

---

# Tausand AB1000 Matlab library tester

A set of commands to test Tausand\_AB1000\_MatlabLibrary to be used in Matlab's command window

Author: David Guzmán. Tausand Electronics, Colombia.

Created: 2020-09. Last revision: 2021-03-15. Version: 1.1.

Contact email: [dguzman@tausand.com](mailto:dguzman@tausand.com). Website: <http://www.tausand.com>

```
findDevices(1504)    %searches for Tausand Abacus AB1504 devices
findDevices()       %searches for any device of the Tausand Abacus
                    family
my_ports=findDevices()    %searches for any device of the Tausand
                    Abacus family
%my_abacus=openAbacus('COM28'); %change COM port depending on your
                    results of 'findDevices'
my_abacus=openAbacus(my_ports{1}); %connects to the first found port

configureByName(my_abacus,['sampling'],[2500]) %sets sampling to 2.5s
configureChannel(my_abacus,'B',20,50) %sets delay to 20ns and sleep to
                    50ns, or the closest valid values
configureChannel(my_abacus,'a',10,59) %sets delay to 10ns and sleep to
                    59ns, or the closest valid values
configureCoincidenceWindow(my_abacus,133) %sets to 133ns, or the
                    closest valid value
configureDelay(my_abacus,'c',35) %sets to 35ms, or the closest valid
                    value
configureSleep(my_abacus,'a',35) %sets to 35ms, or the closest valid
                    value
configureMultipleCoincidence(my_abacus,"ABD")
configureSamplingTime(my_abacus,555) %sets to 555ms, or the closest
                    valid value
countersIdQuery(my_abacus)
idnQuery(my_abacus)
queryAllSettings(my_abacus)
[my_sett_data,my_sett_labels]=queryAllSettings(my_abacus)
queryCoincidenceWindow(my_abacus)
%queryDelay(my_abacus) %this line leads to an error due to a lack of
                    channel selection: a,b,c or d.
queryDelay(my_abacus,'A') %this is correctly used
queryMultipleCoincidence(my_abacus)
querySamplingTime(my_abacus)
queryAllSettings(my_abacus)
readMeasurement(my_abacus)
[my_meas_data,my_meas_labels]=readMeasurement(my_abacus)
timeLeftQuery(my_abacus)
waitForAcquisitionComplete(my_abacus)
configureSamplingTime(my_abacus,5000)
waitForAcquisitionComplete(my_abacus,true)
closeAbacus(my_abacus)
```

Progress: 1/2

---

*Progress: 2/2*

*ans =*

*"COM23"*

*Progress: 1/2*

*Progress: 2/2*

*ans =*

*"COM23"*

*Progress: 1/2*

*Progress: 2/2*

*my\_ports =*

*"COM23"*

*ans =*

*uint32*

*20*

*ans =*

*uint32*

*10*

*ans =*

*uint32*

*130*

*ans =*

*uint32*

*36*

*ans =*

*uint32*

*36*

---

`ans =`

`'ABD'`

`ans =`

`uint32`

`560`

`ans =`

`uint32`

`47`

`ans =`

`'Tausand Abacus AB1504'`

`ans =`

`11×1 uint32 column vector`

`560`

`130`

`10`

`20`

`36`

`0`

`36`

`50`

`0`

`0`

`208`

`my_sett_data =`

`11×1 uint32 column vector`

`560`

`130`

`10`

`20`

`36`

`0`

`36`

---

```

50
0
0
208

my_sett_labels =

11x1 string array

"sampling"
"coincidence_window"
"delay_A"
"delay_B"
"delay_C"
"delay_D"
"sleep_A"
"sleep_B"
"sleep_C"
"sleep_D"
"config_multiple_1"

ans =

uint32

130

ans =

uint32

10

ans =

'ABD'

ans =

uint32

560

ans =

11x1 uint32 column vector

560

```

---

---

```
130
10
20
36
0
36
50
0
0
208
```

```
ans =
```

```
13×1 uint32 column vector
```

```
1
0
0
0
0
0
0
0
0
0
0
0
0
549
```

```
my_meas_data =
```

```
13×1 uint32 column vector
```

```
1
0
0
0
0
0
0
0
0
0
0
0
0
307
```

```
my_meas_labels =
```

```
13×1 string array
```

---

```
"counters_ID"  
"counter_A"  
"counter_B"  
"counter_C"  
"counter_D"  
"counter_AB"  
"counter_AC"  
"counter_AD"  
"counter_BC"  
"counter_BD"  
"counter_CD"  
"counter_multiple_1"  
"time_left"
```

```
ans =
```

```
uint32
```

```
265
```

```
ans =
```

```
uint32
```

```
5000
```

```
Current ID is 2  
Next data is available in 5.0s  
Now, current ID is 1
```

*Published with MATLAB® R2017a*