KEITHLEY   A GREATER MEASURE OF CONFIDENCE

Plotting Data in Real Time with Model 4200-SCS User Test Modules

The Keithley Interactive Test Environment (KITE) software on the Model 4200-SCS Semiconductor Characterization System includes two types of test modules: Interactive Test Modules (ITMs) and User Test Modules (UTMs). Users can choose which type of module to use based on their preferences and the type of test desired. ITMs offer a very convenient way to set up an I-V test—all that's required is a simple point-and-click of the mouse. ITMs also allow plotting data in real time (i.e., the graph drains as the test progresses). On the other hand, with a UTM, the user can program a test using C language and the instrument libraries provided. UTMs are designed to provide the maximum flexibility for controlling a 4200-SCS and other external instruments, so users can create just about any desired test.

Until recently, the UTMs in KTE software were limited because they didn't support plotting data in real time. Being forced to wait until the test is finished before viewing the results can be extremely inconvenient when long test times are involved. This Tech Note describes how Version 5.0 of the software makes it possible to plot the data from UTMs in real time.

Three new functions have been added to the existing Linear parametric Test (LPT) library in KTE version 5.0: "PostDataDouble(),""PostDataInt()," and "Post-DataString()." The protocols of the three functions are:

```
PostDataDouble(char *, double *)

PostDataInt(char *, int *)

PostDataString(char *, char *)
```

These functions are used immediately after one measurement point is finished and a data value is assigned to the corresponding output variable. They will then transfer the data value from memory to the Data sheet in the UTM and get it plotted on the graph. "PostDataDouble()" is used to transfer a "double" type data point from memory back to the Data sheet. "PostDataInt()" is used to transfer a "integer" type data point from memory back to the Data sheet. "PostDataString()" is used to transfer a string from memory back to the Data sheet. Each function should be used according to the type defined for the corresponding output parameter. For example, if the output parameter is defined as "double," then "PostDataDouble()" function should be used to bring data back to the Data sheet, then plot it in real time.

The first parameter in all three functions is the variable name, defined as "char *." For example, if the output variable name is DrainI, then "DrainI" (with quotes) should be used in the first parameter. The second parameter is the value of the variable to be transferred. For example, if DrainI[10] should be transferred, then one should call "PostDataDouble("DrainI," &DrainI[10])."

Plotting Data in Real Time with
Model 4200-SCS User Test Modules

When using the new functions to transfer data into the Data sheet in real time, make sure the data is already located in the 4200-SCS's memory. That means "sweep" type of measurements are unsuitable for real-time transfer, because data is not ready until the sweep is finished. The following examples illustrate how to enable real-time plotting in UTMs.

1. I-V measurements using the 4200-SCS's SMUs

As mentioned previously, if real-time plotting is desired, sweep type measurements can't be used. Therefore, sweep-related functions, such as "smeasx()," "sintgx()," and "sweepx()," can't be used. Instead, one should form a "for" loop in the program to perform a point-by-point measurement. In the following example, SMU1 is used to force voltage, then measure current. Programmed voltage and measured current are then output for plotting.

Test sequence without real-time plotting:

```
#include "keithley.h"

int IV(double startv, double stopv, int numpoint, double
* V, int Vsize, double * I, int Isize)

{
    // error checking
    if ((numpoint != Vsize) ||(numpoint != Isize))
    return –1;


    rtfary(V); // return force array of Voltage
    sintgi(SMU1, I); // measure current
    // setup sweep with 0.01 second sweep delay and then
    trigger the measurement
    sweepv(SMU1, startv,stopv,numpoint–1,0.01);


    return (OK);
}
```

Test sequence with real-time plotting:

```
#include "keithley.h"


int IV(double startv, double stopv, int numpoint, double
* V, int Vsize, double * I, int Isize)
{
    int index;
    double stepv;
```

```
// error checking
if ((numpoint != Vsize) ||(numpoint != Isize))
    return –1;


    //calculate stepv
if (numpoint != 1)
    stepv = (stopv – startv)/(numpoint –1);
// measurement loop
for (index = 0; index < numpoint; index ++)
{
    // calculate voltage array
    V[index] = startv + stepv*index;
    forcev(SMU1, V[index]); // force voltage
    intgi(SMU1,&I[index]); // measure current
    // transfer V data
    PostDataDouble("V," &V[index]);
    // transfer I data
    PostDataDouble("I," &I[index]);
}
return (OK);
}
```

2. Taking measurements using external instruments

The same idea applies to measurements made using external instruments. A trigger—sweep—data method can't be used because real-time plotting requires real-time data and with this method the data is not available until the sweep is complete. The following example illustrates getting real-time data from external instruments.

Test sequence without real-time plotting:

```
initialize instrument;
setup sweep;
setup trigger;
trigger measurement;
serial poll the instrument until measurement is done;
retrieve data from instrument;
post measurement clean up;
done;
```

Plotting Data in Real Time with
Model 4200-SCS User Test Modules

Test sequence with real-time plotting:

```
initialize instrument;

setup single point measurement mode;

setup single trigger mode;

turn on output;

calculate number of data point;

// measurement loop

for (index = 0; index < numpoint; index ++)

{

     setup force value in each step;

     take measurement;

     // transfer data for real time plotting

     PostDataDouble("Variable Name,"
&MeasureArray[index]);

}

post measurement clean up;

done;
```

KEITHLEY