# Magnet mapper

## Overview

The magnet mapper is a magnetic sensor on a boom that moves in 3 axes.

The boom is moved along a particular axis by a stepper motor (for the x-axis, actually two motors working in tandem), via a stage mounted on a lead screw (two screws and stages in the case of the x-axis). The motors are Fuyu-branded NEMA-23 motors (57HS5417-22B20-500) and might have come pre-assembled with the screws.

With the development of the rotary table, another motor controls the fourth axis. While the height of this motor is slightly taller, this model (57HS56-3004A) is still compliant with the NEMA-23 form factor with the same control protocols.

Each motor that drives the boom is controlled by its own driver (the two x-axis motors are controlled by a single driver). The precise model of the driver is unknown but it is probably the Fuyu FMDD50D40NOM driver (<https://www.fuyumotion.com/nema23-stepper-motor-driver-for-fuyu-linear-motion-guide-product/>, as at 13 December 2021).

The motor driver than controls the rotary table is a DM542, which is very similar to the Fuyu one mentioned above ([Leadshine DM542 Digital Stepper Driver 20-50 VDC with 1.0-4.2A (kitaez-cnc.com)](https://kitaez-cnc.com/f/dm542.pdf)).

In turn, the drivers are driven by the MC 405 Trio motion controller. Documentation is available in the [Motion Perfect](http://www.triomotion.uk/public/software/motionPerfectv5.php) software. An out-of-date PDF manual that still seems to be reasonably applicable is here: <http://www.ps-log.si/dokumenti/Trio/Manual_6.pdf> (as at 13 December 2021).

The Trio controller receives commands, such as MOVE or DATUM, over an ethernet link (raw TCP/IP connection). For instance, the controller may be driven by typing commands directly into a PuTTY terminal, or by establishing a TCP/IP connection in LabView and programmatically sending commands.

The magnetic sensor is a P15A Hall-effect sensor from AHS Ltd (<https://www.ahsltd.com/p15a-hall-sensor>, as at 13 December 2021). It detects the magnetic field in the direction perpendicular to the face of the sensor. If the sensor is properly aligned, this is the z-axis direction of the magnet mapper.

## Coordinate system

### Direction

Clockwise motion of any of the motors (looking out of the motor along the screw) brings the stage on the screw closer to the motor. Anti-clockwise motion moves the stage away. *At the origin of the coordinate system* (see below) the stages on the y- and z-axes are close to their motors and significant motion is only possible by moving away from the motors (anti-clockwise motion). This requires sending the MC 405 command a *negative* coordinate. Conversely, at the origin the stages on the x-axis are far from their respective motors and significant motion is possible only by moving towards the motor (clockwise motion). This requires sending the MC 405 a *positive* coordinate.

Therefore, once the origin has been properly set on the MC 405:

* the x-axis runs effectively from 0 to a large positive number (as at 13 December 2021, this is set to 110mm in the LabView software);
* the y-axis runs effectively from 0 to a large negative number (-210mm); and
* the z-axis runs effectively from 0 to another large negative number (-185mm).
* The rotational axis does not have a displacement limit initialised.

Care should be taken not to provide a coordinate with the wrong sign and crash the stages into the motors.

### Origin

In general terms the origin is where the stages on the y- and z-axes nearly as close as they can be to their motors, and the stages on the x-axes are nearly as far as they can be from their motors. This corresponds to the back-right corner of the magnet mapper assembly.

Specifically, there are optical sensors on each axis for determining the origin of the co-ordinate system (setting the “datum”). When initialising the mapper using the DATUM command on the MC 405 the boom should moved towards the origin until the sensors are triggered. The boom is then moved very slightly back to take up any backlash and the zero-points for the relevant axes are set in the MC 405. The position of this origin can regularly shift by up to about 300 microns whenever the MagMapper is reinitialized.

When setting the origin using the LabView code, it is assumed that the *physical* mapper position is initially zero or positive for the x-axis, and zero or negative for the y- and z-axes (relative to the coordinate directions discussed above). If this is not the case then stages will crash into the motor or the end of the lead screw, because the stages will be moving away from the optical sensor instead of towards them.

### Measurement scale

The measurement scale is set in software on the MC 405 and is 1mm per 2,560 pulses (see the UNITS command). This is derived as follows:

* The stepper motors have 200 steps per revolution. That is, each pulse on the motor inputs will move the motor 1.8⁰.
* The lead screws have a pitch of 10mm; a full rotation of the motor will move the stage by 10mm. A single step on the motor will therefore move the stage by 0.05mm, and 20 steps on the motor will move the stage by 1mm.
* The stepper drivers are jumpered to do 8 micro-steps per step, so it takes 8 × 20 = 160 pulses on the driver to make 20 steps on the motor (and advance the stage by 1mm).
* The MC 405 internally divides the pulse rate by 16, so it takes 16 × 160 = 2,560 MC 405 “pulses” to advance the stage by 1mm.

Once the units are set in the MC 405, coordinates are effectively millimetres from the origin (taking care to maintain the correct sign, as indicated above).

A fourth stepper motor (rotational axis) has also been implemented in later stages of development. Because the native 200 subdivisions in the motor was sufficed for the practical context of measurements, the use of microstepping was not desired. However, it was found that the DM542 driver behaved abruptly running at the native 200 subdivisions. The abrupt performance was solved by double-stepping on the 400 subdivisions setting. The unit constant specified for this axis was 32 rather than 2,560 so running the command MOVE(0,0,0,1) would increment the motor 1.8⁰ clockwise (1 ÷ 200).

## Attaching the magnetic sensor

The P15A sensor requires a driving current, and then outputs a differential voltage that is proportional to the magnetic field.

The datasheet shows current passing from pin 1 (positive) to pin 4 (negative) of the device, with the differential voltage measured on pins 2 and 4. However, as at 13 December 2021 the sensor on the mapper was calibrated with the current and voltage leads interchanged, so the sensor is being run with current on pins 2 and 4 and voltage on pins 1 and 3.

The current should be 1mA, provided by a constant-current source (a lower current will only require corrections to be made to the calibration, but a higher current is likely to damage the sensor).

The calibration for the sensor is appended at the end of this document. The original Excel file is available here (as at 13 December 2021): R:\01 HTS Large Scale\11 HTS Dynamos\08 Student work\03 TJB VUW 2020\HallCalibration\PPMS20210212\2021.07.09 P15A Hall Sensor Calibration for Magnet Mapper.xlsx. The calibration shows voltage (in volts) as a linear function of magnetic field (in Tesla), i.e. V = a+bB. It is likely that you will want to invert the relationship. Note that the calibration is temperature dependent; the relationship for the correct temperature must be selected or interpolated if at a temperature that is not listed in the calibration document. The calibration also depends on the current. The voltage-field relationship should be linear in current but this has not been experimentally tested (in particular, care should be taken with the constant term).

As at 13 December 2021 the voltage is being measured using an NI USB-6210 multifunction data acquisition device, with the sensor voltage leads connected to ports AI0+ and AI0- on the device. Any other device may be used but the LabView code might require changes (see below).

## LabView code for movement and magnetic field measurements

### Connection settings

The LabView code makes a connection to the MC 405 controller, and another connection to an NI USB-6210 multifunction data acquisition device.

The connection to the MC 405 controller requires an address. As at 13 December 2021 this is: TCPIP0::192.168.0.250::23::SOCKET. 192.168.9.250 is the IP address of the MC 405 controller and 23 is the port to be used for communication. You can find (and alter) this setting on the far left of the the *main.vi* block diagram.

The connection to the NI USB-6210 also requires an address. As at 13 December 2021 this has been set to Dev2/ai0 using the DAQ Assistant function. You can change this by double-clicking on the DAQ Assistant function in case 64 of the Case block in *main.vi* (there is a True / False block inside the case block; the DAQ Assistant function is found under the False setting). It is assumed that a differential voltage can be read from the relevant address at a sample rate of 1KS/s.

### Running the code

To start, ensure that the MC 405 is on, connected to the LabView PC by ethernet, and can be pinged from the PC at the correct address (see above). Then run *main.vi*.

Next, initialise the controller by pressing the Initialise button. This will set the origin and scaling on each axis, set up limits to prevent crashing the stages on the screws, and make it possible to run other commands. Before running the initialisation ensure that the path from the sensor boom to the origin is clear. Once the controller is initialised it should not need to be initialised again unless it is turned off. In particular, as long as the controller does not lose power at any point you can stop and restart the LabView code without having to re-initialise. This is helpful because you cannot rely on the origin being precisely the same each time you initialise (it will be close but not necessarily sub-millimetre close).

Following initialisation, a typical sequence of operations is to carefully move the sensor to some known position (e.g. just above one corner of the surface of a magnet), and then to run a scan of the magnetic field at points on a 3D grid relative to this position. To do this:

* To move the sensor to the coordinate (X,Y,Z), fill in the X, Y, and Z coordinate boxes and press the Move absolute button. Ensure that you use the correct sign for each coordinate (see above). Absolute position is displayed in the Current Position control (it might take a short time to update after each move).
* To move the sensor to the coordinate (X + X increment, Y + Y increment, Z + Z increment), where (X,Y,Z) is the current position of the sensor, fill in the X increment, Y increment, and Z increment coordinate boxes and press the Move relative button. Ensure that you use the correct sign for each increment (see above). (This action also updates the Absolute position control).
* To scan the magnetic field over a 3-D grid of points, enter values in the X Start, Y Start, Z Start, X End, Y End, Z End, X Step, Y Step, and Z Step boxes and then press the Read grid button. You will be prompted for a file name to store the field readings. As at 13 December 2021 the code then makes a large array of all the points where measurements are to take place, and this can take some time (depending on the number of points). Readings will begin after the array is created. Take care to enter the correct sign for all coordinates and to make sure that the sign of the step size is consistent with the start and end points (e.g. if start is -50mm and end is -25mm, step should be 5mm and *not* -5mm). The code does some checks for validity of coordinates and will silently end the command (so not scan anything) if it finds they are invalid. It is fine to have a zero step size if the beginning and end coordinate are the same (as they will be for a 1-D or 2-D scan). The format of each line in the data file is x, y, z, average voltage, st.dev. voltage, number of voltage readings (when a measurement is taken the voltage is sampled many times and then averaged).

Note that you can also read the Hall sensor voltage directly using a test panel in the NI MAX software, which can be useful for initial positions. However, an active NI MAX reading can prevent the LabView code from reading from the DAQ output, so some care is required.

If you stop the VI while a command is running and then start it again, or there is a software error, the code will sometimes get into a state where it is reading an incomplete or nonsense response from the MC 405. In that case, clicking the Clear buffers command might help. In general, clicking this button between other commands should do no harm and might avoid errors.

### Troubleshooting

Here are some suggestions for troubleshooting if things do not work:

Before running the code:

* check that you can ping the MC 405 from the PC, at the address the code expects;
* check that there is a DAQ device connected to the Hall sensor and the PC at the hardware address that LabView expects (use NI MAX to check);
* check that current is being supplied to the Hall sensor.

If the Read grid command silently fails (the screws don’t turn even after a significant delay) check that you have entered consistent values for start, end, and step, and in particular that the sign of the step value is correct.

If there are errors you might try the following, proceeding down the list only if earlier steps don’t resolve the problem:

* click the Clear buffers button before running the next command (including after restarting the code);
* check that you are not acquiring data from the DAQ device using any other software (e.g. through a test panel in NI MAX, or other LabView code);
* stop the code, close any open VISA sessions using the procedure below, and restart the code;
* close and re-open LabView; and
* turn the MC 405 off and on and reboot the PC (the MC 405 will require re-initialisation).

To close open VISA sessions:

Run the visa.llb library (as at 13 December 2021, using LabView 2020, this is in the C:\Program Files\National Instruments\LabVIEW 2020\vi.lib\Utility directory on my PC). When the library opens, run *Open VISA Session Monitor.vi* and select the sessions you want to close.

### Structure of the code

At the core of the code is a sub-VI that accepts a string, sends it to the MC 405 controller over a raw TCP/IP connection, and outputs the sanitised response of the MC 405 as another string. As at 13 December 2021 this sub-VI is called *sendcmd.vi*. In principle you can feed any valid MC 405 command into the sub-VI, the command will be executed, and you will get the response of the MC 405 (which might be empty for some commands) back from the sub-VI. *sendcmd.vi* is therefore a good building block if you want to extend or otherwise customise the code.

One level up from *sendcmd.vi* is *processcmd.vi*. *processcmd.vi* exists to implement a proxy for the WAIT IDLE command on the MC 405, which pauses until motor motion stops. Using the WAIT IDLE command directly is problematic: in completely normal circumstances where the stages are being moved a long way, the response will take a long time, causing the READ command in LabView to time out. To avoid this, the quasi-command ALT\_WAIT is added in LabView. If *processcmd.vi* detects the ALT\_WAIT command it intercepts it and:

1. sends a command (PRINT IDLE) to query the idle status of the relevant axis on the MC 405;
2. reads the status back; and
   1. if the status is anything other than idle (-1), waits until at least 0.333s until after the command was sent and then goes back to step (1); or
   2. if the status is idle (-1), waits until at least 0.333s until after the command was sent and then exits.

The ALT\_WAIT command may be called as:

* ALT\_WAIT
* ALT\_WAIT axis(x)

where x is the number of the axis for which idleness is being tested. If the first form is used than the active axis (e.g. determined by some previous BASE command on the MC 405) is the axis used.

Any command that does not contain ALT\_WAIT is passed through untouched to *sendcmd.vi*. If you don’t need the ALT\_WAIT functionality you can just use *sendcmd.vi* directly.

The rest of the LabView code is really just assembling parameters and commands to pass to *processcmd.vi*, with the exception of the code that takes a magnetic field measurement. At the moment the field measurement is taken using the built-in DAQ Assistant function. You can edit the inputs to the DAQ Assistant function to control the number of samples taken and the sampling rate.

### Implementation and Amendments of the Rotary Table

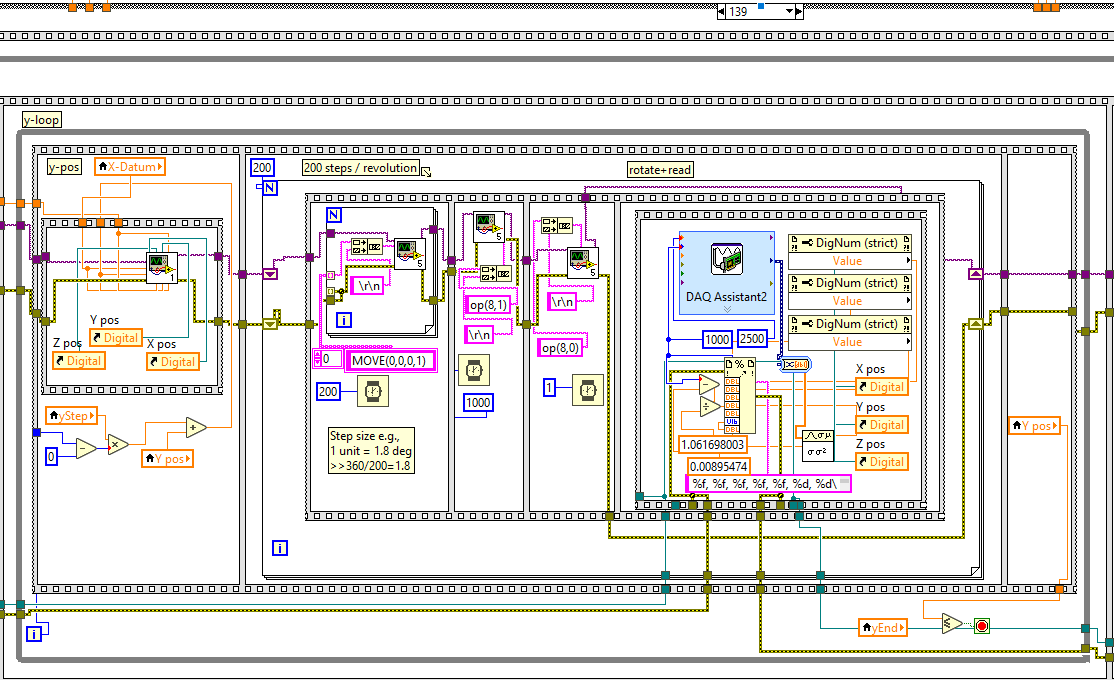
At the time of writing, the mechatronic system of the turntable currently exists without a chassis.

**Disengagement of the Stepper Motor:**

The current implementation currently disengages the stepper motor after every reading for about 1000 ms and then again engages following. This was for testing the EMI. It was found that an engaged stepper motor affects the readings by about 4 mT when the probe is about 50mm above the top base. The inherited configuration takes an averaged sample over one second from the hall sensor.

One issue that may is arise during motor disengagement is that the stepper will shift backward a fraction of a degree. It is currently unknown whether this is constant or varies slightly. If a <4 mT interference is not considered substantial, it may be wiser to keep the stepper engaged constantly throughout running the program.

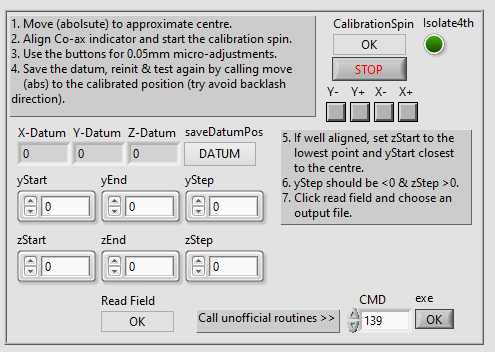
Depending on whether the sampling duration wants to be changed and the engagement toggling is to be kept, the timing values pictured below may need to be adjusted to avoid measuring whilst rotating.



**Instructions for operation:**

1. Use the *Move absolute* panel in the LabVIEW program to approximately align the boom and coaxial indicator to the centre of the rotary table and magnet.
2. Click the *CalibrateSpin* button to start rotating the motor.
3. Use the four buttons below the stop button to micro-adjust the x & y axis by increments of 0.05mm. Tune until the indicator settles to negligible oscillation.
4. Cease the spinning by clicking the stop button.
5. Click *DATUM* to save the position.
6. Re-run the initialisation routine and then use the *Move absolute* panel again, but this time, navigate to the saved position. Spin the motor up again and ensure on the coaxial indicator backlash was not an issue during calibration. If it was, recalibrate micro adjusting in only ONE direction for each axis.
7. If the coaxial indicator still looks settled, set the start and end positions, as well as the step size in mm. The step size for the z-axis should be greater than 0 as you should be traversing upward. The step size for the y-axis should be less than zero, to primarily traverse in the direction avoiding backlash.
8. Clicking the *Read Field* button should commence the measurement process.

Note: The older *Read grid* routine logs each entry into six columns. This measurement script also logs a 7th column, automatically converting the voltage into a Tesla reading, calibrated for ambient room temperature.



## Copy of calibration file

This file is: R:\01 HTS Large Scale\11 HTS Dynamos\08 Student work\03 TJB VUW 2020\HallCalibration\PPMS20210212\2021.07.09 P15A Hall Sensor Calibration for Magnet Mapper.xlsx

ATTENTION: The Hall sensor was calibrated with current through V+ and V- leads, and voltage measured across I+ and I- leads (i.e. the wrong way around). To ensure accuracy of results using this calibration, it should be used in the same way. Attach current source to V+ and V-. Read Hall voltage across I+ and I-. DO NOT SUPPLY MORE THAN 1mA OF CURRENT.

Sensor is a **P15A surface mount sensor** from Advanced Hall Sensors Ltd. The sensor was placed in the Physical Property Measurement System (PPMS, from Quantum Devices). Hall voltage was measured at a range of temperatures and magnetic fields, with a **driving current of 1mA**.

Data were plotted as:

(1) voltage vs applied magnetic field, with temperature fixed

(2) voltage vs temperature, with magnetic field fixed

Straight lines were fitted to each using ordinary least squares.

**Important**: for (2) the fits **exclude** all temperatures above 125K.

Results of fits are below.

e.g. at T=20K (actual temperatures ranging from 19.99023K to 20.01276K), V = .000485 + 1.201885 B

e.g. at T=77K, V= .000614 + 1.201747 B

e.g. at B=-0.4T, V= -0.47793 + 3.32E-5 T

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Filename | Sorting method | Sensor | Range start | Range end | Intercept | Slope |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 19.99023 | 20.01276 | 0.000485 | 1.201885 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 24.9942 | 25.00775 | 0.000416 | 1.202456 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 29.97005 | 30.01801 | 0.000412 | 1.202669 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 34.99059 | 35.00625 | 0.000444 | 1.202965 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 39.97814 | 40.02209 | 0.000474 | 1.20318 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 44.99158 | 45.00487 | 0.000487 | 1.203274 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 49.98226 | 50.01663 | 0.000539 | 1.203345 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 54.98881 | 55.02205 | 0.000567 | 1.20352 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 59.99102 | 60.00651 | 0.000582 | 1.203559 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 64.99268 | 65.00578 | 0.000617 | 1.203692 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 69.9943 | 70.00551 | 0.000619 | 1.203045 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 74.9941 | 75.00597 | 0.000634 | 1.202274 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 76.99596 | 77.00404 | 0.000614 | 1.201747 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 79.9957 | 80.00445 | 0.000561 | 1.2008 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 99.99535 | 100.0161 | 0.000449 | 1.196186 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 119.9933 | 120.0193 | 0.001607 | 1.196747 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 288.1408 | 288.2069 | 0.001007 | 1.073912 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 293.1412 | 293.1707 | 0.000895 | 1.061698 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 298.1443 | 298.1668 | 0.000731 | 1.049945 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | bytemp | 302 | 303.1394 | 303.1664 | 0.000181 | 1.036862 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -1.00006 | -0.99979 | -1.2071 | 6.02E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.90003 | -0.89999 | -1.08823 | 6.55E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.80002 | -0.79998 | -0.96646 | 6.20E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.70001 | -0.69998 | -0.84467 | 6.00E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.60001 | -0.59999 | -0.72237 | 5.97E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.50001 | -0.49998 | -0.59879 | 4.12E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.40001 | -0.39998 | -0.47793 | 3.32E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.30001 | -0.29999 | -0.35785 | 2.73E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.20001 | -0.19998 | -0.23849 | 2.19E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.1 | -0.09998 | -0.11951 | 1.78E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.09002 | -0.08999 | -0.10763 | 1.74E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.08001 | -0.07998 | -0.09574 | 1.71E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.07 | -0.06998 | -0.08383 | 1.66E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.06002 | -0.05997 | -0.07194 | 1.62E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.05002 | -0.04999 | -0.06003 | 1.56E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.04001 | -0.03999 | -0.04809 | 1.48E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.03001 | -0.02998 | -0.03613 | 1.37E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.02002 | -0.01998 | -0.02413 | 1.21E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -0.01002 | -0.00998 | -0.01203 | 9.47E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | -9.64E-06 | 8.94E-06 | 0.000392 | 3.94E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.009987 | 0.010026 | 0.012791 | -1.31E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.019984 | 0.020017 | 0.024867 | -3.75E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.029981 | 0.030024 | 0.036877 | -5.39E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.039981 | 0.040014 | 0.048825 | -6.41E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.049995 | 0.050024 | 0.060762 | -7.15E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.05998 | 0.060022 | 0.072662 | -7.66E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.069983 | 0.07001 | 0.084564 | -8.13E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.079989 | 0.080014 | 0.096474 | -8.64E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.089996 | 0.090024 | 0.108367 | -8.90E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.099981 | 0.100003 | 0.120243 | -9.32E-06 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.199984 | 0.200011 | 0.239239 | -1.32E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.299988 | 0.300019 | 0.35862 | -1.83E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.399989 | 0.400018 | 0.478765 | -2.42E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.499988 | 0.500016 | 0.599397 | -2.91E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.599989 | 0.600025 | 0.720969 | -3.59E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.699987 | 0.700018 | 0.843458 | -3.11E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.799984 | 0.800013 | 0.965325 | -2.71E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.899998 | 0.900033 | 1.087393 | -2.94E-05 |
| 2021-07-09 Sensors 302-0-303 \_001 CONVERTED.dat | byfield | 302 | 0.999795 | 1.000035 | 1.207611 | -3.10E-05 |

## Changelog

|  |  |  |
| --- | --- | --- |
| **Date** | **Action** | **Author** |
| 13 December 2021 | Document created | Tony Booth |
| 9 Febuary 2022 | Rotary table amendments | Dylan Sukha |