



Technical support:
+49 (0)7223 / 9493-0

Attention!
Product discontinuation
due to EC RoHS directive
More info: www.addi-data.com



Technical description

ADDICOM PA 755

4-port serial interface

8th edition 03/1996

Copyright

All rights reserved. This manual is intended for the manager and its personnel.
No part of this publication may be reproduced or transmitted by any means.
Offences can have penal consequences.

Guarantee and responsibility

Basically are effective our "general terms of delivery and payment". The manager receives them at the latest with the invoice. Claims for guarantee and responsibility in case of injuries and material damages are excluded, if they are due to one or some of the following causes:

- if the board has not been used for the intended purpose
- improper installation, operation and maintenance of the board
- if the board has been operated with defective safety devices or with not appropriate or non-functioning safety equipment
- nonobservance of the instructions concerning: transport, storage, inserting the board, use, limit values, maintenance, device drivers
- altering the board at the user's own initiative
- altering the source files at the user's own initiative
- not checking properly the parts which are subject to wear
- disasters caused by the intrusion of foreign bodies and by influence beyond the user's control.

Licence for ADDI-DATA software products

Read carefully this licence before using the standard software. The right for using this software is given to the customer, if he/she agrees to the conditions of this licence.

- this software can only be used for configuring ADDI-DATA boards.
- copying the software is forbidden (except for archiving/ saving data and for replacing defective data media).
- deassembling, decompiling, decoding and reverse engineering of the software are forbidden.
- this licence and the software can be transferred to a third party, so far as this party has purchased a board, declares to agree to all the clauses of this licence contract and the preceding owner has not kept copies of the software.

Trademarks

Borland C and Turbo Pascal are registered trademarks of Borland International, INC.

Burr-Brown is a registered trademark of Burr-Brown Corporation

Intel is a registered trademark of Intel Corporation

CompactPCI is a registered trademark of the PCI Industrial Computer Manufacturer Group (PICMG)

Microsoft, MS-DOS, Visual Basic and Windows are registered trademarks of Microsoft Corporation

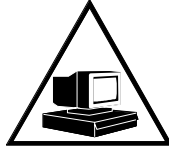
The original version of this manual is in German. You can obtain it on request.

WARNING

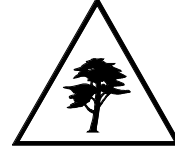
In case of wrong use and if the board is not used for the purpose it is intended for:



people may be injured



the board, PC and peripheral may be destroyed



the environment may be polluted

★★★ Protect yourself, the others and the environment★★★

- **Do read the safety leaflet!**

If this leaflet is not with the manual, please contact us and ask for it.

- **Observe the instructions of the manual!**

Make sure that you do not forget or skip any step. We are not liable for damages resulting from a wrong use of the board.

- **Symbols used**



WARNING!

It designates a possibly dangerous situation.

If the instructions are ignored **the board, PC and/or peripheral may be damaged.**



IMPORTANT!

designates hints and other useful information.

- **Any question?**

Our technical support is at your disposal

1. INTENDED PURPOSE OF THE BOARD	1
2. BOARD FUNCTION	2
2.1 Introduction	2
2.2 Addressing.....	2
2.3 Interrupt.....	2
2.4 Protection of the inputs and outputs.....	3
3. SELECTING THE OPERATING MODE	3
3.1 Jumper position	3
3.2 Operating mode RS 232	3
3.2.1 Electric specifications	3
3.2.2 Communication with modem control signals	4
3.2.3 Jumper configuration in RS 232	5
3.2.4 Communication without modem control signals	5
3.3 Operating mode RS 422	6
3.3.1 Electric specifications	6
3.3.2 RS 422 - Connection principle for a point to point connection	6
3.3.3 Jumper configuration in RS 422	6
3.3.4 Enabling the RS 422 sender	7
3.3.5 Terminal resistor in RS 422	8
3.4 Operating mode RS 485	9
3.4.1 Electric specifications	9
3.4.2 RS 485 - Principle of a point to point connection.....	10
3.4.3 Jumper configuration in RS 485 mode	10
3.4.4 Enabling the RS 485 sender.....	10
3.4.5 Enabling the RS 485 receiver	11
3.4.6 Terminal resistor in RS 485	12
3.5 Operating mode current loop.....	13
3.5.1 Electric specifications	13
3.5.2 Jumper basic configuration for CURRENT LOOP	14
3.5.3 Inverting logic level for emission and reception	14
3.5.4 Sender and receiver - active or passive.....	16
3.5.5 Supply of the constant current sources and of the DC/DC converters in option G.....	17
3.5.5.1 <i>Selecting the supply voltage for constant current sources</i>	17
3.5.5.2 <i>Setting a galvanic isolation between the internal and external supply voltage (current loop)</i>	18
3.5.6 Connection examples in mode current loop	18
3.5.6.1 <i>Sender active, receiver active (Port 1)</i>	19
3.5.6.2 <i>Sender passive, receiver passive (Port 1)</i>	19

4. PROTECTION OF THE INPUTS AND OUTPUTS	20
5. ADDRESSING THE BOARD	21
5.1. Adjusting the base address over DIP switch	21
5.1.1 Decoding address bits A3 and A4.....	22
5.2 Addressing at COM1, COM2, COM3, COM4 (over jumper)	23
5.3 Simultaneous addressing over DIP switch and jumper	25
5.4 I/O Map.....	28
6. INTERRUPT HANDLING	29
6.1 Single interrupts.....	29
6.1.1 Selecting single interrupts.....	29
6.1.2 Selecting the IRQ line to the PC bus for single interrupts.....	30
6.1.2.1 <i>Example 1</i>	30
6.1.2.2 <i>Example 2</i>	31
6.1.2.3 <i>Example 3</i>	32
6.2 Common interrupt.....	33
6.2.1 Interrupt routine for managing several boards with the same interrupt request line	34
6.2.2 Jumper configuration for selecting the common interrupt.....	35
6.3 Principle of the different interrupt request releases	36
7. AN EXAMPLE OF BOARD CONFIGURATION	37
8. MECHANICAL SET-UP	38
9. DRIVER	39
9.1 Introduction	39
9.2 Floppy disk	39
9.3 Board PA 755.....	39
9.4 File CONFIG.SYS	40
9.4.1 Configuring the ports in common interrupt	40
9.4.2 Configuring the ports in single interrupt.....	40
9.4.3 Configuring the ports in single and common interrupt.....	40
9.4.4 Default setting for the serial ports.....	41
9.5 Using DOS functions	41
9.6 Programming with DRV755.SYS	42
9.6.1 Pascal	42

9.7 IOCTL functions	42
9.7.1 Example of a direct call in C	43
9.7.2 Example of an indirect call in Pascal.....	44
9.7.3 Setting a new port configuration.....	45
9.7.4 Reading the configuration of a serial port	48
9.8 Timeout	49
9.9 Using XMODE.....	50
9.10 Operating modes	50
9.10.1 Installing several drivers.....	50
10. APPENDIX.....	A
10.1 Block diagram	A
10.2 Component scheme.....	B
10.3 37 pole pin connector - signals.....	D
10.3 Connection cable ST077, 37 pole to 4 x 25 pole.....	E
10.4 Connection cable ST078, 37 pole to 4 x 9 pole	E

1. INTENDED PURPOSE OF THE BOARD

The board **PA 755** allows to exchange data through 4 serial ports between an industrial process and a personal computer (PC). It is to be used in a free PC ISA slot. The PC is to comply with the EU directive 89/336/EEC and the specifications for EMC protection. Products complying with these specifications bear the CE mark.

Serial data is exchanged with external communication devices through the 37 pole MIN-D connector of the board **PA 755** in the chosen transmission mode (RS 232, RS 422, RS 485 or 20 mA current loop).

Communication occurs over a shielded cable whose shielding is earthed on the both sides via metallized plastic caps. The connection cable is to comply with the following specifications:

- metallized plastic cap
- shielded cable
- cable shield folded back and firmly screwed to the connector housing.

The use of the board in a PC could change the PC features regarding to noise emission and immunity. Increased noise emission or decreased noise immunity could result in the system not being conform anymore. Check the PC's and cable's shielding capacity prior to putting the device into operation.

The use of the board according to its intended purpose includes observing all advices given in this manual and the safety leaflet. Uses beyond these specifications are not allowed.

The manufacturer is not liable for any damages which would result from the non-observance of this clause.

For a quick and safe installation of this board please read absolutely following chapters :

5. Adressing the board

2. Board function

2.1 Introduction

The board **PA 755** is an interface with four asynchronous serial ports. Via jumper, you can configure individually for each port the following operating modes:

- RS 232
- RS 422
- RS 485
- Current loop

The board is available in the following versions:

- **PA755VRC** 4 ports ; RS 232, RS 422/RS 485, current loop
(without DC/DC converter)
- **PA755VR** 4 ports, RS 232, RS 422/RS 485
- **PA755VC** 4 ports, RS 232, current loop
- **Option G** Each port operated in current loop can be separated galvanically over a DC/DC converter
(only versions **PA755VRC** and **PA755VC**). See chapter 3.5.

2.2 Addressing

This board is addressable:

- via DIP switch:
a base address is selected (see chapter 5.1)
- over jumper:
a standard address i.e. COM1, COM2, COM3 or COM4 is selected (see chapter 5.2),
- or by combining these two addressing modes (see chapter 5.3).

2.3 Interrupt

Interrupts can be generated in different ways. Are available:

- the common interrupt:
An interrupt for the four ports (see chapter 6.2)
- four single interrupts:
An interrupt for each port (see chapter 6.1)
- Common and single interrupts combined:
Ex.: two ports are managed over common interrupt and the two ports remaining over single interrupts.

2.4 Protection of the inputs and outputs

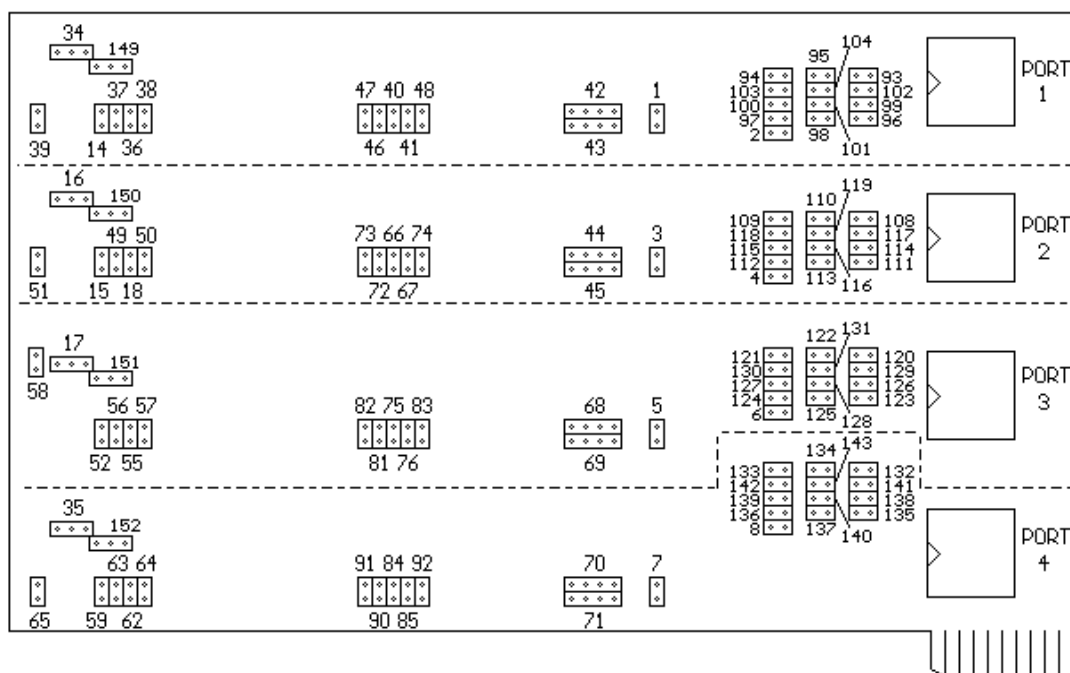
All inputs and outputs are protected against overvoltage through TRANSIL diodes. See chapter 4.

3. Selecting the operating mode

3.1 Jumper position

You will find in the figure below the numbers of jumpers involved in the selection of the operating mode. On the component scheme they are designated by BR (for bridge = jumper).

Attention: Most of the jumper settings represented in this manual are for Port 1. They have to be considered as models for the three other ports.



3.2 Operating mode RS 232

3.2.1 Electric specifications

Output stage

Output voltage: min. +/-9V
Short-circuit capacity:..... +/- 45mA

Receiver

Maximum input voltage:..... limited to +/-26V over Transil
..... diodes

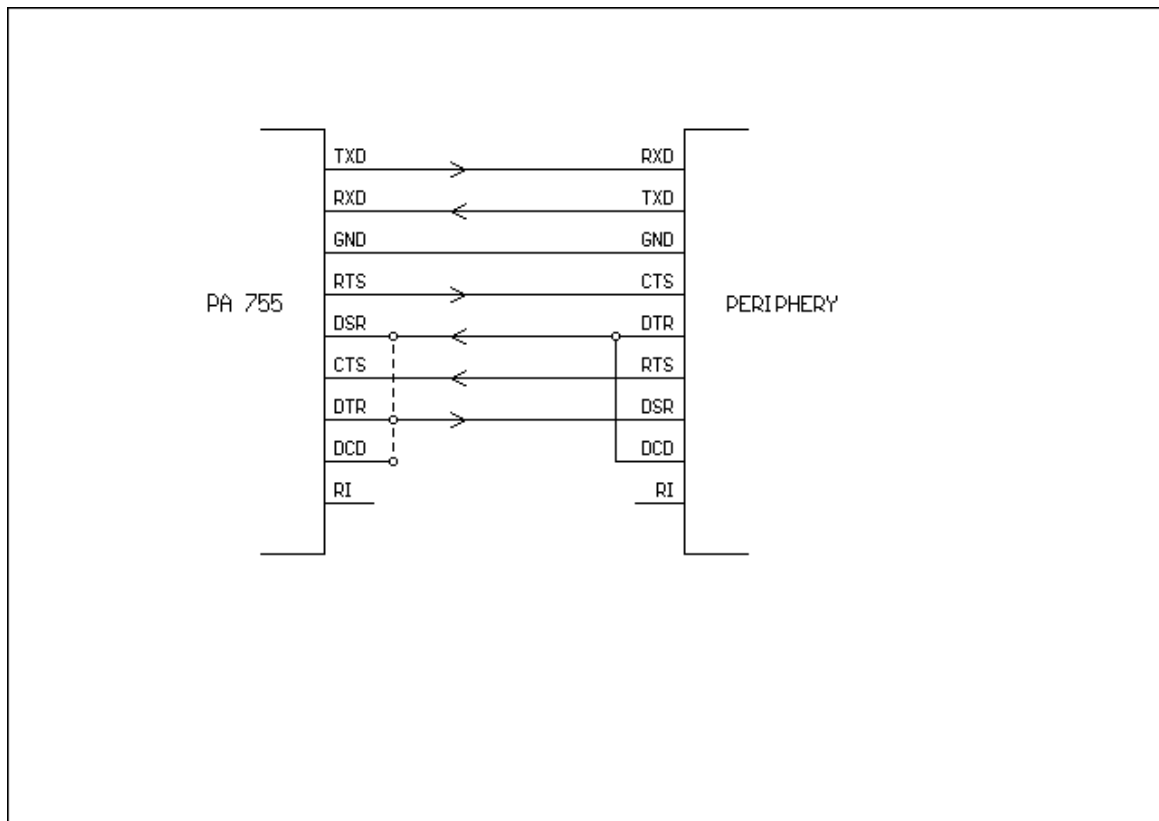
3.2.2 Communication with modem control signals

The modem control signals RTS, CTS, DSR, DTR, DCD, RI are available for each port at the front connector. You can thus connect peripherals managing these modem control signals.

If the periphery does not hand over all the signals shown in the connection principle 3a, the inputs/outputs not used by board **PA 755** can be cabled together, according to their function, directly at the front connector.

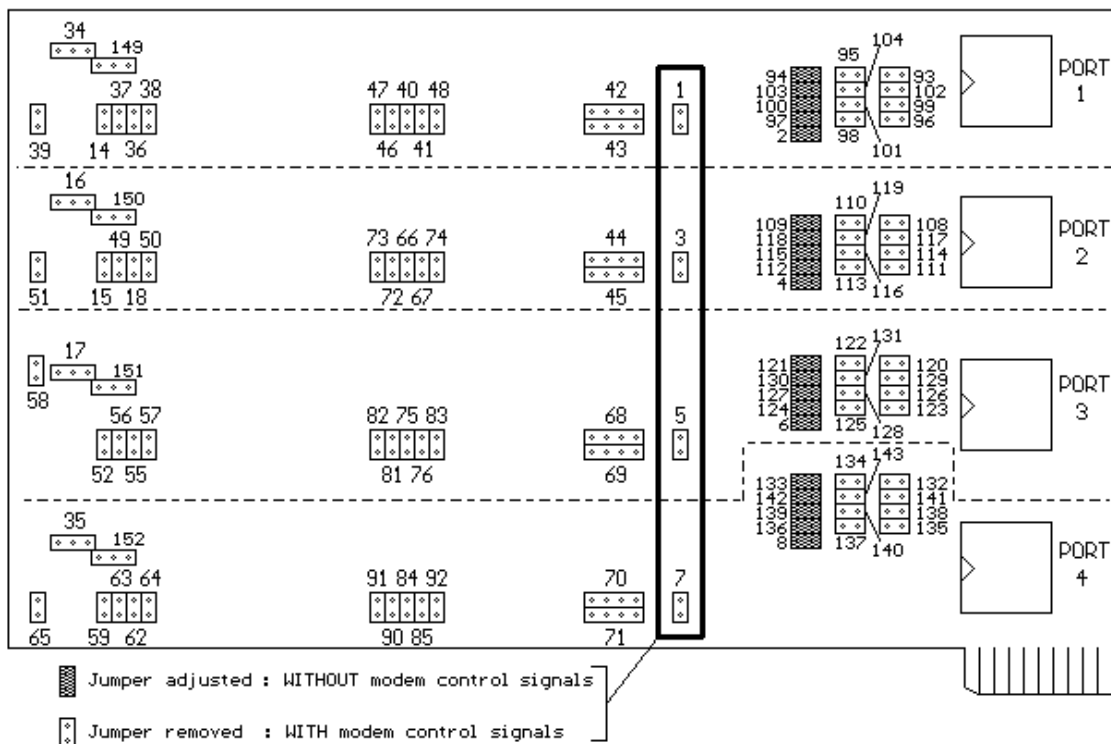
Ex.: if the periphery hands over no DTR signal, the DTR signal of the **PA 755** can be used for controlling its own DSR signal. In this case the signals DTR and DSR have to be wired together at the front connector. The signals DTR, DSR, DCD are thus cabled together directly at the connector.

Fig. 3a - CONNECTION PRINCIPLE



3.2.3 Jumper configuration in RS 232

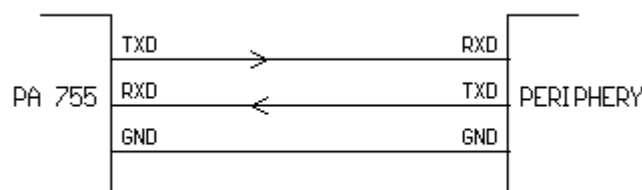
Fig. 3b - JUMPER CONFIGURATION IN RS 232 WITH AND WITHOUT MODEM CONTROL SIGNALS



3.2.4 Communication without modem control signals

This adjustment is valid if only the signals TXD, RXD and GND are used as a three-line connection. See jumper configuration on figure 3b.

Fig. 3c - CONNECTION PRINCIPLE



The RS232 received signal RXD is led outside for the receiver over jumpers BR2, BR4, BR6, BR8 (BR2 for Port 1). The corresponding jumper must be removed when the board is operated in RS 422, RS 485 and current loop.

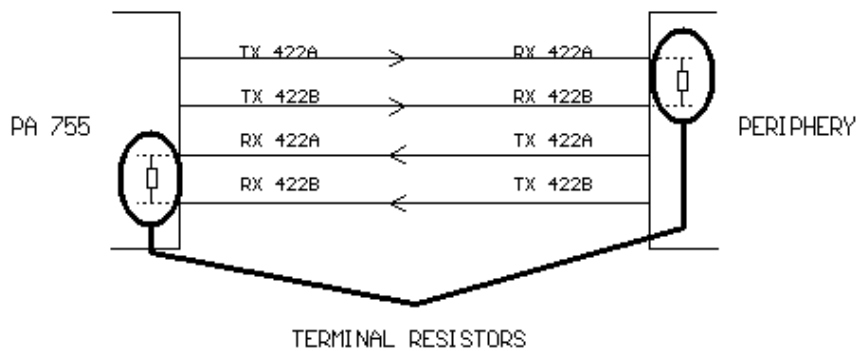
3.3 Operating mode RS 422

3.3.1 Electric specifications

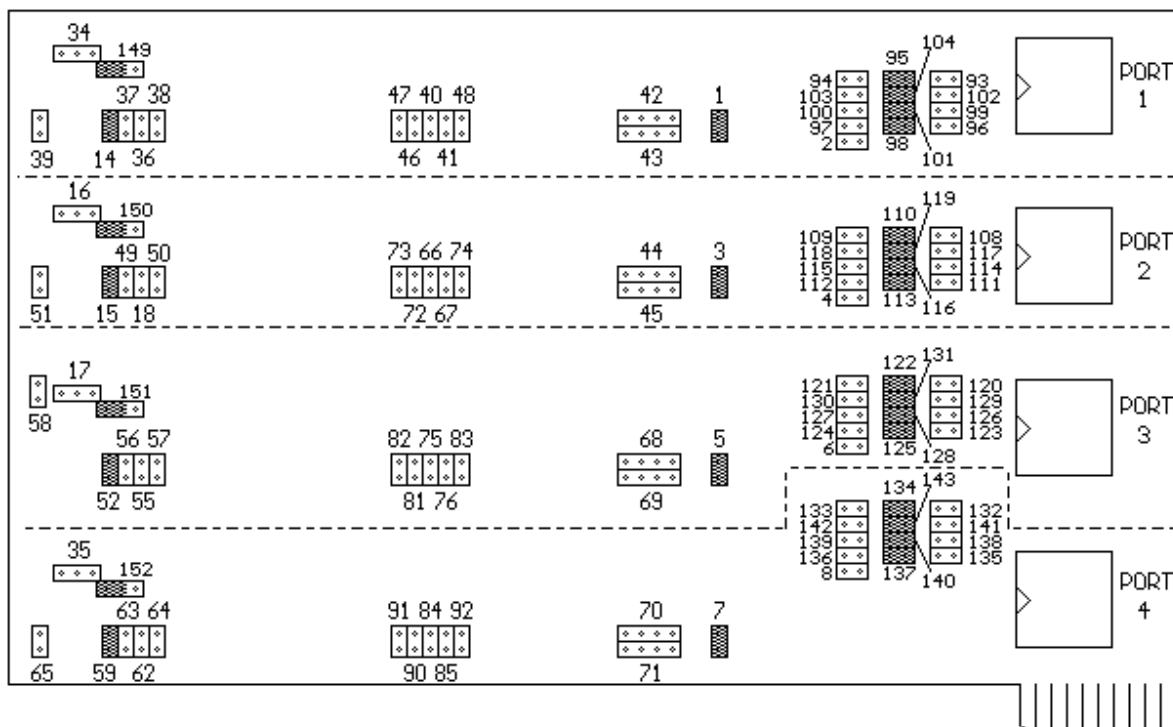
Sender	Signal without load:.....max. +/-5V Signal with load:.....max. +/-2V Load of output stage :.....>= 100 ohm
Receiver	Input voltage:.....max. +/-7V Sensitivity:.....min. +/-200mV

3.3.2 RS 422 - Connection principle for a point to point connection

Terminal resistors: 100 ohm



3.3.3 Jumper configuration in RS 422



3.3.4 Enabling the RS 422 sender

Before transmission in RS 422, the sender must be enabled by selecting one of the three following signals:

1. RTS

This signal originates from the USART and is commanded by software over bit **D1** of the **Modem Control Register** (see I/O map in chapter 5.4).

- "0" blocks the sender
- "1" enables the sender

2. DIR

This signal originates from the USART and is commanded by software over Bit **D0** of the **RS 422/485 Enable Register** (see I/O Map in chapter 5.4).

- "0" blocks the sender
- "1" enables the sender

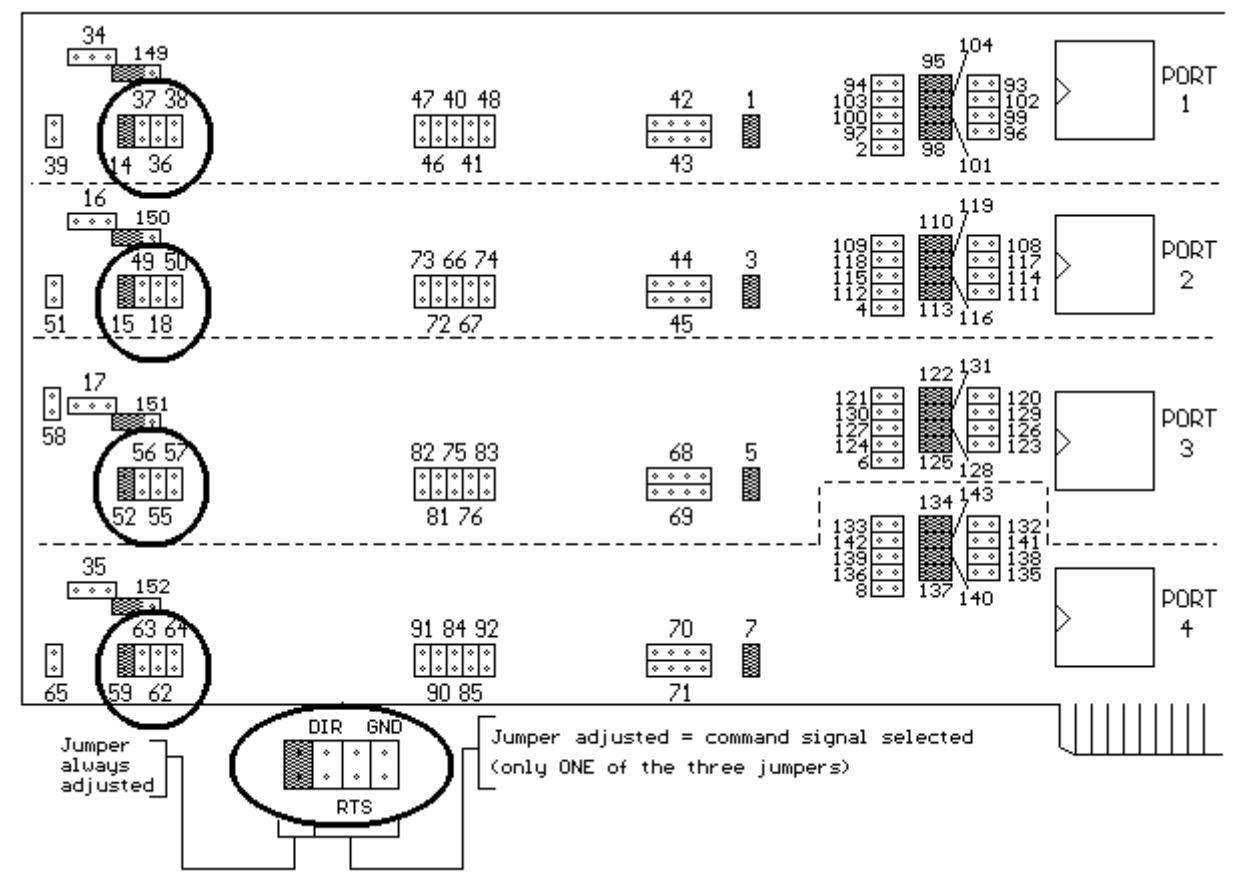
3. GND

The sender is always active. It can't be deactivated by software.

You select the wished signal over jumper (fig. 3d).

USART = Universal Serial Asynchronous Receiver/Transmitter

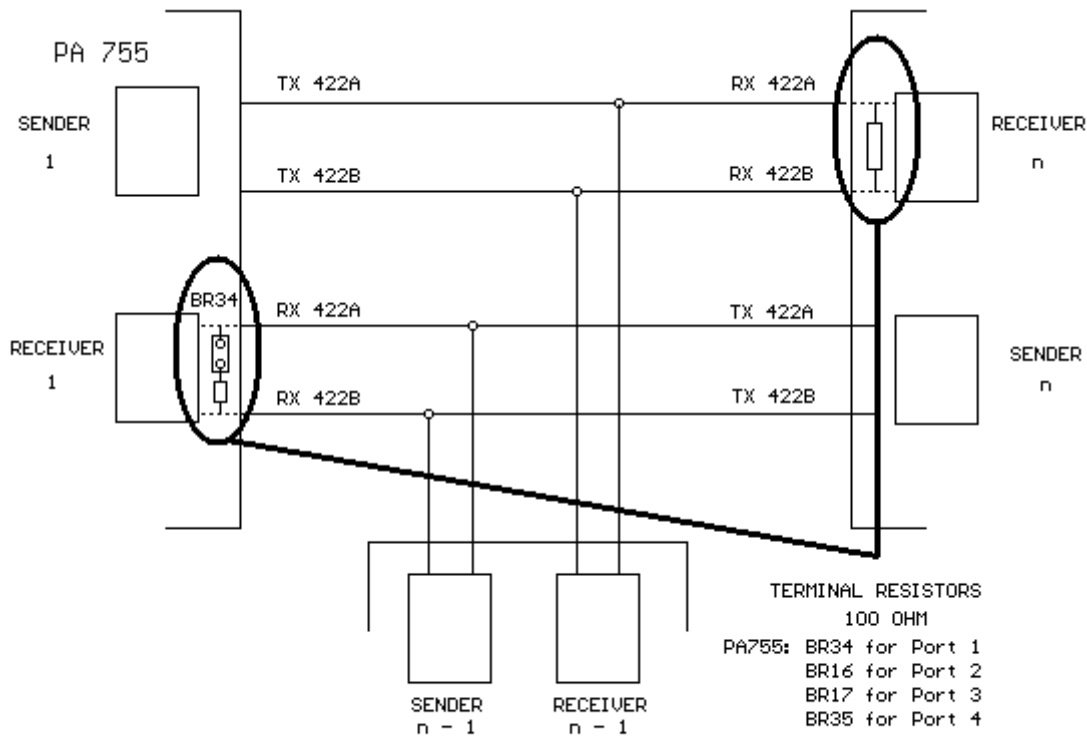
Fig. 3d - ENABLING THE RS 422 SENDER WITH THE SIGNALS RTS, DIR AND GND



3.3.5 Terminal resistor in RS 422

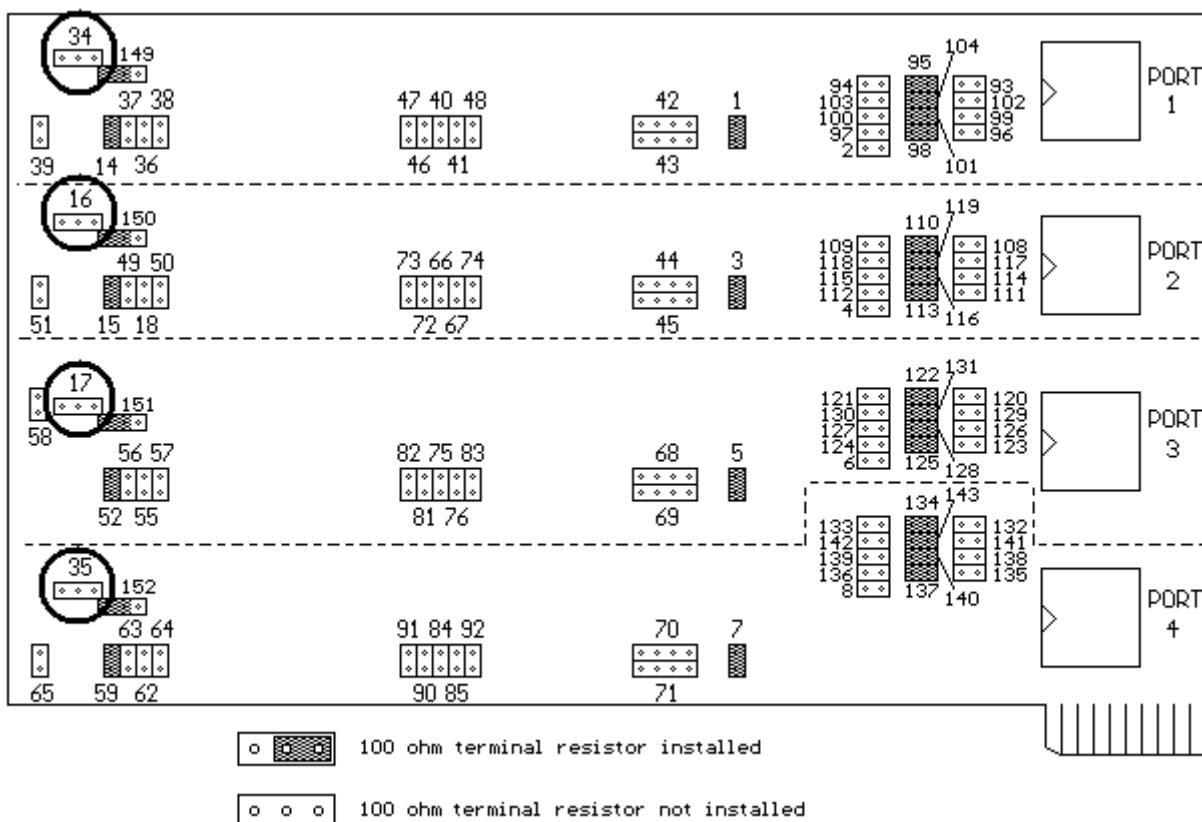
When the board is operated in RS 422, make sure to equip the RS 422 receiver, located at the end of the line, with a **100 ohm** terminal resistor. Board **PA 755** is therefore provided with 4 terminal resistors (one for each port), which can be installed via jumper on the corresponding receive lines of the RS 422 receiver.

Fig. 3e - CONNECTION PRINCIPLE WITH SEVERAL PARTICIPANTS



The terminal resistors are installed on the RS 422 receive line over jumpers BR34, BR16, BR17 and BR35. An adjusted jumper indicates that a terminal resistor is installed on the receive line of the RS 422 receiver. The figure below shows to what port is allocated each of these jumpers.

Fig. 3f - JUMPER CONFIGURATION FOR OPERATION WITH TERMINAL RESISTORS ON THE RECEIVE LINE



3.4 Operating mode RS 485

In contrast to the RS 422 connection requiring a four-wire line, only a two-wire line is necessary in RS 485 mode.

Sender and receiver of the board as well as periphery use the same two-wire line. Therefore make sure that bus conflicts do not happen while sending. Bus conflicts are avoided since the board is able to command the RS 485 sender in different ways.

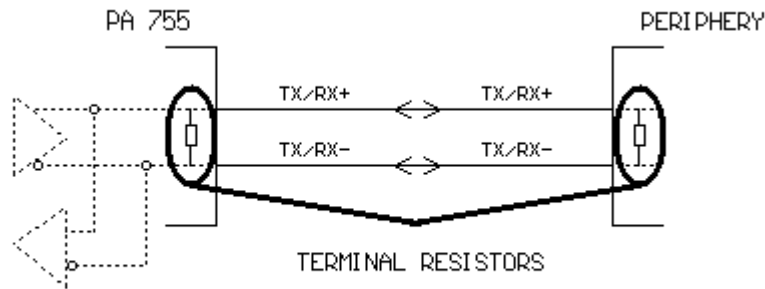
For excluding conflicts safely on the bus, certain RS 485 communication protocols must be observed. These protocols are not delivered with the board.

3.4.1 Electric specifications

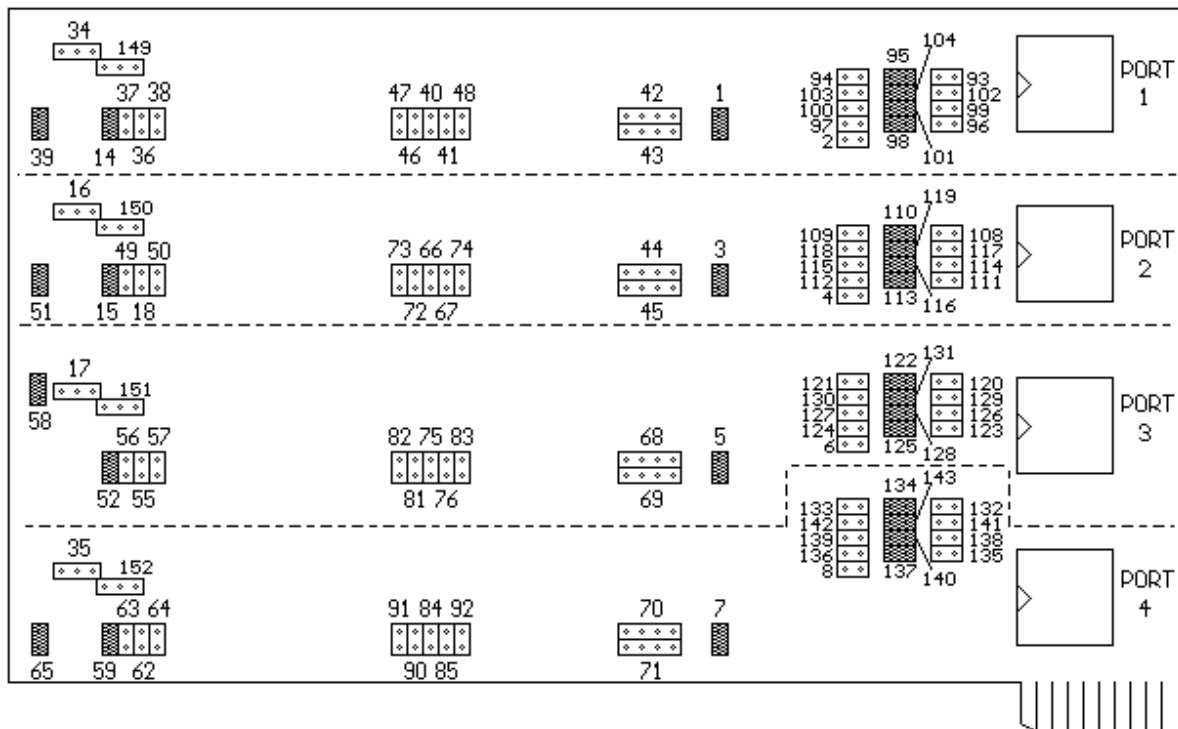
Sender	Signal without load: max. +/-5V Signal with load: min. +/-1,5V Load of output stage :..... >= 120 ohm
Receiver	Input voltage: max. 7-12V Sensitivity: min. +/-200mV

3.4.2 RS 485 - Principle of a point to point connection

Terminal resistors: 120 ohm



3.4.3 Jumper configuration in RS 485 mode



3.4.4 Enabling the RS 485 sender

Before transmission in RS 485, the sender must be enabled by selecting one of the three following signals:

1. RTS

This signal originates from the USART and is commanded by software over bit **D1** of the **Modem Control Register** (see I/O map in chapter 5.4).

- "0" blocks the sender
- "1" enables the sender

2. DIR

This signal originates from the USART and is commanded by software over bit **D0** of the **RS 422/485 Enable Register** (see I/O map in chapter 5.4).

- "0" blocks the sender
- "1" enables the sender

USART = Universal Serial Asynchronous Receiver/Transmitter

3. GND

The sender is always active. It can't be deactivated by software.

You select the wished signal over jumper (fig. 3d).

3.4.5 Enabling the RS 485 receiver

From rev. D it is possible over jumpers BR149, BR150, BR151, BR152:

- either to disconnect permanently the RS 485 receiver,
- or to release alternately the sender or the receiver over the control bits RTS or DIR (refer to chapter 3.4.4). Thus if you release the sender, the receiver is blocked. If the sender is blocked, the receiver is released.

Fig. 3g - JUMPER ADJUSTMENT

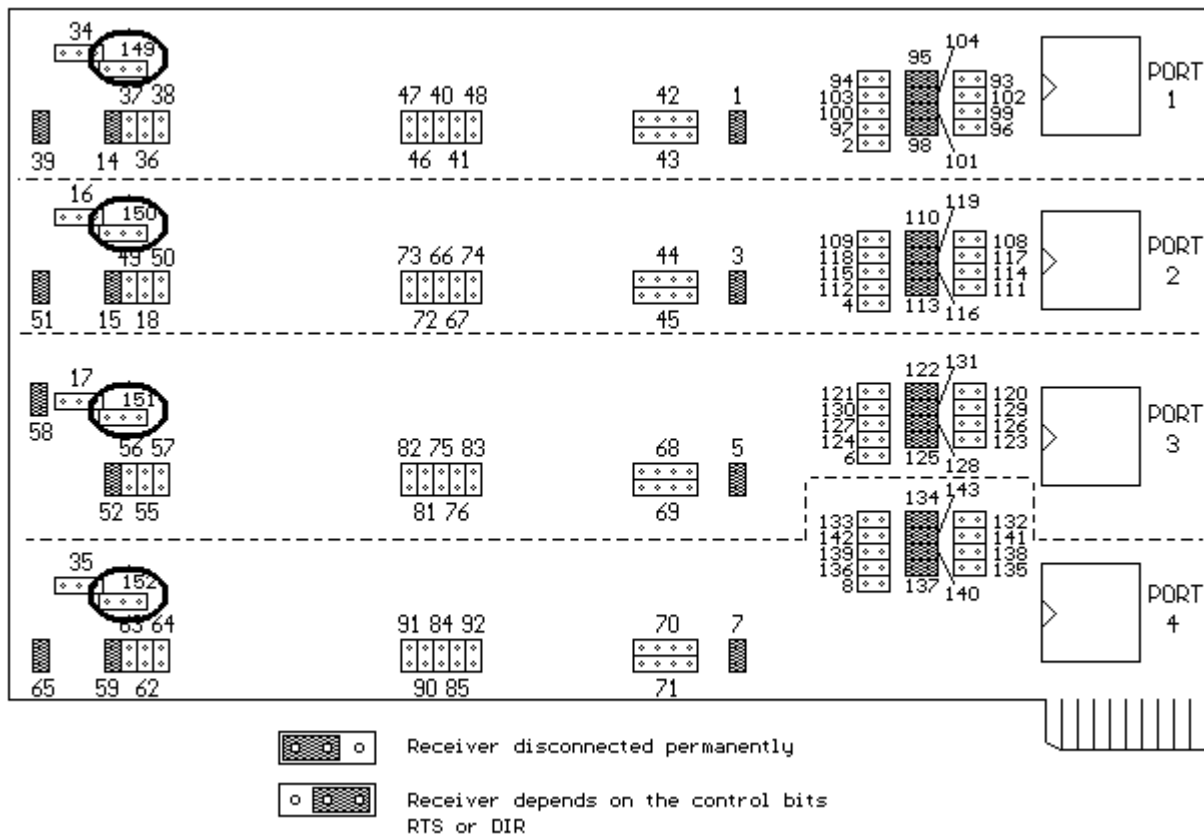
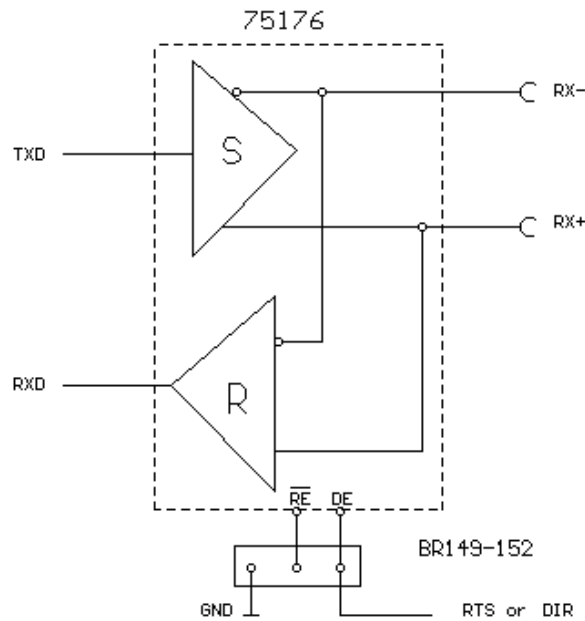


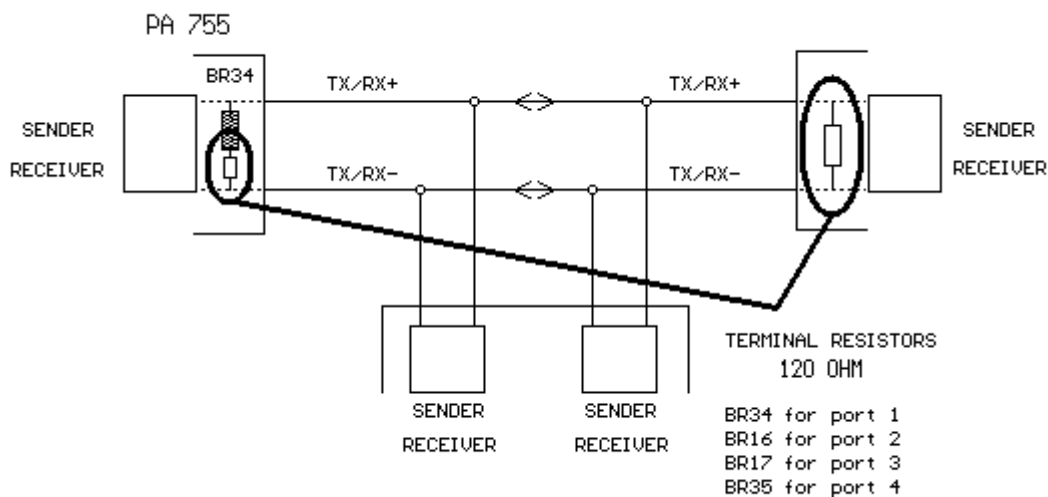
Fig. 3h - PRINCIPLE



3.4.6 Terminal resistor in RS 485

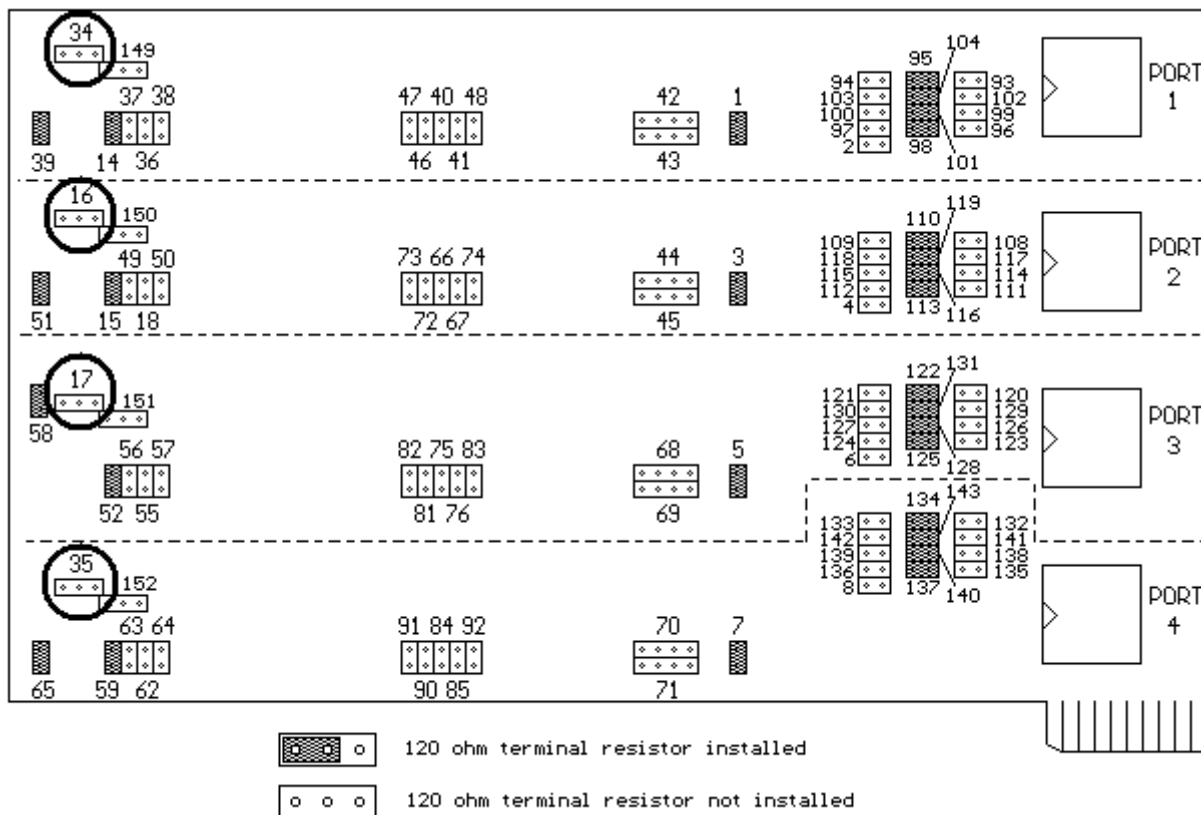
When the board is operated in RS 485, make sure to equip the RS 485 receivers, located at the end of line, with a **120 ohm** terminal resistor. Board **PA 755** is therefore provided with 4 terminal resistors (one for each port), which can be installed via jumper on the corresponding receive line of the RS 485 receiver.

Fig. 3i - CONNECTION PRINCIPLE WITH SEVERAL PARTICIPANTS



The terminal resistors are installed on the RS 485 receive line over jumpers BR34, BR16, BR17 and BR35. An adjusted jumper indicates that a terminal resistor is installed on the receive line of the RS 485 receiver. The figure below shows to what port is allocated each of these jumpers.

Fig. 3j - JUMPER CONFIGURATION FOR OPERATION WITH TERMINAL RESISTORS ON THE RECEIVE LINE



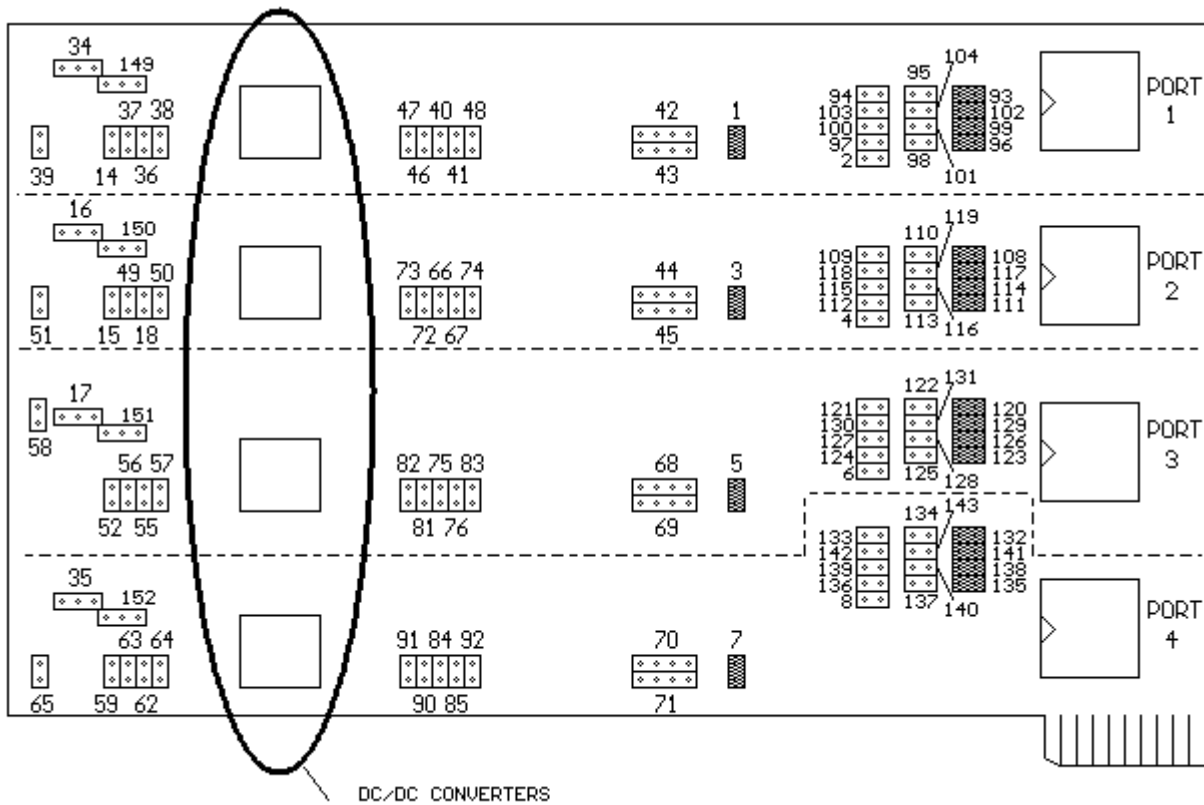
3.5 Operating mode current loop

Board **PA 755** can be used in different ways as a constant current interface. The board is able to send and receive actively and passively. The board's 4 ports are provided each with two 20mA constant current sources for sending and receiving. They are galvanically isolated from the system through optical couplers, and in option G over DC/DC converter. The maximum isolation voltage = 500 V DC. You will find the different configurations in the figures below.

3.5.1 Electric specifications

Sender	Constant current: 20mA Load: < 350 ohm Input protection: +/- 26V
Receiver	Constant current: 20mA Load: < 350 ohm Input protection: +/- 26V

3.5.2 Jumper basic configuration for CURRENT LOOP



The hatched jumpers constitute the basic configuration for current loop. In the figure above, all the 4 ports are in current loop. You can configure each port individually. For setting the passive or the active mode, the basic configuration must be adjusted.

3.5.3 Inverting logic level for emission and reception

It allows to adapt the board to the most various types of level transmissions.

With jumpers BR40 and BR41 (Port 1) you decide whether the transistor in the output stage is enabled or disabled.

BR40

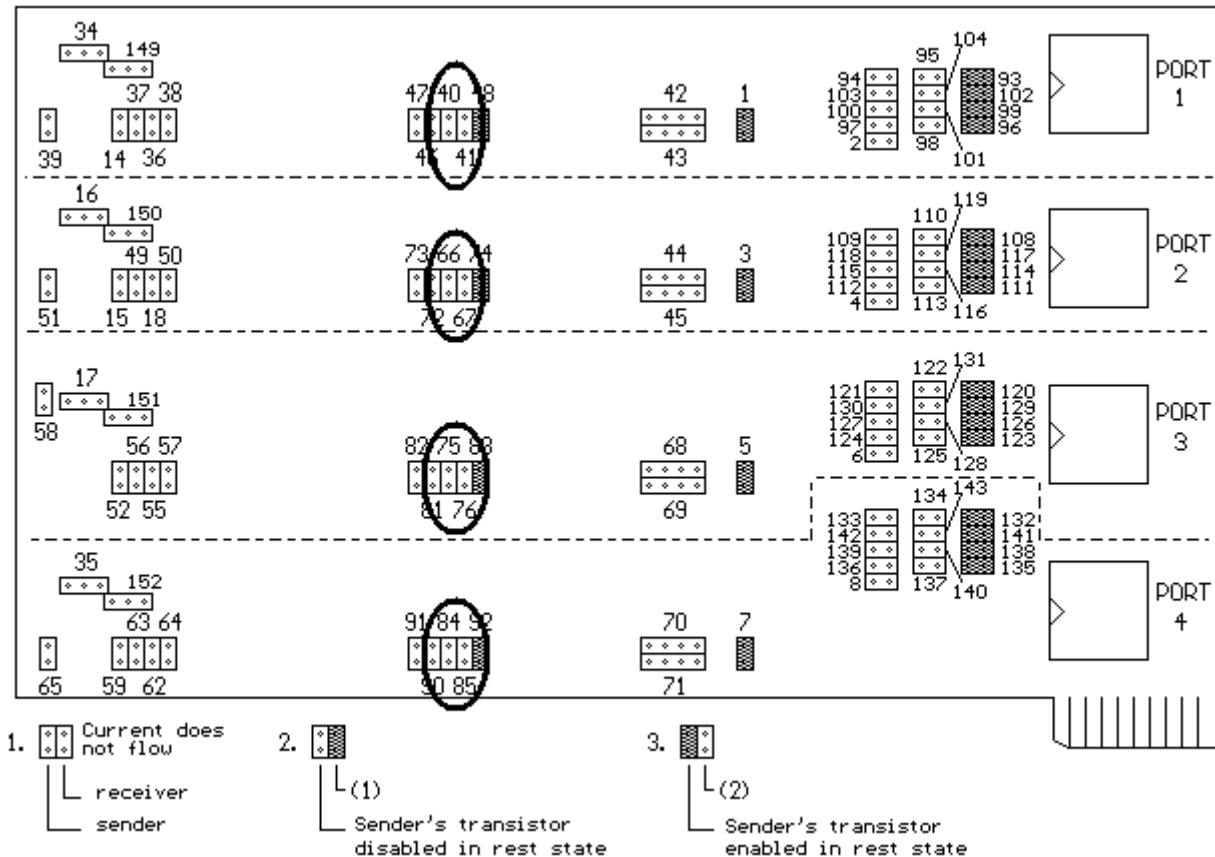
If removed, the sender is in rest state. It means that no current is flowing in the output stage of the sender. If adjusted and in rest state, the output stage of the sender is enabled (20mA).

BR41

It adapts on the board the level of the receivers's receive line. A current flow of 20mA produces in the input circuit, without jumper BR41, a logic "1" at the receiver's input. With BR41, this level is inverted and the result is a logic "0" (BREAK). This logic "0" can be inverted in his turn: no current flow in the input circuit, and if jumper BR41 is adjusted, produce a logic "1" at the input of the board. A logic "0" is produced when a current of 20mA flows in the input circuit.

These configuration possibilities demonstrate that each input and output can be adapted individually to the most various requirements of 20mA signals.

Fig. 3k - LEVEL INVERSION FOR EMISSION AND RECEPTION



- (1) at 20mA the reception level = "1" in the input circuit
- (2) at 20mA the reception level = "0" in the input circuit (BREAK)

3.5.4 Sender and receiver - active or passive

You can determine over jumper field BR42 whether the sender is in active or passive mode.

In passive mode a current of 20mA is supplied by the connected periphery. In active mode a current of 20mA is taken from the internal constant current source.

Since current is supplied by a constant current source and observed in active mode, different line lengths are automatically compensated in the range of the permissible load. The maximum permissible load is mentioned in chapter 3.5.1.

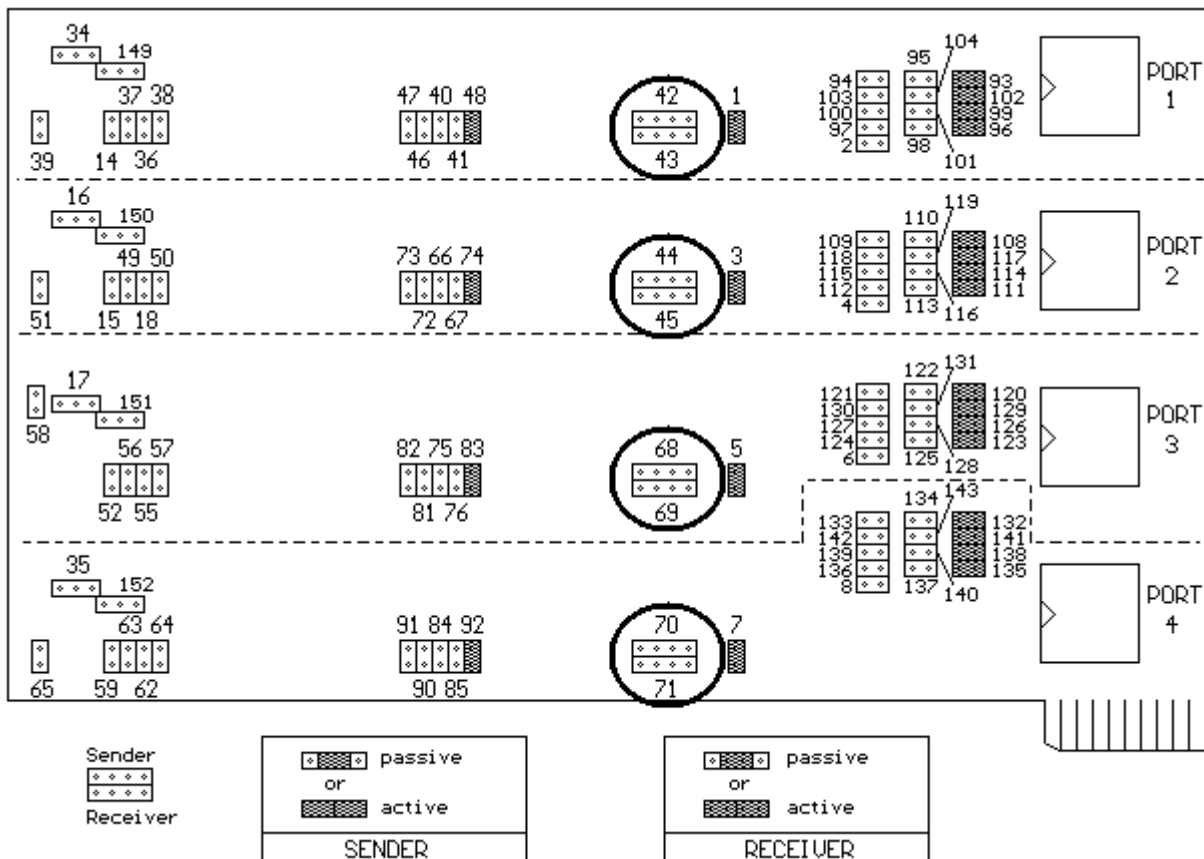
The figure 31 represents the settings for active and passive emission. The receiver as well as the sender can be set on two operating modes:

- 20mA are supplied by the external sender
In this case the receiver is passive.
- or 20mA are taken from the internal constant current source.
In this case the externally connected sender is active.

The adjustment for the receiver occurs over jumper field BR43.

In other cases are valid the technical specifications of the sender. Figure 31 shows the jumper configurations for the active und passive receiver.

Fig. 31 - SENDER AND RECEIVER CONFIGURED IN ACTIVE OR PASSIVE MODE



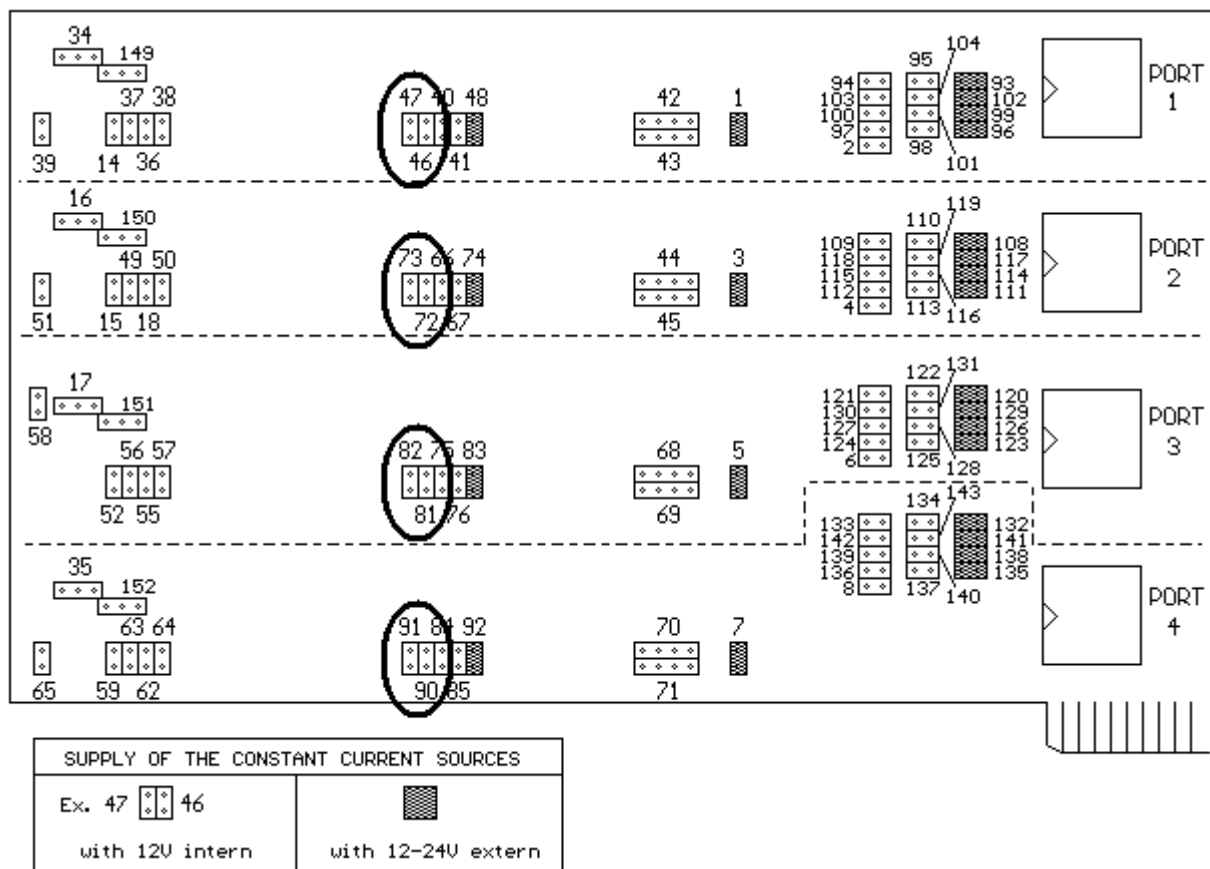
3.5.5 Supply of the constant current sources and of the DC/DC converters in option G

With versions **PA755VRC** and **PA 755VC** it is possible to feed the current sources with the intern 12V of the PC. There is in this case no galvanic separation to the internal system ground. If a galvanic separation to the system ground is needed, it is possible to feed the supply voltage of the constant current sources externally over the 37 pole MIN-D connector. For Port 1 you would adjust jumper BR46 (+12V to +24V extern) resp. BR47 (0V extern).

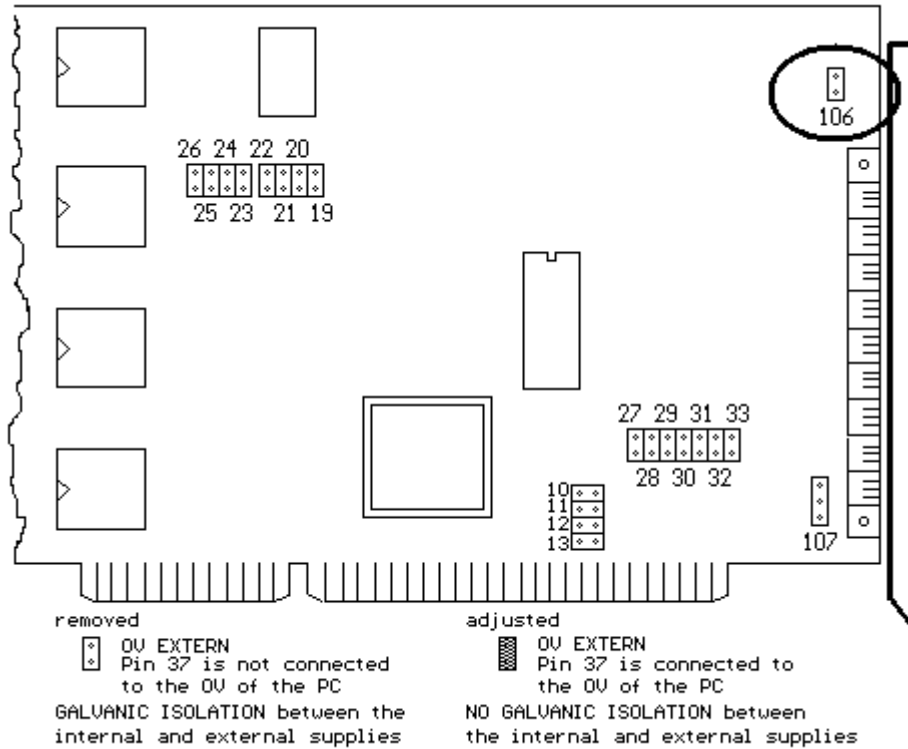
The ports functioning with the internal supply voltage are galvanically interconnected over the internal ground. It is the same for ports supplied with the external voltage. They are also galvanically interconnected over the external ground. In the case where both modes are combined the internal and external grounds are galvanically isolated.

In option G, all the 4 ports are galvanically isolated one another and from the whole system. But it is possible to set over jumper BR106 a galvanic connection between the internal PC ground and the external signal source. Refer to configurations in chapters 3.5.5.1 and 3.5.5.2. The external voltage is fed over the 37 pole MIN-D connector, at which pin 37 corresponds to 0V extern, and pin 19 to +24V extern. This external voltage is protected on the board by fuse F1 (630mA). See right side of the component scheme in chapter 10.1.

3.5.5.1 Selecting the supply voltage for constant current sources



3.5.5.2 Setting a galvanic isolation between the internal and external supply voltage (current loop)

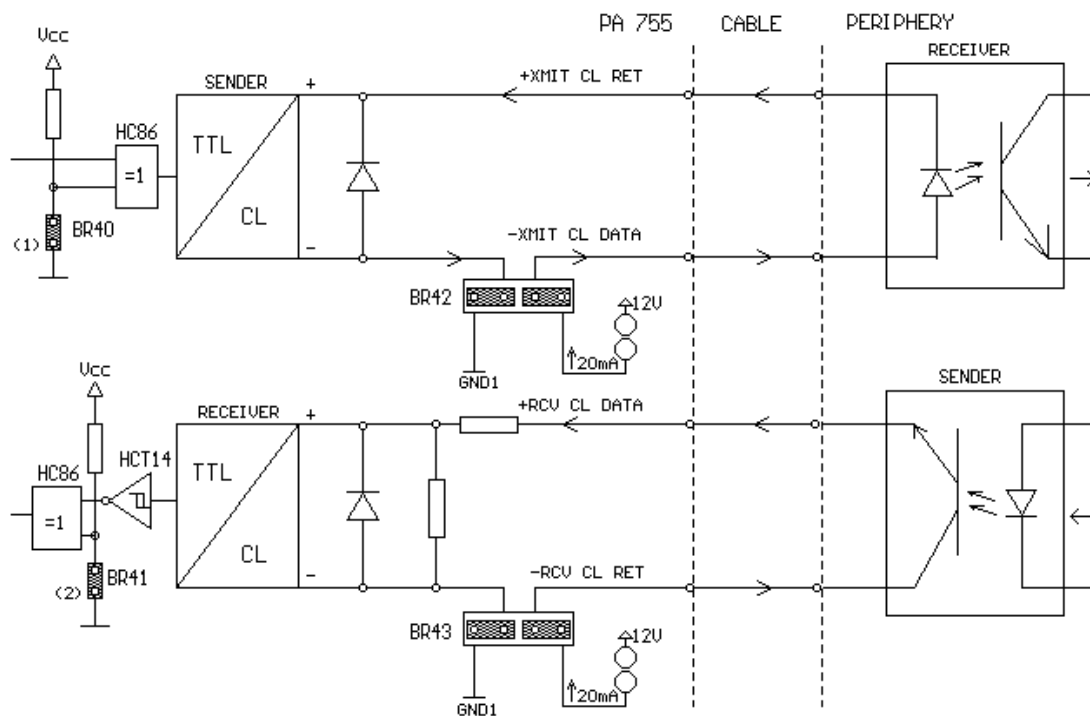


3.5.6 Connection examples in mode current loop

The two following examples represent the principle of the active and passive settings. They are valid for Port 1.

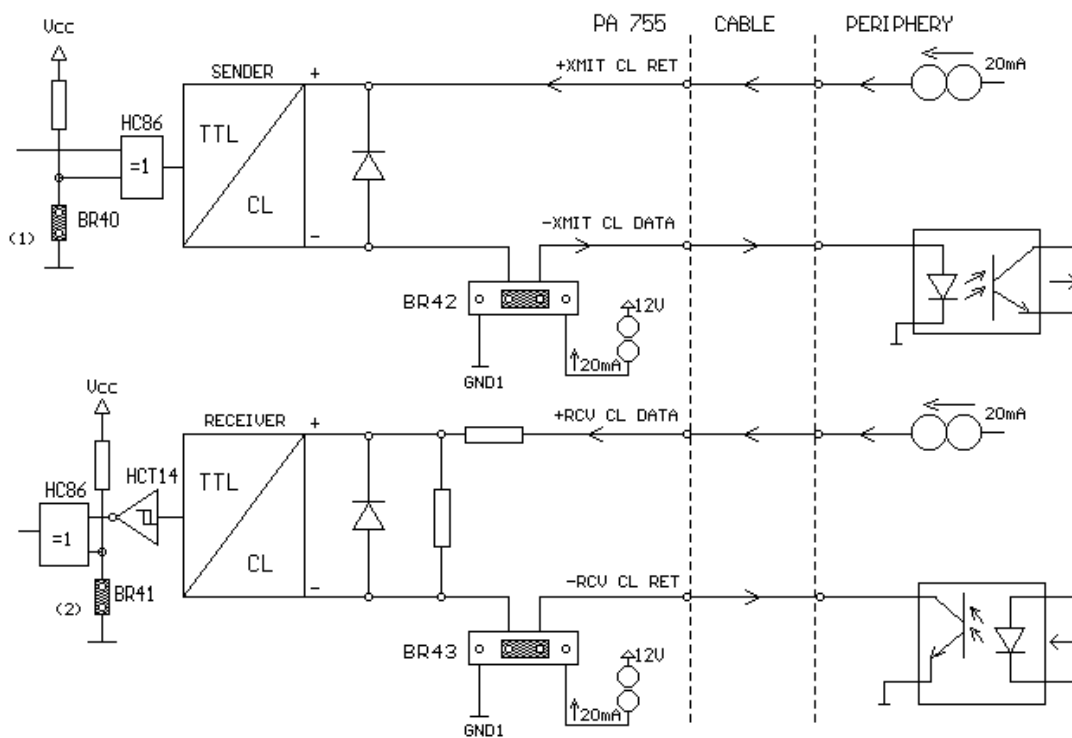
3.5.6.1 Sender active, receiver active (Port 1)

20mA constant current supplied from the internal constant current sources.



3.5.6.2 Sender passive, receiver passive (Port 1)

20mA constant current supplied from the external periphery.

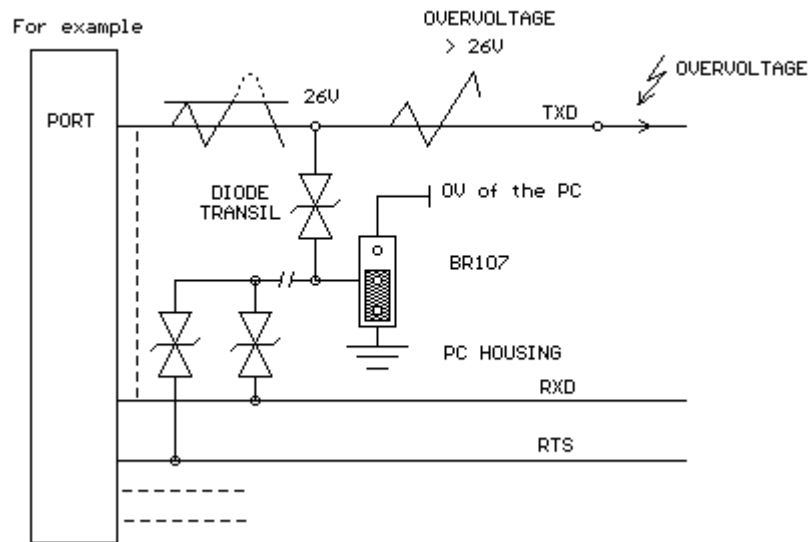


- (1) Logic level inversion for emission (see chapter 3.5.3)
- (2) Logic level inversion for reception (see chapter 3.5.3)

4. Protection of the inputs and outputs

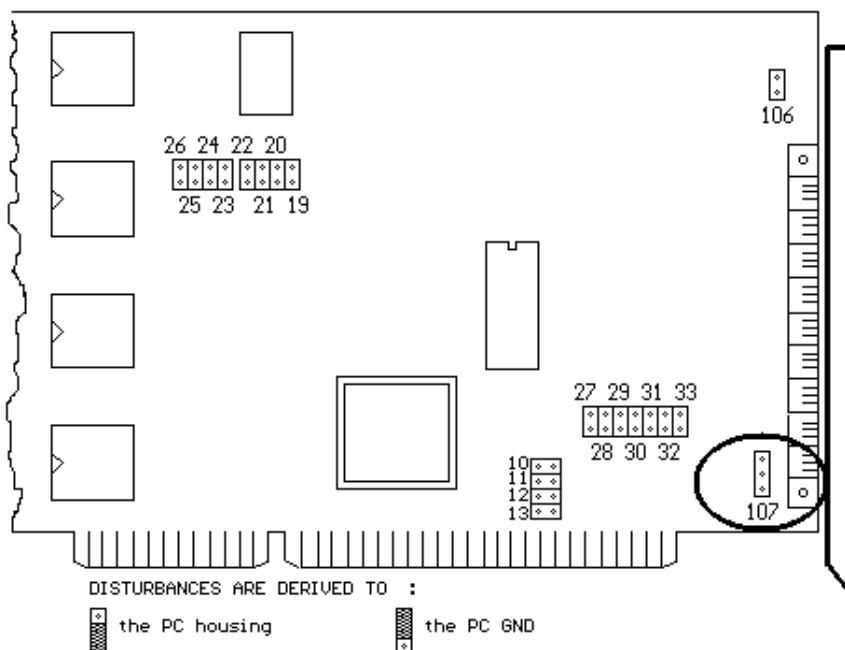
All inputs and outputs connected to the outside world over the 37 pole MIN-D connector are protected with TRANSIL diodes against overvoltage and transients. The TRANSIL diodes limit disturbances at a level of 26V.

Fig. 4a - OVERVOLTAGE DIODE IN INPUT AND OUTPUT LINES



Over jumper BR107, the user can decide whether current, which flows in the case of disturbance through the protection diodes, is derived to the PC housing or the internal PC supply ground. We advise you to derive this current, which can be energy-rich in case of disturbance, to the PC housing.

Fig. 4b - DERIVATION OF DISTURBANCES TO THE PC HOUSING OR THE INTERNAL PC GROUND



At delivery, derivation to the PC housing is selected.

5. Addressing the board

There are basically three different ways for addressing board **PA 755**. Whatever the addressing chosen, each port occupies 8 bytes.

The base address is adjusted over DIP switch. In this case the board will occupy 32 continuous bytes in the 64KB I/O address space of the PC. The following ports will then be responded with an offset of 8 bytes.

The second possibility consists in configuring all the ports directly as COM1 to COM4. In this case the COM interfaces will occupy addresses defined by DOS.

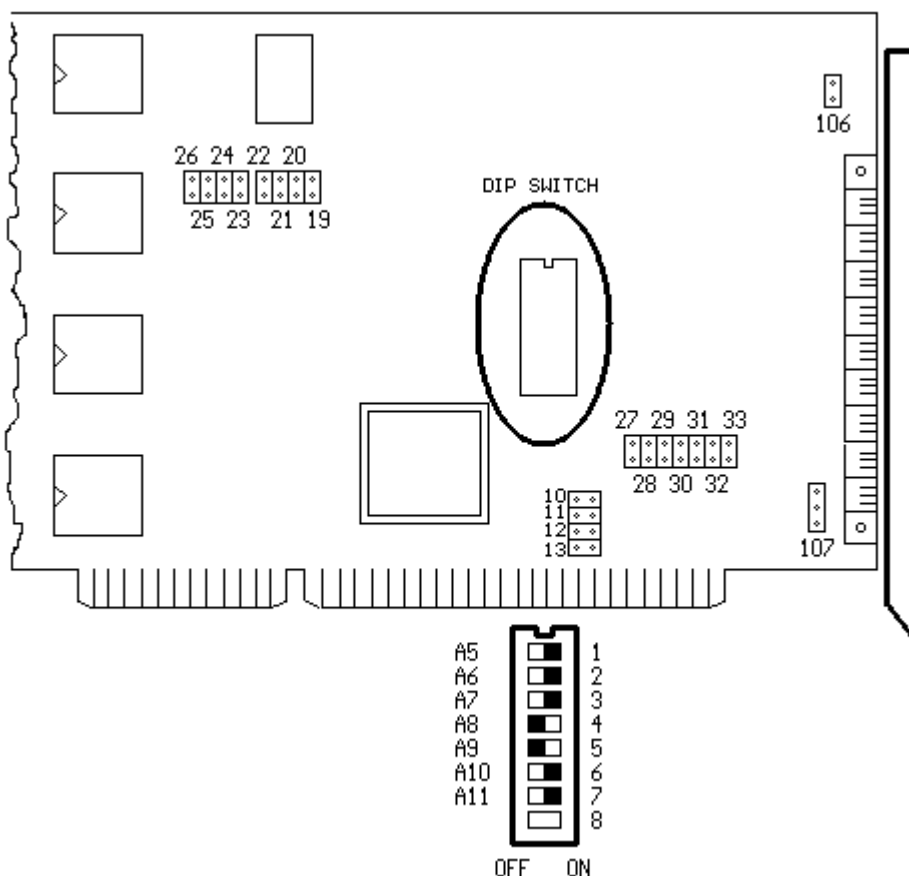
The last possibility combines the two addressing modes mentioned previously. Indeed one part of the ports will be assigned over DIP switch and the other part over the addresses of the COM interfaces.

5.1. Adjusting the base address over DIP switch

If the base address is only adjusted via a DIP switch, the board will occupy 32 continuous bytes within the 64KB I/O address space of your PC. The base address is selected over an 8-pole DIP switch which is set on 300H at delivery.

Fig. 5a - ADJUSTING THE BASE ADDRESS VIA DIP SWITCH

0 = logic '0' = switch in 'ON' position (at delivery on 300H)
 1 = logic '1' = switch in 'OFF' position



With this configuration the different ports occupy the following base addresses.

- Port 1 ---> address 300H
 - Port 2 ---> address 308H
 - Port 3 ---> address 310H
 - Port 4 ---> address 318H
- (always an offset of 8 Bytes)

The board occupies 32 I/O bytes. If you want to install another board, you will address it from 320H upwards.

5.1.1 Decoding address bits A3 and A4

Ports 1, 2, 3 and 4 are decoded internally over address bits A3 and A4. In order that the ports are always addressed internally in the right logic succession, make sure that the address bits A3 and A4 of the base address are on logic "0". See figure below. A special feature of board **PA 755** is that the user can configure each port as a COM interface. Each port is then automatically recognized as such when the PC is booted. If the board is plugged in a XT only the addresses COM1 and COM2 can be configured, whereas if plugged in an AT the addresses COM1 to COM4 are configured.

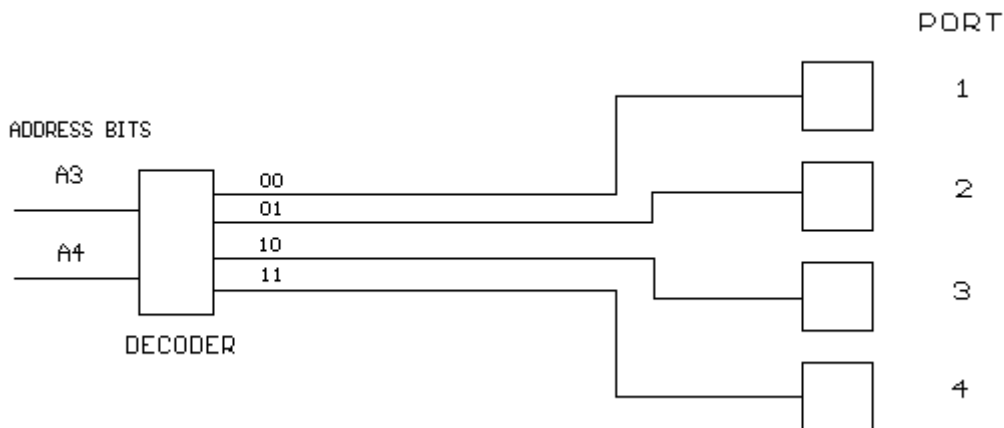
BASE ADDRESS SET ON 0300H

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0				3				0				0			

WRONG ADJUSTMENT: BASE ADDRESS SET ON 0310H

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0
0				3				1				0			

PORT SELECTION THROUGH ADDRESS BITS A3 AND A4



TRUTH TABLE

Address bit		Port
A4	A3	
0	0	Port 1
0	1	Port 2
1	0	Port 3
1	1	Port 4

The base address is decoded in the right order, if the address bits A3 and A4 (= "0") select the **first** port.

If the base address was adjusted on 0310H (A4 = "1"), the address bits A3 and A4 **would not select the first** but the third port (see truth table). A continuous addressing would lead to a shifted decoding of the ports.

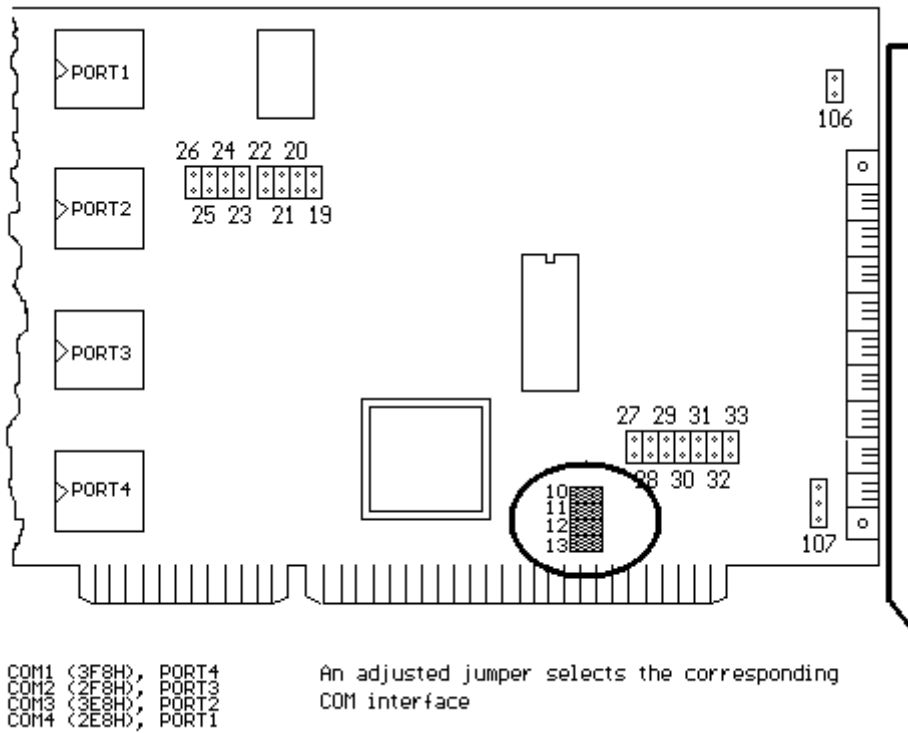
5.2 Addressing at COM1, COM2, COM3, COM4 (over jumper)

Each port can be configured as a COM interface for being recognized automatically as such from the PC when booted. COM1 and COM2 are for XTs, COM1 to COM4 for ATs.

To each COM interface is assigned a port:

- COM1 --> Port 4
- COM2 --> Port 3
- COM3 --> Port 2
- COM4 --> Port 1

Fig. 5b - ADDRESSING AS COM1-COM4 OVER JUMPER



5.3 Simultaneous addressing over DIP switch and jumper

Attention! The individual configurations over DIP switch and over jumper described in chapters 5.1 and 5.2 are not valid in this case.

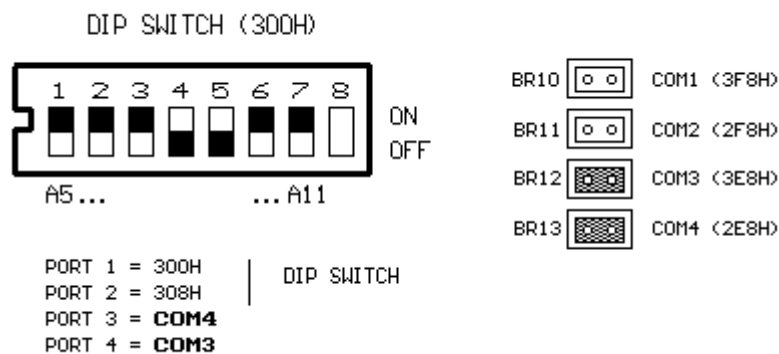
This configuration combines the addressings over DIP switch and jumper. Since there is resulting a wide range of configurations, it is not possible to assign easily to each port a base address.

The following figures describe three configuration possibilities.

To Port 1 is always assigned the base address selected via DIP switch. Each next port, that has not been defined as COM interface over jumper, is located in the logic sequence with an offset of 8 bytes to the previous port.

For the ports addressed via DIP switch, the sequence starts with Port 1 and continues in the increasing order. Whereas the sequence of ports configured as COM interfaces starts with Port 4 and continues in the decreasing order. The highest COM interface is assigned to the smallest port available which has not been addressed via DIP switch.

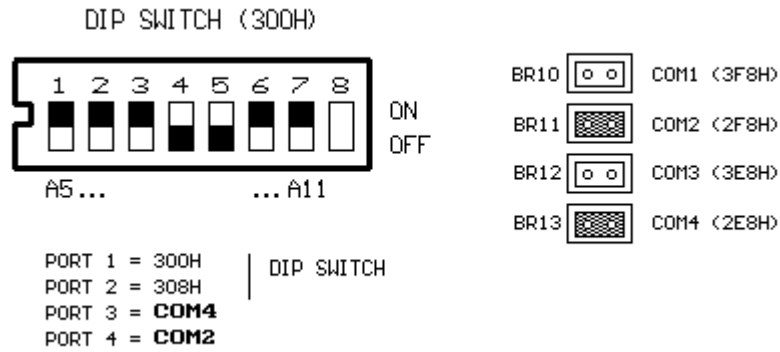
Example 1



The base address is set on 300H. COM4 and COM3 are selected. The ports are assigned as follows: Port 1 and Port 2 are configured over DIP switch. You will find below the address allocation of the different ports:

Port 1	300H
Port 2	308H
Port 3	COM4
Port 4	COM3.

Example 2

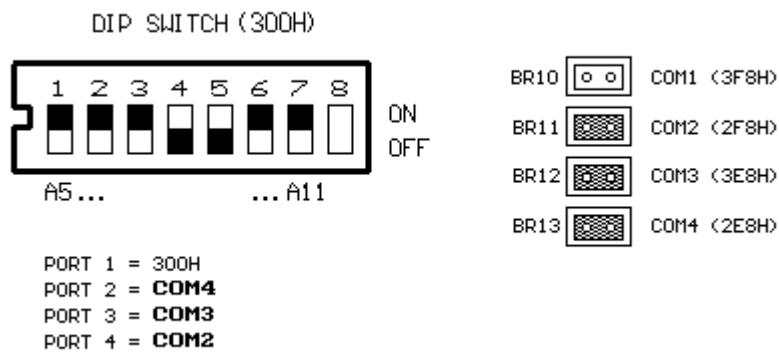


The base address is set on 300H. COM4 and COM2 are selected. The ports are assigned as follows: Port 1 and Port 2 are configured via DIP switch. The address allocation of the ports is the following:

- Port 1 300H
- Port 2 308H
- Port 3 COM4
- Port 4 COM2.

Two COM interfaces are selected. Port 3 is still configured as COM4, and Port 4 now as COM2. In the first example Port 4 was configured as COM3.

Example 3



In example 3, one port is set at the base address 300H. Three ports are configured as COM2, COM3 and COM4. Thus the following allocation:

Port 2	COM4
Port 3	COM3
Port 4	COM2

If you adjust jumper BR10 instead of BR11, COM1 would be assigned to Port 4. Port 2 and Port 3 would not be changed.

The same is valid if COM1, COM2 and COM4 are used instead of COM3 (jumper BR12). In this case the addresses would be allocated as follows:

Port 2	COM4
Port 3	COM2
Port 4	COM1

If only COM1 and COM2 are needed, jumpers BR10 and BR11 must be adjusted, BR12 and BR13 must be removed. Thus the following address allocation:

Port 1	300H
Port 2	308H
Port 3	COM3
Port 4	COM2

Remark to the next chapter (5.4 I/O map)

Port 1 occupies yyyy to yyyy +7

Port 2 occupies yyyy +8 to yyyy +15

Port 3 occupies yyyy +16 to yyyy +23

Port 4 occupies yyyy +24 to yyyy +31

yyyy = Base address

5.4 I/O Map

Adresse	Register	D7	D6	D5	D4	D3	D2	D1	D0	Read/Write
yyyy	Receiver Buffer	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0	R
yyyy +1	Transmitter Holding	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0	W
yyyy +1	Interrupt Enable	0	0	0	0	Modem Status	Receiver Line Status	Transmitter Holding Register empty	Receiver Data available	R/W
yyyy +2	Interrupt Identification	0	0	0	0	0	Interrupt ID Bit 1	Interrupt ID Bit 0	„0“ if Interrupt is pending	R
yyyy +3	Line Control	Divisor Latch Access Bit (DLAB)	Set Break	Stick Parity	Even Parity Select (EPS)	Parity Enable PEN	Number of Stop Bits (STB)	Word Length Select Bit 1 (WLS1)	Word Length Select Bit 0 (WLS0)	R/W
yyyy +4	Modem Control	0	0	0	Loop	Out 2	Out 1	Request to send (RTS) 2) „1“ Enable „0“ Disable	Data Terminal Ready (DTR)	R/W
yyyy +5	Line Status	0	Transmitter empty (TEMT)	Transmitter Holding Register (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	Data Ready (DR)	R/W
yyyy +6	Modem Status	Data Carrier Detect (DCD)	Ring Indicator (RI)	Data Set Ready (DSR)	Clear to Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Data Clear to Send (DCTS)	R/W
yyyy +7	2) RS422/RS485 Enable (DIR)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	2) Bit 0 „1“ Enable „0“ Disable	R/W

1) Bit 0 ist das LSB. Es ist das zuerst gesandte oder empfangene Bit 2) Steuert den Sender in RS 422 oder RS 485 Mode

6. Interrupt handling

Board PA 755 can execute data transfer either in polling or in interrupt mode. Interrupts can be managed in three different ways:

- Common interrupt:

The individual interrupts of the four ports are connected to a common interrupt.

- Single interrupt:

One interrupt is used per port.

- Common and single interrupts combined:

Ex.: Two ports are managed by common interrupt and the two others by single interrupts.

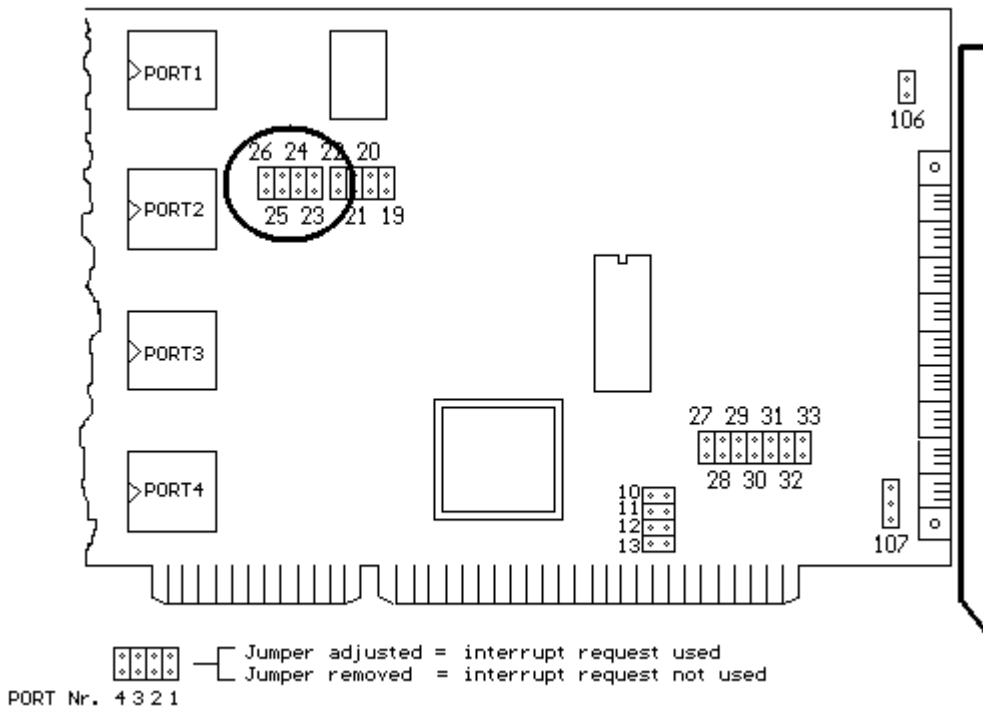
6.1 Single interrupts

Up to four single interrupts can be selected over jumpers BR23, BR24, BR25 and BR26 (BR23 for Port 1...BR26 for Port 4).

If one of these jumpers is adjusted, the corresponding port is then pre-selected for servicing single interrupts. Chapter 6.1.2 describes how the IRQ line is connected to the ISA bus.

If several ports are configured as COM1, COM2, COM3 or COM4, **only single interrupts** can be selected for these ports.

6.1.1 Selecting single interrupts



6.1.2 Selecting the IRQ line to the PC bus for single interrupts

The COM interfaces supported by DOS use standardly the following IRQ lines:

- Port 4 as COM1 --> IRQ4
- Port 3 as COM2 --> IRQ3
- Port 2 as COM3 --> IRQ10
- Port 1 as COM4 --> IRQ11

The single interrupts are assigned to the IRQ lines 4, 3, 10 and 11 of the PC bus over jumpers BR19, BR20, BR21 and BR22. An adjusted jumper establishes a connection to the PC bus. The assignment of the COM interfaces to the IRQ lines is only valid if the board is configured as COM1, COM2, COM3 and COM4 over jumper.

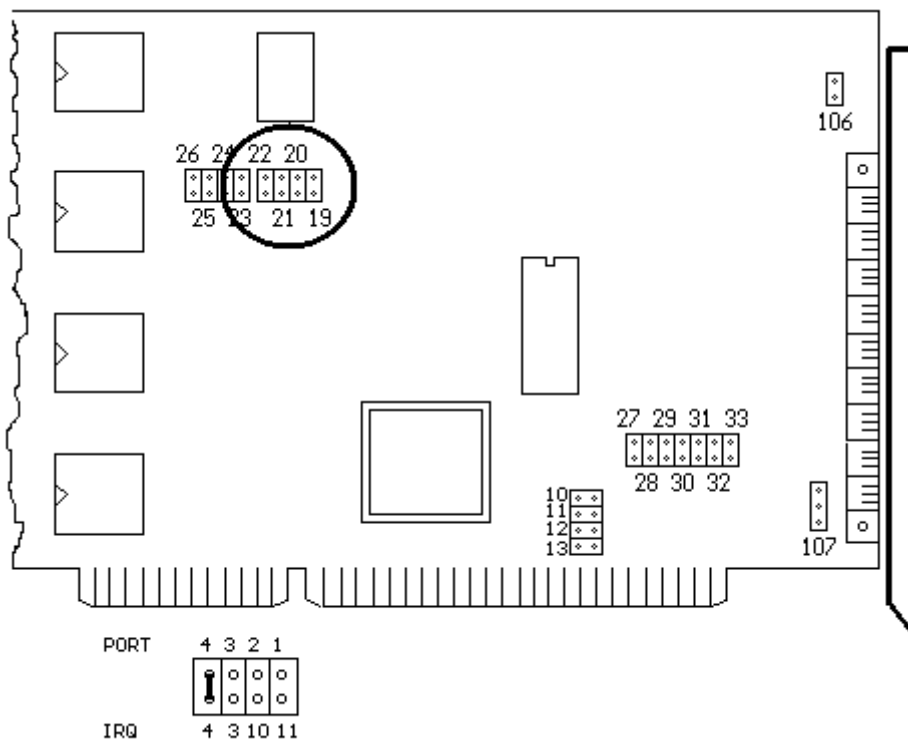
This adjustment is not valid in all other cases:

- if the board is configured over DIP switch
- or when both addressing modes, over DIP switch and jumper, are combined.

Refer to the two following chapters for adjustment.

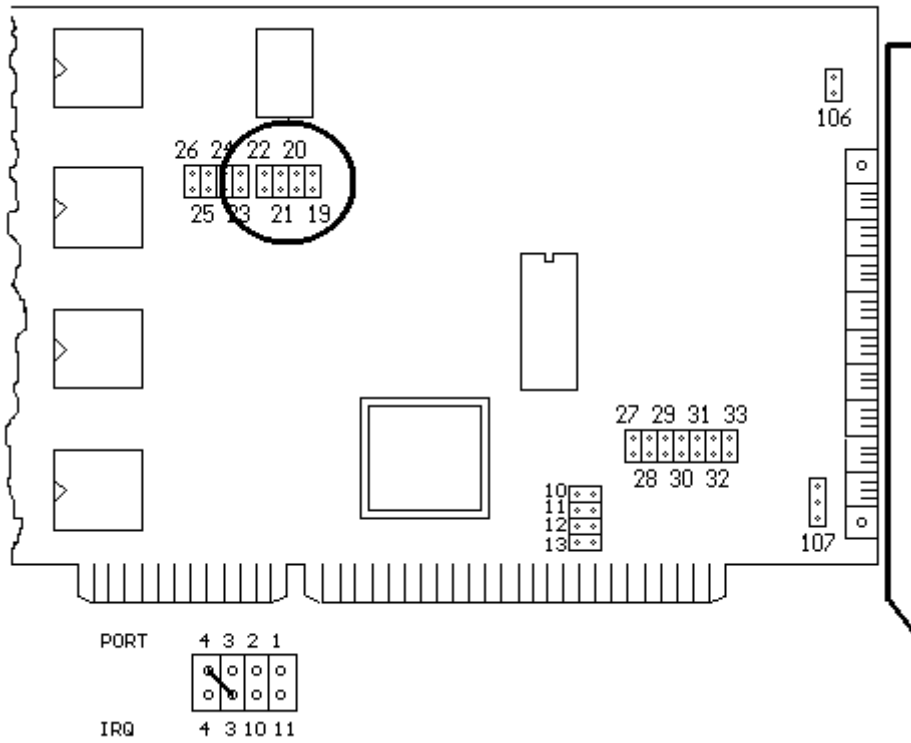
6.1.2.1 Example 1

A port is configured as COM1. The three other ports are addressed via DIP switch. There is resulting the following single interrupt setting: COM1 is assigned to Port 4. Jumper BR23 is adjusted for selecting for Port 1 single interrupts. Since COM1 is automatically assigned to Port 4, Port 4 must be connected with IRQ4 as follows.



6.1.2.2 Example 2

COM2 is addressed over jumper and the other ports over DIP switch. The single interrupt line of COM2 is selected over the adjusted jumper BR26. For assigning COM2, i.e. Port 4, to IRQ3, you establish the connection as shown in the following figure



6.1.2.3 Example 3

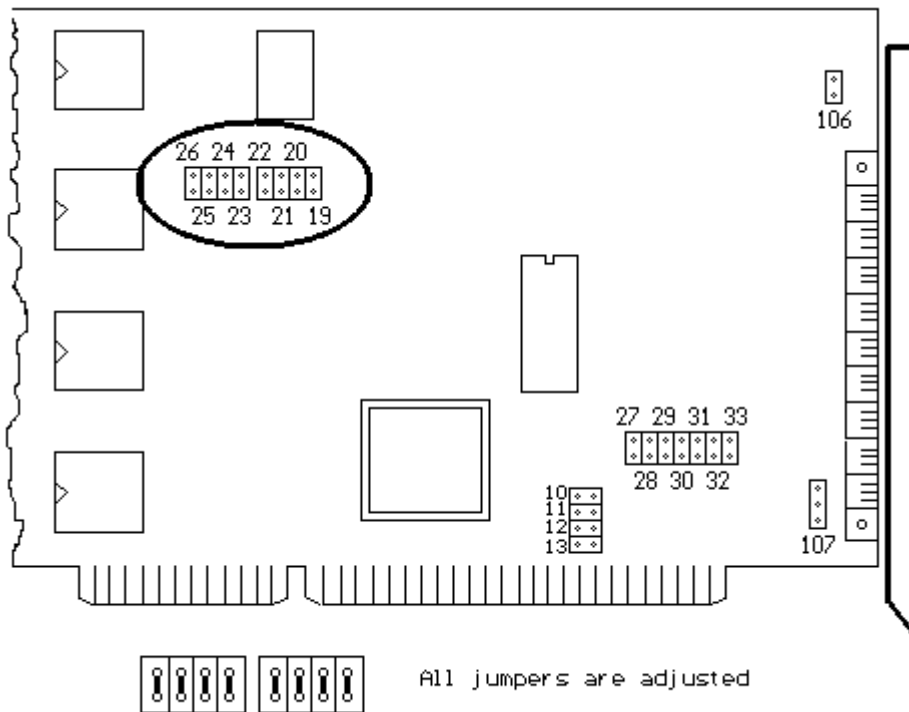
Exception!

The four ports are configured as follows:

- Port 4 as COM1
- Port 3 as COM2
- Port 2 as COM3
- Port 1 as COM4

The single interrupts are selected via jumpers BR19, BR20, BR21, BR22, BR23, BR24, BR25 and BR26. Make sure that they are adjusted.

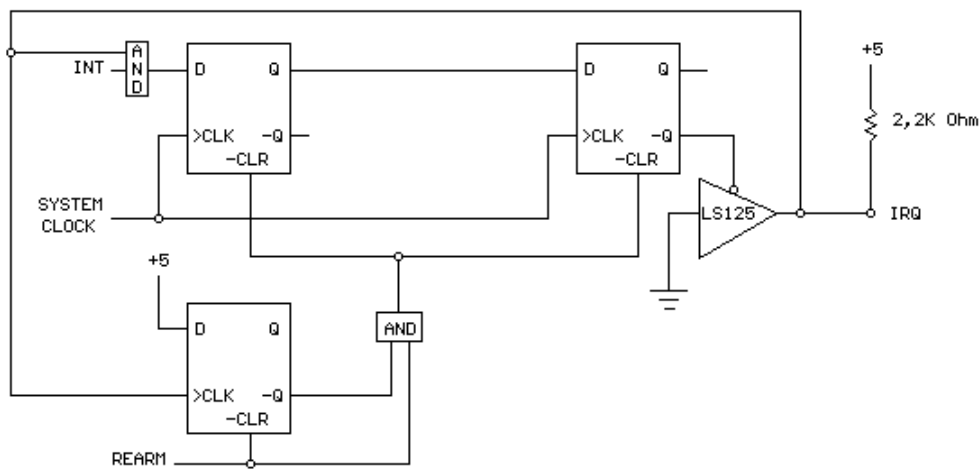
In this exceptional case, the selection of the IRQ line described in chapter 6.1.2 is not valid.



6.2 Common interrupt

An extension board in a PC can usually manage only one free interrupt line. Eventhough the user disposes of very few interrupt lines, several **PA 755** boards can share an interrupt request line through the board's intelligent interrupt logic. The common interrupt takes charge of executing one after the other the single interrupts. An interrupt driver manages common interrupt (see example in chapter 6.2.1).

Fig. 6a - INTERRUPT REQUEST LOGIC FOR COMMON INTERRUPT



You will find in the table below the interrupt request lines and the corresponding REARM address.

IRQ line	REARM address
IRQ3	02F3H
IRQ4	02F4H
IRQ10	06F2H
IRQ11	06F3H
IRQ12	06F4H
IRQ14	06F6H
IRQ15	06F7H

The interrupt request hardware logic is reset by a dummy I/O reading on the REARM address assigned to the IRQ. If another request is present for one of the ports the IRQ line is activated again.

Attention when several boards are used!

- An interrupt is generated on the IRQ line by the leading edge of a negative pulse followed by an high level. The interrupt controller reacts to leading edges.
- The interrupt service routine must establish by reading the different status registers, which board and port have required the interrupt.
- The operations required by the port for processing have to be executed.
- An EOI command to the interrupt controller enables it again.
- The REARM signal on the board must reset the interrupt request logic in order that a new pulse can be produced.

6.2.1 Interrupt routine for managing several boards with the same interrupt request line

Programming example: 3 boards PA 755 use the same IRQ request line IRQ12

```

ISR PROC FAR

    CLI
    PUSH xx                ; For example different registers
    .
    .
    .                    ; Board 1 --> Port 1
    MOV DX,1_Base_Address +2 ; For example Port 1
    IN AL,DX              ; Read interrupt identification of Port 1
    .                    ; Processing
    .                    ; Processing

    .                    ; Board 3 --> Port 2
    MOV DX, 3_Base_Address +10 ; For example Port 2
    IN AL,DX              ; Read interrupt identification of Port 2
    .                    ; Processing
    .                    ; Processing

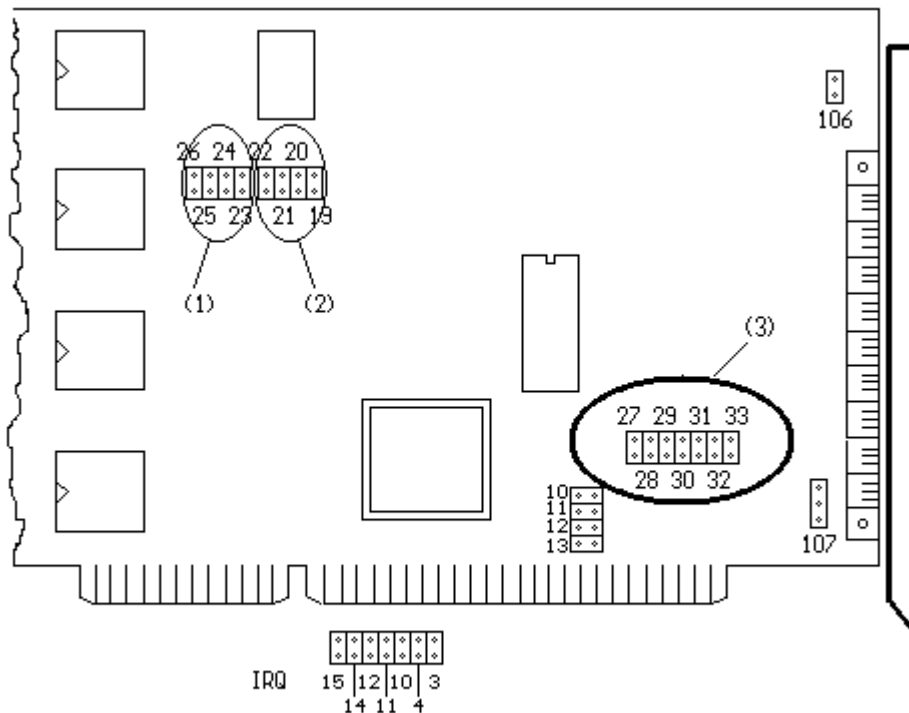
    MOV DX,INT_CTRL1      ; EOI
    MOV AL,EOI
    OUT DX,AL
    POP xx
    .
    .
    .
    MOV DX,6F4H           ; REARM address
    OUT DX,AL             ; Dummy writing, REARM address for IRQ12
    POP DX
    STI
    IRET

END PROC ISR

```


6.2.2 Jumper configuration for selecting the common interrupt

Only the ports **not managed** over single interrupts can be configured in common interrupt. The following IRQ request lines are available:

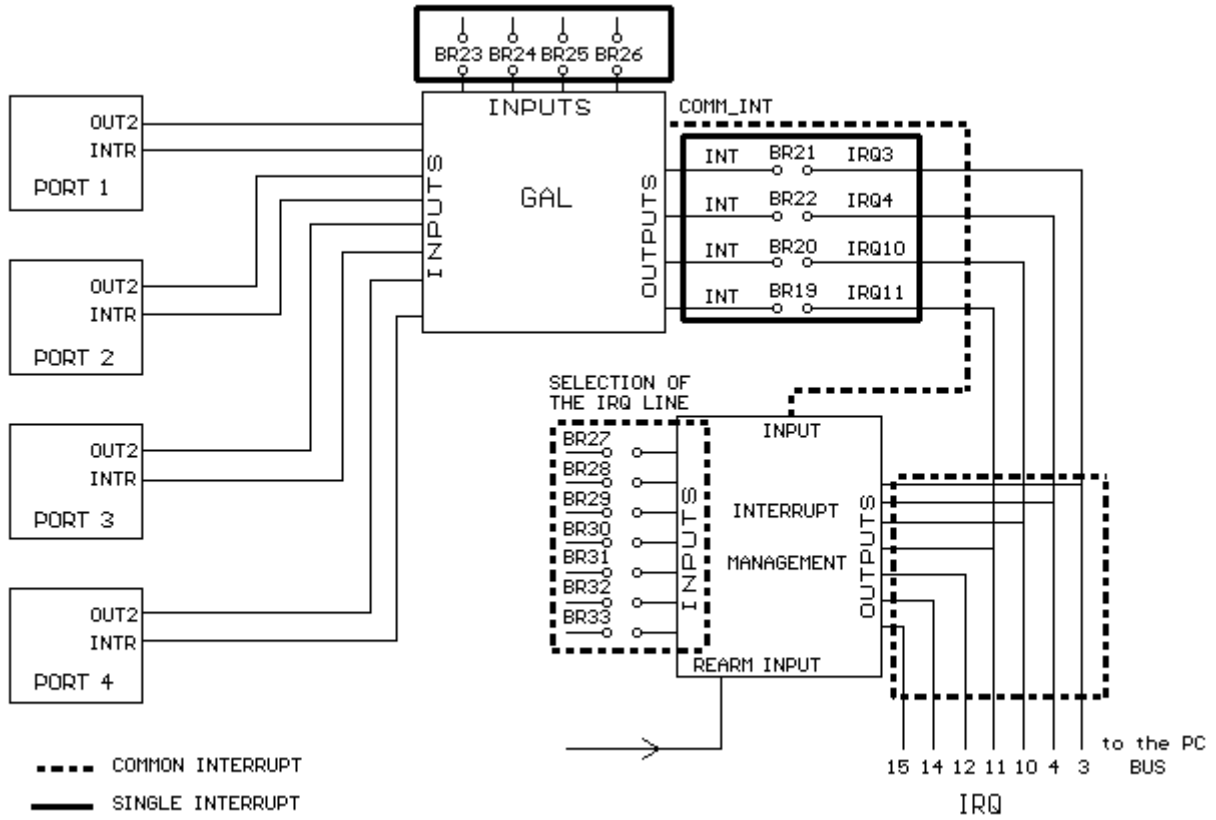


If the line IRQ3-4-10 or 11 is used for common interrupt, make sure that the jumpers selecting single interrupts, i.e BR21, BR22, BR20 and BR19 **are removed**.

IRQ	(3) with jumper	(2) without jumper	(1) without
IRQ3	BR33	BR21	*
IRQ4	BR32	BR22	*
IRQ10	BR31	BR20	*
IRQ11	BR30	BR19	*
IRQ12	BR29		
IRQ14	BR28		
IRQ15	BR27		

- (1) Make sure that the ports managed over common interrupt can not request single interrupts. The corresponding jumper BR23, BR24, BR25 resp. BR26 must be removed (see 6.1.1).

6.3 Principle of the different interrupt request releases



7. An example of board configuration

- Base address: 300H
- COM1 and COM3 are selected: COM1 (3F8H), COM3 (3E8H)
- Single : COM1 ---> IRQ4
- interrupt COM3 ---> IRQ10 | IBM standard

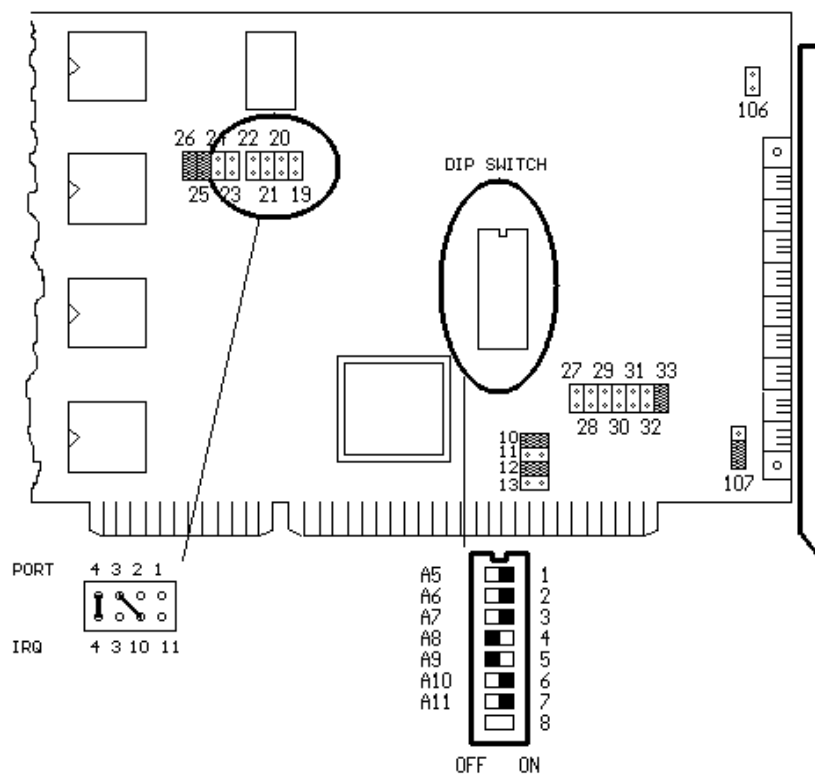
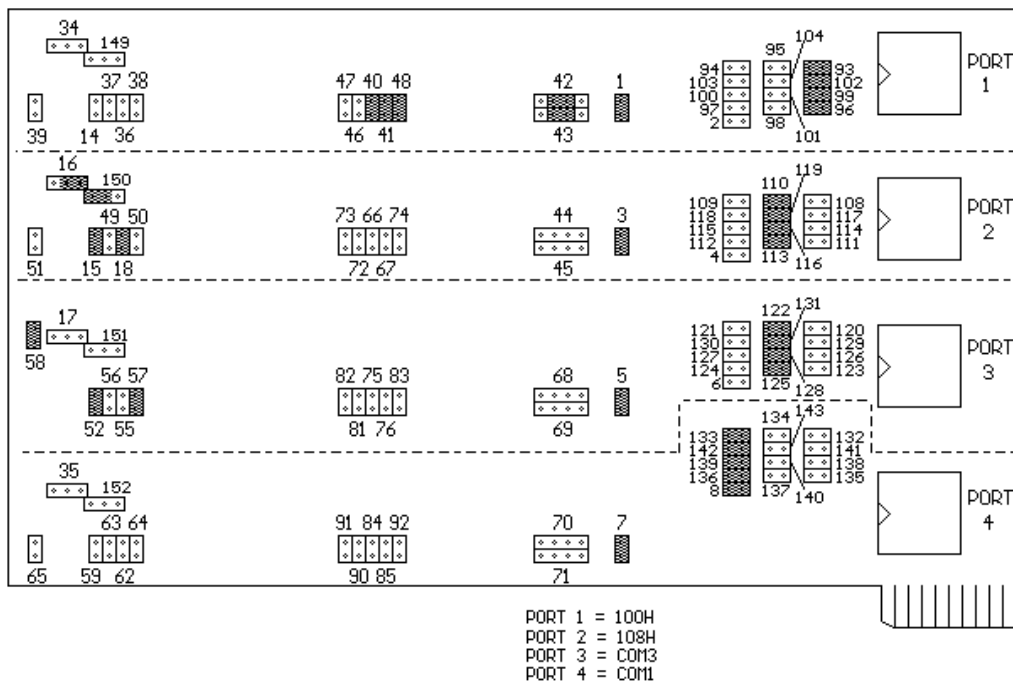
The two other ports are managed with common interrupt (IRQ3)

- COM1 and COM3 are operated as follows:
 - * COM1 in RS 232 without modem control signals
 - * COM3 in RS 485 and continuous sending (GND)
- The port's sender and receiver, addressed at 300H, must function in passive current loop (20mA in rest state)
- The port addressed at 308H must be operated in RS 422.

The sender is enabled over the RTS signal. The receiver is equipped with a terminal resistor.
- The inputs are protected since overvoltage is derived to the PC housing.

Port	Address	IRQ	Mode	Wait States	Derivation of the input protection
4	3F8H COM 1	IRQ 4	RS232 without Modem Control	5	GND of the PC housing
3	3E8H COM 3	IRQ 10	RS485 with continuous Senden (activated through GND)	5	GND of the PC housing
2	308H	Common interrupt on IRQ 3	RS 422 Activated through RTS with terminal resistor	5	GND of the PC housing
1	300H	Common interrupt on IRQ 3	CL Sender and Receiver in CL, passive 20 mA in rest state	5	GND of the PC housing

7.1 Jumper settings



8 Mechanical set-up

For a mechanical and electrical connection a layer of 1,6 mm is used with following measures:
 336 x 99 mm.

The connection with a micro-computer Bus system is realised through a 62-pole direct connector. The board is directly inserted in the PC and is screwed on the back cover of the device.
 For the insertion of the board, please consider the necessary information in chapter 1.

9 Driver

9.1 Introduction

DRV755 is a device driver for IBM PC/XT/AT and compatibles. With this driver it is possible to use under MS-DOS up to 4 serial ports of board PA 755 designated XCOM1 to XCOM4. DRV755 is equipped with receiver buffers (512 characters each). Transfer rates up to 56 000 Bauds are therefore possible. This driver uses interrupts for transmitting or receiving data.

Data can be sent and received with any programming language (C, Pascal, Basic etc).

A communications protocol (RTS/CTS) is used and managed automatically by the driver and can be easily inhibited.

At any time you can change the transfer rate and the communication parameters individually for each serial port of board PA 755.

For example the function COPY of MS-DOS allows to send a file on one of the serial ports of the board.

9.2 Floppy disk

The floppy disk contains the following files:

CONFIG.SYS	Example for installing the driver in file CONFIG.SYS
DEMOx.C	Programming example
DEMOx.Pas	Programming example
XMODE.C	Source file
ACTYPE.H	Header file
XMODE.EXE	Initilization program equivalent to the command MODE of MS-DOS. It permits initializing each port of the board.
DRV755.SYS	Driver program to install in CONFIG.SYS.
PA755.EXE	Initialization program for COM3 and COM4
DRV755.EXE	Install program

To install the driver: A: INSTALL C:

9.3 Board PA 755

The serial ports COM1 and COM2 are the standards for PC/XT, COM1 to COM4 are the standards for PC/AT. If you want to use more than 4 ports, you need a board **PA 755** and a driver for having a more easy-to-handle interface between the user program and the board.

The driver is loaded in the memory when the PC is booted. It occupies about 9K bytes and remains resident. For loading the driver automatically while the PC is booted, you have to add a command line in the file CONFIG.SYS (see chapter 9.4).

9.4 File CONFIG.SYS

9.4.1 Configuring the ports in common interrupt

Symbol signification

X = XCOM
 S = Single interrupt
 C = Common interrupt
 , = And

Ex.: X1 = XCOM1
 X1,2 = XCOM1 and XCOM2
 S3 = Single interrupt at IRQ3
 C15 = Common interrupt at IRQ15

```

DEVICE = C:\PA755\DRV755.SYS - A 300 - X1,2 - C3
          |           |           |           |
          driver     driver     base       XCOM
          directory  name       address    used
          common
          interrupt
    
```

This configuration permits the access to 2 serial ports (XCOM1 and XCOM2) in common interrupt (IRQ3).

Can be used in common interrupt: IRQ3, IRQ4, IRQ10, IRQ11, IRQ12, IRQ14 and IRQ15.

9.4.2 Configuring the ports in single interrupt

```

DEVICE = C:\PA755\DRV755.SYS - A 300 - X1 - S3 - X2 - S10
          |           |           |           |           |
          driver     driver     base       XCOM     XCOM
          directory  name       address    used     used
          single     single
          interrupts
    
```

This configuration permits the access to 2 serial ports (XCOM1 and XCOM2) in single interrupt for each serial port. To XCOM1 corresponds IRQ3, and to XCOM2 corresponds IRQ10.

Can be used in single interrupts: IRQ3, IRQ4, IRQ10 and IRQ11.

9.4.3 Configuring the ports in single and common interrupt

```

DEVICE = C:\PA755\DRV755.SYS - A 300 - X1,3 - C15 - X2 - S10
          |           |           |           |           |
          driver     driver     base       XCOM     XCOM     XCOM
          directory  name       address    used     used     used
          single     single     single
          common
          interrupt
    
```

This configuration permits the access to 3 serial ports: XCOM1, XCOM2, XCOM3. XCOM1 and XCOM3 use IRQ15 in common interrupt. XCOM2 uses IRQ10 as a single interrupt.

9.4.4 Default setting for the serial ports

The following default setting is valid for the 4 ports.

Baud rate: 9600 Bds
 Parity: None
 Data bit 8
 XON: 11Hex
 XOFF: 13Hex
 Protocol: RTS/CTS

These parameters can be changed by software (see chapter 9.7.3).

If a port is configured in COM1 to COM4 (see chapter 5.2), the driver will recognize it automatically and will initialize it as follows
(for DOS versions < 4.0).

Baud rate: 9600 Bds
 Parity: None
 Data bit 8
 Stop bit 1

Attention!

With all DOS versions, COM3 and COM4 are always managed by DOS and not by the driver.

DOS versior	COM used	XCOM used	Software to use
< 4.0	1 to 4	0	PA755.EXE
< 4.0	at least 1	at least 1	DRV755.SYS
< 4.0	0	1 to 4	DRV755.SYS
>= 4.0	1 to 4	0	-
>= 4.0	at least 1	at least 1	DRV755.SYS
>= 4.0	0	1 to 4	DRV755.SYS

9.5 Using DOS functions

Since the driver DRV755.SYS uses peripheral names, it can be directly called in the command line with MS-DOS functions.

Ex.:

- COPY ABC.TXT XCOM1 Copies file ABC.TXT on XCOM1
- COPY XCOM1 ABC.TXT Reads from port XCOM1 and copies data in ABC.TXT. For this example, it is necessary that the code 1AH (end of file under DOS) is sent at the end of receiving. Otherwise MS-DOS would expect infinitely on this end of file code and would not give the hand.

9.6 Programming with DRV755.SYS

The access to a serial port is possible with any programming language and is carried out in the same way as for accessing to a disk file.

9.6.1 Pascal

```
PROGRAM TEST

USES
    CRT, DOS;

VAR
    Input, Output : TEXT;          (* TEXT file *)
    Send           : STRING [80];  (* Buffer containing the message to
    be sent*)
    Receive        : STRING [80];  (* Buffer containing the message to
    be received *)
BEGIN
    ASSIGN (Input, 'XCOM1') ; RESET (Input);          (* Open the
communication *)
    ASSIGN (Output, 'XCOM1') ; REWRITE (Output);      (* channel XCOM1
*)
    SEND := "This is the message to send";
    RECEIVE := " ";
    WRITE (Output, Send);          (* Send data *)
    READ (Input, Receive);        (* Receive data *)
    CLOSE (Output);
    CLOSE (Input);
END.
```

You will find programming examples on the diskette. The communication type with the driver is the same as for using the file.

9.7 IOCTL functions

The functions IOCTL (I/O control) of MS-DOS allow to configure easily the driver. The following parameters can be changed: baud rate, parity, number of data and stop bits, and the communication parameter protocols RTS/CTS or XON/XOFF.

You can access to the IOCTL functions, either directly over a keyword of the programming language, if it is possible, or indirectly over a MS-DOS interrupt 21Hex by using the function 44Hex.

9.7.1 Example of a direct call in C

```
/*
-----
Project      : -                               Compiler   : BORLAND C++
Module name  : DEMO3.C                         Version     : 2.00
-----
Descript.   : Demonstration for using the driver with board PA755.
-----
*/
#include <dos.h>
#include <fcntl.h>
#include <stdio.h>
#include <process.h>
#include <string.h>
#include <io.h>
#include <conio.h>

#define SEND      27
#define IOCTL     50
#define RECEIVE   512

/*
-----
Function name : main
-----
*/
void main (void)
{
    char Ioctl  [IOCTL];           // Configuration buffer.
    int  Serial2;
    int  Serial1;                 // Handler number returned by 'open'

    Serial1 = open ("XCOM1", O_RDWR | O_BINARY); // Opening XCOM1
(PA755).
    Serial2 = open ("XCOM2", O_RDWR | O_BINARY); // Opening XCOM2
(PA755).
    if (!Serial1 || !Serial2)
    {
        printf ("XCOMx can not be opened !!!");
        exit (1);
    }

    ioctl (Serial1, 2, Ioctl, IOCTL); // Reading configuration.
    printf ("\nConfiguration XCOM2: %s", Ioctl);
    ioctl (Serial2, 2, Ioctl, IOCTL); // Reading configuration.
    printf ("\nConfiguration XCOM1: %s", Ioctl);

    strcpy (Ioctl, "b1200;pN;s1;d8;o17;f19;c0\n");

    ioctl (Serial2, 3, Ioctl, IOCTL); // Writing configuration.
    ioctl (Serial1, 3, Ioctl, IOCTL); // Writing configuration.

    ioctl (Serial1, 2, Ioctl, IOCTL); // Reading configuration.
    printf ("\nConfiguration XCOM2: %s", Ioctl);

    ioctl (Serial2, 2, Ioctl, IOCTL); // Reading configuration.
    printf ("\nConfiguration XCOM1: %s", Ioctl);

    close (Serial1);
    close (Serial2);
}
```

9.7.2 Example of an indirect call in Pascal

```

( *
-----
Project      : -                      Compiler   : TP
Module name  : DEMO3.PAS              Version    : 5.00
-----
  Descript.  : Demonstration for using the driver with board PA755.
-----
*)
PROGRAM DEMO3;

USES
  CRT, DOS;

TYPE
  iotyp = string[80];

VAR Input, Output : TEXT;
    IoInBuf       : iotyp;
    regs          : REGISTERS;

( *
-----
  Function name : IoctlIn
-----
  Task          :
                Reads the configuration of the communication port with a DOS
                function
-----
*)
PROCEDURE IoctlIn (var Handler : Text; var Buf : iotyp);
BEGIN
  regs.BX :=
MEM[SEG(Handler):OFS(Handler)]+256*MEM[SEG(Handler):OFS(Handler)+1];
  regs.AX := $4402;
  regs.CX := LENGTH (Buf);
  regs.DX := OFS (Buf)+1;
  regs.DS := SEG (Buf);
  msdos (regs);
END;

( *
-----
  Function name : IoctlOut
-----
  Task          :
                Initializing the configuration of the communications port.
-----
*)
PROCEDURE IoctlOut (var Handler : Text; var Buf : iotyp);
BEGIN
  regs.BX :=
MEM[SEG(Handler):OFS(Handler)]+256*MEM[SEG(Handler):OFS(Handler)+1];
  regs.AX := $4403;
  regs.CX := LENGTH (Buf);
  regs.DX := OFS (Buf)+1;
  regs.DS := SEG (Buf);
  msdos (regs);
END;

VAR
  Send      : STRING[80];
  Receive   : STRING[255];
  i         : BOOLEAN;
BEGIN
  ASSIGN (Input, 'XCOM1'); RESET (Input);
  ASSIGN (Output, 'XCOM1'); REWRITE (Output);

  Send     := 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
  IoInBuf := ' ';

```

```

IoctlIn (Input, IoInBuf);
WRITELN ('a) Read configuration of XCOM1 : ', IoInBuf);

IoInBuf := 'b1200;pN;s1;d8;o17;f19;c1' + chr(13) + chr(10);
IoctlOut (Output, IoInBuf);

IoctlIn (Input, IoInBuf);
WRITELN ('b) Read configuration of XCOM1 : ', IoInBuf);

CLOSE (Input); CLOSE (Output);
END

```

9.7.3 Setting a new port configuration

The configuration line must end with CR,LF (0DHex, 0AHex)

a) Parametering the transfer rate

Keyword: bxxxx or Bxxxx

xxxx is the wished transfer rate

Possible rates: 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 56000 Bauds

If the value xxxx is not in the list below, a rate of 96 000 Bauds is set automatically.

Example in C

```

strcpy (BufIO; "b9600\n\r");          1
ioctl (Handler, 3, BufIO, strlen(BufIO));
    |
    Setting configuration

```

b) Parametering the parity bit

Keyword: px or Px

x refers to the parity wished

- N = None
- O = Odd
- E = Even

¹ \n\r = n for line feed - r for carriage return

Example in C

```
strcpy (BufIO; "b9600; pN\n\r");1
ioctl (Handler, 3, BufIO, strlen(BufIO));
      |
      Setting configuration
```

c) Parametering the number of data bits

Keyword: dx or Dx

x represents the number of bits which have to be contained in one data word (5, 6, 7, 8).

Example in C

```
strcpy (BufIO, "b9600; pN; d8\n\r");
ioctl (Handler, 3, BufIO, strlen(BufIO));
      |
      Setting configuration
```

d) Parametering the number of stop bits

Keyword: Sx or sx

x : represents the number of stop bits to add to the word that you wish to send over the serial port.

1,2 are possible numbers

Example in C

```
strcpy (BufIO; "b9600; pN; s1; d8\n\r");
ioctl (Handler, 3, BufIO, strlen(BufIO));
      |
      Setting configuration
```

e) Parametering the XON code (if protocol is enabled - see § g)

Keyword: Oxx or oxx

When the receiver buffer is half-empty (256 characters), the driver sends the XON code to the periphery which can carry on emission. The xx value is expressed in decimals.

Default: XON = 11 (Hexdecimal)

Example in C

```
strcpy (BufIO; "b9600; pN; s1; d8; o17\n\r");
ioctl (Handler, 3, BufIO, strlen(BufIO));
      |
      Setting configuration
```

¹ \n\r = n for line feed - r for carriage return

f) Parametering the XOFF code (if protocol is enabled - see § g)

Keyword: Fxx or fxx

When the three-quarters of the receiver buffer are full (384 characters) the driver sends the XOFF code to the periphery which stops emission. The xx value is expressed in decimals.

Default: XOFF = 13 (Hexdecimal)

Example in C

```
strcpy (BufIO; "b9600; pN; s1; d8; o17; f19\n\r"); 1
ioctl (Handler, 3, BufIO, strlen(BufIO));
      |
      Setting configuration
```

g) Parametering the protocol

Keyword: Cx or cx

- x = 0 Disables the XON/XOFF protocol and enables the RTS/CTS protocol.
- x = 1 Enables the XON/XOFF protocol and disables the RTS/CTS protocol

Example in C

```
strcpy (BufIO; "b9600; pN; s1; d8; o17; f19; c1\n\r");
ioctl (Handler, 3, BufIO, strlen(BufIO));
      |
      Setting configuration
```

h) Parameting the timeout

Keyword: Tx or tx

- x = 0 Disables the timeout
- 0<x<65536 Enables the timeout (x = 18 = 1s)

Example in C

```
strcpy (BufIO, "b9600;c0;t18");
ioctl (Handler, 3, BufIO, strlen (BufIO));
      |
      Setting configuration
```

¹ \n\r = n for line feed - r for carriage return

9.7.4 Reading the configuration of a serial port

Example in C

```
ioctl (Handler, 2, BufIO, 80); /* Reading configuration */
printf ("\n configuration XCOMx:%s", BufIO);
```

a) Format of buffer BufIO

```
Baud rate; Data length; Parity; Stop; AdrUsart; IrqUsart; XonXoff;
Prot; Error; Xoncode; Xoffcode; Buffer; Timeout; \n\r;
ODH0AH (cr, lf) 1
```

Baud rate	Transfer rate in bits per second initialized
Data length	Number of bits contained in one data word (5, 6, 7, 8)
Parity	Parity type N = None O = Odd E = Even
Stop	Number of stop bits (1, 2)
AdrUsart	Port address in hexadecimal
IrqUsart	Interrupt number in hexadecimal
XonXoff	State of transmit line 0 = emission is possible 1 = Xoff has been detected --> emission is not possible
Prot = 0	Protocol RTS/CTS is active. The driver controls the filling rate of the receiver buffer. When the three-quarters of the buffer are full (384 characters), the driver sets RTS to "0" and the periphery can no longer send data. When the receiver buffer is half-empty (256 characters), the driver sets RTS to "1" and the periphery can send data again.
Prot = 1	Protocol XON/XOFF is active. The driver controls the filling rate of the receiver buffer. If the buffer is three-quarters full (384 characters) the driver sends XOFF on the serial port and the periphery can no longer send data. If the buffer is half empty (256 characters) the driver sends XON on the serial port and the periphery can again send data.
Errors	1 = Overrun 2 = Parity 4 = Framing 8 = Break These error codes refer to errors happening in the USART and not in the receiver buffer. Errors are only valid for the last character received.
XonCode	between 0-255

¹ \n\r = n for line feed - r for carriage return

9.9 Using XMODE

XMODE is intended for initializing the transfer rate of each serial port. When driver DRV755 is loaded XMODE uses the same parameters as function MODE under MS-DOS.

```
XMODE XCOM1: 9600, N, 8, 1, 18
          |      |   |   |   |
          a      b   c   d   e   f
```

- a) Name of the port to be initialized
- b) Baud rate
- c) Parity
- d) Word data length
- e) Number of stop bits
- f) Timeout value

Attention!

If driver DRV755.SYS has not been loaded in CONFIG.SYS, program XMODE can crash down the PC. All the Read and Write operations of your user program are crashed down on the serial ports.

9.10 Operating modes

The driver supports all the operating modes except RS 485.

- Protocol XON/OFF

Can be used in RS 232, RS 422 and current loop.

- Protocol RTS/CTS

Can be used only in RS 232.

- No protocol

If you do not want to use any protocol, communication must occur **without** modem control signals (see jumper adjustment in chapter 3.2.3 and 3.2.4), and parameter the RTS/CTS protocol (see chapter 9.7.3).

This mode can be set in RS 232, RS 422 and current loop. In this mode the receiver buffer is not managed. If the three-quarters of the buffer are full, it will no longer fill up the characters that it receives and it will lose them.

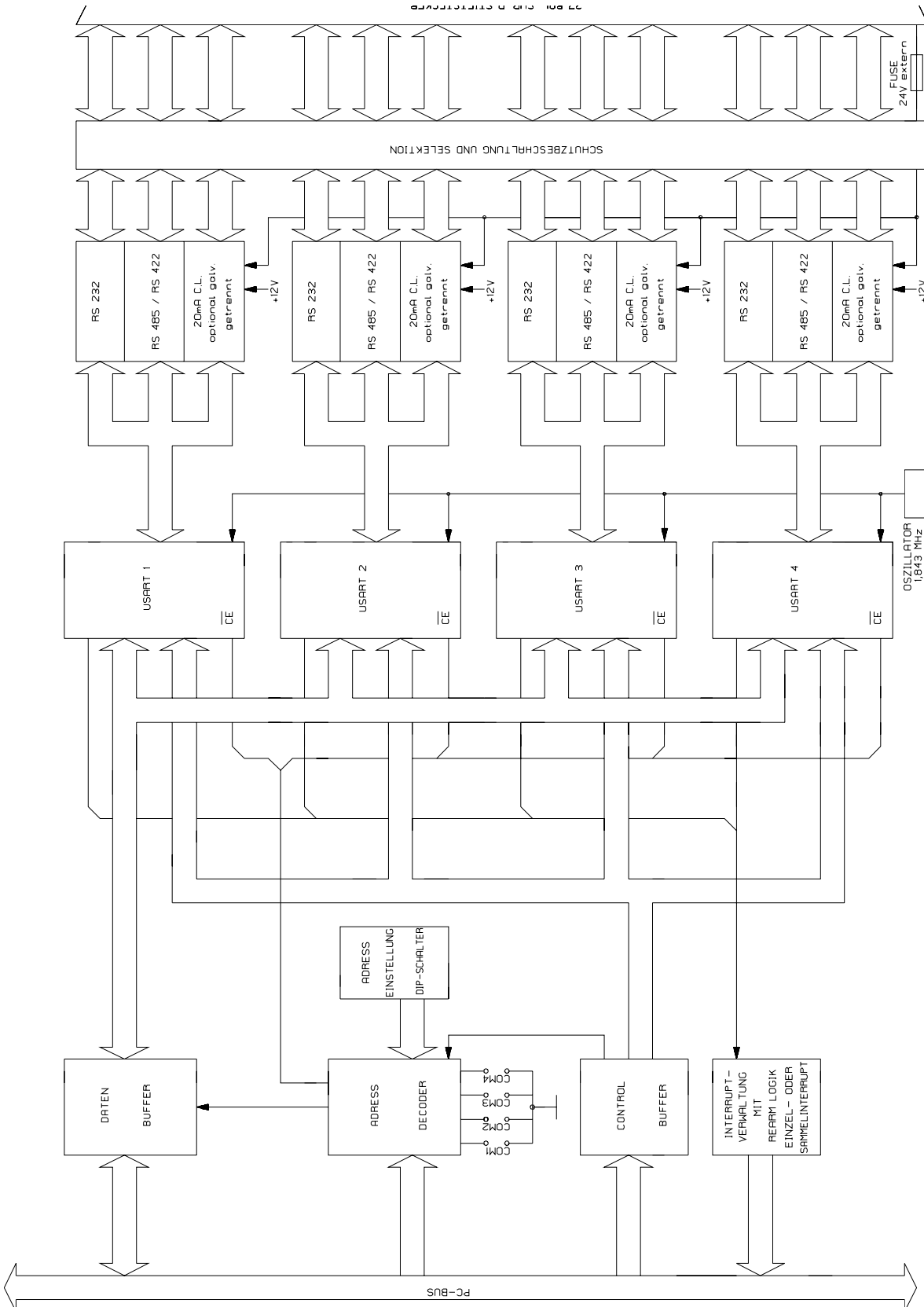
In RS 422, the driver enables emission with RTS, DIR and GND (see chapter 3.3.4) with the XON/XOFF protocol and without protocol.

9.10.1 Installing several drivers

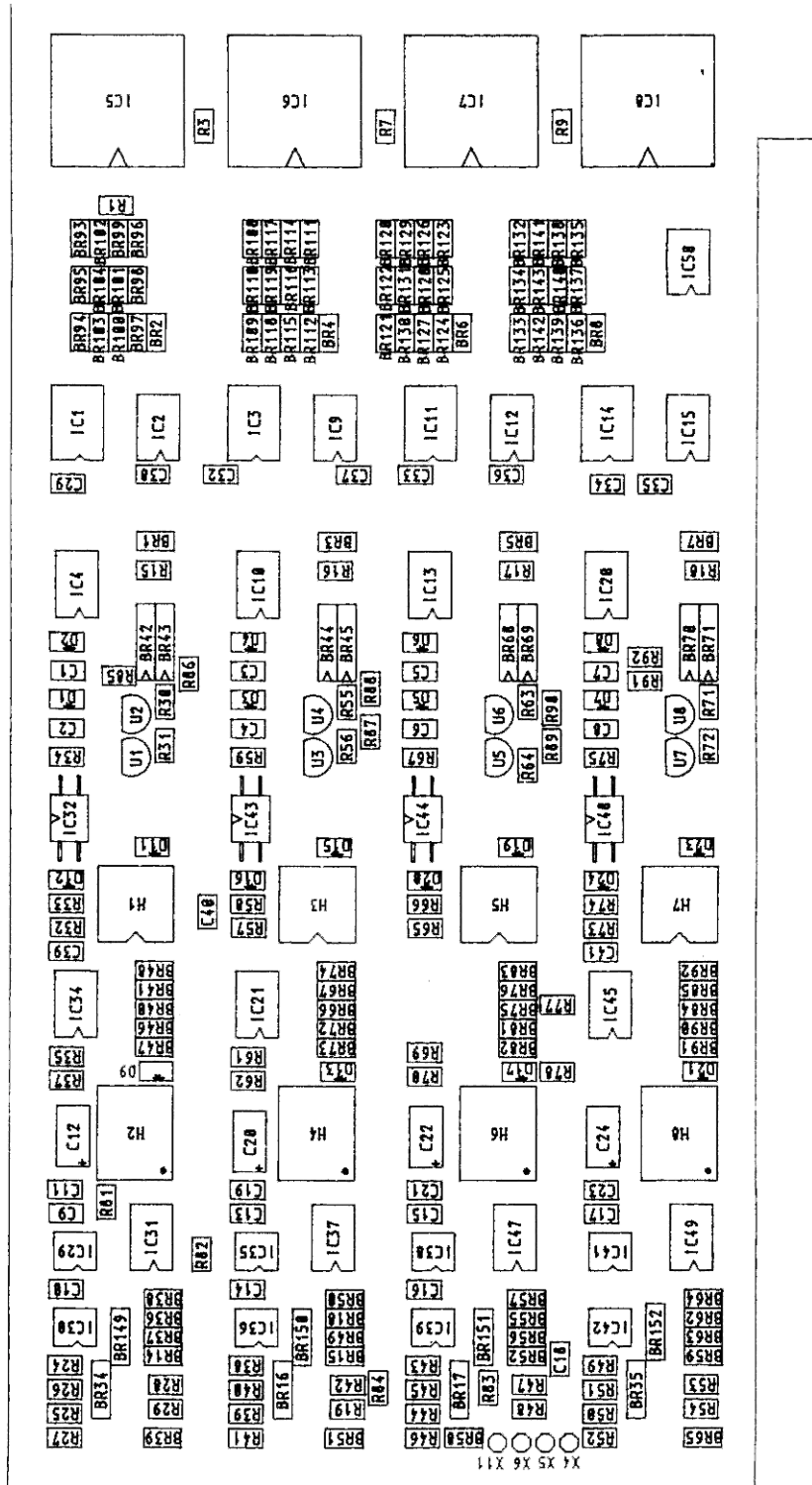
You can use several boards **PA 755** in your PC. For each board, you have to install the driver. Driver 1 will support XCOM1 to XCOM4, driver 2 will support XCOM5 to XCOM8 etc.

10. Appendix

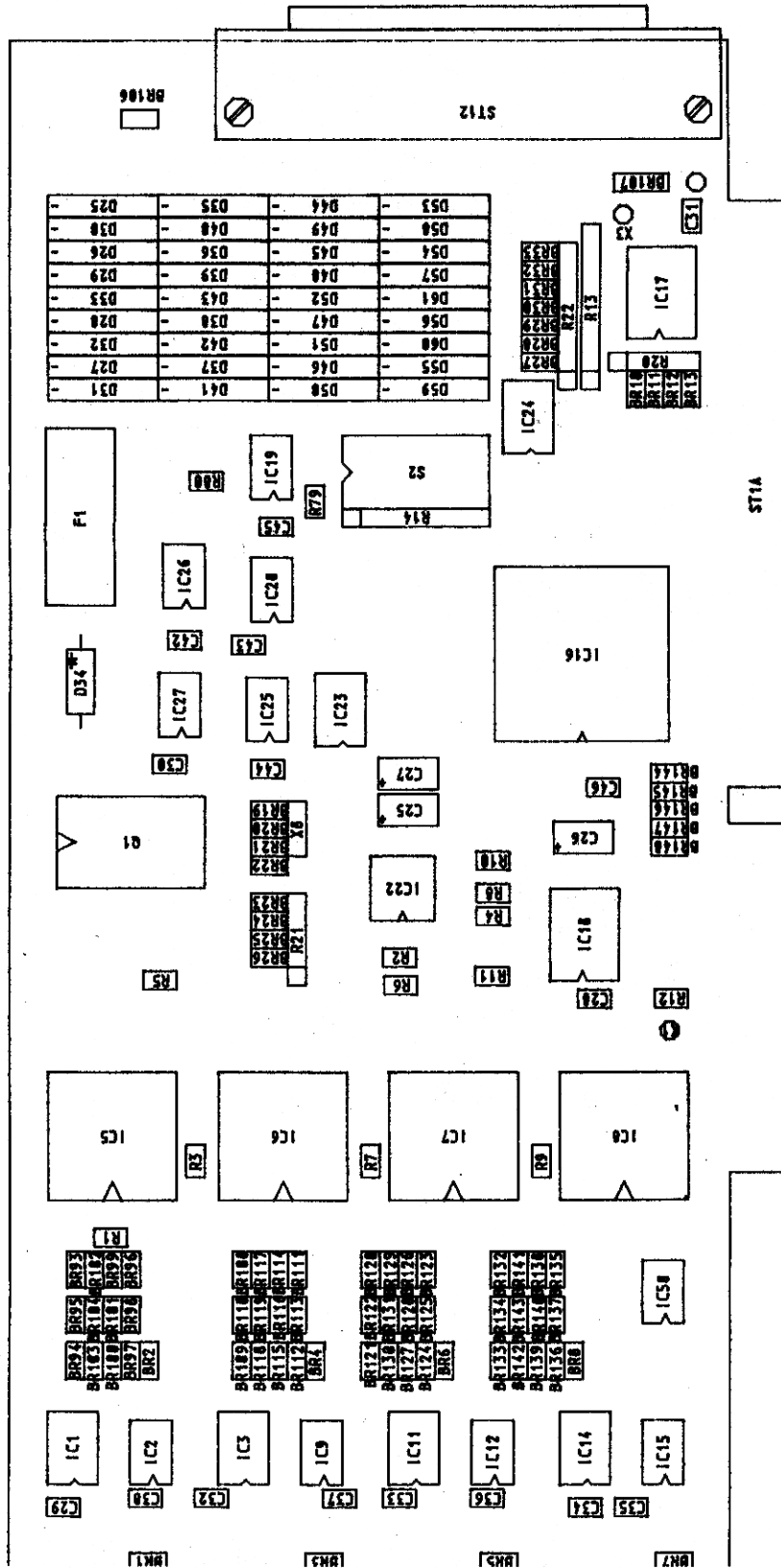
10.1 Block diagram



10.2 Component scheme (left side)



10.1 component scheme (right side)

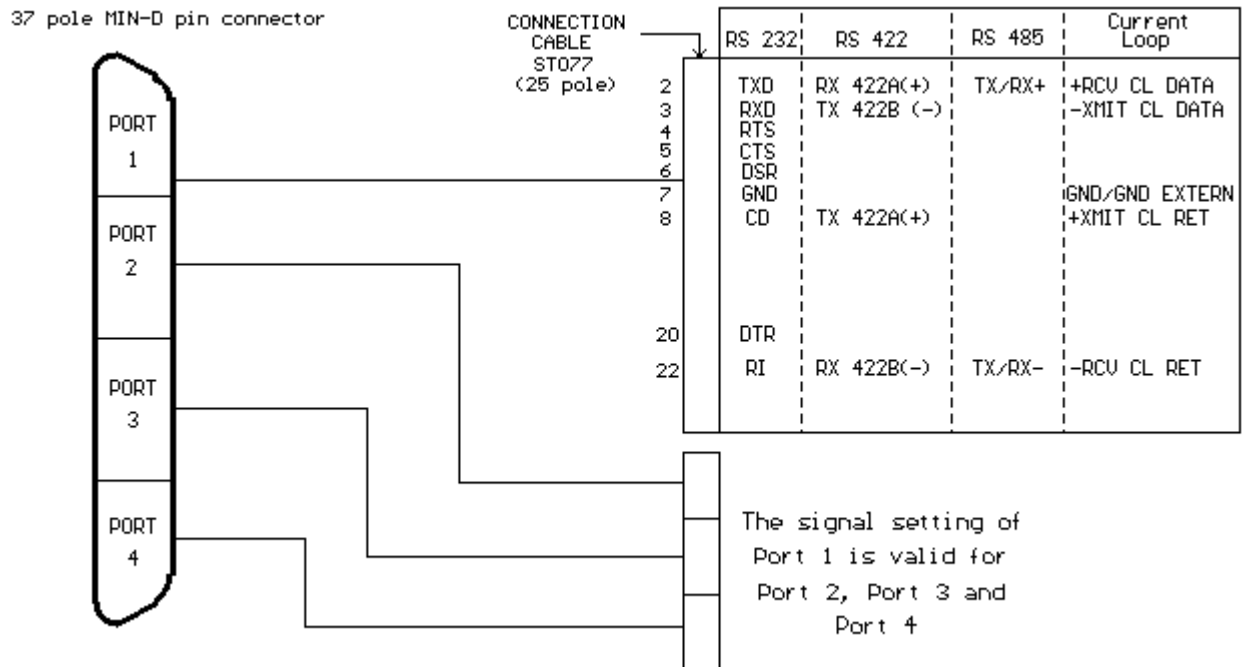


10.3 37 pole pin connector - signals

The periphery is connected to the board over a 37 pole MIN-D pin connector. Pin 19 is not used for the standard connection cables (see chapters 10.3 and 10.4).

RS 485	Current Loop	RS 422	RS 232	RS 232	RS 422	Current Loop	RS 485		
TX/RX-	+24V EXTERN -RCV CL RET	RX 422B (-)	RI	19	GND	RX 422A (+) TX 422B (-) TX 422A (+)	GND/GND EXTERN	TX/RX+	PORT 1
			CTS	37	DTR				
			RTS		TXD				
			DSR		RXD				
					CD				
TX/RX+	GND +RCV CL DATA -XMIT CL DATA +XMIT CL RET	RX 422A (+) TX 422B (-) TX 422A (+)	GND	1	RI	RX 422B (-)	-RCV CL RET	TX/RX-	PORT 2
			DTR		CTS				
			TXD		RTS				
			RXD		DSR				
			CD						
TX/RX-	-RCV CL RET	RX 422B (-)	RI	1	GND	RX 422A (+) TX 422B (-) TX 422A (+)	GND	TX/RX+	PORT 3
			CTS		DTR				
			RTS		TXD				
			DSR		RXD				
					CD				
TX/RX+	GND +RCV CL DATA -XMIT CL DATA +XMIT CL RET	RX 422A (+) TX 422B (-) TX 422A (+)	GND	1	RI	RX 422B (-)	-RCV CL RET	TX/RX-	PORT 4
			DTR		CTS				
			TXD		RTS				
			RXD	20	DSR				
			CD	1					

10.4 Connection cable ST077, 37 pole to 4 x 25 pole



10.4 Connection cable ST078, 37 pole to 4 x 9 pole

