Generating a high-current pulse train

In this section:

Introduction

This example application demonstrates how to use the Model 2460 High-Current Interactive SourceMeter® instrument to generate high-current pulses (up to 7 A) and measure voltage.

Making pulsed I-V measurements may be necessary to avoid self-heating effects, device damage, or to observe the response of a device to a pulsed signal.

In this example, the trigger model controls the timing. The minimum pulse width and period may vary depending on the magnitude of the current. If you pulse low current (less than 1 µA), you may need to use a longer pulse width to get a settled voltage measurement. We recommend that you use an oscilloscope to verify the timing accuracy when you troubleshoot your code.

Equipment required

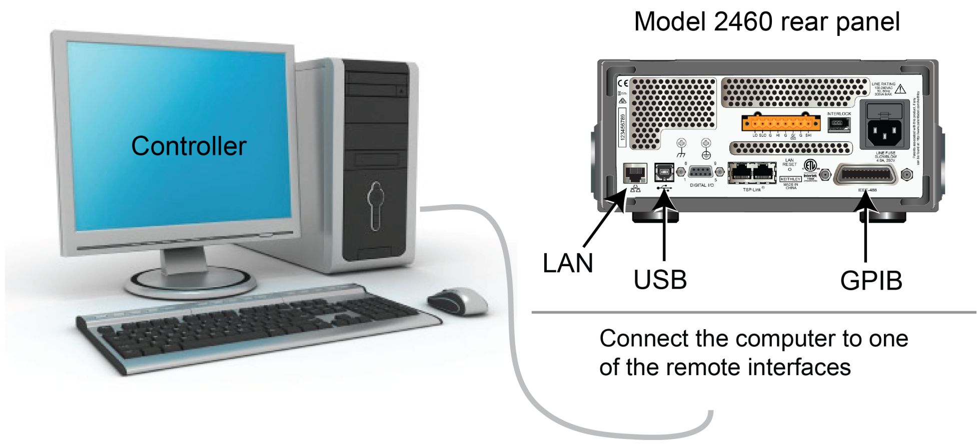
- One Model 2460 High-Current Interactive SourceMeter® instrument
- For front-panel connections, use four insulated banana cables such as the Keithley Instruments Model 8608 High-Performance Clip Lead Set (one set included with the Model 2460; you will need another set)
- For rear-panel connections, use one Model 2460-KIT Screw-Terminal Connector Kit (provided with the Model 2460), or you can use one set of Model 2460-BAN Banana Test Leads/Adapter Cables (with appropriate connections to the device)
- One GPIB, USB, or ethernet cable to connect the Model 2460 to a computer

Set up remote communications

This application is configured to run remotely. You can run this application from any of the supported communication interfaces for the instrument (GPIB, USB, or ethernet).

The following figure shows the rear-panel connection locations for the remote communication interfaces. For additional information about how to set up remote communications, see Remote communications interfaces (on page 3-1).

Figure 48: Model 2460 remote interface connections



Device connections

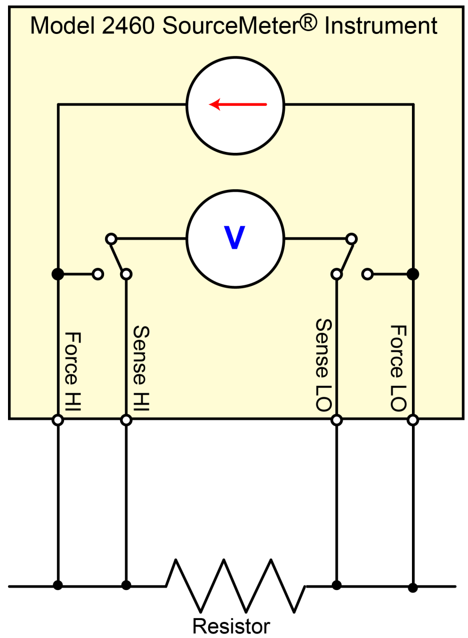
For best measurement accuracy and to eliminate the effects of test lead resistance when sourcing high current, connect the Model 2460 to the device under test (DUT) using the 4-wire sense method.

To use the 4-wire sense connection method:

- Connect the FORCE HI and SENSE HI leads to one end of the DUT.
- Connect the FORCE LO and SENSE LO leads to the other end of the DUT.
- Make the connections as close the DUT as possible to exclude the test-lead resistance from the measurement.

The following figure shows the schematic for outputting high-current pulses to a resistor.

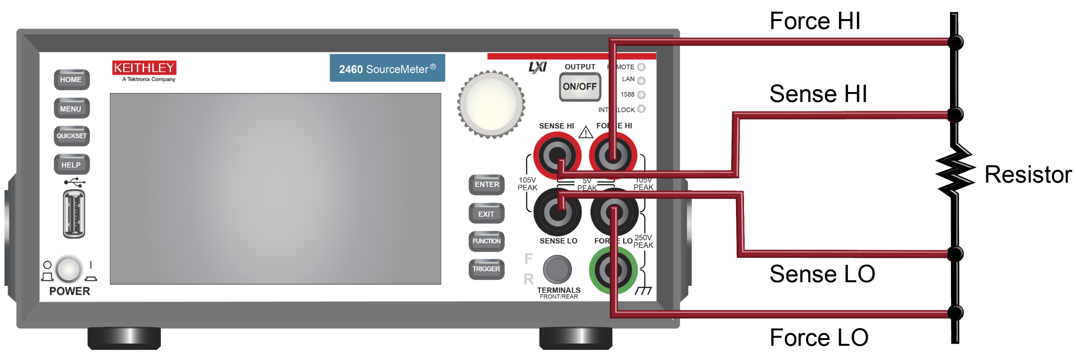
Figure 49: Model 2460 4-wire connections to a resistor



The following figures show the physical connections for the front and rear panels. The front-panel connections are safety banana jacks; the rear-panel connection is a screw terminal.

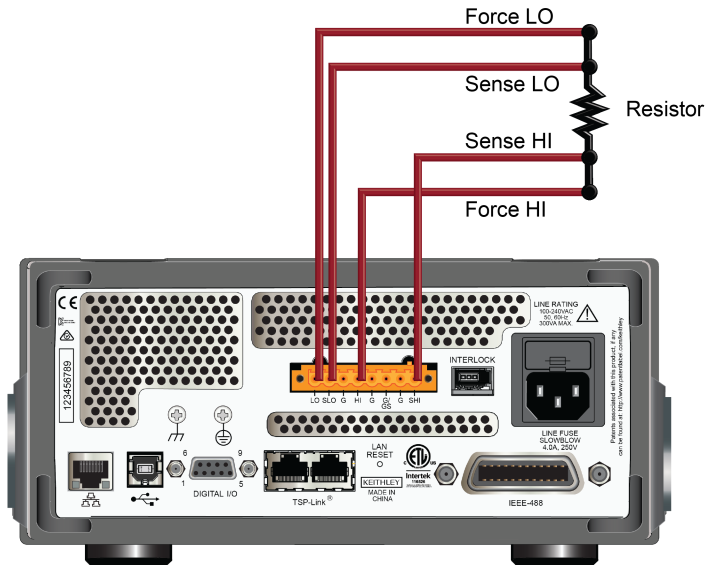
The figure below shows the front-panel connections. You can use four insulated banana cables that are rated to a maximum current of at least 7 A, such as two sets of the Keithley Instruments Model 8608 High-Performance Clip Lead Set (one set is included with the Model 2460).

Figure 50: Model 2460 4-wire connections to the front panel



The figure below shows the rear-panel connections. You can make these connections with either the Model 2460-KIT Screw-Terminal Connector Kit (included with the Model 2460) or a Model 2460-BAN Banana Test Leads/Adapter Cable with appropriate cabling.

Figure 51: Model 2460 4-wire connections to the rear panel



High-speed, high-current pulses from a remote interface

In this application, the Model 2460 is configured from a remote interface to generate a high-current pulse train using Test Script Processor (TSP®) commands. A voltage reading is taken during each pulse.

To ensure precise timing control, you will specify pulse parameters, configure the current source and voltage measurements, and set up the trigger model to control the timing of the pulses and measurements.

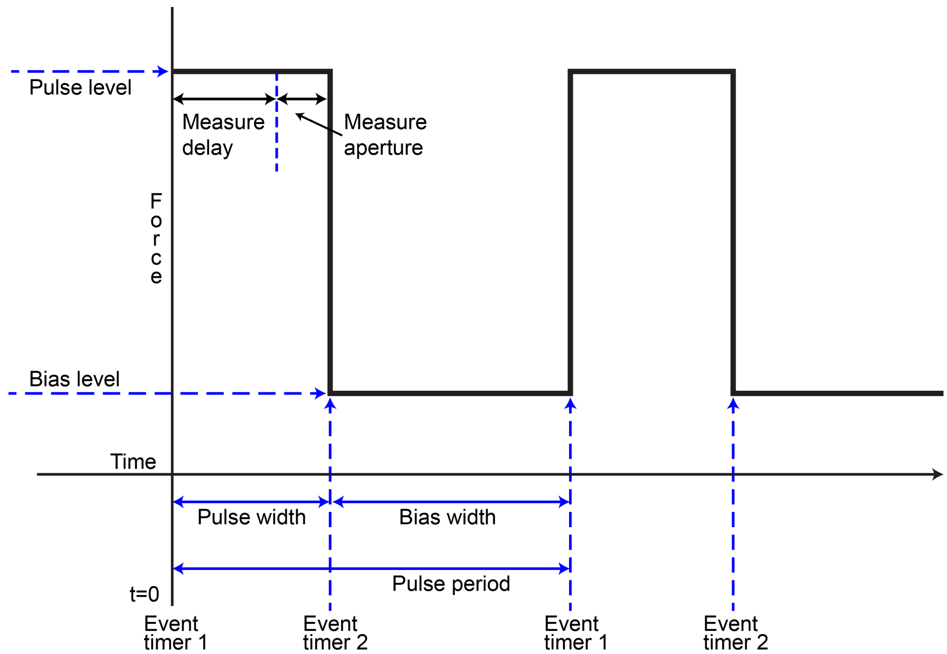
Define the pulse test parameters

The following pulse test parameters, which define the magnitude and timing of the pulse, are used in this example application. The actual values for these test parameters will vary depending on your specific application.

Test parameter	Value	Definition
`biaslevel`	`0 A`	Sets the offset current of the pulse train.
`pulselevel`	`6 A`	Sets the amplitude current of each pulse from zero (not from bias level).
`biaswidth`	`3e-3 s`	Sets the time at the bias level.
`pulsewidth`	`1e-3 s`	Sets the time at the amplitude level for each pulse.
`period`	`pulsewidth + biaswidth`	Specifies the amount of time it takes for the pulse to repeat itself (calculated value in this example resolves to 4 ms).
`points`	`10`	Sets the total number of pulses to generate.
`limit`	`7 V`	Sets the source limit level.
`measuredelay`	`pulsewidth -((1/localnode.linefreq) *smu.measure.nplc + 450e-6)`	Specifies the amount of time that the pulse remains at the pulse level before the measurement is taken. The value `450e-6` is overhead associated with converting a raw measurement into a reading.

The following figure illustrates these test parameters.

Figure 52: Timing diagram for the example



Set up the source and measure functions

To generate high-speed, high-current pulses and measure the voltage during each pulse, it is important to configure the measure and source functions appropriately.

To minimize measure time:

- Set the voltage measure range to a fixed range.
- Set the integration rate of the measurement to 0.01 NPLC (166.67 µs).
- Run autozero once to perform a one-time autozero operation before the start of the trigger model.

For more precise source times:

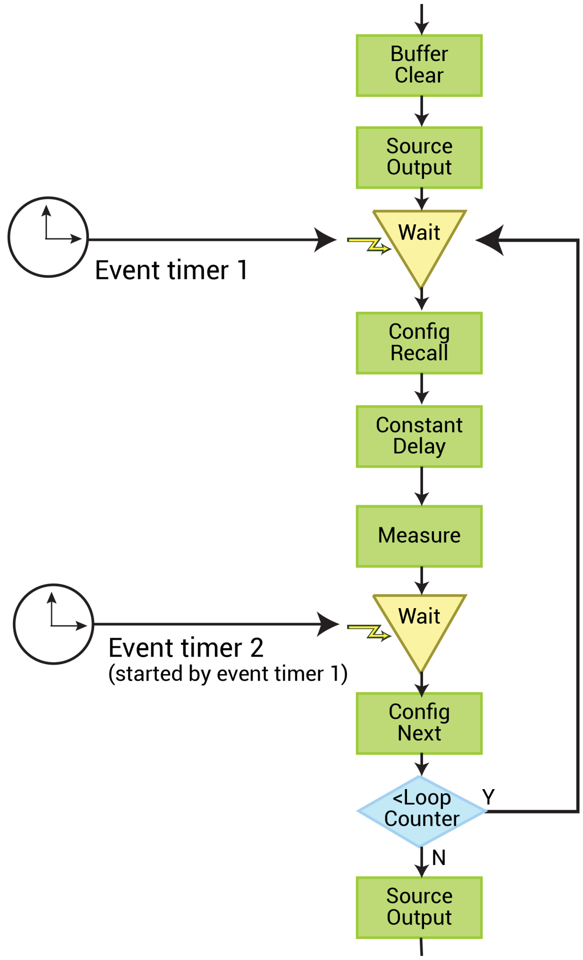
- Turn off the source readback feature.
- If you want to use source readback, you must allow enough time for two measurements: The voltage measurement and the current source measurement.
- Set the source delay time to zero.

In this example, the two values for the source (`pulselevel` and `biaslevel`) are stored in a source configuration list named `OutputList`. This configuration list is recalled by the trigger model.

Set up the trigger model using TSP commands

After you have set up the source and measure functions, configure the trigger model using the `trigger.model.setblock` commands. The blocks in the trigger model execute sequentially, as shown in the following figure.

Figure 53: TriggerFlow diagram used to generate a pulse train



The trigger model recalls the values in the source configuration list, triggers the voltage measurement during the pulse level, and provides accurate timing of the pulse width and period using two trigger event timers (event timer 1 and event timer 2).

Event timer 1 controls the pulse period, and event timer 2 controls the pulse width. For the most precise timing, it is important that the Model 2460 is waiting at the Wait blocks when the timer events are generated. To guarantee that the actual pulse width matches the programmed pulse width, the Config Recall, Constant Delay, and Measure blocks must complete before the event timer 2 delay elapses.

The Constant Delay block uses the `measuredelay` value to set the time delay between when the pulse level is output and the measurement is taken. The `measuredelay` value is a calculated delay that places the measurement as close to the end of the pulse as possible.

The Branch Counter block controls how many pulses are generated based on the value of the `points` parameter.

TSP commands to output a high-current pulse train

NOTE

The following TSP code is designed to be run from Keithley Instruments Test Script Builder (TSB). TSB is a software tool that is available from the Keithley Instruments website. You can install and use TSB to write code and develop scripts for TSP-enabled instruments. Information about how to use TSB is in the online help for TSB and in the “Introduction to TSP operation” section of the *Model 2460 Reference Manual*.

To use other programming environments, you may need to make changes to the example TSP code.

By default, the Model 2460 is configured to use the SCPI command set. You must select the TSP command set before sending TSP commands to the instrument.

To enable TSP commands:

1. Press the **MENU** key.
2. Under System, select **Settings**.
3. For Command Set, select **TSP**.
4. At the prompt to reboot, select **Yes**.

This following example code generates 10 pulses with a magnitude of 6 A and a pulse width of 1 ms. The pulse period is 4 ms and the load is 1 Ω. The voltage readings are stored in the default buffer, `defbuffer1`.

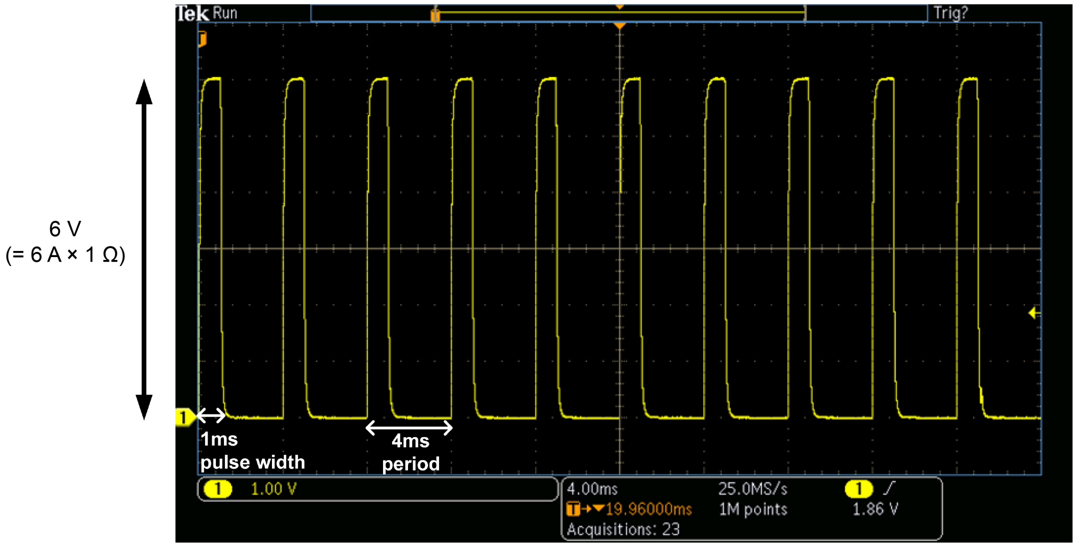
Send the following commands for this application:

```
reset()
--Set up the pulse parameters (user-specified).
biaslevel = 0
pulselevel = 6
biaswidth = 3e-3
pulsewidth = 1e-3
period = pulsewidth + biaswidth
points = 10
limit = 7
--[[Set the source to output current and create a source configuration list.
]]
smu.source.configlist.create("OutputList")
smu.source.func = smu.FUNC_DC_CURRENT
smu.source.readback = smu.OFF
--Set up the measure functions.
smu.measure.func = smu.FUNC_DC_VOLTAGE
smu.measure.nplc = 0.01
smu.measure.terminals = smu.TERMINALS_FRONT
smu.measure.range = limit
smu.measure.sense = smu.SENSE_4WIRE
measuredelay = pulsewidth -((1/localnode.linefreq)*smu.measure.nplc + 450e-6)
if measuredelay < 50e-6 then measuredelay = 50e-6 end
--[[Specify a source range large enough to fit both the bias and level.
]]
smu.source.range = math.max(math.abs(biaslevel), math.abs(pulselevel))
smu.source.delay = 0
smu.source.vlimit.level = limit
--[[Set to pulselevel (amplitude)and save the settings to the configuration list.
]]
smu.source.level = pulselevel
```

```
smu.source.configlist.store("OutputList")
--Set to biaslevel and save settings to the configuration list.
smu.source.level = biaslevel
smu.source.configlist.store("OutputList")
--Set up the timers.
--[[Set timer[1] to control the period of the pulse train. The effective count will
   be points because trigger.timer[1].start.generate = trigger.ON
]]
trigger.timer[1].reset()
trigger.timer[1].start.generate = trigger.ON
trigger.timer[1].delay = period
trigger.timer[1].count = points - 1
--Set timer[2] to control the width of the pulses.
trigger.timer[2].reset()
trigger.timer[2].start.stimulus = trigger.EVENT_TIMER1
trigger.timer[2].start.generate = trigger.OFF
trigger.timer[2].delay = pulsewidth
trigger.timer[2].count = 1
trigger.timer[2].enable = trigger.ON
--Set up the trigger model.
trigger.model.setblock(1, trigger.BLOCK_BUFFER_CLEAR)
trigger.model.setblock(2, trigger.BLOCK_SOURCE_OUTPUT, smu.ON)
trigger.model.setblock(3, trigger.BLOCK_WAIT, trigger.EVENT_TIMER1)
trigger.model.setblock(4, trigger.BLOCK_CONFIG_RECALL, "OutputList")
trigger.model.setblock(5, trigger.BLOCK_DELAY_CONSTANT, measuredelay)
trigger.model.setblock(6, trigger.BLOCK_MEASURE)
trigger.model.setblock(7, trigger.BLOCK_WAIT, trigger.EVENT_TIMER2)
trigger.model.setblock(8, trigger.BLOCK_CONFIG_NEXT, "OutputList")
trigger.model.setblock(9, trigger.BLOCK_BRANCH_COUNTER, points, 3)
trigger.model.setblock(10, trigger.BLOCK_SOURCE_OUTPUT, smu.OFF)
--Start the trigger model.
defbuffer1.clear()
smu.measure.autozero.once()
trigger.model.initiate()
delay(0.001)
trigger.timer[1].enable = trigger.ON
waitcomplete()
```

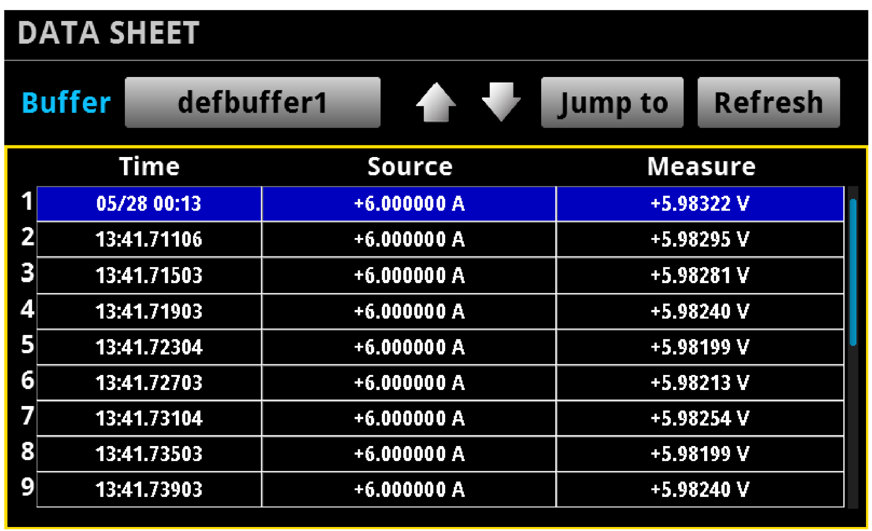
In this example, the output is measured across the load using an oscilloscope. The following figure shows the pulse train displayed on the oscilloscope. The figure shows that the trigger model event timers generate events at uniform time intervals. This results in precise timing control.

Figure 54: Oscilloscope view of ten 1 ms, 6 A pulses into a 1 Ω load with a period of 4 ms



The voltage measurements made during each pulse are saved in the buffer. You can send this data to a remote interface using an appropriate `print` or `printbuffer` command, or it can be viewed from the front-panel Data Sheet (press the **MENU** key and select **Sheet**). The following figure shows a Data Sheet that contains measurement values taken in this example application.

Figure 55: Data sheet showing measurement results



	Time	Source	Measure
1	05/28 00:13	+6.000000 A	+5.98322 V
2	13:41.71106	+6.000000 A	+5.98295 V
3	13:41.71503	+6.000000 A	+5.98281 V
4	13:41.71903	+6.000000 A	+5.98240 V
5	13:41.72304	+6.000000 A	+5.98199 V
6	13:41.72703	+6.000000 A	+5.98213 V
7	13:41.73104	+6.000000 A	+5.98254 V
8	13:41.73503	+6.000000 A	+5.98199 V
9	13:41.73903	+6.000000 A	+5.98240 V

Note that the timestamps in the Time column show that the readings were taken precisely 4 ms apart, as configured in the trigger model.