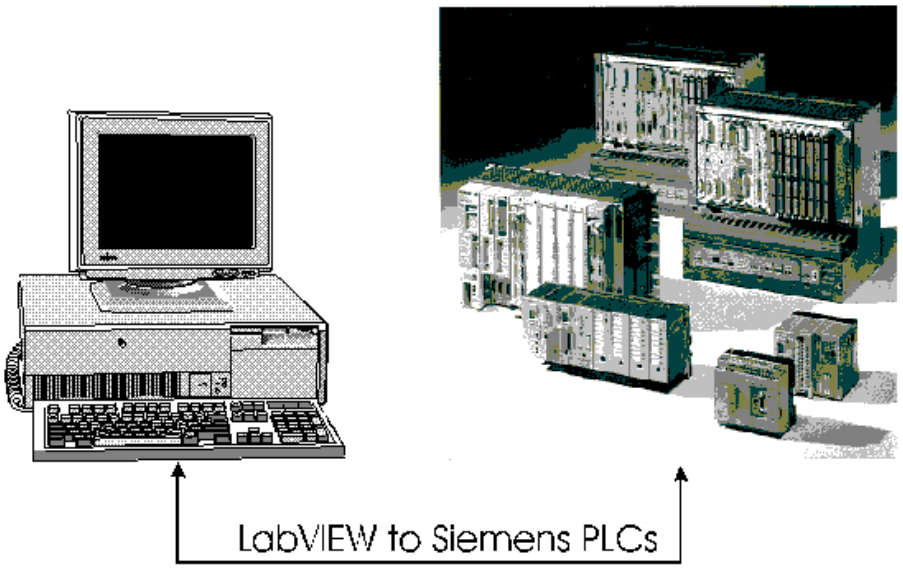
CIT Engineering NV
Kleinhoefstraat 6 - B-2440 Geel - Belgium
Tel. +32 (14) 57 96 50 - Fax +32 (14) 57 96 51
E-mail : info@citeng.com
Website : www.citeng.com
Bank : CERA 733-3093437-40
BTW : BE 459 001 426

SinecVIEW

Sinec-L1 protocol for LabVIEW 5.1

User Manual

Windows 3.1/95/NT Mac and PowerMac version



User Manual

SinecVIEW
Sinec-L1 protocol for LabVIEW 5.1

For :

- AT-PC running DOS/Windows 3.11/95/NT
- Macintosh
- PowerMac

Note:

We are constantly attempting to improve the quality of our manuals and products.

Should you have any suggestions or recommendations for improvement, we would appreciate it if you would send your recommendation to CIT

Revision 5.1 D
December, 08

Comment [DH1]:

Limited warranty & copyright

Important warning

Because of the variety of uses of this product, and the impossibility of our anticipating every use, those responsible for using and applying this product must satisfy themselves that it is appropriate and acceptable for their application. That responsibility includes using this product in accordance with all engineering, programming and safety practices and particularly in accordance with the use of Siemens PLCs as recommended by the Siemens Company.

License

You have a nonexclusive license to use the SinecVIEW software. You may use this product on one computer at a time. For backup purposes only, you may copy SinecVIEW software to other disks which you must retain in your possession and control. You may not copy SinecVIEW software to any other computer by any means for any other purpose.

Limited warranty

CIT warrants both this manual and the SinecVIEW program disks to be free from defects for one year from the date of purchase. If during this warranty period, you find that this manual or the SinecVIEW disks contain physical defects, return the defective item along with proof of purchase and description of the defect. You will receive a replacement at no charge.

CIT makes no warranty or representation, express or implied, with respect to this software, its performance or its fitness for a particular purpose. As a result, the software is sold “as is” and you, The purchaser, assume the entire risk to quality and performance. In no event shall CIT be liable for direct or indirect, special incidental or consequential damages resulting from any defect in the software or its documentation. Such damages include but are not limited to loss or profit, opportunity, program or data.

This warranty excludes all other warranties, oral or written, express or implied.

Copyright

Intended Audience

This manual assumes you are familiar with Macintosh computers or with Windows 3.1/95/NT and the basic Macintosh or Windows operations, such as pointing and clicking, launching an application and moving files. Because SinecVIEW works only with LabVIEW 5.1 software and Siemens PLCs , familiarity with LabVIEW 5.1 and Siemens PLCs is assumed. If you have just purchased LabVIEW 5.1 or PLCs, you should familiarize yourself with those products before you attempt to install or use SinecVIEW.

Table of Contents

1. Introduction

This is a brief manual describing the use of the **Sinec-L1** library as well as for an AT-PC running Windows 3.1/95/NT as for a Macintosh.

Sinec-L1 is a member of **SinecVIEW,** a set of three drivers for LabVIEW 4.0 to communicate with Siemens PLCs of the S5 series.

- Sinec-L1 (AT-PC Windows 3.1/95/NT /Mac and PowerMac)
- Sinec-L2-FDL (AT-PC Windows 3.1 only)
- 3964R (AT-PC Windows 3.1/95/NT /Mac and PowerMac)
- Sinec-L2-DP (AT-PC Windows 3.1/NT)

With SinecVIEW there are four different ways to connect the Siemens PLCs to LabVIEW 3.1.1.

- The first way is by the **Sinec L1** protocol.

The Sinec-L1 VI can communicate on a L1 point-to-point basis for single PLC applications or on a L1 network basis (up to 30 PLCs). The PC is always the Master. The PLC can be connected via a Siemens BT 777 bus terminal or a programming cable (only point-to-point) to a serial port of the PC. No special plug-in board is required for the PC.

- A second way (AT-PC/Windows 3.1 only!) is by the **Sinec L2-FDL** protocol.

The Simatic S5 apparatus can be connected to the PC via Siemens hardware in a network of PLCs and PCs (up to 127 stations). You need a special L2 communication plug-in board in your PC (CP 5412 (A1) or CP 5412 (A2)) .

- The third way to access data on the Siemens PLC is by the serial **RK512 (3964R)** protocol.

With a communication processor (for S5: CP524 or CP525 and CPU 928B with RS323C interface module; for S7-400: CP 441-2) that is configured with the RK512 (3964R) protocol it is possible to connect the module directly to one of the serial ports of the PC. No special plug-in board is required for the PC.

- A fourth way (AT-PC/Windows 3.1/NT) is by the **Sinec L2-DP** protocol.

The Simatic S5 and -S7 apparatus can be connected to the PC via Siemens hardware in a network of PLCs and PCs (up to 125 stations). You need a special L2 communication plug-in board in your PC (CP 5412 (A2)).

These drivers can be ordered separate.

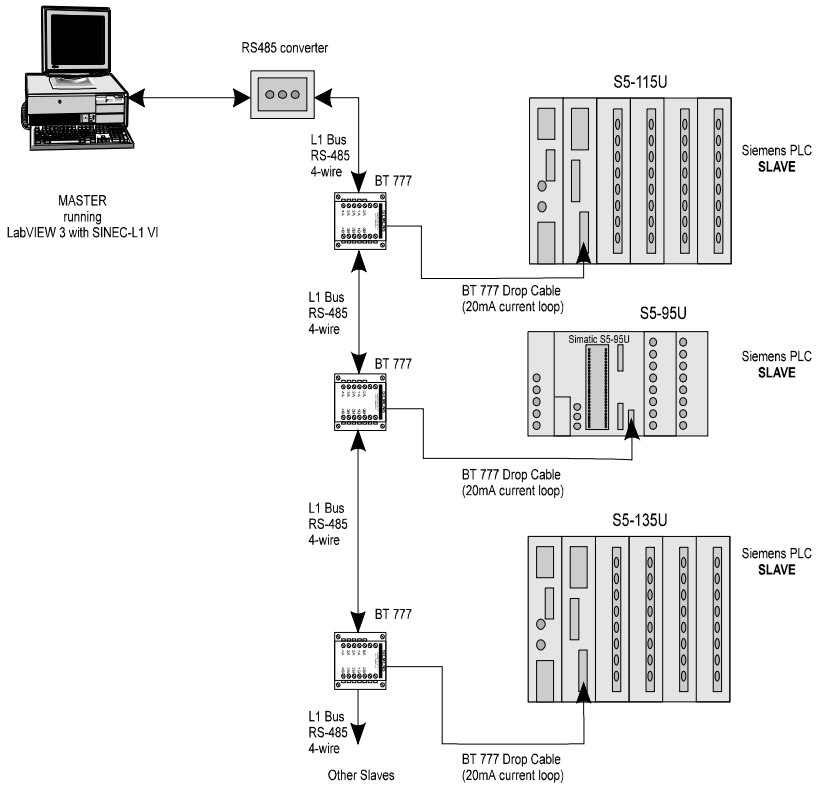
2. The Siemens Sinec L1 Network

2.1. The Sinec L1 Network

The Siemens Sinec L1 low speed network allows Siemens Simatic S5 PLCs of the U family to communicate with each other over a common bus (4-wire, RS-485). The network uses the MASTER-SLAVE principle. The data transmission rate is 9600 Baud. Up to 30 slaves can be connected by the standard Siemens BT-777 bus terminals.

With the L1 protocol the PC becomes the Master who coordinates and monitors the data traffic on the network, so you do not need a PLC configured as Master.

The following illustration shows a typical L1 network configuration.

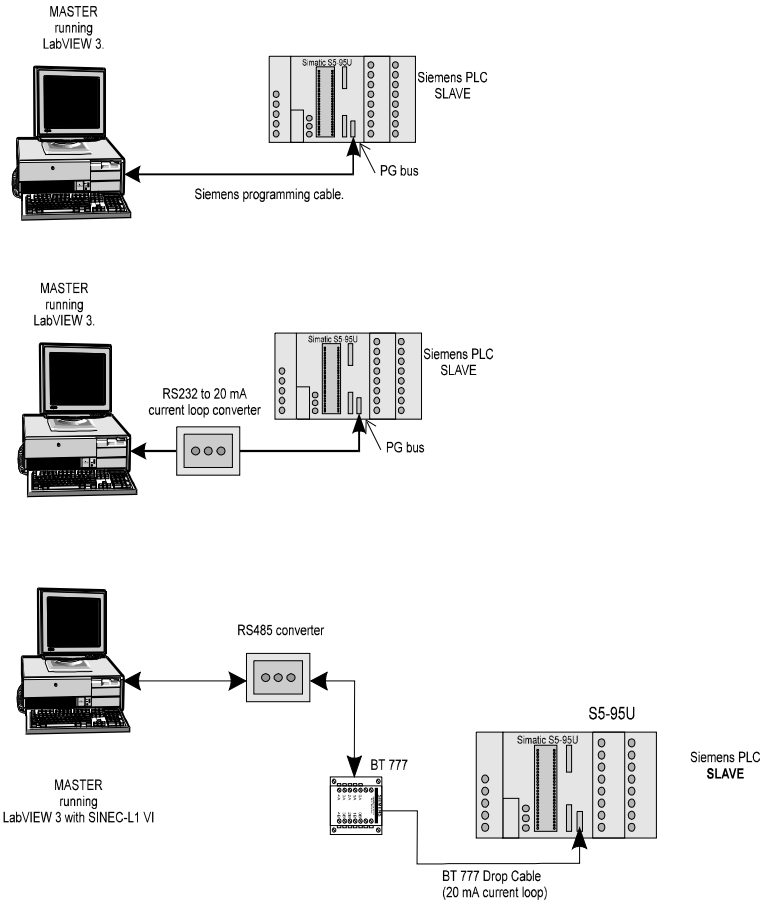


The L1 network.

2.2. The Sinec L1 point-to-point connection

With the Sinec L1 protocol you also can communicate with a PLC on a ***point-to-point*** basis for single PLC applications.
You can use a Siemens programming cable (only short distance) or an RS-232 to 20 mA current loop converter box to establish a connection from the PC's serial port to the programming port (PG) of the PLC (only for 95U, 100U or 115 U).
You also can use a BT 777 bus terminal in combination with a RS-485 converter for larger distances.

The following diagram illustrates the different point-to-point connections (for an AT-PC with RS-232 serial ports).



L1 point-to-point connection.

For a more detailed description of the Sinec-L1 network we refer to the Siemens documentation.

3. Installing The Sinec- L1 Library

3.1. Software installation

First backup your distribution disks and work from your backups.

You must already have installed LabVIEW 5.1 before you can install SinecVIEW.

3.2. AT-PC Windows 3.1 version

 Files included on the disk.

- SINECL1.LLB — Sinec L1 VI library for LabVIEW 5.1

Containing:

  - SinecL1.vi — The Sinec L1 VI
  - Manage PLC data.vi — Utility VI for Sinec L1
  - Example.vi — An example VI for the S5-95U
  - Errorh.vi — An error handler vi

- /S5-95U/SINEC1ST.S5D — Step 5 code for the Example VI
- L1_16.DLL — DLL for Windows 3.1
- /SERIALDR/SERIAL.386 — update serial driver Windows 3.1.1
- /SERIALDR/WG1001.txt — notes update serial driver

Install the VI library by simply copying the file SINECL1.LLB from your floppy drive to a location on your local drive and copy the *.DLL files to your LabVIEW or Windows directory.

Remark: ***Be sure that there are no previous versions of the sinecl1.dll on your local drive.***

IMPORTANT

When you use Windows for Workgroups version 3.11 with a PCI-based machine with a 16550 UART serial port chip (check this with MSD.EXE) you may experience the following problem :

- error 12 (BCC error) when data is transferred from PLC to PC
- no communication at all

This error is due to a bug in the MS Windows serial driver (read the document WG1001.txt for a full explanation) and the use of the buffer of the 16550 UART

If you encounter this problem then :

1. Update the serial.386 driver (read WG1001.txt)

2. Disable the FIFO-buffer of your serial port by editting the Windows\SYSTEM.INI file and change/add the following line in the 386Enh-section

[386Enh]
COMnFIFO=0

where n is the number of the COM port you use for Sinec_L1

if your Sinec_L1 is installed on, say, COM 2 then

[386Enh]
COM2FIFO=0

3.3. Macintosh and PowerMac version

 Files included on the disk.

- SinecL1PPC.VI — Sinec L1 VI for PowerMac
- SinecL168K.VI — Sinec L1 VI for Macintosh (68xxx)
- Manage PLC data.VI — Utility VI for Sinec L1
- Errorh.vi — An error handler vi
- S5-95U/Example.VI — An example VI for the S5-95U
- S5-95U/Sinec1ST.S5D — Step 5 code for the Example VI

Install the VI library by simply copying the files from your floppy drive to a location on your local drive.

3.4. AT-PC Windows 95/NT version

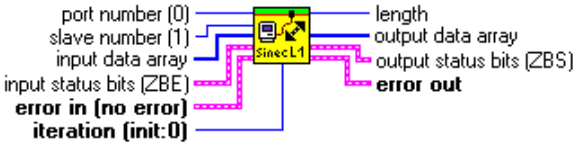
 Files included on the disk.

- L1_32.vi — The Sinec L1 VI
- Manage PLC data.vi — Utility VI for Sinec L1
- Errorh.vi — An error handler vi
- Example.vi — An example VI for the S5-95U
- /S5-95U/SINEC1ST.S5D — Step 5 code for the Example VI
- L1_32.DLL — DLL for Windows 95/NT

Install the VI library by simply copying the files from your floppy drive to a location on your local drive and copy the *.DLL files to your LabVIEW or Windows directory.

4. Labview's VIs Description

4.1. The Sinec L1 VI



SinecL1 handles all communication with the PLCs. It sends the telegram to the specified node in your network via the specified serial port and returns the Slave response to the telegram.

Input wiring

- ***port number (0)*** : the serial port you are communicating with.

  PC

  When you use the serial port VIs under Windows 3.1, the port number parameter can have the following values:

  0: COM1
  1: COM2
  2: COM3
  3: COM4
  4: COM5
  etc.

  If you add extra ports to your computer with a plug-in board, be sure that the configuration is correct (base address and IRQ).

  MAC

  0 =  modem port
  1 =  printer port

  On the Macintosh, port 0 is the modem, using the drivers .ain and .aout. Port 1 is the printer, using the drivers .bin and .bout. To get more ports on a Macintosh, you must install other boards, with the accompanying drivers.

Remark : ***If you use a plug-in board on the Macintosh to get more serial ports, do not forget to add the appropriate values for the 'input driver names' and 'output driver names' into the LabVIEW Global serpOpen.vi***

- ***slave number (1)*** : the node number of the slave that you are addressing.

  1-30 : normal slave
  31 : broadcast (to all slaves)

- ***input data array*** : the data that you would like to send to the PLC. The maximum telegram length should not exceed 64 data bytes. If so, the VI automatically drops the bytes exceeding the maximum length.

- ***input status byte (ZBE)*** : Zustandsbyte Empfang

  The ZBE is always the first byte in the telegram from the master to a Slave.

  Format ZBE :
  bit 0 : none
  bit 1 : slave_aus
  One or more Slaves dropped out.
  1 : A Slave dropped out.
  0 : No drop out.
  bit 2 : bus_run
  Forces bus in run mode
  1 : Bus is in RUN mode
  0 : Bus is in STOP mode
  bit 3 : PG
  Function not defined.
  bit 4 : interrupt
  1 : Request interrupt
  0 : No request.
  bit 5 : QZ
  Quelle/Ziel (Source/Destination)
  1 : Slave receives and transmits data.
  0 : Slave only receives data.
  bit 6 : AG_run
  Force Slave in RUN mode.
  1: AG RUN (with ZBE.7 = 0)
  0 : AG STOP (with ZBE.7=1)
  bit 7 : AG_stop
  Force Slave in STOP mode.

1 : AG STOP (with ZBE.6 = 0)
0 : AG RUN (with ZBE.6=1)

error in (no error) : error in is a cluster that describes the error status before this VI executes. If error in indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its error out cluster. If no error has occurred, then this VI executes normally and sets its own error status in error out. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

- ***status*** : status is TRUE if an error occurred before this VI was called, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.

- ***code*** : code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.

- ***source*** : source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.

- ***iteration (init : 0*)** : if iteration is 0, the VI initializes the serial port with the hardware specifications for SinecL1, and then performs a transceive cycle. If iteration is greater than zero, the SinecL1 VI assumes that the serial port is already configured and just performs a tranceive cycle. You usually wire this input to a loop iteration terminal. Not for windows 95/NT.

Output wiring

- ***length*** : The length of the telegram that the slave returned.

- ***output data array*** : The data bytes that the Slave responded, if no error occurred.

- ***output status byte (ZBS)*** : Zustandsbyte Senden.
  When a Slave receives a telegram, the ZBS is always sent as first byte to the master.

  Format ZBS
  bit 0 : error
  An error occurred in the last data transfer.
  1 : Error.
  0 : No error.
  bit 1 : none1

No function defined.
bit 2 : none2
No function defined.
bit 3 : PG
No function defined.
bit 4 : interrupt
Interrupt request from Slave
bit 5 : QZ
Slave requests 'Querverkehr'. Slave sends data to other Slave.
1: Querverkehr
0 : NO Querverkehr
bit 6 : AG_run
Slave is in RUN mode.
1 : AG RUN (with ZBS.7=0)
0 : AG STOP (with ZBS.7=1)
bit 7 : AG_stop
Slave is in STOP mode.
1 : AG STOP (with ZBS.6=0)
0 : AG RUN (with ZBS.6=1)

- ***error out*** : Error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, error out is the same as error in. Otherwise, error out shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.

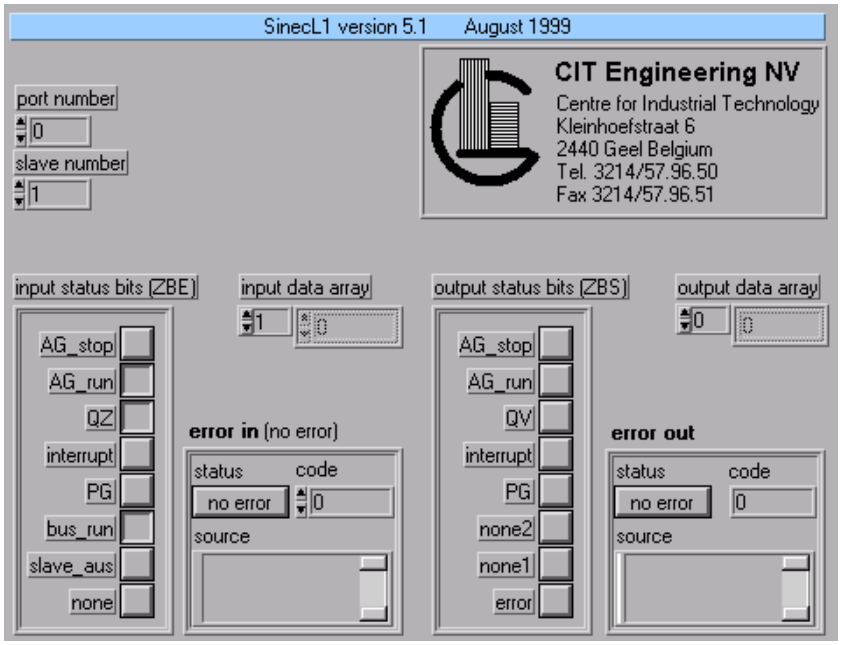
    - ***status*** : status is TRUE if an error occurred before this VI was called, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.

    - ***code*** : code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.

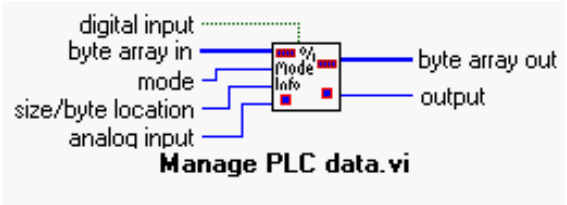
    - ***source*** : source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.

Remark: ***Node numbers go from 1 to 30. The PC is node 0, meaning "the Master". The PLC allows you to send a maximum of 64 data bytes. If the telegram you have to send, contains more than 64 data bytes, only the first 64 will be sent on the Bus. The number of data bytes returned is limited to 64.***

The Sinec L1 front panel.


SinecL1 version 5.1 August 1999
CIT Engineering NV
Centre for Industrial Technology
Kleinhoefstraat 6
2440 Geel Belgium
Tel. 3214/57.96.50
Fax 3214/57.96.51
port number
0
slave number
1
input status bits (ZBE)
AG_stop
AG_run
QZ
interrupt
PG
bus_run
slave_aus
none
input data array
1
0
error in (no error)
status
code
no error
0
source
output status bits (ZBS)
AG_stop
AG_run
QV
interrupt
PG
none2
none1
error
output data array
0
0
error out
status
code
no error
0
source

4.2. The utility VI



A VI used to write and read data (words/bytes/bits) to and from a data array (for the Sinec L1 VI).

There are 7 different operating modes.

Input wiring

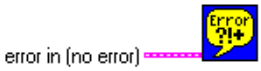
- ***digital input*** : Boolean input when mode=Write bit.
- ***byte array in*** : The input byte array.
- ***mode*** :The mode selection :
  0. Initialize
  1. Read word
  2. Read byte
  3. Read bit
  4. Write word
  5. Write byte
  6. Write bit
- ***size/byte location*** :Dimension size of the array (mode = initialize). Index of the element in the array (all other modes).
- ***analog input*** : Data input when mode=Write word/Write byte. Bit index (of byte) when mode=Read bit/Write bit.

Output wiring

- ***byte array out*** : Output byte array.
- ***output*** : Data output when mode = Read word/Read byte/Read bit.

We refer to the block diagram of the Example VI to learn how to use this VI.

4.3. Specific error for the SinecL1 protocol



ERRORH.VI

This error handler is used primarily to inform the user if an input error exists, to describe the error, and to identify where it occurred. The information for this is derived from the inputs error in, error code, and error source, and from an internal error description table. The table describes all errors that can be created by LabVIEW or its associated I/O operations. The handler has provisions to take alternative actions, such as to cancel or set an error status, and to test for and describe user-defined errors.

Specific errors generated by SinecL1

7011	=	Header information is not correct
7012	=	BCC not correct
7013	=	Got NAK
7016	=	No Data
7123	=	Did not got Slave number from PLC
8234	=	Slave number from PLC not correct

5. Hardware Connection

Important warning : ***When installing or removing interface cables, always power down the PLC, the interface module and the computer. Otherwise, you may damage the computer, the PLC or the interface module.***

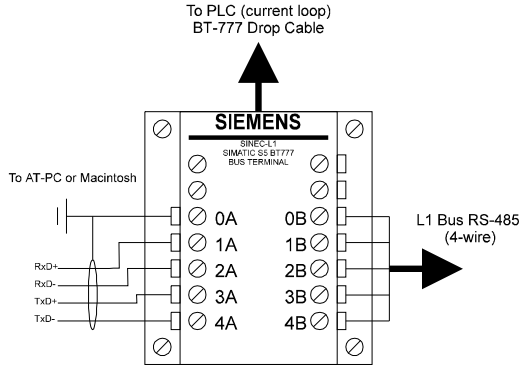
5.1. The BT 777 bus terminal

Normally a BT 777 bus terminal is used to connect your computer to the Sinec L1 network, but in a single PLC application you also can use a Siemens programming cable or a 20 mA current loop converter to establish communication on a point-to-point basis.

The bus terminal uses RS-485 as communication standard. RS-485 is an updated version of RS-422 and is used more and more as the standard interface for different units. It is designed for data buses with up to 32 units and is suitable for multidrop networks with Master/Slaves. It is recommended for distances up to 1.200 m.

Connecting your computer's serial port to a BT 777 bus terminal depends on the computer configuration (AT-PC/Macintosh/PowerMac) you are using.

The figure below shows the connection of the BT 777 bus terminal.



The connection of the BT 777 bus terminal

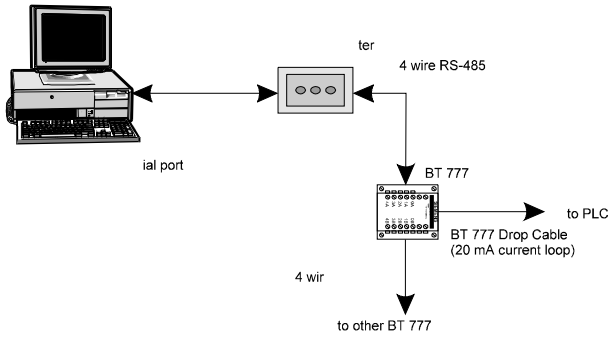
We refer to the Siemens documentation for more information about the bus terminal.

5.2. Connecting an AT-PC to the BT 777.

The most common interface for data communication via the serial port of an AT-PC is the RS-232 standard.

It is possible to directly connect (see picture) the serial port of an AT-PC to the Sinec L1 bus but we don not recommend it because of the danger to overload the PC's communication ICs. When you directly connect from RS-232 to the bus terminal, the distance between PC and bus terminal is limited (<15m).

It is better to use an RS-232 to a 4-wire RS-485 converter or a plug-in communication board that supports 4-wire RS-485.

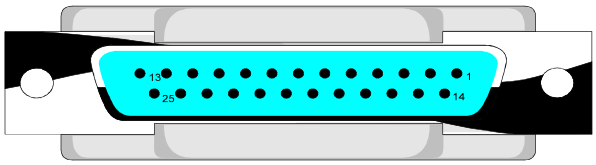


RS-232 to RS-485

Important note : ***Direct connection to the BT 777 can cause overloading of your serial port. We strongly recommend to use an RS-485 converter.***

The picture below shows the pin assignment of a DB-25 connector as most commonly used RS-232 interface connector.

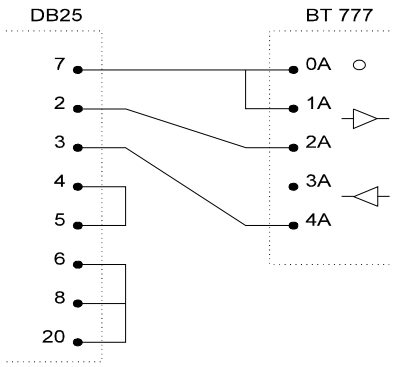
DIN DB-25 connector



1 : Shield	4 : RTS	7 : Signal ground
2 : TxD	5 : CTS	8 : DCD
3 : RxD	6 : DSR	20 : DTR

The pin assignment of a DB-25 connector

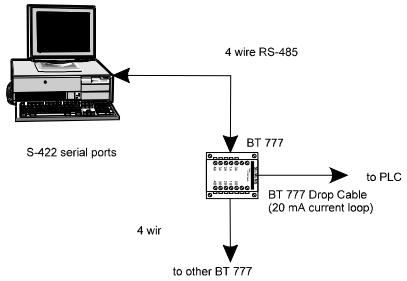
The figure below shows the physical wiring of the computer's serial (RS-232) DB-25 connector and the Sinec L1 BT 777 bus terminal. If you use this connection the distance between PC and the BT 777 is limited (<15m).



connection RS-232 and BT 777

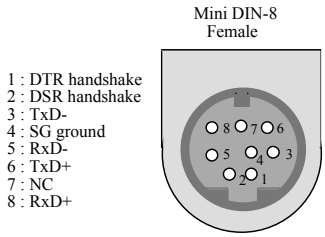
5.3. Connecting a Macintosh PC to the BT 777

The protocol of the normal serial ports of a Macintosh is RS-422. Normally RS-422 and RS-485 are electrical compatible. It is possible to directly connect a serial port of your Macintosh to the Sinec L1 bus.



Macintosh 68K to BT 777

The picture below shows the pin assignment of the mini DIN-8 connector (serial port of a Macintosh). You do not have to connect the DTR and DSR signals to the bus terminal.



It is possible that the communication ICs of your Macintosh cannot supply enough current to the BT 777 bus terminal. So it is possible that you do not get an error free communication with the Sinec L1 VI.

If problems arise, we recommend to use a RS-422/RS-485 driver or repeater to connect your Macintosh to the bus terminal. Be sure that the used repeater can supply sufficient current for the bus terminal.

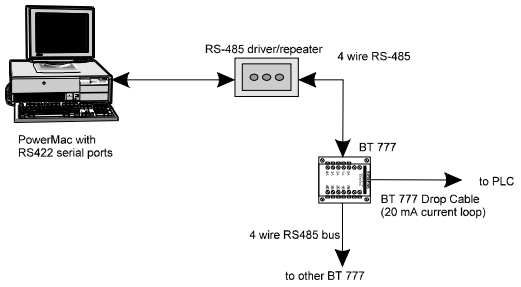
Important note : ***Direct connection to the BT 777 can cause overloading of your serial port. If problems arise we recommend to use an RS-422/RS-485 driver or repeater.***

5.4. Connecting a PowerMac to the BT 777

The protocol of the normal serial ports of a PowerMac is RS-422. Normally RS-422 and RS-485 are electrical compatible, but with our test configuration (a 8100 PowerMac) it was not possible to get an error free communication. The communication IC of the PowerMac could not supply sufficient current to the bus terminal.

We recommend to use a RS-422/RS-485 driver or repeater to connect your PowerMac to the bus terminal. Be sure that the used repeater can supply sufficient current for the bus terminal.

If you do not get an error free communication with the Sinec L1 VI it is likely that the used converter is not suitable.



PowerMac to BT 777

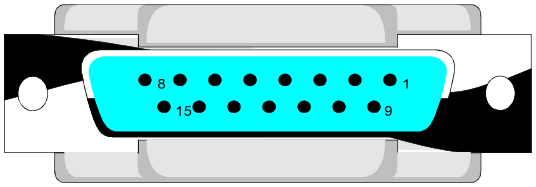
Important note : ***Direct connection to the BT 777 can cause overloading of your serial port. If problems arise we recommend to use an RS-422/RS-485 driver or repeater.***

5.5. Connecting a PLC to the BT 777

The BT 777 houses a drop cable with a 15-pin connector that connects the PLC to the network bus. This connector may be plugged into the PG port of Siemens PLC (only 95U, 100U and 115U) or in the lowest port of the CP 530 communication card (115U or higher).

This drop cable uses a 20 mA current loop interface to communicate with the PLC.

DIN-15 connector/female



1 : mass external	5 : mass internal	9 : RxD+	14 : Vpg + 5.2V
2 : RxD-	6 : TxD+	11 : T/20mA	15 : mass internal
3 : Vpg + 5.2V	7 : TxD-	12 : mass internal	
4&10 : -	8 : mass external	13 : R/20mA	

6. Example For The S5-95U

The Sinec L1 pack comes with a well-documented example VI. The example VI is located in the Sinec L1 library (Example.vi) together with a few utility VIs.

The VI shows the connection between a Siemens SIMATIC S5 95U PLC and a PC hosted LabVIEW. The PLC program is stored in a separate directory on your disk by the name sinec1st.s5d. Note that this file has a Siemens Step 5 format that can only be read with a Siemens DOS PG.

6.1. Configuring your PLC for the Example VI

The correct parameters must be entered into DB1 in order to allow the S5-95U to work correctly with the Sinec-L1 protocol.

The configuration of the S95U for the example must appear as shown below.

```
DB1
0:   KC ='DB1 OBA: AI 8 ; OBI:    ';
12:  KC ='   ; OBC: CAP 100   CBP ';
24:  KC ='100    ; SL1: SLN 1  SF ';
36:  KC ='DB2  DW0   EF DB3  DW0  ';
48:  KC ='  KBE MB100      KBS MB1';
60:  KC ='01      PGN 1  ;  SDP: N';
72:  KC ='T 128 PBUS N ; TFB: OB13';
84:  KC =' 60000  ; #CLP: STW MW10';
96:  KC ='2        CLK DB5  DW0   ';
108:KC ='  SET 3  01.10.91 12:00:';
120:KC ='00    OHS 000000:00:00  ';
132:KC ='  TIS 3 01.10. 12:00:00 ';
144:KC ='   SIP Y SAV Y  CF 00   ';
156:KC =';#    END   ';
```

- SLN (slave node) = 1
- SF (send mailbox) = DB2 starting at DW1
- EF (receive mailbox) = DB3 starting at DW1
- KBE (receive coordinate byte) = MB100
- KBS (send coordinate byte) = MB101

Remark: ***While editing DB1 be careful not to affect the spacing within the datablock***.

For further information concerning the configuration of the S95U refer to the Siemens manual.

The remaining test code is shown below:

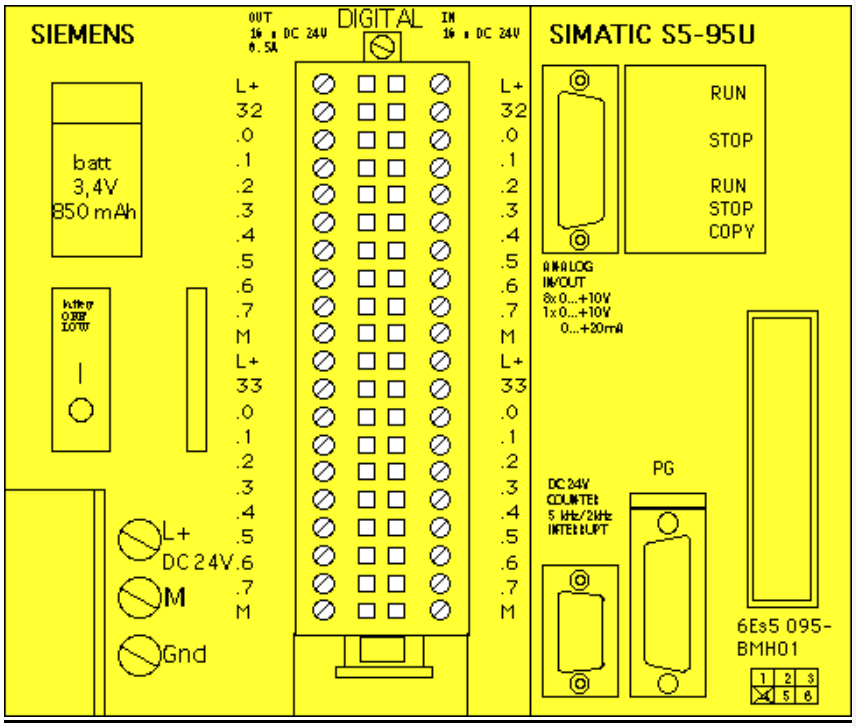
```
OB1
000   :SPA FB2
NAME  :ZENDEN          //OB1 runs every cycle
004   :BE
```

```
FB2
NAME  :ZENDEN
      :A DB3        // DB3 is configured as receive mailbox.
      :U M 100.7    // last message done ?
      :SPB =M0
      :L DR0        // load source of master
      :L KF+        // load address of master (0)
      :><F          // not equal?
      :SPB =M1
      :L DW1
      :T AW32       // first 16 outputs of PLC

      :UN M 100.7
      :S M 100.7    // enable receive bit


M0    :A DB2        // DB2 is configured as send mailbox.
      :LKF+2        // this value indicates how many bytes the PLC has
                    // to send to the PC. (starting with DW 1 (DB2))
      :T DL0        // transfer to mailbox length header
      :LKF+0        // load destination address -> master
      :T DR0        // transfer to destination header
      :L EW32       // first 16 inputs of PLC
      :T DW1

      :S M 101.7    // enable send bit

      :BE
```

Remark: ***Do not forget to declare DB2 and DB3. Fill these DBs with zeros!***

6.2. LabVIEW_Example.Vi

On the figure below you can see how easy it can be to create a clear front panel.
The picture was painted and then pasted as an image in this example. In this example we send two bytes to the PLC (AB32 AB33) and receive two bytes (EB32 EB33) from the PLC.
Do not forget to chose the right port and Slave number before running the VI.



If you look at the example you will see 16 LED's on the digital input of the PLC and 16 LED's on the digital output of the PLC. Just click the output LED's and the corresponding led on the PLC will light up (relay activated). Activate the digital inputs on the PLC and the corresponding LED's on the frontpanel will light up.

Remark: ***Do not forget the power supply (L+ and M) of the digital inputs and outputs.!***