1 What CanzeRead is for

CanzeRead program (or CanzeLogRead, as is currently named on Github), has the purpose of interpret CanZE-created log files, and create from text files that can be easily read through spreadsheets, text editors and plotting files. These text files are a straightforward structure (see later) and “ADF”£ extension (stands for Ascii Data File”). They can read and interpreted “as are”, by the following plotting and post-processing program:

http://ceraolo-plotxy.ing.unipi.it/default.htm

which was used to make the pictures shown in this document.

Because CanZE logs contain samples having a different timestamp each, the better way to store curves and prepare them for further visualization and post-processing, appears creating individual text files for individual variables.

Time in each CanZE log row is constituted by a huge string containing year, month, day, hour, minute, and seconds (with milliseconds).

It is believed that the best way to make plots is as having on the abscissa the elapsed time from trip inception.

Names of the different variables stored by CanZE can be, in some cases, very long, and contain spaces; therefore they are nt very adequate as variable names for plotting.

Therefore CanzeRead uses short names for different variables, as follows:

- In case no specific information is present (see next dots), the short name is equal to the variable code for instance variable coded as 793.622001.24, which corresponds to “Mains phase 1 current RMS value” takes as short name 793.622001.24;
- For some special classes of names, which refer to many variables each: cell voltages, cell temperatures, balance switches, short names are internally defined by the program. For instance, cell voltages are vCell01, vCell02, …, cell temperatures tempCell01, tempCell02, … balance switches are balanceSwwitch01, balanceSwitch02, etc.
- Users can override any short name automatically defined by the program as per the previous two points, providing customised short names in a text file, which, to be read, must be located in the same folder in which the log file to be interpreted resides.

This is a simple example of file **`ShortNames.txt`** (four rows):

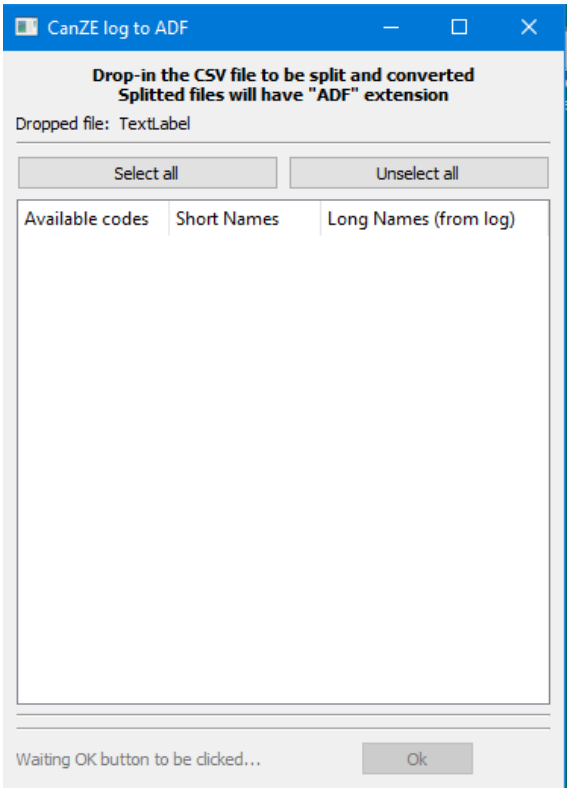
```
7ec.623203.24 vBatHV
7ec.623204.24 iBatHV
7ec.622002.24 batSOC
793.62504a.24 pMains
```

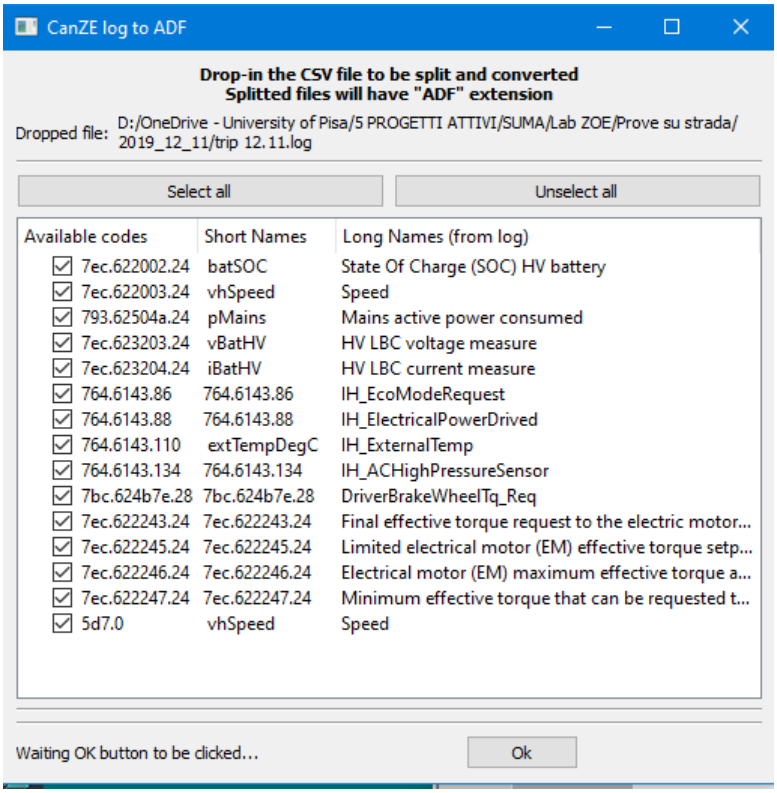
2 Usage

The program usage is rather straightforward: it is sufficient to run it, and drop on it the CaZE log file to be read. CanZerRead will display the variables read from the file, in three ways: code, short name, long name. The usera cah nselect all of them or part of them to be used to create output files.

The output file names will contain date info from the log file, plus the variable short name.

Below an example of the program screen, before and after having dropped the file “trip 12.11.log” on it:





The first few rows of the log file are as follows:

```
Datetime,SID,Name,Value,Unit
20191211154043903,7ec.622002.24,State Of Charge (SOC) HV battery,79.04,%
20191211154043926,7ec.622002.24,State Of Charge (SOC) HV battery,79.04,%
20191211154044382,7ec.622003.24,Speed,0.00,km/h
20191211154044810,793.62504a.24,Mains active power consumed,0,W
20191211154044825,793.62504a.24,Mains active power consumed,0,W
20191211154044828,793.62504a.24,Mains active power consumed,0,W
20191211154045349,7ec.623203.24,HV LBC voltage measure,378.5,V
20191211154045363,7ec.623203.24,HV LBC voltage measure,378.5,V
20191211154045749,7ec.623204.24,HV LBC current measure,-0.8,A
20191211154045760,7ec.623204.24,HV LBC current measure,-0.8,A
```

The first rows of two created files are as follows (separator between the two columns is a tab “\t” character).

First rows of file “trip 12.11_batSOC.ADF”:

```
//2019-12-11; 15:40  Long name: "State Of Charge (SOC) HV battery"
t            batSOC
0            79.04
0.023        79.04
5.97         79.04
5.985        79.04
11.008       79.04
```

First rows of file “trip 12.11_7bc.624b7e.28.ADF”

```
//2019-12-11; 15:40  Long name: "DriverBrakeWheelTq_Req"
t            7bc.624b7e.28
3.071        3034
8.731        3056
13.814       527
20.87        2
25.469       2
```

3 Notes

Log files may be very long. However, the variables logged should appear in the first rows of the log file. Therefore, to avoid overheads, variable codes and names are searched in log files only in the first 1000 rows.