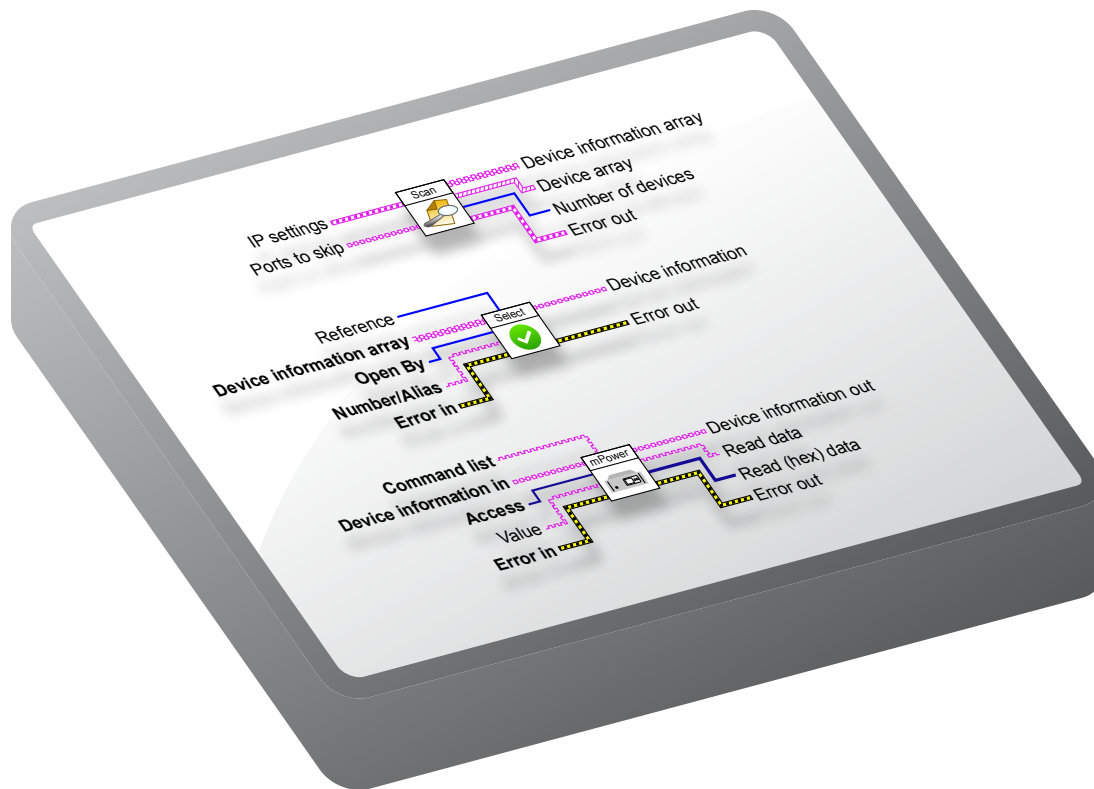




mPower™ DC 3xx Series

Programmable Power Supplies

LabView VI Guide and Reference



Oct 2023 : P/N 501048-DC3LV-C

© 2020, 2022, 2023 Marway Power Systems, Inc. All rights reserved. Some portions © Elektro-Automatik, used with permission.

mPower™, mPower DC™, mPower 3 Series™, mPower 300™, mPower 310™, mPower 320™, mPower 311™, and mPower 411™ are trademarks of Marway Power Systems, Inc. All other trademarks are the property of their respective owners.

Global Support Contacts

Web: www.marway.com
Email: support@marway.com
sales@marway.com
Phone: 800-462-7929 (7am–5pm PST)

There may be updates to this documentation and the software it describes at:
<http://www.marway.com/mpower-dc-power-supplies>

1 Getting Ready

1.1 Applicability	4
1.2 Additional Documentation	4
1.3 Command Timing	4

2 Communication Setup VIs

2.1 Device_scan.vi	5
2.2 Device_select.vi	8
2.3 Device_close.vi	10

3 Series Main VIs

3.1 300 Series PS9000_1U.vi	11
3.2 310 Series PSI9000_2U.vi	11
3.3 310 Series PSI9000_3U.vi	11
3.4 320 Series PSI9000_3U_slave_rear.vi	12
3.5 320 Series PSI9000_3U_slave_front.vi	12
3.6 311 Series PSI10000_4U.vi	12
3.7 411 Series PSB10000_4U.vi	13
3.8 System VI Inputs / Outputs	14

4 Function Generator VIs

4.1 Overview	16
4.2 Commonly used VI parameters	17
4.3 FG_arbitrary_config.vi	18
4.4 FG_arbitrary_load.vi	19
4.5 FG_arbitrary_save.vi	20
4.6 FG_din40839_config.vi	21
4.7 FG_ramp_config.vi	22
4.8 FG_rectangle_config.vi	23
4.9 FG_sine_config.vi	24
4.10 FG_trapezoid_config.vi	25
4.11 FG_triangle_config.vi	26
4.12 FG_xy_fc_calculation.vi	27
4.13 FG_xy_pv_calculation.vi	29
4.14 FG_xy_table_load.vi	31
4.15 FG_xy_pv_en50530_csi.vi	32
4.16 FG_xy_pv_en50530_man.vi	34
4.17 FG_xy_pv_en50530_thin_film.vi	36
4.18 FG_xy_pv_en50530_read_data.vi	38

5 Error Code Reference

1 Getting Ready

1.1 Applicability

This Programming Guide covers a multitude of hardware interfaces and software protocols available across the mPower DC Series of programmable autoranging DC power supplies. Not all features described in this Guide are available on all models of the mPower DC series.

- The mPower 411 Series models have USB, Ethernet, and the flexible AnyBus remote interfaces, which accepts a number of different hardware protocol modules.
- The mPower 311 Series models have USB, Ethernet, and the flexible AnyBus remote interfaces, which accepts a number of different hardware protocol modules.
- The mPower 300 Series models typically have USB and Ethernet remote interfaces. This Series does not have a function generator, so remote programmability for the function generator does not apply.
- The legacy mPower 310 Series models come with either a GPIB interface, or the flexible AnyBus slot, which accepts a number of different hardware protocol modules.
- The legacy mPower 320 Series models come with USB Type B on the rear and front panels. The port on the front panel supports a subset of the ModBus and SCPI commands relative to the rear port.

The reader will have to be aware of exactly which model and hardware interfaces are to be programmed, and therefore which software capabilities are also available. Throughout the Guide, an attempt will be made to try to make it clear which models the discussed features are applicable to. If there is no specific mention of model compatibility, it can be assumed the commands are applicable to all models.

1.2 Additional Documentation

All mPower VI communication is based on the ModBus protocol, and the latter's terminology of registers. The command list is configured as a dropdown, containing the register numbers in front of the commands as cross reference.

The register numbers for each of the mPower Series are documented in separate documents available from our website (<http://www.marway.com/mpower-dc-power-supplies/software>). Additionally, for reference and examples of ModBus in general, the commands, and some of the interface hardware, you will need either the Operating Guides of current models, or the general programming guide *Marway-mPower-DC3-Programming-Guide-PS.pdf* for the legacy models.

1.3 Command Timing

Program code from a PC can send commands to a power supply faster than each command can be processed. In custom scripting, it is the responsibility of the program code to make sure that the power supply is ready for each command (often by inserting intentional time delays between actions). In a LabView VI, it is the VIs responsibility to do the same.




A key factor which governs the amount of time required between commands is that an error may occur, and the VIs have to wait a certain time in order to catch the response.

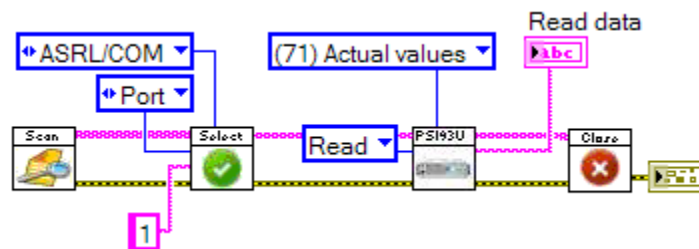
It is recommended to implement a timing rhythm in applications where the main VIs are not repeated faster than every 30 ms.

Furthermore, it is not possible to let two or more VIs run at the same time, even when placed in parallel. At least not with the standard LabView version. This is based on the LabView internal handling. This means that if you place two of the same VIs in parallel, for example in order to control two units in parallel and at the same time, and the processing time of one VI is about 45 ms, then both VIs will use up approximately 90 ms. This definitely applies to VIs which send data over Ethernet/LAN, but may also apply for all VISA interface communication related VIs.

2 Communication Setup VIs

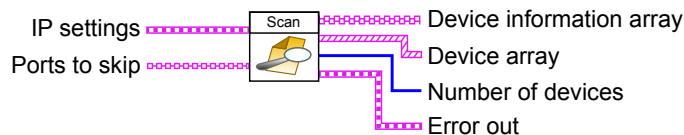
In order to correctly establish communication with an mPower power supply, please follow these steps:

1. Place VI **Device_scan.vi** , which will scan the system for connected, compatible devices.
2. Behind **Device_scan.vi** place VI **Device_select.vi** . This VI selects a specific device for remote control. It is also required to be placed if only one device is connected, or only one was found.
3. Behind **Device_select.vi**, you then place the **device specific VI** for your device series, here for example **PSI9000_3U.vi** for the 310 Series 3U power supply.
4. The communication is terminated by the **Device_close.vi** .



Note that for mPower 310 Series AnyBus RS-232 interfaces, set the baud rate to 115200 for the module as this is the fixed rate defined internally in the VI.

2.1 Device_scan.vi



This VI scans selected hardware ports like RS232, USB, GPIB, Ethernet of the computer for compatible devices.

Note: if Ethernet is not used, the input **IP settings** should be left unconnected in order to minimize scan time.

Important: Read below about input **Ports to skip**.

In the event no compatible device is found, because a wrong IP or IP range was entered, or the USB driver for the USB port is not correctly installed, the VI will return an error.

IP settings

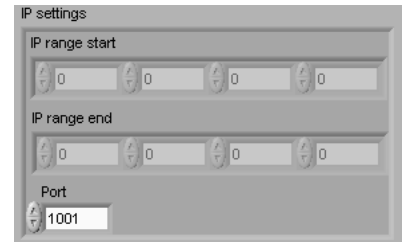
IP settings input is required for detection of devices connected via Ethernet/LAN. If Ethernet is not used, this port can be left blank. The VI will scan one IP or a range of IPs for compatible device. When using only one IP, IP range start and IP range can be the same, but IP range end is not required. When only IP range start is given, only the start IP is scanned.

The IP range may only vary on the fourth number, anything else is not accepted.

The port input has to have the same value as the port setting of the power supply's Ethernet interface. The scan will only detect systems with that port. If several ports were used on the systems, the scan VI has to be run several times with a different port input.

Note: this input does not set the port for the Ethernet interface, it only uses it for communication.

The port can only be changed by accessing the Ethernet web interface by opening the system's IP in a browser, and selecting CONFIGURATION. The port can only be changed there. The port value is stored in the Ethernet module, not in the system.



Ports to skip

This input is important for mixed environments where systems from various manufacturers are connected to one PC. During a normal scan, this VI will open any COM port that is present and send data. For systems other than supported by this VI set, this can cause complications. Therefore, list all ports in this string array which are to be skipped during the scan. Apart from avoiding hanging devices caused by telegrams of an alien protocol, this also decreases the time used for a scan.

The port names have to be entered in typical LabView format, like COM3 (alias) or ASRL3::INSTR, for example.



Device information array

Device information array is a two-dimensional array containing information required for the other VIs. This output is normally connected to the corresponding input of VI **Device_select.vi**, which will then select one of the found, compatible devices. The indexed array is then output by **Device_select.vi** at output **Device information**. The indexed array contains the following elements:

Index	Name	Description
0	COM Part	Port type to which the system is connected as VISA resource string ¹
1	Type	System Device Type (see Appendix)
2	Serial Number	System Serial number
3	Article Number	System Article number (see Appendix)
4	(not used)	(not used)
5	Nominal voltage	Nominal voltage (e.g. 80, 200, 360, 500, 750)
6	Nominal current	Nominal current in amps (e.g. 12, 25, 30, 50, 100)
7	Nominal power	Nominal power in watts (e.g. 1500, 3000)
8	(not used)	(not used)
9	Nominal Resistance	Max. internal resistance (310 Series PSI9000_nU.vi only)
10	Class	System Device Class (see Appendix)
11	Port description	For example "USB Serial" (similar to Windows device manager)
12	Port init	
13	Interface type	Name of interface card
14	Interface software version	Firmware version of an IF-XX interface card
15	Interface article number	Article number of an IF-XX interface card

16	Interface serial number	Serial number of an IF-XX interface card
17	Device software version (KE)	Firmware version of KE component (see Operating Guide block diagram)
18	Device software version (HMI)	Firmware version of HMI component (see Operating Guide block diagram)
19	Device software version (DR)	Firmware version of DR component (see Operating Guide block diagram)
20	Protocol	Always MODBUS_RTU for with Modbus via USB or Ethernet or Anybus

Device array




Device array is a cluster array which contains information about the detected devices.

Number of devices

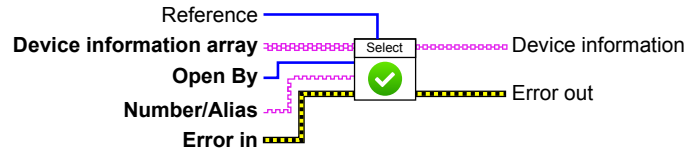
Returns the number of detected, compatible systems.

Error out

Error out contains error information. If **Error in** contains error information, it is led through to **Error out**. Alternatively, it contains the error status of the VI. For more information about the error, right-click the display element **Error out** and choose “Explain error” from the context menu.

Field	Description
	Status is TRUE (red cross) if an error occurs, or FALSE (green hook) if not. Default is FALSE .
	Code is the error or warning code. If Status is TRUE , then Code contains an error code unequal to zero. If Status is FALSE , then Code will be zero or contains a warning code.
	Source in most cases, will be the name of the VI, or the name of the function which caused the error.

2.2 Device_select.vi



This VI selects one of the detected, compatible systems connected to the PC for further control. It can be placed multiple times in parallel to control several systems at once. The selection is done by one of the following:

- Serial number
- Type of hardware port

The inputs **Reference**, **Open by**, and **Number/Alias** depend on each other, though **Open by** has priority.

When Open by = Port

A system is selected via a certain hardware interface. For input **Reference**, the type string **ASRL/COM**, **GPIB**, or **TCPIP** is selected.

The input **Number/Alias** will then define the alias name for serial ports, such as COM1, or the GPIB address, or the IP address for TCPIP.

When Open by = Serial number

A system is selected by its serial number. This only required input **Number/Alias**, which defines the serial number of the particular device as a string. This number is compared to the ones of the found devices and if found, the device is opened. Otherwise, an error is returned.

Device information array

Device information array is a two-dimensional array containing information about the found systems. This input has to be connected to the corresponding output of VI **Device_scan.vi**. **Device_select.vi** selects one of the detected, compatible systems, and outputs it to **Device information**.

Open by

Open by defines the way how a device is opened—offering the choices of: Serial number, Port.

Reference

Reference is only used and required if **Open by** = Port was chosen. The desired hardware interface is selected here (GPIB, ASRL, or TCPIP), defined by input **Number/Alias**.

Example: You want to select the device with the GPIB port GPIB0::1:INSTR from the **Device information array**. You would then select **Open by** = Port, **Reference** = GPIB, and in the **Number/Alias** you enter its GPIB address.

Number/Alias

This input is used to define an alias for serial ports, like for example COM5, or a GPIB address number, or an IP address like 192.168.2.1, depending on what was selected by priority input **Open by**.



Device information

This output contains the device information of the device that was selected from **Device information array**. See above.



Error out

Error out contains error information. If **Error in** contains error information, it is led through to **Error out**. Alternatively, it contains the error status of the VI. For more information about the error, right-click the display element **Error out** and choose “Explain error” from the context menu.

Field	Description
	Status is TRUE (red cross) if an error occurs, or FALSE (green hook) if not. Default is FALSE .
	Code is the error or warning code. If Status is TRUE , then Code contains an error code unequal to zero. If Status is FALSE , then Code will be zero or contains a warning code.
	Source in most cases, will be the name of the VI, or the name of the function which caused the error.

2.3 Device_close.vi






This VI closes the communication port that was used with the device specified in **Device information in**.

Device information in

Device information in is a string array containing information required to control the device. It is usually connected to the output of the main VI. (See the VI for your mPower model for the information array details.)

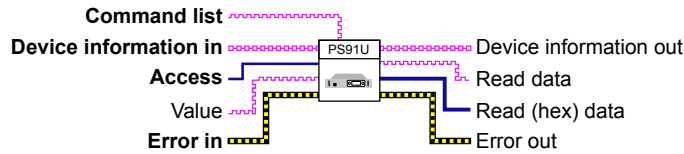
Error out

Error out contains error information. If **Error in** contains error information, it is led through to **Error out**. Alternatively, it contains the error status of the VI. For more information about the error, right-click the display element **Error out** and choose “Explain error” from the context menu.

Field	Description
	Status is TRUE (red cross) if an error occurs, or FALSE (green hook) if not. Default is FALSE .
	Code is the error or warning code. If Status is TRUE , then Code contains an error code unequal to zero. If Status is FALSE , then Code will be zero or contains a warning code.
	Source in most cases, will be the name of the VI, or the name of the function which caused the error.

3 Series Main VIs

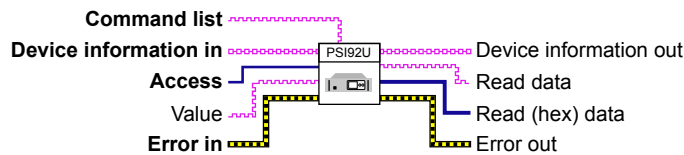
3.1 300 Series PS9000_1U.vi



Supported interfaces: USB, Ethernet

This VI is for the mPower 300 Series 1U power supplies. See “3.7 411 Series PSB10000_4U.vi” on page 13 for details.

3.2 310 Series PSI9000_2U.vi

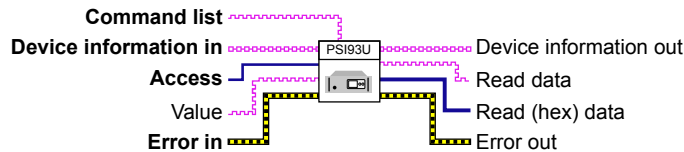


Supported interfaces: USB (rear), Ethernet, ModBus TCP, RS-232, GPIB

This VI supports the mPower 310 Series 2U systems. See “3.7 411 Series PSB10000_4U.vi” on page 13 for details.

The 310 Series includes a function generator, and the PSI9000_2U.vi contains commands for the function generator only for run control. All other function generator control is managed through a series of VIs with each dedicated to a particular function type. Additionally, the FG_xy_table_load.vi which would be used to load table data from CSV files. These are documented later in this Guide.

3.3 310 Series PSI9000_3U.vi

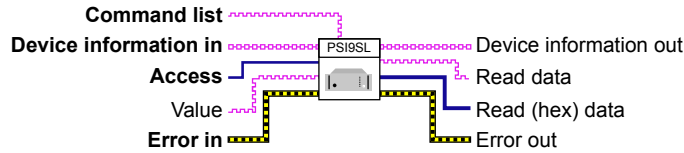


Supported interfaces: USB (rear), Ethernet, ModBus TCP, RS-232, GPIB

This VI supports the mPower 310 Series 3U systems. See “3.7 411 Series PSB10000_4U.vi” on page 13 for details.

The 310 Series includes a function generator, and the PSI9000_3U.vi contains commands for the function generator only for run control. All other function generator control is managed through a series of VIs with each dedicated to a particular function type. Additionally, the FG_xy_table_load.vi which would be used to load table data from CSV files. These are documented later in this Guide.

3.4 320 Series PSI9000_3U_slave_rear.vi



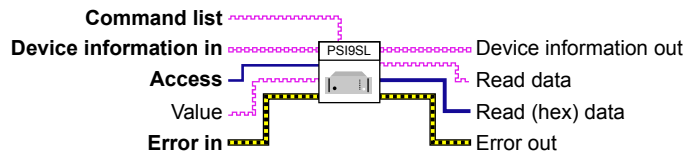
Supported interfaces: USB (rear panel only)

This VI supports the mPower 320 Series slave systems. See “3.7 411 Series PSB10000_4U.vi” on page 13 for details.

These systems usually run as slaves in a customized master-slave setup with a 310 Series master unit, and one or more 320 Series slaves. Technically, a 320 Series unit can be used as a master, but would need configured remotely to do so (and it would not have a function generator). See the Operating Guide for more information if using the 320 unit as a master is of interest. Remote control of a 320 unit is only possible in stand-alone operation, or when the unit is going to run as master. Once configured as a slave, the 320 unit cannot be individually remote controlled.

Systems of this series have an additional Type B USB port on the front for quick access and configuration. This port has a reduced set of commands, so there are two different VIs for the front USB and the rear USB port. The rear USB port offers the full set of commands. The corresponding command set is connected to a ModBus register list as reference.

3.5 320 Series PSI9000_3U_slave_front.vi

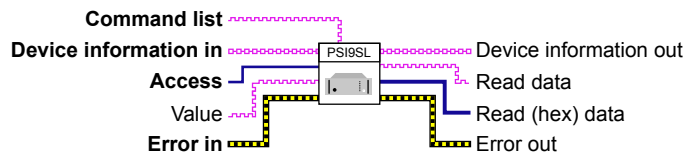


Supported interfaces: USB (front panel only)

This VI supports the mPower 320 Series slave systems, and in particular, the limited command set of the front USB.

See “3.7 411 Series PSB10000_4U.vi” on page 13 for details.

3.6 311 Series PSI10000_4U.vi

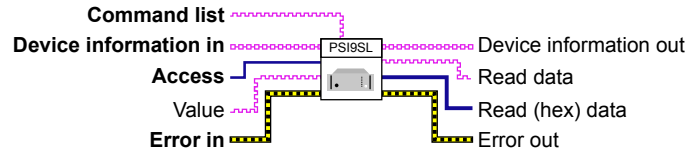


Supported interfaces: USB (rear), Ethernet, ModBus TCP, RS-232

This VI supports the mPower 311 Series 4U systems. See “3.7 411 Series PSB10000_4U.vi” on page 13 for details. It can be used for the 3U and 2U systems as well (as there’s no dedicated VIs for those as of this writing).

The 311 Series includes a function generator, and the PSI10000_4U.vi contains commands for the function generator only for run control. All other function generator control is managed through a series of VIs with each dedicated to a particular function type. Additionally, the FG_xy_table_load.vi which would be used to load table data from CSV files. These are documented later in this Guide.

3.7 411 Series PSB10000_4U.vi



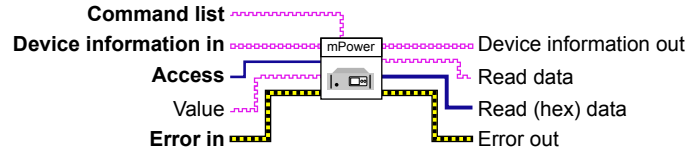
Supported interfaces: USB (rear), Ethernet, ModBus TCP, RS-232

This VI supports the mPower 411 Series 4U systems. See “[3.7 411 Series PSB10000_4U.vi](#)” on page 13 for details. It can be used for the 3U and 2U systems as well (as there’s no dedicated VIs for those as of this writing).

The 411 Series includes a function generator, and the PSB10000_4U.vi contains commands for the function generator only for run control. All other function generator control is managed through a series of VIs with each dedicated to a particular function type. Additionally, the FG_xy_table_load.vi which would be used to load table data from CSV files. These are documented later in this Guide.

3.8 System VI Inputs / Outputs

For all models, the pattern of the VI connections are the same. This section explains the inputs and outputs.



abc Device information in

Device information in is a string array containing information required to control the device.

Index	Description
0	Communication port as VISA resource string ¹
1	Device type
2	Serial number
3	Article number
5	Nominal voltage (e.g. 80, 200, 360, 500, 750)
6	Nominal current in amps (e.g. 12, 25, 30, 50, 100)
7	Nominal power in watts (e.g. 1500, 3000)
10	Device class
20	Fixed parameter, always: MODBUS_RTU

¹ <http://zone.ni.com/reference/en-XX/help/370131S-01/ni-visa/visaresourcesyntaxandexamples/>

Command list

Command list includes all the features required to control an mPower 300 Series 1U system. Select the desired command from the list. Refer to the ModBus register list for more information about the commands.

Access

The **Access** input defines whether the command selected from **Command list** is **Read** or **Write**. If the command does not support writing, then an error will occur.

Value

Value is any string or value which is to be sent with the command selected by **Command list**. Values are only required when using **Access** mode **Write**. The VI uses strings, similar to SCPI language, and understands hexadecimal values. For example, a command from the register list is described to have the parameters 0x0000 (OFF) and 0xFF00 (ON). Either string of "0x0000" or "OFF" can be used as the value. If there is more than one parameter required, each value is separated by semicolons (no spaces).

Read data




Read data returns the data requested by the last read command. If the last command was a write command, then “OK” is returned. Returned data will be formatted by the VI to a readable string. For example, if the system returns the alarm status OVP in the form of a bit, the VI will output “OVP” as a string.

Read data (hex)

Same data as **Read data** above, but as a hexadecimal byte array.

Error out

Error out contains error information. If **Error in** contains error information, it is led through to **Error out**. Alternatively, it contains the error status of the VI. For more information about the error, right-click the display element **Error out** and choose “Explain error” from the context menu.

Field	Description
	Status is TRUE (red cross) if an error occurs, or FALSE (green hook) if not. Default is FALSE .
	Code is the error or warning code. If Status is TRUE , then Code contains an error code unequal to zero. If Status is FALSE , then Code will be zero or contains a warning code.
	Source in most cases, will be the name of the VI, or the name of the function which caused the error.

4 Function Generator VIs

4.1 Overview

The function generator is a feature available in the mPower 310 Series power supplies. These VIs are compatible with that series of systems. (The 300 Series and 320 Series do not include a function generator.)

In remote control, the VIs provide the same functionality as when manually configuring and operating the function generator. However, there are some minor differences. For example, the 99 sequence points of the arbitrary generator can't be supplied by a LabView array, and must be loaded from a CSV file. Using CSV files generated from MS Excel or text editors is more efficient anyway, and loading files offers the ability to do fast switching between several sequences.

The function generator VIs described here are only for configuration or calculating/loading table data. The run control commands for starting, stopping or exiting the function generator is integrated into the series main VIs (e.g. 310 Series PSI9000_2U and 310 Series PSI9000_3U VIs).

To prepare for using the function generator using these VIs, follow this process:

- First, switch the device into remote control.
- Then use a configuration VI to configure the desired function generator (XY or arbitrary). This also selects whether the function will be applied to voltage or current.
- Next, use a load data VI to complete the setup.
- Finally, use the main VI to control the function generator (start, stop, exit).

The following function generator VIs are available:

VI Name	VI Purpose
FG_arbitrary_config.vi	For the configuration of the arbitrary generator
FG_arbitrary_load.vi	For loading the 99 arbitrary generator sequence points from a CSV file
FG_arbitrary_save.vi	For saving the 99 arbitrary generator sequence points into a CSV file
FG_din40839_config.vi	For the configuration of the DIN40839
FG_ramp_config.vi	For the configuration of the function Ramp
FG_rectangle_config.vi	For the configuration of the function Rectangle
FG_sine_config.vi	For the configuration of the function Sine
FG_trapezoid_config.vi	For the configuration of the function Trapezoid
FG_triangle_config.vi	For the configuration of the function Triangle
FG_xy_table_load.vi	For loading of a table with exactly 4096 values from a CSV file
FG_xy_fc_calculation.vi	For calculation of XY table data for a simulated fuel cell
FG_xy_pv_calculation.vi	For calculation of XY table data for a simulated solar panel
FG_xy_pv_EN50530_csi.vi	For calculation of XY table data for a simulated solar panel according to standard EN 50530 and panel technology "cSi". The VI is also used to load the day trend curve from a file.
FG_xy_pv_EN50530_man.vi	For calculation of XY table data for a simulated solar panel according to standard EN 50530, where all available parameters of all modes and technologies are freely adjustable within their limits. The VI is also used to load the day trend curve from a file.
FG_xy_pv_EN50530_read_data.vi	Used to read data from the device, recording during the solar panel simulation.

FG_xy_pv_EN50530_thin_film.vi	For calculation of XY table data for a simulated solar panel according to standard EN 50530 and panel technology "thin film". The VI is also used to load the day trend curve from a file.
-------------------------------	--

4.2 Commonly used VI parameters

These parameters below are used in many of the VIs. Instead of repeating their documentation, the details can be found here, and subsequent VI sections will refer to these sections.

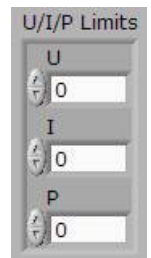
Device information in

Device information in is a string array containing information required by the VI. This input is to be connected to **Device information out** of the corresponding series main VI, or another function generator related VI.

U/I/P Limits




This input requires that you define limits for values on the dc output of your device which are not affected by the function generator. These limits correspond to the set values of voltage (U), current (I) and power (P) the same as you can adjust them when controlling the system manually or remotely in normal operation. Any limit value which is governed by the function generator operation will be ignored. (A voltage limit of 45 will be ignored if the function generator has an instruction to set it to 60.)

Don't leave values at zero. For example, let's say you want to apply a sine wave function to the dc output current. Using the **Limits** input you still have to define the values for voltage (U) and power (P), otherwise the load might not consume any power, because the power would have been limited to 0.

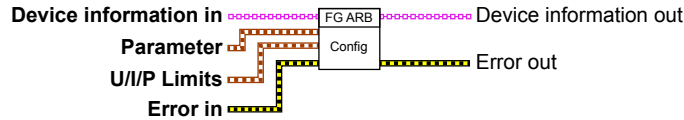


Error out

Error out contains error information. If **Error in** contains error information, it is led through to **Error out**. Alternatively, it contains the error status of the VI. For more information about the error, right-click the display element **Error out** and choose "Explain error" from the context menu.

Field	Description
	Status is TRUE (red cross) if an error occurs, or FALSE (green hook) if not. Default is FALSE .
	Code is the error or warning code. If Status is TRUE , then Code contains an error code unequal to zero. If Status is FALSE , then Code will be zero or contains a warning code.
	Source in most cases, will be the name of the VI, or the name of the function which caused the error.

4.3 FG_arbitrary_config.vi



This VI is for the configuration of the arbitrary function generator. There are separate VIs for loading and saving a sequence, and a series main VI is used to control the running of the function generator after it is set up.

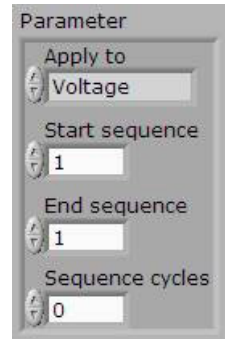
Device information in

See page 17.

Parameter

The **Parameter** variable is a cluster input requiring following elements which are named similar to the settings you can manually configure on the device:

- **Apply to** defines whether the arbitrary generator applies the function to the dc voltage (U) or the dc current (I) on the output of the system.
- **Start sequence** defines which of the 100 available sequences will be the starting point for the function run. It means, you can build sequence blocks of various lengths. When using only one sequence, the start sequence value has to be equal to the end sequence value. Keep in mind that Start sequence \leq End sequence.
- **End sequence** defines, which of the 100 available sequences will be the end point for the function run. It means, you can build sequence block of various length. When using only one sequence, the end sequence value has to be equal to the start sequence value. Keep in mind that End sequence \geq Start sequence.
- **Sequence cycles** defines the number of cycles (i. e. repetitions) the sequence block, defined by Start sequence and End sequence, is run through. 0 = Infinite cycles to run until stopped by user. 1–999 = Finite number of cycles to run, and the function generator will stop automatically when done.



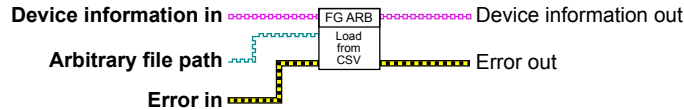
U/I/P Limits

See page 17.

Error out

See page 17.

4.4 FG_arbitrary_load.vi



This VI is for the loading of CSV file data into the arbitrary function generator. It must be located and executed *after* the **FG_arbitrary_config** VI. The format of the CSV file, and the file name must follow the specifications described below. The loading will be aborted if there's a file name or file format errors, or values in the file that don't match the specifications of the power supply system. **Error out** will be populated.

Device information in

See page 17.

Arbitrary file path

This is the full path and file name to a *.csv file, which must follow a certain file name pattern:

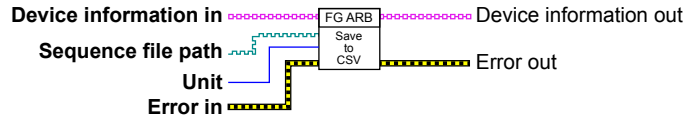
- The file name must either begin with **WAVE_U** or **WAVE_I** (upper or lower case).
- **WAVE_U** causes the VI to interpret values in the file as voltage (arbitrary mode U), and the file is checked against the system's nominal voltage rating.
- **WAVE_I** causes the VI to interpret values in the file as current (arbitrary mode I), and the file is checked against the system's nominal current rating.

The file must be of type CSV (simple text file, values separated by semicolons according to Excel style). The format of the data is described in the mPower DC Power Supplies Operating Guide (in the chapter about the function generator).

Error out

See page 17.

4.5 FG_arbitrary_save.vi



This VI is for the saving CSV file data from the arbitrary function generator. The sequence data can come from two sources, either they were previously loaded into the device with **FG_arbitrary_load.vi**, or they are from the control panel of the power supply (where you can manually edit the sequence). There are two important things to know:

- Data which has been loaded into the power supply through **FG_arbitrary_load.vi** is only temporary. It is not stored in the system. If it has been altered after being loaded, it must be saved (assuming you want the changes saved).
- Data entered through the control panel has to be manually loaded (by accessing the function generator controls in the control panel menus) before saving it with this VI.

Device information in

See page 17.

Sequence file path

This is the full path and file name to a *.csv file to save the sequence data to.

The created file will be of type CSV (simple text file, values separated by semicolons according to Excel style). The format of the data is described in the mPower DC Power Supplies Operating Guide (in the chapter about the function generator).

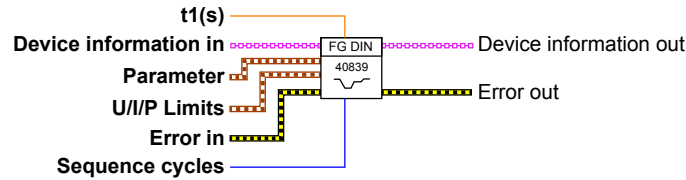
Unit

Unit defines whether the arbitrary sequence data to be read from the system was previously loaded for voltage or current. It is required to select the correct unit, so the saved values can be translated to proper real values from the internal percent (of nominal value) format. (See the “Hex Percent and Decimal Value Conversion” topic in the *Programming Guide*.)

Error out

See page 17.

4.6 FG_din40839_config.vi



This VI defines sequence parameters for the power supply, which are then used by the DIN 40839 function to emulate the battery voltage progression during a car engine start according to German standard DIN 40839. This function implies that it can only be applied to the DC output voltage of the power supply.

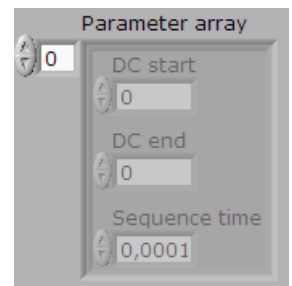
In case the DIN curve shall contain a section where a sinusoidal voltage progression is required, this can't be achieved with this VI, because it uses only linear voltage progression based upon the arbitrary generator using ramps. To achieve a modified curve, single sequences of the arbitrary generator can be used.

abc Device information in — See page 17.

EST Parameter array

This input requires an array of five (5) clusters. These five clusters each represent a sequence, building the complete DIN curve. See also the Operating Guide for an overview, and the assignment of sequences to sections of the DIN curve.

- **DC start:** Starting voltage at the begin of a sequence. Adjustable range: 0–Unom. Note that **DC start** has to be the same value as **DC end** of the previous sequence (except for the first sequence).
- **DC end:** Final voltage at the end of a sequence. Adjustable range: 0–Unom. Note that **DC end** has to be the same value as **DC start** of the next sequence (except for the last sequence).
- **Sequence time:** defines the time in seconds for the voltage progression, i.e. ramp between **DC start** and **DC end**. Adjustable range is 0.0001s–36000s.



EST U/I/P Limits — See page 17.

DBL t1

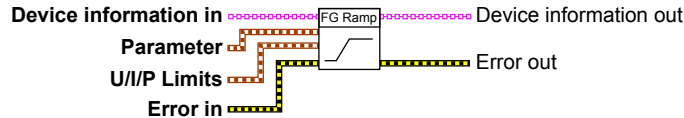
The t1 time is the delay at the end of a cycle before the next cycle starts. Adjustable range is 0.0001s–36000s.

U16 Sequence cycles

Defines the number of cycles (repetitions) the complete DIN curve is run through. 0 = Infinite cycles to run until stopped by user. 1–999 = Finite number of cycles to run, after which the function generator will stop automatically.

EST Error out — See page 17.

4.7 FG_ramp_config.vi



This VI is for defining sequence parameters for the power supply, which are then used by the ramp function.

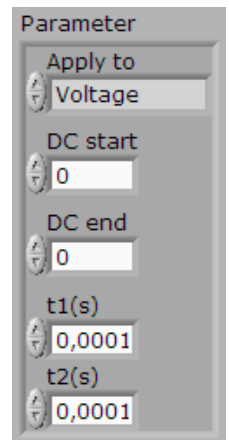
Device information in

See page 17.

Parameter

The **Parameter** variable is a cluster input requiring following elements which are named similar to the settings you can manually configure on the system's control panel:

- **Apply to** defines whether the arbitrary generator applies the function to the dc voltage (U) or the dc current (I) on the output of the system.
- **DC start** is the desired voltage at the start of the ramp. The adjustable range is 0–Unom or 0–Inom. Keep in mind that to achieve a rising ramp, **DC start** has to be lower than **DC end**. Vice versa, when wanting to achieve a falling ramp.
- **DC end** is the desired voltage at the end of the ramp. The adjustable range is 0–Unom or 0–Inom. Keep in mind that to achieve a rising ramp, **DC start** has to be lower than **DC end**. Vice versa, when wanting to achieve a falling ramp.
- **t1** defines the time in seconds to elapse before the start of the ramp, in order to have a pause before the next repetition. If t1 is set to the minimum value, the resulting generated signal would be a sawtooth pattern. Adjustable range os 0.0001–36000 seconds.
- **t2** defines the time in seconds for the voltage progression—the ramp between DC start and DC end. Adjustable range os 0.0001–36000 seconds.



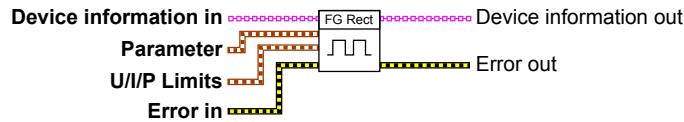
U/I/P Limits

See page 17.

Error out

See page 17.

4.8 FG_rectangle_config.vi



This VI is for defining sequence parameters for the power supply, which are then used by the rectangle function.

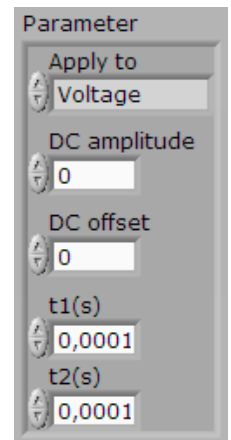
Device information in

See page 17.

Parameter

The **Parameter** variable is a cluster input requiring following elements which are named similar to the settings you can manually configure on the system's control panel:

- **Apply to** defines whether the arbitrary generator applies the function to the dc voltage (U) or the dc current (I) on the output of the system.
- **DC amplitude** is the desired amplitude of the generated rectangular signal, in either amps or volts. The adjustable range is DC offset–Unom or DC offset–Inom. The amplitude of the signal adds to the offset, so the resulting voltage or current the function is applied to is DC amplitude + DC offset.
- **DC offset** is the desired offset from zero for the generated rectangular signal, in either amps or volts. The adjustable range is 0–(Unom – DC amplitude) or 0–(Inom – DC amplitude). The offset shifts the generated signal positively on the Y axis.
- **t1** defines the time in seconds for the amplitude being low (i.e. the generated signal is at DC offset). The adjustable range is 0.0001–36000 seconds.
- **t2** defines the time in seconds for the amplitude being high (i.e. the generated signal is at DC amplitude). The adjustable range is 0.0001–36000 seconds.



Adding t1 and t2 results in the period of the rectangular signal or frequency as $f = 1/(t1+t2)$. With both time values being separately adjustable, the duty cycle can be varied.

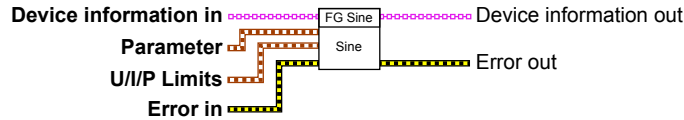
U/I/P Limits

See page 17.

Error out

See page 17.

4.9 FG_sine_config.vi



This VI is for defining sequence parameters for the power supply, which are then used by the sine function.

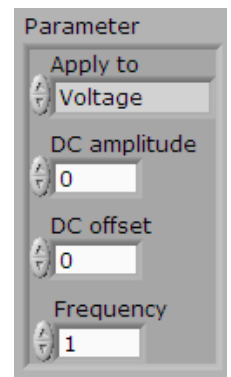
Device information in

See page 17.

Parameter

The **Parameter** variable is a cluster input requiring following elements which are named similar to the settings you can manually configure on the system's control panel:

- **Apply to** defines whether the arbitrary generator applies the function to the dc voltage (U) or the dc current (I) on the output of the system.
- **DC amplitude** is the desired amplitude of the generated sine wave, in either amps or volts. The adjustable range is 0–**DC offset**. (The **DC offset** is the center line of the DC sine wave signal, thus the **DC amplitude** can and must never be higher than the **DC offset**.)
- **DC offset** is the required offset from zero for the generated sine wave, in either amps or volts. The adjustable range is 0–(Unom – DC amplitude) or 0–(Inom – DC amplitude). The offset shifts the generated signal positively on the Y axis. Value **DC offset** always has to be equal to or bigger than **DC amplitude**.
- **Frequency** defines the frequency of the generated sine wave signal (integer values, 1–1000 Hz).



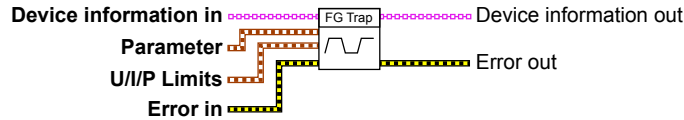
U/I/P Limits

See page 17.

Error out

See page 17.

4.10 FG_trapezoid_config.vi



This VI is for defining sequence parameters for the power supply, which are then used by the trapezoid function.

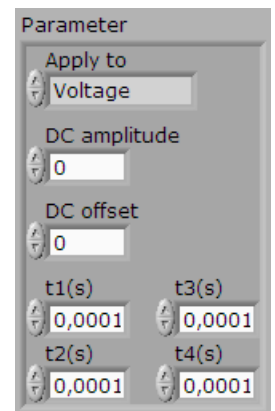
Device information in

See page 17.

Parameter

The **Parameter** variable is a cluster input requiring following elements which are named similar to the settings you can manually configure on the system's control panel:

- **Apply to** defines whether the arbitrary generator applies the function to the dc voltage (U) or the dc current (I) on the output of the system.
- **DC amplitude** is the desired top level of the generated trapezoid signal, in either amps or volts. The adjustable range is DC offset–Unom or DC offset–Inom. The amplitude of the signal adds to the offset, so the resulting voltage or current the function is applied to is DC amplitude + DC offset.
- **DC offset** is the desired offset from zero for the generated trapezoid signal, in either amps or volts. The adjustable range is 0–(Unom – DC amplitude) or 0–(Inom – DC amplitude). The offset shifts the generated signal positively on the Y axis.
- **t1** defines the rising ramp/edge time (range is 0.0001–36000 seconds).
- **t2** defines the duration of the top level time (range is 0.0001–36000 seconds).
- **t3** defines the falling ramp/edge time (range is 0.0001–36000 seconds).
- **t4** defines the duration of the low level time (range is 0.0001–36000 seconds).



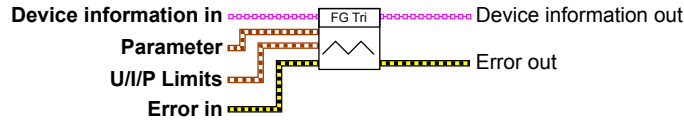
U/I/P Limits

See page 17.

Error out

See page 17.

4.11 FG_triangle_config.vi



This VI is for defining sequence parameters for the power supply, which are then used by the triangle function.

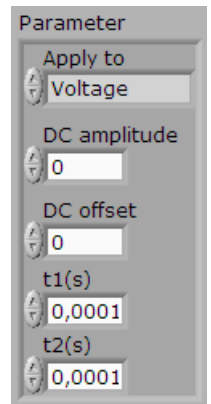
Device information in

See page 17.

Parameter

The **Parameter** variable is a cluster input requiring following elements which are named similar to the settings you can manually configure on the system's control panel:

- **Apply to** defines whether the arbitrary generator applies the function to the dc voltage (U) or the dc current (I) on the output of the system.
- **DC amplitude** is the desired top level of the generated triangle signal, in either amps or volts. The adjustable range is DC offset–Unom or DC offset–Inom. The amplitude of the signal adds to the offset, so the resulting voltage or current the function is applied to is DC amplitude + DC offset.
- **DC offset** is the desired offset from zero for the generated triangle signal, in either amps or volts. The adjustable range is 0–(Unom – DC amplitude) or 0–(Inom – DC amplitude). The offset shifts the generated signal positively on the Y axis.
- **t1** defines the rising ramp/edge time (range is 0.0001–36000 seconds).
- **t2** defines the falling ramp/edge time (range is 0.0001–36000 seconds).



Adding **t1** and **t2** results in the period of the triangular signal or frequency as $f = 1/(t1+t2)$.

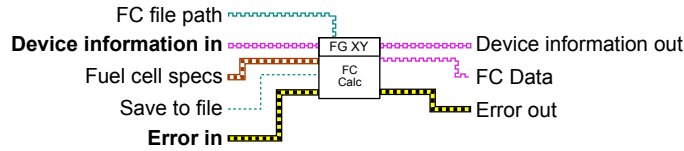
U/I/P Limits

See page 17.

Error out

See page 17.

4.12 FG_xy_fc_calculation.vi



This VI is used to calculate Fuel Cell XY table data for use with the XY function generator. It requires entering four set points which are distributed along the typical curve of a fuel cell. Alternatively, the table data can be calculated externally, by any means, and loaded with the `FG_xy_load.vi`. Table data generated using the VI can be saved using `FG_xy_load.vi` and re-loaded later.

Device information in

See page 17.

Save to file

`TRUE` enables saving the calculated table data to a CSV file, which has to be specified with input `FC file path`.

FC file path

This input defines the path where to save the calculated FC curve table data. If a file name is given, it must use the naming conventions for FC table files as described in the Operating Guide of the power supply. Existing files of the same name will be overwritten. If only a target folder is given, the VI will automatically create files with the convention: `FC_<date>_<time>.csv`.

If input `Save to file` is `TRUE` the data will be stored into a file, in parallel to providing it on the output `FC data` which can be used to directly submit the data to the `FG_xy_load.vi`.

Fuel cell specs

This input is used to enter four setpoints. The first point is at 0% (open circuit) position, therefore the current automatically starts at zero. Similarly, but vice versa, the last point is at 100% (short-circuit), and the voltage is fixed at zero. The other two points require to set two values.

The range for every value is 0 to nominal rating for the power supply, but not every combination is valid. If the calculation results are erroneous, the VI will abort and return error 9971. The calculation doesn't take into account the maximum power of the device. So, a U (Point 3) of 60 V, and an I (Point 3) of 400 A for a 15 kW model would be acceptable inputs, but the power supply couldn't run that curve 1:1 (it would require 24 kW).

- **Point 1:** Uoc (open circuit voltage). Note: Uoc must be greater than U (Point 2) and U (Point 3)
- **Point 2:** U (Point 2) & I (Point 2). Note: U (Point 2) must be greater than U (Point 3), I (Point 2) must be smaller than I (Point 3)
- **Point 3:** U (Point 3) & I (Point 3). Note: U (Point 3) must be smaller than U (Point 2), I (Point 3) must be smaller than Isc
- **Point 4:** Isc (short-circuit current). Note: Isc must be greater than I (Point 3) and I (Point 2)

Label	Value
Uoc	0
U (Point 2)	0
I (Point 2)	0
U (Point 3)	0
I (Point 3)	0
Isc	0

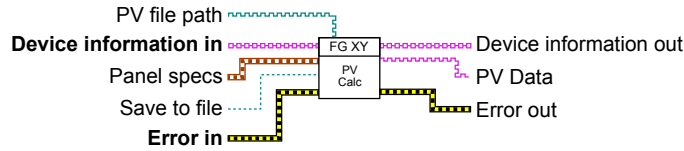
FC data

Contains the calculated FC curve table data (4096 values for 0-125% Unom) as a string for direct feed to `FG_xy_load.vi`, which will load the XY generator with the data.

Error out

See page 17.

4.13 FG_xy_pv_calculation.vi



This VI is used to calculate Fuel Cell XY table data for use with the XY function generator. It requires entering four typical technical specification values of solar panels. Alternatively, the table data can be calculated externally, by any means, and loaded with the **FG_xy_load.vi**. Table data generated using the VI can be saved using **FG_xy_load.vi** and re-loaded later.

Device information in

See page 17.

Save to file

TRUE enables saving the calculated table data to a CSV file, which has to be specified with input **PV file path**.

PV file path

This input defines the path where to save the calculated PV table data. If a file name is given, it must use the naming conventions for PV table files as described in the Operating Guide of the power supply. Existing files of the same name will be overwritten. If only a target folder is given, the VI will automatically create files with the convention:

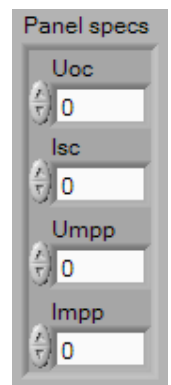
PV_<date>_<time>.csv.

If input **Save to file** is **TRUE** the data will be stored into a file, in parallel to providing it on the output **PV data** which can be used to directly submit the data to the **FG_xy_load.vi**.

Panel specs

This input is used you to enter four values which are typical for solar panels. They are used for the calculation of the PV curve table data, and later for the panel simulation by the power supply. The values must be sensible in relation to the power supply rating. For example, when trying to simulate a 24 V solar panel with a 1500 V power supply the results will not be as expected. It is better to use the simulation within 50–100% of the rated output voltage of the device.

- **Uoc** = Open circuit voltage of the panel (max. panel voltage)
- **Isc** = Short-circuit current (max. panel current)
- **Umpp** = Voltage in the Maximum Power Point
- **Impp** = Current in the Maximum Power Point



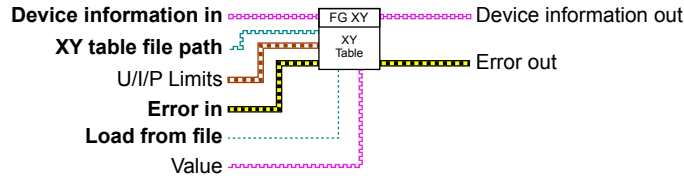
PV data

Contains the calculated PV curve table data (4096 values for 0-125% Unom) as a string for direct feed to `FG_xy_load.vi`, which will load the XY generator with the data.

Error out


See page 17.


4.14 FG_xy_table_load.vi




This VI supports the XY function generator. It is used to load sequence data for the XY function generator from a file of type CSV, or directly from another VI which calculates table data such as `FG_xy_pv_calculation.vi` and `FG_xy_fc_calculation.vi`.

The format of the CSV file, and the file name must follow the specifications described below. The loading will be aborted if there's a file name or file format errors, or values in the file that don't match the specifications of the power supply system. **Error out** will be populated.

 **Device information in** — See page 17.

 **U/I/P Limits** — See page 17.

 **Load from file**

When **TRUE**, loads table data from the CSV file specified by input **XY table file path**. When **FALSE** loads data directly from input **Value** which has previously calculated data from `FG_xy_pv_calculation.vi` or `FG_xy_fc_calculation.vi`.

 **XY table file path**

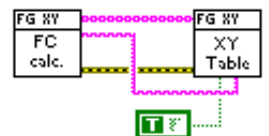
This is the full path and file name to a *.csv file, which must follow a certain file name pattern:


- The file name must either begin with **UI** or **IU** (upper or lower case).
- **UI** causes the VI to interpret values in the file as voltage (which are checked against the system's nominal voltage rating). The values are used as a function of $U = f(I)$.
- **IU** causes the VI to interpret values in the file as current (which are checked against the system's nominal current rating). The values are used as a function of $I = f(U)$.

The file must be of type CSV (simple text file, values separated by semicolons according to Excel style). The format of the data is described in the mPower DC Power Supplies Operating Guide (in the chapter about the function generator).

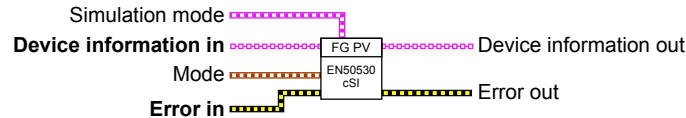
 **Value**

Serves to feed XY table data directly from one of the calculation VIs such as `FC_xy_pv_calculation.vi` and `FC_xy_fc_calculation.vi`. Input **Value** is simply connected to the corresponding output of the calculation VI. The input only accepts data if input **Load from file** is set to **FALSE**.



 **Error out** — See page 17.

4.15 FG_xy_pv_en50530_csi.vi



This VI is used to configure c-SI panel technology of the extended photovoltaics function for solar panel simulation according to EN 50530. Unlike the regular PV simulation, the table data is calculated by the power supply. This allows the extended PV simulation to permanently re-calculate the XY table during runtime, and to record simulation data. The configuration requires entering a specific set of technical properties of solar panels. All parameters required for c-SI technology, such as FFu etc., are rigid, and defined by the standard.

Refer to the standard EN 50530 for definition and additional information about the parameters, and refer to the ModBus register list of the power supply for the mPower-specific adjustable ranges of these parameters.

Device information in

See page 17.

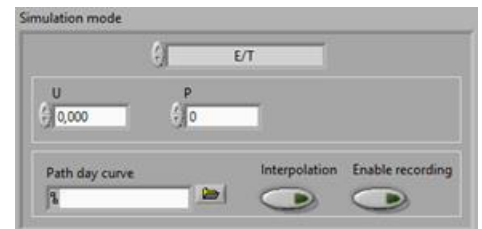
Simulation mode

This input is used to select the simulation mode between:

- **E/T** = Continuous simulation with adjustable temperature and irradiation
- **U/I** = Continuous simulation with adjustable MPP values (Umpp, Impp)
- **DAY U/I** = Day trend mode for U/I
- **DAY E/T** = Day trend mode for E/T

Necessary parameters:

- **Path day curve** = contains the path to the day curve (max. 10000 points) to load for modes DAY U/I and DAY E/T.
- **U** = Set value of voltage (same as single set value command) to limit the output voltage during simulation; this is usually set to the open circuit voltage (Uoc) of the panel
- **P** = Set value of power (same as single set value command) to limit the output power during simulation



Optional parameters:

- **Interpolation** = can be activated to achieve a smoother trend between two points on the loaded day curve. The device would then calculate intermediate steps.
- **Enable recording** = can be activated to record additional simulation data, which are max. 576000 sets of six values: actual values of U, I and P, plus Umpp, Impp and Pmpp

Parameter

This input defines technology parameters for the manual technology mode. In other modes (see the different EN 50530 VIs) these parameters cannot be altered and are fixed to values according to the standard.

- FFu = Fill factor for voltage
- FFi = Fill factor for current
- Cr = Correction factor Cr for Uoc in m^2/W
- Cg = Correction factor Cg for Uoc in W/m^2
- Cu = Correction factor Cu for Uoc
- alpha = Temperature coefficient for Isc in $1/\text{C}^\circ$
- beta = Temperature coefficient for Uoc in $1/\text{C}^\circ$

Parameter

FFu	Cr	Cg
0,000	0,000000	0,000000
FFi		alpha
0,000		0,000000
Cu		beta
0,000000		0,000000

Mode

This input determines the calculation base for the PV simulation curve.

With selection **MPP**, the curve is calculated based on the two input values Umpp and Impp. In mode **ULIK**, it's based on Uoc and Isc. The other two values not related to the selected mode can be ignored.

Note: No matter what mode has been selected, the values entered here must be reasonable or else the curve calculation will fail.

Mode

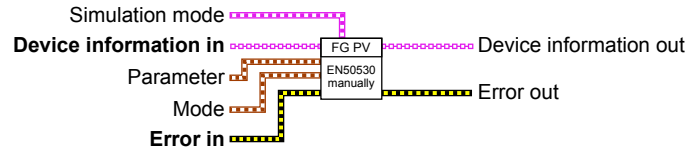
MPP

Uoc	Umpp
0,000	0,000
Isc	Impp
0,000	0,000

Error out

See page 17.

4.16 FG_xy_pv_en50530_man.vi



This VI is used to manually configure the technology parameters of the extended photovoltaics function for solar panel simulation according to EN 50530. Unlike the regular PV simulation, the table data is calculated by the power supply. This allows the extended PV simulation to permanently re-calculate the XY table during runtime, and to record simulation data. The configuration requires entering a specific set of technical properties of solar panels.

Refer to the standard EN 50530 for definition and additional information about the parameters, and refer to the ModBus register list of the power supply for the mPower-specific adjustable ranges of these parameters.

Device information in

See page 17.

Simulation mode

This input is used to select the simulation mode between:

- **E/T** = Continuous simulation with adjustable temperature and irradiation
- **U/I** = Continuous simulation with adjustable MPP values (U_{mpp} , I_{mpp})
- **DAY U/I** = Day trend mode for U/I
- **DAY E/T** = Day trend mode for E/T

Necessary parameters:

- **Path day curve** = contains the path to the day curve (max. 10000 points) to load for modes DAY U/I and DAY E/T.
- **U** = Set value of voltage (same as single set value command) to limit the output voltage during simulation; this is usually set to the open circuit voltage (U_{oc}) of the panel
- **P** = Set value of power (same as single set value command) to limit the output power during simulation



Optional parameters:

- **Interpolation** = can be activated to achieve a smoother trend between two points on the loaded day curve. The device would then calculate intermediate steps.
- **Enable recording** = can be activated to record additional simulation data, which are max. 576000 sets of six values: actual values of U, I and P, plus U_{mpp} , I_{mpp} and P_{mpp}



Parameter

This input defines technology parameters for the manual technology mode. In other modes (see the different EN 50530 VIs) these parameters cannot be altered and are fixed to values according to the standard.

- FFu = Fill factor for voltage
- FFi = Fill factor for current
- Cr = Correction factor Cr for Uoc in m^2/W
- Cg = Correction factor Cg for Uoc in W/m^2
- Cu = Correction factor Cu for Uoc
- alpha = Temperature coefficient for Isc in $1/C^\circ$
- beta = Temperature coefficient for Uoc in $1/C^\circ$

Parameter		
FFu	Cr	Cg
0,000	0,000000	0,000000
FFi	alpha	
0,000	0,000000	
Cu	beta	
0,000000	0,000000	



Mode

This input determines the calculation base for the PV simulation curve.

With selection **MPP**, the curve is calculated based on the two input values Umpp and Impp. In mode **ULIK**, it's based on Uoc and Isc. The other two values not related to the selected mode can be ignored.

Note: No matter what mode has been selected, the values entered here must be reasonable or else the curve calculation will fail.

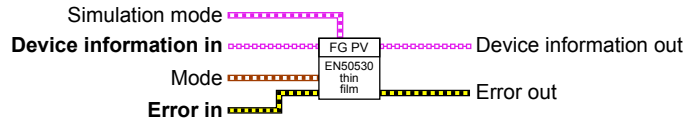
Mode	
MPP	
Uoc	Umpp
0,000	0,000
Isc	Impp
0,000	0,000



Error out

See page 17.

4.17 FG_xy_pv_en50530_thin_film.vi



This VI is used to configure the thin film technology parameters of the extended photovoltaics function for solar panel simulation according to EN 50530. Unlike the regular PV simulation, the table data is calculated by the power supply. This allows the extended PV simulation to permanently re-calculate the XY table during runtime, and to record simulation data. The configuration requires entering a specific set of technical properties of solar panels. All parameters required for thin film technology, such as FFu etc., are rigid, and defined by the standard.

Refer to the standard EN 50530 for definition and additional information about the parameters, and refer to the ModBus register list of the power supply for the mPower-specific adjustable ranges of these parameters.

Device information in

See page 17.

Simulation mode

This input is used to select the simulation mode between:

- **E/T** = Continuous simulation with adjustable temperature and irradiation
- **U/I** = Continuous simulation with adjustable MPP values (Umpp, Impp)
- **DAY U/I** = Day trend mode for U/I
- **DAY E/T** = Day trend mode for E/T

Necessary parameters:

- **Path day curve** = contains the path to the day curve (max. 10000 points) to load for modes DAY U/I and DAY E/T.
- **U** = Set value of voltage (same as single set value command) to limit the output voltage during simulation; this is usually set to the open circuit voltage (Uoc) of the panel
- **P** = Set value of power (same as single set value command) to limit the output power during simulation



Optional parameters:

- **Interpolation** = can be activated to achieve a smoother trend between two points on the loaded day curve. The device would then calculate intermediate steps.
- **Enable recording** = can be activated to record additional simulation data, which are max. 576000 sets of six values: actual values of U, I and P, plus Umpp, Impp and Pmpp



Parameter

This input defines technology parameters for the manual technology mode. In other modes (see the different EN 50530 VIs) these parameters cannot be altered and are fixed to values according to the standard.

- FFu = Fill factor for voltage
- FFi = Fill factor for current
- Cr = Correction factor Cr for Uoc in m^2/W
- Cg = Correction factor Cg for Uoc in W/m^2
- Cu = Correction factor Cu for Uoc
- alpha = Temperature coefficient for Isc in $1/C^\circ$
- beta = Temperature coefficient for Uoc in $1/C^\circ$

Parameter

FFu	Cr	Cg
0,000	0,000000	0,000000
FFi	alpha	
0,000	0,000000	
Cu	beta	
0,000000	0,000000	



Mode

This input determines the calculation base for the PV simulation curve.

With selection **MPP**, the curve is calculated based on the two input values Umpp and Impp. In mode **ULIK**, it's based on Uoc and Isc. The other two values not related to the selected mode can be ignored.

Note: No matter what mode has been selected, the values entered here must be reasonable or else the curve calculation will fail.

Mode

MPP

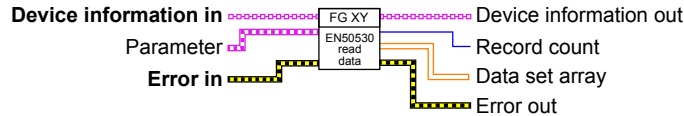
Uoc	Umpp
0,000	0,000
Isc	Impp
0,000	0,000



Error out

See page 17.

4.18 FG_xy_pv_en50530_read_data.vi



This VI can be used to read data from the power supply which has been recorded during a function run of the extended photovoltaics EN 50530 solar panel simulation. This data recording feature can be activated before or during the simulation, and records up to 576,000 sets of six values (actual values of U, I, and P, plus U_{mpp} , I_{mpp} and P_{mpp}) every 100 ms (resulting in a maximum recording time of 16 hours). Recording is started and stopped with the simulation.


However, once the data buffer is filled, it will be overwritten from the beginning, and the counter **Record count** reset. So, after more than 16 hours of recording time, the buffer is filled, and the counter does not return the correct count. Also, the VI will read the number of data sets indicated by the current counter value.

 **Device information in** — See page 17.

 **Parameter**

Parameter can be used to enable writing the data read from the power supply to a file. The file path is generated by the VI as **PV_RECORD_xx**, where **xx** increments for every file found with the same prefix.

The file format is identical to the file format when saving the data from the control panel of the system to a USB stick. It contains one header line and 6+1 columns.


 **Recorded data**

This output provides a two-dimensional array which contains the data sets which have been read from the power supply. The maximum number of rows in this array is 576,000. The number of filled rows is identical to the value in output **Record count**. The rows have six columns with values which are defined as (from left to right):s

- actual voltage in Volts
- actual current in Amps,
- actual power in Watts,
- voltage in the maximum power point (U_{mpp}),
- current in the maximum power point (I_{mpp}),
- power in the maximum power point (P_{mpp} , calculated from $P = U * I$)

 **Record count**

Returns the counter used internally by the system while recording data. This counter is also used to read the same number of data sets from the system. If a smaller portion of data is required, for example the first 1000 entries, these will have to be extracted afterwards either from the array **Recorded data**, or from the saved file.

 **Error out** — See page 17.

5 Error Code Reference

The Error Codes table below contains all error codes which can be output from the various VIs which feature an **Error out**. If we reference the **Error out** data, the **Code** component has four digits. The highest two digits define an error code range which is always 9900. The lowest two digits define the error code shown in the Error Codes table.

Error Codes	
Code	Description
0	"No error"
01–57	See <i>Programming Guide</i> , ModBus section "Communication errors"
67	"Tolerance value out of range or Tsr not in range of 0,03...1s or Tsf not in range of 0,03...1s" (Any combination of these can occur. Indicates one or multiple wrong values were given for the last command)
69	"Data read error" (when reading from the unit)
70	"No data read"
71	General error, no specific text
77	"Value out of range" (the value as entered into input "Value" is too high or too low)
78	"IP is incorrect" (only when trying to write an IP of wrong format to the device)
79	"String too long" (for example, when trying to write command "User text")
81	"1st or 2nd or 3rd or 4th or 5th sequence number to process in function is wrong. It must be 0–5" "Mode wrong. It must be UIP or UIR." "Repetitions of function wrong. It must be 0–255" "Example: 1;1;1;1;1;UIP;255" Any combination of these can occur if the function manager settings were wrong.
92	"Data type in input 'value' wrong" (occurs if the string/value(s) at input "value" are in wrong format/type)
93	"Stop point activation wrong. Must be 'on' or 'off'." "Stop after x repetitions of sequence wrong, it must be 0–255" "Stop after x repetitions of function wrong, it must be 0–255" "Sequence number wrong, it must be 1–5" "Sequence point wrong, it must be 0–9" "Example: 1;1;1;1;1;UIP;255" Any combination of these can occur if function settings are wrong. Refer to the user's guide of your device for further information about the function manager and function setup.
96	"Object not supported" (if a command has been chosen that is not supported for the selected device)
97	"No device with ... found" (device(s) at a certain hardware port can not be detected)
98	"There are x devices with the same port" "There are x devices with the same device node" "There are x devices with the same serial number" Any combination of these can occur. Normally, this does not prevent opening the devices, but if, for example, two devices with the same serial number are found (should actually never occur), then they can't be opened by their serial number, but by their device node
99	"No device found during scan"

mPower™ DC 3xx Series
Programmable Power Supplies
LabView VI Guide P/N: 501048-DC3LV-C

© 2023, Marway Power Systems, Inc. All rights reserved.



Marway Power Solutions
1721 S. Grand Ave., Santa Ana, CA 92705
800-462-7929 • marway@marway.com