Programmer’s Guide

PeakTech4055MV/4060 Function Generator

1. SCPI command language

SCPI (Standard Commands for Programmable Instruments) is a standardized set of commands and be based on ASCII code, this command is widely used in electrical measurement instruments to realize programming control of the instruments through corresponding remote control interface.

1.1 Command format: the command format of SCPI is as follows:

```
    [SOURce]:FREQuency   {<Value>[MHz|kHz|Hz|mHz]|MINimum|MAXimum}
```

Brace { }: contains the parameter options of the command string.

Separating character|: separate several parameter options, only one parameter option could be chosen at one time.

Angular bracket < >: indicates this option is a parameter value.

Square bracket [ ]: parameters contains in this character is optional and could be omitted.

The sign { }, │, < > and[ ]in the command example are all shown for convenient expression, and not sent with command string and not allowed in practical applications.

1.2 Command abbreviation: command for instrument is either can be used in abbreviation format which is simple and brief for program writing, or be used in full format which is clear for program meaning and easy to read. The commands given by this guide is full format, among which written by uppercase stands for abbreviation format, abbreviation format is shorter than 4 letters. The other format besides that will result in mistake.

The command will not distinguish uppercase and lowercase, and it allows both uppercase and lowercase, or mixing uppercase and lowercase is also allowed. For example, “FREQ 1KHZ”, “freq 1khz” and “Freq 1kHz”are all acceptable with same running result. But for unit, letter M is different from letter m, and users can’t mix-use. For example, 1 mHz and 1MHz are both acceptable but with different running result.

1.3 Command separator: with hierarchical structure, SCPI command can be divided into Root, Subnode and Endnode commands. Using colon “:” to separate the keywords, and separate command keyword and parameters with space, if the command contains many parameters, you can use comma “,” to separate.

For example, Apply:Sin   1 kHz, 5.2 Vpp, -0.2Vdc

Use semicolon “;” to link several commands with same scale and under one subsystem, in this case the higher scale command could be omitted and the program becomes simpler.

For example: AM:INTernal:FREQuency 3kHz

AM:INTernal:FUNCtion SINusoid

Above two commands could be linked with semicolon “;” and are simplified as one command as:

AM:INTernal:FREQuency 3kHz;FUNCtion SINusoid

Use a semicolon and a colon “;:” to link several commands under different subsystems, every command should start with root command.

For example: AM:STATe ON;:FREQuency 100kHz;:AM:DEPTh?

1.4 Parameter type: parameter type has 4 formats as following.

1.4.1 Numeric value parameter: numeric value parameter is presented by decimal number, composed by digits, minus and decimal point, such as -8.253. Floating-point number could be used to indicate either, such as 1.0E+06. You also can use two special values Minimum and Maximum to instead parameter value of command. Min set the parameter to be the allowed minimum value, and Max set the parameter to be the allowed maximum value. You can add the unit in the end of parameter value, such as kHz, Vrms and so on. If not adding, you can use the basic unit Hz, V, s and so on. The unit omitting parameter can make the program simple, but sometimes adding unit can be more convenient, for example, Freq 10MHz is more simple than 10000000.

Command of value parameter, for example, Frequency 1000, or Amplitude Max.

1.4.2 Discrete parameter: discrete parameter only has a few of value, and be same as commands, you can use full or abbreviation format, or mixing uppercase and lowercase is also allowed.

The command of discrete parameter such as FSKey:Source Internal.

1.4.3 Boolean parameter: a Boolean parameter specifies a single binary condition which is either true or false. For “True”, the parameter value is “ON” or “1”, and for “False”, the parameter value is “OFF” or “0”.

The command of Boolean parameter such as FM: State On.

1.4.4 Character string parameter: this parameter is constituted by ASCII characters, and enclosed by a pair of quotation marks. Such as, “No error”.

1.5 Parameter query: users can add interrogation “?” in the end of commands, so you can query the current value of most parameters. For example, the present CHA is sine wave, Freq 1 kHz, output port is open.

Sending query SOURce:Function?, can return discrete parameter “SIN”, it indicates that the current function is sine wave. For discrete parameter, both the query and return are abbreviation format with the uppercase.

Sending query SOURce:Frequency?, can return value parameter “1.000000E+03”, it indicates that the current frequency is 1kHz. For value parameter, basic unit is default when query or return a value without a unit with it, the format of the value is floating-point.

Sending query ‘Output?’, can return Boolean parameter “1”. It indicates that the current state of output port is open. For Boolean parameter, query or return “1”or“0”.

Sending query System:Error?, can return the character string parameter “No error”, it indicates that no error happened now. For character string parameter, query or return a character string enclosed in a pair of quotation marks.

1.6 Universal command: universal commands start with *, with length of three characters, and could have parameters.

Example of universal command:*RST

1.7 Command end character: total characters in a command string should not be more than 60. Each end of character string should be added an end character (shift character of ASCII code 10), indicates an end of character string in order to avoid a mistake. It is suggested that the end character is written in the sending function when programmable so that it is not necessary for adding it at the end of each command and it will never be lost carelessly.

2. Command set

The instrument set the SCPI command of most functions, except the arbitrary edition and parameter calibration function, since the operations of the two are complex and remote control interface is not used usually.

The command keyword in the command set is written in full format, the uppercase is the abbreviation of this command, the unit of value could not be omitted, and the names of waveforms have only abbreviation format.

2.1 Direct configuration command

```
[SOURce]:APPLy:SINusoid   [<frequency>[,<amplitude>[,<offset>]]]
[SOURce]:APPLy:SQUare   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:RAMP   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:NOISE   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:PPULS   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:NPULS   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:STAIR   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:HSINE   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:LSINE   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:REXP   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:RLOG   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:TANG   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:SINC   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:ROUND   [<frequency >[,< amplitude [,<offset >]]]
[SOURce]:APPLy:CARD   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy:QUAKE   [<frequency >[,< amplitude >[,< offset >]]]
[SOURce]:APPLy？
```

Users may use Apply command to configure directly the four parameters of the function generator: function, frequency, amplitude and offset, the sequence of which could not be changed. For example:

Configuration continuously outputs sine, frequency 10kHz, amplitude 1.2Vpp,offset 0.5Vdc,the command of which is as follows:

```
SOURce:Apply:Sin   10kHz, 1.2, 0.5
```

Where frequency, amplitude and offset three parameters could be omitted from the last to the first, the omitted parameters keeps as the current value. Such as:

```
    SOURce:Apply:Sin  10kHz,1.2        set Frequency and Amplitude, offset is omitted.
    SOURce:Apply:Sin  10kHz            set Frequency, amplitude and offset are omitted.
    SOURce:Apply:Sin                   frequency, amplitude and offset are omitted.
```

Use query command Source:Apply? To return current configured function, frequency, amplitude and offset value. Such as: “SIN,1.000000E+04, 1.200000E+00, 0.500000E+ 00”. Use Apply command to configure directly and conveniently four parameters. Single command is more applicable when changing specifically parameters.

2.2 Output configuration command

```
[SOURce]:FUNCtion  {SINusoid|SQUare|RAMP|NOISe|PPULS|NPULS|STAIR
                   |HSINE|LSINE|REXP|RLOG|TANG|SINC|ROUND|CARD|QUAKE}
 [SOURce]:FUNCtion?                          /* Output function
[SOURce]:FUNCtion:SQUare:DCYCle  {<Value %>|MINimum|MAXimum}
[SOURce]:FUNCtion:SQUare:DCYCle?        /* Square duty cycle
[SOURce]:FUNCtion:RAMP:SYMMetry  {<Value %>|MINimum|MAXimum}
[SOURce]:FUNCtion:RAMP:SYMMetry?        /* Ramp symmetry
[SOURce]:FREQuency[:CW]   {<Value MHz|kHz|Hz|mHz>|MINimum|MAXimum}
[SOURce]:FREQuency[:CW]?                /* Frequency(CW means continuous waveform)
[SOURce]:PERiod  {<Value s|ms>|MINimum|MAXimum}
[SOURce]:PERiod?                   /* Period
[SOURce]:VOLTage[:AMPLitude]   {<Value Vrms|mVrms|Vpp|mVpp>
                               |MINimum|MAXimum}
[SOURce]:VOLTage[:AMPLitude]?           /* Amplitude
[SOURce]:VOLTage:OFFSet  {<Value Vdc|mVdc>|MINimum|MAXimum}
[SOURce]:VOLTage:OFFSet?                /* DC offset
[SOURce]:VOLTage:ATTenuation:   {<Value dB>|MINimum|MAXimum|AUTO}
[SOURce]:VOLTage:ATTenuation?           /* Amplitude attenuation
[SOURce]:VOLTage:UNIT  {Vpp|Vrms}
[SOURce]:VOLTage:UNIT?                  /* Amplitude unit
OUTPut:POLarity  {NORMal|INVerted}
```

```
OUTPut:POLarity?                          /* Output polarity

OUTPut[:STATe]   {ON|OFF}

OUTPut[:STATe]?                           /* Output state
```

2.3 Frequency modulation (FM) command

```
FM:DEViation   {<Value MHz|kHz|Hz|mHz>|MINimum|MAXimum}

FM:DEViation?                                  /* Frequency deviation

FM:INTernal:FREQuency   {<Value kHz|Hz|mHz>|MINimum|MAXimum}

FM:INTernal:FREQuency?                         /* Modulation frequency

FM:INTernal:FUNCtion   {SINusoid|SQUare|RAMP|NOISe|PPULS|NPULS|STAIR

                    |HSINE|LSINE|REXP|RLOG|TANG|SINC|ROUND|CARD|QUAKE}

FM:INTernal:FUNCtion?                          /* Modulation function

FM:STATe   {ON|OFF}

FM:STATe?                                      /* Modulation state
```

2.4 Amplitude modulation (AM) command

```
AM:DEPTh   {<Value %>|MINimum|MAXimum}

AM:DEPTh?                                        /* Modulation amplitude depth

AM:INTernal:FREQuency   {<Value kHz|Hz|mHz>|MINimum|MAXimum}

AM:INTernal:FREQuency?

AM:INTernal:FUNCtion  {SINusoid|SQUare|RAMP|NOISe|PPULS|NPULS|STAIR

                   |HSINE|LSINE|REXP|RLOG|TANG|SINC|ROUND|CARD|QUAKE}

AM:INTernal:FUNCtion?

AM:STATe   {ON|OFF}

AM:STATe?
```

2.5 Phase modulation (PM) command

```
PM:DEViation   {<Value deg>|MINimum|MAXimum}

PM:DEViation?                                    /* Phase deviation

PM:INTernal:FREQuency   {<Value kHz|Hz|mHz>|MINimum|MAXimum}

PM:INTernal:FREQuency?
```

```
PM:INTernal:FUNCtion  {SINusoid|SQUare|RAMP|NOISe|PPULS|NPULS|STAIR
                     |HSINE|LSINE|REXP|RLOG|TANG|SINC|ROUND|CARD|QUAKE}
PM:INTernal:FUNCtion?
PM:STATe  {ON|OFF}
PM:STATe?
```

2.6 Pulse width modulation (PWM) command

```
PWM[:DEViation]:DCYCle  {<Value %>|MINimum|MAXimum}
PWM[:DEViation]:DCYCle?                     /* Pulse width deviation (duty cycle)
PWM:INTernal:FREQuency  {<Value kHz|Hz|mHz>|MINimum|MAXimum}
PWM:INTernal:FREQuency?
PWM:INTernal:FUNCtion  {SINusoid|SQUare|RAMP|NOISe|PPULS|NPULS|STAIR
                        |HSINE|LSINE|REXP|RLOG|TANG|SINC|ROUND|CARD|QUAKE}
PWM:INTernal:FUNCtion?
PWM:STATe  {ON|OFF}
PWM:STATe?
```

2.7 Frequency shift keying (FSK) command

```
FSKey:FREQuency  {<Value MHz|kHz|Hz|mHz>|MINimum|MAXimum}
FSKey:FREQuency?                                /* Hop frequency
FSKey:INTernal:RATE  {<Value kHz|Hz|mHz>|MINimum|MAXimum}
FSKey:INTernal:RATE?                         /* Hop rate
FSKey:SOURce  {INTernal|EXTernal}
FSKey:SOURce?                                   /* Trigger source
FSKey:STATe  {ON|OFF}
FSKey:STATe?
```

2.8 Frequency sweeping command

```
FREQuency:STARt  {<Value MHz|kHz|Hz|mHz>|MINimum|MAXimum}
FREQuency:STARt?                                   /* Start frequency
FREQuency:STOP  {<Value MHz|kHz|Hz|mHz>|MINimum|MAXimum}
FREQuency:STOP?                                    /* End frequency
```

```
SWEep:SPACing  {LINear|LOGarithmic}
SWEep:SPACing?                                   /* Sweeping mode (step interval)
SWEep:TIME  {<Value s|ms>|MINimum|MAXimum}
SWEep:TIME?                                      /* Sweeping time
SWEep:STATe  {ON|OFF}
SWEep:STATe?                                     /* Sweeping state
TRIGger:SOURce  {IMMediate|EXTernal}
TRIGger:SOURce?                                  /* Trigger source
*TRG                                             /* Single trigger
```

2.9 Burst command

```
BURSt:NCYCles  {<Value >|MINimum|MAXimum}
BURSt:NCYCles?                                   /* Pulse count
BURSt:INTernal:PERiod  {<Value s|ms>|MINimum|MAXimum}
BURSt:INTernal:PERiod?                           /* Repeated period
BURSt:PHASe  {<Value deg>|MINimum|MAXimum}
BURSt:PHASe?                                     /* Start phase
BURSt:STATe  {ON|OFF}
BURSt:STATe?                                     /* Trigger state
TRIGger:SOURce  {IMMediate|EXTernal}
TRIGger:SOURce?                                  /* Trigger source
*TRG                                             /* Single trigger
```

2.10 System command

```
*RST                                             /* system reset
*CLS                                             /* clear error queue
SYSTem:ERRor?                                    /* query error queue
SYSTem:LOCal                                     /* return to local
```

3. Error information

3.1 Error queue: Instrument has an error memory, with an error queue arraying in a first-in-first-out (FIFO) way. Each time an error occurred, the queue will store it and the generator will make an alarm at the same time. The queue stores 20 errors at most, and when exceed, the last error information will be defined as “Queue overflow”, indicating queue overflow and will not store new error.

3.2 Error reading: use error query command SYSTem:ERRor? to read an error, the first error read is also the first one to be stored. When an error is read, it will be cleared at the same time. If there is no stored error in this queue, or all the stored errors have been read out, “No error” will be returned.

3.3 Error clearance: use *CLS command or cut off the power of the generator to clear error queue, which will be not cleared when using resetting command *RST.

3.4 Error message: the form of error message is “-100, Queue overflow”. There are two error messages:“-1xx” indicates command error, “-2xx” indicates run error, such as:

“-100, Queue overflow”:error queue overflow, the quantity of stored error exceeds 20.

“-101, First level command error”:First scale (root) command error, may be writing error, such as, Swep

“-102, Second level command error”:Second scale command error, may be writing error, or this command is not matched with the first scale command, such as, FM:Depth 20％

“-103, Third level command error”:Third scale command error, may be writing error, or this command is not matched with the second scale command, such as, Fskey:Internal:Frequency 3kHz.

“-104, Invalid parameter”:invalid parameter, may be parameter writing error, or the parameter type is not accordance with current option, such as, FM:State 1

“-105, Invalid suffix(unit)”:invalid postfix (data unit),may be writing error of unit, or this unit is not in accordance with the current data, such as, Burst:Ncycles 3 cyc

“-106, Syntax error”:syntax error, of which the possibility is complex, such as using comma instead of space:Frequency, 6kHz

‘-107, Missing parameter’: missing parameter, there is no parameter behind the commands that should have one, for example, VOLTage:OFFSet.

“-201, Current function must be continuous”:Current function must be continuous, like the duty cycle of square could be set only under continuous function.

“-202, Current waveform not able to use Vrms”:Current waveform could not be described by Vrms, Vrms could be only used as amplitude unit when setting amplitude of sine, square and ramp.

“-203, *TRG only use in sweep or burst”:single trigger command is only applicable to frequency sweeping or burst functions.

“-204, Data out of range, value clipped to limit”:the data setting exceed its range, limit the setting value within the allowable range of the generator.

4. Application example

If users are not so good at SCPI language, they may make some mistakes when programming, such as :

A. Mistake in writing: the command set lists the full formats of all command keywords, where the uppercases are the abbreviations, and writings of command keywords besides those are incorrect. Besides, the use of the colon, comma, semicolon, interrogation and space must comply with the SCPI rule, or errors may happen easily.

B. Mistake in command path: SCPI command adopts hierarchical structure, and the command set gives corresponding paths of different command keywords. Such as, command string func:Squ:Symm is incorrect, since command keyword Symm could be under Ramp path only.

C. Value does not match with its unit: the command set gives units accepted by different parameters value, such as, command string Sour:Freq 1Vpp is incorrect, since voltage unit Vpp is not accepted by frequency value.

D. Not all the command keywords could be used as query command, but the command with interrogation ? listed in the command set could be used as query command. Such as FUNC:SQUare ? is incorrect.

But, when programming, mistakes could hardly be made if users program fully obey with the command rules listed in the command set.

With below examples, the users may understand more clearly the applications of each command, learn more about the program rules of SCPI language and realize the remote control of the instrument.

4.1 Continuous output configuration

Configure continuously output ramp, with symmetry of 25％,frequency 12.5kHz,amplitude 1.5Vpp,DC offset 0.8Vdc,output port open. The command is as follows:

```
SOURce:FUNCtion:RAMP:SYMMetry  25％
SOURce:FREQuency 12.5E3            /* use floating-point format
SOURce:VOLTage:AMPLitude  1.5Vpp
SOURce:VOLTage:OFFSet  0.8
OUTPut:STATe ON
```

4.2 Query continuous output

```
SOURce:Apply？                 /* query four basic parameters of continuous output
Return “RAMP,1.250000E+04,1.500000E+00,8.000000E-01”
SOURce:FUNCtion:RAMP:SYMMetry?   /* query symmetry of continuous output
                                    ramp
Return“2.500000E+01”
```

4.3 Amplitude modulation

Configure amplitude modulation output, with carrier of sine, frequency 20kHz,amplitude 2Vpp,DC offset 0Vdc,modulation depth 80％,modulation frequency 2kHz,modulation waveform sine, modulation open. The command is as follows:

```
SOURce:Apply:Sin  20kHz,2,0     /* configure four carrier parameters directly
AM:DEPTh 80％
AM:INTernal:FREQuency  2k        /* order of magnitude character may be used
AM:INTernal:FUNCtion SINusoid
AM:STATe ON
```

4.4 Frequency sweeping

Configure frequency sweeping output, start frequency 1kHz,end frequency 100kHz,sweeping time 10s,logarithm sweeping, external trigger source, trigger sweeping once. The command is as follows:

```
FREQuency:STARt 1kHz;STOP 1E5     /* use semicolon to link two commands with
                                     same scale
```

```
SWEep:TIME 10s
SWEep:SPACing LOG;: TRIGger:SOURce EXTernal   /* use ;: to link two root
                                                              commands
*TRG                                   /* single trigger
```

4.5 Query error queue

Make two errors first then query them, the command as:

```
*CLS                      /*clear error queue
FREQu: 1kHz               /* the first level command is error，“u” is the abbreviation
VOLTage 8Vrms                       /*amplitude value beyond allowed value
SYSTem:ERRor?                       /* query the first error
Return“-101, First level command error”: the first level command is error
SYSTem:ERRor?         /*query the second error
Return“-204, Data out of range, value clipped to limit”:Data out of range, value clipped
                                                        to limit
SYSTem:ERRor?                       /* error queue is read out
Return “No error”:          there is no mistake
```