telem Enable | |  | | Boolean | |
|  |  | | 2 | | LS Telem Enable | |  | | Boolean | |
|  |  | | 3 | | Debug Telem Enable | |  | | Boolean | |
|  |  | | 4 | | Slave Telem Enable | |  | | Boolean | |
|  |  | |  | | spare | |  | | 3 bits | |
|  |  | | 8 | | HS Telem Rate | | Hz | | Unsigned8 | |
|  |  | | 16 | | MS Telem Rate | | Hz | | Unsigned8 | |
|  |  | | 24 | | LS Telem Rate | | Hz | | Unsigned8 | |
| 21 | 2015 | | 0 | | MS Telem Index 0 | |  | | Unsigned8 | |
|  |  | | 8 | | MS Telem Index 1 | |  | | Unsigned8 | |
|  |  | | 16 | | MS Telem Index 2 | |  | | Unsigned8 | |
|  |  | | 24 | | MS Telem Index 3 | |  | | Unsigned8 | |
| 69 | 2045 | | x | | Effort Sensor Slope | | Nm/count or N/count | | Float | |
| 70 | 2046 | | x | | Effort Sensor Offset | | Nm or N | | Float | |

## MS telem index

|  |  |
| --- | --- |
| **Index** | **Parameter Name** |
| 0 | MCTL\_Pos |
| 1 | MCTL\_Des\_Pos |
| 2 | MCTL\_Cmd\_Pos |
| 3 | MCTL\_Mech\_Pos |
| 4 | MCTL\_Vel |
| 5 | MCTL\_Des\_Vel |
| 6 | MCTL\_Cmd\_Vel |
| 7 | MCTL\_Effort |
| 8 | MCTL\_Des\_Effort |
| 9 | MCTL\_Cmd\_Effort |
| 10 | MCTL\_Iq |
| 11 | MCTL\_Des\_Vq |
| 12 | MCTL\_Cmd\_Iq |
| 13 | MCTL\_Inertia |
| 14 | MCTL\_Damping |
| 15 | MCTL\_Stiffness |
| 16 | MCTL\_Imp\_Pos |
| 17 | MCTL\_Imp\_Vel |
| 18 | MCTL\_Bus\_Volts |
| 19 | MCTL\_Bus\_Amps |
| 20 | MCTL\_PCB\_Temp |
| 21 | MCTL\_Coil\_Temp |
| 22 | ADC\_Strain |
| 23 | ADC\_Position |
| 24 | ADC\_Position (Alternative) |
| 25 | FOC\_Elec\_Pos |
| 26 | FOC\_Elec\_Counts |

## Telemetry scaling factors

#define POSITION\_CONV\_DEFAULT (6.28318530717959f)

#define VELOCITY\_CONV\_DEFAULT (8.0f)

#define EFFORT\_CONV\_DEFAULT (256.0f)

#define CURRENT\_CONV\_DEFAULT (64.0f)

#define VOLTAGE\_CONV\_DEFAULT (64.0f)

#define TEMPERATURE\_CONV\_DEFAULT (128.0f)

#define INERTIA\_CONV\_DEFAULT (32.0f)

#define DAMPING\_CONV\_DEFAULT (128.0f)

#define STIFFNESS\_CONV\_DEFAULT (1024.0f)

I have one quick question here. The initial setting number is 437851926 with number 20 status as a 7.

Is it the correct way to change 437851926 to 0 to see the absolute encoder value?

As far as your question regarding the parameter 21. 0 will set the medium speed telemetry to publish 4 messages, all of which will be absolute encoder value. This is good for initial setup. It will keep you from mixing up the medium speed telemetry messages, which won’t be labelled very well. I imagine that once you get successful messages you might want to get other useful telemetry messages. So once you get good communication you might try setting parameter 21 to 50462976. This will set hdt\_ms\_telem0 to the absolute encoder position, hdt\_ms\_telem1 to desired position, hdt\_ms\_telem2 to commanded position, and hdt\_ms\_telem3 to the mechanical position. These are all various sensor inputs before or after the gearing. Maybe that will be useful for you.

Here’s a more descriptive version:

hdt\_ms\_telem0 to the absolute encoder position – This is the position sensor of the actuator.

hdt\_ms\_telem1 to desired position – This is similar to the commanded position, but it will truncate the value if the commanded position is outside of the position limits.

hdt\_ms\_telem2 to commanded position – This pretty self-explanatory. The actuator might not be able to reach a commanded position if it isn’t valid.

hdt\_ms\_telem3 to the mechanical position – This is the mechanical position of the rotor.

Absolute encoder position and mechanical position are probably of most interest to you. One thing to note on the large actuators is that they have some ‘lash,’ meaning if you grab the arm you can physically move it a few degrees due to compliance in between the gears. The absolute encoder position should show you this lash because the sensor comes after the gearbox. The mechanical position will not show you this lash because it shows rotor position, which is before the gearbox.

For telemetry scaling:

The medium speed telemetry is packed in 16 bit, signed integers. This is how it’s packed:

(LSB) = least significant bit (MSB) = Most significant bit

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Medium Speed Telemetry | | | |  |  |  |  |  |  |
|  |  | Bit | | | | | | | |
|  |  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Byte | 0 | MS Telem Feedback 1 (LSB) | | | | | | | |
| 1 | MS Telem Feedback 1 (MSB) | | | | | | | |
| 2 | MS Telem Feedback 2 (LSB) | | | | | | | |
| 3 | MS Telem Feedback 2 (MSB) | | | | | | | |
| 4 | MS Telem Feedback 3 (LSB) | | | | | | | |
| 5 | MS Telem Feedback 3 (MSB) | | | | | | | |
| 6 | MS Telem Feedback 4 (LSB) | | | | | | | |
| 7 | MS Telem Feedback 4 (MSB) | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |

Each of the 4 is in radians in the current setting so you just multiply that by the scaling factor after separating the messages.