Section 8

Grading and binning resistors

In this section:

Introduction

This application example demonstrates how to use the DMM6500 to perform benchtop binning operations. It uses the trigger model and digital I/O to control external component-handling devices.

The DMM6500 can do simple pass-or-fail testing and grading and sorting. Grading resistors is a common application that is done by monitoring multiple limits until the first failure is received. Binning resistors is also common but unlike grading, involves monitoring limits until the first pass is received.

Refer to the "Grading and binning resistors" topic in the *Model DMM6500 Reference Manual* for more information.

Equipment required

- One DMM6500
- One Model 7700 20-channel differential multiplexer module
- One computer setup for communication with the instrument
- One device or component to be tested

Device connections

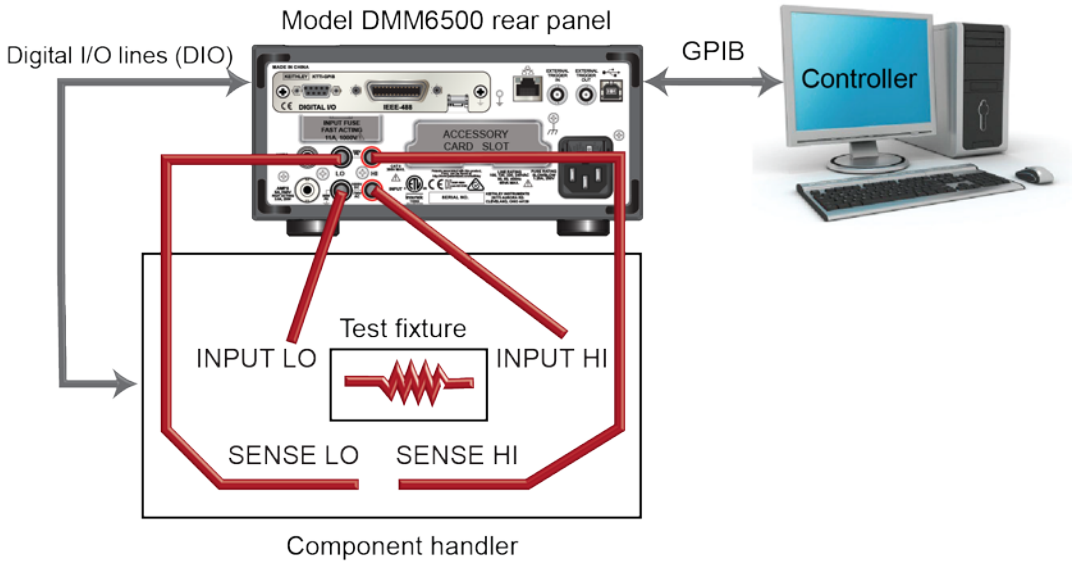
This example application uses a DMM6500 to perform benchtop binning operations. Output signals (grading results) are sent from the instrument to the component handler, which bins the devices.

The figure below shows the rear-panel connections from the DMM6500 to the test fixture and the digital lines to the component handler. The optional GPIB communication card connects the controller with the component handler.

NOTE

Digital lines and GPIB communications require the KTTI-GPIB communications accessory card to be installed in the instrument.

Figure 40: Device connections for component binning



WARNING

To prevent electric shock, test connections must be configured such that the user cannot come in contact with test leads or any device under test (DUT) that is in contact with the conductors. It is good practice to disconnect DUTs from the instrument before powering the instrument. Safe installation requires proper shields, barriers, and grounding to prevent contact with test leads.

There is no internal connection between protective earth (safety ground) and the LO terminals of the DMM6500. Therefore, hazardous voltages (more than 30 V_{rms}) can appear on LO terminals. This can occur when the instrument is operating in any mode. To prevent hazardous voltage from appearing on the LO terminals, connect the LO terminal to protective earth (safety ground) if your application allows it. You can connect the LO terminal to the chassis ground terminal on the front panel or the chassis ground screw terminal on the rear panel. Note that the front-panel terminals are isolated from the rear-panel terminals. Therefore, if you are using the front-panel terminals, ground to the front-panel LO terminal. If using the rear-panel terminals, ground to the rear panel LO terminal. Failure to follow these guidelines can result in injury, death, or instrument damage.

Resistor grading and binning test

This resistance-grading application uses limit tests to inspect a single resistor under test against multiple limits until the first failure occurs. When the resistor fails, it is placed into a designated resistance tolerance bin as defined by the limits.

Resistors are placed into bins based on the bit patterns that are assigned to the limits. In this example, the DMM6500 GradeBinning trigger model template is used to simplify the application. This trigger model template grades components, resistors in this case, into four tolerance levels (for example, 20 %, 10 %, 5 %, and 1 %) as defined by limits 1 to 4. A single spot measurement is inspected against multiple limits, which tighten progressively around the same nominal value. Since there is no need to continue limit checking once the appropriate tolerance level for a resistor under test is determined, this application typically immediately bins the tested resistors.

Because the limits are inspected in ascending numeric order, the measured resistance is checked first against Limit 1, which is the 20 % limit. If the resistor fails this limit inspection, its resistance value is outside of the 20 % tolerance band and the trigger model outputs the Limit 1 Fail Pattern, which causes the component handler to place the resistor in the Limit 1 fail bin (20 % fail bin).

If a resistor passes the 20 % limit test, the resistance value is checked against Limit 2, which is the 10 % limit value. If the resistor fails this limit inspection, the resistance is outside of the 10% tolerance band. The trigger model outputs the Limit 2 Fail Pattern, which causes the component handler to place the resistor in the Limit 2 fail bin (10% fail bin).

If a resistor passes the 10 % limit test, the resistance value is checked against Limit 3, which is the 5 % limit value, and so on. If a resistor passes all the limit tests, the trigger model outputs the Overall Pass Bit Pattern, which causes the component handler to place the resistor in the all pass bin.

For this example, the same fail pattern is assigned to both the lower and upper bounds of the limits. Therefore, a fail bin contains resistance values in the range R-P % to R+P %. P in this example is 20, 10, 5, or 1. You can assign different bit patterns for different limit values.

For this application, you will:

- Reset the instrument.
- Select the 4-wire resistance function.
- Enable offset compensation.
- Set auto zero to Once.
- Set up digital I/O lines one to four as outputs to component handler.
- Set up digital I/O line five for trigger-model control, detecting the trigger as the start-of-test input.
- Set up digital I/O line six as the end-of-test output notification.
- Initiate the GradeBinning trigger-model template.
- Display "Test Completed" message on the front panel.

Trigger model template: grade and binning test

The trigger model template contains the settings for the number of components, digital I/O, and limits. The command parameters for the template are described in the following command and table.

SCPI command usage:

```
:TRIGger:LOAD "GradeBinning", <components>, <startInLine>, <startDelay>,
   <endDelay>, <limit1High>, <limit1Low>, <limit1Pattern>, <allPattern>,
   <limit2High>, <limit2Low>, <limit2Pattern>, <limit3High>, <limit3Low>,
   <limit3Pattern>, <limit4High>, <limit4Low>, <limit4Pattern>, "<bufferName>"
```

TSP command usage:

```
trigger.model.load("GradeBinning", components, startInLine, startDelay, endDelay,
   limit1High, limit1Low, limit1Pattern, allPattern, limit2High, limit2Low,
   limit2Pattern, limit3High, limit3Low, limit3Pattern, limit4High, limit4Low,
   limit4Pattern, bufferName)
```

Parameter list

components	100
startInLine	Digital I/O line 5
startDelay	100 ms
endDelay	100 ms
limit1High	R = 100 Ω, P = 20%, 100+20% = 120 Ω
limit1Low	R = 100 Ω, P = 20%, 100-20% = 80 Ω
limit1Pattern	Bin 1 Fail Pattern 15: drive all digital I/O lines high (1111)
allPattern	All Pass Pattern 4: drive line 3 high (0100)
limit2High	R = 100 Ω, P =10%, 100+10% = 110 Ω
limit2Low	R = 100 Ω, P =10%, 100-10% = 90 Ω
limit2Pattern	Bin 2 Fail Pattern 1: drive line 1 high (0001)
limit3High	R = 100 Ω, P =5%, 100+5% = 105 Ω
limit3Low	R = 100 Ω, P =5%, 100-5% = 95 Ω
limit3Pattern	Bin 3 Fail Pattern 2: drive line 2 high (0010)
limit4High	R = 100 Ω, P =1%, 100+1% = 101 Ω
limit4Low	R = 100 Ω, P =1%, 100-1% = 99 Ω
limit4Pattern	Bin 4 Fail Pattern 3: drive line 1 and 2 high (0011)
bufferName	The reading buffer is set to `bufferVar`

Using SCPI commands

This sequence of SCPI commands grades resistors into bins based on the measured accuracy.

You may need to make changes so that this code will run in your programming environment.

Send the following commands for this example application:

Commands	Descriptions
`*RST`	• Reset the DMM6500
`:TRAC:MAKE "bufferVar", 1000000`	• Make a buffer named `bufferVar` with a capacity of 1,000,000 readings
`:TRAC:CLE "bufferVar"`	• Clear `bufferVar`
`:SENS:FUNC "FRES"`	• Set instrument to measure 4-wire resistance
`:SENS:FRES:NPLC 1`	• Set the number of power line cycles (NPLC) to 1
`:SENS:AZER:ONCE`	• Enable autozero once
`:SENS:FRES:OCOM ON`	• Enable offset compensation for more accurate resistance readings

`:DIGital:LINE1:MODE DIG, OUT`
`:DIG:LINE2:MODE DIG, OUT`
`:DIG:LINE3:MODE DIG, OUT`
`:DIG:LINE4:MODE DIG, OUT`
	• Configure digital I/O lines 1 through 4 as digital outputs; these are used to output binning code to the component handler

`:DIG:LINE1:STAT 0`
`:DIG:LINE2:STAT 0`
`:DIG:LINE3:STAT 0`
`:DIG:LINE4:STAT 0`
	• Set the states of digital I/O lines 1 through 4 to bit low

`:DIG:LINE5:MODE TRIG, IN`	• Configure digital I/O line 5 as trigger input to detect start-of-test trigger
`:TRIGger:DIG5:IN:EDGE FALL`	• Set trigger detector to detect falling edge on digital I/O line 5
`:DIG:LINE6:MODE TRIG, OUT` `:TRIGger:DIGital6:OUT:LOGic NEG`	• Configure digital I/O line 6 as a trigger output used to send end-of-test trigger with negative logic and output pulse width of 10 μs
`:TRIG:DIG6:OUT:PULSewidth 10e-6` `:TRIG:DIG6:OUT:STIMulus NOT1`	• The trigger pulse occurs when the Notify block generates an event
`:TRIGger:LOAD "GradeBinning", 100, 5, .1, .1, 120, 80, 15, 4, 110, 90, 1, 105, 95, 2, 101, 99, 3, "bufferVar"`	• Define the GradeBinning trigger model template
`INIT`	• Initiate the trigger model
`*WAI`	
`:DISP:SCR SWIPE_USER`	• Set the front-panel display to the USER swipe screen
`:DISP:USER1: TEXT "Test Completed"`	• Display "Test Completed" when the binning test is complete

Using TSP commands

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 webpage on the Tektronix 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 *DMM6500 Reference Manual*.

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

By default, the DMM6500 uses 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 sequence of TSP commands will grade resistors into established bins of accuracy. After the code executes, the data is displayed in the Instrument Console of Test Script Builder.

Send the following commands for this example application:

```
-- Reset the instrument to default settings
reset()
-- Create a user-defined reading buffer that can store up to 1 million readings
bufferVar = buffer.make(1000000)
bufferVar.clear()
--Set the measure function to 4-wire resistance
dmm.measure.func = dmm.FUNC_4W_RESISTANCE
-- Set the number of power line cycles 1 PLC
dmm.measure.nplc = 1
-- Immediately update autozero reference measurements and then disable the autozero
   function
dmm.measure.autozero.once()
-- Enable offset compensation for more accurate resistance reading
dmm.measure.offsetcompensation.enable = dmm.ON
-- Configure digital I/O lines 1 through 4 as digital outputs. These I/O lines are
   used to output binning code to component handler
digio.line[1].mode = digio.MODE_DIGITAL_OUT
digio.line[2].mode = digio.MODE_DIGITAL_OUT
digio.line[3].mode = digio.MODE_DIGITAL_OUT
digio.line[4].mode = digio.MODE_DIGITAL_OUT
-- Clear digital I/O lines to 0
digio.line[1].state = digio.STATE_LOW
digio.line[2].state = digio.STATE_LOW
digio.line[3].state = digio.STATE_LOW
digio.line[4].state = digio.STATE_LOW
-- Configure digital I/0 line 5 as a trigger input used to detect
-- the start-of-test trigger from the component handler
digio.line[5].mode = digio.MODE_TRIGGER_IN
-- Set trigger detector to detect falling edge
trigger.digin[5].edge = trigger.EDGE_FALLING
```

```
-- Configure digital I/0 line 6 as a trigger output used to send
-- an end-of-test trigger to the component handler
digio.line[6].mode = digio.MODE_TRIGGER_OUT
-- Output a falling edge trigger
trigger.digout[6].logic = trigger.LOGIC_NEGATIVE
-- Set width of output trigger pulse to 10 us
trigger.digout[6].pulsewidth = 10E-6
-- Trigger pulse is output when the Notify Block generates an event
trigger.digout[6].stimulus = trigger.EVENT_NOTIFY2
--Load Component Binning trigger model template
trigger.model.load("GradeBinning", 100, 5, .1, .1, 120, 80, 15, 4, 110, 90, 1, 105,
   95, 2, 101, 99, 3, bufferVar)
--Initiate trigger model and wait until finished
trigger.model.initiate()
waitcomplete()
--Display on front panel USER swipe screen after binning test completes
display.changescreen(display.SCREEN_USER_SWIPE)
display.settext(display.TEXT1, "Test Completed")
```