

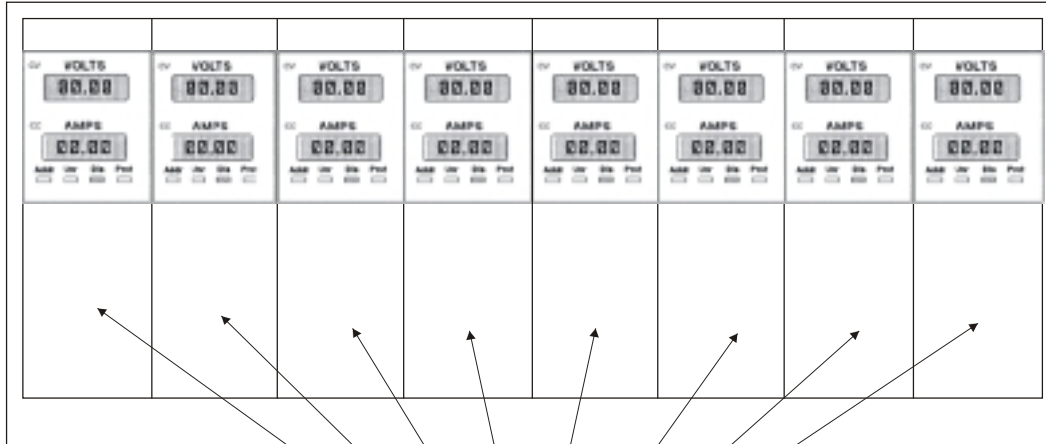
Series 2040 Test Systems

**Modular
Power
System
Programming
Guide**

**Part Number 4200-0213
Version 1.1**

***Digalog Functional Calls For The
Agilent/HP Series 661xxA
MPS Power Modules***

MPS Power Modules



Power Modules

66101A	0 - 8V/16A
66102A	0 - 20V/7.5A
66103A	0 - 35V/4.5A
66104A	0 - 60V/2.5A
66105A	0 - 120V/1.25A
66106A	0 - 200V/0.75A

MPSGpibInit

This functional call initializes the operating system's variables and hardware needed by the other functional calls for communicating with the MPS power modules within the selected MPS mainframe over the IEEE-488 bus. It should be executed only once (at the start of the user's overall test program).

Also, this call configures each power module to disable its output due to an over-voltage, over-current, over-temperature, or remote-inhibit "protection" condition and to activate its DFI output in response to any of these "protection" conditions. To enable any MPS power module which has activated its "protection" condition and shutdown to notify all other power modules in the mainframe to do likewise, connect one jumper wire between the "FLT +" and "INH+" terminals of the removable digital connector on the back of the mainframe chassis and another jumper wire between its "FLT-" and "INH -/Common" terminals.

Visual Basic Declaration:

```
Public Sub MPSGpibInit(ByVal GpibCntrlrIndex As Byte, ByVal MFAddress As Byte)
```

Call MPSGpibInit(GpibCntrlrIndex, MFAddress)

WHERE:

GpibCntrlrIndex	Index value of the GPIB controller board located in the computer which communicates with the selected MPS mainframe.
=	0 to 9
MFAddress	GPIB Address of the selected MPS Mainframe.
=	0 to 30

EXAMPLE:

```
Dim GpibCntrlrIndex As Byte
Dim MFAddress As Byte
```

```
GpibCntrlrIndex = 0
```

```
MFAddress = 10
```

```
Call MPSGpibInit(GpibCntrlrIndex, MFAddress) ..... Configure GPIB Controller Board "GPIB0"
..... to communicate with MPS Mainframe
..... whose GPIB address is 10.
```

MPSReset

This functional call resets the output of the selected MPS power module by opening its output relay contacts (if equipped) and reprogramming its output levels to 0V@(minimal source current). (Refer to the OEM's user's guide for more details.)

Visual Basic Declaration:

Public Sub MPSReset(ByVal Unit As Integer)

Call MPSReset(Unit)

WHERE:

Unit

- = -1 Reset all MPS power modules.
- = 0 to 7. Mainframe slot index of the selected MPS power module.

EXAMPLE:

Dim Unit As Byte

Unit = 0

Call MPSReset(Unit) Reset MPS Unit #0.

MPSPwr

This functional call programs the selected MPS power module to the specified output settings. The validity of the voltage and current level values is **not** verified before being sent “as is” to the selected MPS power module. If the output is to be disabled, then this command will be sent first to the power module prior to all other programming commands. If the output is to be enabled, then this command will be sent last after all other programming commands. This call will return control to the user’s test program after the specified settling time has elapsed regardless of the actual output level of the selected MPS power module.

Visual Basic Declaration:

```
Public Sub MPSPwr(ByVal Unit As Byte, ByVal Polarity As Byte, ByVal Voltage As
Double, ByVal Current As Double, ByVal Mode As Byte, ByVal SettlingTime As
Double)
```

Call MPSPwr(Unit, Polarity, Voltage, Current, Mode, SettlingTime)

WHERE:

Unit	=	0	Mainframe slot index of the selected MPS power module. to 7
Polarity	=	0	Output relay polarity is reversed i.e., opposite of that of the selected power module’s output.
	=	1	Output relay polarity is normal/straight-through i.e., identical to that of the selected power module’s output.
Voltage	=	0	to “Maximum Programmable Output Voltage” value of the programming output voltage level for the selected MPS power module (in units of Volts). (Refer to the OEM’s user’s guide for more details.)
Current	=	0	to “Maximum Programmable Output Current” value of the programming output current level for the selected MPS power module (in units of Amperes). (Refer to the OEM’s user’s guide for more details.)

2040 MPS Programming Guide

Mode

- = 0 Disable output of the selected MPS power module.
- = 1 Enable output of the selected MPS power module.

SettlingTime

- = 0 to 65 Time to wait for the output to settle to its programmed value after being enabled (in units of seconds).

EXAMPLE:

Dim Unit As Byte
Dim Polarity As Byte
Dim Voltage As Double
Dim Current As Double
Dim Mode As Byte
SettlingTime As Double

Unit = 0
Polarity = 1
Voltage = 5.25
Current = 0.25
Mode = 1
SettlingTime = 0.25

Call MPSPwr(Unit, Polarity, Voltage, Current, Mode, SettlingTime) Program MPS unit #0
..... to 5.25V@0.25A with its output polarity normal
..... and output mode enabled, and then wait 0.25s
..... before continuing to the next line of the user's test program.

MPSChek

This functional call returns the output voltage and current levels of the selected MPS power module measured by the unit itself via its sense lines.

Visual Basic Declaration:

```
Public Sub MPSChek(ByVal Unit As Byte, ByRef Voltage As Double, ByRef Current As Double)
```

Call MPSChek(Unit, Voltage, Current)

WHERE:

Unit = 0 to 7 Mainframe slot index of the selected MPS power module.

Voltage = Value of the measured output voltage level returned by the selected power module (in units of Volts).

Current = Value of the measured output current level returned by the selected power module (in units of Amperes).

EXAMPLE:

```
Dim Unit As Byte  
Dim Voltage As Double  
Dim Current As Double
```

```
Unit = 0  
Call MPSChek(Unit, Voltage, Current) ..... Return output voltage and current of MPS unit #0.
```


MPSGpibWrite

This functional call sends the specified message to the selected MPS power module. The maximum possible message length is limited to the amount of available (free) computer memory.

Visual Basic Declaration:

```
Public Sub MPSGpibWrite(ByVal Unit As Byte, ByVal GpibStr As String)
```

Call MPSGpibWrite(Unit, GpibStr)

WHERE:

Unit = 0 to 7 Mainframe slot index of the selected MPS power module.

GpibStr Message to be sent to the selected MPS power module.

EXAMPLE:

```
Dim Unit As Byte  
Dim GpibStr As String
```

```
Unit = 0
```

```
GpibStr = "*IDN?"
```

```
Call MPSGpibWrite(Unit, GpibStr) ..... Send ID command to MPS unit #0.
```

```
Call MPSGpibRead(Unit, GpibStr) ..... Retrieve reply message from MPS unit #0.
```

MPSGpibRead

This functional call retrieves the message sent by the selected MPS power module in reply to a previous GPIB command. The maximum possible message length is 1024 bytes.

Visual Basic Declaration:

```
Public Sub MPSGpibRead(ByVal Unit As Byte, GpibStr As String)
```

Call MPSGpibRead(Unit, GpibStr)

WHERE:

Unit	=	0	Mainframe slot index of the selected MPS power module. to 7
GpibStr			Reply message sent by the selected MPS power module.

EXAMPLE:

```
Dim Unit As Byte
Dim GpibStr As String
```

```
Unit = 0
GpibStr = "*IDN?"
Call MPSGpibWrite(Unit, GpibStr) ..... Send ID command to MPS unit #0.
Call MPSGpibRead(Unit, GpibStr) ..... Retrieve reply message from MPS unit #0.
```

Related Error Codes

25088 / 098:000	GPIB System (EDVR) error; Possibly due to non-initialization or over-initialization of OS variables and GPIB controller board.
25090 / 098:002	GPIB No-Listener (ENOL) error; Possibly due to an incorrectly specified GPIB address
25094 / 098:006	GPIB I/O Aborted/Timeout (EABO) error; Possibly due to lack of reply from GPIB device
25095 / 098:007	GPIB non-existent board (ENEB) error; Possibly due to an incorrectly specified GPIB controller board index number/identity.
27421 / 107:029	No 8-Channel P/S Interface Board in Testhead
27422 / 107:030	Bad parameter value in MPS functional call
27423 / 107:031	No 8-Channel P/S Interface Selftest Board in Selftest Unit
27424 / 107:032	The MPS module is set for local sensing, not remote sensing.
27425 / 107:033	An open circuit exists between the MPS power module's output connections and their corresponding Patchboard Receiver pins.
27426 / 107:034	The MPS power module could not be properly set up to detect the TRIGGER IN signal.
27427 / 107:035	The MPS power module did not detect the TRIGGER IN signal.