```
% NOTE: we assume that the scope has been correctly configured to capture
% 10Mb data; furthermore, this configuration has been saved to 'Setup 1'
% on the scope.  If 'Setup 1' does not contain the appropriate settings
% file, it may retrieved from the compact flash card accompanying the
% scope--it has been saved as 'dilon10mb.set'.  Having loaded the settings
% file, the setup should then be resaved to 'Setup 1'.

%
% SESSION VAR
%
% number of records to capture
n = 5;
% length of record
l = 500e6;

%
% CONNECTION VAR
%
% set hw address for connection to scope (if hw address has changed--i.e.,
% the scope is plugged into a different USB port--then refer to
% 'matlab_visa_connection.txt' to determine port settings

% vu = visa('ni', 'USB0::0x0699::0x0401::C010098::INSTR');

if ~exist('SCP')
    SCP = visa('tek', 'GPIB8::1::INSTR'); %found via 'instrhwinfo('visa','tek')'

    % set transfer buffer to record length:
    %   difference of query(vu,'DATA:START?') and query(vu,'DATA:STOP?')
    SCP.InputBufferSize=l;
end

if isempty(SCP)
    disp('Scope not present!');
    return;
end

% save directory (remember trailing slash)
recDir = 'C:\Users\rgerdes\Desktop\tmp\';

% open connection to scope
fopen(SCP);

% configure scope for 10Mb capture (recall 'Setup 1'); may need to recall
% manually
% fprintf(capture_scope,'*RCL 0');

% determine y-increment (voltage scale)
yinc = str2num(query(SCP,'WFMI:YMU?'));

% sample rate
s = str2num(query(SCP,'HOR:MODE:SAMPLER?'));

% set data collection points
fprintf(SCP,'DATA:START 1'); %beginning of record, in buffer of 1*10^6
fprintf(SCP,'DATA:STOP 500e3'); %end of record, in buffer

i = 0; %current rec cnt
badRecs = 0;

% begin timer
tic;
```

```
% instruct scope to take single measurement
fprintf(SCP,'ACQ:STATE ON');

disp('Run initiated...');
while (i < n )
    % wait for scope to finish measurement (check scope state every 1/10
    % of a second)
    disp('Waiting for trigger...')
    while str2num(query(SCP,'ACQ:STATE?')) == 1
            pause(0.1);
    end
    disp('...triggered!')

    % get channel data
    fprintf(SCP,'DAT:SOURCE CH1');
    fprintf(SCP,'CURV?');
    ch1 = binblockread(SCP,'int8')';

    fprintf(SCP,'DAT:SOURCE CH2');
    fprintf(SCP,'CURV?');
    ch2 = binblockread(SCP,'int8')';

    % to save time, trigger new measurement on scope
    fprintf(SCP,'ACQ:STATE ON');

    % recover differential signal
    rec = yinc*(ch2-ch1);

    ts = now;

    is = num2str(i);
    f = [recDir 'sample' ...
        strrep(num2str(zeros(1,5-length(is))), ' ', '') is '.mat'];
    save(f,'ts','s','yinc','rec');

    t = round(toc); % get current running time
    disp(['Record: ' num2str(i+1) '/' num2str(n) ...
        ' (%' num2str(i/n*100) '); ' ...
        'Elapsed time: ' num2str(floor(t/3600)) ' hr. ' ...
        num2str(mod(floor(t/60),60)) ' min. ' ...
        num2str(mod(t,60)) ' sec.']);

    % increment good recs counter
    i = i + 1;
end

disp('...run complete');

% close connection to scope
fclose(SCP);
```