

# Transferring data from the TDS3014 oscilloscope to a computer

This document describes a Matlab script that captures data from a Tektronix TDS3014 oscilloscope and stores it in a file.

**Downloading files:** The files necessary to run this script are [MatlabTDS3014.m](#) and [MatlabTDS3014.fig](#). These need to be downloaded and stored somewhere that Matlab can find them. If you are in Hicks Hall, these should already be loaded on your computer and accessible.

**Setting up the oscilloscope:** Make sure that the scope is attached to the computer through the serial cable and that it is turned on, or MatLab stops working until the scope is attached properly. Also, make sure that the baud rate is set to 38400, which is done by hitting “Utility,” then “System configure” until the I/O menu appears. Then, set the baud rate (on the right top side). If you are in Hicks, the scopes should already be properly configured.

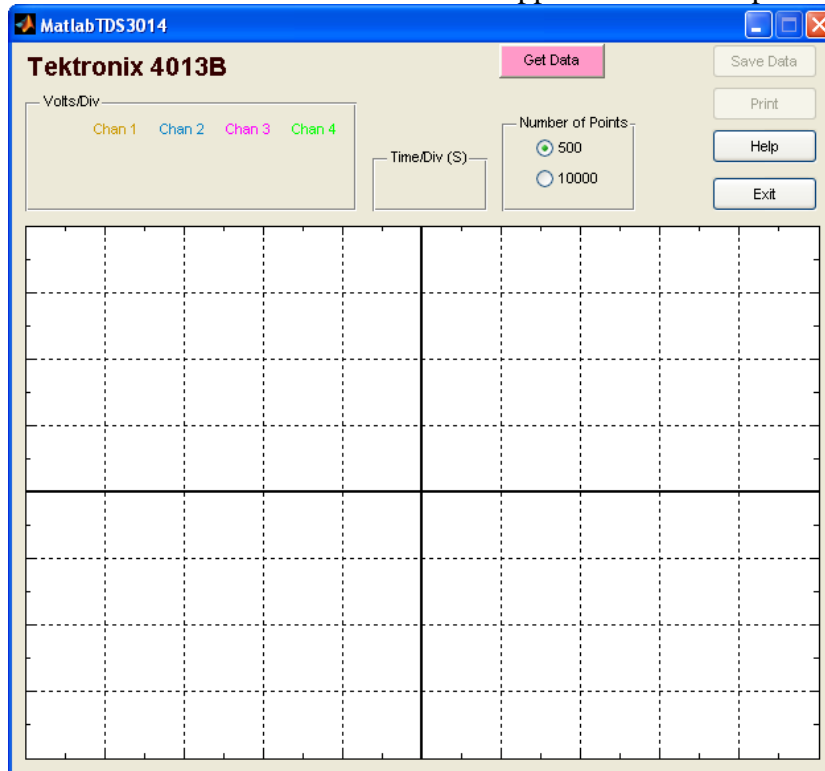
A pdf version is available at <http://www.swarthmore.edu/NatSci/echeeve1/Ref/TDS4013/MatlabTDS3014.pdf>

---

To collect data from the oscilloscope, start Matlab and run the script *MatlabTDS3014*. This program sends data from the scope to the computer. To run it type:

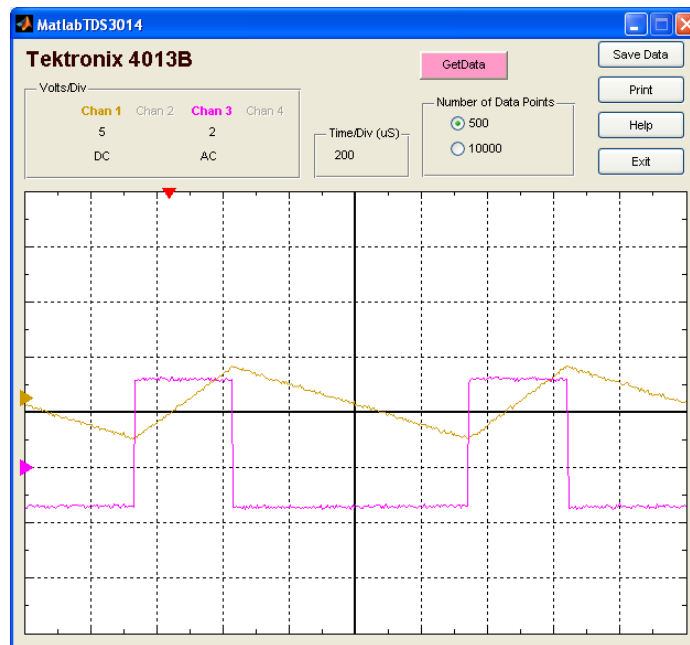
```
>> MatlabTDS3014
```

from the Matlab prompt. A window similar to the one below appears on the computer:



The program is fairly straightforward to use. It will load the data for the horizontal scale, the vertical scale, the coupling status, the time shift, the voltage shifts, and the oscilloscope screen for all four channels. You can choose how many data points to take from the machine — 500 or 10,000. Taking 500 points takes about 0.5s per channel, whereas taking 10,000 points takes between 8 and 10 seconds per channel. You should choose the one that will give you the precision you need to do the lab without unduly wasting your time. Also, the “Help” button links this document online

When you are ready to collect data hit the "Get Data" button on the Matlab GUI. A sample result is shown below with channels 1 and 3 active. The colors are chosen to match those of the oscilloscope.



Note the small triangles on the left of the screen that show the location of ground (similar to the ground markers on the oscilloscope screen). Also, the red triangle on the top references the time shift, as is shown in red on the oscilloscope.

At this point you can save the data by hitting the "Save data" button. You will also be prompted for a short comment which can help you identify the contents of the file later (e.g., "Lab 1, Circuit 3"). You can also print the figure - there is a black and white printer in the lab (Hicks 314) and Baggins is the printer in the computer lab downstairs.

## Loading the data into Matlab

To work with the data from within Matlab, load a previous saved file (in this case called "FILENAME").

```
>> load('FILENAME')
```

This will load three variables, accessible with the "whos" command.

```
>> whos
  Name      Size      Bytes  Class      Attributes
  data      1x4        12240  cell
  fileComment 1x40       80     char
  t          500x1      4000   double
```

The variable "data" is actually a cell array and holds the oscilloscope data, with one section from each channel (data{1} is the data from the first channel, etc - note that the array indices use a curly brace, and not a parentheses), "t" is the corresponding time vector, and "fileComment" holds the comment you entered as well as the date and time at which the data was collected (which might be useful)

To plot the data from channel 1, use the following Matlab command (again, note that the characters surrounding the 1 are curly braces {}, not parentheses). This denotes a "cell array" in Matlab - you won't really have to worry about them very much.

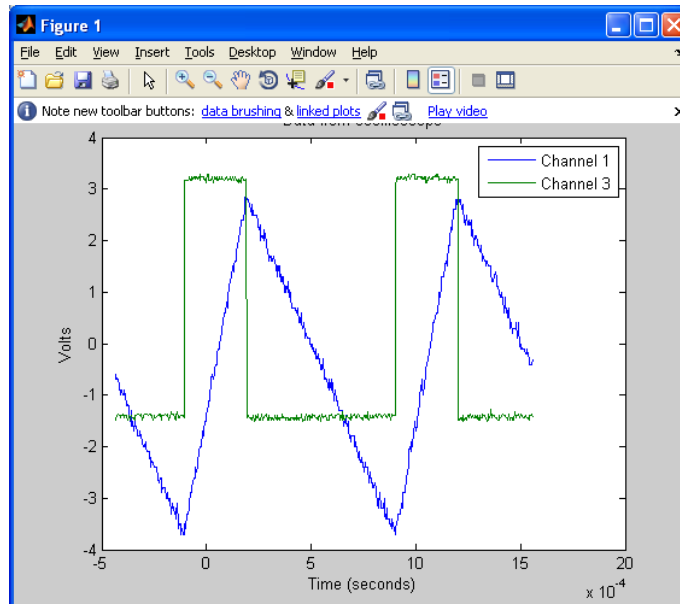
```
>> plot(t, data{1})
```

To plot channel 1 and 2 with axes labeled and a legend enter:

### Matlab code section 1:

```
>> plot(t, data{1}, t, data{3})
>> legend('Channel 1','Channel 3')
>> title('Data from oscilloscope')
>> xlabel('Time (seconds)')
>> ylabel('Volts');
```

When you do this, the data will plot voltage vs time, not scaled and shifted as it is on the oscilloscope screen. For example ground (voltage=0) will be the same for all channels, while it may differ on the oscilloscope screen, also the voltage scales will also be the same). Since you are outside of the MatlabTDS3014 program at this point, Matlab reverts to the graphs no longer match the oscilloscope coloring scheme.



Note that this looks a little different than the original data because 0 volts is in the middle of the screen, whereas on the oscilloscope, the grounds may be at different locations.

Note: you could also save the data to another variable before plotting.

```
>> OutputVoltage=data{1};
>> plot(t, OutputVoltage)
>> title('Data from oscilloscope')
>> xlabel('Time (seconds)')
>> ylabel('Output Voltage (Volts)s');
```

If you want to work with a subset of the data, it may be easier to copy it to another array ("c1" and "c2" in the example below).

### Matlab code section 2:

```
>> myIndex=find(t>=0); %Find where t is equal to or greater than zero.
>> c1=data{1};
>> c3=data{3};
>> plot(t(myIndex), c1(myIndex), t(myIndex), c3(myIndex))
```

