Introduction

MagnoMonitor:

The MagnoMonitor program is designed to monitor and analyze magnetic fields. It allows the user to connect to a transmitter device that measures magnetic fields in three orthogonal axes, x, y, and z. The program continuously receives data from the transmitter over the RS-232 serial interface and provides real-time monitoring and processing. The program has a main panel with associated controls and an interactive interface with tabs to switch between views: live chart, 3D graph, and Fourier transform. The program has a main panel with associated controls, and an interactive interface with tabs to switch between views: live chart, 3D graph, and Fourier transform.

- The user configures the communication port according to the operating standards of the measuring device using a dedicated control button. An LED indicates the connection status.

- The measured values for the x, y, and z components of the magnetic field are processed and displayed in numeric controls on the user interface.

- This program calculates the total strength (intensity) of the magnetic field by using the Euclidean norm of the x, y, z vector. The strip chart displays the intensity, while numeric controls show the maximum and minimum magnitudes calculated.

- A strip chart is used to visualize the real-time changes in the magnitude of the magnetic field over time. The chart scrolls horizontally, displaying the data points as new data arrives.

- The distance between points on the x-axis (time axis) is determined by the sampling rate of the measuring device and the time window for the strip chart.

- The program gives the option to log the received x, y, and z data for further analysis or documentation. The data is written as a text (*.txt) file to the project directory with the default name, or a different location and name can be selected.

- The program features a 3D graph – Measurement's Studio ActiveX control, which showcases the variations in magnetic field values throughout time. The x, y, and z components are represented in sets on the three axes of the graph, while the exact values are shown in a table.

- After the data acquisition is complete, the program performs a Fourier transform of the recorded magnetic field data using the FFT algorithm. Then the spectrum in the frequency domain is displayed on a graph. The Fourier transform helps identify the frequency components present in the magnetic field signal, which can be valuable for analyzing harmonic interference or detecting specific patterns.

Transmitter:

The Transmitter is a simulator, a companion program that emulates the behavior of an actual measuring instrument. It loads data representing magnetic field measurements, simulates the data probing, and transmits it via the RS-232 serial interface. The Transmitter reads pre-recorded magnetic field data from a file. The data represents measurements in the x, y, and z directions, corresponding to different points in time.

- The Transmitter reads pre-recorded magnetic field data from a file. The data represent measurements in the x, y, and z directions, corresponding to different points in time.

- The program simulates real-time probing of the magnetic field in three directions at a fixed sampling rate and transmits it, similar to how the actual instrument transmits data.

- The program reads the data in a floating-point format. It is then converted into bytes (char type) and arranged in a specific order to create a packet of consecutive bytes. The packet is sent with proper checksum to guarantee the integrity of the data during transmission.

- The Transmitter includes a real-time LED indicator. The LED blinks to indicate that data is being sent. This visual feedback allows users to know when the transmission is active.

RS-232 Configurator:

The RS-232 Configurator is a shared library (DLL) that contains functions and a UIR panel to configure and manage the RS-232 communication settings. The DLL provides a convenient way to configure the RS-232 communication settings. The UIR panel allows configuring the parameters, such as the COM port, baud rate, parity, data bits, and stop bits. It allows choosing the handshaking operation by selecting CTS or X mode.

- This program has an error-handling mechanism related to RS-232 communication. It retrieves the error message associated with the most recent RS-232 error and displays it in a popup window.

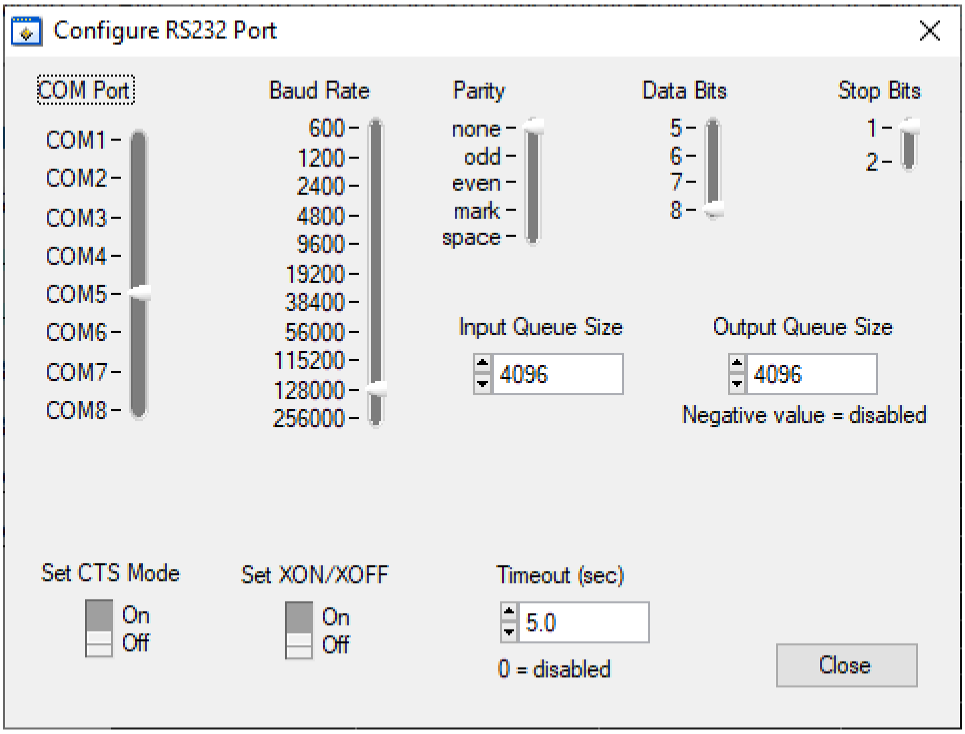
- The MagnoMonitor program allows users to interface with a real transmitter device to study and interpret magnetic field data for various applications, such as scientific research, engineering, and environmental monitoring.

- The Transmitter allows testing the MagnoMonitor program's functionality without the need for an actual measuring instrument. By using pre-recorded data, the simulator generates a representative output that enables users to evaluate the Analyzer program's performance and features in a controlled environment.

- The RS-232 Configurator provides a convenient way for users to configure the RS-232 communication settings for both the MagnoMonitor and the Transmitter programs. It encapsulates the low-level communication details, reducing code duplication and improving maintainability. By using a DLL, common functionality is shared between the programs.

Graphical interface

RS-232 Configurator:


Configure RS232 Port

COM Port: COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8

Baud Rate: 600, 1200, 2400, 4800, 9600, 19200, 38400, 56000, 115200, 128000, 256000

Parity: none, odd, even, mark, space

Data Bits: 5, 6, 7, 8

Stop Bits: 1, 2

Input Queue Size: 4096

Output Queue Size: 4096

Negative value = disabled

Set CTS Mode: On / Off

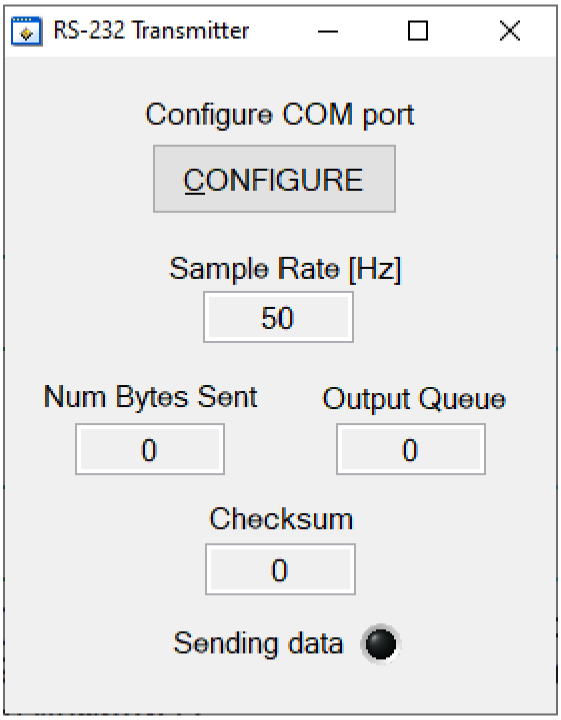
Set XON/XOFF: On / Off

Timeout (sec): 5.0

0 = disabled

Transmitter:

Idle


RS-232 Transmitter

Configure COM port

Sample Rate [Hz]: 50

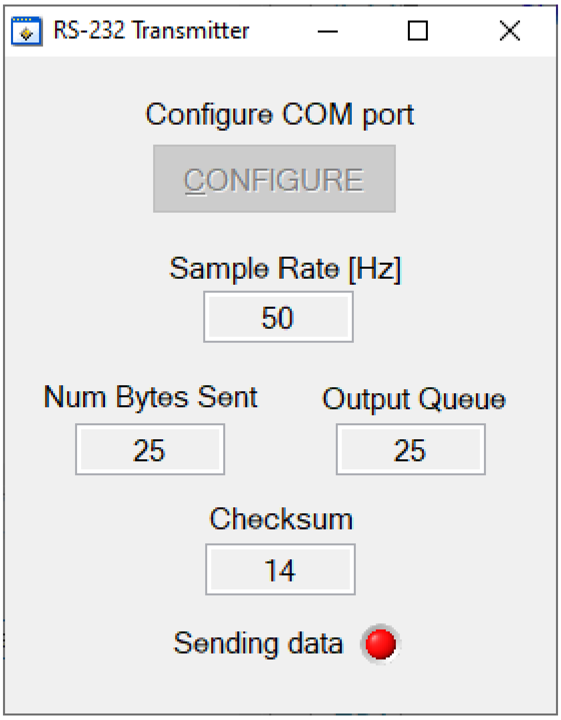
Num Bytes Sent: 0

Output Queue: 0

Checksum: 0

Sending data ●

Running


RS-232 Transmitter

Configure COM port

Sample Rate [Hz]: 50

Num Bytes Sent: 25

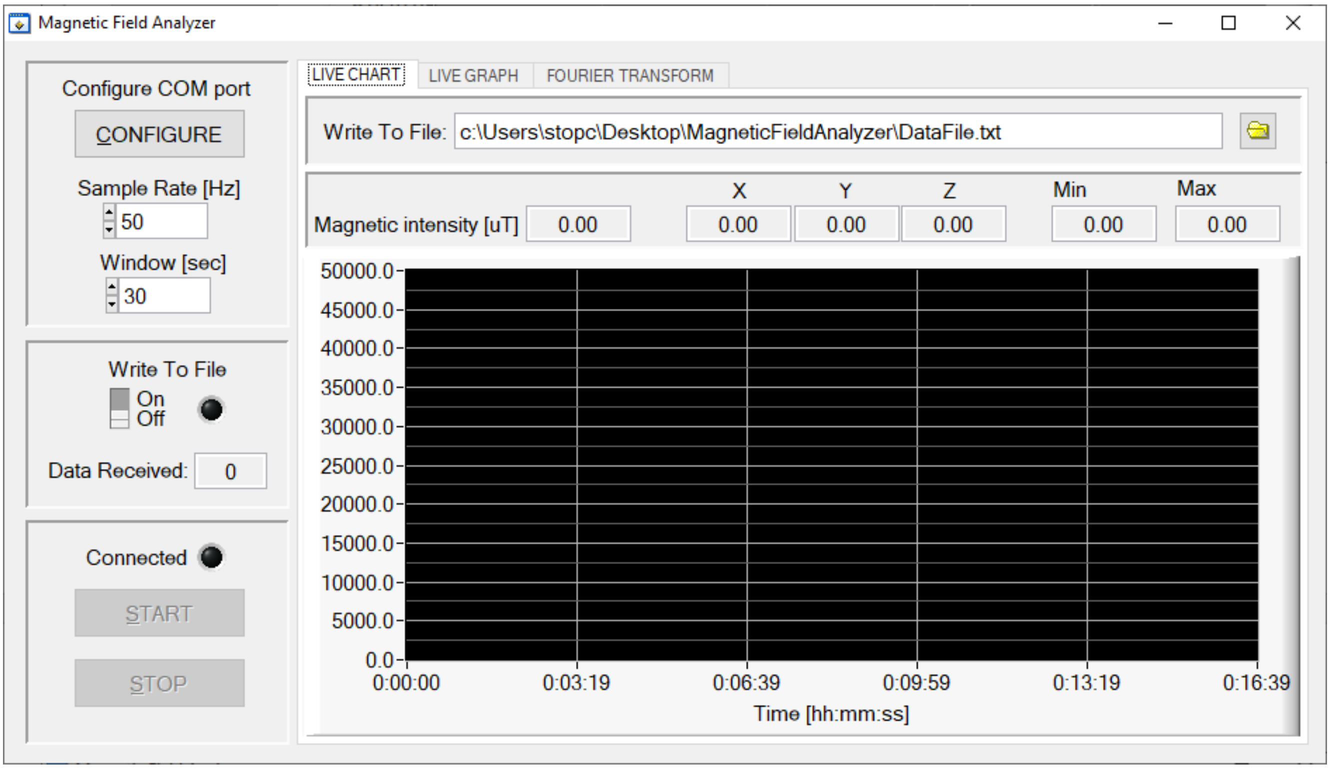
Output Queue: 25

Checksum: 14

Sending data ●

MagnoMonitor:

Idle



Sample Rate [Hz] 50

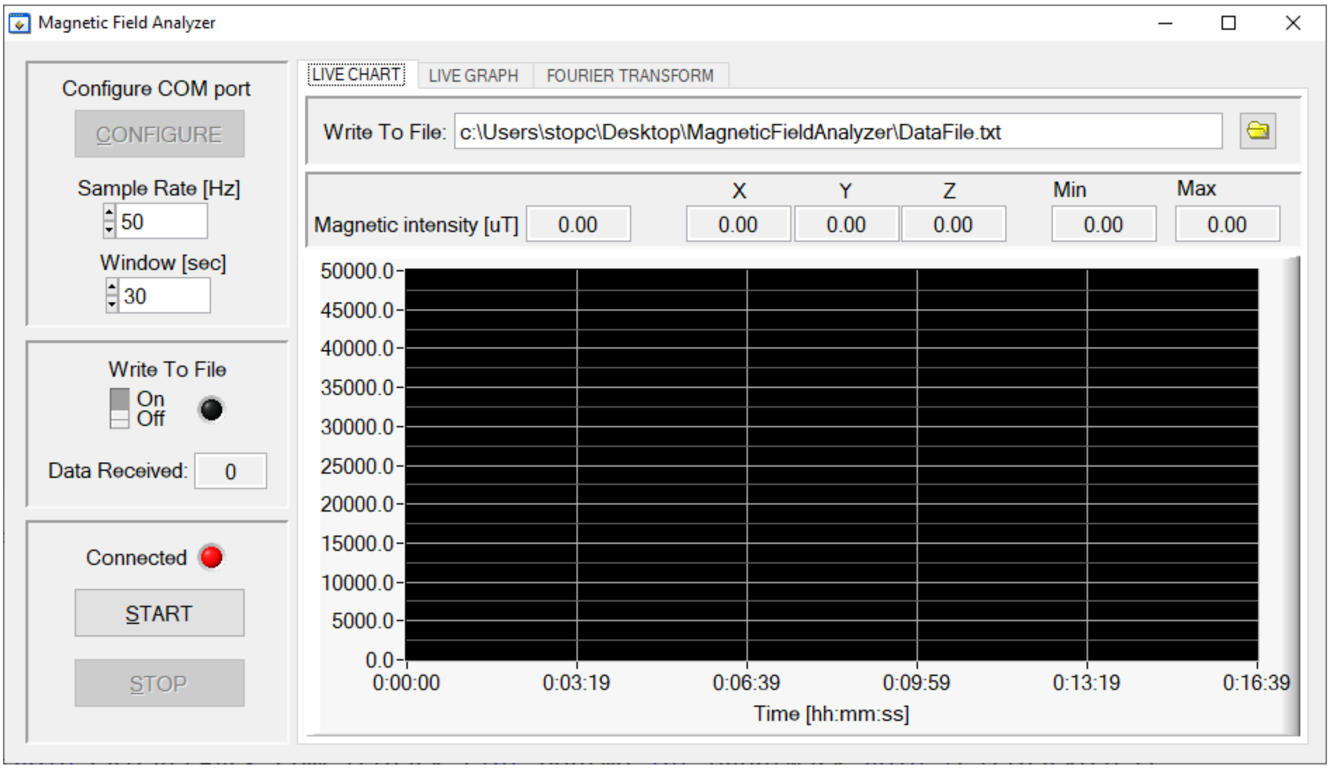
Window [sec] 30

Data Received: 0

Write To File: c:\Users\stopc\Desktop\MagneticFieldAnalyzer\DataFile.txt

		X	Y	Z	Min	Max
Magnetic intensity [uT]	0.00	0.00	0.00	0.00	0.00	0.00

Connected



Sample Rate [Hz] 50

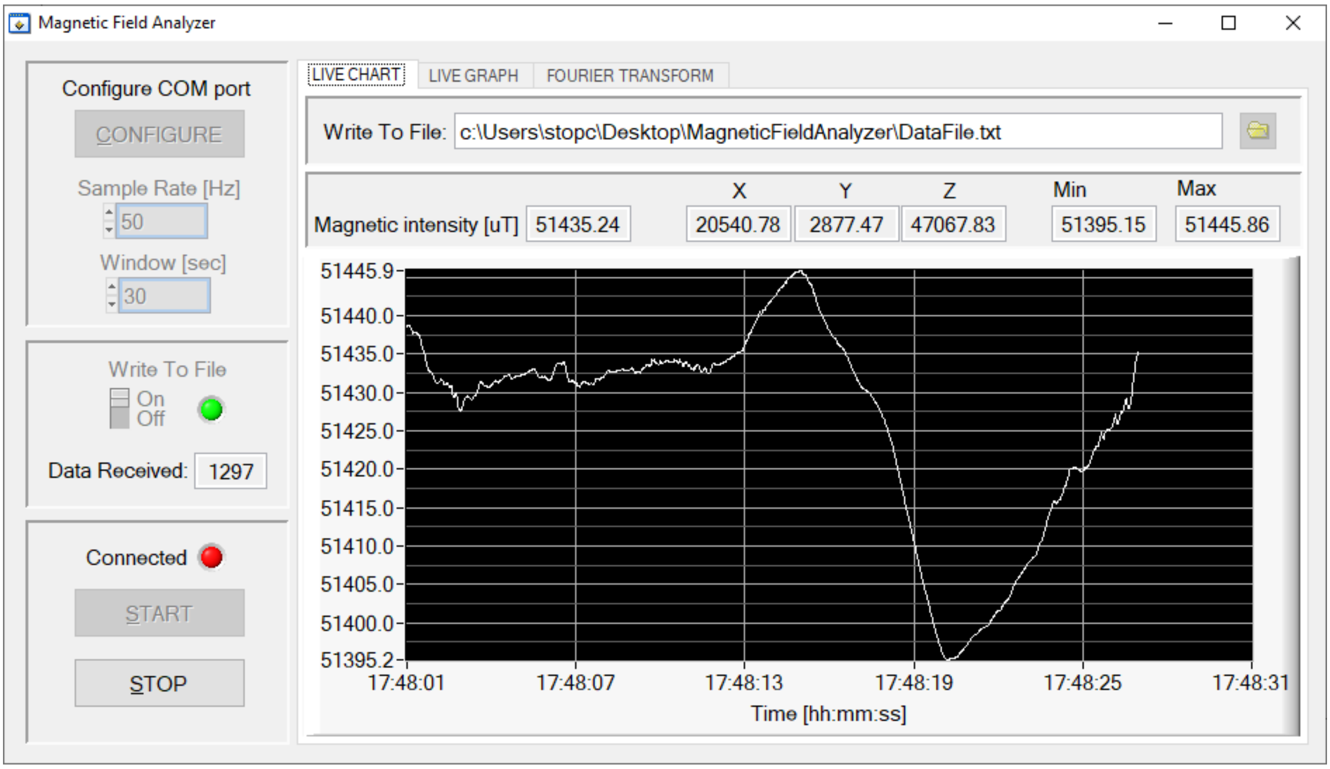
Window [sec] 30

Data Received: 0

Write To File: c:\Users\stopc\Desktop\MagneticFieldAnalyzer\DataFile.txt

		X	Y	Z	Min	Max
Magnetic intensity [uT]	0.00	0.00	0.00	0.00	0.00	0.00

Running



Magnetic Field Analyzer

Configure COM port

Sample Rate [Hz]: 50

Window [sec]: 30

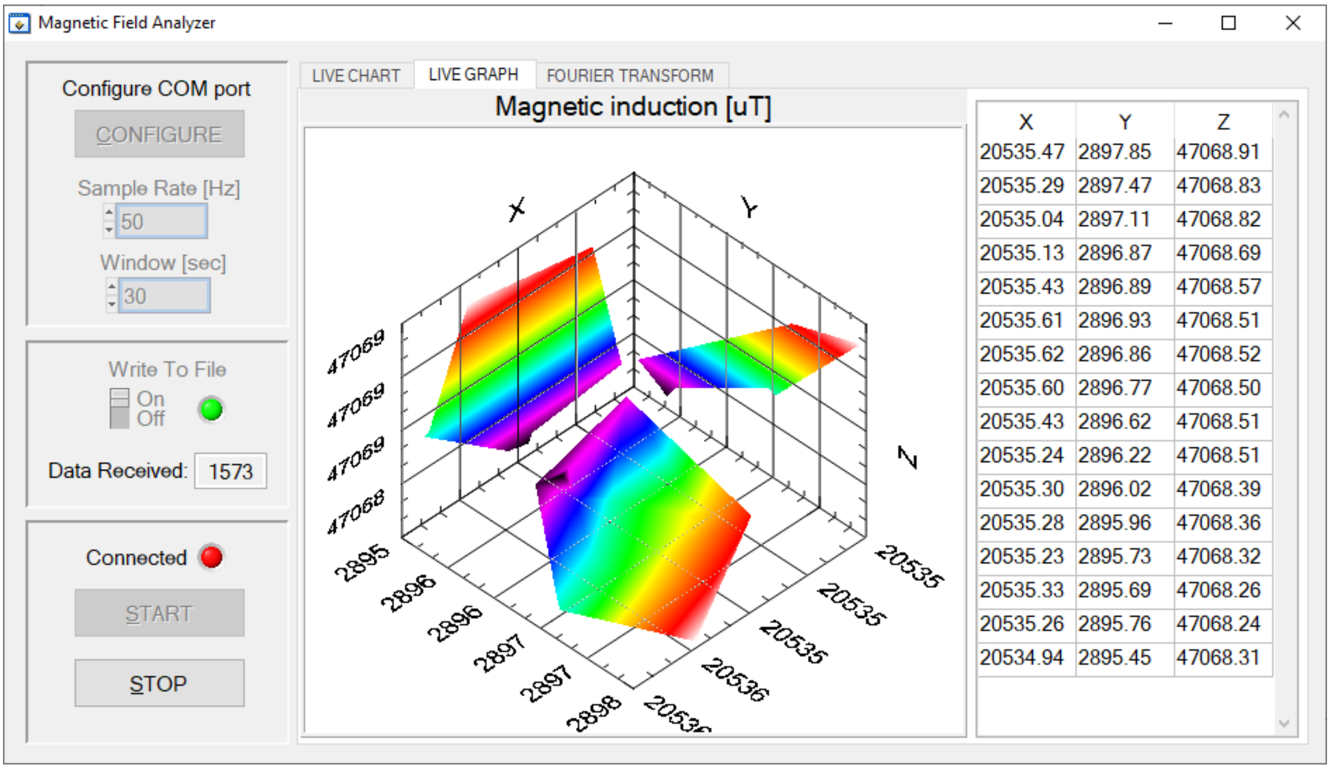
Write To File: On / Off ●

Data Received: 1297

Connected ●

Write To File: c:\Users\stopc\Desktop\MagneticFieldAnalyzer\DataFile.txt

Magnetic intensity [uT]	X	Y	Z	Min	Max
51435.24	20540.78	2877.47	47067.83	51395.15	51445.86



Magnetic Field Analyzer

Configure COM port

Sample Rate [Hz]: 50

Window [sec]: 30

Write To File: On / Off ●

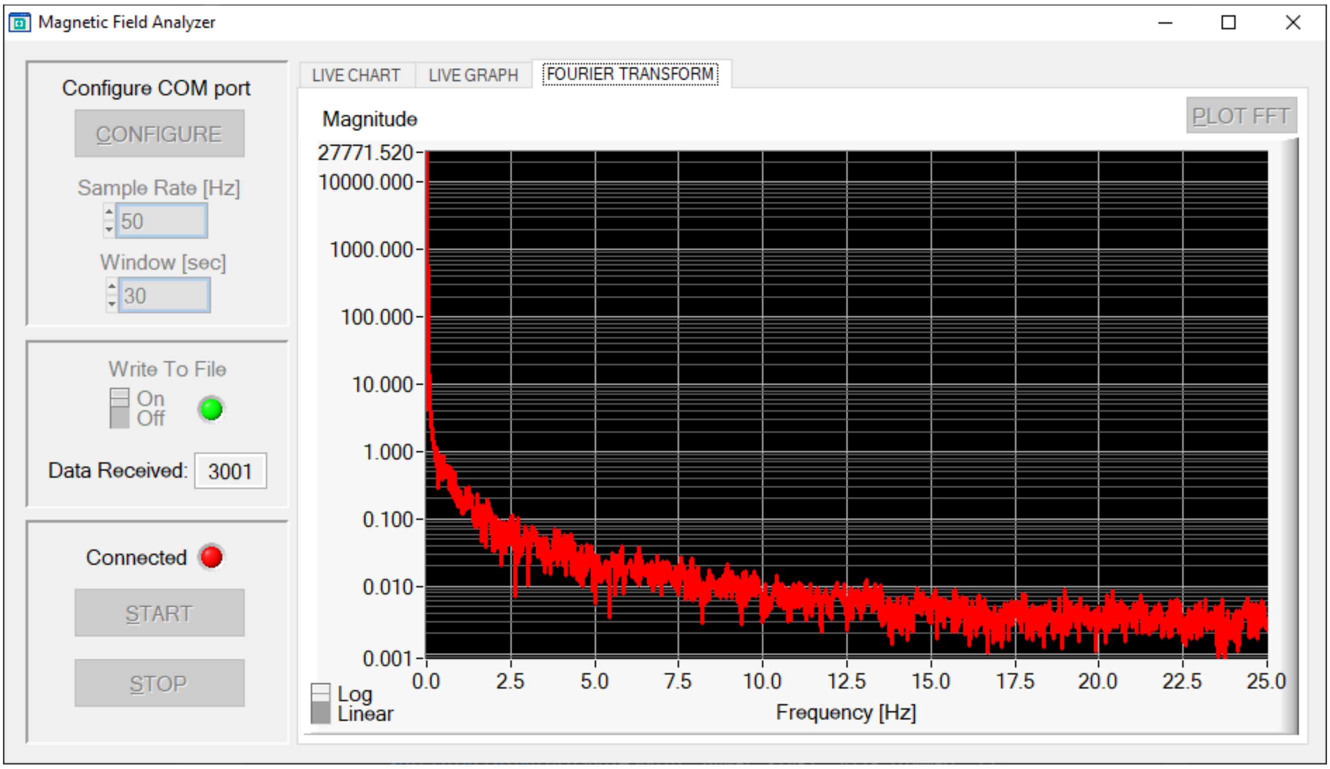
Data Received: 1573

Connected ●

Magnetic induction [uT]

X	Y	Z
20535.47	2897.85	47068.91
20535.29	2897.47	47068.83
20535.04	2897.11	47068.82
20535.13	2896.87	47068.69
20535.43	2896.89	47068.57
20535.61	2896.93	47068.51
20535.62	2896.86	47068.52
20535.60	2896.77	47068.50
20535.43	2896.62	47068.51
20535.24	2896.22	47068.51
20535.30	2896.02	47068.39
20535.28	2895.96	47068.36
20535.23	2895.73	47068.32
20535.33	2895.69	47068.26
20535.26	2895.76	47068.24
20534.94	2895.45	47068.31

Stopped

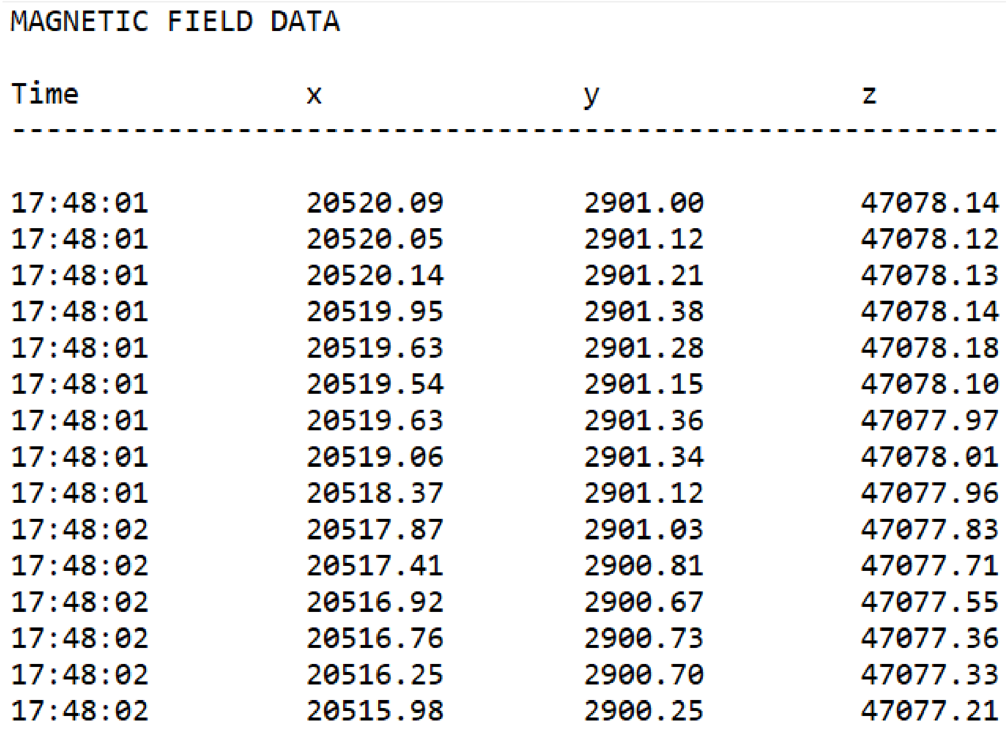


Sample Rate [Hz] 50

Window [sec] 30

Data Received: 3001

Data stored in a file



```
MAGNETIC FIELD DATA

Time                x                   y                   z
---------------------------------------------------------------------

17:48:01            20520.09            2901.00             47078.14
17:48:01            20520.05            2901.12             47078.12
17:48:01            20520.14            2901.21             47078.13
17:48:01            20519.95            2901.38             47078.14
17:48:01            20519.63            2901.28             47078.18
17:48:01            20519.54            2901.15             47078.10
17:48:01            20519.63            2901.36             47077.97
17:48:01            20519.06            2901.34             47078.01
17:48:01            20518.37            2901.12             47077.96
17:48:02            20517.87            2901.03             47077.83
17:48:02            20517.41            2900.81             47077.71
17:48:02            20516.92            2900.67             47077.55
17:48:02            20516.76            2900.73             47077.36
17:48:02            20516.25            2900.70             47077.33
17:48:02            20515.98            2900.25             47077.21
```

Operational instructions

Initial Setup:

1. Connect the measuring device (transmitter) to the computer using an RS-232 serial interface.
   - For the simulator: Use a virtual serial COM port – com0com software with default settings. Create two ports within the range of 1-8.
2. Launch the MagnoMonitor program.
   - For the simulator: Launch the Transmitter program.

Communication Port Configuration:

1. Click on the "CONFIGURE" button to access the communication settings.
2. Select the appropriate communication port that corresponds to the connected transmitter device.
   - For the simulator: Connect to one of the two ports that were created.
3. Set the required communication parameters according to the operating standards of the measuring device.
   - For the simulator: Baud rate: 128000, Parity: none, Data bits: 8, Stop bits: 1, CTS Mode: Off, Xmode: XOFF, Input/Output Queue Size: 4096.
4. Once the configuration is complete, click on "Confirm" to apply the settings.
   - For the simulator: Perform the same steps for Transmitter program.

Data Logging (Optional):

1. To log the received data, click on the toggle button under "Write to File."
2. To choose the desired location, click on the icon 📁 and choose the file name for the log file (stored as a text *.txt file).

Sample Rate and Window:

1. On the main panel, choose the appropriate sample rate that corresponds to the transmitter.
2. On the main panel, choose a time window for the strip chart in seconds.

Data Acquisition and Observing:

1. Once the connection is established, an LED indicator and a START button will light up.
2. Click on the “START” button to begin data acquisition and processing.
3. Use the tabs provided to switch between different views: live chart, 3D graph.
4. Click on the “STOP” button to stop the data acquisition and processing.
5. Navigate to the Fourier transform tab and click on the “PLOT FFT” button to calculate and plot on the graph the Fourier transform of the data received.
6. Click on the toggle button “Log/Linear” to choose the Magnitude axis representation.
7. Click on the close button to exit the program.

LED Indicators:

- The MagnoMonitor main panel includes two LEDs: Red LED indicates the connection status. The green LED indicates whether the writing to file option is on.
- The transmitter includes a real-time LED indicator that is blinking while data is being transmitted, providing visual feedback that the transmission is active.

Main algorithms and functions

To handle asynchronous data reception, the MagnoMonitor program uses multithreading. The main thread is responsible for user interface updates, while a separate thread is dedicated to handling serial communication and data reception.

ConfigCallBack:

The function prepares the program to receive and process data from the connected transmitter through the COM port. When the "Configure" button is clicked, the function invokes DLLConfigPort to set up the COM port and checks for RS-232 errors. It schedules a thread function for data reading, creates a new thread-safe queue task to securely transfer incoming data between threads, and installs a task callback.

ComCallback:

The com callback function is triggered on two events:

- When the function detects connection/disconnection (DSR line changes state event), the LED and START buttons are activated/deactivated.

- When a 25-byte packet arrives at the serial port, the program acquires a lock to prevent it from being discarded before the function completes execution. The function calculates a checksum to ensure data integrity. If the checksum is valid, it extracts x, y, and z values as a double type from the data packet, performs light processing, and updates the UI controls with the received data for real-time display. The processing involves calculating the Euclidean norm (2-norm): $intesity = \sqrt{x^2 + y^2 + z^2}$. If data logging is enabled, it writes the received data along with timestamps to the file. Then, the data is written to a thread-safe queue to avoid race conditions and ensure data integrity during further processing.

ProcessDataFromQueueCallback:

The task callback is triggered to process data when a certain number of vectors are available in the thread-safe queue. The function extracts from the data blocks separate values and saves sets of vectors to dynamically allocated arrays x, y, and z. Then it calls the VisualizeData function to display the sets on a 3D graph using the provided Instruments CW 3D Graph 8.0 functions. The exact values of the data points are displayed in a table.

PlotFFT:

The function can be called after the data acquisition is complete. It applies a Hamming window to the vectors in the time domain to reduce spectral leakage. The algorithm calculates the Fast Fourier Transform (FFT) in each direction, which is an efficient method for finding the Fourier transform of discrete data. Then, the resulting magnitude of each vector is arranged in a matrix. By applying the 2-norm to each row (vector) of the matrix, the function calculates the magnitude of each frequency component. This provides a single value that represents the overall strength of the frequency component across all three Cartesian coordinates. The function calculates the frequency resolution of the FFT: $\Delta f = \frac{sampling\_rate}{num\_vecotrs}$. It normalizes the magnitudes of the FFT by dividing each frequency component by half of the vector number (DC components are divided by the total number of vectors) to ensure accurate representation. The frequency array is generated up to the Nyquist frequency. The final output provides a frequency domain representation of the input signal and is plotted on the graph.

Problems and Solutions

The main problem was in testing and debugging the MagnoMonitor program operation. That is why the Transmitter – a simulator of data transmission was built. One challenge was to detect the connection status and control the transition of data. The solution involves checking and responding to changes in the DSR line status. To control the transition, a break signal is sent by the program, which is interpreted by the transmitter as a signal to start sending data.

The program does not have a resetting mechanism that would allow it to return to its default state without having to be relaunched. I did not incorporate this logic during the program's design, and attempting to add it later proved to require subtle tweaking. As a temporary solution, I have disabled any buttons that could disrupt the program's logic after the first interaction.

The program has little error-handling mechanisms as it would require more complex logic.

If the data acquisition is interrupted by the MagnoMonitor without disconnecting (such as during debugging), its input queue is filled. Reconnecting to the same port afterward causes issues as it seems to receive old data even though the COM port was flushed at the beginning and end. No solution has been found for this problem.

Switching to the tab with a 3D graph the first time while acquiring data occasionally stops the acquisition. A temporary solution was to force open the tab on the program launch. The cause of the issue wasn't found.

The code block for processing com callbacks is not properly locking, resulting in incomplete file writing upon quitting the program. No solution has been found for this issue.

Suggestions for Program Improvement:

- Add cursors and numeric controls to the graphs, enabling more interactions with graphs.
- Add harmonic analysis, allowing for the analysis and visualization of harmonics in the data.
- Expand data-saving options, enabling users to save different types of data.
- Implement more error-checking and handling mechanisms and provide informative error messages.
- Make the program circular, offering an option to restart from any point in the data acquisition process.
- Provide the option to choose different units for the data.
- Create additional graphs with multiple plots to visualize each vector individually.
- Introduce options to choose different window functions and antialiasing filters for signal processing.