Purpose

This document provides an example of using the Matlab helper scripts "Import_HP54600B_ScopeData.m", "CurveTracerTest_Id_vs_Vgs.m" and "CurveTracerTest_Id_vs_Vds2.m".
Note: The script "Import_SADF_Data.m" is used in exactly the same manner as "Import_HP54600B_ScopeData.m".

The following LabView VI's were used to generate the Simple ASCII Data Format (SADF) files that were subsequently imported into and processed with Matlab.

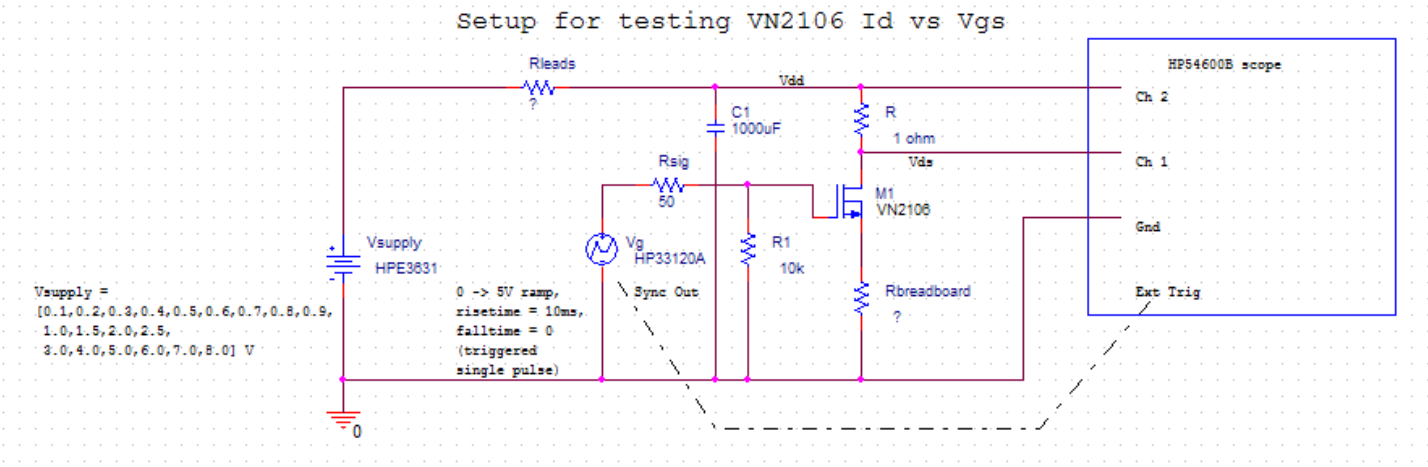
- Power MOSFET Vgs Sweep.vi
  Uses the Triggered_Ramp, Setup_Output, Send Software Trigger, and HP54600B_ReadTraces VI's to measure Id vs Vgs of the VN2106 MOSFET. This VI does not plot nicely – that was done with Matlab using the helper script. See next pages.

- Power MOSFET Vds Sweep.vi
  Uses the Triggered_Rectangular_Pulse, Setup_Output, Send Software Trigger, and HP54600B_ReadTraces VI's to measure Id vs Vds of the VN2106 MOSFET. This VI does not plot nicely – that was done with Matlab using the helper script. See next pages.

The context is the I/V characterization of a VN2106 power MOSFET. This device has a high enough gain to make it impossible to characterize with the same method used for the CD4007 MOSFET devices. It simply draws too much current (> 1 ampere). If steady-state measurements were made at those current levels and any appreciable Vds, the device would burn up. This problem led to the development of LabView VI's to generate single ramp and rectangular pulses to minimize the time the device would spend in a conducting state. In addition, the HP54600B oscilloscope was used to capture the voltage drop across a 1 ohm current sense drain resistor while the gate voltage was ramped (in the case of a Vgs sweep) or pulsed (in the case of a Vds sweep).

Since the resulting arrays of measurements points from the oscilloscope needed some mathematical operations performed, this provided an excellent opportunity to provide scripts and functions enabling the import of SADF data into Matlab.

The example test procedure begins on the next page. Note: the noise in the waveforms is due to oscilloscope quantization at the volts/div settings necessary to keep the waveforms always on the screen.



Here the Labview Triggered_Ramp VI is used to send a single ramp waveform to the VN2106 MOSFET gate for each value of Vds. The Labview E3631 VI sets the Vds voltages. Since the E3621 power supply cannot put out more than 1 amp from the +/- 25V outputs, C1 is used to store some charge locally, and the ramp is kept short. Nevertheless, 5V is used as the maximum Vgs.

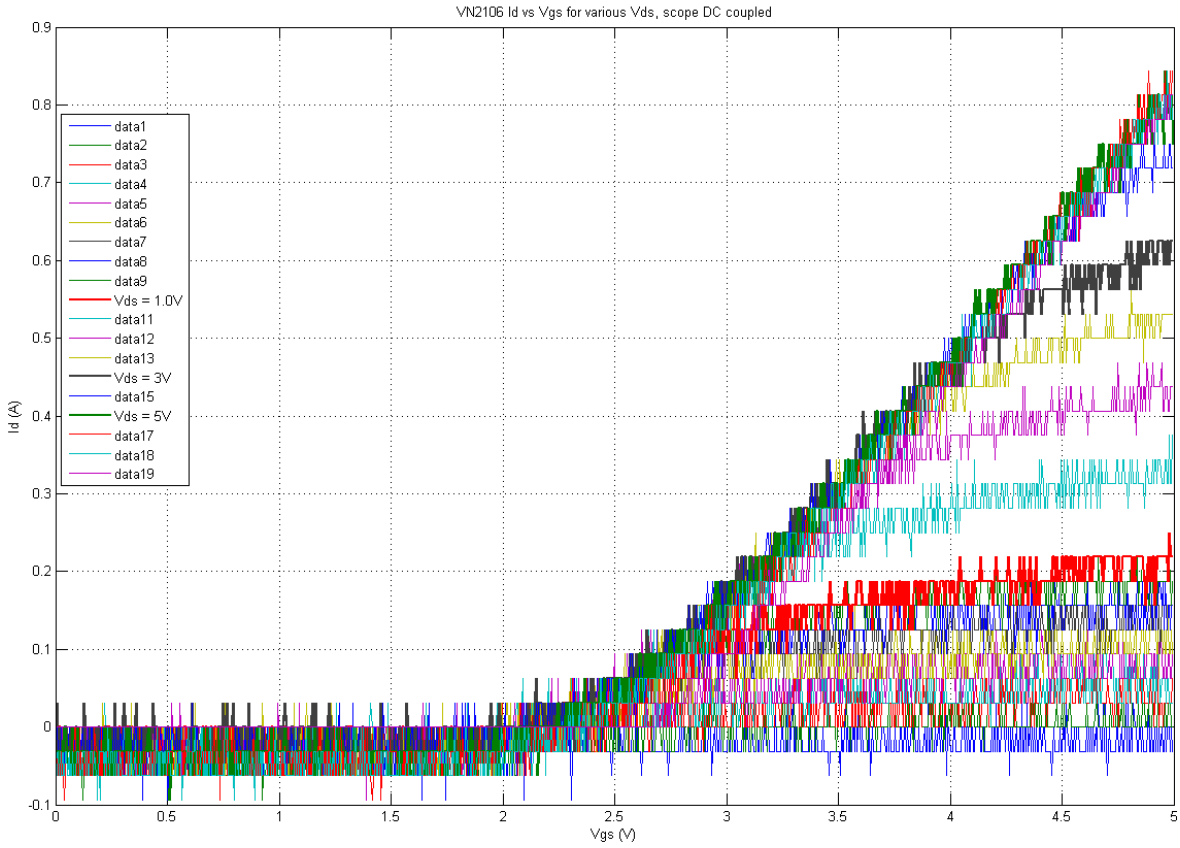
The Labview HP54600B_ReadTraces VI is used to read the Vds and Vdd (Ch1&2) traces from the scope and save the data to a *.CSV file. Each new acquisition (each new value of Vds) is appended to the same file. The resulting file can be directly read by Excel, or (as below) can be read into Matlab using the Import_HP54600B_ScopeData helper script.

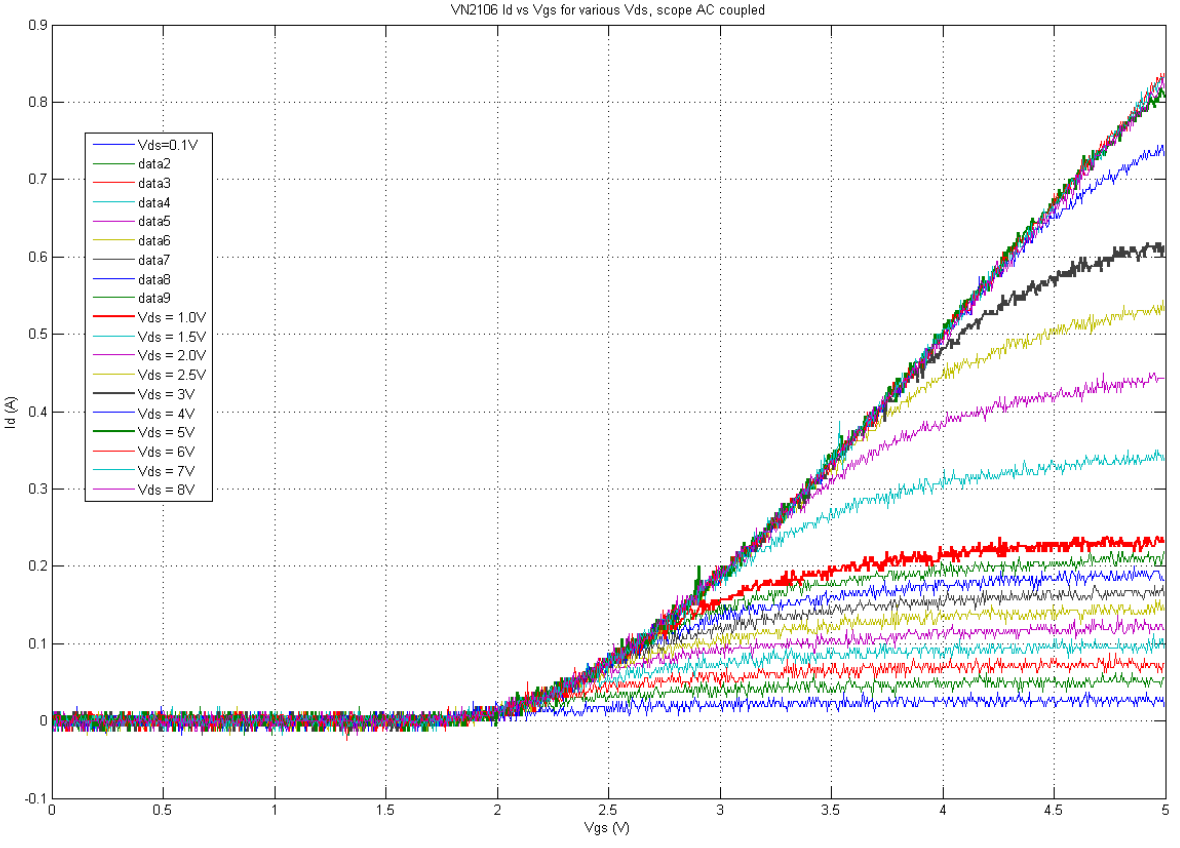
Two sets of resultant data are included. ScopeData6.csv has the scope set for DC coupled inputs. Unfortunately, in order to fit the whole range of Vds on the screen, the voltage resolution (already poor for the HP54600B) is pretty awful. ScopeData7.csv has the scope set for AC coupled inputs in order to increase the gain on the vertical channels, under the assumption that the ramp pulse is fast enough to not experience low-pass distortion from the AC coupling. This appeared to be a valid assumption.

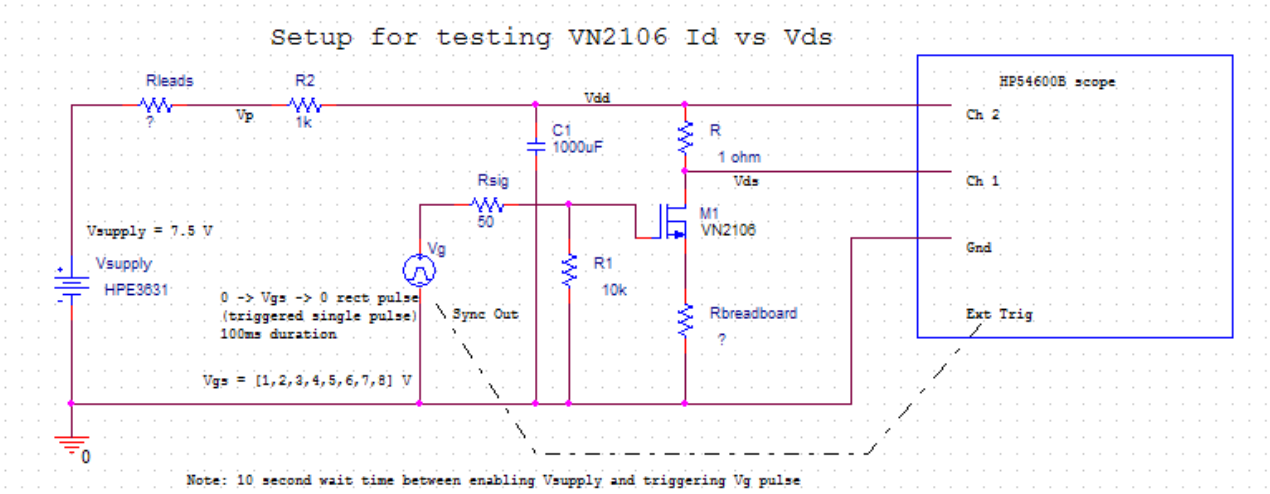
The example Matlab script below uses the function “Import_HP54600B_ScopeData” to select and read a *.CSV file created by a Labview program using “HP54600B_ReadTraces.vi”.

This shows how simple it can be. The plots on the next page were created with this script and then annotated “manually” in Matlab.

```
[Data,Setup]=Import_HP54600B_ScopeData;

Ids=Data(:,:,3)-Data(:,:,2);
Time=Data(:,:,1);
Vgs=Time*500;
plot(Vgs',Ids');
```


VN2106 Id vs Vgs for various Vds, scope DC coupled
Id (A)
Vgs (V)
data1
data2
data3
data4
data5
data6
data7
data8
data9
Vds = 1.0V
data11
data12
data13
Vds = 3V
data15
Vds = 5V
data17
data18
data19


VN2106 Id vs Vgs for various Vds, scope AC coupled
Id (A)
Vgs (V)
Vds=0.1V
data2
data3
data4
data5
data6
data7
data8
data9
Vds = 1.0V
Vds = 1.5V
Vds = 2.0V
Vds = 2.5V
Vds = 3V
Vds = 4V
Vds = 5V
Vds = 6V
Vds = 7V
Vds = 8V



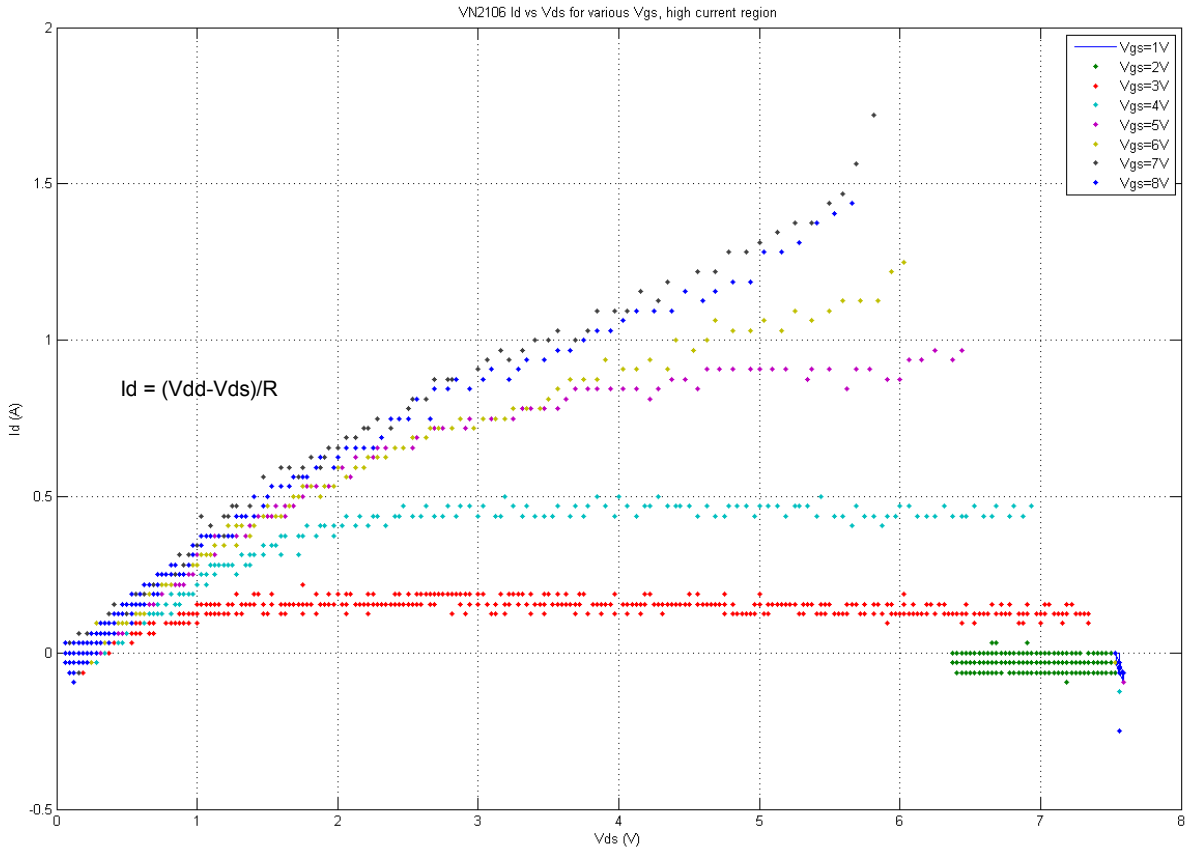
Since the E3631 power supply cannot put out more than 1 amp on its +/- 25V outputs, and the VN2106 MOSFET would burn up using the old power supply stepping method of measuring Id vs Vds, C1 is used to store a small fixed amount of energy right on the breadboard. The Labview pulse generation VI is used to send a single pulse of known amplitude to the MOSFET gate. This pulse is long enough to discharge C1 at a rate determined by the MOSFET conductance. The scope captures Vds and the voltage across R as C1 discharges, and the Labview program saves the scope data to a *.CSV file. The Matlab code (below) then reads in the data and plots Id vs Vds.

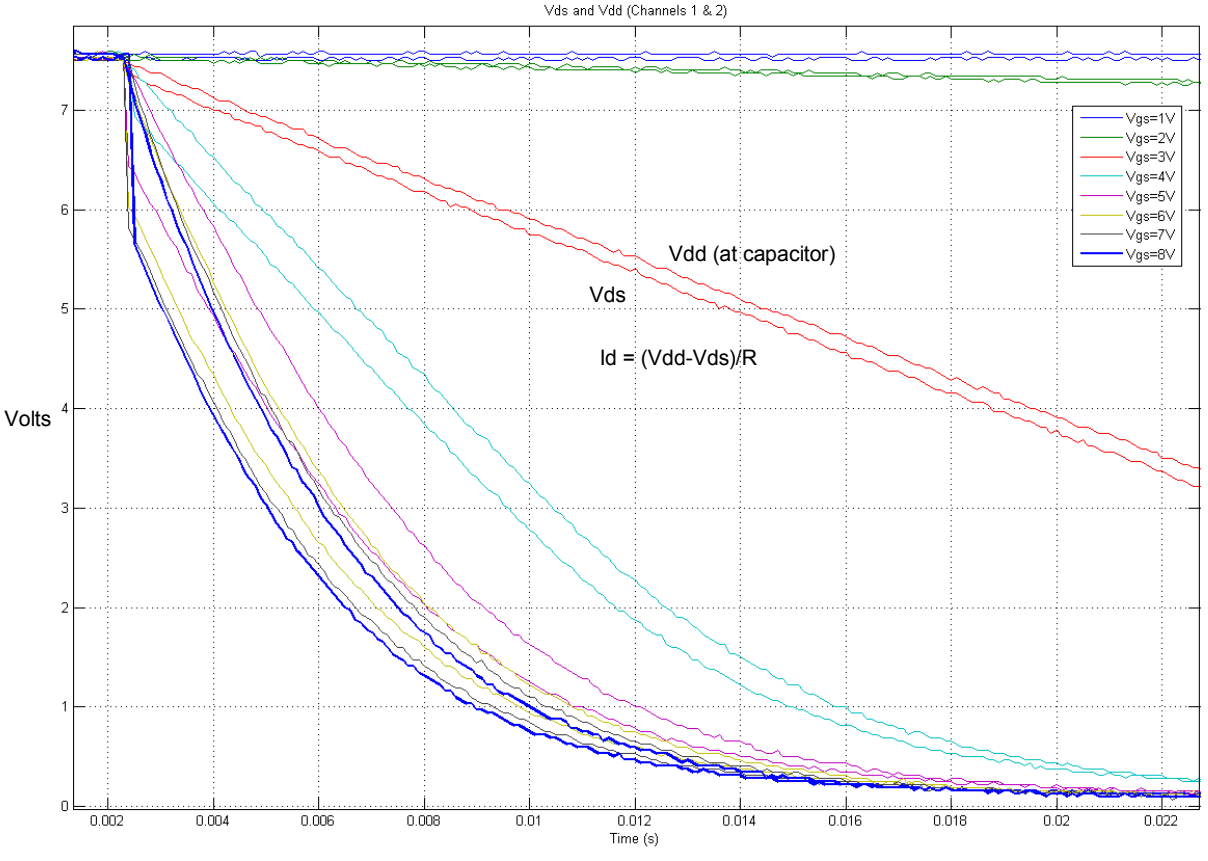
One set of resultant data is included. ScopeDataA9.csv has the scope set for DC coupled inputs. Unfortunately, in order to fit the whole range of Vds on the screen, the voltage resolution (already poor for the HP54600B) is pretty awful. AC coupling was not tried (yet).

The Matlab code snippet below uses the function “Import_HP54600B_ScopeData” to select and read a *.CSV file created by a Labview program using “HP54600B_ReadTraces.vi”.

This code (“CurveTracerTest_Id_vs_Vds2.m”) is slightly more complicated than the previous.
The first plot on the next page was created with an older version of this script and then annotated “manually” in Matlab. The second plot was created with this script as Matlab figure 5. Some text was added with OpenOffice's presentation software during the creation of this report.

```
[Data,Setup]=Import_HP54600B_ScopeData;

Time=Data(:,:,1);
Vds=Data(:,:,2);
Vdd=Data(:,:,3);
Ids=Vdd-Vds;     % note Rd = 1 ohm

figure(1);
plot(Vds',Ids');
title(Setup{1,1}{1}{1});
xlabel('Vds (V)');
ylabel('Id (A)');
grid on;
legend({'Vgs=1V','Vgs=2V','Vgs=3V','Vgs=4V','Vgs=5V','Vgs=6V','Vgs=7V','Vgs=8V'});

.
.
.

figure(5);
hold off;
plot(Time',Vds');
hold on;
plot(Time',Vdd');
title('Vds and Vdd (Channels 1 & 2)');
xlabel('Time (s)');
ylabel('Amps');
grid on;
legend({'Vgs=1V','Vgs=2V','Vgs=3V','Vgs=4V','Vgs=5V','Vgs=6V','Vgs=7V','Vgs=8V'});
```


VN2106 Id vs Vds for various Vgs, high current region
Vgs=1V
Vgs=2V
Vgs=3V
Vgs=4V
Vgs=5V
Vgs=6V
Vgs=7V
Vgs=8V
Id = (Vdd-Vds)/R
Id (A)
Vds (V)
2
1.5
1
0.5
0
-0.5
0
1
2
3
4
5
6
7
8


Vds and Vdd (Channels 1 & 2)
Vgs=1V
Vgs=2V
Vgs=3V
Vgs=4V
Vgs=5V
Vgs=6V
Vgs=7V
Vgs=8V
Vdd (at capacitor)
Vds
Id = (Vdd-Vds)/R
Volts
Time (s)
7
6
5
4
3
2
1
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
0.022

Example *.CSV file from HP54600B_ReadTraces.vi

This is the setup portion (single quote delimited) and top part of the numerical data section.
The very first line is the user's comment, with the acquisition date and time automatically prepended.

```
'12/26/2007  4:44 PM  Id vs Vds curve tracer test 9, scope DC coupled, C=1000uF'
'Scope Parameter Name', 'Value'
'*IDN?', 'HEWLETT-PACKARD,54600B,0,A.01.27'
':TIMEBASE:RANGE?', '+1.00000000E-001'
':TIMEBASE:MODE?', 'NORM'
':TIMEBASE:REFERENCE?', 'CENT'
':TIMEBASE:VERNIER?', 'OFF'
':TIMEBASE:DELAY?', '+5.00000000E-002'
':ACQUIRE:TYPE?', 'NORM'
':ACQUIRE:COUNT?', '+8'
':TRIGGER:SOURCE?', 'EXT'
':TRIGGER:LEVEL?', '+1.99000000E+000'
':TRIGGER:HOLDOFF?', '+2.00000000E-007'
':TRIGGER:COUPLING?', 'DC'
':TRIGGER:SLOPE?', 'NEG'
':TRIGGER:REJECT?', 'OFF'
':TRIGGER:NREJECT?', 'ON'
':TRIGGER:MODE?', 'NORM'
':CHAN1:RANGE?', '+8.00000000E+000'
':CHAN1:COUPLING?', 'DC'
':CHAN1:BWLIMIT?', 'OFF'
':CHAN1:INVERT?', 'OFF'
':CHAN1:VERNIER?', 'OFF'
':CHAN1:PROBE?', 'X10'
':CHAN1:OFFSET?', '+4.00000000E+000'
':CHAN2:RANGE?', '+8.00000000E+000'
':CHAN2:COUPLING?', 'DC'
':CHAN2:BWLIMIT?', 'OFF'
':CHAN2:INVERT?', 'OFF'
':CHAN2:VERNIER?', 'OFF'
':CHAN2:PROBE?', 'X10'
':CHAN2:OFFSET?', '+4.00000000E+000'
':WAVEFORM:POINTS?', '+1000'
':CHAN:MATH?', 'OFF'
'Time (s)', 'Channel 1 (V)', 'Channel 2 (V)'
0.000000, 7.562500, 7.500000
0.000100, 7.562500, 7.531250
0.000200, 7.562500, 7.531250
0.000300, 7.562500, 7.531250
0.000400, 7.562500, 7.531250
0.000500, 7.562500, 7.531250
.
.
.
.
```