

Sequencer

DC Polarizations, Ramps and Loops

1 Introduction.....	2
2 Starting Sequencer Program.....	2
2.1 Choosing Different Potentiostats	3
3 Writing a Sequence.....	4
3.1 Hold at OCP	4
3.2 Potential Ramp.....	5
3.2.1 Potential Ramp Starting from Actual Potential	5
3.3 Current Ramp.....	7
3.4 Hold at a Constant Potential/Current	7
3.5 Set Sampling Rate	9
3.6 Defining Loops	9
4 Sample Sequences	10
5 Global Control Settings and Limits	12
6 Additional Acquisition Channels	14
7 Incorporating a Sequence in SCRIPT	15

1 Introduction

In research labs and industries, simple electrochemical processes are repeated multiple times to test different systems under investigation. The repetition of the following DC processes can be easily automated using the “Sequencer” software.

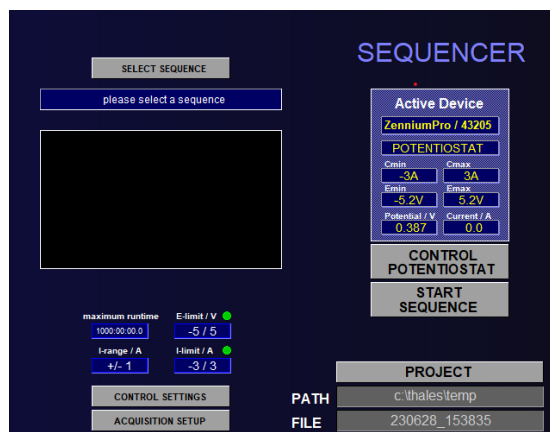
- Potential ramp
- Current ramp
- Constant potential/current vs time
- Hold at open circuit voltage (OCP)

2 Starting Sequencer Program

Click on the *EXE* icon in the classic mode of the *Thales* software and open the *sequencer.rtm* file. This will open the sequencer window. The *.rtm* file is available in folder *c:\Thales*.



Thales software – Classic mode



Sequencer

The sequencer software requires a “sequence”, a text file, defining the measurement procedure and parameters. In the sequencer software, pre-written sequence is uploaded for the measurement. Different sample sequences are provided in the folder *c:\thales\script\sequencer\sequences*. A sequence can be uploaded in the sequencer window via **SELECT SEQUENCE**. The appropriate potential and current ranges can either be defined directly in the sequence or can be modified via **CONTROL SETTINGS** before running the measurement. The valid current and voltage limits for the sequence will be shown above the **CONTROL SETTING** button in the sequencer software.

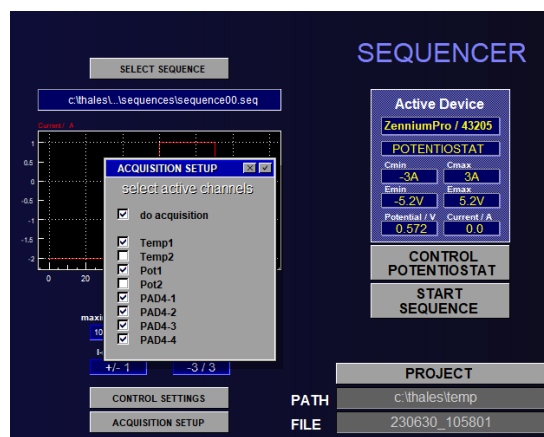
On the right side of the sequencer software, the active device, with its maximum current and voltage limits, is shown. One may select another device (e.g., a PP2X2 or an EL1002, an extension of the ZENNIUM potentiostat) via the **CONTROL POTENTIOSTAT** button. After selecting a device, loading a sequence and setting

the appropriate potential/current ranges, the measurement can be started by clicking on START SEQUENCE.

The PROJECT indicates the path where the measured data file will be saved. A default filename is set to DATE_TIME format. The "Sequencer" actively writes the measured data (in blocks) in a text file during the measurement and does not wait for the measurement to end for generating the measured data file. Since the sequencer is writing the data in real-time, a sequence may be run for a maximum runtime of up to 1,000 hours. As the measured data is continuously saved during the measurement hence the user has to define the folder path and file name before starting a measurement. Clicking on PROJECT will open a sub-menu where one can define the pathway to the folder where the measurement file will be saved. In addition, the name of the measured data file can also be defined via the sub-menu accessible through PROJECT.

The ACQUISITION SETUP shows the active channels from the Signal Acquisition Setup. Here, the user may select the ACQ channels (i.e., PAD4 channels), which can be recorded with the main channel. Up to 8 ACQ channels can be recorded in the sequencer. The "do acquisition" will activate the recording of data from the selected ACQ channels.

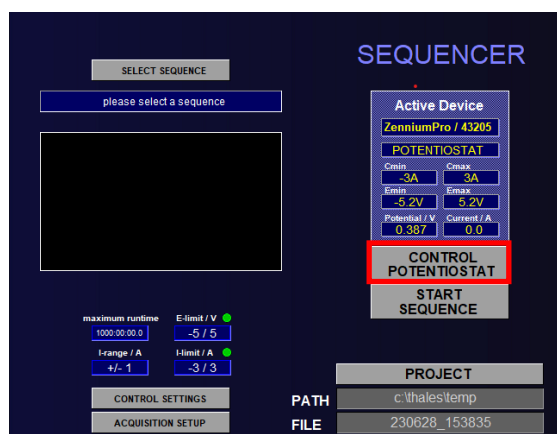
The ACQ channels will only appear in the sequencer software after they are acquired in the Signal Acquisition window. Read the [Signal ACQ manual](#) to learn, how to set up the acquisition channels.



2.1 Choosing Different Potentiostats

A power potentiostat or an electronic load can be chosen for high current/voltage applications. To choose another potentiostat, click on "CONTROL POTENTIOSTAT". This will open the TEST SAMPLING & POTENTIOSTAT CONTROL window.

In the TEST SAMPLING & POTENTIOSTAT CONTROL window, click on DEVICE. This will open a sub-window, where the EPC42 port number must be given to which the power potentiostat or the electronic load is connected.



3 Writing a Sequence

A sequence is written in the THALES EDITOR window and follows a basic format as shown below.

Sequence format

```
start_cycle      'start sequence
samples(20)      'measure 20 readings per second
do_loop(5)       'do 5 loops
.
.               'sequence body
.
end_loop         'end loop
end_cycle        'end sequence

start_online     'define global parameter
.
.               'global parameter i.e., current/voltage limits (see Chapter 5)
end_online       'end
```

We recommend writing the global parameters (start_online/end_online) for each sequence directly in the sequence file.

Commands are written in pink and comments are written in green color. More information about color coding is provided in the manual [Script - an introduction](#). The limit for the sampling frequency is from 1 mHz to 1 kHz.

The green-colored comments are not processed by the sequencer and should not be present in the sequence file. Comments are shown in this manual to explain the sequences.

The user may choose any name while saving a sequence however a standard name format for the sequences exists which must be followed if the user wants to employ the sequencer software in Zahner's Script or Remote program. The standard name convention follows the format "sequence##.seq". Here ## are numbers like 00, 01, 02..., 99. In the following, simple commands are explained which can be used to build a sequence.

There must be no blank lines between the keywords **start_cycle** and **end_cycle**!

3.1 Hold at OCP

The system under investigation can be scanned at an open circuit potential (OCP) for a defined time period with the following command.

ocp(time)

Parameters: time: [s]

Functional limits are not available in OCP scan.

3.2 Potential Ramp

A potential ramp can be applied by providing the starting and end potential values in volts (V). The slope of the ramp or time for measurement can be provided in V/s or s, respectively.

Command for potential ramp with defined slope:

ramp_pot_s(starting_potential,end_potential,slope[,Imin,Imax])

[] in the commands are not required and are only used in this manual to mark the optional parts of the commands. Using [] in commands may lead to error while running the sequence.

Parameters: starting potential: [V]
end potential: [V]
slope: [V/s]
optional: minimum current limit (Imin): [A]
maximum current limit (Imax): [A]

Command for potential ramp with defined time:

ramp_pot_t(starting_potential,end_potential,time[,Imin,Imax])

Parameters: starting potential: [V]
end potential: [V]
time: [s]
optional: minimum current limit (Imin): [A]
maximum current limit (Imax): [A]

Imin (A) and Imax (A) are optional and define the minimum and maximum current limits, respectively. Upon reaching such a functional limit, the defined ramp terminates the progress of the actual command and the sequencer continues with the next command in the sequence. The current limit is provided to avoid the unwanted high currents which may damage the system under investigation. If such a limit is not necessary then it can be omitted from the command.

3.2.1 Potential Ramp Starting from Actual Potential

A potential ramp can also be initiated from the actual potential of the potentiostat to a defined end potential. This is possible with the following command

```
ramp_pot_t(PACT,end_potential,time[,Imin,Imax])
```

Parameters: *PACT*: ACTual Potential
 end_potential: [V]
 time: [s]
optional: *minimum current limit (Imin)*: [A]
 maximum current limit (Imax): [A]

Here, *PACT* represents the actual potential of the sample/system under investigation (i.e., OCP when no potential is applied). The command can also be modified for the slope as well.

```
ramp_pot_s(PACT,end_potential,slope[,Imin,Imax])
```

The user may also define the potential or the current parameter like “*PACT*+1” or “*CACT*+1” (*CACT*: ACTual Current), respectively. For *PACT*+1, the set potential value will be the actual potential value+1 (V). An example is shown below.

```
ocp(time)  
ramp_pot_t(PACT,PACT+1,10)
```

In the two lines of code, the open circuit potential will be measured in the first command. In the second command, for the potential ramp, the actual potential (here OCP) will be taken as starting potential and OCP+1 V will be taken as the end potential.

```
ocp(time)  
ramp_pot_t(PACT,PACT+1,10)  
ramp_pot_t(PACT,PACT+1,10)
```

The variable *PACT* is assigned a new actual potential value at the end of each command. In the second example with three lines of code, the *PACT* value will be the OCP after first command. After second command, the *PACT* value will be overwritten and will be OCP+1 and in third command the starting voltage will be OCP+1 and the end voltage will be OCP+2 (*PACT*+1).

Sequencer only allows + and – operators.

The user also has the option to define any parameter using the decimal separator (i.e., 0.1) or using the SI prefix (i.e., 100m for 0.1).

Examples: A potential ramp, from 0 V to 100 mV with a slope of 0.1 V/s (or time-period of 1 s).

```
ramp_pot_s(0,0.1,0.1)
```

```
ramp_pot_t(0,100m,1,-10m,100m)
```

Here in the time-defined ramp, an optional current range of -10 mA to 100 mA is also provided.

3.3 Current Ramp

The current ramp can be applied by providing the starting and end current values in ampere (A). The slope of the ramp or time for measurement can be provided in A/s or s, respectively.

Command for current ramp with defined slope:

```
ramp_cur_s(starting_current,end_current,slope[,Emin,Emax])
```

Parameters: starting current: [A]
 end current: [A]
 slope: [A/s]
optional: minimum potential limit (Emin): [V]
 maximum potential limit (Emax): [V]

Command for current ramp with defined time:

```
ramp_cur_t(starting_current,end_current,time[,Emin,Emax])
```

Parameters: starting current: [A]
 end current: [A]
 time: [s]
optional: minimum potential limit (Emin): [V]
 maximum potential limit (Emax): [V]

Emin (V) and Emax (V) are optional and are used to provide lower and upper potential limits. When reaching such a functional limit, the defined ramp terminates the actual command and the sequencer continues with the next command in the sequence. The potential limit is provided to avoid unwanted over- and under-voltages which may damage the system under investigation. If such a limit is not necessary then it can be omitted from the command.

3.4 Hold at a Constant Potential/Current

A constant potential/current can also be applied for a defined time period.

Command for potentiostatic polarization:

```
hold_pot(potential,time[,Imin,Imax])
```

Parameters: potential: [V]
 time: [s]

optional: minimum current limit (I_{min}): [A]
 maximum current limit (I_{max}): [A]

One can also write the same command with PACT.

hold_pot(PACT,time[, I_{min} , I_{max}])

Parameters: PACT: actual potential
 time: [s]

optional: minimum current limit (I_{min}): [A]
 maximum current limit (I_{max}): [A]

PACT is the actual potential of the system under investigation. I_{min} (A) and I_{max} (A) are optional and are used to define the minimum and maximum functional limits of the current. When reaching such a functional limit, the defined polarization terminates prematurely and the sequencer continues with the next command in the sequence. The current limit is provided to avoid the unwanted high currents which may damage the system under investigation. If such a limit is not necessary then it can be omitted from the command.

Galvanostatic polarization:

hold_cur(current,time[, E_{min} , E_{max}])

Parameters: current: [A]
 time: [s]

optional: minimum potential limit (E_{min}): [V]
 maximum potential limit (E_{max}): [V]

One can also write the same command with CACT.

hold_cur(CACT,time[, E_{min} , E_{max}])

Parameters: CACT: actual current
 time: [s]

optional: minimum potential limit (E_{min}): [V]
 maximum potential limit (E_{max}): [V]

CACT is the actual current flowing through the system under investigation. E_{min} (V) and E_{max} (V) are optional and are used to provide minimum and maximum limits of the potential. When reaching such a functional limit, the defined galvanostatic polarization terminates prematurely and the sequencer continues to the next command in the sequence. The potential limit is provided to avoid unwanted over- and under-voltages which may damage the system under investigation. If such a limit is not necessary then it can be omitted from the command.

3.5 Set Sampling Rate

Define the data acquisition sampling rate at which the data should be recorded.

```
samples (sample_rate)
```

Parameters: sample_rate: [Samples/s]

The user may change the sampling rate at any time during a sequence hence different commands of a sequence can be recorded with different sampling rate. The defined sample rate is effective until a new sample rate is defined.

```
samples (sample_rate)
    command1
    command2
    ...

    samples (sample_rate)
        command1
        command2
        ...
```

Sequencer supports a maximum sampling rate of 1000 Hz for the main channel and 200 Hz for the ACQ channels.

3.6 Defining Loops

The sequencer allows loops to repeat a sequence of commands for a defined number of times.

```
do_loop (number_of_cycles)
    command1
    command2
    ...
end_loop
```

The user may also have integrated loops (loops within loops).

```
do_loop (number_of_cycles_1)
    command1
    command2
    ...

    do_loop (number_of_cycles2)
        command1
        command2
        ...
    end_loop
end_loop
```

4 Sample Sequences

In this section, different sample sequences are provided which can be used in the sequencer program.

Sequence 1:

A potential ramp from 0 V to 1 V in 10 seconds is repeated 50 times with a sampling rate of 20 samples per second.

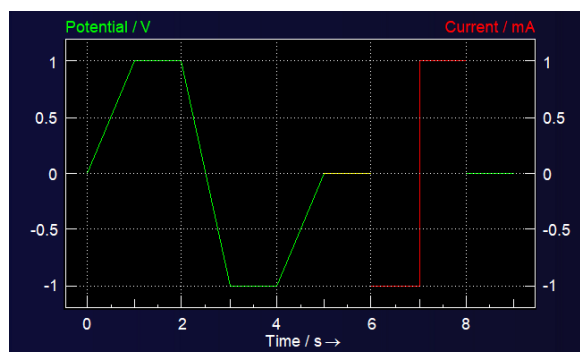
```
start_cycle                                'start sequence
  samples(20)                             'measure 20 readings per second
  do_loop(50)                             'do 50 loops
    ramp_pot_t(0,1,10,-3,3)              'starting potential = 0 V, end potential = 1 V,
                                          time = 10 s, minimum current limit = -3 A,
                                          maximum current limit = 3 A
  end_loop                                'end loop
end_cycle                                 'end sequence
```

The green-colored comments are not processed by the sequencer and should not be present in the sequence file. Comments are shown in this manual to explain the sequences.

Sequence 2

The sequence is visualized in the graph on the right side.

The potential measurements (potentiostatic) are shown with green color, OCP with yellow, and the current measurements (galvanostatic) with red color.



```
start_cycle                                'start sequence
  samples(20)                             'set sampling rate to 20 samples per second
  do_loop(20)                             'do 20 loops
    ramp_pot_t(0,1,1)                     'potential ramp from 0 V to 1 V with 1 V/s
    hold_pot(1,1)                         'polarize at 1 V for 1 second
    ramp_pot_s(1,-1,2)                    'potential ramp from 1 V to -1 V with 2 V/s
    hold_pot(-1,1)                        'polarize at -1 V for 1 second
    ramp_pot_s(-1,0,1)                    'potential ramp from -1 V to 0 V with 1 V/s
    samples(10)                           'set sampling rate to 10 samples per second
    ocp(1)                                'ocp scan for 1s
    hold_cur(-0.001,1)                    'galvanostatic polarization at -1 mA for 1 second
    hold_cur(1m,1)                        'galvanostatic polarization at 1 mA for 1 second
    hold_pot(PACT,1)                      'polarize at the actual potential for 1 second
  end_loop                                'end loop
end_cycle                                 'end sequence
```

Starting from the OCP measurement the sampling rate is reduced to 10 samples per second.

In this example, the current limits are not provided as they are optional.

In a sequence, between two commands a “dead time” of up to 100 ms is present. During these circa 100 ms the sequencer will not record any data. Hence the user must use a recommended minimum of 1 second time for each command. For example, a fast pulsing in the sequencer software with a pulsing time of less than 1 s will not be properly carried out.

Sequence 3

In the sequence, the user may use variables despite the intended values. These variables can then be defined between *start_variables* and *end_variables*. This way user does not have to change the values through the complete sequence and only changing the values for the *variables* will be sufficient.

```
start_cycle                                     'start sequence
  samples(20)                                 'set sample rate to 20 samples per second
  do_loop(5)                                  'do 5 loops
    ramp_pot_t(POT0,POT1,TIM1)                'potential ramp from POT0 to POT1 in TIM1
    hold_pot(POT1,TIM2)                       'polarize POT1 for TIM2
    ramp_pot_t(POT1,POT0,TIM1)                'potential ramp from POT1 to POT0 in TIM1
  end_loop                                     'end loop
  ramp_cur_t(CUR0,CUR1,TIM1)                  'current ramp from CUR0 to CUR1 in TIM1
  ramp_cur_t(CUR1,CUR0,TIM1)                  'current ramp from CUR1 to CUR0 in TIM1
end_cycle                                     'end sequence
```

```
start_variables 'DEFINE VARIABLE
  POT0=0        '0 V
  POT1=1        '1 V
  TIM1=1        '1 s
  TIM2=5        '5 s
  CUR0=0        '0 A
  CUR1=10m      '0.01 A
end_variables
```

Restrictions: In sequencer only 3 variables TIM, POT, and CUR can be used for time, potential, and current, respectively. The differentiation between similar variables (i.e., potential variables) can be made by using a number with the variable. Here, numbers from 0-99 are allowed in the sequencer.

The comments are not allowed in sequences. The sequencer will not load the sequences where the comments are also written in the sequences. In this manual, comments are written in the sequences to explain the commands used in the sequences.

5 Global Control Settings and Limits

In a sequence, different “control settings” can also be defined. This will automatically change the CONTROL SETTINGS in the Sequencer software when a sequence file is uploaded and the user does not have to modify the CONTROL SETTINGS manually. Upon reaching the defined global current or potential limits the sequence will be terminated. This is done to prevent under- and over-voltages or currents.

```

start_cycle      'start sequence
  samples(20)    'measure 20 readings per second
  do_loop(5)     'do 5 loops
  .              'sequence body
  .
  end_loop       'end loop
end_cycle        'end sequence

start_online     'start global settings and online display configuration
  max_ti=1000    'maximum measurement time in hours - (range: 0.1 - 1000 h)
  cur_hi=2m      'upper current limit in amperes - (dummy value: 2 mA)
  cur_lo=0       'lower current limit in amperes - (dummy value: 0 mA)
  pot_hi=400m    'Upper potential limit in volts - (dummy value: 400 mV)
  pot_lo=-100m   'lower potential limit in volts - (dummy value: -100 mV)
  cur_ra=100m    'current range in amperes - (dummy value: 100 mA)
  pot_of=0       'potential latency window (POT-OFF)
                  'pot_of=-1 → do not turn off
                  'pot_of= t → POT-OFF after "t" seconds
                  'pot_of= 0 → turn off immediately
                  'pot_of= 2 → turn off after 2 seconds if the potential does not
                  decrease to the desired potential range in the 2 seconds
  cur_of=0       'current latency window (Current off) -Same as POT-OFF
  end_on=0       'defines if the potentiostat should be left turned on with the
                  last sequence setting or should the potentiostat be turned off
                  after the sequence finishes (1=on / 0=off)
  rodrops=10     'Resistance value in ohms for IR-drop compensation. "rodrops=0"
                  will deactivate the IR-drop compensation.- (dummy value: 10 Ω).
  acqena=0       'Enable (0) or disable (-1) ACQ recording.
end_online       'end global configuration

```

The “pot-of” and the “cur-of” decide what to do if the potential and the current values exceed the described global limits, respectively. The “pot-of” and the “cur-of” are named “potential latency window” and “current latency window” in the Potentiostat & Control settings.

The “cur_ra” defines the current range or the shunt in the potentiostat for the potentiostatic measurements in the sequence. If a smaller current range is selected than needed for the desired potentiostatic measurement then the potentiostat will not be able to correctly carry out the potentiostatic measurement as the smaller shunt will limit the functionality of the potentiostat. As Auto-ranging of the shunts is not available during the potentiostatic measurements in the sequencer hence it is crucial that the user choose a correct current range before starting the sequence. For the galvanostatic measurements, an appropriate shunt will be set before running each galvanostatic command in the sequence.

POT bandwidth:

During current ramps, if the potentiostat shows noise then in the CONTROL SETTINGS, change the POT bandwidth to a higher value. POT bandwidth values from 0 to 3 are possible. A comparatively higher POT bandwidth value entails

1. Shorter rise time
2. Higher noise
3. Lower immunity to oscillations

and vice versa. The POT Bandwidth with different values comprises different ranges.

Bandwidth 0 = 10 MHz

Bandwidth 1 = 1 MHz

Bandwidth 2 = 100 kHz

Bandwidth 3 = 20 kHz

Parameter	Value
max runtime / h:	1K
upper current limit / A:	2m
lower current limit / A:	0
upper potential limit / V:	400m
lower potential limit / V:	-100m
current range / A:	100m
potential latency window:	0
current latency window:	0
POT bandwidth (10MHz):	0
POTSTATE at END:	0
timebase (0=6/1=12/2=30):	0
auto filename:	1
R-Odrop:	10
Odrop IntTime/ms:	1

It is highly advised that the user defines the global parameters in each sequence file to avoid any problem with the measurement.

IR-drop compensation:

The sequencer continuously carries out the IR-drop compensation during the measurement. The "Odrop IntTime" in the "Potentiostat and Control settings" window defines the time span for which the potentiostat integrates the current for the IR-drop compensation. A very small "Odrop IntTime" values means that the time span to integrate the current is very short. In such cases, the noise in the current signal may effect the IR-drop compensation.

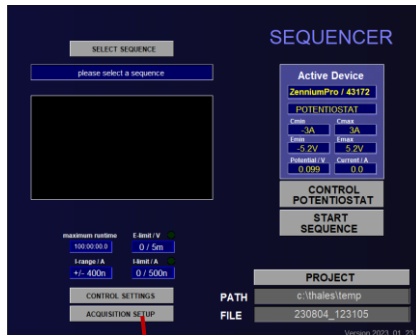
R in R-Odrop is reset to 0 after every measurement. This is done to prevent unintended R-Odrop compensation in the next sequence measurement.

When a value (r) for IR-drop compensation is set in the global settings of the sequencer software then the Zahner's potentiostat applies a higher voltage than defined using the following formula to compensate for the voltage drop across the series resistance.

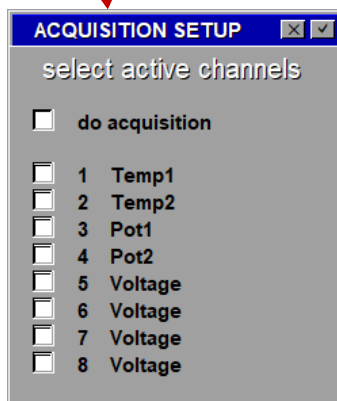
$$\text{Applied Voltage} = \text{Set Voltage} \cdot \frac{(R + r)}{R}$$

Here R is the resistance of test object and r is the series resistance. Depending upon the r , it could be that the applied voltage value varies greatly from the set voltage which may trigger the global parameter limit prematurely ending the measurement.

6 Additional Acquisition Channels



Open the ACQUISITION SETUP window



Do acquisition = global enable/disable for ACQ recording

Select the additional acquisition channels for the SEQUENCER recording. All acquisition channels which are displayed in the Thales ACQ software are available.

Setup ACQ input channels and defining available channels (displayed channels) is described in the [signal acquisition manual](#).

7 Incorporating a Sequence in SCRIPT

The "Sequencer" only runs the DC tests. If the user wants to carry out different DC and AC measurements in a single run then the sequencer can be used with Zahner's script program. For this, the user may write the sequence(s) and then save the sequences in the folder `c:\thales\script\sequencer\sequences` following the standard naming format. Once the sequence(s) are saved then the user can call them in the script program and then run the sequences before or after AC measurements (i.e., EIS) or other standard measurements (CV).

Example:

The following script carries out a two sequences with an EIS in between.

1. Start
2. Sequence 1
3. EIS measurement
4. Sequence 2
5. End

```
SCRIPT1
SEQUENCE%=1          'load c:\thales\script\sequencer\sequences\sequence01
gosubLOADSEQUENCE    'go to sequencer module
gosubEXECUTESEQ      'do sequence
if (SEQERROR%=0) trueif
endif
MEAS_OPEN_EIS(65,"c:\thales\script","eis")      'open EIS rule file
MEAS_EIS                                           'record EIS spectrum
MEAS_SAVE_EIS(65,"@c:\thales\script","eis0")     'save measured EIS
        SEQUENCE%=2          'load c:\thales\script\sequencer\sequences\sequence02
        gosubLOADSEQUENCE    'go to sequencer module
        gosubEXECUTESEQ      'do sequence
gotoMENU                                           'go to menu

'****As the CODE below allows the incorporation of the sequences in the
script program hence it must not be changed and should be added without
modification to the end of the script.****
:
LOADSEQUENCE::
pushSEQUENCE%
gosub"seq*,p",LOADSEQEXT
BRSEQDONE::
pullSEQERROR%
lprintSEQERROR%
if (SEQERROR%=0) trueif
    SEQUENCE_LOAD%=-1
    RESULT$="SELOK:"
    a$=RESULT$+" sequence"+fnSTR$(SEQUENCE%)+ " loaded"
falseif
    SEQUENCE_LOAD%=0
    RESULT$="SELEERROR:"
    a$=RESULT$+" in sequence"+fnSTR$(SEQUENCE%)
endif
lprinta$
return
:
EXECUTESEQ::
```



```

gosubSET_SEQU_FILE
gosub"seq*,p",DOSEQEXT
pullSEQERROR%,SEQERR$
if (SEQERROR%=0) trueif
    RESULT$="SEQOK:"
falseif
    RESULT$="SEQERROR"+fnSTR$(SEQERROR%)+ " "+SEQERR$+":"
endif
lprintRESULT$
return
:
START::
    on (1-INIT%) gosubDEFAULTS
    INIT%=1
return
:
DEFAULTS::
    gosubPF
    gosub"seq*,p",INIT
    INIT%=1
return
:
SET_SEQU_FILE::
    SEQ_PATH$="@c:\thales\temp"
    SAVEFILE$="Sequence"+fnSTR$(SEQUENCE%)
    pushSEQ_PATH$,SAVEFILE$
    fnSEQ_EXE ("pullREMO_ASPATH$,REMO_ROOTFILE$")
return
:
deffnSEQ_EXE (a$,i,a)
    whileinstr (a$,"'")
        i=instr (a$,"'")
        a$=left$(a$,i-1)+chr$(34)+mid$(a$,i+2)
    wend
    a$=a$+":push0:return"
    pusha$
    gosub"seq*,p",EXECUTE
    pulla
fnenda
:
SCRIPT_END

```

The user may carry out any measurement or a sequence of measurements before or after running a sequence. In the script, only the first few lines are the main script body, the rest is the code which is required to run the sequencer in the script program. If a user wants to run the sequencer in their script then they can just copy paste the code for running the sequencer directly from this manual in their script and can easily run the sequencer in the script program.