Section 9

Measuring power using digitizing and TSP-Link

In this section:

Introduction

This application example demonstrates how to configure two DMM6500 instruments to use TSP-Link to measure the power consumed by a Bluetooth® low energy (BLE) device.

For this example, one DMM6500 measures digitized voltage while the other instrument measures digitized current. Using TSP-Link, these measurements are made simultaneously and the results communicated between the two instruments. Using TSP scripting, the average power consumption during the test period is calculated using the following formula in which P_{ave} is the average power and n is the number of points: $P_{ave} = \frac{I_1V_1 + I_2V_2 + \cdots + I_nV_n}{n}$. This formula calculates power at each point along the waveform by multiplying corresponding currents and voltages, adding them together, and dividing them by the total number of data points.

This application example multiplies measured current and measured voltage, point-by-point and averages those power calculations to find the average power usage. Average power measurements provide insight into device performance. This is more accurate than simply multiplying the average current by the average voltage to find average power.

Some applications calculate power by measuring current measurements and then multiplying them by the known battery voltage. The advantage of digitizing both current and voltage simultaneously is increased accuracy because the exact voltage is known at each current measurement.

These measurements are especially important when the device requires a battery to operate, because minimizing power consumption maximizes battery life.

Equipment required

This application requires the following equipment:

- Two DMM6500 instruments
- Two KTTI-TSPLink communication and digital I/O accessory cards
- One computer set up for communication with the instrument
- One ethernet crossover cable
- Several insulated banana cables (such as the Model 1757 Standard Test Lead Kit)
- One device or component to be tested

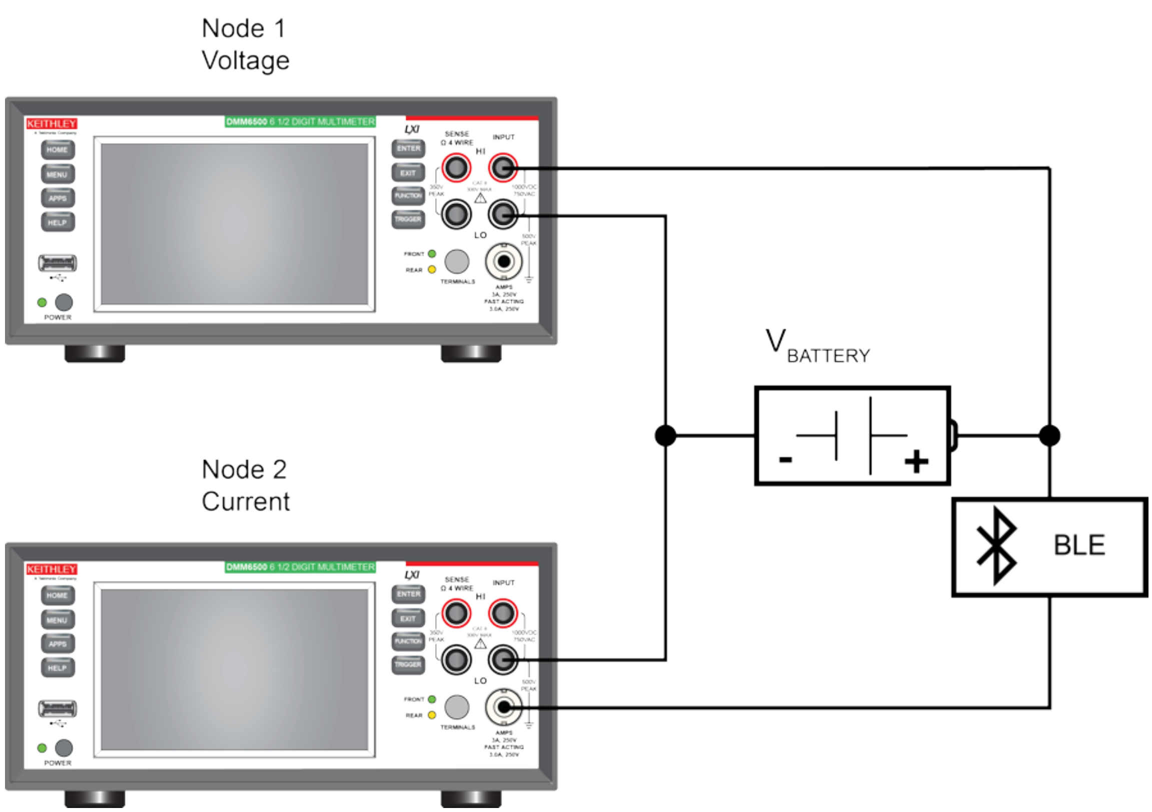
Device connections

1. This application requires two KTTI-TSPLink communication and digital I/O accessory cards. You insert one card into the accessory card slot on the rear panel of each instrument. Connected the instruments with a crossover cable connected to the KTTI-TSPLink communication cards.

Figure 41: Two DMMs with TSP-Link

2. To connect to the Bluetooth low energy device, connect the test leads of the instrument measuring current in series with the battery of the device. The test leads of the instrument measuring voltage are connected in parallel with the battery.

Figure 42: Two Nodes Measuring Current and Voltage



3. Connect the computer to the DMM6500 that will be set to node one.

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. Failure to recognize and observe normal safety precautions could result in personal injury or death.

Measuring power using digitizing and TSP-Link

For this application, you will:

- Set the TSP-Link nodes for the DMM6500 instruments to one and two.
- Reset the instruments.
- Initialize TSP-Link.
- Configure the input and output TSP-Link triggers for the DMM6500 instruments.
- Set up the digitize function.
- Set up trigger models on both instruments.
- Initiate measurements on both DMM6500 instruments.
- Use buffer statistics to find the average current and voltage.
- Find average power of a Bluetooth device.
- Print the results to the USER swipe screen.

Setting up the nodes for TSP code

Before executing the TSP code, you must set up the nodes on the instruments and configure the TSP-Link network.

To set up TSP-Link on the DMM6500:

1. Press the **MENU** key.
2. Under System, select **Communication**.
3. Select the **TSP-Link** tab.
4. Set the Node to **1** on the instrument that is used to measure voltage
5. Repeat these steps on the other instrument and set the Node to **2** on the instrument that is used to measure current.
6. Select **Initialize** on each instrument.

NOTE

The communication from the computer is made directly to the first DMM6500 (node 1) in the following TSP code. This makes the first DMM6500 the master in this TSP-Link network, with the other DMM6500 as the subordinate. You can change the master to be the other DMM6500. However, you must modify the code and how the TSP-Link network is initialized. The master node does not require the `node[x].` prefix.

If you need to improve program execution speed, remove all the master node prefixes in the following TSP code.

Using SCPI commands

This example cannot be replicated in SCPI code because TSP-Link-related events and commands are only available through the TSP command language.

However, you can use other instrument trigger interfaces, such as digital I/O or external trigger I/O, to replace TSP-Link and achieve similar operation.

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](tek.com/keithley) (tek.com/keithley). 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 measure power using digitizing and TSP-Link . After the code executes, the data is displayed in the Instrument Console of Test Script Builder.

Send the following commands for this example application:

```
-- Initiate the tsp-link network
tsplink.initialize()
-- Set a delay of 0.5 seconds.
delay(0.5)
-- Reset master instrument at node 1
node[1].reset()
-- Set up TSP-link trigger line 1 to trigger the subordinate node digitizer.
node[1].tsplink.line[1].mode = tsplink.MODE_TRIGGER_OPEN_DRAIN
node[1].trigger.tsplinkout[1].stimulus = trigger.EVENT_NOTIFY1
-- Set up digitize voltage function settings.
node[1].dmm.digitize.func = dmm.FUNC_DIGITIZE_VOLTAGE
node[1].dmm.digitize.samplerate = 5000
-- Set up digitize voltage range based on voltage applying to the BLE device.
node[1].dmm.digitize.range = 10
-- Set up reading buffers.
node[1].defbuffer1.capacity = 50000
-- Set up trigger model.
node[1].trigger.model.setblock(1, trigger.BLOCK_NOTIFY, trigger.EVENT_NOTIFY1)
node[1].trigger.model.setblock(2, trigger.BLOCK_WAIT, trigger.EVENT_TSPLINK1)
node[1].trigger.model.setblock(3, trigger.BLOCK_MEASURE_DIGITIZE, defbuffer1,
   50000)

-- Reset instrument at node 2.
node[2].reset()
-- Set up TSP-link trigger line 1 to receive a trigger from master node.
node[2].tsplink.line[1].mode = node[2].tsplink.MODE_TRIGGER_OPEN_DRAIN
-- Set up digitize current function settings.
node[2].dmm.digitize.func = node[2].dmm.FUNC_DIGITIZE_CURRENT
node[2].dmm.digitize.samplerate = 5000
-- Set up digitize current range based on maximum current the BLE device can draw.
```

```
node[2].dmm.digitize.range = 1
-- Set up reading buffers.
node[2].defbuffer1.capacity = 50000
-- Set up trigger model.
node[2].trigger.model.setblock(1, node[2].trigger.BLOCK_WAIT,
   node[2].trigger.EVENT_TSPLINK1)
node[2].trigger.model.setblock(2, node[2].trigger.BLOCK_MEASURE_DIGITIZE,
   defbuffer1, 50000)

-- Show graph of measurements on swipe screens.
node[1].display.changescreen(node[1].display.SCREEN_GRAPH_SWIPE)
node[2].display.changescreen(node[2].display.SCREEN_GRAPH_SWIPE)
delay(1.0)
-- Initiate trigger model on both instruments.
node[2].trigger.model.initiate()
trigger.model.initiate()
-- Wait for test to complete.
waitcomplete()

-- Retrieve buffer statistics.
voltage_buffer = node[1].defbuffer1
voltage_stats = node[1].buffer.getstats(voltage_buffer)
avgVolt = voltage_stats.mean
print(avgVolt .. " Volts")

current_buffer = node[2].defbuffer1
current_stats = node[2].buffer.getstats(current_buffer)
avgCurr = current_stats.mean
print(avgCurr .. " Amps")

-- Print results to the USER swipe screen.
node[1].display.changescreen(display.SCREEN_USER_SWIPE)
node[1].display.settext(display.TEXT1, "AVG V: " .. string.format("%.2e", avgVolt)
   .. " V")
node[1].display.settext(display.TEXT2, "Average Power: Calculating... ")
node[2].display.changescreen(display.SCREEN_USER_SWIPE)
node[2].display.settext(display.TEXT1, "AVG I: " .. string.format("%.2e", avgCurr)
   .. " A")
node[2].display.settext(display.TEXT2, "Average Power: Calculating... ")

-- Calculate power using reading index-based method.
power_total = 0
num_readings = current_buffer.n

-- Iterate through each current and voltage reading, and calculate power.
for i = 1, num_readings do
            current = current_buffer.readings[i]
            voltage = voltage_buffer.readings[i]

            -- Keep track of the total power
            power_total = power_total + current*voltage
end

-- Find average power by dividing total power by the number of readings.
average_power = power_total / num_readings
print(average_power .. " Watts")
```

```
--Print results to USER swipe screen.
node[1].display.changescreen(display.SCREEN_USER_SWIPE)
node[1].display.settext(display.TEXT2, "Average Power: ".. string.format("%8f",
   average_power) .. " W")
node[2].display.settext(display.TEXT2, "Average Power: ".. string.format("%8f",
   average_power) .. " W")
```

Results

The voltage and current waveforms captured on the DMM6500 clearly show the power consumption resulting from the use of the Bluetooth device. You can identify the transmission state of the device by the areas of high current consumption and the visible drop in voltage from the battery. Because these measurements are triggered within 2 µs of each other, the voltage and current data is synced nearly point for point.

You can expand on this example by importing the data from each of the instruments' buffers to a computer and analyzing the data more closely.

Figure 43: Master node measuring voltage

Figure 44: Master node voltage waveform

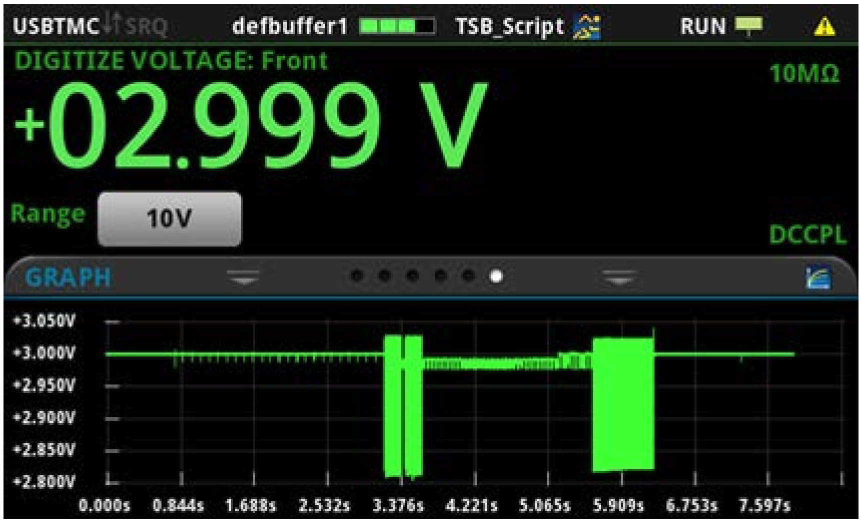


Figure 45: Subordinate node measuring current

Figure 46: Subordinate node current waveform