

Generic Power Meter Format Specification

Generic power meter configuration files are used to configure the EMC32 power meter driver in order to enable it to remotely control a power meter for which no dedicated driver exists via its GPIB bus interface.

Two example files `RsNrvd.DeviceConfiguration` and `Fluke.DeviceConfiguration` are available from the EMC32 installation CD-ROM. Please refer to them or the example listing below for further clarification of the format specification.

Listing of `DemoPowerMeter.DeviceConfiguration` can be found in `GenericPowerFileListing.pdf`

A power meter configuration file must conform to the following basic formal rules:

- The file shall be written in ASCII text format.
- It must be located in the `\Execute\Configuration` subfolder of the EMC32 main installation folder.
- Its name must be `<xxx>.DeviceConfiguration`, where `<xxx>` stands for an arbitrary descriptive name.
- Its contents shall conform to the syntax of Windows initialization files (extension `.ini`), that is, all contents shall be arranged in sections, each section containing an arbitrary number of lines, each line being composed by an entry string followed by a '=' character and the data string associated to the entry.
- Comment lines are allowed at any place throughout the file and must start with a ';' character.

The following sections and entries in the file are mandatory:

- Section `FileInfo`. This section identifies a file for EMC32. If not available, EMC32 will refuse to open the file. For the contents of this section please refer to the demo file printout above.
- Section `General`. This section must contain at least the line `"Driver=GenericPowerMeter"`. This line will identify the file as a configuration file for a generic power meter. If this line is not present, the file will not be displayed in the configuration file selection box in the Generic Power Meter Properties dialog.
- Section `Measure`. This section must contain at least two lines:

One line reading `"Count=<cnt>"`, where `<cnt>` stands for an integer number which must be greater than 0.

A second line reading `"GpibLine1=<cmd>"` where `<cmd>` stands for the IEEE bus command to be sent as measurement value query.

Without these two lines, the file will not be accepted.

The rest of the file's content is optional:

- Section `GpibSettings`. Contains one line `"EOITermination=<i>"` to set the termination character for IEEE communication. `<i>` stands for an integer according to the following meaning:

`i = 1`: Termination character is a Carriage Return character (hexadecimal 0D)

`i = 2`: Termination character is a Line Feed character (hexadecimal 0A). This is the default if the line is not present.

`i = 3`: All strings are terminated with a CR and a LF character in sequence.

Additionally, the section may contain a line `"GpibTimeout=<time>"` where `<time>` stands for an integer determining the value in milli-seconds to be used as general timeout for IEEE bus communication.

- Sections `Initialize`, `Channel`, `Unit`, `Speed`, `Zero`, `Trigger`:

In each of these sections a set of commands may be defined which will be sent to the device when programming the corresponding parameter:

`Initialize` --> device initialization during software start

`Channel` --> measuring channel selection (for a multi-channel device) at test start

`Unit` --> setting the measurement unit to dBm at test start

`Speed` --> measurement speed is set at test start

`Zero` --> perform a zeroing on the device at test start

`Trigger` --> trigger a data acquisition prior to each measurement query

Defining IEEE bus command sets always follows the same philosophy:

A line "Count=<cnt>", where <cnt> is an integer, tells the driver that cnt command lines will follow, each line defining a command.

The command lines themselves then shall read "GpibLine<i>=<cmd>" where <i> is an index ranging from 1 to the command line count defined before, and <cmd> defines the IEEE bus command.

Additionally, <cmd> may contain a waiting time indication for the command: for example, "@1000@ABC" defines that EMC32 shall delay further execution for 1000 ms after sending the command "ABC" to the device.

The use of any of these sections and command sets is optional. Even defining command strings for setting different measurement speeds is not required although the settings dialog will allow for different speed selections. If no strings are defined, no strings are sent; if less than four strings are defined, the last in the list is used when trying to use non-existing strings.

- Sections Identify, Measure:

These sections contain query commands. Identify will be used at software start-up to check whether the device found at the given IEEE bus address is the supposed power meter. Measure will be used each time a measurement value is queried from the device.

Defining IEEE bus queries always follows the same philosophy:

A line "Count=<cnt>", where <cnt> is an integer of value 0 or 1. "0" means that no command is available, "1" means that a query command is available. "0" can not be used in the Measure section (see above).

A line "GpibLine1=<cmd>" will define which command to send as a query to the device. A waiting time can be defined in the command string as described above.

Additionally, some section specific entries can be used:

A line "GpibResponse1=<resp>" can be defined in the Identify section. The device's answer will be checked to contain the string given by <resp>. Only if this sub-string can be found, the device will be considered as identified and will be set to physical mode.

A line "HeaderOffset=<i>" can be defined in the Measure section. <i> stands for an integer and indicates the number of characters to skip at the beginning of the string returned to the measurement query. The measurement value itself will be the numerical conversion of the string after skipping these offset characters.