

PMP 7605 GUI:

INSTALLING Windows® APPLICATION

The available download also includes a file named PMP 7605 GUI.zip. Copy the contents of this folder to any desired folder on your computer. This causes the Settings.ini file to be written to the desired folder on software exit. This action cannot occur if the application is run within the zip file.

Make sure to first unzip all the contents of the zip file into a folder on your computer. After you unzip the contents, your folder should look like the following.

Double click on 'setup' to run the application. The first time that you run this application you will see the following message.

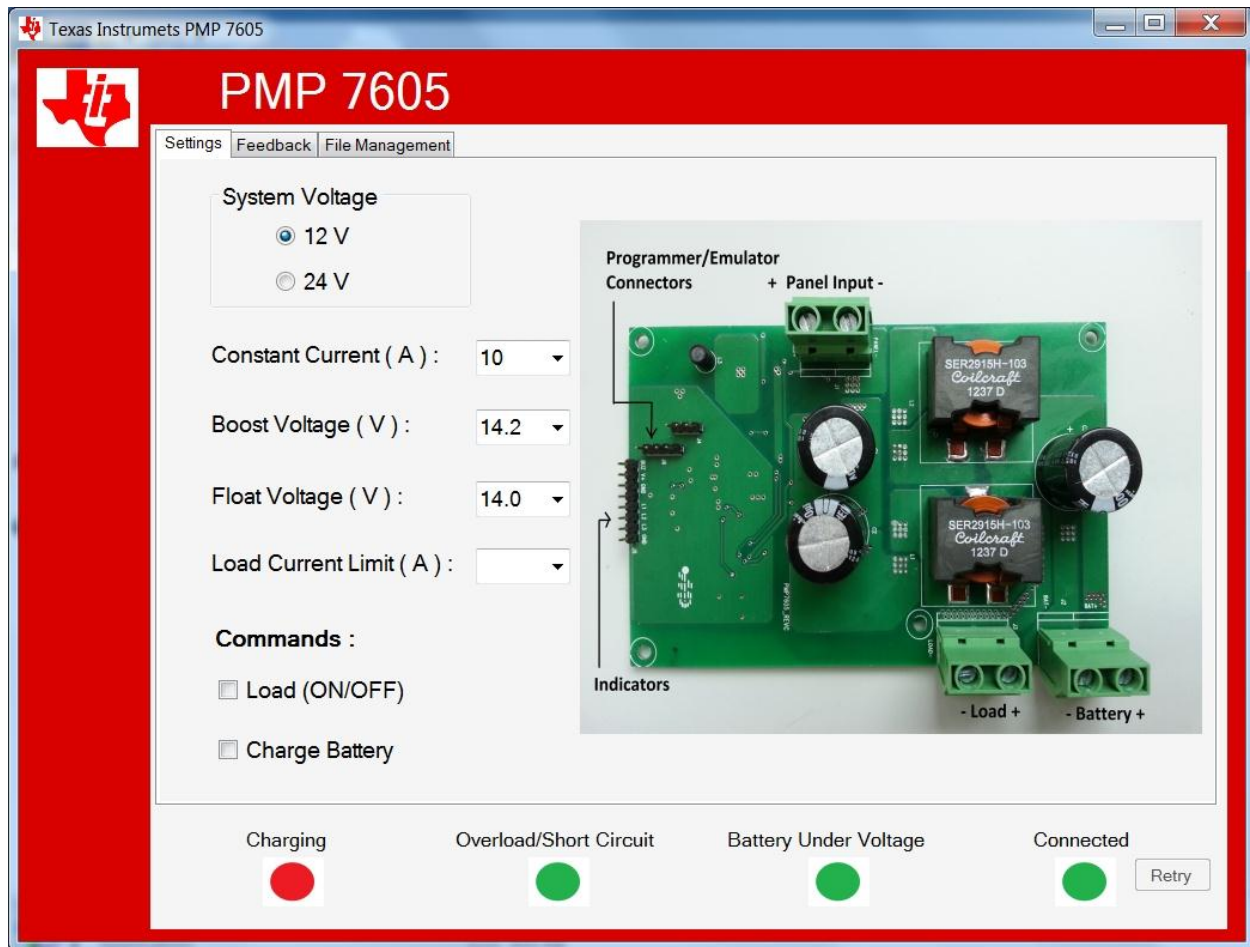
Click the 'Install' button. You should see the setup in action. Once done extracting the required files, the application will launch itself.

RUNNING THE APPLICATION

For future use, the application can be opened from the following path: Start → All Programs → Texas Instruments Inc. → test

4.1 Connecting the Hardware to GUI

MSP430 launch pad is used to establish the connection with the hardware. The same launch pad can be used to program the controller as well. For programming the controller, take out Vcc, test, reset and ground pins out of the launch pad and connect to the respective pins of the board. To connect the hardware to GUI connect Rx,Tx of launch pad to L3,L2 indicator pins respectively and short the grounds.



Indicators:

GUI has 4 indicators.

Connection status is indicated by the right most LED indicator. Whenever the GUI is launched, it will automatically detect the port and connect itself with the board. If the board is not present/powered up, it won't be detected and error message will be displayed.

If the connection could not be established please check for the following.

- The EVM is power is present.
- The 3V3 (JP1) jumper is in place.
- The connection cable between the PC and Hardware is connected.

In such case, check the connections again the press 'Retry' button.

Once the communication is established, if the communication is lost in the middle, error message will be displayed. Check the connections and launch the application again.

Other indicators such as charging, overload/short circuit and battery under voltage indicate the corresponding status of the system.

Basically the GUI has 3 tabs: Settings, Feedback and File management

Settings:

System Voltage: Depending upon the battery being used user can select either 12v or 24v.

Depending upon the system voltage, range for battery charge profile parameters will be updated.

Constant Current: Here user can choose the CC limit of the battery. Available options are 5,10,15 and 20A.

Boost Voltage and Float Voltage: For 12v battery these parameters range is 13.6v to 14.4v

And for 24v, range is 27.2v to 28.4v.

Load Current Limit: Here user can specify the maximum limit for load current. Available options are 3,6,9 and 12A.

Commands:

Charge battery and Load enable are the two commands. Ticking the checkbox will execute the corresponding commands.

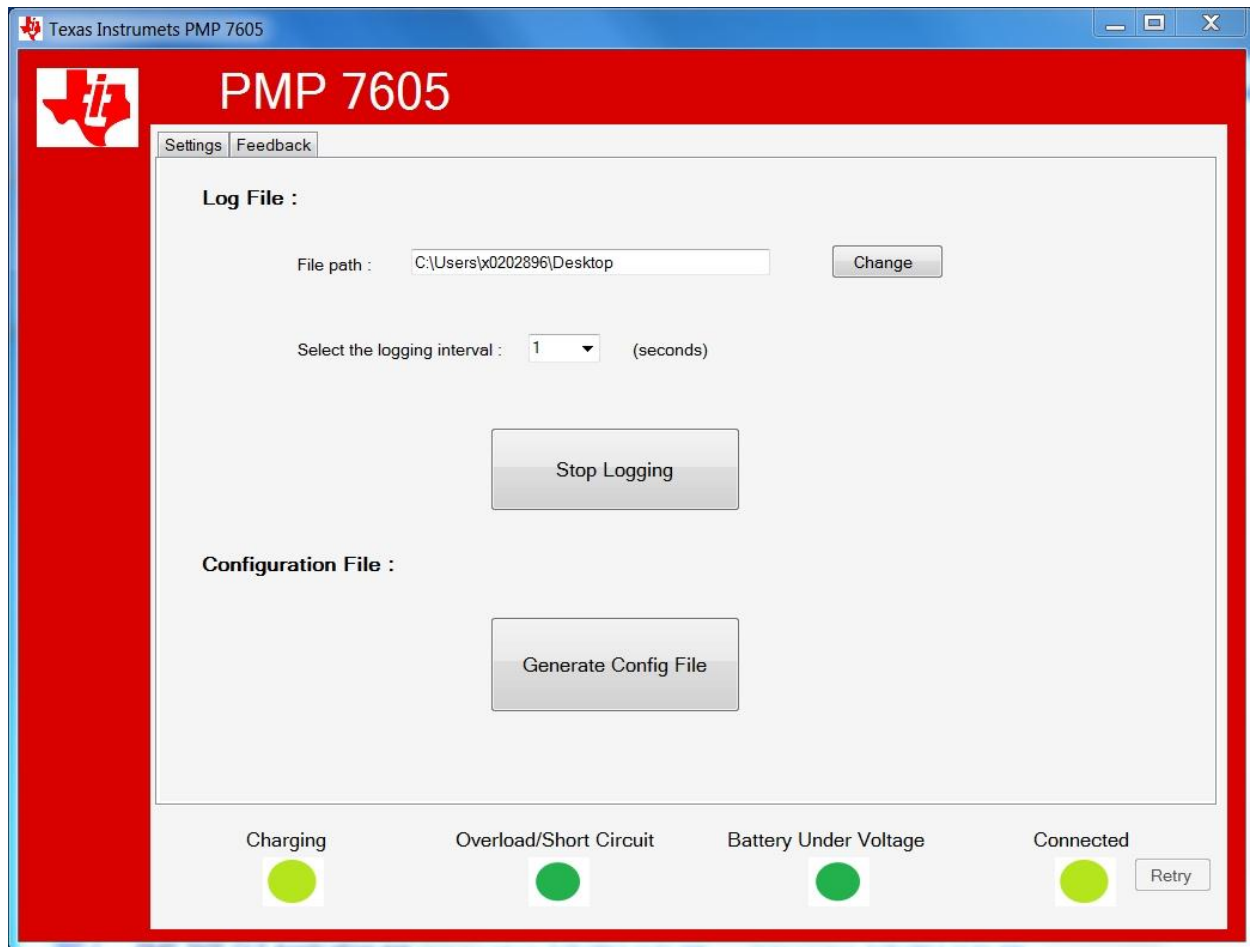
User can change the battery profile parameters even during the run time i.e when battery is charging and/or load is on.

Feedback:

Here user can view the plots of panel voltage, panel current, battery voltage and battery current with respect to time in real time. These are the readings observed by the microcontroller. Ranges for these plots will change corresponding to the system voltage.



File management:



File management basically involves two options :

Log file:

Here user can select the location of the log file. File will be saved in the selected location with name 'PMP7605 Log File' in .txt format. If there is already one text file with same name in that location, new readings will be appended to that file at the end. If there is no file with the same name in that location, new file will be created.

Here user can also select the logging time interval. Available logging intervals are 1, 5, 10 and 30 seconds. On clicking the 'Start Logging' button, system parameters such as panel voltage, panel current, battery voltage and battery current will be logged in real time along with time stamp into the text file. Changing path and logging interval during logging will have no effect. Events such as over load, battery under voltage will be logged into the file along with timestamp. Logging can be stopped by clicking on 'Stop Logging' button.

Configuration File:

Configuration file will contain the values of system parameters selected in the GUI. On clicking the 'Generate Configuration File' button, a window will pop up asking for the destination location. If there is already one configuration file in the destination location with the same name, it will notify saying 'There is already one file named config in the destination. Would you like to replace?'. If yes, old config file will be updated with the new values otherwise user can again select the destination. It will be saved with the name 'Config' in .C format.

RUNNING THE HARDWARE IN STAND-ALONE MODE

The firmware loaded by default in the MCU on board is a User Experience code. Using this firmware will need the GUI to work properly.

However, once the parameters are tuned, the output of the GUI can be used to operate the hardware in a stand-alone mode (without GUI). These parameters are given by the GUI in form of the 'config.c' file which is further used to change the code inside the MCU on hardware.

To use the system without GUI, user can follow these steps:

1. Add the generated Config.c file to the CCS project.
2. Remove/comment the statement **'#define GUI'** at the beginning in main.c, GUI.c and config.c files.

To use the system without GUI, add the statement **'#define GUI'** at the beginning in main.c, GUI.c and config.c files.