Maxa dashboard code

Main classes:

* MainWindow – main
* Charts – handles creating the charts, adding new data points, removes redundant data points and old data points to avoid excessive memory usage.
* ErrorCodes – declares all maxa error codes with related register, description, and bit mask to access the specific error.
* Registers – declares all maxa registers addresses and some bit masks to control certain functions
* FileWriter -handles writing data read from the maxa to a csv file.
* Maxa – This class handles the communication with the maxa. It updates UI variables and uses the FileWriter class to write the data to the log file.
* NotifyNewData – This class is used for the data binding between the software and the UI elements.
* UserMessages – Displays messages to the user on a dedicated window and controls their removal after they expire.

MainWindow main responsibilities:

At startup:

* Initialize UI elements, and fill combo boxes with comboBoxItems
* Start a 2 second timer (regulating data collection from Maxa)
* Initialize setpoint label and charts datapoints

On User Input:

* Set SCV file name and path
* Attempt to connect communication with Maxa
* Activate and deactivate data recording to CSV file
* Apply setpoints and operation mode to Maxa

Continuously:

* Handle OnClick events
* Verify and auto correct invalid setpoints set by user
* Verify continuous communication with Maxa
* When disconnected from Maxa notify user and attempt to reconnect
* Update charts with new data points

Main loop (called by timer every time it elapses):

* Verify last communication cycle ended
* Verify connection flag
* Attempt to reset error (if any activate)
* Set new setpoints and operation mode (if requested by user)
* Activate special functions (if requested by user)
* Read all data points from Maxa
* Write data to CSV (if requested by user)
* Update active errors flag
* If communication is not successful for longer than 30 seconds – notify user
* Attempt to reconnect if communication disconnected
* Flag that the communication cycle has ended

Charts main responsibilities:

* Generate series for the 2 charts (temperatures and pressures)
* Add data points to the series
* Remove redundant data points when it doesn't affect the series visually
* Remove excessive data points from series to limit memory usage
* Hide and unhide series in the temperature chart
* Refresh time axis values to show new datapoints

ErrorCodes usage:

This class declares all the errors from the Maxa modbus table and generates a bit mask to retrieve each error. That class was planned to have more usage but currently only the bit mask for each error is being used (by the ReadErrors function in the Maxa class).

Registers:

This static class contains all addresses of the registers in the Maxa machine and also some bit masks for registers 7201, 7202, 7214 and 7216

FileWriter:

Uses a public dictionary to enable other classes (specifically Maxa class) to add the data to be written to the CSV file. On the first writing attempt it first writes the headers to the file, after that it only writes the data points. It uses a different thread to perform the actual writing to the file.

Maxa:

This static class is responsible for the communication with the Maxa machine - reading and writing. It updates the UI variables in NotifyNewData, and the data to be writen in the csv file if relevant using the fileWriter.dataDictionary.

NotifyNewData:

This class implements the INotifyPropertyChanged interface and is used for data binding. The class declares a property for each UI element required for data binding and calls the INotifyPropertyChanged whenever a property is being updated either by the user or by the software.

UserMessages:

This class handles the display of messages to the user. It gets a stackPanel on initialization to display messages on. It handles the adding of new messages after verifying that they are not already displayed. It also removes of old messages when their time ends.