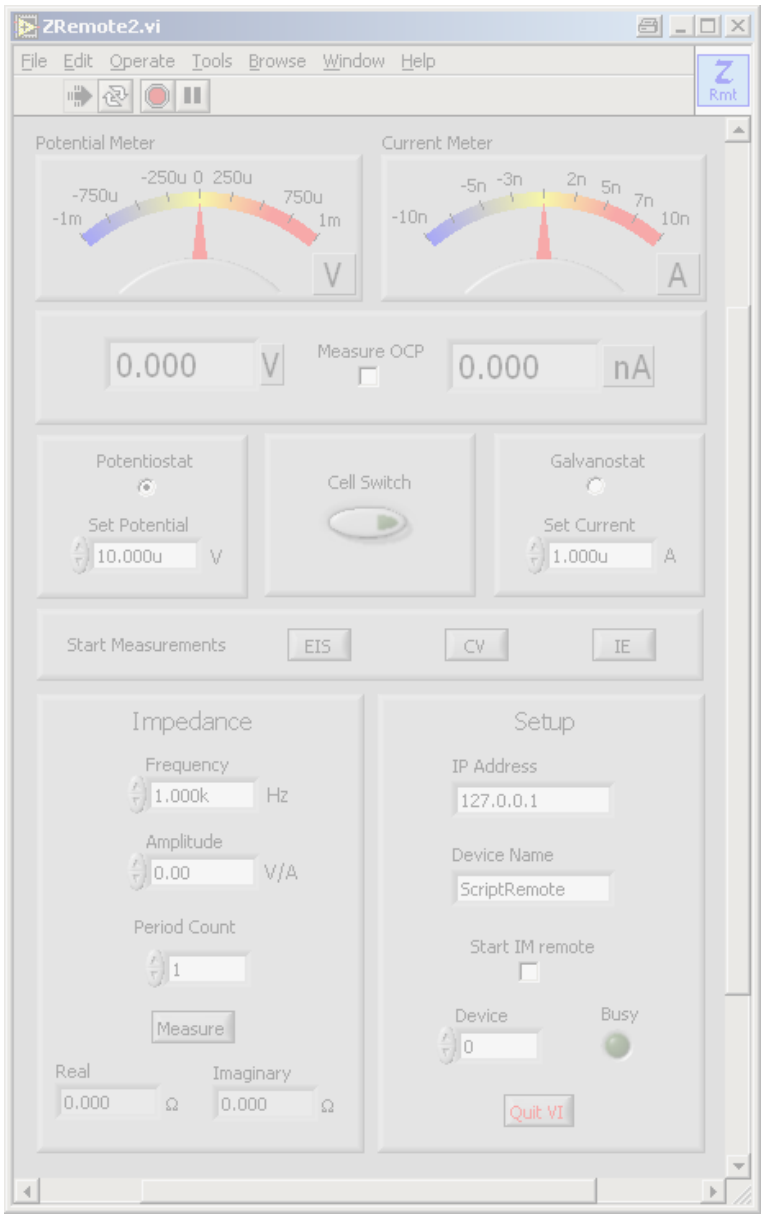
Remote

Remote Control of a *Zennium/IM6* by LabVIEW



1. Introduction

Normally, the *Zennium/IM6* systems are controlled by the Thales software package which is optimized for measuring and analyzing electrochemical data. In addition, the *Zennium/IM6* systems are prepared to be controlled by external software using the TCP/IP protocol. This is not limited to the local Thales-PC but is also possible over network.

For that purpose, a network card running the TCP/IP protocol has to be installed in the Thales-PC. A DLL-file is used to interface between external software (e.g. LabVIEW) and the Thales interface. A SCRIPT procedure running on the *Zennium/IM6* will call the specific procedures for each of the commands and parameters.

A LabVIEW VI controlling the basic functions of the electrochemical workstation is supplied by Zahner. This LabVIEW VI can also be used as a basis for user applications.

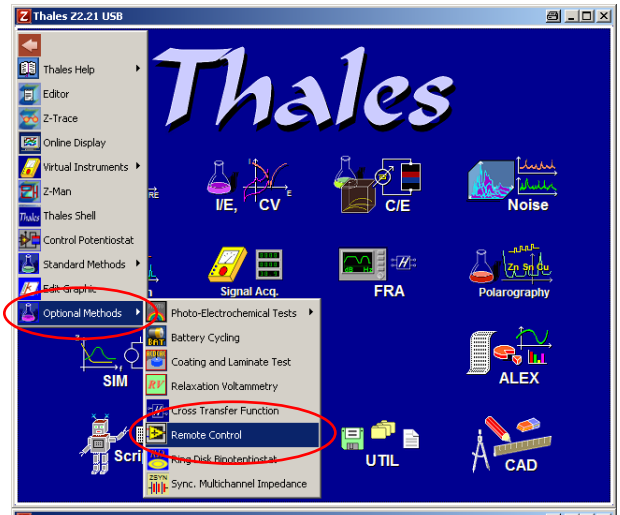
2. Thales Server

In order to enable remote control of Thales a special application has to be started. This can be done either by starting the runtime file, linking a Thales script source or automatically start the remote application during startup of Thales. Select one of the methods described in chapters 2.1, 2.2 or 2.3.

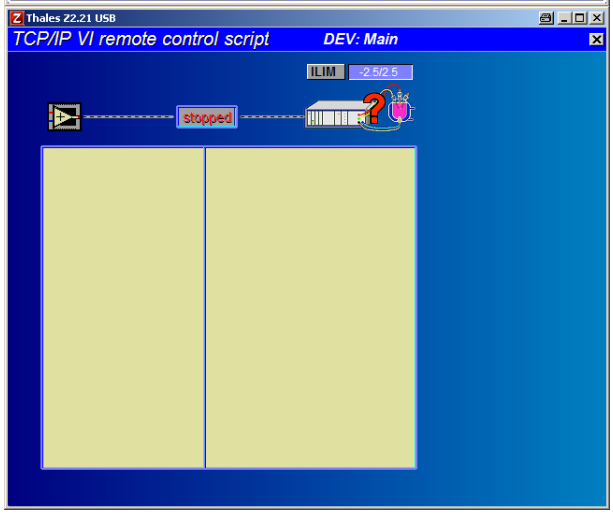
2.1 Running the supplied runtime file

In case no modifications of the remote application are necessary starting the runtime file is the most convenient way.

Start the Thales application and click the red *Z*-icon to open the pull down menu.



In the pull down menu choose Optional Methods and click Remote Control.

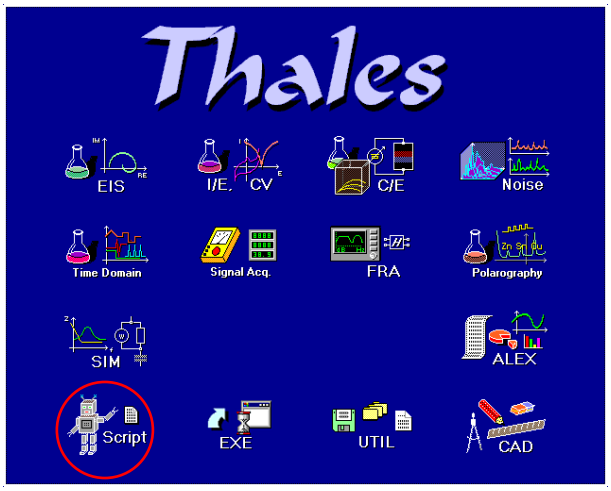


The remote application starts automatically and awaits a client connection.
At this stage, the communication indicator reads stopped because the communication is not set up so far. It will start as soon as the LabVIEW VI or another controller software is started.
This screen has to be active during the whole remote session. Quitting this screen will stop the remote session.

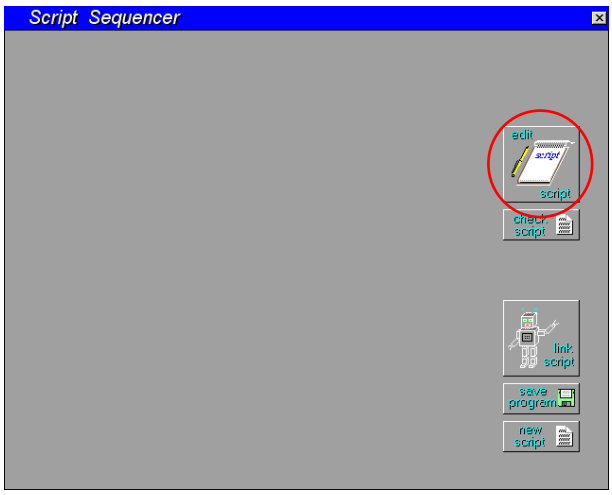
Alternatively the runtime file can be loaded with the EXE-icon of the Thales main screen. Navigate to c:\thales\examples\applications and select remote.rtm.

2.2 Linking the script source

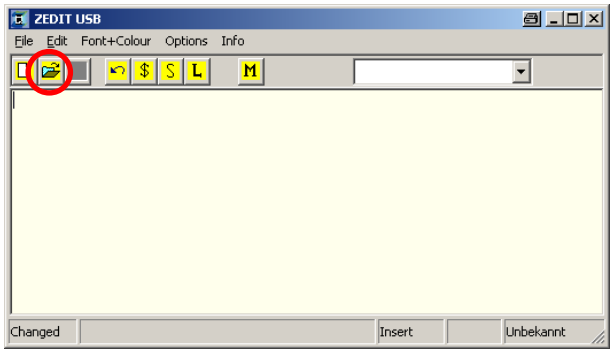
In case modifications of the remote application are desired the script source is provided for editing.



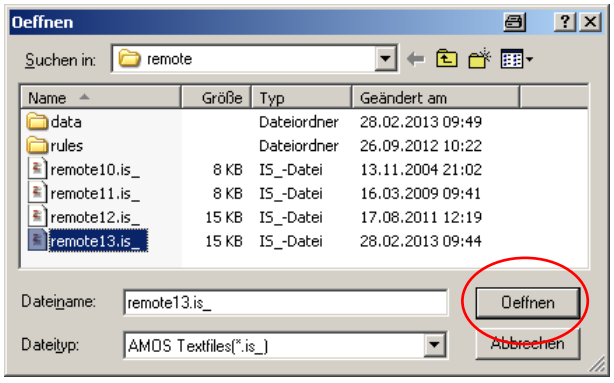
Start the Thales application and activate Script.



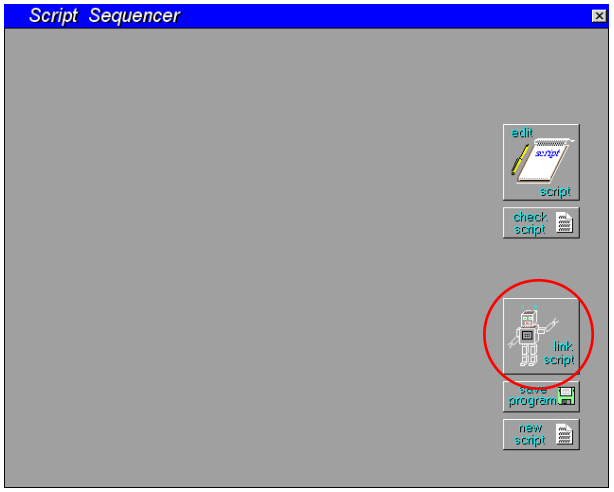
Click the edit script button in order to open the Zahner editor ZEdit.



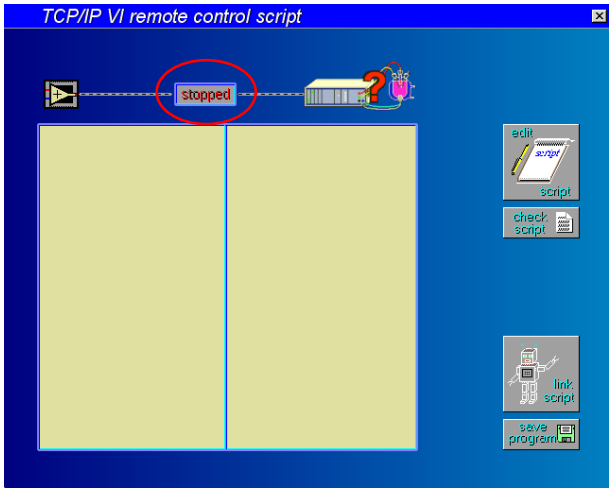
In ZEdit click the import button.



Select the source file with the highest version number, (here *remote13.is_*) in the folder c:\THALES\script\remote and load it. The source code can now be edited in ZEdit if necessary.



Return to the Thales window and click the button link script.



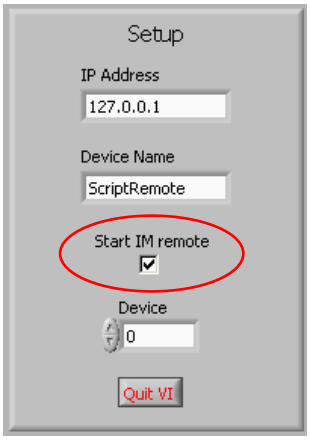
After linking the script is started automatically.
At this stage, the communication indicator reads stopped because the communication is not set up so far. It will start as soon as the LabVIEW VI or another controller software is started.
This screen has to be active during the whole remote session. Quitting this screen will stop the remote session.
Using the save program button a runtime file can be saved for future use as described in chapter 2.1.

2.3 Remote Boot of a *Zennium/IM6*

When using the *Zennium/IM6* at distant places it can even start the remote server without user interaction. It can be simply activated by switching the “Start IM remote” option on before running the LabVIEW VI.

For a more comprehensive explanation of the Remote Boot feature, please refer to the Devcli.pdf – manual, paragraph “5. Synchronization”, function “DevRestartViaPD”.

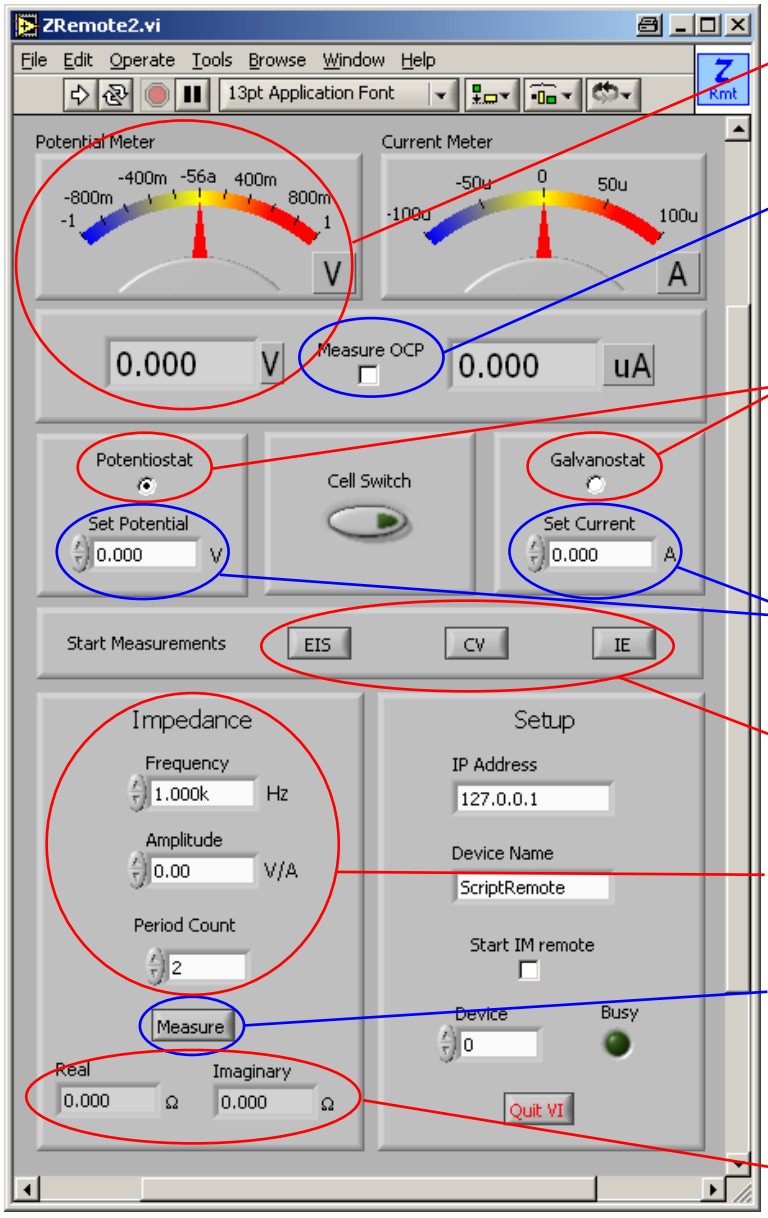
Please note, the former technique of sending an NMI to autostart the *Zennium* is no deprecated.



3. LabVIEW Client

The sample application ZRemote.vi is located in the folder c:\THALES\VI\IM. It uses the VIs Autoscale.vi, HandleUserErrors.vi, Sendcommand.vi, Startup.vi, and the library flclient6.llb. This library integrates the function calls to DevCli.dll into LabVIEW.

All LabVIEW VIs are saved in the LabVIEW 6.1 file format so they can be used with older versions of LabVIEW. But they are also compatible with recent versions like 8.5.



Analog and digital auto-ranging meters display the measured potential.

Measure the **o**pen **c**ircuit **p**otential i.e. potential when cell is switched off

Set operating mode to potentiostatic or galvanostatic mode.

Enter set point potential (potentiostatic mode) and set point current (galvanostatic mode).

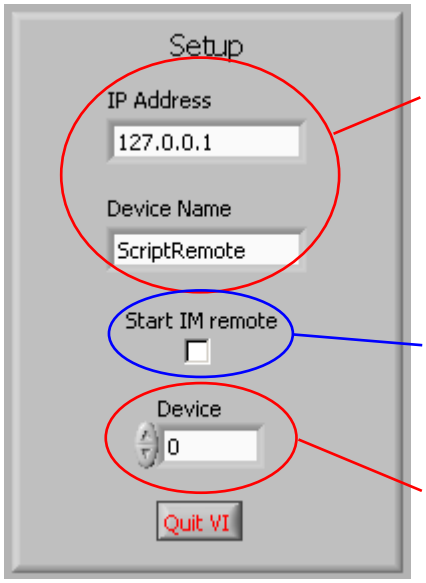
Start a complete EIS, CV, IE measurement

Enter parameters for a impedance measurement

Start impedance measure-ment with preset parameters.

Result of last impedance measurement

The box on the lower right hand side deals with setup of communication and choice of the used device.



IP-Address of Thales-PC acting as server and device name coded in the remote script (default: ScriptRemote).
These inputs are disabled while execution of the LabVIEW VI. Preset these settings before starting the LabVIEW VI.

Start remote server at start of LabVIEW VI. This checkbox is disabled during execution. Preset before starting the LabVIEW VI.

Active potentiostat: 0 internal potentiostat,1, 2, 3, … channels of EPC40, RMux

Please note the following hints:

Do not use the Stop-Button (red circle) of LabVIEW to quit the LabVIEW VI. The communication is not terminated correctly in this case. Always use the “Quit VI” button on the front panel.
If the Stop-Button was used inadvertently restart LabVIEW to reset communication.

The radio buttons for potentiostatic and galvanostatic control can be both activated when the LabVIEW VI is not running. At start of the VI the Potentiostat button is read out and the Galvanostat button set accordingly.

At startup the LabVIEW VI tries to establish communication for a maximum time of 5 s. If this fails an error message is displayed. Activating “Start IM remote” extends this timeout to 60 s.

The analog meters are autoranging. To avoid flickering of the range the last five values are taken into account when switching to a lower scale. If a new value exceeds the current scale rescaling takes place instantaneously.

4. Communication Functions of Remote Control

The communication functions are described in the DevCli.dll Programmer's Reference Manual.

5. Instruction Set of Remote Control

5.1 Format of command strings

Communication uses ASCII-strings of the following format:
X:Command:
X number of the communication channel from 1 to 9. Up to now Thales Remote does not distinguish different channels, so channels 2 to 9 should be considered as reserved.
Command one of the keywords given below.
The command has to be delimited by a colon (:).

Several commands can be sent in one string. In this case all commands are separated by colons (:), e.g.
X:Command1:Command2:Command3:

Sequence of execution is fixed to DEV%, Pset, Cset, Frq, Ampl, Nw, Pot, Gal, GAL, SENDSETUP, CURRENT, POTENTIAL, IMPEDANCE, EIS, CV, IE.

Values beyond the limits of the active device are ignored.

Examples of valid communication strings:

1:Pset=1.5:	Sets the potential to 1.5 V
1:CURRENT:POTENTIAL:	Returns the actual current and potential

5.2 Set up parameters

DEV%=X

Set the active potentiostat. Setting non existent devices is ignored.

Parameter X

0 main potentiostat of IM6
> 0 external potentiostat connected to channel X of EPC

Pset=X

Set the potential. Values beyond the potential limit of the active device are ignored.

Parameter X

Potential in V.

Cset=X

Set the current. Values beyond the current limit of the active device are ignored.

Parameter X

Current in A.

Frq=X

Set the frequency. Values beyond the frequency range of the active device are ignored.

Parameter X

Frequency in Hz. Range IM6 10µHz – 8 MHz, Zennium 10 µHz – 4 MHz

Ampl=X

Set the AC amplitude. Values beyond the limits of the active device are ignored.

Parameter X

AC amplitude in **m**V (potentiostatic) or **m**A (galvanostatic). Range 0 – 1000.

Nw=X

Set the number of AC periods per measurement point. Values beyond the limits are ignored.

Parameter X

Number of AC periods. Range 1 - 100

Pot=X

Switch the potentiostat on or off.

Parameter X

-1 potentiostat enabled
0 potentiostat disabled

Gal=X and GAL=Y

Set the ECW to potentiostatic, galvanostatic or pseudo-galvanostatic mode. Although Gal and GAL are two independent commands they are normally used together.

Parameters X and Y

The following table gives the arguments for setting the different modes.

	GAL = 1	GAL = 0	GAL = -1
Gal = 0	-	potentiostatic	pseudo-galvanostatic
Gal = -1	galvanostatic	-	-

GlobalAck=X

Switch acknowledge of set commands on or off

Parameter X

1 acknowledge of set commands switched on (default)
0 acknowledge of set commands switched off (for legacy clients)

In former versions of the remote application, not all of the commands return an “acknowledge-string”, i.e. return a “result” to the client. This restriction is now eliminated, resulting in a simpler handshaking.
By default the remote application will now return “OK” if no other response is to be sent. To disable this feature, the client must send the command “GlobalAck=0”. Please note that when activated (“GlobalAck=1”) the client must read the “acknowledge-string” using the DevReadCont-function.
This feature is important considering the new functions implemented in the devcli.dll in October 2018. Please refer to the manual “devcli.pdf”, paragraph 5. Synchronization.

5.3 Measure single values

CURRENT

Measures and returns cell current in Ampere.

Parameters

None

Return value

current= -#.@@@^^^^: (see ANDIbasic manual, command “using” for meaning of wildcards)
Example: current= -9.937e-07:

POTENTIAL

Measures and returns cell potential in Volt.

Parameters

None

Return value

potential= -##.@@@: (see ANDIbasic manual, command “using” for meaning of wildcards)
Example: potential= -0.100:

IMPEDANCE

Measures and returns impedance at the preset frequency as real and imaginary part in Ω. Before calling IMPEDANCE frequency and amplitude have to be set and cell has to be switched on. Otherwise “Impedance disabled!” is returned. The command CURRENT should be issued before IMPECANCE is called in order to automatically set the optimal current range of the ECW.

Parameters

None

Return value

Impedance disabled!:
Potentiostatic loop interrupted!:
impedance= -#.@@@^^^^,-#.@@@^^^^: (see ANDIbasic manual, command “using” for meaning of wildcards)

Example: impedance= 4.926e+04,-1.072e+06:

5.4 Run Complete Measurements

5.4.1 EIS – Impedance Spectrum

EIS

Measures a complete impedance spectrum and saves the result to hard disk. The settings (frequency range, etc.) can be either read from a rule file or separately set by the client before starting the measurement. The two alternatives are described below.

The measured spectrum is saved in both Zahner binary and ASCII format to the files eis_*timestamp*.ism and eis_*timestamp*.ism in the path c:\thales\script\remote\data.
The paths and filenames can be changed in the Script source remote*XX*.is_ if necessary.

Parameters

None

Return value

EIS DONE: - is sent after measurement is completed.

5.4.1.1 Parameters from rule file

In case of fixed parameters for the EIS measurements the use of a rule file is a handy way of concurrently setting all parameters (frequency range, etc.). As a standard the parameters are read from the file eis_rule.ism in the path c:\thales\script\remote\rules\.
In order to create a rule file enter the EIS program and preset all desired parameters. Record a few measurement points and save the measurement as eis_rule.ism.
When starting up remote control use of rule files is the default setting. As described in chapter 5.4.1.2 this can be switched to individual parameters sent by the client. In order to return to the use of rule files send the command

UseRuleFile=1

5.4.1.2 Parameters sent separately by client

When using frequently changing parameters for the EIS measurements the use of rule files can be cumbersome. Therefore the parameters can be set individually by the remote client. In order to activate use of individual parameters send the command:

UseRuleFile=0

This enables the use of the following parameters. **value** is considered an integer or floating point number (also in exponential representation) for the parameter. State of potentiostat and all parameters have to be set before the measurement is started with **EIS**.

Fmin=value

Minimal frequency of the spectrum measured.

Range: 1e-5 to 8e6 depending on used device.

Fstart=value

Starting frequency of the spectrum measured.

Range: Within Fmin and Fmax, 1e-5 to 8e6 depending on used device.

Fmax=value

Maximum frequency of the spectrum measured.

Range: 1e-5 to 8e6 depending on used device.

dfl=value

Steps per decade below 66 Hz.

Range: 1 to 124

dfm=value

Steps per decade above 66 Hz.

Range: 1 to 124

Nwl=value

Number of measured periods below 66 Hz.

Range: 1 to 999

Nws=value

Number of measured periods above 66 Hz.

Range: 1 to 999

ScanStrategy=value

0 single sine
1 multi sine
2 frequency table

ScanDirection=value

0 from Fstart to Fmax and down to Fmin
1 from Fstart to Fmin and up to Fmax

5.4.2 CV – Cyclic Voltammogram

CV

Run a complete cyclic voltammetry measurement and save the result to hard disk. The settings (vertex potentials, etc.) are read from the file cv_rule.ism in the path c:\thales\script\remote\rules\.
In order to create a rule file enter the CV program and preset the parameters. Record a few measurement points and save the measurement.

The measured cyclic voltammogram is saved in both Zahner binary and ASCII format to the files cv_*timestamp*.ism and cv_*timestamp*.ism in the path c:\thales\script\remote\data.
The paths and filenames can be changed in the Script source remoteXX.is_ if necessary.

Parameters

None

Return value

CV DONE: - is sent after measurement is completed.

5.4.3 IE – Steady State Current/Voltage Curve

IE

Measure a complete current voltage curve and save the result to hard disk. The settings (vertex potentials, etc.) are read from the file ie_rule.ism in the path c:\thales\script\remote\rules\.
In order to create a rule file enter the I/E program and preset the parameters. Record a few measurement points and save the measurement.

The measured I/E-curve is saved in both Zahner binary and ASCII format to the files ie_*timestamp*.ism and ie_*timestamp*.ism in the path c:\thales\script\remote\data.
The paths and filenames can be changed in the Script source remoteXX.is_ if necessary.

Parameters

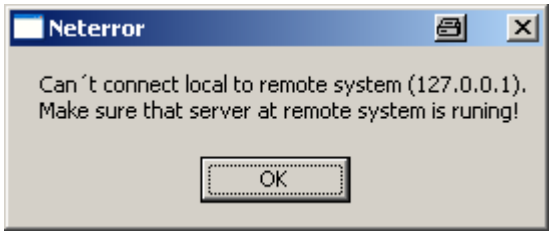
None

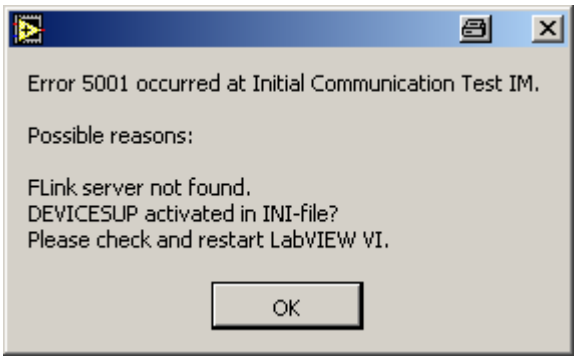
Return value

IE DONE: - is sent after measurement is completed.

6. Troubleshooting

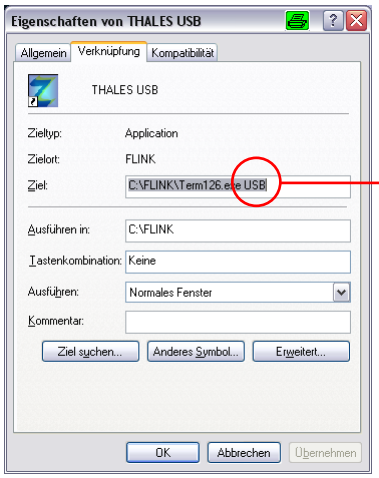
When running Zremote2 the following error messages are displayed. Zremote2 cannot connect to Thales.





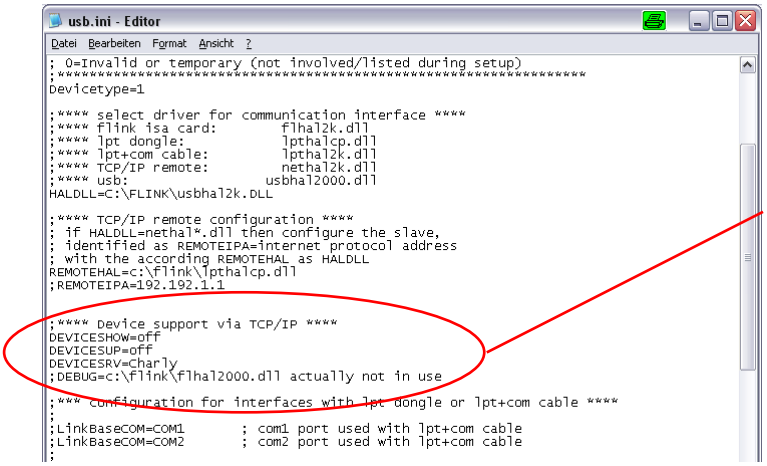
Make sure the TCP/IP protocol is installed and active on your PC.

Right-click the *Thales* icon on your desktop and select *Properties*. Check which .ini file is in use. This entry you will find in the *Target* line after the entry “c:\flink\termXXX.exe” with XXX as version number. For example: “c:\flink\term129.exe USB” means that the “usb.ini” file is in use.



This is the name of the .ini file used by the Thales driver (in this example usb.ini)

Open this .ini file which is located in the c:\flink folder of your computer hard drive. Check whether the parameter “DEVICESUP=on” (section “Device support via TCP/IP”) is set. If it is set to “off” change it to “on”. Save and close the editor window.



TCP/IP settings in the usb.ini file. Set the parameters as follows:

DEVICESUP=on
DEVICESRV=Charly
(Instead of Charly put in the server name or IP-address, e.g. 127.0.0.1 here)

Leave all other settings as they are!
Save the edited .ini file and close the *Notepad*.

Please also check, the Windows® firewall does not block incoming connections to AMOS Terminalemulation.

Data files are not saved when using the EIS / CV / IE command.

Ensure the output folder for data exists and the user running the Thales software has sufficient rights to write to this folder.
Running remote.rtm the data folder is c:\thales\script\remote\data. This folder can be changed by modifying the script source remote13.is_. Locate the line `DATAPATH$="c:\thales\script\remote\data"` and change the path.

After update of Thales to version Z2.21 or later third party LabVIEW VIs cannot communicate with Thales.
In order to improve stability communication was changed to a FIFO-buffered mode. Replace DevRead with DevReadCont and DevWrite with DevWriteCont in the LabVIEW VI in order to use the FIFO-mode.