MOD.UI USER MANUAL

1.0 Introduction

The purpose of this Application is to communicate with any MODBUS RTU enabled device over the serial port of a Windows computer. Setup files are used to define the parameters that need to be read/written. The Application will generate tabs and buttons corresponding to the specified parameters. The user can then connect with the device to read and display all the values of all said parameters. Some of these can be modified by the user and written to the device.

2.0 Flow

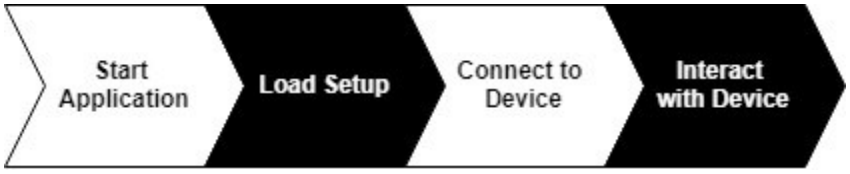

Fig 2.1 Steps to use Application

2.1 Start Application

2.2 Click on ‘LOAD SETUP’ in the top-left corner; this will open a File Dialogue window. Select the required setup file:

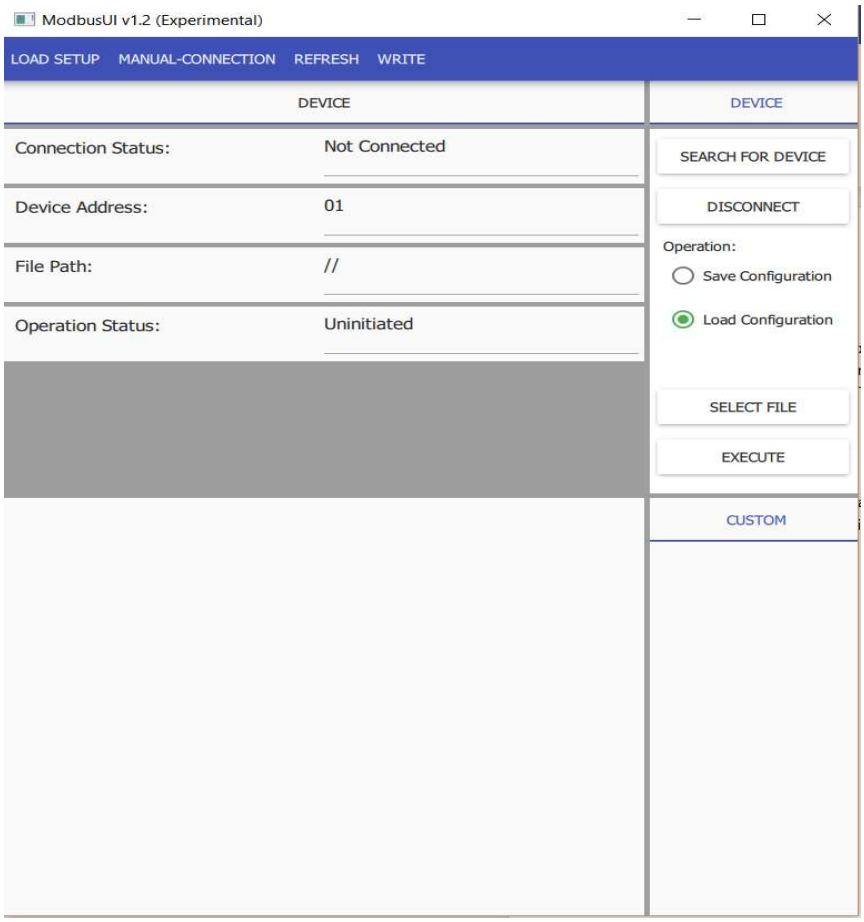

Fig 2.1: Starting Appearance

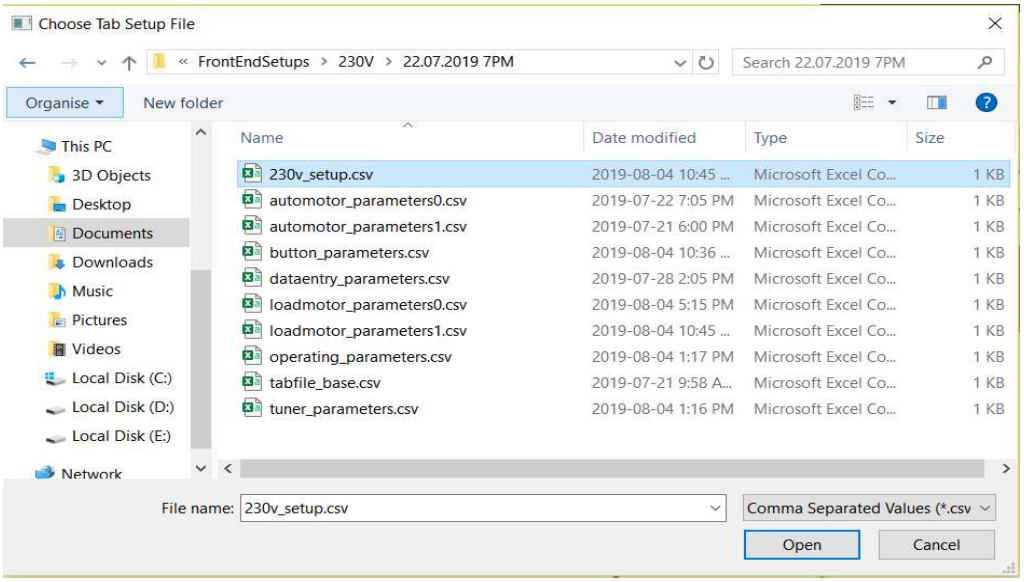

Fig 2.2: File Dialogue

2.3 The application will now generate a layout that will allow the user to read and write to all the parameters specified in the setup files. The following is an example only; the only constant is the 'Device' box in the top-right corner.

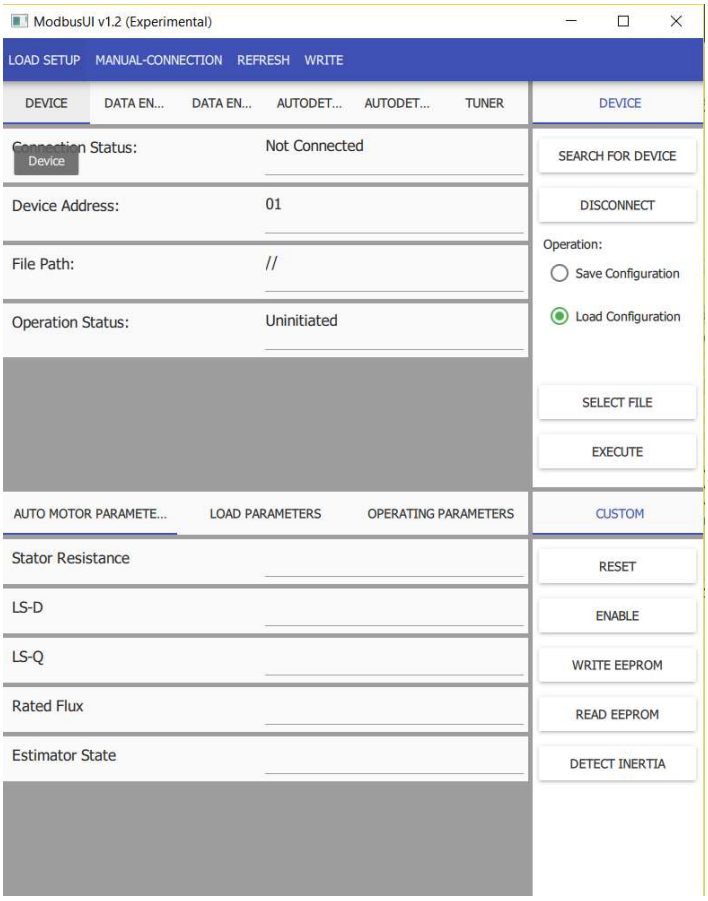

Fig 2.3 After loading setup file (example only)

The Stable parameters are contained within the upper tabs. The Mutable parameters are contained within the lower tabs. The Button parameters are contained within the 'Custom' box in the lower-right corner. This will be explained further in the Setup File section.

2.4 Plug the Drive into the USB port. Now, either click 'SEARCH FOR DEVICE' to automatically search the available COM Ports or click 'MANUAL-CONNECTION' to manually select and open a COM Port. If the COM Port was manually opened, click on 'SEARCH FOR DEVICE' to find the MODBUS Device on that COM Port.

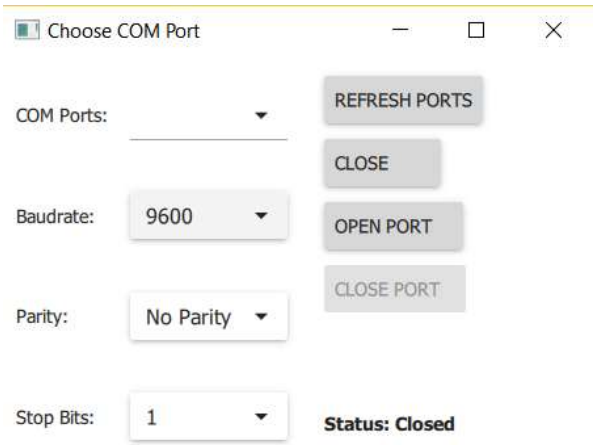

Fig 2.4 Manual Window; hit REFRESH PORTS after plugging in device

2.5 Assuming the device is now connected, the user may interact with the device using two functions:

2.5.1 READING PARAMETERS: Upon connecting, the application will first read all the (Stable and Mutable) parameters sequentially and update their values in the UI. Then, it will continuously poll and update the Mutable Parameters as well as the Button Parameters. Clicking 'REFRESH' in the bar will refresh the Stable values.

2.5.2 WRITING PARAMETERS: the user can go to any tab, modify the values of any parameters; clicking 'WRITE' in the bar along the top will result in these values being written to the device. Clicking on the Buttons, however, will immediately queue a Write command to invert their value (if 0, it will write 1, else it will write 0).

3.0 Setup Files

3.1 Purpose: Setup files are used to instruct the application on what parameters are to be interacted with and how they are to be treated.

3.2 Structure: There are three types of setup files, the Root File and Button File, each of which there should be only one, and the Tab Files, of which there can be multiple.

3.2.1 Root File: a CSV-format file with 3 columns, it is essentially a vertical list of which tab files are to be taken as input, their titles as well as their type (stable, mutable or button). Columns:

3.2.1.1 Title: this field should contain the 'name' of the tab; this will be displayed as the tab header once it has been loaded into the application

3.2.1.2 Type: this field should be either 0, 1 or 2. 0 indicates that the file contains Stable parameters. 1 indicates that the file contains Mutable parameters. 2 indicates that the file contains Button parameters

3.2.1.3 Path: this field should contain a path to the file **relative to the root file**. If the tab files are in the same folder as the root file, then simply put in their names.

	A	B	C	D	E
1	title	type	path		
2	Data Entry	0	dataentry_parameters.csv		
3	AutoDetect Motor	0	automotor_parameters0.csv		
4	AutoDetect Load	0	loadmotor_parameters0.csv		
5	Tuner	0	tuner_parameters.csv		
6	Auto Motor Parameters	1	automotor_parameters1.csv		
7	Load Parameters	1	loadmotor_parameters1.csv		
8	Operating Parameters	1	operating_parameters.csv		
9	Buttons	2	button_parameters.csv		

Fig 3.1 Example Root File

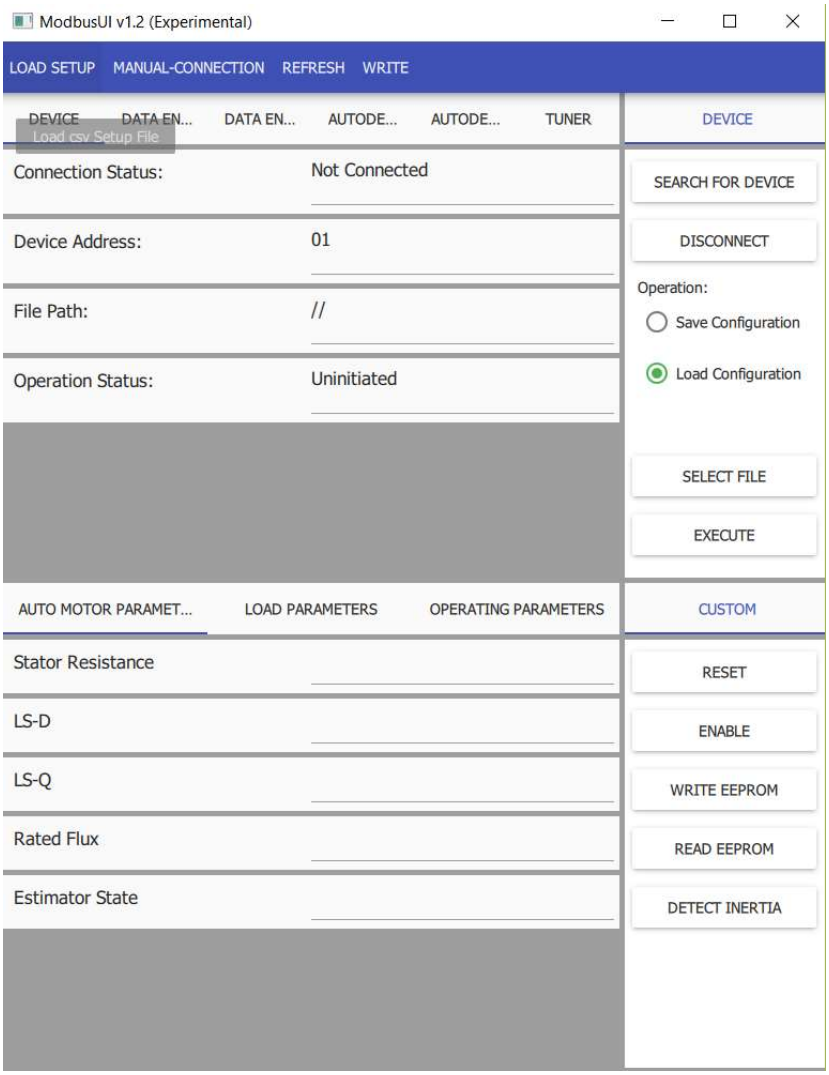

Fig 3.2 Resulting Application UI

Note that the Titles of the Tabs correspond to the Name in the root file. Further note that the Type 0 (Stable) tabs are in the upper panel, whereas the Type 1 (Mutable) tabs are in the lower panel and Type 2 (Button) file contents are in the 'CUSTOM' box.

3.2.2 Tab File: a CSV-format file with 5 columns, it is a vertical list of parameters. The parameters in any given Tab File will be grouped together, regardless of whether their addresses are contiguous. Columns:

3.2.2.1 Name: Contains the name of the parameter as it should be displayed in the application

3.2.2.2 Size: Specifies whether the parameter is 16-bit or 32-bit; 1 indicates 16 bits and 2 indicates 32 bits (ie two 16-bit fields)

3.2.2.3 Factor: Indicates how the data received from the device should be parsed. If factor is set to 0, it will be parsed as an integer. If it is 1, it will be treated as a Q24 value (that is, parsed as an integer, type-

casted to float and then divided by 2^{24}). If it is 2, it will be treated as a Q24 value but also multiplied once by 1000. Finally, if it is 3 it is treated as a Q20 value (same as Q24, but divided by 2^{20} instead of 2^{24}).

3.2.2.4 Address: Should contain the MODBUS address of the parameter. If the Address is set to X, and the Size value set to 2, it is assumed that the address is both X as well as X + 1.

3.2.2.5 Description: An optional column containing a description of the parameter. This description can be seen in the application as a tooltip by hovering over either the associated button or text area (whichever is applicable)

	A	B	C	D	E
1	name	size	factor	address	description
2	Stator Resistance	2	1	18	Resistance of the Stator
3	LS-D	2	1	20	LSD
4	LS-Q	2	1	22	LSQ
5	Rated Flux	2	1	24	Rated Flux of the Motor
6	Estimator State	1	0	97	Estimator state

Fig 3.3 automotor_parameters1.csv from the setup file in Fig 2.2.1.1

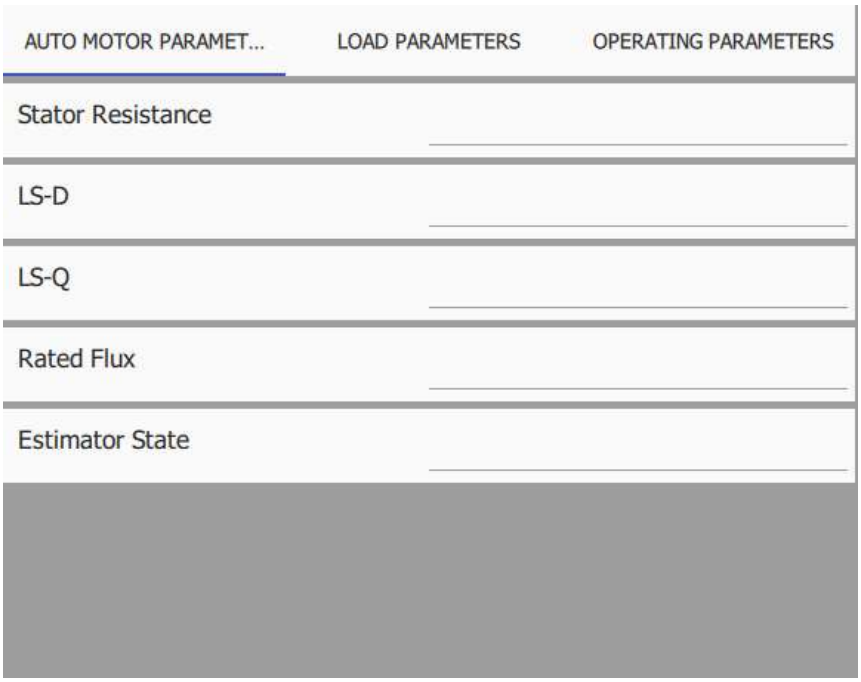

Fig 3.4 Resulting Tab in the application UI

3.2.3 Button File: a CSV-format file indistinguishable from Tab Files. The application only knows to treat them as button files due to the Type field in the Root File. Each parameter in the Button File will be treated as a binary value. Each parameter will have an associated button that is RED when the value is read as 0 from the device, and GREEN when the value is read as 1

from the device. Clicking a button while it is RED will result in the application sending a write command to set that parameter to 1 in the device. Similarly, clicking the button while it is GREEN will queue a Write command that will set the parameter to 0 in the device.