Tausand AB1000 Matlab library example: Multiple Read Example

Table of Contents

Reads continously and saves data from a Tausand Abacus coincidence counter. Uses functions in the Tausand_AB1000_MatlabLibrary. Handles errors and retry readings and connections when lost. To be used in Matlab's command window.

Author: David Guzmán. Tausand Electronics, Colombia.

Created: 2021-03. Last revision: 2021-03-16. Version: 1.1.

Contact email: dguzman@tausand.com. Website: http://www.tausand.com

Constants to be defined by user

Change this parameter to set how many samples to read:

```
samples_to_read = 10;
```

Change this parameter to set your sampling time. 1000=1s:

```
my_sampling_time_ms = 1000;
```

Change this port to the adequate one:

```
my_port = 'COM23';
```

Define the desired channels to be read. Example:

```
channels_to_read = ["A","B","C","AB","AC","multiple_1"];
```

where string "multiple_1" corresponds to a multi-fold measurement, to be configured, e.g. 'ABC'

Establish a connection

```
disp("******************************");
disp("MATLAB multiple read example");
disp("******************************");
disp("1. Establish a connection");


my_abacus = openAbacus(my_port)
```

```
device_idn = idnQuery(my_abacus);
disp(['Device IDN: ',device_idn]);
```

```
*****************************
MATLAB multiple read example
*****************************
1. Establish a connection

   Serial Port Object : Serial-COM23 AB1504

   Communication Settings
      Port:               COM23
      BaudRate:           115200
      Terminator:         'LF'

   Communication State
      Status:             open
      RecordStatus:       off

   Read/Write State
      TransferStatus:     idle
      BytesAvailable:     0
      ValuesReceived:     21
      ValuesSent:         7


Device IDN: Tausand Abacus AB1504
```

Write and read new settings

```
disp("*******************************");
disp("2. Write and read new settings");
```

```
*****************************
2. Write and read new settings
```

Write settings, using configureByName function:

```
disp(['Setting sampling time to
 ',num2str(my_sampling_time_ms),'ms.']);
configureByName(my_abacus,"sampling",my_sampling_time_ms);
```

```
Setting sampling time to 1000ms.
```

Several configurations may be applied with a single command line:

```
disp('Configuring coincidence window to 50ns.');
disp('Configuring delay in channels A, B to 0 and 10ns.');
disp('Configuring sleep in channels A and B to 20ns.');
configureByName(my_abacus,...
    ["coincidence_window","delay_A","delay_B","sleep_A","sleep_B"],...
    [50,0,10,20,20]);
    %this sets: coincidence_window=50ns, delay_A=0ns, delay_B=10ns,
    %sleep_A=0ns, sleep_B=20ns.
```

```
Configuring coincidence window to 50ns.
Configuring delay in channels A, B to 0 and 10ns.
Configuring sleep in channels A and B to 20ns.
```

Configure multiple coincidence counter

```
disp("Configuring multi-fold coincidences to 3-fold ABC.");
configureMultipleCoincidence(my_abacus,"ABC");
```

```
Configuring multi-fold coincidences to 3-fold ABC.
```

Upgrade 'TAUSAND:timeout' warning to an error, to catch them.

```
my_warn = warning('error', 'TAUSAND:timeout');
```

Read current settings

```
max_try=5;
for attempt=1:max_try
    try
        [setting_values,setting_labels]=queryAllSettings(my_abacus);
        current_settings=[setting_labels,setting_values];
        disp('Current settings are:');
        disp([setting_labels,setting_values]);
        break;
    catch ME
        switch ME.identifier
            case {'TAUSAND:unexpectedReadByte',...
                    'TAUSAND:checksumFailed','TAUSAND:timeout'}
                %ignore these errors, just retry.
            case 'MATLAB:serial:fwrite:opfailed'
                %if connection is lost, maybe device has been
                %disconnected
                closeAbacus(my_abacus)
                try
                    openAbacus(my_abacus)
                catch
                    %ignore error
                end
            otherwise
                warning('Unexpected error. Device connection closed.')
                closeAbacus(my_abacus)
                rethrow(ME)
        end
    end
end
warning(my_warn.state, 'TAUSAND:timeout'); % Restore this warning back
 to their previous (non-error) state
if (attempt == max_try)
    error('TAUSAND:timeout',['Communication error after
 ',int2str(max_try),' attempts']);
end
```

```
Current settings are:
    "sampling"              "1000"
```

```
    "coincidence_window"    "50"
    "delay_A"               "0"
    "delay_B"               "10"
    "delay_C"               "0"
    "delay_D"               "0"
    "sleep_A"               "20"
    "sleep_B"               "20"
    "sleep_C"               "0"
    "sleep_D"               "0"
    "config_multiple_1"     "224"
```

Create plaintext file

```
disp("******************************");
disp("3. Create file");
date_time_string = string(datetime,'yyyy-MM-dd_HHmmss');
file_name =
 strcat('data_multipleReadExample_',date_time_string,'.txt');
```

```
*****************************
3. Create file
```

Create file,

```
myfile = fopen(file_name, 'a'); %create file in append mode
disp(strcat("File ",file_name," has been created"));
```

```
File data_multipleReadExample_2021-03-16_172336.txt has been created
```

and write file headers,

```
fwrite(myfile,['multipleReadExample',newline]);
fwrite(myfile,['--------------------',newline]);
fwrite(myfile,['Begin time: ',char(date_time_string),newline]);
fwrite(myfile,['Device IDN: ',device_idn,newline]);
fwrite(myfile,['Serial port: ',my_abacus.port,newline]);
fwrite(myfile,['Settings: ',newline]);
fprintf(myfile,'%s\t%s\n',[setting_labels,num2str(setting_values)]');
fwrite(myfile,[newline,newline]);
```

Multiple read using function waitAndGetValues

```
disp("******************************");
disp('4. Multiple read using waitAndGetValues function begins');
tMultipleReadExample = tic;
for sample=1:samples_to_read
    try
        if sample == 1
            %get column header and data
            [my_data,my_headers] =
 waitAndGetValues(my_abacus,channels_to_read);
            column_headers = ["PC_time",my_headers'];
            fprintf('\t%s',column_headers); %print in command window
```

```
            fprintf('\n');
            fprintf(myfile,'%s\t',column_headers); %print in file
            fwrite(myfile,newline);
        else
            my_data = waitAndGetValues(my_abacus,channels_to_read);
        end
        s=strcat(sprintf('%.3f',toc(tMultipleReadExample)),sprintf('\t
%d',my_data),string(newline));
        fwrite(myfile,s);
        t = [num2str(sample),'/',num2str(samples_to_read)];
        fprintf('%s\t%s',t,s);    %print data in command window
    catch ME
        switch ME.identifier
            case {'TAUSAND:unexpectedReadByte',...
                    'TAUSAND:checksumFailed','TAUSAND:timeout'}
                %ignore these errors, just continue.
            case 'MATLAB:serial:fwrite:opfailed'
                %if connection is lost, maybe device has been
                %disconnected
                closeAbacus(my_abacus)
                try
                    openAbacus(my_abacus)
                catch
                    %ignore error
                end
            otherwise
                warning('Unexpected error. Device connection closed.
 File access closed.')
                fclose(myfile);
                closeAbacus(my_abacus)
                rethrow(ME)
        end
    end
end

*****************************
4. Multiple read using waitAndGetValues function begins
 PC_time counters_ID counter_A counter_B counter_C counter_AB
 counter_AC counter_multiple_1
1/10 0.857 62 3472 31251 31251 3472 3472 3472
2/10 2.034 63 3472 31251 31251 3472 3472 3472
3/10 2.935 64 3473 31250 31250 3473 3473 3473
4/10 3.908 65 3472 31251 31251 3472 3472 3472
5/10 4.909 66 3472 31251 31251 3472 3472 3472
6/10 5.903 67 3472 31250 31250 3472 3472 3472
7/10 6.966 68 3473 31251 31251 3473 3473 3473
8/10 7.906 69 3472 31251 31251 3472 3472 3472
9/10 8.946 70 3472 31250 31250 3472 3472 3472
10/10 9.968 71 3473 31251 31251 3473 3473 3473
```

Close connection and close file

```
fclose(myfile);
```

```
disp(strcat("File ",file_name," has been closed."));
closeAbacus(my_abacus)
disp(strcat("Connection to device in port ",my_abacus.name," has been
 closed."));
```

```
File data_multipleReadExample_2021-03-16_172336.txt has been closed.
Connection to device in port Serial-COM23 AB1504 has been closed.
```

Published with MATLAB® R2017a