MDT USB Magnetometer User Manual

Version 2.1

August 21, 2017

General Description

The MDT USB Magnetometer is a single axis digital magnetometer product family that is intended for the measurement of magnetic fields at frequencies less than 100 Hz. The MDT USB Magnetometer combines a MDT TMR2705 TMR full bridge magnetic field sensor or other MDT sensor with plug and play USB data acquisition electronics to provide a single axis digital magnetometer in a simple low cost form factor. It is designed such that the user may use it directly combined with the proprietary MDT graphical user interface (GUI), controlled with a terminal emulator program, or integrated into the user's custom written program. The GUI can be run on an 8" Windows Tablet, a notebook, or desktop computer. Additionally, the probe electronics may be reprogrammed by the user using freely available open source Arduino development tools. The MDT USB Magnetometer is thus a complete development tool that allows a user a simple means for experimenting with MDT TMR sensors, sensor applications, and developing microcode and algorithms.

Applications

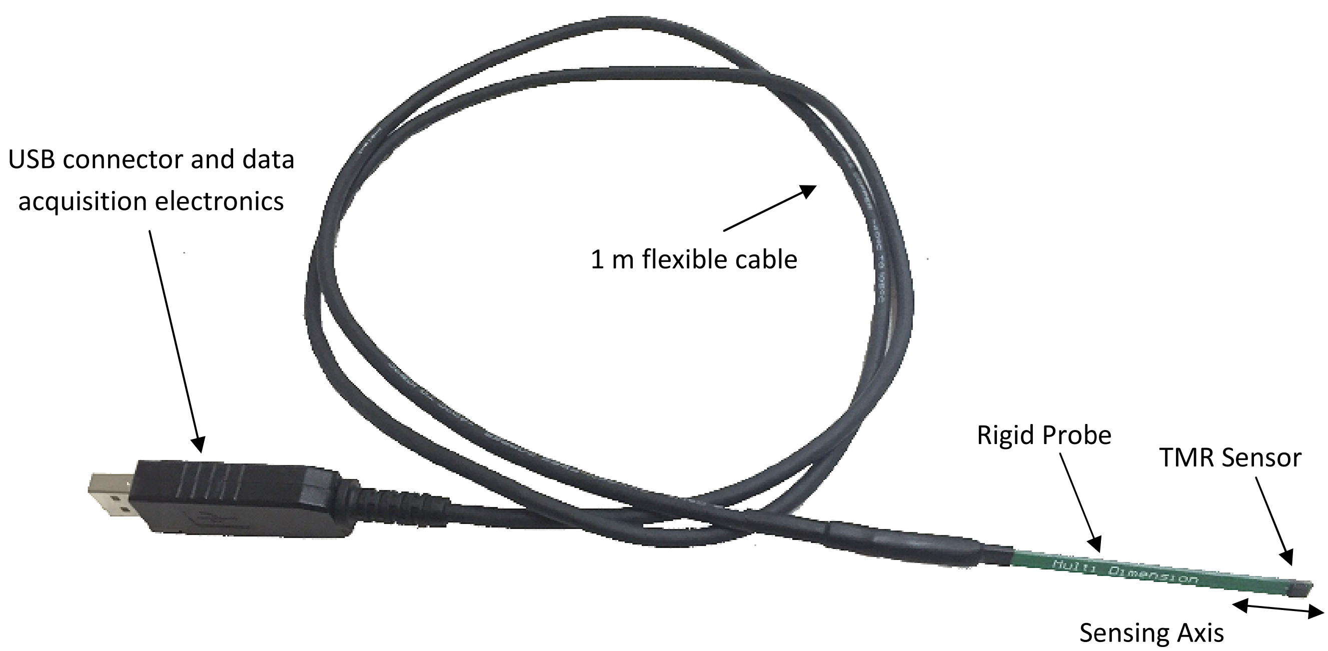
- High precision, Low speed (<100 Hz) magnetic field measurement
- magnetic field range dependent on sensor series (<1 mOe to 40 Oe, <1 Oe to 30,000 Oe)
- Data acquisition systems
- Remanent flux inspection
- Non destructive testing
- Motion and fields of magnetic objects
- Magnetic field data logging
- Current measurement
- Detection of magnetic objects
- Laboratory measurements
- Sensor signal processing algorithm development

Features

- High accuracy TMR magnetic field sensor or high field Hall sensor
- Low power operating directly from a USB port
- Resolution, speed, and sampling can be adjusted, and in low speed mode can be < 0.1 mG
- USB plug and play
- Ability to run multiple USB sensors simultaneously.
- Fully open architecture and documented interface permitting custom code development
- Compatible with any programming language that can access a COM port
- Can be used directly with a terminal emulator program on Windows, Mac, etc.
- Low cost in volume and can be considered for disposable applications
- Data acquisition hardware in the USB connector has a bootloader and it can be custom programmed through a USB port using freely available Arduino software
- Easy user calibration in Earth's field with no specialized equipment required
- Data logging graphical interface for Windows
- Standard FTDI USB drivers for compatibility and reliability

The frequency limitation of the MDT USB Magnetometer is due to the electronic design, not an inherent limitation of the sensor. The goal of this product family is to produce an affordable magnetometer that is applicable for most users' measurement needs. The MDT USB Magnetometer thus trades speed for reduced electronics cost. Generally, TMR sensors can achieve speeds of several tens of MHz. Please consult MDT for higher speed measurement applications.

The MDT USB Magnetometer has the following configuration:



A magnified view of an axial TMR probe shows the location of the sensor.

Note, the cable length may be extended without degrading performance of the magnetometer by using a female to male USB A extension cable.

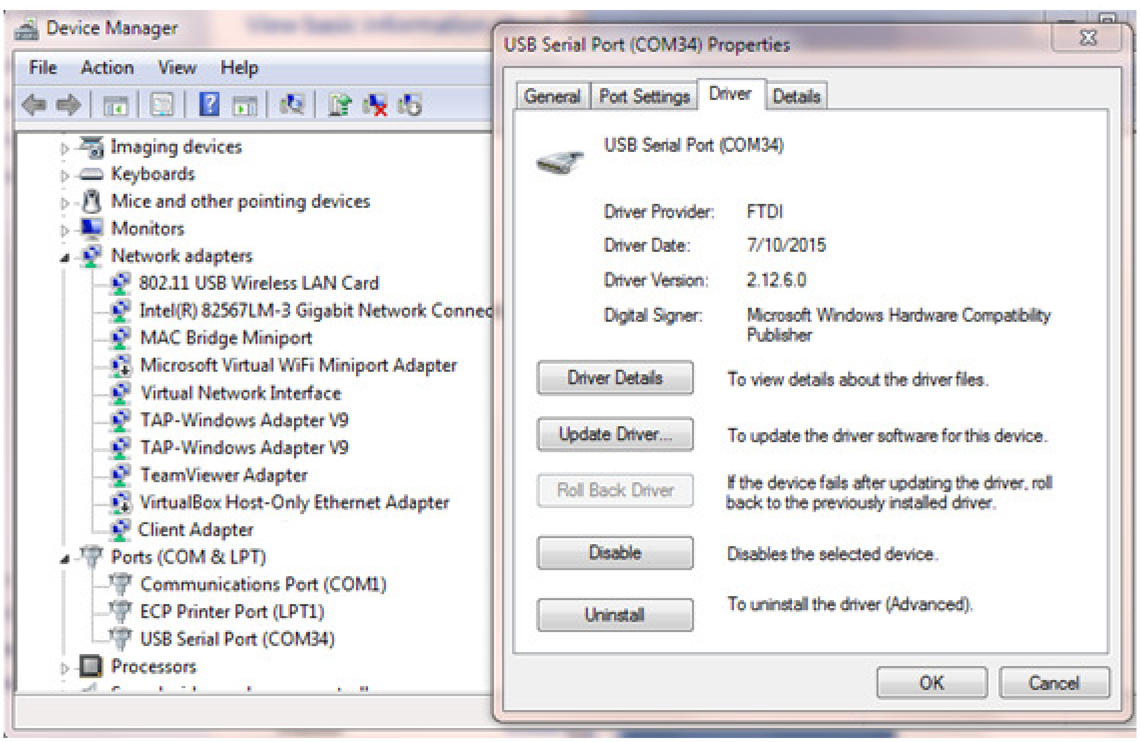
Installation

Installation in most cases is very simple. In the typical setup, the USB Magnetometer is plugged into a USB port, and the Windows operating system automatically installs the correct driver for communicating with the USB Magnetometer. The user then needs to copy the USBMagVx.exe file onto the computer. The steps are generalized as:

(1) Insert the USB Magnetometer into a USB port and allow the operating system to install the USB COM port driver.

(2) Copy the USBMagVx.exe file into a directory of choice. This executable program is included in the USB Magnetometer directory.

<u>Details of Step (1)</u> - The first time a USB Magnetometer is plugged into a USB port of a computer, the computer operating system searches for a driver for the FTDI serial to USB convertor chip in the USB Magnetometer. The USB Magnetometer will show up in the computer Device Manager as a USB Serial Port. For example, in the screenshot below, the USB magnetometer shows up as COM34. The COM port is assigned by the operating system, and when plugged into different computers, the USB Magnetometer may be assigned different COM ports on different computers. Likewise, if two USB magnetometers are plugged into the same computer, and then the different USB Magnetometers will have different COM ports assigned to them. This allows multiple magnetometers to be operated simultaneously with the same computer.



If you see a new "USB Serial Port (COMxx)" assigned after the USB Magnetometer is connected, then the operating system has properly configured the USB Magnetometer driver. Depending on the user's computer, other USB Serial Ports may be present, especially if other FTDI-based devices are present, so the best way to verify the correct installation of the USB Magnetometer driver is to remove and reinsert the USB Magnetometer form the computer's USB port and look for a corresponding change in the Device Manager COM port listing. Note, if the driver is automatically downloaded by the operating system, it may require as long as 10 minutes for this procedure to complete. Windows will provide a notification that it is searching for drivers and installing them in most circumstances.

In some circumstances, the operating system may not be able to find the FTDI COM port driver. This may happen for example, when no internet connection is available. In this case, the USB Serial Port driver may be installed manually. It is available for download at:

http://www.ftdichip.com/Drivers/VCP.htm

Please download the appropriate VCP (Virtual COM Port) executable installer program for your operating system and computer hardware. Note, there are drivers available for Windows, Mac, Linux, and Android as well.

Details of Step (2) - The USB Magnetometer user interface can be copied onto the user's compter. It is presently only available for Windows computers and has been verified on Windows XP, 7, 8, and 10. The installer is located at

<Removable drive letter>:\MDT USB Magnetometer\USBMagVx.exe

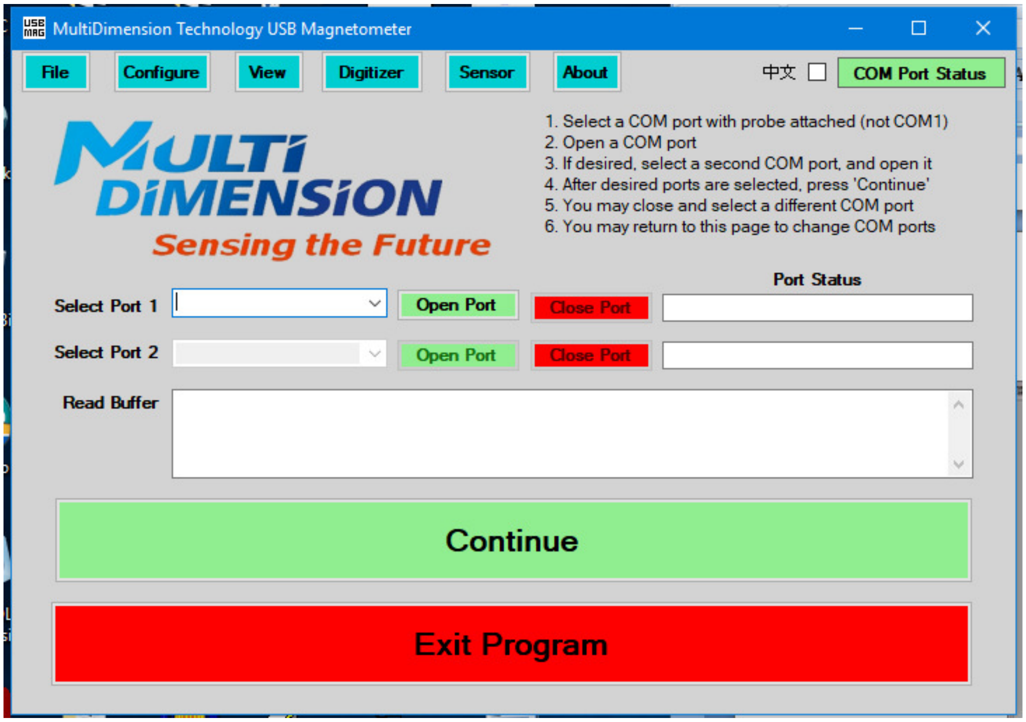
The program can be copied directly to the desktop, or it may be placed in a program folder and associated with a shortcut in the start menu. In Windows 10, this can be done as follows:

1. use the %appdata%\Roaming\Microsoft\Windows\Start Menu\Programs location, which belongs to your user account, so no additional privileges are required.

2. Create a folder there and copy the USBMagVx.exe program into the folder.

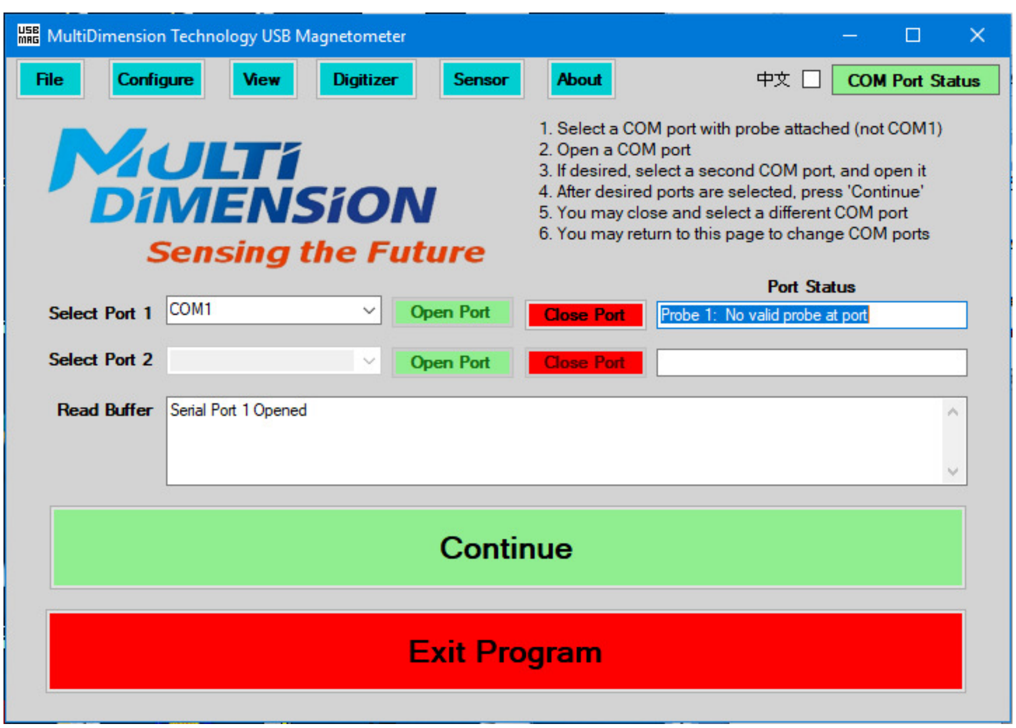
3. The folder will then appear on your start menu after a few minutes or after restart.

Once the program is installed, please start the USB Magnetometer GUI program.

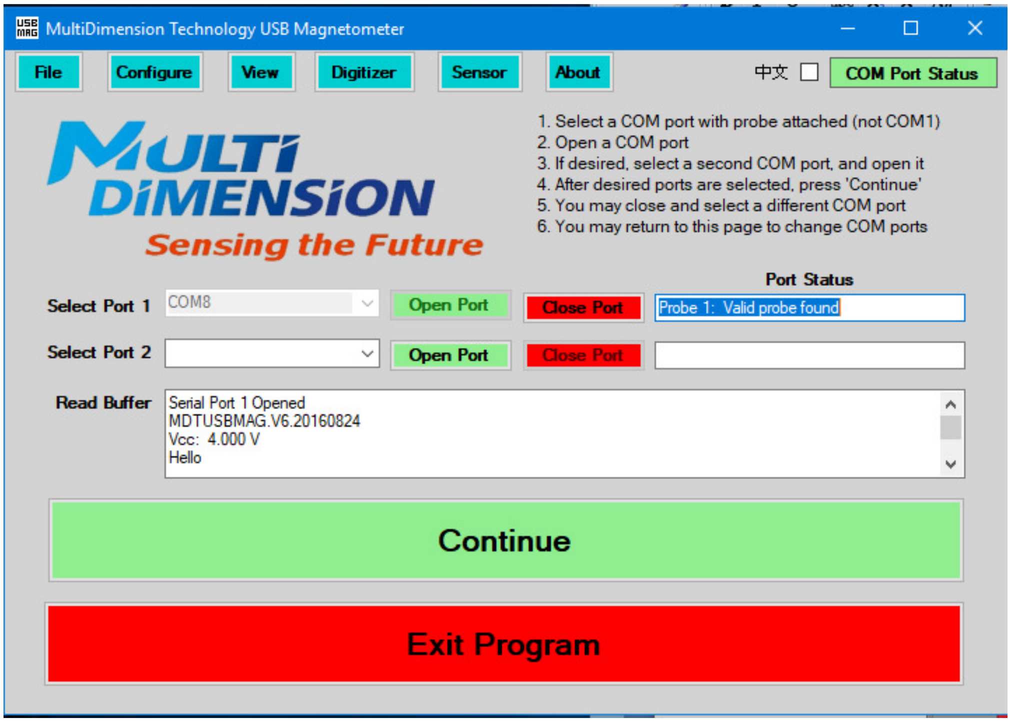


Pull down the "Select Port 1" down menu and select the appropriate port for the USB Magnetometer. Then press the "Press to continue" button. You may optionally quit the program at this point by pressing the "Exit Program" button.

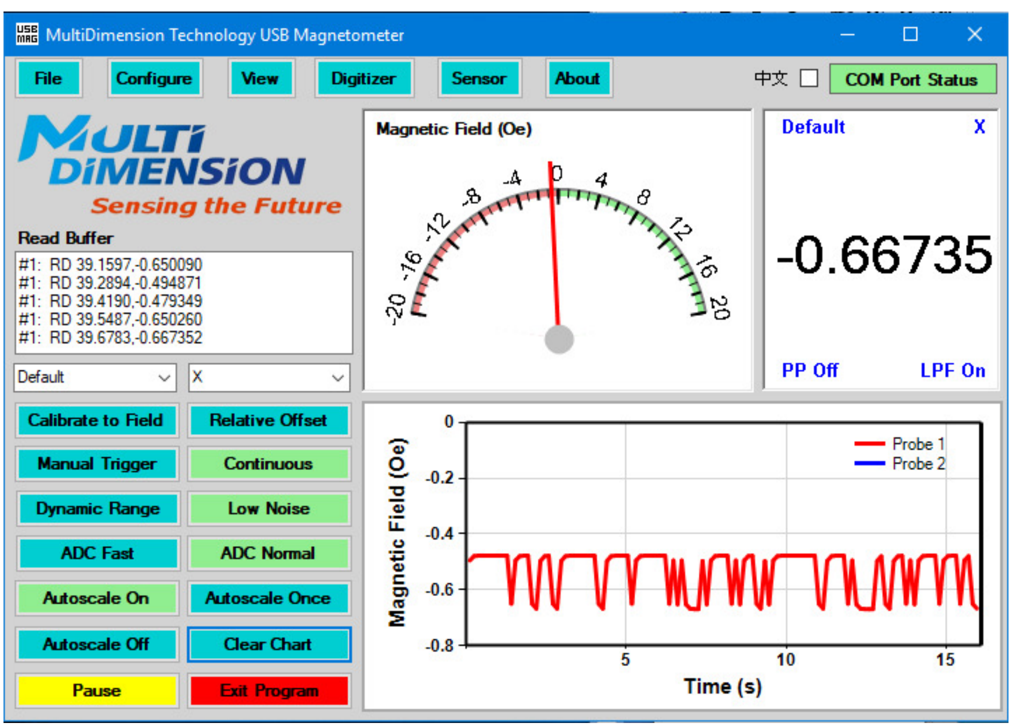
Do not worry about choosing an incorrect port. The software automatically tests the device at the selected port number, and if incorrect, you will see a message as follows:



Once the proper port is selected, you will see the "valid port" message in the Read Buffer window of the dialog. You may also see the red LED of the USB connector flashing.



A second probe can be opened at this point if the user wishes to monitor two probes simultaneously. In any event, after the "Continue" button is pressed, the program will go through an obvious initialization routine, and the user will be redirected to the main viewing window.



If everything went well, data should start streaming to the chart at the lower right, and the Read Buffer and Magnetic Field displays will change as the measured field value changes. Depending on the probe that is being used, you can verify operation either by moving the probe in Earth's field, or you can move a magnet past the probe to verify the hardware is responding.

Troubleshooting the Installation

If the USB Magnetometer does not show up as a COM port in the Serial Port drop down menu, then check the following:

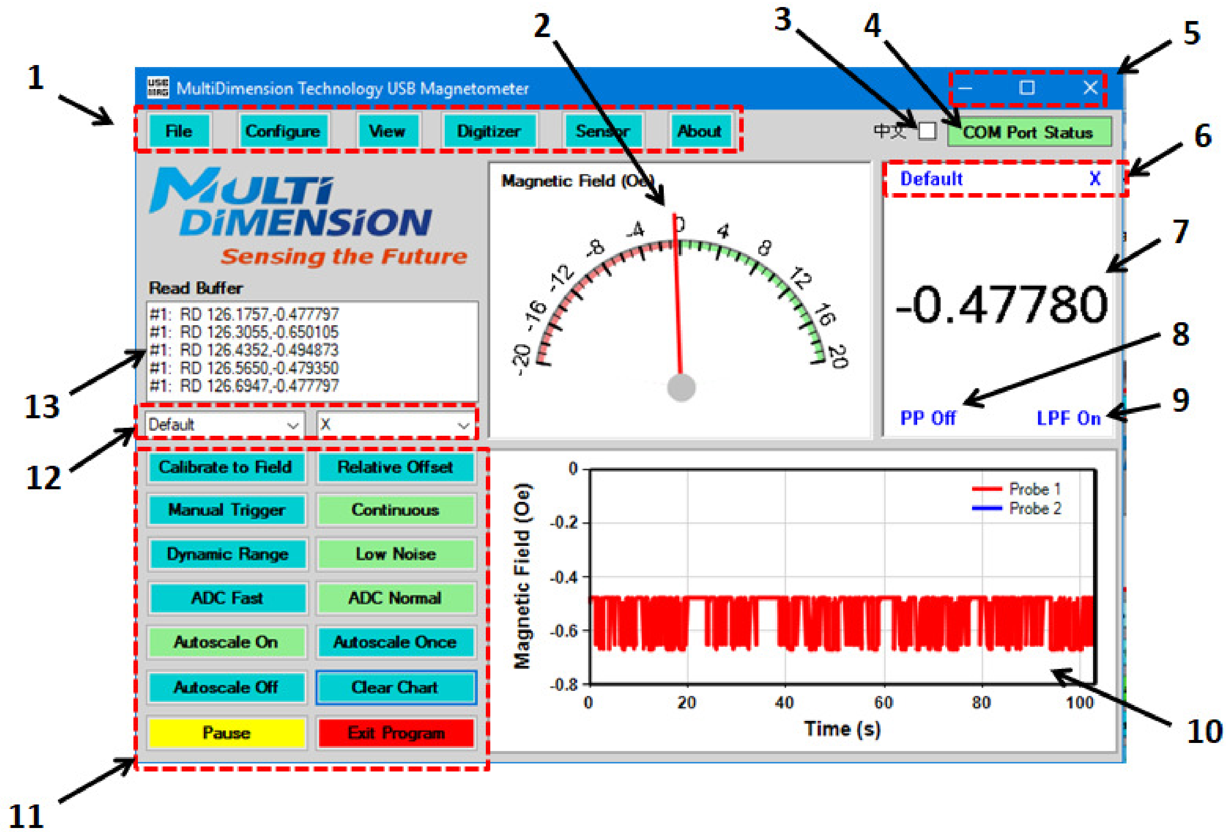
1. Does the USB Magnetometer show up as a USB Serial Port in the Device Manager? If not, try unplugging the USB Magnetometer from and plugging back into the computer's USB port. If this fails, then reinstall the FTDI drivers manually.

2. Close and restart the MDT USB Magnetometer GUI.

3. Verify no other programs are using the COM port. The MDT USB Magnetometer GUI cannot run simultaneously with a terminal emulator program.

Graphical User Interface (GUI) Operation

The layout of the GUI and a brief description of the features is now presented.



Item 1 – Runtime Menu Area.

Item 2 – Magnetic Field (Oe) analog meter.

Item 3 – Language selection control.

Item 4 – COM Port Error indicator LED. Normal operation is indicated by green. An error requiring restarting the program is indicated by red.

Item 5 – Standard accelerators used for minimizing, maximizing, and closing the window.

Item 6 – Selected Probe indicators.

Item 7 – Digital display indicating the magnetic field detected by the magnetic sensor.

Item 8 – Post processing (PP) indicator used to indicate if data smoothing has been applied to the H (Oe) vs. time (s) chart. Data smoothing is moving window is applied to the data after it is acquired.

Item 9 – Indicator used to indicate the digitizer low pass filter is enabled.

Item 10 – H (Oe) vs. time (s) chart

Item 11 – Quick Access Buttons for setting runtime parameters.

Item 12 – Controls for selecting the probe and axis to monitor in the analog gauge and digital meter.

Item 13 – Real-time display of the COM port output values from the USB Magnetometer and GUI messages.

Quick Access Button Details

Button Name	Function
Calibrate to Field	Opens a file dialog window used for calibrating the sensitivity and offset of the sensor.
Set/Disable Relative Offset	Applies and reversibly removes an offset value to/from the sensor output. The offset value is computed automatically based on the present reading. When removed, the offset returns to the originally calibrated value. This is useful for zeroing a measurement when there is a DC background field. Note the button name toggles between enable and disable depending on the status of the magnetometer.
Set Man. Trigger	"Set Manual Trigger." This button is used in order to change the sampling from continuous streaming read mode to a mode where readings are only acquired when the "Set Man. Trigger" button is pressed. This is useful when mapping fields around a magnetic object. Combined with data averaging, it enables very precise mapping of magnetic fields. Note this button color changes to yellow and flashes when the Manual Trigger Mode is enabled.
Continuous	This button sets continuous streaming read mode. It often needs to be pressed after selecting runtime menu items. This button changes to "Set Continuous" when the manual trigger mode is enabled.
Dynamic Range	Preamplifier gain set to 1x to allow the maximum possible magnetic field range.
Low Noise	Preamplifier gain set to 8x to allow the best possible magnetic field resolution.
ADC Fast	The ADC is set to 12-bit resolution. This provides about 100 Hz sampling frequency.
ADC Normal	The ADC is set to 16-bit resolution. This provides about 8 Hz sampling frequency.
Autoscale On	The maximum and minimum values of the vertical scale of the chart constantly update to the maximum and minimum values of the data displayed in the chart.
Autoscale Once	Set the maximum and minimum values of the vertical axis of the chart to the present maximum and minimum values of the data displayed in the chart. Do not update the maximum and minimum values of the vertical axis as the data changes.
Autoscale Off	Sets the vertical scale of the chart to the present maximum and minimum values. Do not update the maximum and minimum values of the vertical axis as the data changes.
Clear Chart	Clear all chart data and reset the timer to zero.
Pause/Resume	This button is used to temporarily pause a measurement. Note however, the timer will continue to count time since the first measurement point.
Stop	This button is used to quit the program.

Runtime Menu Details

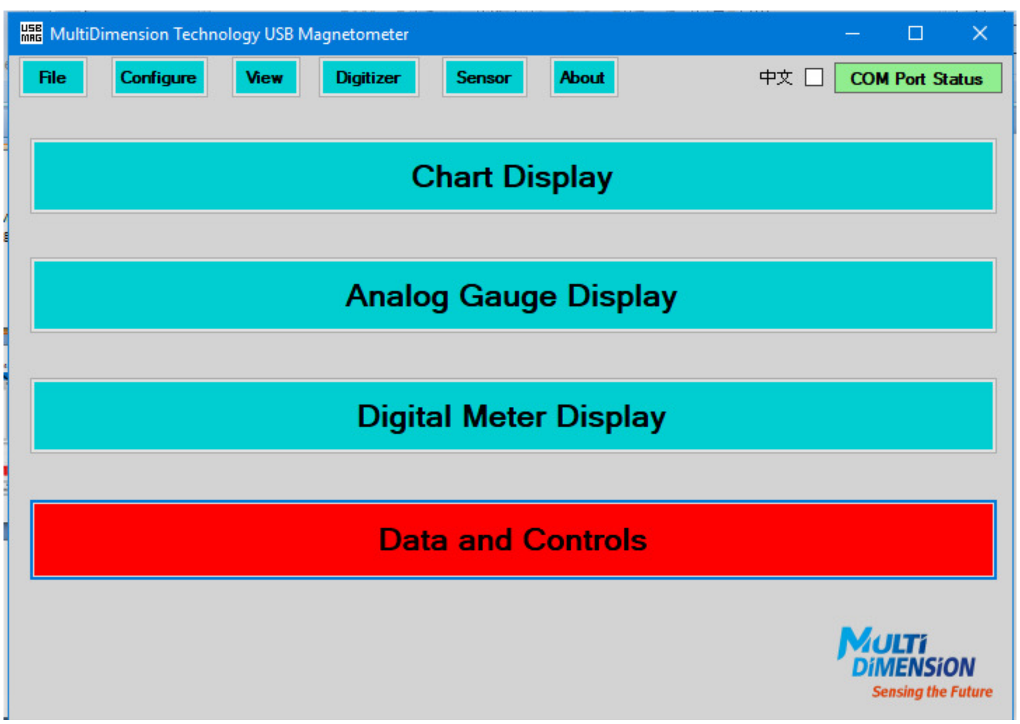
Top Level	Submenu	Function
File	Export Chart Data	Opens a file dialog window, and then exports the data presently displayed in the H (Oe) vs. time (s) chart to a CSV file defined by the user.
File	Stop	Quits the program.
Configure	COM: Open Port	Open the selected COM Port
Configure	COM: Close Port	Close the selected COM Port
Configure	COM: Continue	Initialize Open COM Ports and begin logging data
Configure	COM: Exit Program	Close any open COM Ports and exit the program
Configure	Chart: Clear Data	Clears the data in the H (Oe) vs. time (s) chart and resets the time axis.
Configure	Analog Gauge: Set to Chart Maximum	The analog gauge maximum and minimum values will be set to reflect the maximum absolute value of the chart limits.
Configure	Analog Gauge: Set to Chart Maximum	The analog gauge maximum and minimum values will be set to the default for the presently selected probe.
Configure	Analog Gauge: Force Range	The analog gauge maximum and minimum values will be set to the user specified value
Configure	Chart: Data Points	The maximum number of data points displayed on the chart
Configure	Chart: Rolling Buffer	When selected, the chart window will display the most recent number of data points acquired and discard earlier data points. This is useful for visualizing data.
Configure	Chart: Fill Buffer and Stop	When selected, the chart window will fill up to the number of data points selected, and then stop acquiring data. This is useful for unattended data logging.
Configure	Chart: Chart Legend On?	Displays/hides the chart legend
Configure	Chart: Smooth Data	Uses a moving window to smooth the data in the H (Oe) vs. time (s) chart. The smoothing operation is applied to the data that is exported.
Configure	Chart: Chart Clear	Clear the accumulated data buffer and reset the time to zero
Configure	Chart: Autoscale Once	The chart will autoscale once based on present data displayed, and then turn Autoscale off.
Configure	Chart: Autoscale On	The chart is set to autoscale continuously.
Configure	Chart: Autoscale Off	This fixes the maximum and minimum values of the vertical axis of the chart at their present values. Note, the chart scale maximum and minimum values can be edited in this case.
Configure	Chart: Return	Returns the view to the default "Data and Controls" viewing mode.
Configure	Return	Returns the view to the default "Data and Controls" viewing mode.

View	Chart Display	Displays only one large real-time H vs. time chart
View	Analog Gauge Display	Displays a large analog gauge, and a small digital meter. The analog gauge and digital meter show data from the currently selected probe, and the range of the analog gauge can be selected from the range selector at the bottom right.
View	Digital Meter Display	Displays a large digital meter, and small selectors for choosing the probe and axis to monitor. The display also contains indicators for acquiring statistics on the selected waveform. The statistics measurements are enabled when the “Stats” button is pressed.
View	Data and Controls	Returns the view to the default “Data and Controls” viewing mode.
Digitizer	Enable LPF	Turn on real-time filtering of the USB Magnetometer data, where a causal digital low pass filter is applied to the data in real time before it is sent over the USB port.
Digitizer	Disable LPF	Disables the real-time low pass filtering.
Digitizer	Command Line	Open a terminal emulator window for allowing a user to enter text commands directly to the magnetometer.
Digitizer	Set ADC Bits	Set the number of digitizer bits.
Digitizer	Dynamic Range	Preamplifier gain set to 1x to allow maximum field range.
Digitizer	ADC Fast	Preset for fast data acquisition mode. This mode is 12-bit and LPF On. The speed is roughly 100 Hz.
Digitizer	Number of Samples	Set the number of samples to be averaged for each data point.
Digitizer	Low Noise	Preamplifier gain set to maximum to permit lowest noise measurement.
Digitizer	Normal Mode	Preset for 16-bit ADC resolution with LPF On. The speed is roughly 8 Hz.
Digitizer	Return	Returns the view to the default “Data and Controls” viewing mode.
Sensor	Auto Offset	Computes an average offset voltage based on the present values read from the TMR sensor, and overwrites the calibration value used during run-time. It does not overwrite the offset stored in EEPROM. This is useful if the user has a zero-gauss chamber and wishes to fine tune the offset for a particular measurement.
Sensor	Manual Offset	Average offset voltage based on the present values read from the TMR sensor, and overwrites the calibration value used during run-time. It does not overwrite the offset stored in EEPROM. This is useful if the user has a zero-gauss chamber and wishes to fine tune the offset for a particular measurement.
Sensor	Manual Sensitivity	Allows the user to overwrite the presently used sensitivity calibration, but it does not overwrite the permanent value stored in the EEPROM.

Sensor	Display Calibration Coefficients	Allows the user to view all probe calibration data, the sensor ID, and the firmware revision. It also specifies the status of the calibration and linearization parameters stored in the EEPROM.
Sensor	Calibration to Field	Opens the TMR sensor calibration dialog. It permits the user to temporarily overwrite the problem calibration or set new values permanently in the EEPROM.
Sensor	Return	Returns the view to the default "Data and Controls" viewing mode.
About		Provides a description of recent relevant changes and contact information for suggested changes.

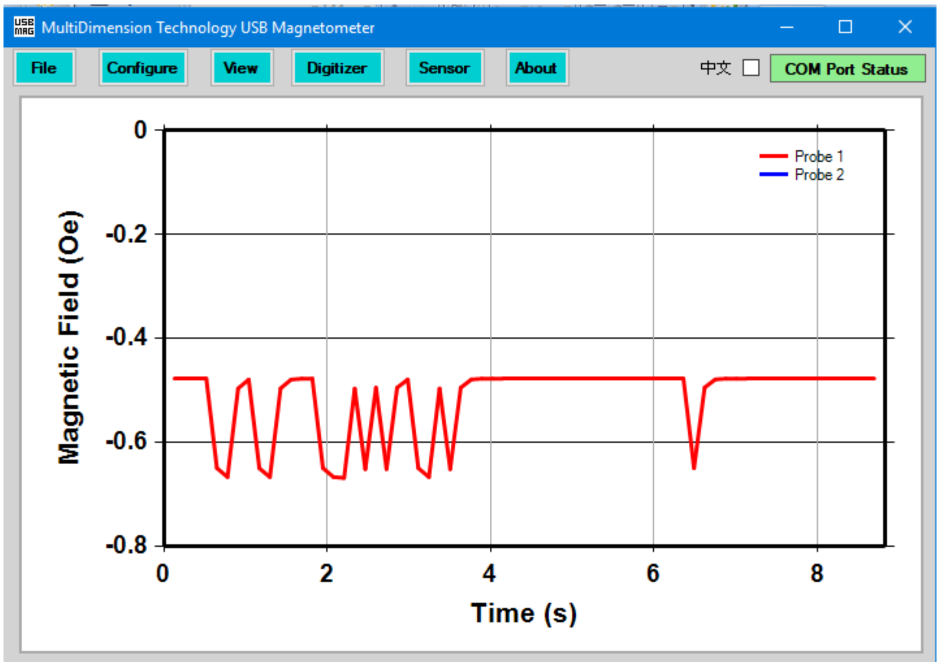
<u>Viewing Modes</u>

Three different viewing modes may be accessed in view submenu

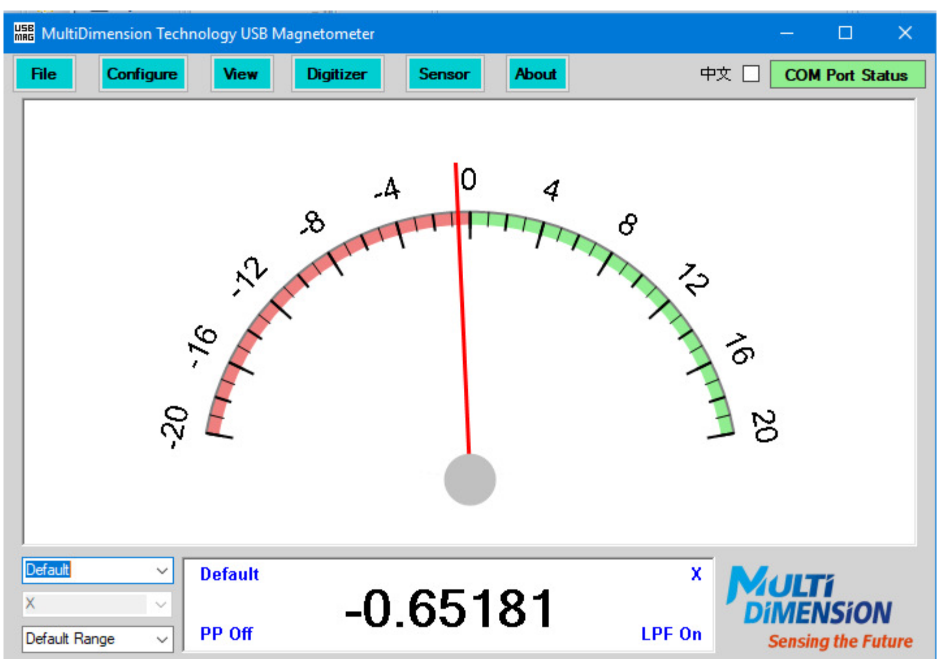


These appear as follows:

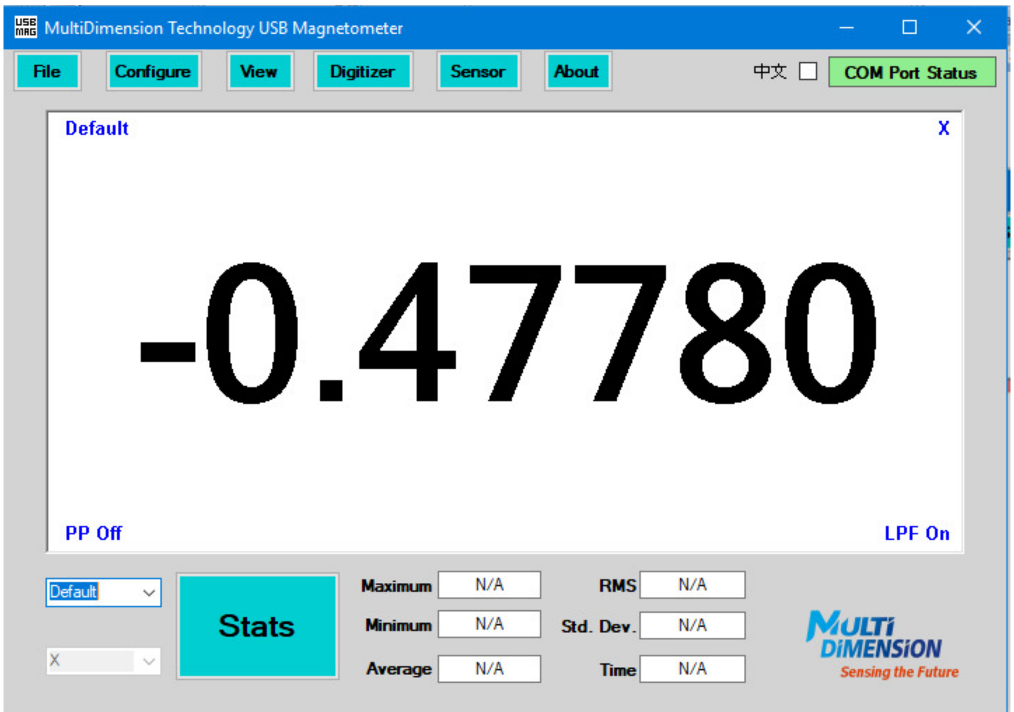
1. Chart Display – this chart displays the same date seen on the default chart page.



2. Analog Gauge Display – here the selected probe and axis may be changed. The analog gauge range may be selected from the various presets in the selection box at the lower left. There is a small digital meter below the analog gauge.



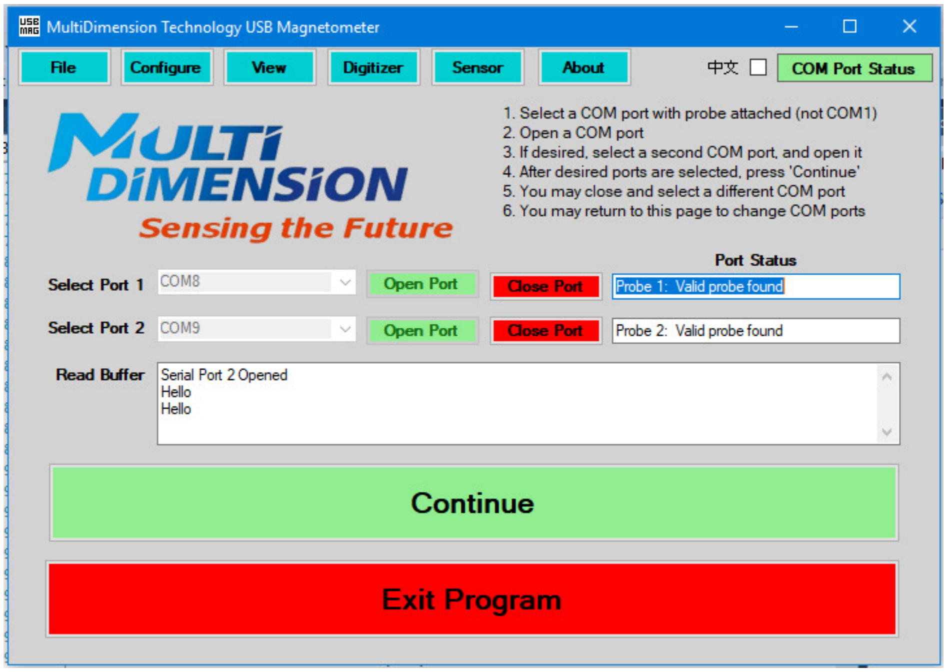
3. Digital Meter Display – This chart shows a large digital display of the presently selected probe and axis. The “Stats” button is used to acquire statistics on the present measurement. The data provided is the Maximum value, the minimum value, the mean average, the RMS average, the standard deviation, and the elapsed time of the measurement. Note the standard deviation indicator shows the RMS noise of the measurement under the present configuration as affected by ADC and averaging settings.



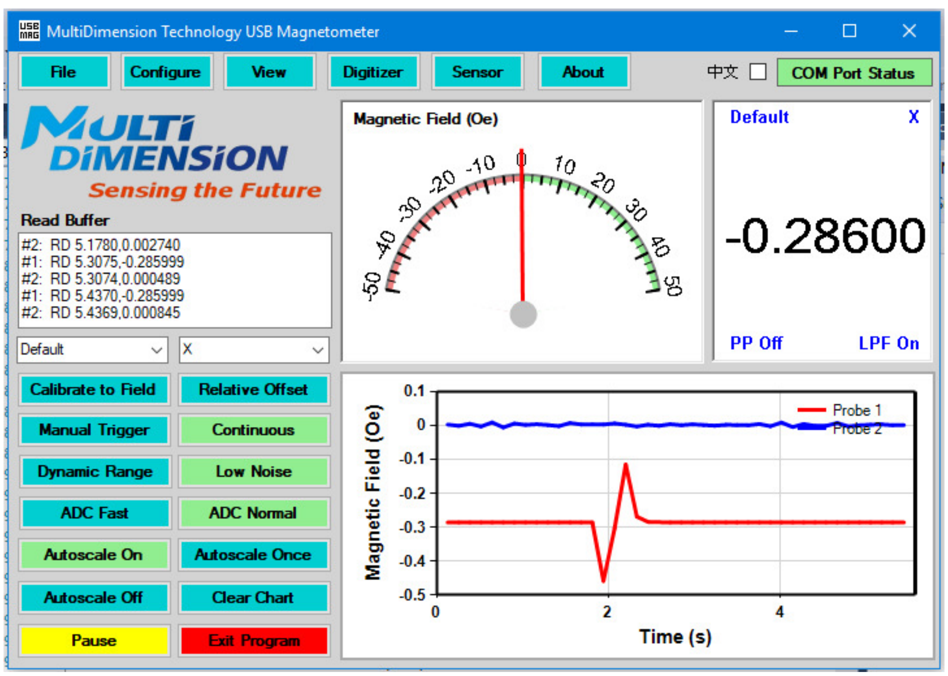
Note the “Data and Controls” button on the “View” submenu will return the GUI to the default viewing mode.

Reading Multiple USB Magnetometers Simultaneously

The USB Magnetometer probes do not interfere with each other, and several may be accessed simultaneously. The USB Magnetometers need not be of the same type. Additionally, multiple copies can be run on the same computer, each of which accesses different USB Magnetometer probes simultaneously. These capabilities are useful for building arrays of probes. The number of probes is limited by the user’s computer hardware.

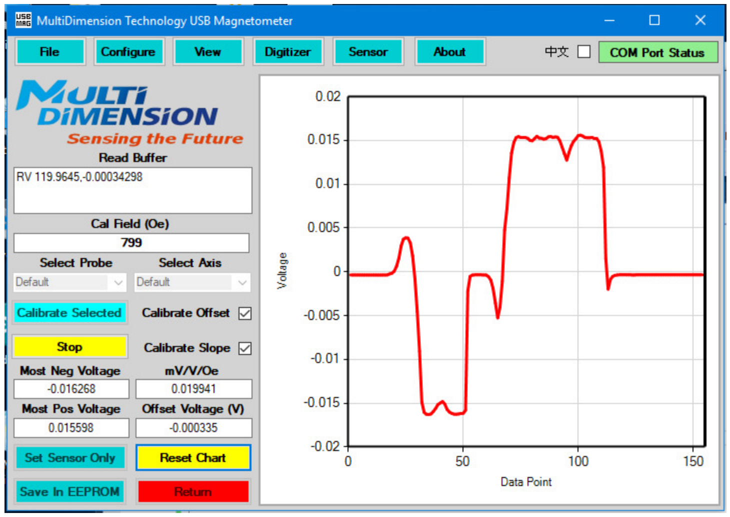


After pressing continue, traces for both magnetometers should appear in the chart, and the Read Buffer will show data for both probes.



Probe Calibration

The probe may be calibrated each use to correct offset and sensitivity. There are several methods to do this. The most convenient, which does not require any special equipment is an Earth's field calibration. To do so, select the "Calibrate to Field" button, and a window similar to that shown below will appear.



The calibration proceeds using a known reference magnetic field. The reference can be a calibration magnet, such as the MDTCAL01 used for high field calibration of probes such as the TMR2510x series, or Earth's field for probes such as the TMR2705x series.

For Earth's field calibration, the user needs to know the magnitude of Earth's field at his or her location, and that can be obtained from a website like The National Center for Environmental Information, which lists world-wide magnetic field values

http://www.ngdc.noaa.gov/geomag-web/?model=igrf#igrfwmm

So, for example if we look up a city in China, for example Nanjing, we get this result:

Magnetic Field

Model Used:	WMM2015
Latitude:	32° 2' 53" N
Longitude:	118° 46' 8" E
Elevation:	0.0 km Mean Sea Level

Date	Declination (+ E | - W)	Inclination (+ D | - U)	Horizontal Intensity	North Comp (+ N | - S)	East Comp (+ E | - W)	Vertical Comp (+ D | - U)	Total Field
2016-02-23	-5° 31' 8"	48° 35' 48"	32,949.8 nT	32,797.1 nT	-3,168.9 nT	37,369.8 nT	49,821.6 nT
Change/year	-0° 4' 16"/yr	0° 5' 35"/yr	-30.6 nT/yr	-34.4 nT/yr	-37.7 nT/yr	87.7 nT/yr	45.5 nT/yr
Uncertainty	0° 17'	0° 13'	133 nT	138 nT	89 nT	165 nT	152 nT

The total field value listed at the top of the right most column is the value we need to use. We need to convert it to Oe for the USB Magnetometer. The conversion is simple.

1 nT ÷ 10^5 ~ 1 Oe

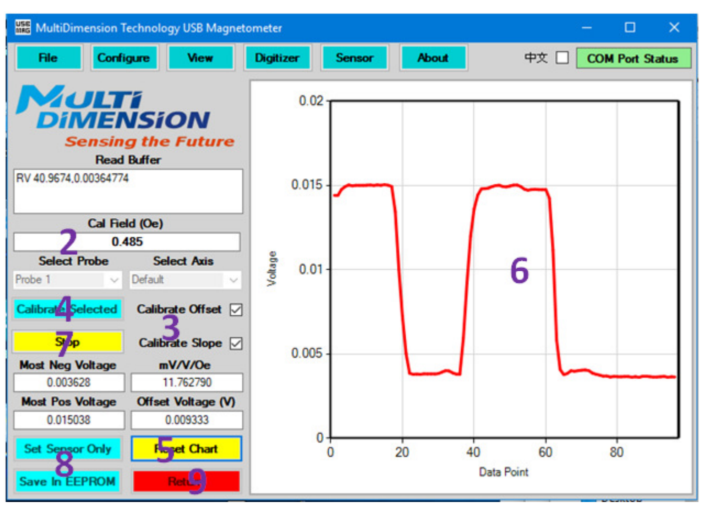
So move the decimal point in the nT data 5 places to the left.

49,821.6 nT → 0.498216 Oe

Two minor points that can be helpful, although the user does not need to know this, understanding the meaning of the inclination and declination values listed in the left most columns can be helpful for determining if maximum and minimum field values are being performed in an area where there is no distortion. The second column indicates the angle of the magnetometer probe with respect to a level surface for which the maximum magnetic field magnitude will be observed is helpful when first trying to find the orientation of maximum and minimum magnetic field. The declination is the angle by which the magnetic field deviations from true north. So if you find the maximum and minimum values occur along an incorrect direction or at a weird angle with respect to a level surface, then the magnetic field at your location is probably being distorted by a magnetic object.

In any case, even without knowing the inclination and declination, the calibration procedure is:

1. Press the “calibrate to field” button.
2. Set the magnetic field value appropriate for your location (generally around 0.5 Oe)
3. Make sure both sensitivity and offset checkboxes are selected
4. Select the desired probe and Press “Calibrate Selected”
5. Reset the chart in the calibration window
6. Swing the magnetometer around along the North-South axes in space to find the maximum and minimum voltage values, as shown in the sensor raw voltage plot on the right hand side of the window.   The bottom left of the window will show the computed sensitivity and offset. The sensitivity should be close to the sensitivity value for the sensor, and in the case of TMR2705, in the range of 10 to 20 mV/V/Oe.
7. Press “Stop”
8. Save the calibration in either the flash for future use or just write the calibration to the sensor for immediate use only
9. Exit the calibration window
10. If the offset has not been adequately corrected, then perform one of the offset calibration procedures

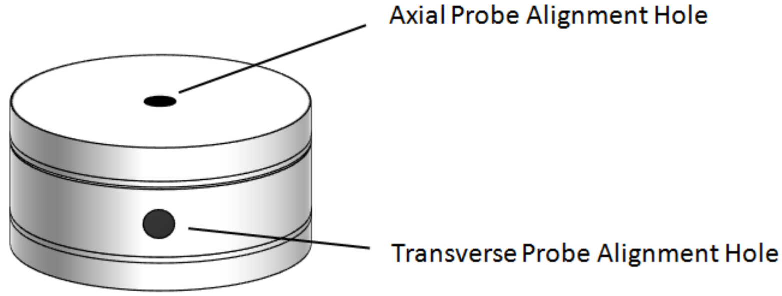


Some tips for accurate calibration.

- Note also that the maximum magnetic field is usually not along an axis that is parallel to the floor. There is always a declination angle, and the maximum field may in fact be largest at a very steep angle.
- Make sure no ferromagnetic objects are nearby, so that the value of Earth's field is not distorted.
- North and South directions may be found using a compass or a cell phone's compass.

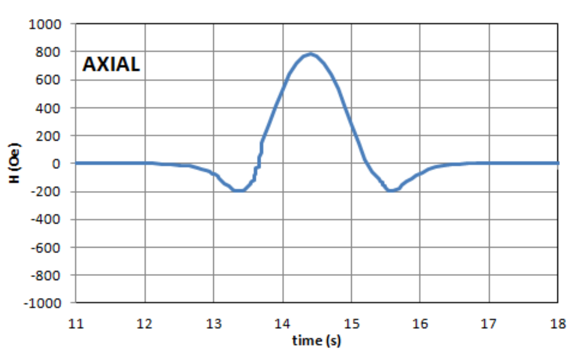
Calibration of TMR2510x Series Probes Using a Calibration Magnet

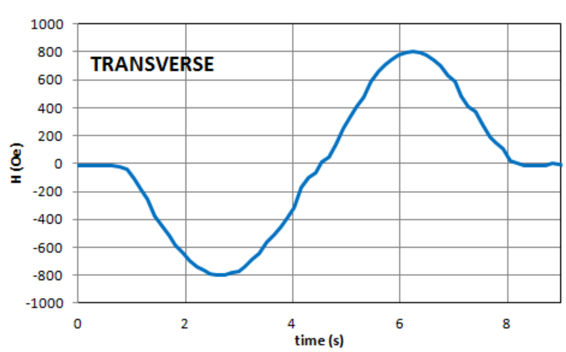
The CAL01 reference magnet provides a stable and convenient method to calibrate the USB 2510x series magnetometers.



CAL01 IS NOT FOR USE WITH THE USB2705 MAGNETOMETER.

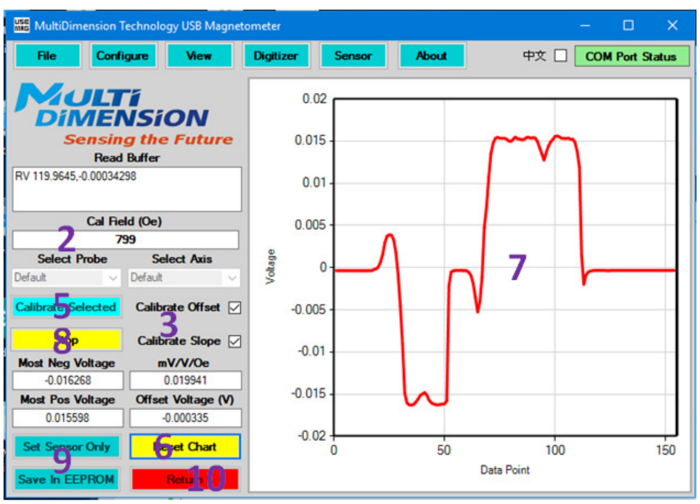
The CAL01 reference magnet is composed of two axially magnetized permanent magnet rings that allow a probe to access the working region of the fixture from both axial and transverse orientations, making the calibration fixture useful for both axial and transverse probe calibration. The plots below show the peak field values when moving an axial probe through the axial hole and when rotating the magnet around a transverse probe in the transverse hole.

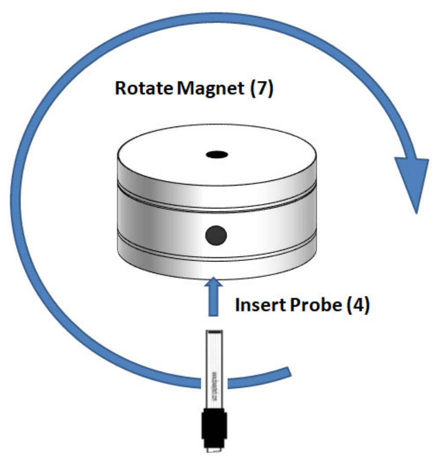




Transverse Probe Calibration Procedure

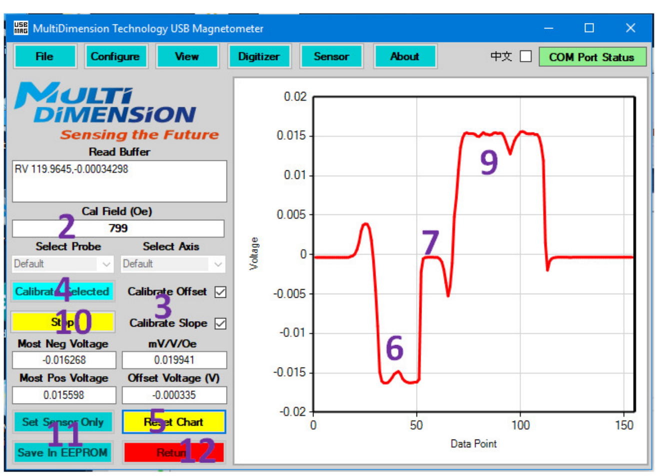
1. Press the “calibrate to field” button.
2. Set the calibration field value to the magnetic field value indicated on the side of the calibration assembly
3. Make sure both sensitivity and offset checkboxes are selected
4. Insert transverse probe all the way into the transverse probe alignment hole.
5. Select the desired probe and Press “Calibrate Selected”
6. Reset the chart in the calibration window
7. Rotate the magnet in order to sample the largest positive voltage and the largest negative voltage. The probe can be wiggled and the magnet rotated back and forth around the field extrema angles in order to sample the maximum and minimum values.
8. Press “Stop”
9. Save the calibration in either the flash for future use or just write the calibration to the sensor for immediate use only
10. Exit the calibration window
11. If the offset has not been adequately corrected, then perform one of the offset calibration procedures

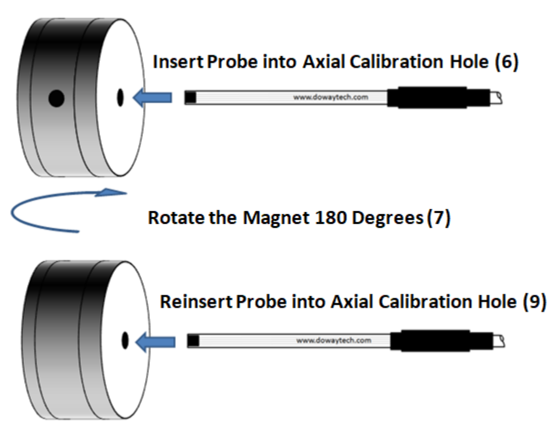




Axial Probe Calibration Procedure

1. Press the “calibrate to field” button.
2. Set the calibration field value to the magnetic field value indicated on the side of the calibration assembly
3. Make sure both sensitivity and offset checkboxes are selected
4. Select the desired probe and Press “Calibrate Selected”
5. Press Reset Chart
6. Push the axial probe into the axial alignment hole to find the maximum positive (negative) voltage reading, you can wiggle the probe some and move in and out of the calibration assembly to verify the maximum value has been sampled.
7. Hold the probe steady and pull the calibration assembly away.
8. Flip the calibration magnet in order to reverse the magnetic field
9. Push the axial probe into the axial alignment hole to find the maximum negative (positive) voltage reading, you can wiggle the probe some and move in and out of the calibration assembly to verify the maximum value has been sampled.
10. Press “Stop”
11. Save the calibration in either the flash for future use or just write the calibration to the sensor for immediate use only
12. Exit the calibration window
13. If the offset has not been adequately corrected, then perform one of the offset calibration procedures
14. If the offset has not been adequately corrected, perform one of the offset calibration procedures





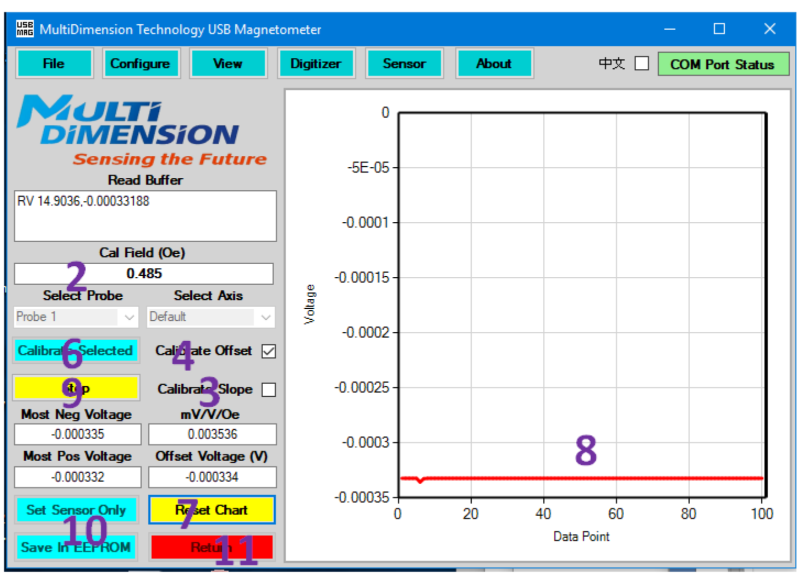
Zeroing a Probe

There are several methods that may be used.

- Autozero the probe with the sensing axis aligned east-west
- Autozero the probe inside a zero gauss chamber
- Manual offset
- Calibration method to permanently change the offset

Calibration Procedure

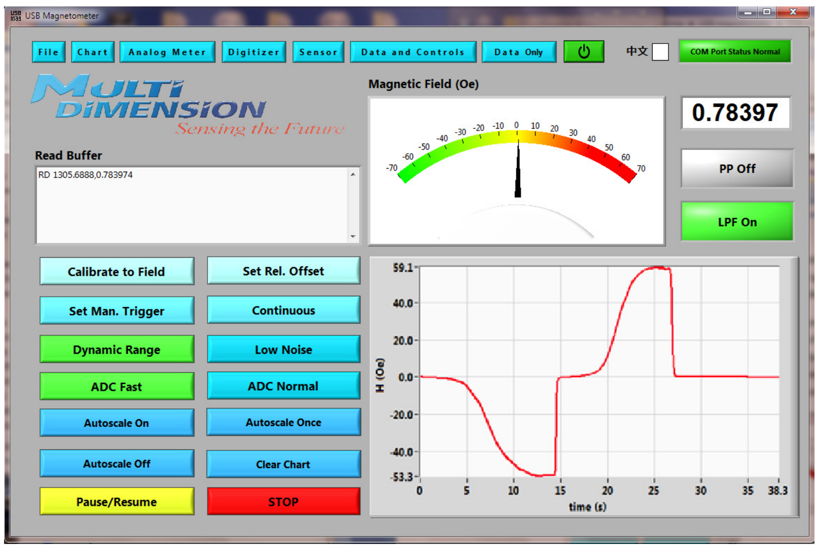
1. Press the “calibrate to field” button.
2. The Cal Field does not matter, any value will be fine
3. Make sure the sensitivity checkbox is **not** selected
4. Make sure the offset checkbox is selected
5. Place the sensor in a zero gauss chamber OR align the sensing axis along the east-west direction
6. Select the desired probe and Press “Calibrate Selected”
7. Press Reset Chart
8. Acquire several seconds of voltage vs time data, verifying that the background value is stable
9. Press “Stop”
10. Save the calibration in either the flash for future use or just write the calibration to the sensor for immediate use only
11. Exit the calibration window



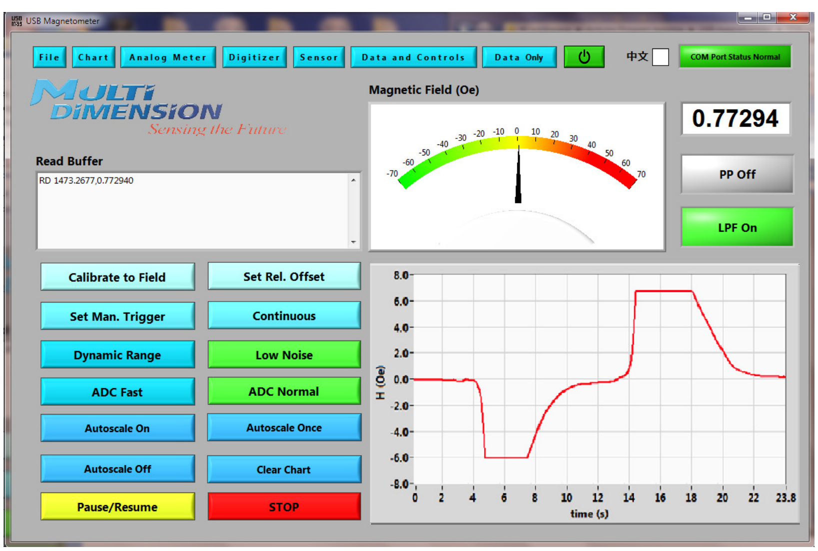
Configuring ADC Range and Resolution

The USB Magnetometer can be configured to measure different field ranges, measurement speeds, resolutions, and noise levels. This is accomplished by changing the gain of the preamplifier in the ADC chip, the number of bits used by the ADC, and the number of samples per data point.

The measurement range may be configured using the “Dynamic Range” and “Low Noise” buttons. When “Dynamic Range” is selected, the preamplifier gain in front of the ADC is set to 1, and the ADC can measure a voltage swing from about 0 to V_{ref}, which is the applied bias voltage of the sensor. The plot below shows a linearized TMR 2705, with “Dynamic Range” selected, as a large magnet is moved towards the sensor, moved back, flipped, moved towards the sensor, and again moved back.

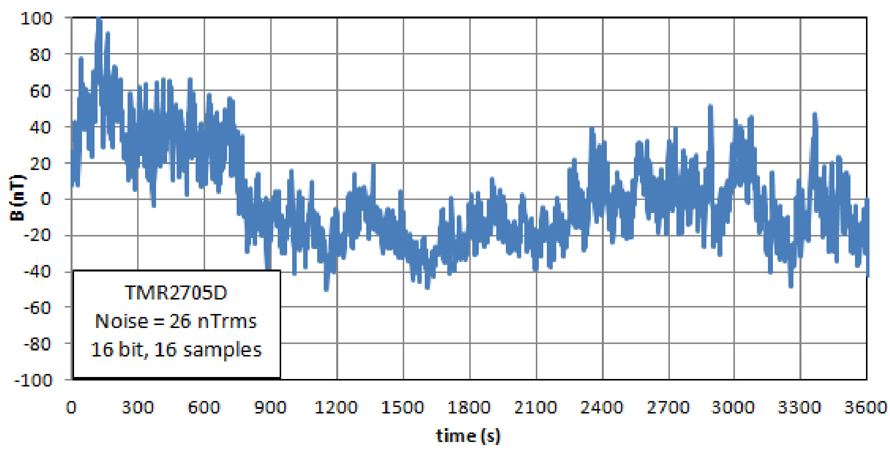


When set to low noise, the internal preamplifier gain is set to 8, and the ADC will saturate at a value no more than 1/8 the applied sensor bias voltage. The following plot shows the identical measurement with “Low Noise” selected.

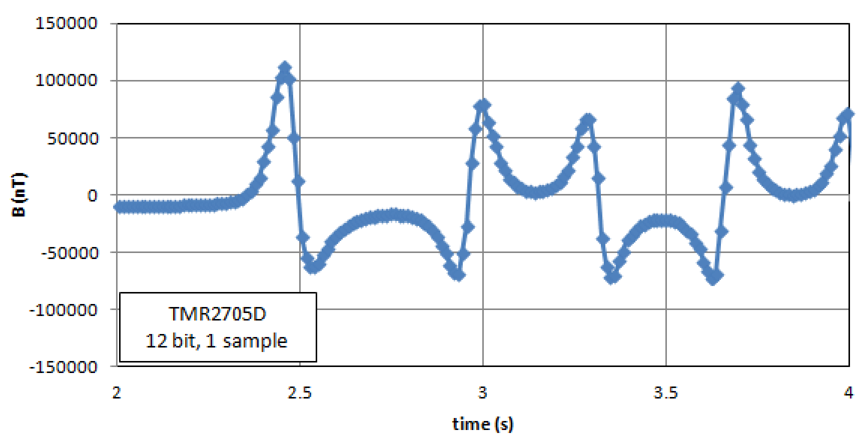


“ADC Fast” and “ADC Normal” may be used to configure speed and resolution. “ADC Fast” corresponds to a 100 Hz sample rate with 12 bit resolution. “ADC Normal” corresponds to a sample rate of about 8 Hz with 16 bit resolution.

Highest resolution and lowest noise is achieved by selecting “ADC Normal” and “Low Noise” settings. Noise can be further reduced and resolution further increased by averaging, and this can be accomplished using the “Digitizer|Number of Samples” menu entry. The Number of samples can be made arbitrarily large, at the expense of measurement speed. The plot below shows an hour long measurement at 16-bit resolution with 16 samples per point. The RMS noise in this configuration is <30 nTrms.



Faster measurements may be accomplished using the “ADC Fast” setting and reducing the number of data points averaged to 1. The next plot shows a magnet being rapidly and repeatedly moved past a sensor in high speed mode without averaging.



Magnetometer Linearization

The USB Magnetometer uses a default linearization algorithm to extend the linear range of the TMR sensor. The linearization is performed prior to sending data over the USB port. The default linearization should be sufficient for most users. The parameters may however be changed by the user using the commands defined in the USB Port Programming section.

The linearized sensor output field is defined in a dimensionless format as

$$H_{corr}\left(\frac{Sensitivity}{1000}\right) = \sum_{i=0}^{5} A_i \left(\frac{V_{raw}}{V_{ref}}\right)^i$$

Where V_{raw} is the raw voltage from the sensor, V_{ref} is the bias voltage of the sensor, and V_{corr} is the linearized sensor voltage. A_i is a coefficient used for linearization. Typically for a TMR sensor, $A_5>A_3>A_1$ and $A_0=A_2=A_4=0$. A polynomial fit of the raw voltage response to this polynomial equation can provide the Ai coefficients for linearization.

Please contact jim.deak@dowaytech.com for details.

Exporting Data

The program can be made to export all data presently displayed in the H (Oe) vs. time (s) chart, by selecting the "File|Export Chart Data" menu entry. The number of data points to be stored may be controlled by changing the "Data Points" control at the right side of the "Field (Oe)" analog meter. The data chart may be cleared by pressing the "Clear Chart" button at the lower right side of the "Field (Oe)" analog meter. Data will be exported into CSV format, and the user may define the name and location of the data file. The data format is standard comma separated variable, and it can be accessed in most spreadsheet programs.

Measurement Examples

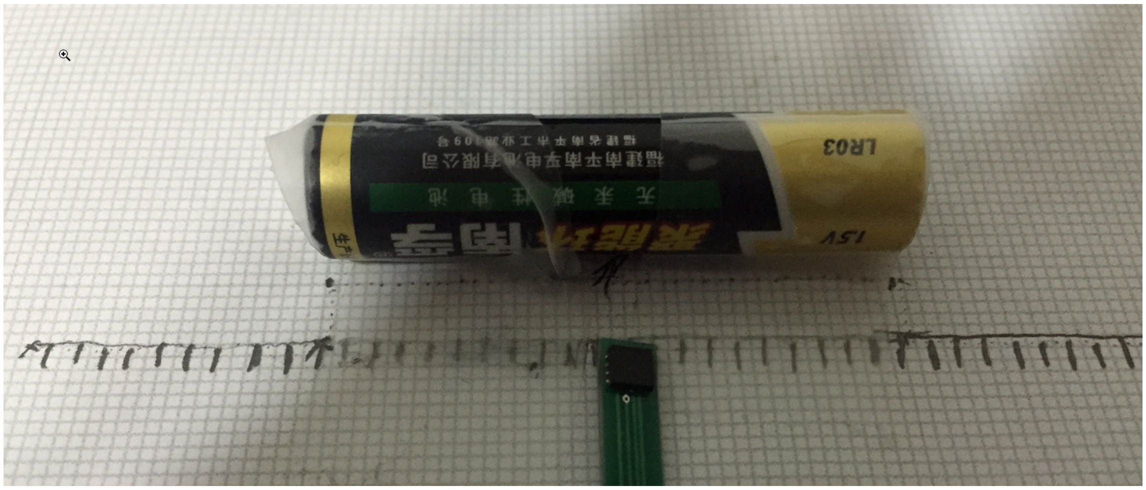
The MDT USB Magnetometer is well suited for precision measurement of magnetic fields that occur in common engineering and scientific applications. Here are a few example measurements obtained using the MDT USB Magnetometer GUI and data export feature.

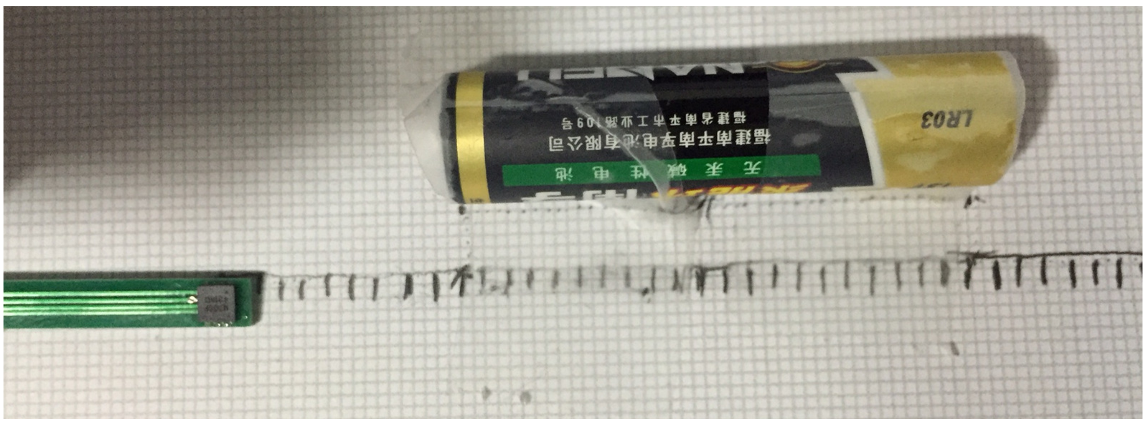
1. Field distribution around an AAA battery

Often it is desirable to deploy magnetic sensors in applications where the sensor must be powered by a battery in close proximity to the sensor. An example of this type of device would be a traffic or parking lot monitoring system. Unfortunately, typical batteries often are composed of ferrous materials that have a significant remanent magnetization. In effect, they behave like a weak permanent magnet. To further complicate the system design, magnetic sensor systems that are used for detecting the presence of automobiles often utilize sensors that saturate at a few Oe, which may be smaller than the remanent magnetic field of the battery. This can be corrected by a hard magnetization offset value added to the sensor output, but only if the sensor is not saturated by the magnetic field. Also, the batteries cannot be shielded, since the magnetic shielding would probably also shield the magnetic field that the magnetic sensor needs to detect. So a practical question that the MDT USB magnetometer can answer is, where should the sensor be located with respect to the battery?

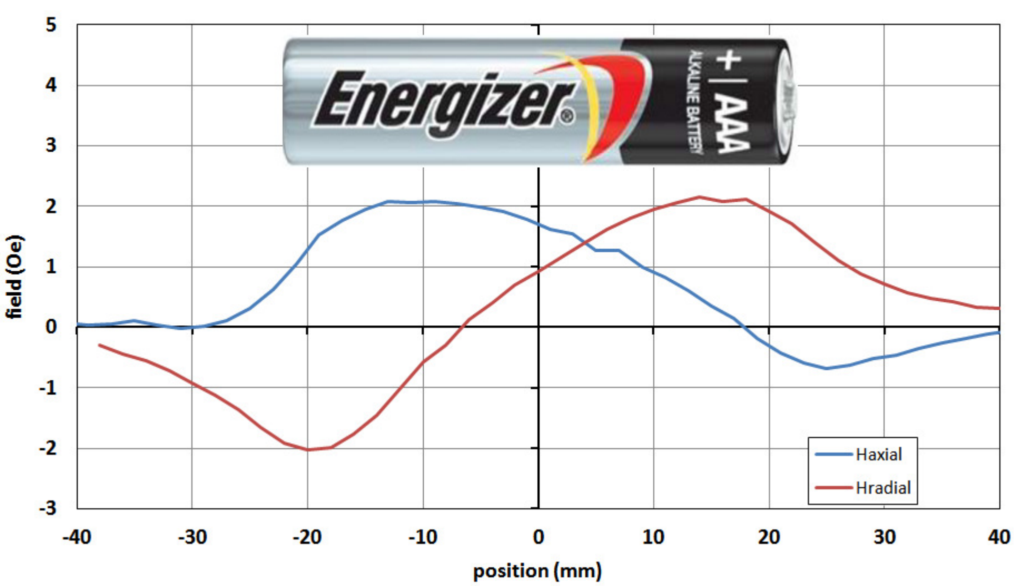
Here we look at the magnetic field along a line that is 5 mm from the edge of an AAA battery. The USB Magnetometer is set for 17 bit resolution, no LPF, 4X speed, manual trigger. Relative offset is used to zero the sensor at properly oriented positions far removed from the battery. The measurement is performed by taping an AAA battery to a piece of graph paper and manually triggering the sensor at specific positions along a line 5 mm distant from the edge of the battery.

Here are typical results.



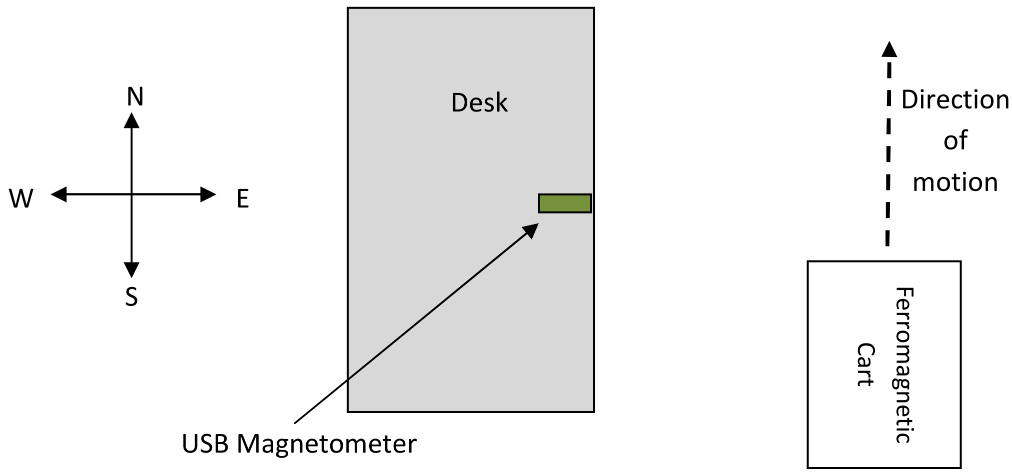


The measured fields show that even at 5 mm from the battery, a remanent magnetic field exceeding 2 Oe can be present. This is strong enough that it could saturate high sensitivity magnetic field sensors, making hard magnetic correction impossible. Note also the magnetic field is not uniform.

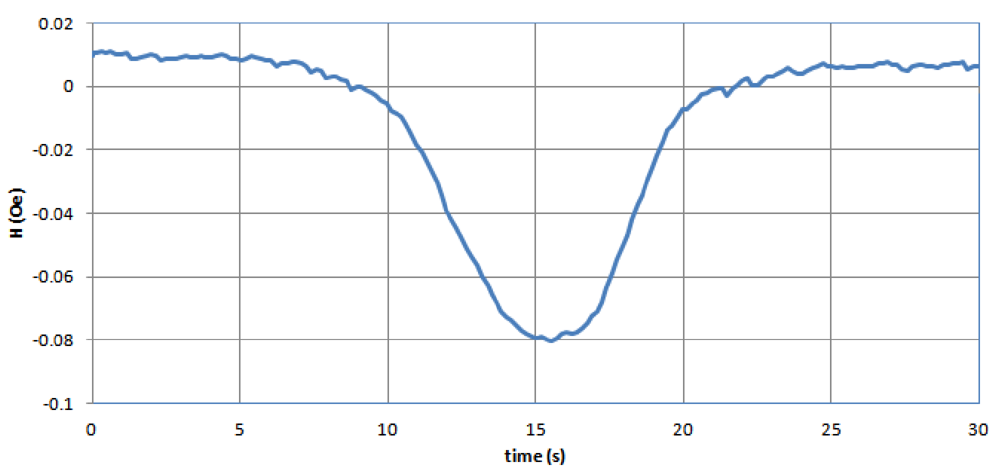


2. A ferrous object moving past a stationary magnetometer

This next chart shows the magnetic field distortion produced by a metal cart moving past a desk.



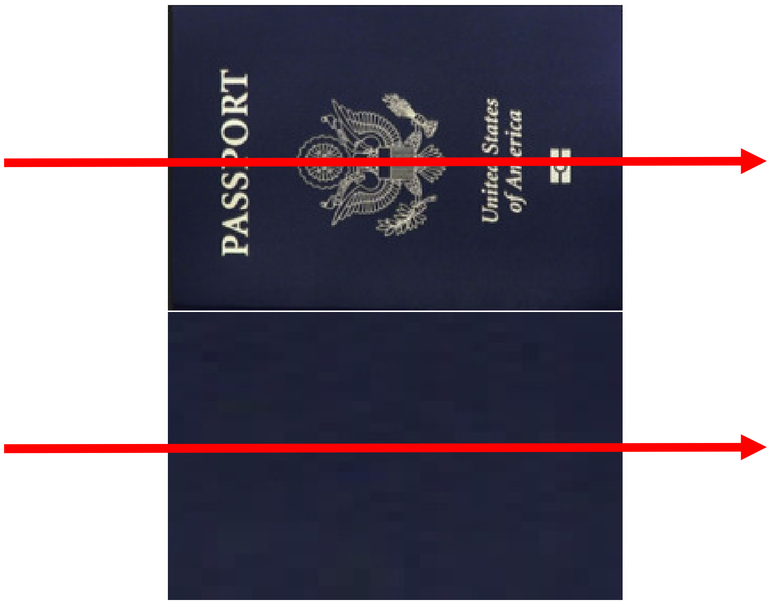
In this case, the magnetometer was placed flat on a desk, aligned mostly along the east-west direction. A large metal cart was pushed by at a distance of about 1 m along the north-south axis. Data acquisition used 16 bit resolution with LPF turned on. The motion of the cart past the desk slightly bends the Earth's magnetic field into the East-West axis and produces a small signal.

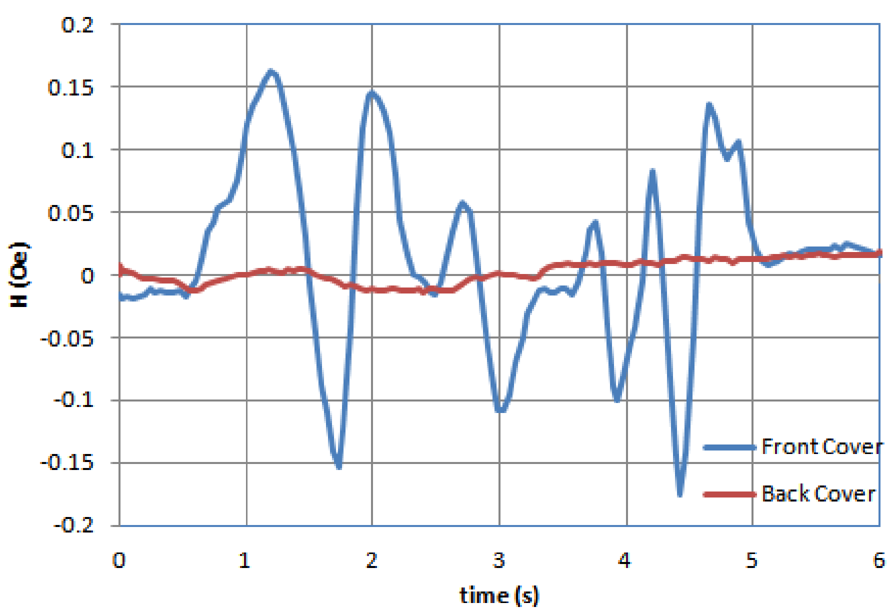


Note the change in magnetic field at the 15 s mark as the cart was moved past. Detecting the presence of an object by looking for the distortions it produces in a magnetic field is a common tracking technique known as magnetic anomaly detection.

3. Hidden magnetic signature in a document

Here the magnetic field across the surface of a United States Passport cover is examined. The passport is held stationary, and the sensor is slid over the front and back cover of a US passport. It takes some practice to keep the magnetometer alignment stable if doing this manually, but it is not too difficult to master. What conclusion can be made?



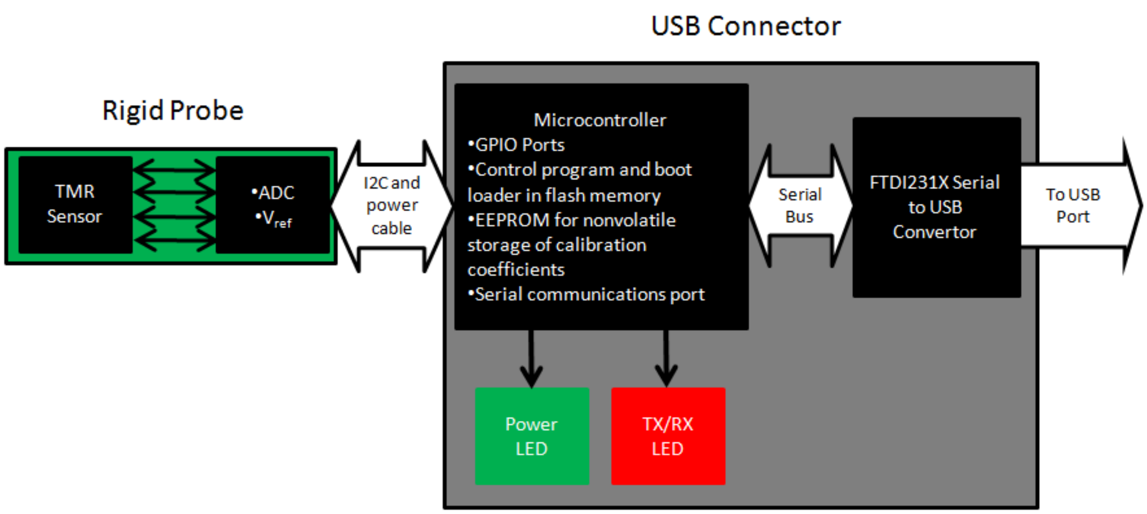


USB Port Programming

The USB Magnetometer was designed to enable users to interface it with their own custom programs. This is useful for writing data acquisition programs, so the user can input data directly to a program without needed to export it from the USB Magnetometer GUI into Excel.

The USB magnetometer may be interfaced with any software language that is able to connect to a COM port and send and receive ASCII commands, including C/C++, Visual Basic, Java, Labview, etc. Additionally, the USB Magnetometer may be operated directly from a serial terminal such as Termite, HyperTerminal, RealTerm, etc, in order to test a sequence of commands or for actual use. A free copy of Termite can be found at http://www.compuphase.com/software_termite.htm.

The USB magnetometer hardware is defined as follows:



A TMR or other sensor at the end of a 1 m cable is used to detect the magnetic field, and it converts the magnetic field to a voltage value. The TMR sensor is biased by a temperature compensated voltage reference. The TMR sensor output is connected to an ADC, which digitally samples the voltage and sends the digital signals to the microcontroller over an I2C bus. The microcontroller converts the sampled sensor voltage to a magnetic field value using calibration coefficients stored within the non-volatile program Flash memory or the user reprogrammable EEPROM. When factory calibrated, the calibration coefficients are stored in EEPROM. These EEPROM values may be overwritten by the user. The microcontroller then sends the magnetic field value to the serial bus, which is connected to the FTDI231X Serial to USB convertor. The I/O of the FTDI231X Serial to USB convertor is a standard USB port. A driver on the computer is used to create a virtual COM port, which is then accessed through the MDT USB magnetometer GUI, a terminal emulator program, or other custom written program.

The startup sequence after reset or connecting the USB Magnetometer to a USB port is as follows:

1. Check the EEPROM to see if the EEPROM calibration is valid, if so load offset and sensitivity parameters to overwrite the default values in firmware
2. Check the EEPROM to see if the linearization coefficients are valid, if so, retrieve them from EEPROM and overwrite the default values in firmware
3. Send out welcome message
4. Report system status
5. Poll for USB port input

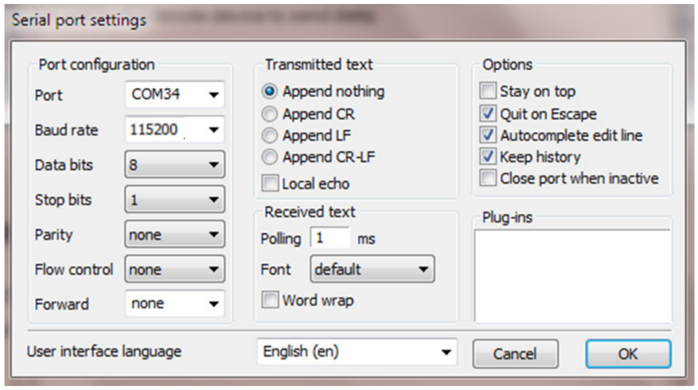
The following pages list the commands used for configuring and reading the USB magnetometer.

USB Magnetometer Serial Port Commands		
Command	Description	Firmware Subroutine
OA	Automatically zero the voltage offset of the sensor. Overrides OM. This should be accomplished in a zero gauss chamber	CorrectOffset()
SM <n.nnn>	Manually input a sensitivity value in units of mV/V/Oe	ManualSensitivity()
OM <n.nnn>	Manual input an offset voltage, in units of volts. overrides OA	ManualOffset()
RS	Automatically find the offset field. This command remembers the previous offset field. Calling RO will remove the offset value and restore the state prior to calling RS.	RelativeSet()
RO	Removes the offset field value calculated during the RS command. It restores the state prior to calling RS.	RelativeOff()
PC	Display the calibration parameters presently being used by the magnetometer.	PrintCal()
DB <N>	Set oversampling in order to achieve a desired number of bits (N) or resolution for the ADC. N is any integer greater than 10, and for practical measurement it should not exceed 18, due to the low effective sampling rate at high oversampling. Increasing N decreases noise.	SetDigitizerBits()
DS <N>	Set the ADC clock speed. Where N is a multiplier from 1 to 4. Increasing the clock speed decreases the digitizer integration period, and thereby increases sensor noise somewhat. Also, better linearity may be obtained at lower clock speeds. That said, the default is set to fast (N=4) and performance is quite good. When the clock speed is changed, the sensor should be recalibrated, due to a change in the calibration of the ADC.	SetDigitizerSpeed()
RM	Set the magnetometer to one-shot trigger mode. Successive calls to RM will result in a single read. This is useful for situations where a user may want to slowly move the probe past a series of fixed locations around a magnetic object, or in the case where the ADC resolution is very high and read time is very long. Overrides the RC and VC commands.	ManualRead()
VM	Same as RM, except the magnetometer will output the uncorrected voltage of the sensor. Overrides the RC and VC commands.	ManualReadVoltage()
RC	Set the magnetometer to continuous read mode. Overrides the RM and VM commands.	ContinuousRead()
VC	Same as the RC command, except the output will be the uncorrected voltage of the sensor. Overrides the RM and VM commands.	ContinuousReadVoltage()
NS <N>	Set the number of data points (N) to be averaged.	SetNumberSamples()
LP <0 or 1>	Turn the low pass filter on (1) or off (0);	SetFilterMode()

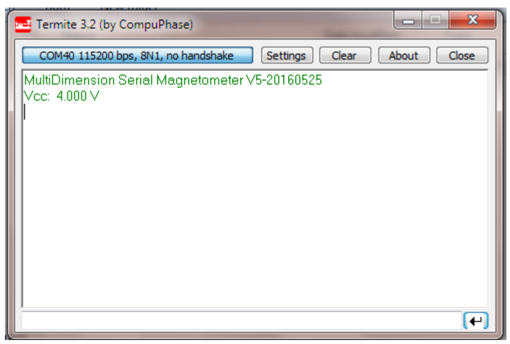
Command	Description	Firmware Subroutine
SS <string>	Write the sensor serial number and other identifying information to the EEPROM. The maximum <string> length is 32 ASCII characters.	SetSensorCodeEEPROM()
GS	Query the sensor ID code.	GetSensorCodeEEPROM()
LS	Enable linearization of the sensor data (default)	SetLinearization()
LO	Disable sensor linearization	RemoveLinearization()
PE	Pause execution, but do not change the system settings. Pause the state of the magnetometer.	Pause()
RE	Resume execution without changing system settings.	Resume()
HI	Used to test for the presence of a USB magnetometer at a particular port number. The valid response from the USB magnetometer is "Hello".	QueryProbe()
OW <n.nnn>	Directly write a user specified offset value in units of volts to the USB magnetometer EEPROM.	WriteOffsetEEPROM()
OR	Query the offset value stored in the EEPROM.	ReadOffsetEEPROM()
SW <n.nnn>	Directly write a user specified sensitivity value in units of mV/V/Oe to the USB magnetometer EEPROM.	WriteSensitivityEEPROM()
SR	Query the sensitivity value stored in the EEPROM.	ReadSensitivityEEPROM()
AR	Read linearization coefficients stored in the EEPROM	ReadLinEEPROM()
AM <m> <n.nnn>	Set present linearization coefficient Am to value n.nnn. Must follow with EA to store in EEPROM.	AManual()
CV <0 or 1>	Set the calibration valid flag in the EEPROM. Once set, the USB magnetometer will read the calibration parameters stored in the EEPROM at startup, rather than using the default value stored in firmware. 0 = not valid, and 1 = valid (default).	CalValid()
LV <0 or 1>	Set the linearization valid flag in the EEPROM. Once set, the USB magnetometer will read the calibration parameters stored in the EEPROM, rather than using the default value stored in microcode. 0 = not valid (default), and 1 = valid.	LinValid()
ES	Write the sensitivity value presently being used to the EEPROM. For example, the probe has been calibrated using the SM command, that sensitivity value will not be stored in EEPROM by default.	WrtCalSlopeEEPROM()
EO	Write the offset value presently being used to the EEPROM. For example, the probe has been calibrated using the OM command, that sensitivity value will not be stored in EEPROM by default.	WrtCalOffsetEEPROM()
EA	Write present linearization parameters to the EEPROM.	WriteLinEEPROM()
AB <0 or 1>	Set output to binary (1) or default ASCII (0)	SetASCIIBinary()
TS <0 or 1>	Turn timestamp on (1) default or off (0)	SetTimeStamp()
PG <1,2,4,8>	Set the preamplifier gain	SetPGA()
QQ	Soft reset of the USB Magnetometer probe.	psuedoReset()

Serial Port Operation Using a RS-232 Terminal Emulator

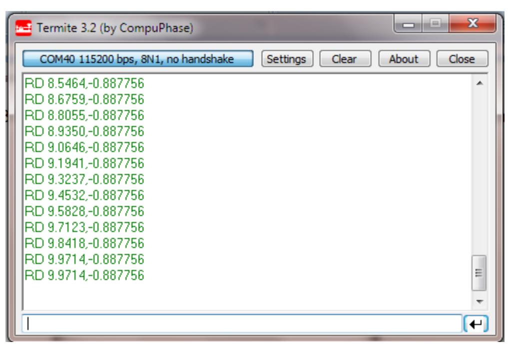
The MDT USB Magnetometer may be operated directly from a terminal emulator program. The terminal emulator programs should be configured as in the following figure, except that the COM port number might be different. The examples below use Termite.



Upon connecting, the USB Magnetometer should reset and display a message similar to the following.

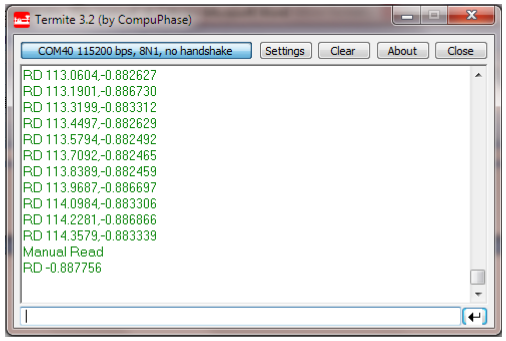


To stream the measured field value in units of Oe, simply send an "RC" command.

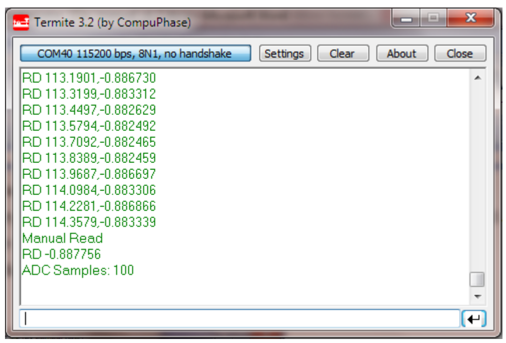


The two numbers after the RD token are time in units of seconds, and magnetic field in units of Oe. When swinging the magnetometer probe around in space, if there are no distortions of Earth's field, the maximum and minimum field values should be approximately ±0.5 Oe.

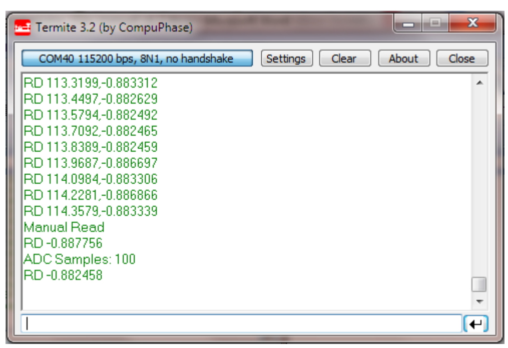
To stop continuously reading the magnetometer, issue a "RM" command. This initiates manual read or "one shot" mode.



Successive calls to the RM command will provide a single read. This is useful for precision measurements of a field at a stationary point, for example, if measuring the field distribution around a magnetic object. The precision of the measurement may be improved by increasing the number of samples or increasing the ADC resolution. For example, the samples could be increased to 100 by issuing an “NS 100” command.



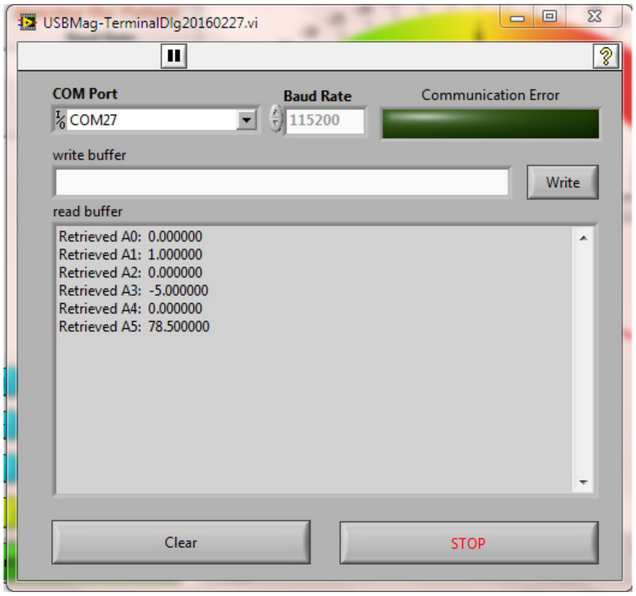
Then after issuing a “RM” command, 100 samples will be averaged, and the resulting precision will be higher. It will however take longer to acquire the data point.



Depending on the terminal emulator program that is being used, data can be exported to a file or cut and pasted into a spreadsheet. Most of the functions available in the GUI can be performed in text mode using a terminal emulator program. The USB magnetometer can be run in this manner on any computer of portable device that has standard COM port drivers.

Command Line Window

The GUI has a built in terminal emulator that may be accessed from the digitizer menu.



Here the “write buffer” input is used to input a serial port command. The command will be sent after pressing the “Write” button. The read buffer shows the real time output of the magnetometer. The “Clear” button is used to clear the “read buffer” indicator. The Stop button is used to exit the dialog.

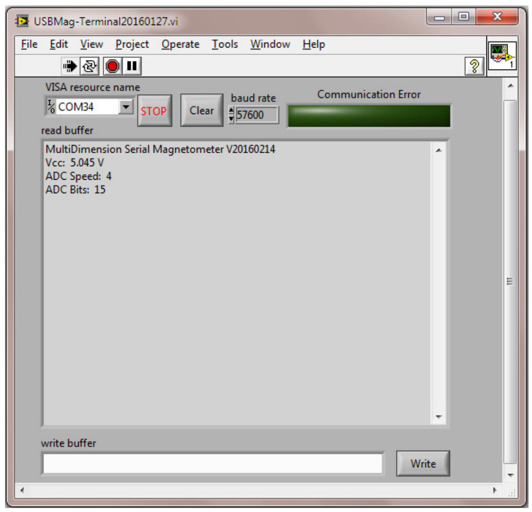
The command line window is useful for testing serial port commands and configuring the USB Magnetometer.

Be warned, that it is possible to make permanent changes to the EEPROM from this window that could affect probe operation. Any other changes may be reversed by quitting the GUI program, removing and reinserting the USB Magnetometer from the USB port, and then restarting the GUI program.

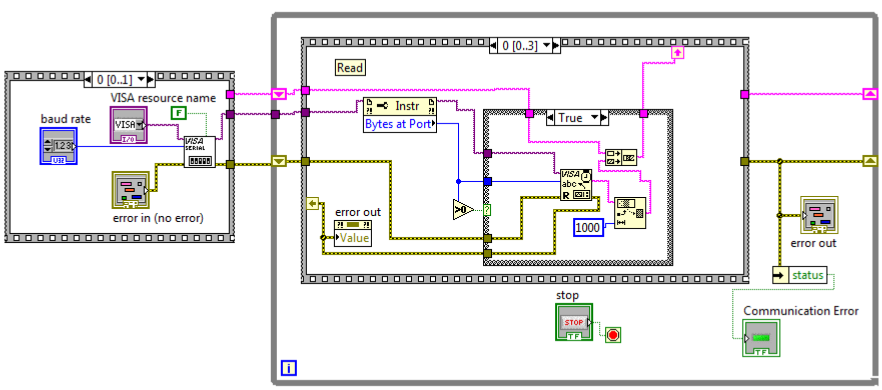
Custom Software Development

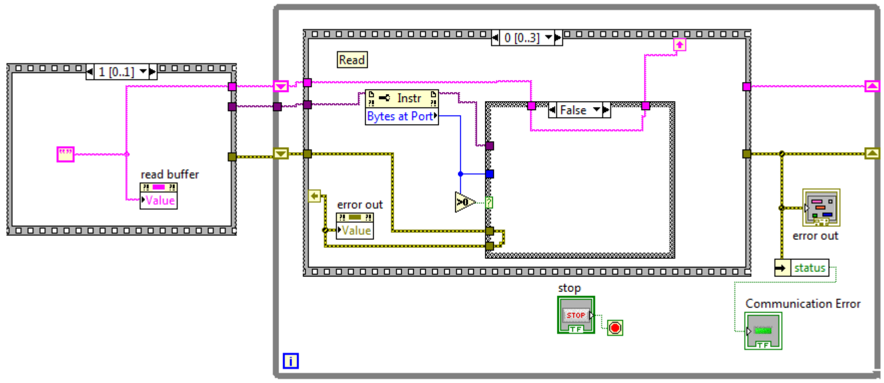
Any programming language that can access a COM port in order to send and receive ASCII data may be used to write a custom interface for the USB Magnetometer. The user may explore the operation of the various commands by using a terminal emulator. Likewise, as long as the USB port can be accessed, and the proper cable is used, programs may be written for mobile devices and tablets. An example terminal emulator program written in Labview and compatible with the standard firmware is shown below.

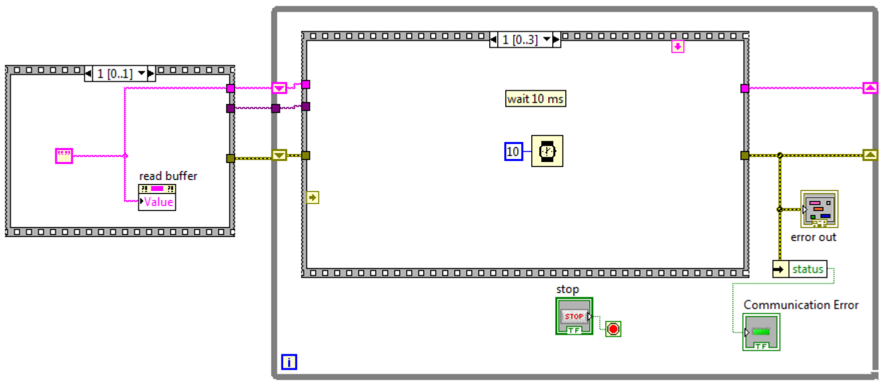
Serial Port Emulator User Interface

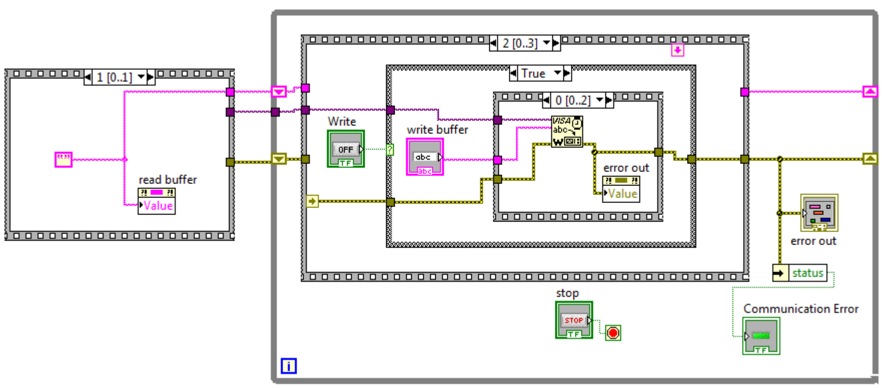


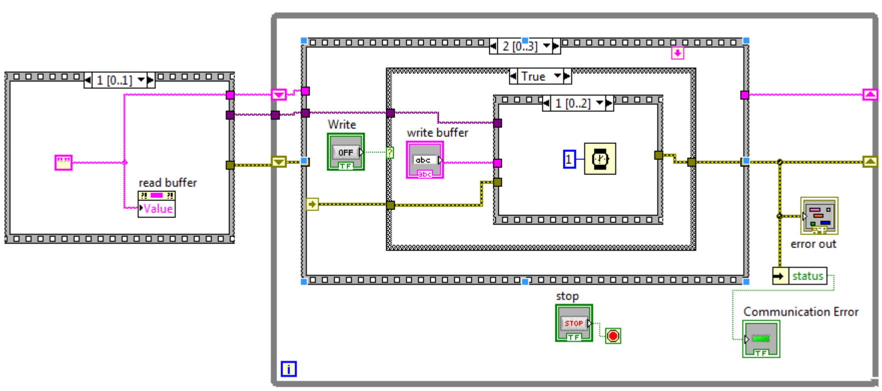
Block Diagram

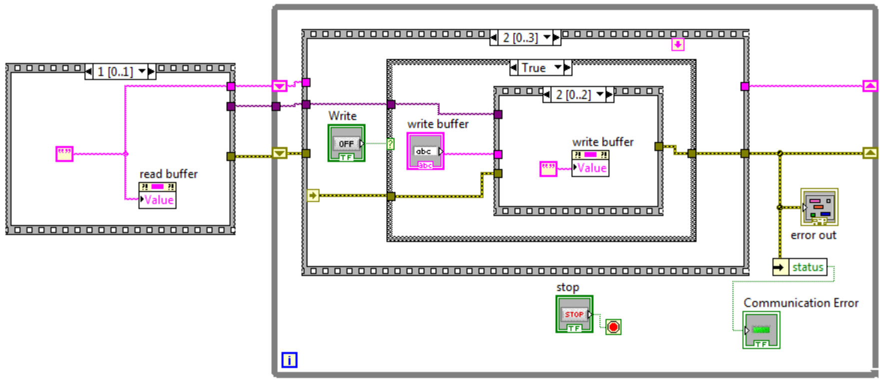


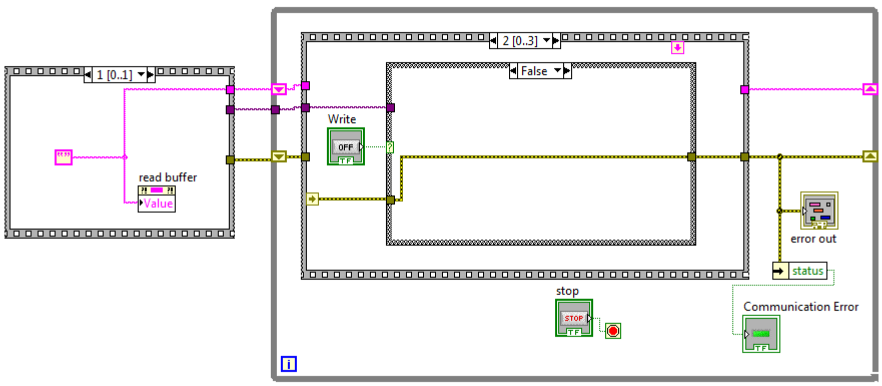

1 [0..1]
read buffer
Value
0 [0..3]
Read
Instr
Bytes at Port
False
error out
Value
error out
status
stop
Communication Error

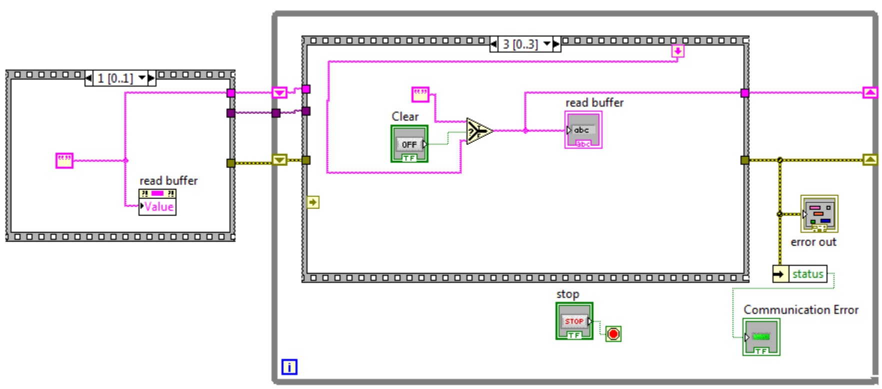

1 [0..1]
read buffer
Value
1 [0..3]
wait 10 ms
10
error out
status
stop
Communication Error


1 [0..1]
read buffer
Value
2 [0..3]
True
0 [0..2]
Write
write buffer
error out
Value
error out
status
stop
Communication Error


1 [0..1]
read buffer
Value
2 [0..3]
True
1 [0..2]
Write
write buffer
1
error out
status
stop
Communication Error







Custom Firmware Development Options

The MDT USB Magnetometer is designed so that custom firmware can be easily uploaded to the microcontroller flash memory through the USB port. This is useful for exploring algorithms for and applications of MDT TMR sensors. In order to make this easy for the user, the USB magnetometer is program compatible with the popular open source Arduino UNO board, and thus the Arduino programming environment or Atmel microcontroller development tools. The user may develop custom firmware at his/her own risk.

The following is the basic information that the user needs to know:

- The microcontroller is an Atmel ATMEGA328P with Arduino boot loader, and it may be programmed using the open source Arduino Development Suite with UNO board settings.
- The ADC is a Texas instruments ADS1100
- The reference voltage is fixed 4.0 V
- The MDT TMR sensor outputs are connected to analog inputs of the ADC
- The power LED is connected to USB $V_{cc.}$
- The Tx/Rx LED is connected to the ATMEGA328 Tx/Rx line
- The Serial to USB convertor is a standard FTDI231X chip. No special drivers are needed.

Details of the Arduino C programming language and development environment may be found at https://www.arduino.cc/en/Reference/HomePage.

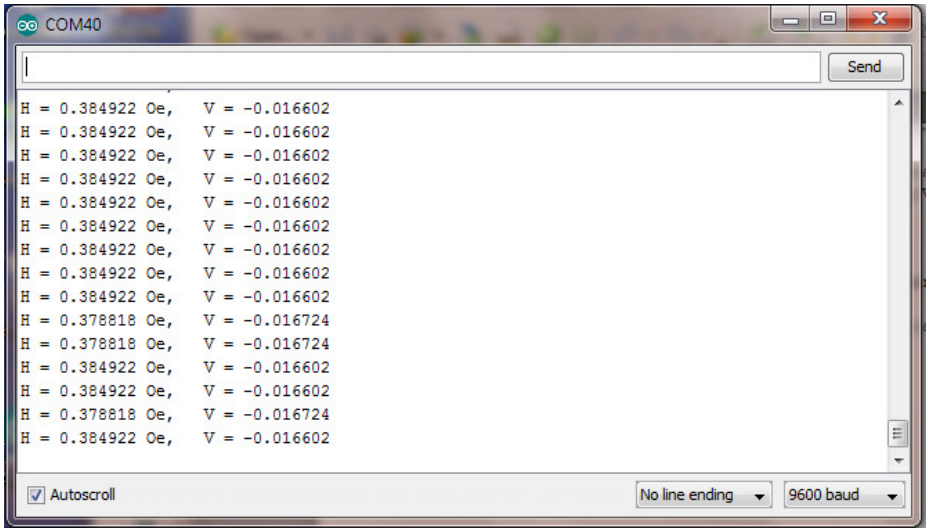
A simple Arduino C program for streaming sensor output to the USB port is provided here:

```
/*****************************************************************************************
 * Read TMR sensor voltage, convert to Oe, and stream to USB port
 *****************************************************************************************/

#include <Wire.h>        // used for I2C bus
#define ADC B1001010     // 7bit I2C address of ADC

// configure sensor
float sensitivity = 5.0;     // sensitivity calibration in units of mV/V/Oe --> needs to be changed
float offset = -0.011475;    // offset calibration in units of volts --> needs to be changed

// configure measurement
int NumSamps = 1;        // number of samples to be averaged
int gain = 1;            // valid options are 1,2,4,8
int bits = 16;           // valid options are 12, 14, 15, and 16

// do not change
float SensorVoltage;     // output voltage of the sensor
float Vref = 4.0;        // Sensor Reference voltage
float Hout;              // measured field

void ConfigureADC(int bits, int PGA, int Continuous, int RequestData) {
  /**************************************************************************
   * Routine used to configure the ADC
   **************************************************************************/

  unsigned int bCom = 0x00;

  // set the number of bits and datarate
  switch (bits) {
    case 12: bCom = bCom | 0x00; break; // 12 bits, 128 bps
    case 14: bCom = bCom | B00000100; break; // 14 bits, 32 bps
    case 15: bCom = bCom | B00001000; break; // 15 bits, 16 bps
    case 16: bCom = bCom | B00001100; break; // 16 bits, 8 bps
    default:  bCom = bCom | B00001100;       // 16 bits, 8 bps
  }
```

```
  // set the PGA gain
  switch (PGA) {
    case 1: bCom = bCom | B00000000; break; // gain = 1
    case 2: bCom = bCom | B00000001; break; // gain = 2
    case 4: bCom = bCom | B00000010; break; // gain = 4
    case 8: bCom = bCom | B00000011; break; // gain = 8
    default:  bCom = bCom | B00000000;      // gain = 1
  }

  // set continuous/single mode
  switch (Continuous) {
    case 0: bCom = bCom | B00010000; break; // single
    case 1: bCom = bCom | B00000000; break; // continuous
  }

  // this is used in single conversion mode to request a read
  switch (RequestData) {
    case 0: bCom = bCom | B00000000; break; // single
    case 1: bCom = bCom | B10000000; break; // continuous
  }

  // write the command
  Wire.beginTransmission(ADC);
    Wire.write(bCom);
  Wire.endTransmission();
}


float ReadSensor(int Nbits) {
  /*****************************************************
   * Routine used to read the ADC.
   * bits must be > 11, <17, and <> 13
   *****************************************************/

  float Vout;
  byte highbyte, lowbyte, configR;

  Wire.requestFrom(ADC, 3);
  while(Wire.available()) // ensure all the data comes in
  {
    highbyte = Wire.read(); // high byte
    lowbyte = Wire.read(); // low byte
    configR = Wire.read();
  }

  // divide by the PGA gain and scale based on ADC bits
  Vout = (highbyte * 256 + lowbyte)/gain * scale(Nbits);
 delay(SetDelay(bits)); // dependent on ADC bits
 return Vout;
}
```

```
float SetDelay(int Nbits) {
  // used for ads1100, see datasheet for details
  float ftmp;
  switch (Nbits) {
    case 12: ftmp = 7.0; break;
    case 14: ftmp = 31.0; break;
    case 15: ftmp = 63.0; break;
    case 16: ftmp = 125.0; break;
    default: ftmp = 125.0; break;
  }
  return ftmp;
}

float scale(int Nbits) {
  // conversion used for digital resolution
  float ftmp;
  switch (Nbits) {
    case 12: ftmp = 1.0/2048.0; break;
    case 14: ftmp = 1.0/8192.0; break;
    case 15: ftmp = 1.0/16384.0; break;
    case 16: ftmp = 1.0/32768.0; break;
    default: ftmp = 1.0/32768.0; break;
  }
  return ftmp;
}

void setup() {
  Serial.begin(9600); // initialize serial communications at 9600 bps:
  Wire.begin();       // open communication with ADC
  delay(10);
  ConfigureADC(bits, gain , 1, 0); // ADC @ 16-bits, 8bps, continuous conversion
}


void loop() {
  int i;
  float tmp = 0.0;

  // read and average the analog voltage of the sensor
  for (i=0; i<NumSamps; i++) tmp += (float)ReadSensor(bits);
  SensorVoltage = Vref*tmp/(float)NumSamps;                    // Sensor voltage
  Hout = (SensorVoltage-offset)*1000.0/sensitivity/Vref;     // Convert voltage to field

  // print the results to the serial monitor
  Serial.print("H = "); Serial.print(Hout, 6);
  Serial.print(" Oe,    V = "); Serial.println(SensorVoltage, 6);

  delay(2); // optional delay to slow down the serial output
}
```

Output from this program should appear as follows:



The user will need to set the appropriate sensitivity and offset values, but this program can provide the starting point for custom firmware development. This program only consumes 20% of the program flash memory, leaving ample room for the user to experiment with various algorithms. Within the USB connector, the microcontroller PCB also has several GPIO and analog inputs broken out to connection points. If necessary and a user does not mind disassembling the USB connector, a user could tap into them to add other input and output signals.

Provided the user does not brick the microcontroller, the MDT USB Magnetometer may be restored to original operation and compatibility with the MDT measurement GUI by uploading a compiled version of the standard firmware, which MDT may make available. Note, uploading custom firmware does not change the calibration coefficients stored in EEPROM. If the calibration coefficients were overwritten by the custom firmware, then the user can simply recalibrate the sensor using the standard GUI calibration routine, and then issuing a “CV 1” command using a terminal emulator program.

For more details, please contact jim.deak@dowaytech.com, to discuss different custom development options. MDT has no plans to release the complete hardware and code specification, but in some special cases MDT may choose to do so. MDT may also modify firmware in some special circumstances at the request of a user. The USB hardware is compatible with most MDT sensors, and MDT may consider custom builds. MDT may from time to time release firmware upgrades that the user can flash to the USB Magnetometer using the Arduino development environment or other compatible software tool.