Auto Sequence Programming Examples for GPIB-M

Example 1

Using GPIB, program and execute a sequence that will step up voltage from 1 to 10 V in 1 V increments per second, then hold the output at 10V until triggered by the external trigger line, ending the sequence. Set the current limit to 30A, and the over voltage trip level to 10.25V for the entire sequence.

Programming:

Command	**Comment**
SYST:REM:SOUR GPIB	Set up for communication with GPIB
PROG:NAME 1	Select auto sequence #1
PROG:STEP1 1, 30, 10.25, 1s	Step 1
PROG:STEP2 2, 30, 10.25, 1	Step 2 - note, by default, step duration is in seconds
PROG:STEP3 3, 30, 10.25, 1	Step 3
PROG:STEP4 4, 30, 10.25, 1	Step 4
PROG:STEP5 5, 30, 10.25, 1	Step 5
PROG:STEP6 6, 30, 10.25, 1	Step 6
PROG:STEP7 7, 30, 10.25, 1	Step 7
PROG:STEP8 8, 30, 10.25, 1	Step 8
PROG:STEP9 9, 30, 10.25, 1	Step 9
PROG:STEP10 10, 30, 10.25, TRIG	Step 10
PROG:TRIG:SOUR EXT	Set the trigger source to the trigger line at the rear panel

Running:

Command	**Comment**
SYST:REM:SOUR GPIB	Set up for communication with GPIB
PROG:NAME 1	Select auto sequence #1
PROG:STAT RUN	Start auto sequence. Assert the trigger line when step 10 is reached to end the program.

Example 2

Same as Example 1, on a CAN-enabled power supply set to multichannel address 2.

Programming:

Command	**Comment**
SYST2:REM:SOUR MCH	Set up for communication via CANbus
SYST2:REM:STAT REM	Enter remote mode

PROG2:NAME 1	Select auto sequence #1
PROG2:STEP1 1, 30, 10.25, 1s	Step 1
PROG2:STEP2 2, 30, 10.25, 1	Step 2 - note, by default, step duration is in seconds
PROG2:STEP3 3, 30, 10.25, 1	Step 3
PROG2:STEP4 4, 30, 10.25, 1	Step 4
PROG2:STEP5 5, 30, 10.25, 1	Step 5
PROG2:STEP6 6, 30, 10.25, 1	Step 6
PROG2:STEP7 7, 30, 10.25, 1	Step 7
PROG2:STEP8 8, 30, 10.25, 1	Step 8
PROG2:STEP9 9, 30, 10.25, 1	Step 9
PROG2:STEP10 10, 30, 10.25, TRIG	Step 10
PROG2:TRIG:SOUR EXT	Set the trigger source to the trigger line at the rear panel

Running:

Command	**Comment**
SYST2:REM:SOUR GPIB	Set up for communication with GPIB
PROG2:NAME 1	Select auto sequence #1
PROG2:STAT RUN	Start auto sequence. Assert the trigger line when step 10 is reached to end the program.

Example 3

Edit the sequence programmed in Example 1 to trigger on a command, and to run twice before ending.

Command	**Comment**
SYST:REM:SOUR GPIB	Set up for communication with GPIB
PROG:NAME 1	Select auto sequence #1
PROG:NAME?	Receive "1"
PROG:STEP:COUNT?	Query the number of steps Receive "10"
PROG:TRIG:SOUR IMM	Set the trigger source to be the SCPI command "INIT:IMM"
PROG:TRIG:SOUR?	Query the trigger source Receive "IMM"
PROG:REP 2	Change sequence repetitions to 2 times.

Running:

Command	**Comment**
PROG:NAME 1	Select auto sequence #1

PROG:STAT RUN	Start auto sequence
INIT:IMM	Send the trigger command when step 10 is reached to advance to the next step. (First repetition.)
INIT:IMM	Send the trigger command when step 10 is reached to advance to the next step. (Second repetition.)

Example 4

If you need to approximate a triangle wave, you may do so in a step-wise fashion. Another method would be to change the slew rate. Note that the slew rate feature offers limited accuracy and the result may be that the output waveform may be truncated, or that the final voltage values of each step may not be reached.

Program a triangle wave with an amplitude of 5 V, a period of 4 seconds, and a 12.5 V DC offset. Current limit should be set to 3 A, and OVP disabled.

Solution:
You want the voltage to change at a rate of 5V/2 seconds. Program this as a slew rate with a step of 0.05 V and an interval of 20ms.

Command	**Comment**
SOUR:SLEW:STEP 0.05	Set slew rate step size to 0.05V
SOUR:SLEW:INT 20ms	Set slew rate interval to 20ms
PROG:NAME 2	Select auto sequence program 2
PROG:STEP1 10,3,,2	Set step 1 to 10V, 2 second duration
PROG:STEP2 15,3,,2	Set step 2 to 15V, 2 second duration
PROG:REP INF	Set sequence to run infinitely

The power supply will take a few seconds to ramp up to the operating voltages when the sequence is first started. To end the sequence, send the command "PROG:STAT STOP."

Note that the slew rate is a configuration of the power supply, not specific to the sequence. Hence, if the slew rate is changed, it will affect the how output looks when the sequence is run.

NOTES

To advance the sequence to the next step:
PROG:STEP:NEXT

To query the number of times the sequence has executed:
PROG:REP:EXEC?

To query the step number that is currently executing:
PROG:STEP:EXEC?

To query the number of steps that have been programmed in a sequence:
PROG:STEP:COUNt?

To exit and return to normal operation,
PROG:EXIT