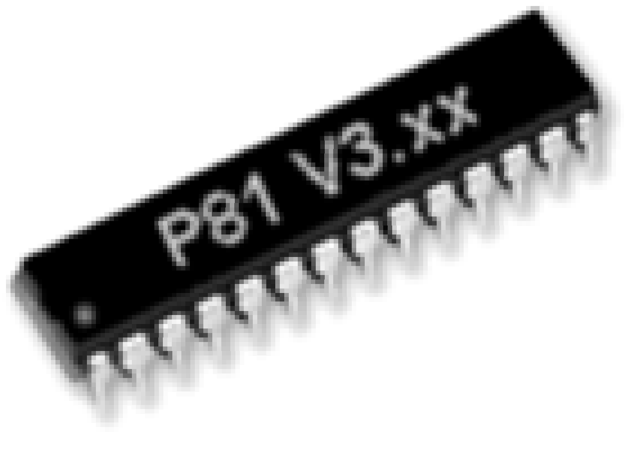
P81 firmware

for TRANSIT (PS-270)

Installation Guide



2003-11-28 Part.no. 5268419

This information is furnished for guidance, and with no guarantee as to its accuracy or completeness; its publication conveys no licence under any patent or other right, nor does the publisher assume liability for any consequence of its use; specifications and availability of goods mentioned in it are subject to change without notice; it is not to be reproduced in any way, in whole or in part, without the written consent of the publisher.

© Nedap IDEAS, P.O. Box 103, NL-7140 AC GROENLO

CONTENTS

1 INTRODUCTION

The P81 firmware is standard TRANSIT (PS-270) firmware with various non-proprietary synchronous communication protocols. It can directly be used to replace a wiegand or magstripe readers in an existing access control system.
The TRANSIT is based on proven microwave technology in the 2.45 GHz ISM band and allows identification of tags at a distance up to 10 meters, even at high speeding passage. The P81 firmware combines microwave identification with low frequency identification at 120 kHz.

The P81 firmware supports a wide range of transponders for various applications. The heavy-duty tag is developed typical for vehicle applications. The window-tags can be mounted easily behind the windshield of a vehicle. The booster-unit is a special window tag, which is able to hold a NEDAP low frequency identification card. This card is read by the booster. The combi-booster combines the features of the window-tag with a booster allowing to identify both vehicle and driver.
The P81 firmware decodes NEDAP PM-transponders, NEDAP Combi-Boosters and manchester encoded transponders such as EM4102 based transponders. HID prox cards are supported when placed inside a combi-booster HID.

The following synchronous communication protocols are supported:

- Barcode Code39
- Magstripe (ISO 7811/2)
- Wiegand 26 bit (H10301)
- Wiegand 32 bit
- Wiegand 37 bit (H10302 / H10304)
- FF 56 bit
- HID Corporate 1000

A simple asynchronous protocol (CR/LF protocol) allows to monitor, debug and setup the reader.

2 GETTING STARTED

This chapter provides simple instructions (only 3 steps) to get the user up and running quickly.

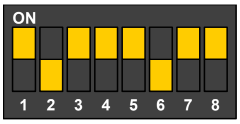
2.1 STEP 1 – HARDWARE CONFIGURATION

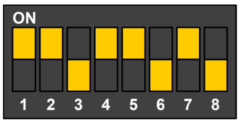
The first step is to configure your hardware with the available DIP-switches and jumpers.

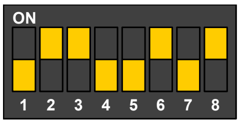
2.1.1 DIP SWITCH SETTINGS

Make sure the DIP-switches are set correctly. The DIP-switches are described in detail in chapter 3.

Below some recommended DIP-switch settings for a few typical applications:

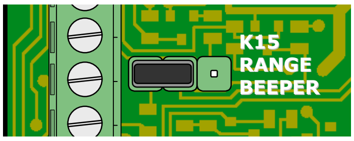
Typical combi-booster HID application:		
Applicable tags:	All (incl. combi-booster HID)	
Communication protocol:	Wiegand 26 (+ RS232 CR/LF 9600 7E1)	

General purpose:		
Applicable tags:	All (also on the low-frequency antenna)	
Communication protocol:	Wiegand 37 (+ RS232 CR/LF 9600 7E1)	

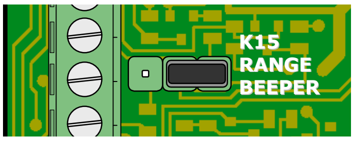
Optimized for fast identification:		
Applicable tags:	Only Nedap tags (no combi-boosters)	
Communication protocol:	RS232 CR/LF 38400 8N1 (+ Magstripe)	

2.1.2 JUMPERS

Enable or disable the BEEPER by setting jumper K15 RANGE-BEEPER.



Beeper OFF



Beeper ON

For a complete description of all other jumpers refer to the TRANSIT installation guide.

2.1.3 FREQUENCY SETTING

The frequency setting only needs to be changed when two TRANSIT readers are near to each other, or when there is much disturbance from other devices in the neighborhood on the same frequency (for example Wi-Fi base stations).
The frequency DIP-switches are described in detail in the installation guide.

2.2 STEP 2 – HARDWARE INSTALLATION

The second step is to connect the wires on your TRANSIT as described below.

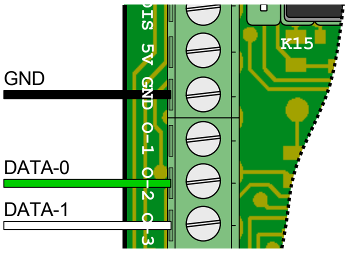
2.2.1 POWER SUPPLY

Connect the AC mains or use a 24VDC power supply. Refer to appendix A when locating these connections.

2.2.2 SYNCHRONOUS COMMUNICATION (WIEGAND, MAGSTRIPE, BARCODE, ETC.)

The synchronous communication interface wiring depends upon the selected communication protocol. In the previous step we selected the communication protocol with the DIP-switches.

In the figure below the wiring for the wiegand protocol is shown.



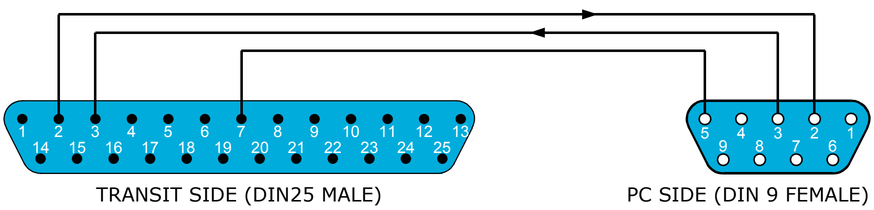
Wiegand wiring

For the other communication protocols see the references below.

- Barcode Code39 — See chapter 5.2.1 on page 15.
- Magstripe (ISO 7811/2) — See chapter 5.3.1 on page 17.
- Wiegand 26 bit (H10301) — See chapter 5.4.1 on page 20.
- Wiegand 32 bit — See chapter 5.4.1 on page 20.
- Wiegand 37 bit (H10302 / H10304) — See chapter 5.4.1 on page 20.
- FF 56 bit — See chapter 5.5.1 on page 22.
- HID Corporate 1000 — See chapter 5.6.1 on page 23.

2.2.3 ASYNCHRONOUS COMMUNICATION (RS-232)

The RS-232 interface is required to complete the software configuration described in chapter 2.3.
The cable must be a so called nul-modem cable. The hardware handshake lines are unused. See the cable specifications below.



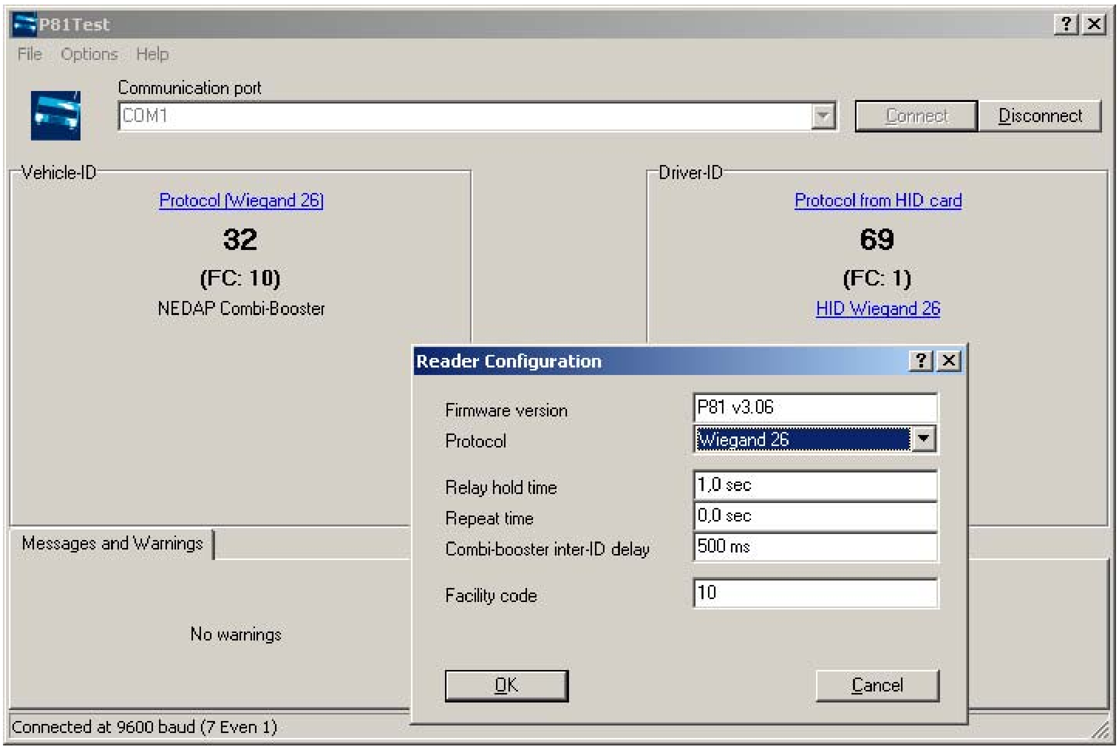
RS-232 cable

2.3 STEP 3 – SOFTWARE CONFIGURATION

The third step is setting up the software configuration options. This step is optional unless your application requires to deviate from the default settings.

2.3.1 P81TEST

The easiest way to configure the software options is by means of the P81Ttest software application, which can be downloaded from http://www.nedapavi.com. This user friendly application is developped to easily configure, debug and monitor your TRANSIT reader.



2.3.2 SETUP MENU

An alternative way to setup the software configuration is described in chapter 6. SETUP MENU.

3 DIP SWITCH SETTINGS

The TRANSIT (PS-270) has 8 DIP-switches, which are used by the P81 firmware as described below. Refer to appendix A when locating the DIP-switches.

ANTENNA SOURCE	8	7	6	5	4	3	2	1
Use default antenna	ON	x	x	x	x	x	x	x
Microwave and low frequency antenna	OFF	x	x	x	x	x	x	x

Set DIP-switch 8 only to OFF when using a TRANSIT reader with an low frequency (120kHz) antenna connected. The firmware then tries to identify transponders on both antennas (microwave and low frequency). When on one antenna a valid transponder is identified it sticks to that antenna source and does not identify anymore on the other antenna. So, when a vehicle is identified with the microwave antenna and this vehicle stays in front of the TRANSIT, nobody is identified at the 120kHz antenna.

FRAMELENGTH	8	7	6	5	4	3	2	1
Framelength 128 bit	x	ON	x	x	x	x	x	x
Framelength 64 bit	x	OFF	x	x	x	x	x	x

Framelength 128 bit is required to identify combi-boosters and EF-coded (RW-80) transponders. Set DIP-switch 7 to OFF when no combi-booster or EF-coded transponders are to be identified. This increases the detection speed. Framelength is only changed after a reset.

NON_NEDAP DECODING	8	7	6	5	4	3	2	1
Only Nedap tags enabled	x	x	ON	x	x	x	x	x
Nedap and non-Nedap tags enabled	x	x	OFF	x	x	x	x	x

Non-Nedap decoding enables the TRANSIT to identify non-Nedap transponders such as EM4102 based transponders and HID prox cards. Set DIP-switch 6 to OFF when non-Nedap transponders are to be identified.

BAUDRATE	8	7	6	5	4	3	2	1
Baudrate 9600	x	x	x	ON	ON	x	x	x
Baudrate 1200	x	x	x	ON	OFF	x	x	x
Baudrate 19200	x	x	x	OFF	ON	x	x	x
Baudrate 38400	x	x	x	OFF	OFF	x	x	x

The baudrate DIP-switches select the asynchronous communication speed. Default baudrate is 9600. RS-232 specifies communication speeds up to 20kbaud to have a maximum cable length of 15 meters or the cable length equal to a capacitance of 2500pF.
The baudrate is only changed after a reset.

COMMUNICATION PROTOCOL	8	7	6	5	4	3	2	1
CR/LF 7E1, Barcode Code39	x	x	x	x	x	ON	ON	ON
CR/LF 8N1, Magstripe ISO 7811/2	x	x	x	x	x	ON	ON	OFF
CR/LF 7E1, Wiegand 26 (H10301)	x	x	x	x	x	ON	OFF	ON
CR/LF 8N1, Wiegand 32	x	x	x	x	x	ON	OFF	OFF
CR/LF 7E1, Wiegand 37 (H10302 / H10304)	x	x	x	x	x	OFF	ON	ON
CR/LF 8N1, FF-56	x	x	x	x	x	OFF	ON	OFF
CR/LF 7E1, HID Corporate 1000	x	x	x	x	x	OFF	OFF	ON

The selected synchronous protocol is only changed after a reset.
DIP-switch 1 also selects the data format for the CR/LF protocol on the asynchronous interface. Where 7E1 means 7 data bits, even parity and 1 stop bit and 8N1 means 8 data bits, no parity and 1 stop bit.

For Wiegand 37 use command message SVF to select whether the facility code is used or not (see page 14).

4 LED INDICATORS

A number of LED's are used by the P81 firmware to indicate the current status. The table below describes the function of each LED. Refer to appendix A when locating the LED's.

LED	Description
STS	Status LED. Slow blinking: System's heartbeat (0.8 sec on / 0.8 sec off). Indicates that the power is on and the processor is running. Fast blinking: Bootloader say's hello. Only indicated for a short period directly after a reset. Twice blinking: Configuration menu active. Off: Abnormal situation. Should never not be off for longer than 1 second.
ID	Identification LED. This green LED starts to blink fast when a valid transponder is identified. The LED stays off when no (valid) transponder is identified.
UL	Unlock LED. The unlock LED is normally off and goes on when a valid transponder is identified. The LED is turned off when no transponder is identified anymore and the relay-hold-time has elapsed. This LED can be connected to a Reflex or DC130 antenna. There is also a relay contact present which has the same function.
NA	Lock LED. Red LED indicating system standby. This LED is normally on and goes off when the unlock LED goes on. This LED can be connected to a Reflex of DC130 antenna.
INP / DOOR	Input status LED This red LED is on when the input contact is closed. The input is not used in the P81 firmware.

Table 1: LED indicators

5 APPLICATION INFORMATION

The P81 firmware supports various non-proprietary synchronous communication protocols. The asynchronous CR/LF protocol is used to monitor, debug and setup the reader.

5.1 CR/LF PROTOCOL

CR/LF protocol is a simple ASCII based communications protocol. No software handshaking is required. Retries or retransmissions are not supported. The CR/LF protocol is used in combination with any of the non-proprietary synchronous protocols.
This chapter describes the application layer of the CR/LF protocol as it is implemented in the P81 firmware.

5.1.1 DATA FORMAT

Baudrate:	9600(default), 1200, 19200 or 38400. Setup with DIP-switches (see chapter 2).
Data bits:	7(default) or 8. Setup with DIP-switches (see chapter 2).
Parity:	even (default) or none. Setup with DIP-switches (see chapter 2).
Stop bits:	1.

5.1.2 MESSAGE FORMAT

The CR/LF protocol supports 7 bit ASCII data communication.

The ASCII control characters are reserved for message handling. The remaining characters (in the range from 20 hex to 7F hex) are valid characters for the data.

Every CR/LF message is terminated with a carriage-return and linefeed character:

`<data> CR LF`

Where:	`<data>`	Any readable ASCII character (in the range from hex 0x20 – 0x7F).
	CR	Carriage return character (hex 0x0D).
	LF	Linefeed character (hex 0x0A).

5.1.3 EVENT MESSAGES

The reader may send the following event messages. Spaces are only added for readability.

N Transponder identified (6 digit CF/DF/GF-code)

Syntax:	`N nnnnnn` CRLF	
Where:	`nnnnnn`	Transponder number in range from 000001 to 999999.

U Transponder identified (80 bit EF-code)

Syntax:	`U xxxxxxxxxxxxxxxxxxxx` CRLF	
Where:	`xxx...xxx`	Identification number 80 bit hexadecimal.
	x	Hexadecimal character made out of 4 bits (nibble) added with the value of character '0'.

U Combi-booster identified

Syntax:	`U 0000aaaaaa bbbbbbbbbb` CRLF	
Where:	`aaaaaa`	Combi-booster identification number in range from 1 to 999999.
	`bbbbbbbbbb`	Card identification number. Can be hexadecimal if a HID or EM4102 card is used.
Notes:	When no card is placed in the combi-booster the second identification number is left blank (filled with zeros).	

`U` EM4102 transponder identified
Syntax: `U 0000000000xxxxxxxxxx` CRLF
Where: `xx..xx` Identification number 40 bit hexadecimal.
`x` Hexadecimal character made out of 4 bits (nibble) added with the value of character '0'.

`O` Reader restart
Syntax: `O` CRLF

`P` Reader reset (all EEPROM settings are restored to their factory defaults)
Syntax: `P` CRLF

Examples:

Heavy-duty tag identified (id no. 000123).
`N000123`CRLF

Window tag identified (id no. 666666).
`N666666`CRLF

Combi-booster (id no. 123456) identified without a card.
`U00001234560000000000`CRLF

Combi-booster (id no. 123456) identified with a NEDAP card (id no. 708).
`U00001234560000000708`CRLF

Combi-booster (id no. 123456) identified with an EM4102 card (id no. 01F7A9C2E5).
`U000012345601F7A9C2E5`CRLF

Combi-booster (id no. 123456) identified with a HID prox card (H10301 format).
`U0000123456000003B207`CRLF

An EF-coded window tag identified (id no. 0000019AAD000123FC92).
`U0000019AAD000123FC92`CRLF

5.1.4 COMMAND MESSAGES

The following command messages may be sent to the reader. Protocol dependent characters are not shown here.

`CLS` **Reset all EEPROM settings and restart**

Description:	Restart the reader and reset all EEPROM settings to their factory defaults. The reader will generate a P-event (reader reset).
Syntax:	`CLS`
Reply:	`CLS`

`QVE` **Request firmware version**

Syntax:	`QVE`	
Reply:	`QVEpppvvv`	
Where:	`ppp`	Firmware name (P81).
	`vvv`	Firmware Version (300 = version 3.00).

`SFC` **Set facility code**

Description:	Set facility code for the Wiegand 26, Wiegand 32 and FF56 protocols.	
Syntax:	`SFCffff`	
Reply:	`SFCffff`	
Where:	`ffff`	Facility code in the range from hex 0000 to hex FFFF. Most significant byte first.
Notes:	For Wiegand 26 and FF56 only the least significant byte of the facility code is used.	

`QFC` **Request facility code**

Description:	See command message SFC above.	
Syntax:	`QFC`	
Reply:	`QFCffff`	
Where:	`ffff`	Facility code in the range from hex 0000 to hex FFFF. Most significant byte first.

`SC1` **Set timer 1 (relay hold time)**

Description:	The relay hold time (also referred to as the unlock-time) is default 1 second. It causes the unlock relay to stay activated for the specified time after the transponder could not be identified anymore. See timing diagram in Figure 1. When during the relay hold time the same transponder is identified again the reader will not generate a new detection event. Changed timer values are stored in EEPROM and are only lost when a 'reset reader' command is performed.	
Syntax:	`SC1xx`	
Reply:	`SC1xx`	
Where:	xx	Relay hold time in the range from 1 to 255 tenths of a second. Specify timer value hexadecimal. E.g. 1 second is hex 0A.
Example:	`SC11E`	Set the relay hold time to 3 seconds.

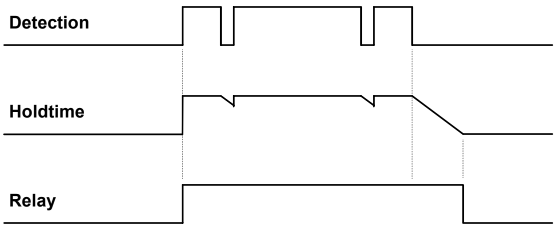


Figure 1: Timing diagram relay-hold-time

`QC1` Request timer 1 (relay hold time)

Description:	See command message SC1 above.	
Syntax:	`QC1`	
Reply:	`QC1xx`	
Where:	xx	Relay hold time in the range from 1 to 255 tenths of a second.

`SC2` Set timer 2 (not used, reserved for alarm-time)

Syntax:	`SC2xx`	
Reply:	`SC2xx`	
Where:	xx	Alarm time in the range from 0 to 255 tenths of a second. Specify timer value hexadecimal. E.g. 1 second is hex 0A.

`QC2` Request timer 2 (not used, reserved for alarm-time)

Description:	See command message SC2 above.	
Syntax:	`QC2`	
Reply:	`QC2xx`	
Where:	xx	Alarm time in the range from 0 to 255 tenths of a second.

`SC3` Set timer 3 (not used, reserved for blocking-time)

Syntax:	`SC3xx`	
Reply:	`SC3xx`	
Where:	xx	Blocking time in the range from 0 to 255 tenths of a second. Specify timer value hexadecimal. E.g. 1 second is hex 0A.

`QC3` Request timer 3 (not used, reserved for blocking-time)

Description:	See command message SC3 above.	
Syntax:	`QC3`	
Reply:	`QC3xx`	
Where:	xx	Blocking time in the range from 0 to 255 tenths of a second.

`SC4` Set timer 4 (repeat-time)

Description:	The repeat time is default 0 seconds, which means that the detection event is only sent once. The repeat time causes the reader to transmit a detection event every 'repeat time' seconds for as long as the transponder is present. See timing diagram in Figure 2. Changed timer values are stored in EEPROM and are only lost when a 'reset reader' command is performed. The repeat timer works for every communication protocol.	
Syntax:	`SC4xx`	
Reply:	`SC4xx`	
Where:	xx	Repeat time in the range from 0 to 255 tenths of a second. Specify timer value hexadecimal. E.g. 1 second is hex 0A.
Example:	`SC400`	Disable the detection event message repeat option.

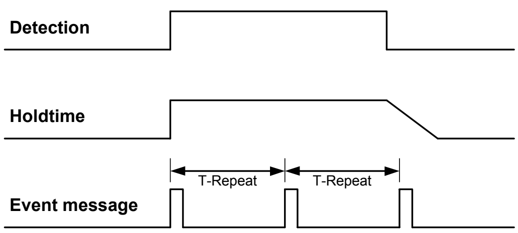


Figure 2: Timing diagram repeat time (T-repeat)

`QC4` Request timer 4 (repeat-time)

Description:	See command message SC4 above.	
Syntax:	`QC4`	
Reply:	`QC4xx`	
Where:	xx	Repeat time in the range from 0 to 255 tenths of a second.

SC5 Set timer 5 (combi-booster inter id delay)

Description: Combi-boosters combine a vehicle id with a driver-id. Both id-numbers may be sent in a separate event message to the host computer. The delay between the two messages can be setup with this timer. First is always the vehicle-id transmitted. The second event message, which contains the driver-id, can be disabled if this timer is set to zero. The vehicle-id message can be disabled if the timer is set to 255 (hex FF).
The second event message is only available for the synchronous communication protocols ISO 7811/2, Wiegand 26/32/37 and HID Corporate 1000. The CR/LF and Code39 protocol send the vehicle-id and driver-id in one event message. See also the timing diagram in Figure 3 below.
The FF-56 bit protocol does not support the combi-booster's driver-id.
Changed timer values are stored in EEPROM and are only lost when a 'reset reader' command is performed.

Syntax: `SC5xx`

Reply: `SC5xx`

Where: xx Combi-booster inter-id delay in the range from 0 to 255 hundredths of a second. Specify timer value hexadecimal. E.g. 100 msec is hex 0A.

Notes: When to the total amount of time required to sent both id numbers is greater than the repeat time, see command message SC4, then the second message containing the vehicle-id follows directly after the first message with the driver-id. See also 'Events D' in Figure 3.

Example: `SC500` Disable the synchronous event message containing the combi-booster's driver-id. See 'Events A' in Figure 3.

Example: `SC50A` Enable both synchronous event messages. First send the vehicle-id, wait for 100 msec and secondly send the driver-id. See 'Events B' in Figure 3.

Example: `SC5FF` Disable the synchronous event message containing the combi-booster's vehicle-id. See 'Events C' in Figure 3.

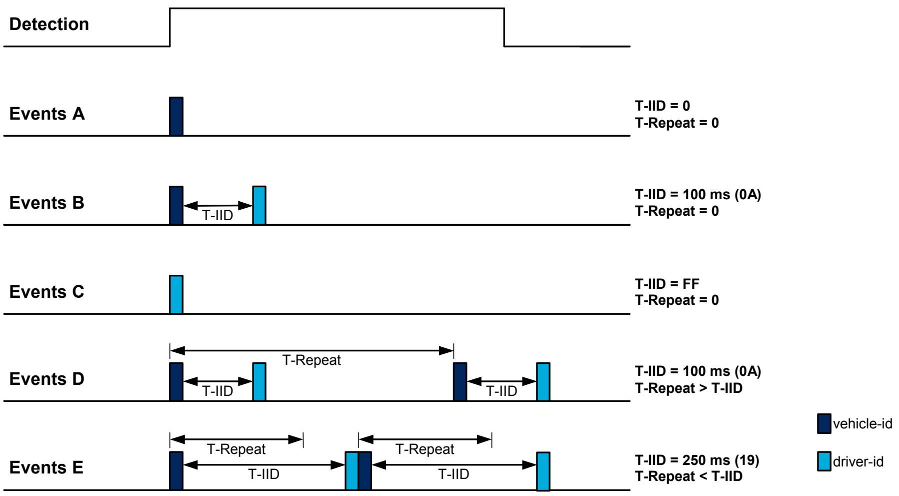


Figure 3: Timing diagram combi-booster inter id delay (T-IID)

QC5 Request timer 5 (combi-booster inter id delay)

Description: See command message SC5 above.

Syntax: `QC5`

Reply: `QC5xx`

Where: xx Combi-booster inter-id delay in the range from 0 to 255 hundredths of a second (0=disable driver-id, FF=disable vehicle-id)

SVF Set various flags

Description: Set flag to select Wiegand 37 / Wiegand 37 with facility code.
Syntax: `SVFxx`
Reply: `SVFxx`
Where: `xx` Bit-coded byte in hexadecimal notation.

VAR FLAGS	7	6	5	4	3	2	1	0
Wiegand 37 without facility code (H10302)								0
Wiegand 37 with facility code (H10304)								1

QVF Request various flags

Description: See command message SVF above.
Syntax: `QVF`
Reply: `QVFxx`
Where: `xx` Bit-coded byte in hexadecimal notation.

5.2 BARCODE CODE39 PROTOCOL

Code39 is commonly used for various barcoding labels such as name badges, inventory and industrial applications. The Code39 barcode is the easiest to use of alpha-numeric barcodes and is designed for character self-checking, eliminating the requirement for check character calculations. The 3 of 9 Code and Code 3 of 9 are just other names for the same symbology.

Each character consists of 9 elements, 5 bars and 4 spaces. Three elements are wide and six elements are small. The ratio between wide and small are 3:1. A bar corresponds with a high level (5V) on the TTL-output, a space with a low-level (0V) on the TTL-output.
The messages contain a start and stop character. The characters are separated by an intercharacter gap. The intercharacter gap is a small space.

5.2.1 CONNECTIONS

The following TTL-level outputs are used. Refer to appendix A when locating these connections.

- BC39 Code39 data `0-3`
- GND Signal ground `GND`

5.2.2 MESSAGE FORMAT

The reader sends a message when a transponder is identified. The transponder may be identified by the microwave or by the low frequency (120kHz) antenna.

Message format for CF, DF or GF-coded (6 digit) transponders.

Syntax: `*NNNNNN*`

Where:

`*`	Start/stop character.
`NNNNNN`	Identification number in the range from 000001 to 999999.
`N`	Decimal character (0-9).
`*`	Start/stop character.

Message format for combi-boosters, EF-coded transponders and non-Nedap transponders.
Syntax: `*XXXXXXXXXXXXXXXXXXXX*`

Where:

`*`	Start/stop character.
`XXX...XXX`	Identification number 80 bit hexadecimal (20 characters).
`X`	Hexadecimal character (0-9, A-F).
`*`	Start/stop character.

Optionally this message may be repeated if the transponder stays within the reading range. Setup the repeat interval with command message SC4 on the asynchronous communication interface (see page 12).

Examples

Window tag identified (id no. 666666).

`*666666*`

Combi-booster (id no. 123456) identified without a card.

`*0000123456000000000*`

Combi-booster (id no. 123456) identified with an EM4102 card (id no. 01F7A9C2E5).

`*000012345601F7A9C2E5*`

Combi-booster (id no. 123456) identified with a HID prox card (H10301 format).

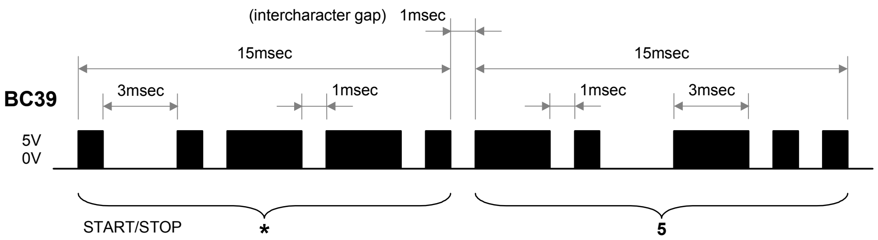
`*0000123456000003B207*`

An EF-coded window tag identified (id no. 0000019AAD000123FC92).

`*0000019AAD000123FC92*`

5.2.3 PROTOCOL TIMING

In the figure below the timing for two characters is specified including the intercharacter gap.



Timing constants:

Small element	1msec
Wide element	3msec
Intercharacter gap	1msec

Figure 4: Code39 protocol timing

5.2.4 CHARACTER SET

The symbology of the Code 39 character set consists of bar code symbols representing characters 0-9, A-Z, the space character and the following symbols: - , . , $, / , + , %. For transmitting id-numbers only the characters 0-9 and A-F are used.

Description	Code39	Description	Code39
0		1	
2		3	
4		5	
6		7	
8		9	
A		B	
C		D	
E		F	
* START/ STOP			

Table 2: Code39 character subset

5.3 MAGSTRIPE ISO 7811/2

The ISO 7811/2 protocol is implemented according to the standard ISO 7811/2 track 2 with ABA format, which is often used in magnetic stripe readers.

The data information of the message is included between start-sentinel and end-sentinel. After the end-sentinel the LRC is sent (Longitudinal Redundancy Check) to check the received data. The contained data is in BCD notation.

5.3.1 CONNECTIONS

The following TTL-level outputs, which are active low, are used. Refer to appendix A when locating these connections.

- /CLS Card Loaded Signal `0-1`
- /RCP Reader Clock Signal `0-2`
- /RDP Reader Data Signal `0-3`
- GND Signal ground `GND`

5.3.2 MESSAGE FORMAT

The reader sends a message when a transponder is identified. The transponder may be identified by the microwave or by the low frequency (120kHz) antenna.

ISO 7811/2 message format (spaces are only added for readability).

Syntax: `<ssen> ccc ssss nnnnnn <esen> <lrc>`

Where:	`<ssen>`	Start sentinel (hex B).
	`ccc`	Customer code (fixed 000).
	`ssss`	Status code (fixed 0000).
	`nnnnnn`	Identification number in the range from 000001 to 999999.
	`<esen>`	End sentinel (hex F).
	`<lrc>`	Longitudinal redundancy check.

Optionally this message may be repeated if the transponder stays within the reading range. Setup the repeat interval with command message SC4 on the asynchronous communication interface (see page 12).
When using combi-boosters an optional second event message containing the driver-id can be enabled with command message SC5 (see page 13).
For each character that is transmitted an odd parity bit is calculated. Therefore each character contains 5 bits. The least significant bit is transmitted first, finally the parity bit is transmitted. After the end-sentinel the LRC is transmitted.

Examples

Window tag identified (id no. 666666).

`B0000000666666F4`

Combi-booster (id no. 123456) identified without a card.

`B0000000123456F3`	First message contains vehicle-id.
`[B0000000000000F4]`	Optional second message with empty driver-id.

Combi-booster (id no. 123456) identified with a NEDAP XS-card (id no. 004444).

`B0000000123456F3`	First message contains vehicle-id.
`[B0000000004444F4]`	Optional second message with driver-id.

Combi-booster (id no. 123456) identified with a HID prox card.

`B0000000123456F3`	First message contains vehicle-id.
`[B0000000000000F4]`	Optional second message with empty driver-id.

An EF-coded window tag identified (id no. 0000019AAD000123FC92). Hex FC92 = 64658.

`B0000000064658FD`

Notes

- *EF-coded transponders contain 80 bits of data which does not fit in the message format. When an EF-coded transponder is identified only the least significant 16 bits are used. This 16 bit binary value is converted to a BCD value in the range from 0 to 65535. This BCD number is transmitted in the ISO 7811/2 message. The message on the asynchronous interface (CR/LF protocol) will contain the complete 80 bits of transponder data. EF identification can be completely disabled with DIP-switch 7. See chapter 2.*
- *EM4102 based transponders contain 40 bits of data which does not fit in the message format. When an EM4102 based transponder is identified only the least significant 16 bits are used. This 16 bit binary value is converted to a BCD value in the range from 0 to 65535. This BCD number is transmitted in the ISO 7811/2 message. The message on the asynchronous interface (CR/LF protocol) will contain the complete 40 bits of transponder data. EM4102 identification can be completely disabled with DIP-switch 6. See chapter 2.*
- *HID prox cards on combi-boosters are not supported in this protocol. The optional second message with driver-id is filled with zeros as it would be if there was no card at all.*

5.3.3 LRC CALCULATION

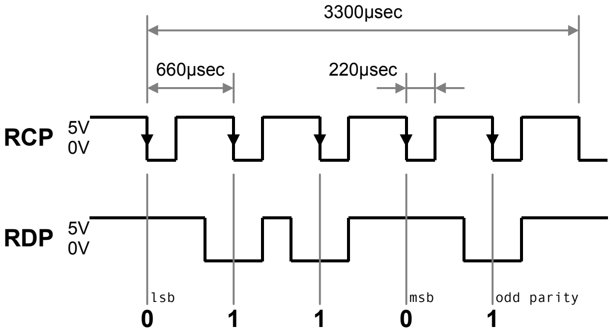
The LRC is the vertical even parity over all data bits including start-, and end-sentinel. In the example below the LRC is calculated on a data message 0000000123456:

HEX	PAR	BIN	Description
B	0	1011	start-sentinel
0	1	0000	data \
0	1	0000	data > customer code (always 000)
0	1	0000	data /
0	1	0000	data \
0	1	0000	data \ status code (always 0000)
0	1	0000	data /
0	1	0000	data /
1	0	0001	data \
2	0	0010	data \
3	1	0011	data \ id-number
4	0	0100	data / (123456)
5	1	0101	data /
6	1	0110	data /
F	1	1111	end-sentinel
3	1	0011	lrc

LRC = (B ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 0 ⊕ 1 ⊕ 2 ⊕ 3 ⊕ 4 ⊕ 5 ⊕ 6 ⊕ F) = 3.

5.3.4 PROTOCOL TIMING

In the figure below the timing for one character is specified. Each bit consists out of one period low (220µsec) and two periods high (440µsec). The bit times have an accuracy of 10 percent. The data-signal RDP is valid and stable on the falling edge of the clock-signal RCP.



Timing constants:

Card present period	74msec
Clock period	660µsec
Clock high	440µsec
Clock low	220µsec
Data preamble	11msec
Data postamble	11msec

Figure 5: ISO 7811/2 protocol timing one character

The card present signal will be active (=low) for about 74 milliseconds. Before and after the data 16 clock pulses are generated, resulting in $2 \times 16 \times 660\mu sec \approx 22$ msec delay.

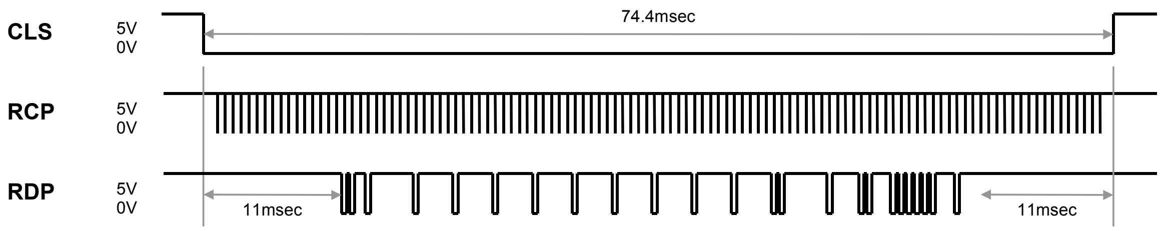


Figure 6: ISO 7811/2 protocol timing complete message

5.3.5 CHARACTER SET

HEX	BIN	Description
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	Reserved for hardware control purposes
B	1011	Start-sentinel
C	1100	Reserved
D	1101	Separator
E	1110	Reserved for hardware control purposes
F	1111	End-sentinel

Table 3: ISO 7811 track 2 character set

5.4 WIEGAND 26/32/37 BIT PROTOCOL

The Wiegand protocol is based upon the Wiegand-effect card readers widely used in access control applications. Wiegand cards incorporate two rows of tiny wire particles representing binary zeros and ones. The Wiegand interface uses two TTL-level outputs that correspond to these zeros and ones.

5.4.1 CONNECTIONS

The following TTL-level outputs are used. Refer to appendix A when locating these connections.

- Data-0 Zeros (active low) `0-2` (Green)
- Data-1 Ones (active low) `0-3` (White)
- GND Signal ground `GND` (Black)

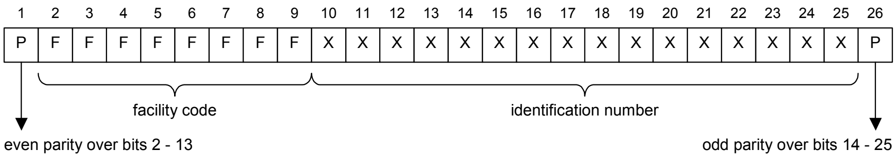
5.4.2 MESSAGE FORMAT

The reader sends a message when a transponder is identified. The transponder may be identified by the microwave or by the low frequency (120kHz) antenna.

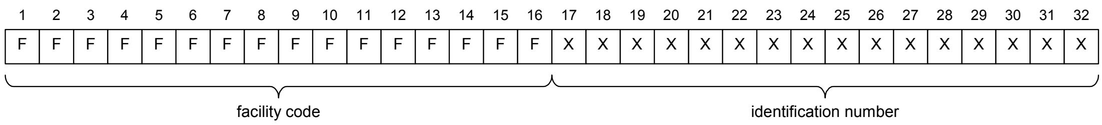
Optionally this message may be repeated if the transponder stays within the reading range. Setup the repeat interval with command message SC4 on the asynchronous communication interface (see page 12).
When using combi-boosters an optional second event message containing the driver-id can be enabled with command message SC5 (see page 13).
NEDAP cards do not contain a facility code. The P81 firmware will add the facility code to the Wiegand messages. Setup this facility code with command message SFC (see page 11).

The message format for HID prox cards is preserved including the HID prox card's facility code. This means that the second Wiegand event message, with the driver-id, may be transmitted in a different format than selected by the DIP switches. Even OEM proprietary formats can be expected. The maximum length for these HID prox cards is 40 bits. For longer formats only the least significant 40 bits are transmitted.

Wiegand 26 bit message format (H10301)
Wiegand 26 is the industry standard format. It consists of a parity bit, followed by 8 facility code bits, followed by 16 bits identification number and finally another parity bit. The facility code and id-number are transmitted with the most significant bit first. The total number of bits is 1+8+16+1 = 26.
The first parity bit is even calculated over the succeeding 12 bits. The last parity bit is odd calculated over preceding 12 bits.



Wiegand 32 bit message format
The Wiegand 32 bit format consists of 16 bits facility code, followed by another 16 bits identification number. The facility code and id-number are transmitted with the most significant bit first. The total number of bits is 32. There are no parity bits.



Notes for Wiegand 26/32

- *The id-number for CF, DF or GF-coded (6 digit) transponders is converted to a binary value. The message will only contain the 16 least significant bits. E.g. maximum id-number is 65535.*

- *EF-coded transponders and EM4102 based transponders contain respectively 80 and 40 bits of data. This does not fit in the Wiegand 26/32 bit message formats. The Wiegand message will only contain the 16 least significant bits.*

Wiegand 37 bit message format (H10302)
Wiegand 37 was developped to provide the industry with an open standard format that supports large id-numbers. The Wiegand 37 bit format consists of a parity bit, followed by 35 bits identification number and finally another parity bit. The id-number is transmitted with the most significant bit first. The total number of bits is 1+35+1 = 37.
The first parity bit is even calculated over the succeeding 18 bits. The last parity bit is odd calculated over preceding 18 bits. Bit 19 is used in calculating both parity bits.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
P	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	P

even parity over bits 2 - 19 — identification number — odd parity over bits 19 - 36

Wiegand 37 bit with facility code message format (H10304)
Wiegand 37 with facility code differs only from Wiegand 37 in that it has a facility code. The Wiegand 37 bit with facility code format contains besides the parity bits, 16 facility code bits and a 19 bits id-number.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
P	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	P

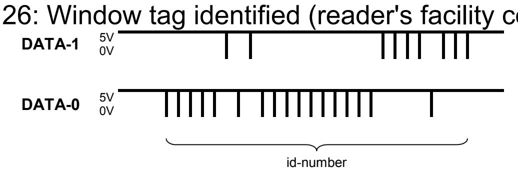
even parity over bits 2 - 19 — facility code — identification number — odd parity over bits 19 - 36

Notes for Wiegand 37 / Wiegand 37 with facility code

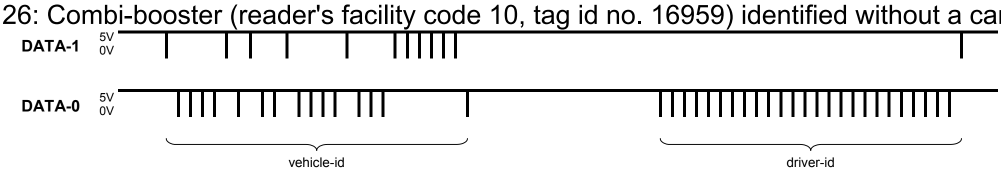
- *EF-coded transponders and EM4102 based transponders contain respectively 80 and 40 bits of data. This does not fit in the Wiegand 37 bit message format.*
  The Wiegand 37 message will only contain the 35 least significant bits.
  The Wiegand 37 with facility code message will only contain the 19 least significant bits.

Examples

Wiegand 26: Window tag identified (reader's facility code 10, tag id no. 123).



Wiegand 26: Combi-booster (reader's facility code 10, tag id no. 16959) identified without a card.



5.4.3 PROTOCOL TIMING

In the figure below the Wiegand protocol timing is specified.

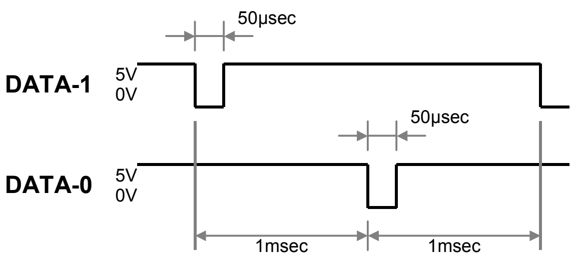


Figure 7: Wiegand protocol timing

Timing constants:

Bit period	1msec
Data low	50µsec

5.5 FF-56 BIT PROTOCOL

The FF56 protocol is a clock and data protocol, which is used in Fritz Fuss access control systems. The data is valid and stable on the rising edge of the clock signal.

5.5.1 CONNECTIONS

The following TTL-level outputs are used. Refer to appendix A when locating these connections.

- FF56C Clock (active high) `O-2`
- /FF56D Data (active low) `O-3`
- GND Signal ground `GND`

5.5.2 MESSAGE FORMAT

The reader sends a message when a transponder is identified. The transponder may be identified by the microwave or by the inductive antenna. Optionally this message may be repeated if the transponder stays within the reading range. Setup the repeat interval with command message SC4 on the asynchronous communication interface (see page 12).
Setup the facility code with command message SFC on the asynchronous interface.

FF-56 bit message format

Syntax: `FFFFFFFFSSSSPSSSSPSSSSPSSSSPNNNNPNNNNPNNNNPNNNNPVVVVPDDD`

Where:

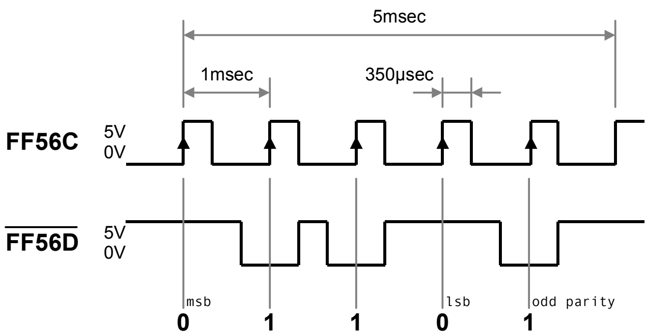
F	Reader facility code (8 bit binary value, most significant bit first).
S	System number (fixed BCD 0000).
N	Identification number in the range from 0001 to 9999.
V	Version (fixed BCD 0).
P	Odd parity over preceding 4 bits.
D	Dummy bits (3 bits binary, fixed bin 001).

Notes

- *Combi-boosters contain a vehicle-id and optionally a driver-id. The FF-56 bit message will only contain the vehicle-id. Maximum vehicle-id number is 9999.*
- *If the id-number exceeds the range from 0001 to 9999 then there is no FF-56 bit message.*
- *When an EF-coded transponder or EM4102 based transponder is identified there will be no FF-56 bit message.*

5.5.3 PROTOCOL TIMING

In the figure below the FF-56 bit protocol timing for one BCD digit including parity bit is shown.



Timing constants:

Clock period	1msec
Clock high	350µsec
Clock low	650µsec

Figure 8: FF-56 bit protocol timing

© Nedap IDEAS, P.O. Box 103, NL-7140 AC GROENLO

5.6 HID CORPORATE 1000 PROTOCOL

The HID Corporate 1000 format is a 35 bit wiegand format with a unique Company ID Code and over 1,000,000 card numbers available for use.

5.6.1 CONNECTIONS

The connections are descibed in chapter 5.4.1.

5.6.2 MESSAGE FORMAT

HID can provide you with a document that details the generic Corporate 1000 Format. You may obtain this document by calling HID at (800) 237-7769 and asking for Technical Support. Technical Support can also answer specific questions that you may have about the Corporate 1000 program.

The reader will send a message when a transponder is identified. The transponder may be identified by the microwave or by the low frequency (120kHz) antenna.

Optionally this message may be repeated if the transponder stays within the reading range. Setup the repeat interval with command message SC4 on the asynchronous communication interface (see page 12).
When using combi-boosters an optional second event message containing the driver-id can be enabled with command message SC5 (see page 13).
NEDAP cards do not contain a facility code. The P81 firmware will add the facility code to the Corporate 1000 messages. Setup this facility code with command message SFC (see page 11).

The message format for HID prox cards is preserved including the HID prox card's facility code. This means that the second event message, with the driver-id, may be transmitted in a different format than selected by the DIP switches. Even OEM proprietary formats can be expected. The maximum length for these HID prox cards is 40 bits. For longer formats only the least significant 40 bits are transmitted.

Notes

- *EF-coded transponders and EM4102 based transponders contain respectively 80 and 40 bits of data. This does not fit in the Corporate 1000 message format. Only the 20 least significant bits are used.*

5.6.3 PROTOCOL TIMING

The protocol timing is described in chapter 5.4.3.

6 SETUP MENU

The complete software configuration of the TRANSIT can be changed by means of the setup menu.

6.1.1 ENTER SETUP MENU

To reach the setup-menu through the serial connection follow these steps:

1. Connect a console terminal or PC running a terminal emulation program to the reader's serial port. The port settings are 9600 baud, 8 data bits, no parity, 1 stop bit, no flow control (the port settings are not changed by the DIP-switches).
2. **Power off the reader**.
3. **Hold down the x key** on the terminal (or emulation), because after power-up three lowercase x characters (xxx) must be seen within 1 second to enter the setup-menu.
4. **Power up the reader** and wait (while still holding the x key) for the message

```
Firmware version: NEDAP P81xxx
Press Enter to go into Setup Mode
```

to appear on your terminal.

5. To enter the setup-menu, you must press Enter within 5 seconds. The setup-menu will appear.

```
Setup Menu:
1. View/Edit settings
7. Factory defaults
8. Exit without save
9. Save & exit setup
Your choice ?
```

Note: While you're in the setup menu the TRANSIT does not read any tags. The status LED periodically flashes twice to indicate that the setup menu is still active.

6.1.2 VIEW/EDIT SETTINGS

Select '1. View/Edit settings' and go through all available settings. To enter a value for a setting, type the value and press Enter, or to leave the current value unchanged, just press Enter.

```
Relay hold time (0A hex) ?
```
Relay hold time in tenths of a second. 0A hex = 10*0.1sec = 1sec. See SC1 on page 11.

```
Repeat time (00 hex) ?
```
Repeat time in tenths of a second. 00 hex = do not repeat (Send once). See SC4 on page 12.

```
Combi-booster inter-id delay (32 hex) ?
```
Delay on the wiegand interface between the two combi-booster events. Also used to disable one of them. Time in hundredths of a second. 32 hex = 50*0.01sec = 0.5sec. See SC5 on page 13.

```
Facility code (000A hex) ?
```
The reader's facility code. Used for cards that do not contain a facility code (such as Nedap cards). Enter in hexadecimal notation. See SFC on page 11.

```
Various flags (00 hex) ?
```
Bit-coded byte with various flags. 00 hex = no flags enabled. See for details SVF on page 14.

6.1.3 FACTORY DEFAULTS

Select '7. Factory defaults' to reset all settings to their factory defaults.

6.1.4 EXIT SETUP MENU

When you are finished make sure to exit the setup menu.
Select '8. Exit without save' to exit the setup menu without saving any changes you have made.
Select '9. Save & exit setup' to exit the setup menu and save any changes you have made.

7 FIRMWARE UPGRADING

The Microchip PIC16F876 is a single chip flash based microcontroller, which allows to upgrade the firmware by the asynchronous serial interface. The upgrading is performed by a freeware application called "PIC downloader" which downloads the firmware file to the microcontroller. The upgrade procedure is described below.

1. Start PICload.exe.
2. Select the RS232 port to which the TRANSIT is connected.
3. Select the firmware file (*.hex, *.ehx).
4. Enable 'download customer codes' when you want to load the customer codes from the hex-file into the TRANSIT.
   Note that the hex-files on www.nedap.net contain DEMO customer codes. Uncheck 'download customer codes' to prevent your customer codes inside the TRANSIT to be overwritten.
5. Click 'Download' to start downloading the firmware.
6. Once the downloading has completed the PIC downloader displays the message 'Download successfully completed' and the TRANSIT starts the upgraded firmware.

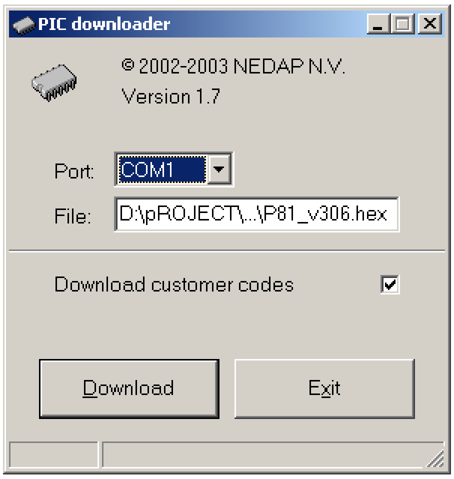


Figure 9: PIC downloader

Note1: Aborted downloads may cause the TRANSIT to stop functioning. In such cases repeat the upgrade procedure until it succeeds.

Note2: If the message 'Searching for bootloader' does not disappear check the cables and the com-port settings. Sometimes it may be required to reset the TRANSIT before the bootloader can be found. Before resetting the TRANSIT make sure that the message 'Searching for bootloader' is still shown in the statusbar.

8 FIRMWARE REVISION HISTORY

Below the P81 firmware modifications are listed. For information on how to obtain the latest release of the P81 firmware contact Nedap.

Version	Date	Notes/Bugs fixed
v3.06	2003-11-13	• Wiegand pulse width now 50µsec (was 100µsec before) • Bug fixed which was introduced in v3.05
v3.05	2003-10-28	• Added setup menu • Default value for inter-id delay now 0.5 sec (instead of 0 = only vehicle-id)
v3.04	2003-07-08	• Bug fixed: full duplex RS232 communication could cause failures • I^2C acknowledges now properly handled.
v3.03	2003-04-24	• Added Combi-booster HID support • Added Corporate 1000 protocol • Added commands SVF/QVF to select Wiegand 37 with facility code • Disable vehicle-id when inter id delay is set to hex FF
v3.02	2003-03-03	• Send ? $^{C}{}_{R}{}^{L}{}_{F}$ instead of NAK when unknown messages are received
v3.01	2003-01-17	• Bug fixed for wiegand 26-bit. odd parity over last 12 bits incorrect
v3.00	2002-10-11	• Initial release

Table 4: Revision history

A HARDWARE

The P61 firmware is developed for the TRANSIT (PS-270) microwave reader. Below an overview of the hardware components is shown. For more details about the connections and electrical specifications refer to the TRANSIT (PS-270) installation guide.

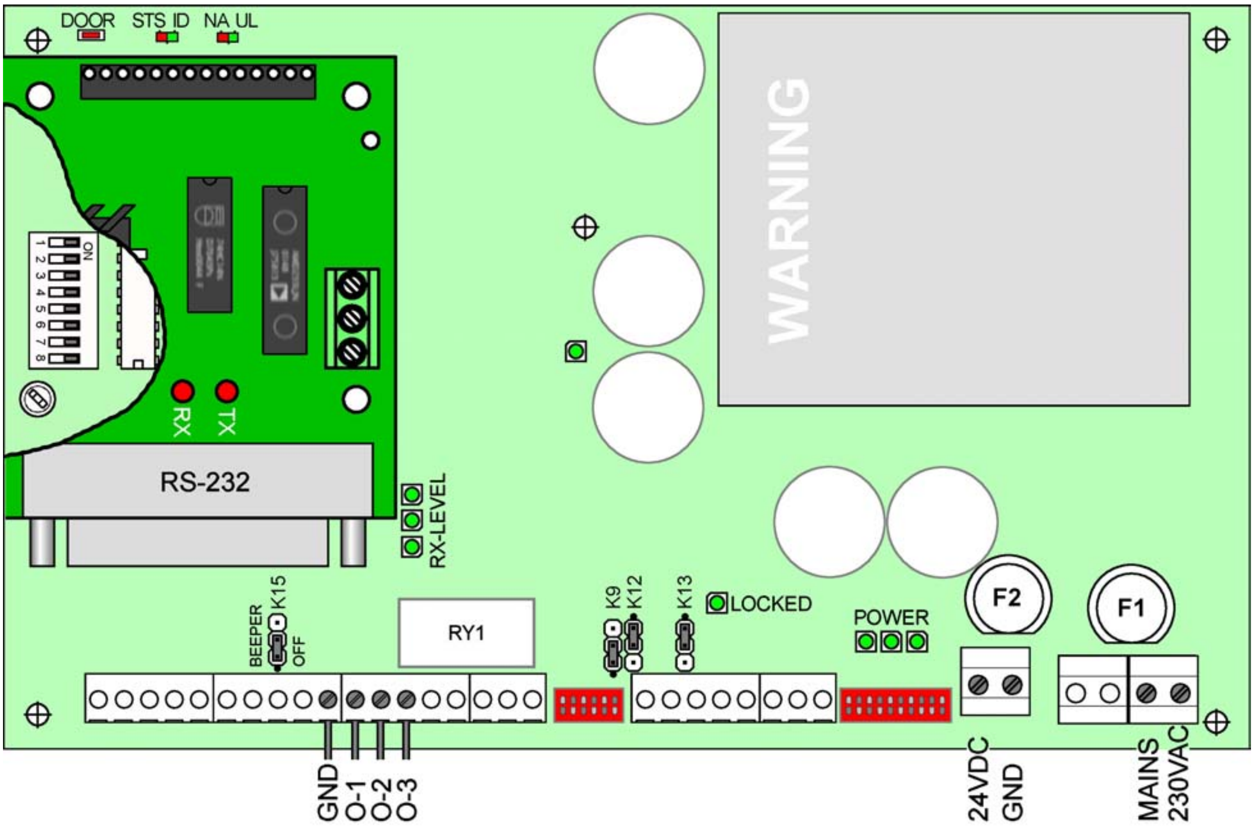


Figure 10: Overview PS-270 board

B ASCII TABLE

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	NUL	32	20	SP	64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	HT	41	29	)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SO	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	78	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL