Manual AP80
Display Controller

- 8 Digit display in DIN-enclosure 144 X 72 mm
- Supply voltage 10 – 35V DC
- All in- and outputs optically isolated
- Input for absolute SSI encoders, incremental encoders, parallel encoders and Start/Stop sensors
- CAN-bus, RS232, RS422/RS485
- 6 Digital inputs and 9 digital outputs
- 24 Programmable cams / 9 outputs
- 48 Programmable nominal values
- Dynamic cam-adjustment (cycletime 250µS)
- Programmable analog output (16 bit)
- Linearization function
- Datamodule for 24 in- and 24 outputs
- Datamodule for 32 (tri-state) outputs

Table of contents

1 Introduction

1.1 General

With the microcontroller based AP80 it is possible connect the following encoder types:

- Incremental with quadrature (two 90°) signals
- Incremental with impulssignal and directionsignal
- Absolute with SSI (Synchron Serial Interface)
- Absolute with start/stop (MTS Sensor)

It is standard equipped with 6 digital inputs and 9 digital outputs, all free programmable. The AP80 incorporates possibilities like CAN-bus, RS232 communication, RS422/485 communication, an analog output and a datamodule with 24 inputs and 24 outputs or 32 tri-state outputs.
All in- and outputs, including the communication-ports, are electrically isolated.

With the 24 programmable cams several functions can be realized like limit detection.

Another feature of the AP80 is the memory for the nominal values. The 48 programmable values can be used to determine the positions of the cams.

Up to 4 values can be read from external thumbwheel switches. These values can be used as positions for the cams or as preset values.

The microcontroller reads the actual sensor value and calculates the display-value and the actual velocity.
A programmable power failure protection makes sure that the actual value is stored in an EEPROM.
The AP80 is configurable for numerous applications by adjusting its parameters. One of the functions is an that the actual sensor value can be calculated with a linearization table for nonlinear movements.

The AP80 can be programmed with the DST80 PC-based software.

1.2 Important information

- ✓ The AP80 is a high-tech electronic product. To ensure safety and a correct functioning of the product it is important that only qualified specialists will install and operate the AP80.
- ✓ If through a failure or fault the AP80 an endangering of persons or damage to plant is possible, this must be prevented using additional safety measures. These must remain operational in all possible modes of the AP80.
- ✓ Necessary repairs to the AP80 are only to be carried out by the manufacturer.

1.3 EMC

To ensure the best possible electromagnetic compatibility, it is recommended to pay attention to shielding and grounding the AP80:

- ✓ Shielding on both sides and with the largest possible contact area.
- ✓ Keep wiring as short as possible.
- ✓ Earth-connections should be short and with the highest possible wiring-diameter.
- ✓ Signal-cables and supply-cables must be separated.
- ✓ The EMC-bracket type EMC-B01 should be used.

1.4 Definitions

1.4.1 Display units AWE

The display units, referred to as AWE, is the value shown on the display without regarding the decimal point. The decimal point is only used for the comfort of the operator, but has no functional meaning.
(display = 347.4 >> AWE = 3474)

1.4.2 Parameter number

A parameter number is always shown in the format P[xxx]. It is possible that a parameter number appears in more than one menu.

1.4.3 Notation

Values can be displayed in different notations like binary or hexadecimal. The character behind the value shows in which notation the value is represented:

100D	Decimal
238H	Hexadecimal
244G	Gray
10010011B	Binary

for example 220D = DCH = 11011100B

1.4.4 Edges

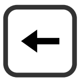
L→H	: rising edge (low to high)
H →L	: falling edge (high to low)

2 Operation

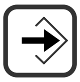
2.1 Key functions

[P] key
- Cycle through monitoring displays
- Activate programming mode (in combination with other keys)

[+1] key
- View type number



[Cursor] key
- View software version
- View custom software version (in combination with the [Enter] key)

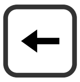


[Enter] key
- View status of inputs and outputs

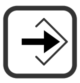
2.2 Key functions in programming mode

[P] key
- one step back in menu
- discontinue programming mode
- discontinue changing nominal values/parameters (edit mode)
- LED is on when programming mode is active

[+1] key
- cycle through menu
- increase nominal value- / parameter number
- increase digit (in edit mode)



[Cursor] key
- activate edit mode
- move one digit to the left (in edit mode)

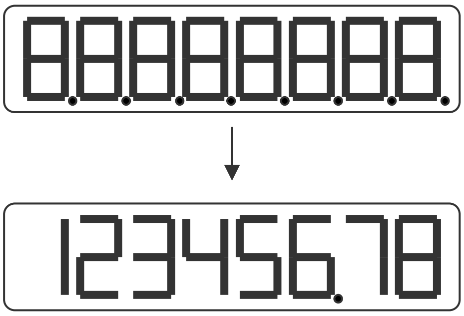


[Enter] key
- enter a submenu or parameter
- increase nominal value-/parameter number
- store a changed value
- clear value, hold down [Cursor] button (edit mode)

2.3 Display functions

2.3.1 Status functions

Power on



Appr. 1sec.

Operating (normal) mode

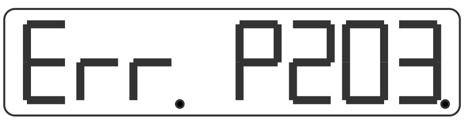
2.3.2 Error messages

There are two groups of errors:

- Parameter errors (error numbers 0…499, preceeded by a “P”)
- General error (error numbers from 500 and up)

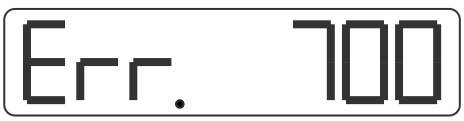
Example:

Parameter error number



Error parameter 203
(Parameter out of min/max value)

General error number



General error 700

2.3.3 Survey of error messages

<u>Error messages:</u>

000...499   Parameter error is displayed as PXXX on the display.

700 = Reference value P[003] >= Counting range P[004]
701 = Adjustment value SSI P[005] >= Counting range
or Adjustment SSI P[005] < Offset P[002] (only if adjustment type = 1,2, or 4)
702 = Adjustment value parallel P[005] >= Counting range
703 = Number of active SSI databits > number of SSI Clock pulses
704 = Transmitting and receiving with AP-Link not possible with the same adress
705 = Offset SSI P[002] > Counting range
708 = No Hold-Reset possible (only possible with incremental and internal frequency)
710 = Minus-sign and data ready on the same output on the data IO module
711 = Data input not possible together with parallel actual value (absolute encoder)
712 = Minus-sign and data valid on the same input on data IO module
713 = Number of active bits on the data IO module too large (minus-sign and or data valid are active)
716 = Umin >= Umax
717 = Imin >= Imax
720 = Changelock nominal values active
721 = Changelock parameters active
725 = Serial RS232 and RS485 both with ASCII at the same time not possible
732 = Function input-2 not valid (equal to input-1)
733 = Function input-3 not valid (equal to input-1...input-2)
734 = Function input-4 not valid (equal to input-1...input-3)
735 = Function input-5 not valid (equal to input-1...input-4)
736 = Function input-6 not valid (equal to input-1...input-5)

800 = SSI error delta-s
801 = SSI error cable failure

810 = Start/Stop error no magnet detected
811 = Start/Stop error time-out no stop signal detected

Error meldingen linearisatie (xx = P1 …P30)

9xx = Linearisatie Xn <= Xn-1
940 = Linearisatie (modus 1 of 2) X1 <> 0
941 = Linearisatie modus 1) Y1 <> 0

Error messages for cams (latst 2 digits = cam number)

1001…1024	length cam = 0 (cam begin = cam end) or cam begin <= cam end (with no counting range active)
1101…1124	length cam <= Hysteresis
1201…1224	(2 * hysteresis) + cam length >= counting range
1301…1324	cam begin and/or cam end outside counting range (incl. hysteresis)

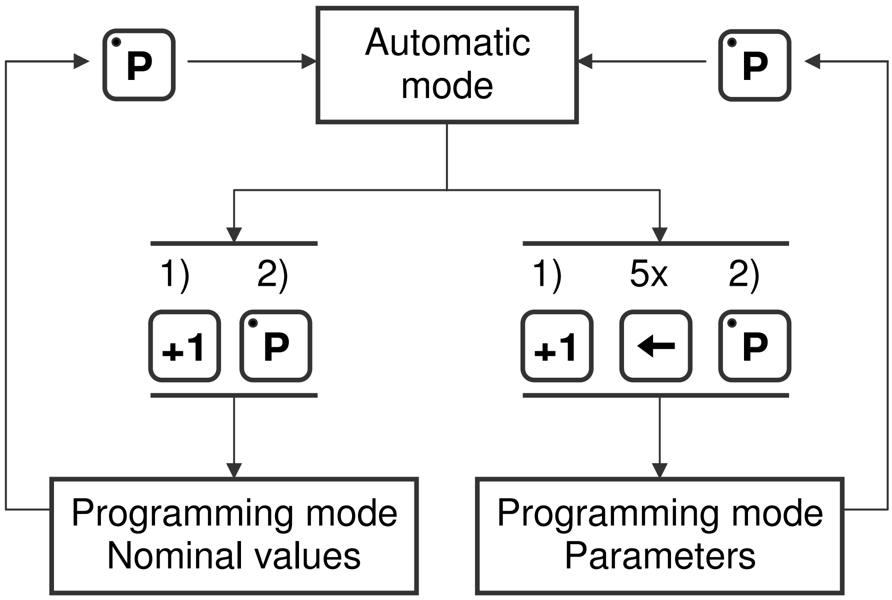
Error messages ASCII

er 1 = parity error
er 2 = frame error
er 3 = overflow error
er 4 = buffer overrun
er 5 = number invalid
er 6 = data invalid (outside min/max value)
er 7 = programming mode parameters/nominal values active

3 Programming

There are three different modes of operation:

- Automatic mode
- Programming mode for nominal values
- Programming mode for parameters



1) hold

2) press once

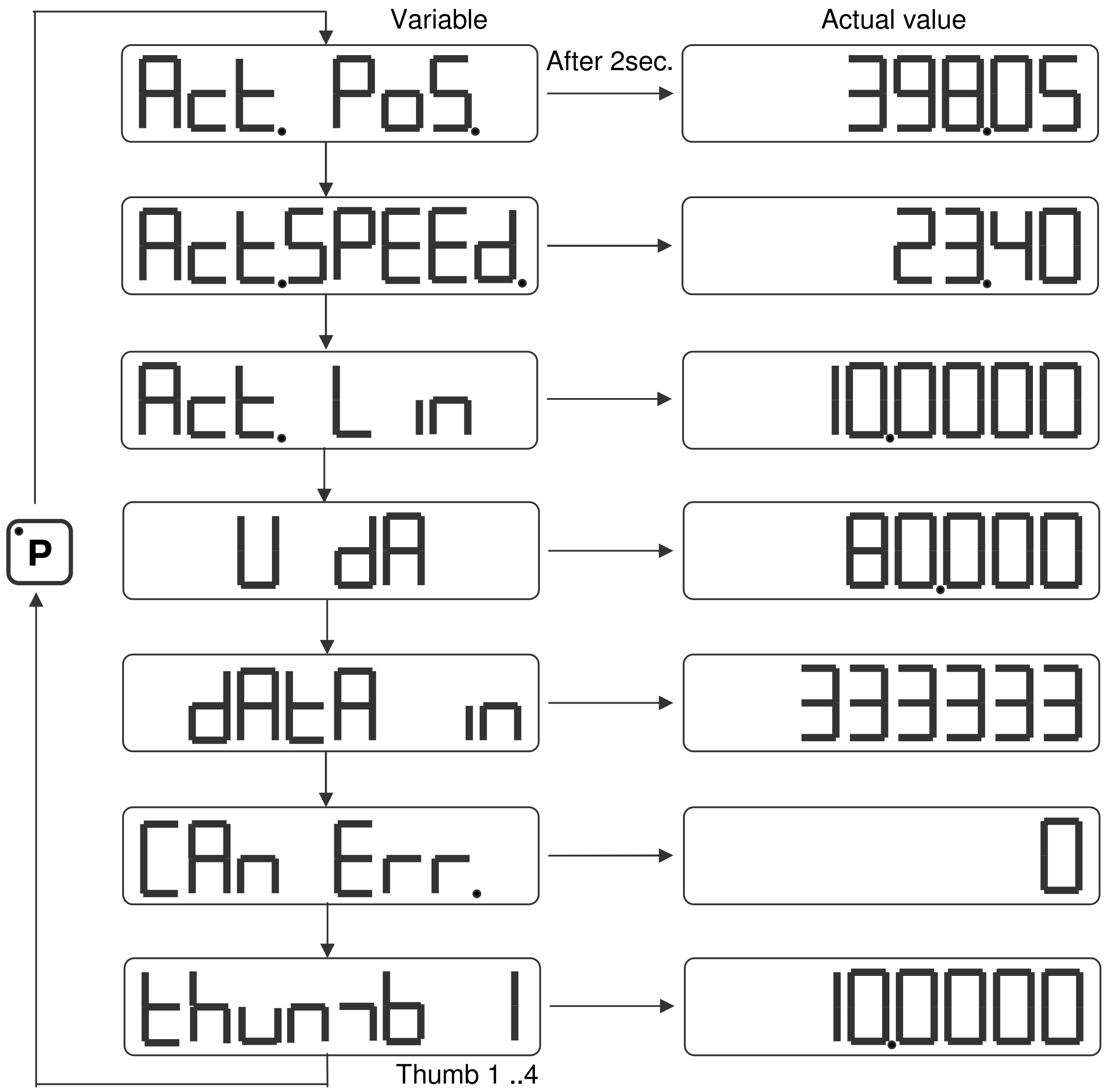
3.1 Automatic mode

In the automatic mode, depending on the type of sensor, the increments are counted or the absolute position is read and the result is shown on the display as the actual value. The velocity is calculated and can be visualized as well.

3.1.1 Monitor function

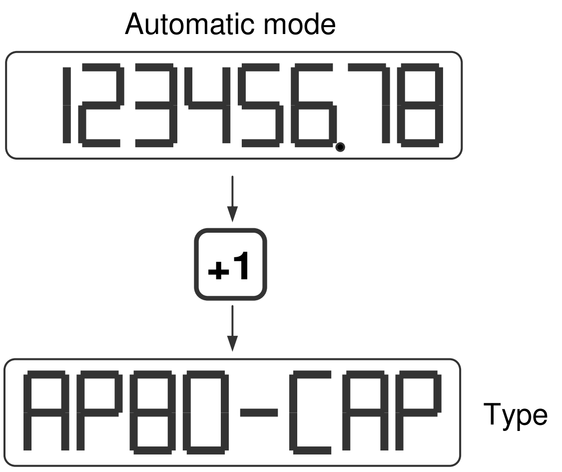
In automatic mode different variables can be displayed. By using the [P] key one can cycle through the different pages:

- Actual position
- Actual velocity
- Actual linearization
- Voltage or current of the D/A-converter
- Data input from the I/O module
- CANbus error
- Actual values thumbwheel sets

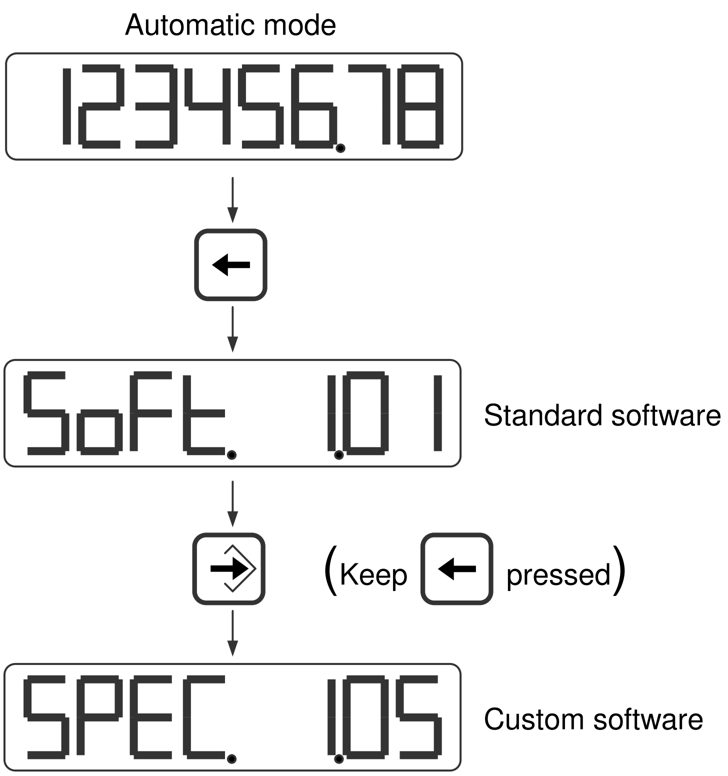


Parameter P[208] determines which option is visible after start-up.

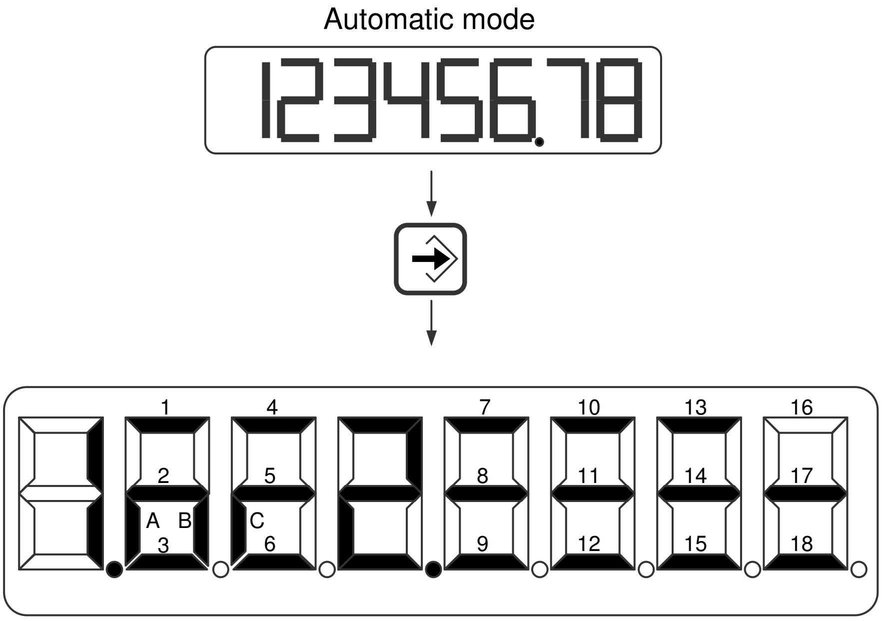
3.1.2 Displaying the type number



3.1.3 Displaying the software-version



3.1.4 Status in- and outputs

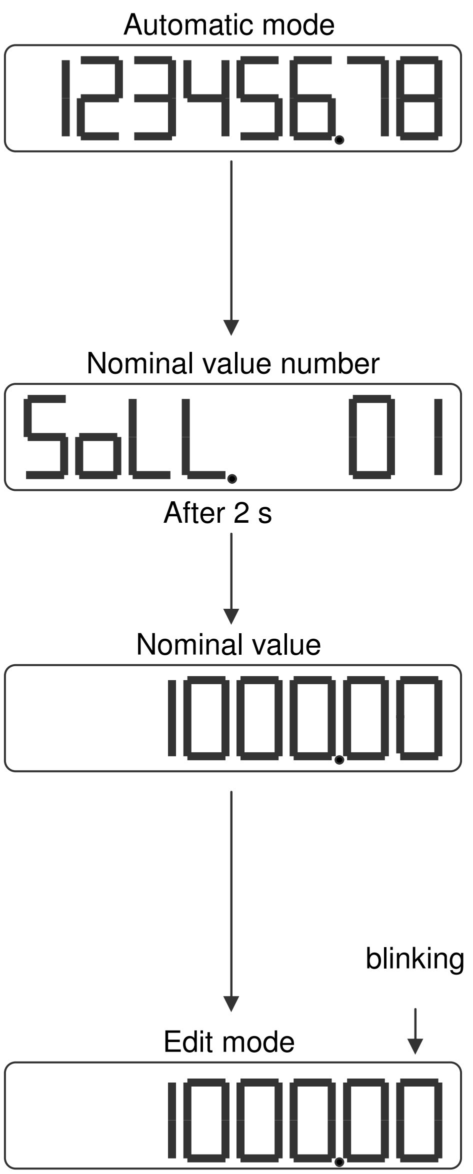


<u>Inputs</u>
1 = input-1
2 = input-2
3 = input-3
4 = input-4
5 = input-5
6 = input-6
A = input K1
B = input K2
C = input K0

<u>Outputs</u>
7 = output-1
8 = output -2
9 = output -3
10 = output -4
11 = output -5
12 = output -6
13 = output -7
14 = output -8
15 = output -9

<u>Outputs</u>
17 = Hold/Reset active
18 = Cams active

3.2 Changing nominal values



<u>Access to nominal values</u>

*) [+1] Then [P]

<u>Select nominal values</u>

- [P] Exit programming mode
- [+1] Nominal value number +1
- [←] Activate edit mode
- [→] Nominal value number +1

<u>Changing values</u>

- [P] Exit edit mode
- [+1] Increase digit
- [←] Move one digit left
- [→] Confirm change
- *) [←] Then [→] Clear input

*) keep pressed

3.3 Changing parameters

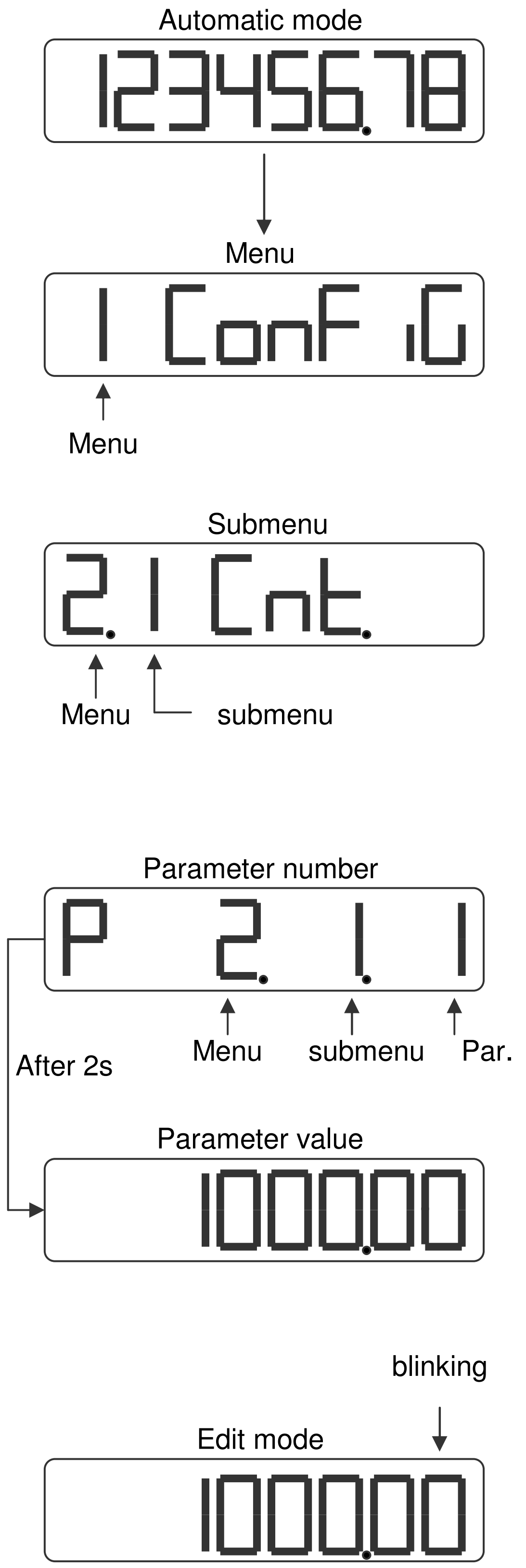
3.3.1 Menus

The parameters are displayed in different menus and submenus:

```
1 ConFiG
2 ActuAL
    2.1 Cnt.
    2.2 SSI
    2.3 FrEG.
    2.4 Par.
    2.5 CanAP
    2.6 StStP
3 Canbus
    3.1 ConFG.
    3.2 Obj/PDO in
    3.3 Obj/PDO out
4 SEriAL
    4.1 ConFG.
    4.2 SEr-1
    4.3 SEr-2
5 InPut
6 OutPut
    6.1 OP1
    ... ...
    ... ...
    6.9 OP9
7 DAtA
    7.1 DAtA in
    7.2 DAtA out
8 AnALoG
    8.1 ConFG.
    8.2 dA-U
    8.3 dA-I
9 Cam
    9.1  CA1
    ...  ...
    ...  ...
    9.24 CA24
10 D-SEt
    10.1 ConF.
    10.2 SEt-1
    10.3 SEt-2
    10.4 SEt-3
    10.5 SEt-4
11 Lin
    11.1 Conf
    11.2 P 1
    ... ...
    ... ...
    11.31 P30
```

Example:
PAr. 6.1.1 means
menu 6, submenu 1, parameter 1

3.3.2 Input parameters



*) keep pressed down

Access parameters

- [+1] *) 5x [←] Then [P]

Menu selection

- [P] Exit programming mode
- [+1] Menu item +1
- [→] To submenu/parameter number

Submenu selection

- [P] Back to menu
- [+1] Submenu item +1
- [→] To parameter number

Selecting parameters

- [P] Back to menu/submenu
- [+1] Parameter number +1
- [←] Activate edit mode
- [→] Parameter number +1

Changing parameters

- [P] Exit edit mode
- [+1] Increase digit-value
- [←] Move 1 digit to the left
- [→] Confirm input
- *) [←] Then [→] Clear input

4 Functions

4.1 Basic function

The basic function of the AP80 is programmed at P[200].
Factory setting P[200] = 0, where the sensor-value will be shown on the display.

When using the counter-input or internal frequency (P[201] = 0 or 2) there is a special function available called “Hold/Reset”. In this function the displayed value will be refreshed based upon an external start- and stopsignal.

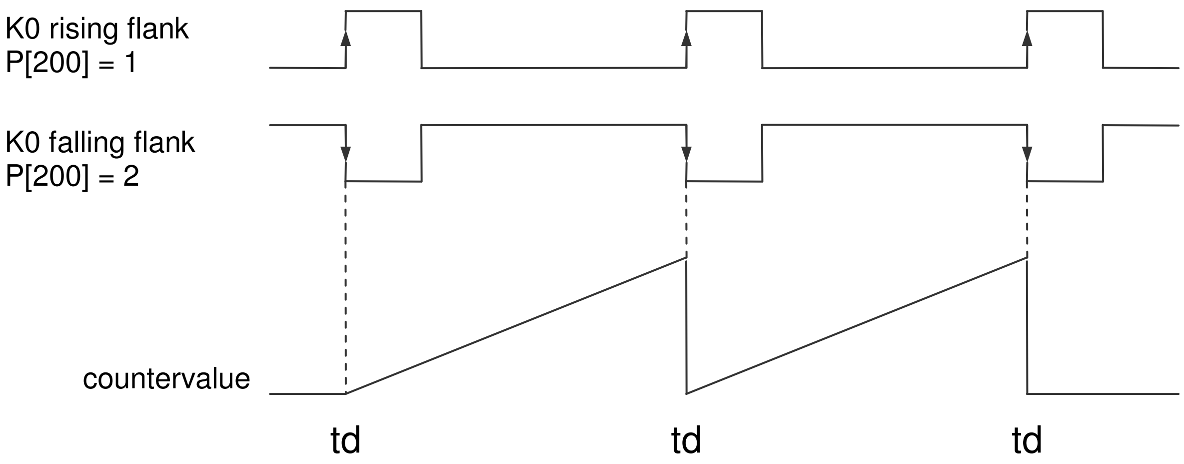
start: internal countervalue is set to 0
stop: display will be refreshed with internal countervalue

<u>Standard</u> (P[200] = 0)

Standard function.

<u>Hold/Reset K0 L →H and Hold/Reset K0 H →L</u> (P[200] = 1 or 2)

Hold/Reset start and stop with rising- or falling flank from input K0

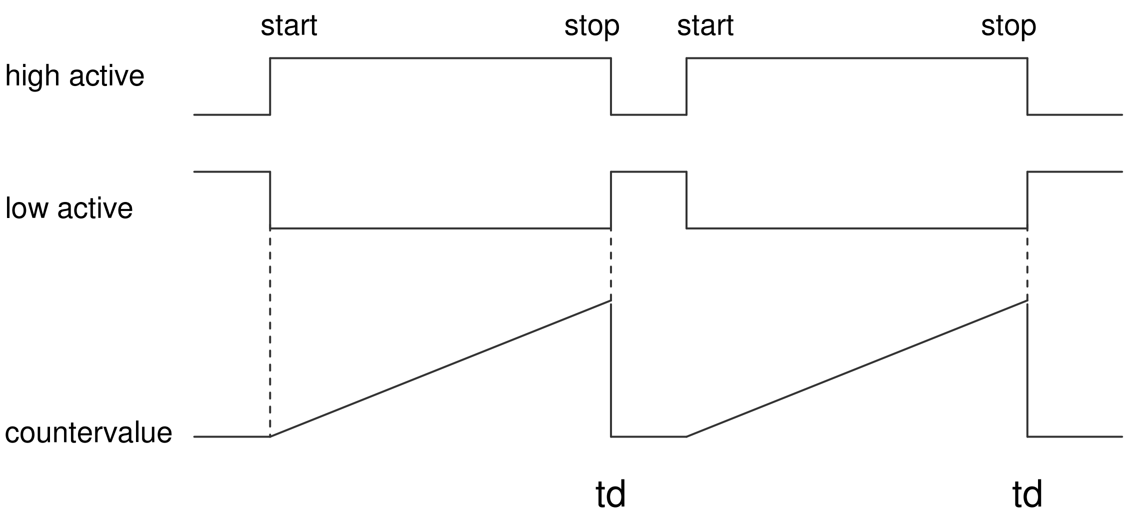


td = countervalue →display

If the Hold/Reset function 3…6 is active, the inputs for start and stop will have to be selected with P[249] … [254].

Hold/Reset start/stop H and Hold/Reset start/stop L (P[200] = 3 or 4)

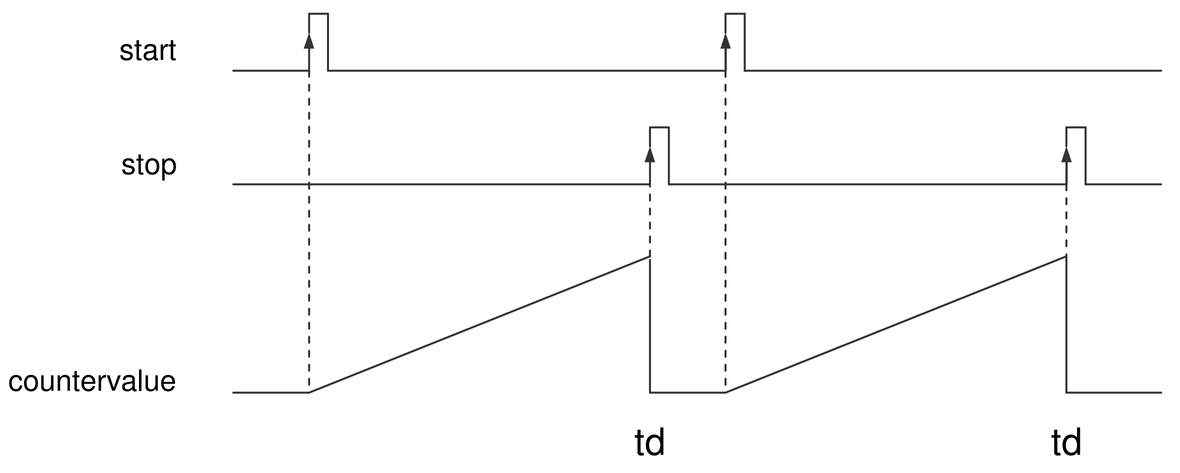
Hold/Reset start and stop using input (1…6) high and low active



td = countervalue →display

Hold/Reset start/stop L →H (P[200] = 5)

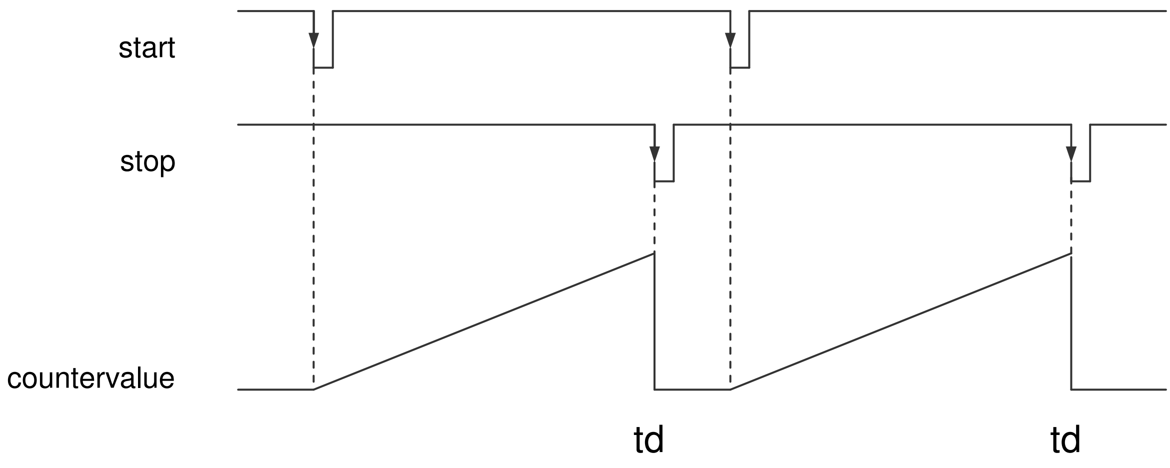
Hold/Reset start and stop using input 1…6 rising flank.



td = countervalue →display

Hold/Reset start/stop H →L (P[200] = 6)

Hold/Reset start and stop using input 1…6 falling flank



td = countervalue →display

4.2 Actual position

The actual position shown on the display is depending on the selected sensor input (P[201]) and several parameters.

Counterinput and internal frequency:

$$\text{Actual position} = \text{Counter} \times \text{FL} \times \text{dir} \times \frac{\text{Mt}}{\text{Mn}}$$

SSI, Start/Stop and parallel input:

$$\text{Actual position} = \text{input} \times \text{dir} \times \frac{\text{Mt}}{\text{Mn}} + \text{N}$$

FL	=	edgemultiplier	P[210]
Mt	=	multiplicator numerator	P[000]
Mn	=	multiplicator denominator	P[001]
N	=	offset	P[002]
dir	=	direction (x1 or x –1)	P[211]

4.3 Actual velocity

The velocity measurement is always active and can be displayed using the [P]-key (always AWE/s)
Two parameters are necessary to configure the measurement:

P[088] = measuring time [AWE/s]
The smaller the measurement time the more dynamic the velocity measurement will be. This time is also the refreshment-interval on the display.

P[202] = integrator
The number of cycles are programmed. The average velocity (unit =AWE/s) of the programmed cycles is calculated.

> *Example:*
> *Measurement time = 50ms, integrator = 10.*
> *The actual velocity will be refreshed every 50 ms and is the average velocity during the last 10 measurements.*

4.4 Multiplicator

By using the multiplicator it is possible to convert the SSI-value to display-units (AWE).

Example:
Encoder with 90º shifted signal and 1024 increments/revolution will have 4096 edges/revolution. If the desired value in the display is 360,0 (= 3600 AWE), the multiplicator will be 3600/4096 = 0,87890625.
It is possible to program the exact values (3600 and 4096) instead of the fraction.

Multiplicator (numerator)	*P[000]*	*=*	*3600*
Multiplicator (denominator)	*P[001]*	*=*	*4096*

Through P[203] it is possible to adjust the decimal point.

4.5 Power failure protection

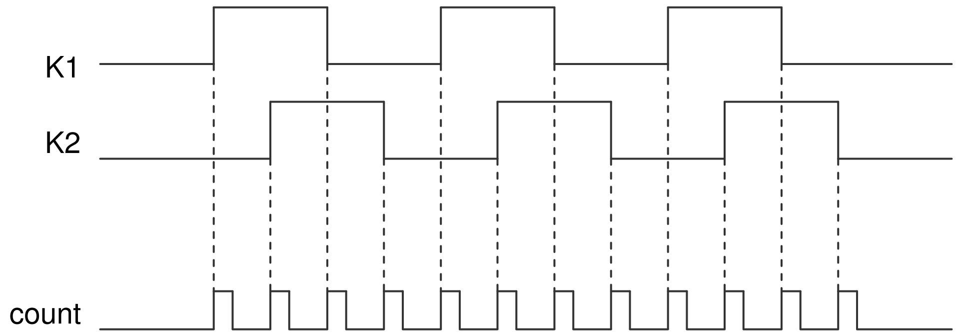
If P[206] = 1 then the actual position of the AP80 will be stored in EEPROM when power is shut down. After power up this value will be restored.

The power failure protection has no meaning with an absolute encoder (SSI or parallel).

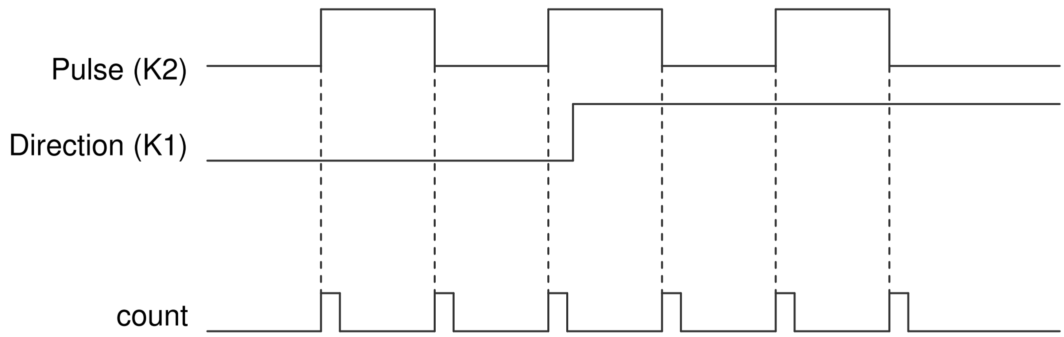
4.6 Edge multiplication (counter input)

There are two possibilities for the counter input:

V-signal x4: edge multiplication x4, 90º *shifted encodersignals.*



S-signal x2: edge multiplication x2, encoder signal with directional signal.



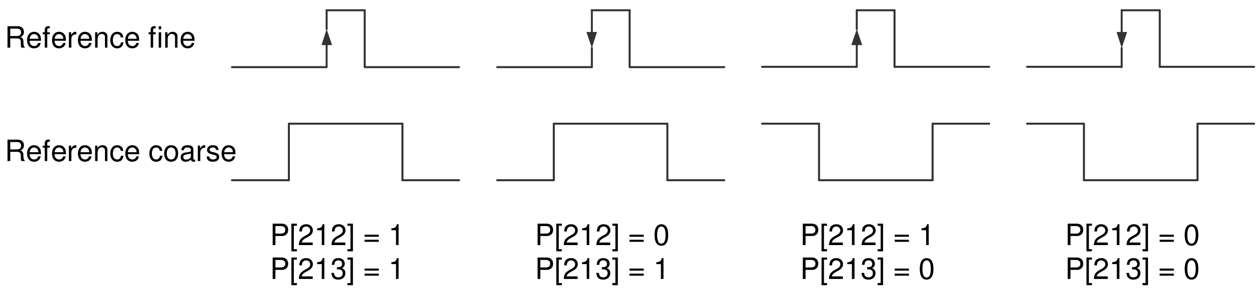
4.7 Preset (counter input and internal frequency)

The preset function can be used in 4 ways:

- Using input K0 (in combination with reference coarse)
- Using parameter service functions P[207]
- Using front keys
- Using input 1…6

4.7.1 Preset using input K0

The function preset is used to set the actual value to a programmed value, stored in P[003], P[204] or 102] or external thumbwheel-1.
The value will be set in case of an active edge from the reference fine signal K0. If reference coarse has been activated (P[213] <> 0) one of the inputs has to be programmed to “**`1 – Reference Coarse"`**.



4.7.2 Preset using service parameter P[207]

When P[207] has the value 111 then the actual position is preset with the value of P[003] when exiting the programming mode. The parameter reference coarse P[213] and direction reference fine P[209] have no function.
After executing this function parameter P[207] will have the value 0 again.

4.7.3 Preset using front keys

The function preset can be executed with the key combination [P] + [ETR].
Holding down the [ETR] key and push the [P] key one time. Then the actual position is preset with the value of P[003]. The parameter referene coarse P[213] and direction reference fine P[209] have no function.
This function is active when: P[212] = variant **`"3 front keys"`**

4.7.4 Preset using input 1…6

The function preset can be executed with a rising edge on input 1…6.
Then the actual position is preset with the value of P[003]. The parameter referene coarse P[213] and direction reference fine P[209] have no function.
This function is active when: P[212] = variant **`"6 rising edge input (1…6)"`**. One of the inputs has to be programmed to **`"17 – SeRef/Adjustment"`**.

4.8 Counting range (counter input and internal frequency)

The counting range used by the counter can be limited when using the counter input or the internal frequency (P[201] = 0 or 2). The counting range can be set either by parameter or nominal value.

4.8.1 Set counting range by parameter "Counting range" P[004]

If P[233] = variant **`"0 parameter counting range"`** the increments for the counting range can be programmed in parameter P[004], ignoring the decimal point. If P[004] = 0 the counting range is deactivated.

$$\text{Counting range P[004]} = \frac{\text{Display range (AWE)} \; * \; \text{Mul (nominator) P[001]}}{\text{Mul (denominator) P[000]}}$$

The number of increments is programmed, ignoring the decimal point.

<u>Counting range P[004]</u>
0 = function not active
1 … counting range

Example:
Incremental encoder, 90º shifted signals, 1000 pulses/rev. and 1,5 rev. = 360,0 degrees.

1000 pulses/rev. is equal to 4000 increments/rev. (edge multiplication x4).
3600 AWE ⇔ 1,5 x 4000 = 6000 increments

Multiplicator (numerator)	*P[000]*	=	*3600*
Multiplicator (denominator)	*P[001]*	=	*6000*
Counting range	*P[004]*	=	*6000 increments*

At P[203] it is possible to program the use of a decimal point.

Display will show: → 359,8 ... 359,9 ... 0,0 ... 0,1 ...0, 2 ←

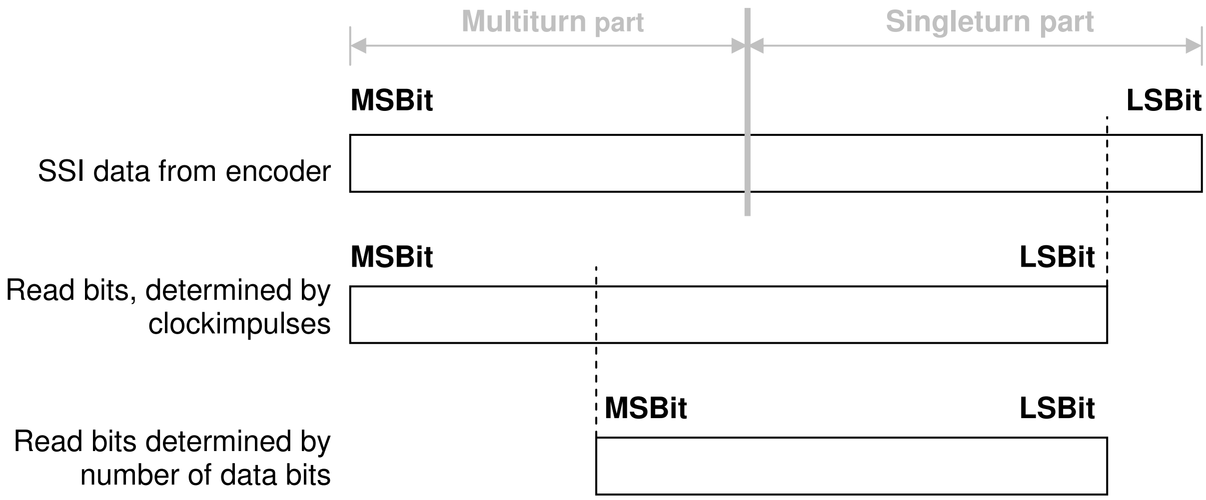
4.8.2 Set counting range by nominal value

If P[233] = variant **"x Nominal value x"** the increments in AWE for the counting range can be programmed by the selected nominal value. In this case P[004] is internally calculated. If the selected nominal value = 0 the counting range is deactivated

4.9 Number of bits SSI

To read the sensor values of SSI-encoders two parameters are important:

Number of clock impulses	P[216]
Number of data bits	P[217]



Normally the most significant bit (MSB) is transmitted first by the encoder.

The number of clock pulses determine the number of bits that will be read by the AP80. Basically this will be the number of bits that the encoder transmits.

> *Example:*
> *The SSI-encoder has a resolution of 4096 positions per revolution and 4096 coded revolutions. The number of clock pulses will be 12 (bit) + 12 (bit) = 24 (bit).*

The number of data bits will normally be the same: 24.
In some cases however it can be desirable to change the number of clock pulses and/or data bits.

Example:
The SSI-encoder has 16 singleturn bits (65535 positions/rev.) and 14 multiturn bits (16384 coded revolutions).

The resolution should be 8192 positions/rev. = 13 bit. The number of revolutions should not be changed (remains 14).

Number of clockpulses will be:	*P[216] = 27 (13+14)*
Number of databits will be:	*P[217] = 27 (13+14)*

4.10 SSI monitoring

The SSI-value is being read and processed every single cycle (250µs).
Due to external electromagnetic interference it may happen that during 1 cycle incorrect information has been read.
To prevent unwanted cam actions switching actions, the SSI value can be monitored.

There are two different ways of monitoring the SSI-value: detection of cable-failure (see also P[232], P[252]) and monitoring the delta-SSI.

The delta-SSI value is a maximum value that two separate readings may differ from each other (P[006]). It is possible to set the number of times that this value may be overrun (P[220]).

If there is a reading which is exceeding the max. difference, the last reading is interpolated. When the max overrun counting is reached there will be a SSI error generated.

4.10.1 Output "SSI error"

It is possible to create a SSI-error signal using one of the outputs 1..6:

Output SSI error
Output-x = option **"2 SSI error"** (Low = SSI error)

4.10.2 Reset "SSI error"

The SSI error-signal can be reset by using one of the following options:

- Activate and exit programming mode parameters
- Reset with PC-program DST80
- Serial communication (ASCII)
- Set one of the inputs 1…6 to **"4 Reset SSI error"** and apply a high signal

4.11 Justage SSI en StartStop sensors

The justage of absolute sensors can be done in 5 ways:

- Using the parameter "Adjustment absolute value" P[005]
- Using the input K0
- Using parameter service functions P[207]
- Using front keys
- Using input 1…6

4.11.1 Juatage using the parameter "Adjustment absolute value"

This function is applicable for rotary SSI encoders and is active when parameter "Adjustment" P[218] or P[219] has been set to: **"3 PAR"**

Parameter "Adjustment absolute value " P[005] can be set to a value to adjust the encoder. This value (which can be negative and positive) will be added to the actual value and for SSI sensors compensated with the display counting range.

$$\textbf{Display counting range} = \frac{\textbf{max SSI value * Multiplicator (numerator) P[000]}}{\textbf{Multiplicator (denominator) P[001]}}$$

4.11.2 Justage Using the input K0

In this function the actual position can be preset to a certain value. The function is active when parameter "adjustment" P[218] is set to one of the following options:

option: **"1 L→H K0 RAM"** (rising edge K0)
option: **"2 H→L K0 RAM"** (falling edge K0)
option: **"4 L→H K0 EEPROM"** (rising edge K0)
option: **"5 L→H K0 RAM Ext Data"** (rising edge K0)
option: **"6 H→L K0 RAM Ext Data"** (falling edge K0)
option: **"8 L→H K0 RAM thumbwheel-1"** (rising edge K0)
option: **"9 H→L K0 RAM thumbwheel-1"** (falling edge K0)

Parameter "adjustment value" P[005]] (or thumbwheel-1) can be set to a value to which the actual position will be adjsted when K0 detects a rising edge, possibly combined with a reference coarse signal ([P213]).
For option 5 and 6 this value is given by the external data input. (AP80/81-CxP).
If through parameter [P213] reference coarse has been activated ([P213] <> 0]), the option **"1 - Reference coarse"** should be selected for one of the inputs 1..6 (P[249]..P[254]).
The calculated offset will be stored in either RAM or EEPROM depending on the option chosen in parameter "adjustment" P[218]. When storing into RAM-memory (option 1,2,5 and 6) this will occur on a interrupt basis and can be performed during movement. The memory however is volatile and the value will not be permanently stored.

Storing the value into EEPROM (option 4) is not initiated by an interrupt. It is recommendable to perform this only when not moving or moving very slow. This value will be stored permanently.

4.11.3 Justage using service parameter P[207]

When P[207] has the value 111 then the actual position is adjusted to the value of P[005] when the programming mode is left.
After executing this function parameter P[207] will have the value 0 again.

4.11.4 Justage using front keys

The function preset the absolute position can be executed with the key combination [P] + [Enter].
Holding down the [Enter] key and push the [P] key once. Then the actual position is preset to the value of P[005].
This function is active when:

Sensor type SSI
P[218] = variant **"5 front keys"**

Sensor type Start/Stop
P[219] = variant **"5 front keys"**

4.11.5 Justage using input 1…6

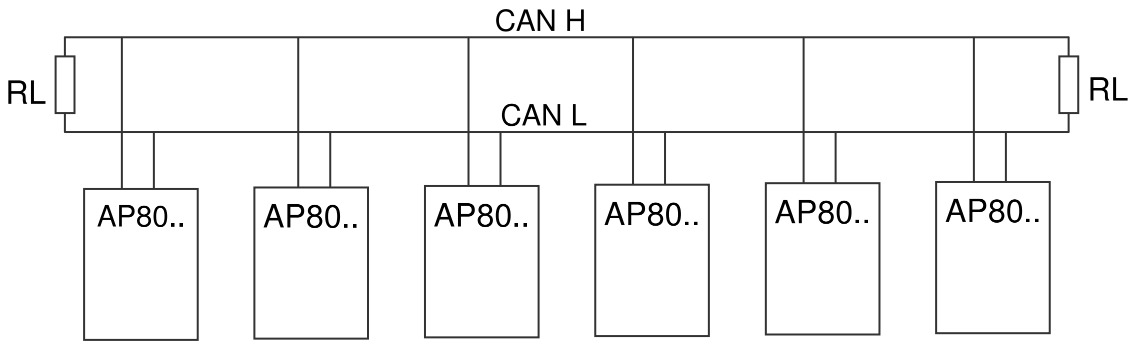
The function preset the absolute position can be executed with a rising edge on input 1…6. Then the actual position is preset to the value of P[005].
This function is active when:

Sensor type SSI
P[218] = variant **"10 rising edge input (1…6)"**
P[249] … [254]) variant **"17 – SetRef/Adjustment"** (only for 1 input)

Sensor type Start/Stop
P[219] = variant **"8 rising edge input (1…6)"**
P[249] … [254]) variant **"17 – SetRef/Adjustment"** (only for 1 input)

4.12 CAN-bus

The baudrate for the CAN-bus is stored in parameter P[228] and has a maximum of 1 Mbit/s. The highest usable baudrate is depending on used cable type and length. The first and last unit must have terminal resistors.

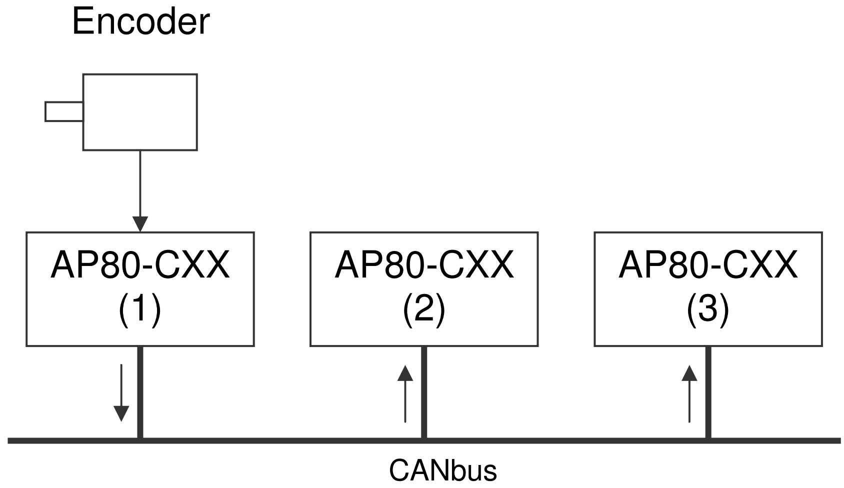


Using switch S3 enables the terminal resistor (120E).

The CAN-bus can be monitored on the display (see also paragraph 3.1.1), where “0” = no error and “1” = bus off.

4.12.1 AP-Link

Using the CAN bus it is possible to sent the actual position and velocity in just one message to one or more other AP80’s (AP40,s).
Example:



AP80-CXX (1) sending data on CANbus
AP80-CXX (2) receiving data from AP80-CXX (1)

AP80-CXX (3) receiving data from AP80-CXX (1)

Settings for example as shown above

	PAR	AP80 (1)	AP80 (2)	AP80 (3)
Input actual position	P[201]	0…3	4	4
Baudrate	P[228]	5	5	5
Obj1/PDO1 In adress	P[089]	-	1	1
Obj1/PDO1 In function	P[229]	0	1	1
Obj1/PDO1 Out adress	P[090]	1	-	-
Obj1/PDO1 Out function	P[230]	1	0	0
Time-out AP-Link	P[079]	-	> 0	> 0
Reset Time-out AP-Link	P[215]	-	0 or 1	0 or 1

In this example the first AP80 (1) is sending the actual position and velocity to adress 1 with 500 kbit/s (dataframe COB-ID 385…511).

AP80 (2) and (3) receive this data as actual position and velocity.
If the time-out value stored in P[079] is exceeded, the display will blink and an output will be set (programmable option).
This time-out error will be reset automatically when new data has been re-ceived, or by performing a reset through one of the inputs 1..6 (choose option 14: “reset time-out error AP-link”).

ATTENTION!
If there are (dynamic) cams positioned on the 0-position while using a closed counting range, the display-counting range P[103] must be programmed.

For example the master (AP80) is sending:

→ 599,8 ... 599,9 ... 0,0 ... 0,1 ...0, 2 ←

The display counting range should be set to 6000.
Parameter P[103] has no influence on the displayed value and is only used for calculating the cams.

P[103] is calculated as follows:

<u>Incremental encoders and internal frequency (P[201] = 0 or 2):</u>

$$\textbf{Display counting range P[103]} = \frac{\textbf{Counting range P[004]} * \textbf{Multiplicator (numerator) P[000]}}{\textbf{Multiplicator (denominator) P[001]}}$$

<u>SSI encoders (P[201] = 1):</u>

$$\textbf{Display counting range P[103]} = \frac{\textbf{Active data bits P[217]}}{\textbf{2}} \quad \textbf{-1}$$

4.12.2 Start/Stop sensor

The AP80 sends a start signal to the sensor and then waits for the stop signal to return from the sensor. The time between the start and stop signal is an measure for the sensor (magnet) position.

Every sensor has it’s own characteristic internal sensor speed. This speed is defined as gradient in m/s. for example 2789,00 m/s.
For MTS sensors this gradient is written on the type shield.
The gradient value can be programmed as Parameter P[115] in the AP80. The measurement position will then be displayed correctly.

Parameter "Sensor length" P[118] can be set to the nominal sensor length so the AP40 can calculate the number of measurements per second.

4.13 ASCII protocol

The serial ports of the AP80, both RS232 and RS422/485, are able to work with the ASCII protocol, however not at the same time.

Using the ASCII protocol, actual values can be read, parameters and nominal values can be stored and read, the status of the digital inputs and outputs can be monitored etc.

4.13.1 Overview functions

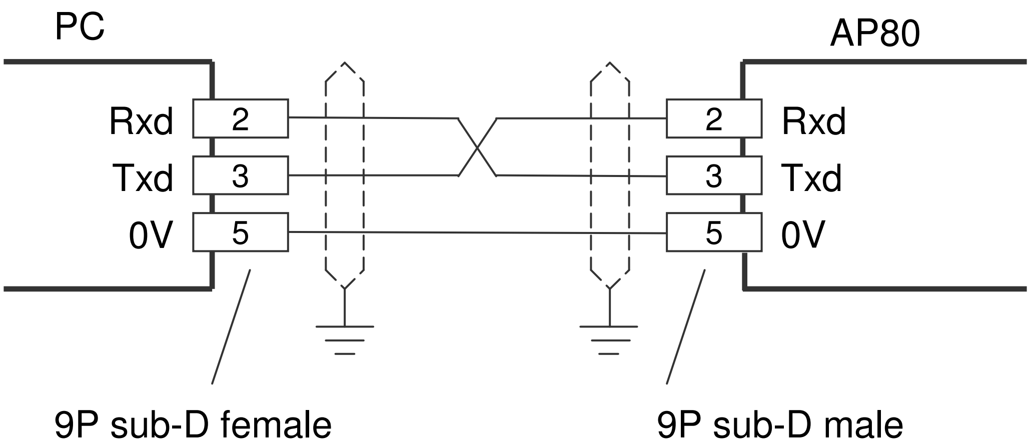
`sc` **select AP80**
`r0` **Read actual position (AWE)**
`r1` **Read actual velocity (AWE/s)**
`r2` **Read actual voltage output (0,1mV units)**
`r3` **Read actual current output (0,1 mA units)**
`r8` **Read actual linearization**
`ri` **Read status inputs**
`ru` **Read status outputs**
`rk` **Read status input K1, K2 and K0**
`wu` **Write outputs**
`ra` **Read status data input**
`rb` **Read status data output**
`wd` **Write data output**
`rp` **Read parameter**
`wp` **Write parameter** (Only EEPROM)
`rs` **Read nominal value**
`ws` **Write nominal value** (RAM + EEPROM)
`rx` **Read software version**
`rt` **Read type number**
`rh` **Read hardware version**
`rf` **Read error number**
`wf` **Reset SSI error**
`rn` **Read status bits**
`rd` **Read actual value thumbwheel sets**
`bp` **Load and activate**

4.13.2 General

Through the ASCII protocol it is possible to communicate with the AP80

Send: Data from PC, PLC → AP80

Receive: Data from AP80 → PC, PLC



(5m RS232 cable: KBL006-005)

Send structure:
Functioncode (space) [argument 1](space)[argument 2] <CR>

Receive structure:
Functioncode (space) [argument1](space)[argument 2] <CR> <LF>

Functioncode (space) [argument1] [argument 2].
Argument 1 and 2 are depending on the function and are separated by a space.

Example:
`wp 20 250` *(write value 250 to parameter 20)*

4.13.3 Functions

`sc` Select AP80

send:	**`sc xx`**
receive:	**`sc xx`**
transmitting parameter:	Unit Id number

The AP80 with the unit Id number is selected, all consecutive commands are relevant for this unit.
An AP80 with unit Id number 0 will always respond. This is the reason that only one unit is allowed to have unit Id number 0.

`r0` Read actual position (AWE)

send:	**`r0`**
receive:	**`r0 xxxxxxxx`**
transmitting parameter:	none

`r1` Read actual velocity (AWE/s)

send:	**`r1`**
receive:	**`r1 xxxxxxxx`**
transmitting parameter:	none

`r2` Read actual voltage output (0,1mV units)

send:	**`r2`**
receive:	**`r2 xxxxxxxx`**
transmitting parameter:	none

`r3` Read actual current output (0,1 mA units)

send:	**`r3`**
receive:	**`r3 xxxxxxxx`**
transmitting parameter:	none

`r8` Read actual linearization (AWE)

send:	**`r3`**

receive: **`r3 xxxxxxxx`**
transmitting parameter: none

`ri` Read status inputs

send: **`ri`**
receive: **`ri xxx`**
transmitting parameter: none

B0 = input K0
B1 = input-1
B2 = input-2
B3 = input-3
B4 = input-4
B5 = input-5
B6 = input-6
B7 = 0

*Example: **`ri 23 gives the following answer:`***
23 → 17H, 0001 0111 B
K0 = "1"
input-1 = "1"
input-2 = "1"
input-3 = "0"
input-4 = "1"
input-5 = "0"
input-6 = "0"

`ru` Read status outputs

send: **`ru`**
receive: **`ru xxx`**
transmitting parameter: none

B0 = output-1
B1 = output-2
B2 = output-3
B3 = output-4
B4 = output-5
B5 = output-6
B6 = output-7
B7 = output-8

B8 = output-9

`rk` Read status input K1, K2 and K0

send:	**`rk`**
receive:	**`rk x`**
transmitting parameter:	none

B0 = K0
B1 = K1 or counting direction
B2 = K2 or counting pulse

`wu` Write outputs

(only valid for outputs with “ASCII protocol” selected)

send:	**`wu xxx`**
receive:	**`wu xxx`**
transmitting parameter:	data for output

B0 = output-1
B1 = output-2
B2 = output-3
B3 = output-4
B4 = output-5
B5 = output-6
B6 = output-7
B7 = output-8
B8 = output-9

Example:
Output-2 and output-9 should be set to “1”:
100000010B = 102H = 258D
send: wu 258
receive: wu 258

`ra` Read status data input

send: **`ra`**
receive: **`ra xxxxxxxx`**
transmitting parameter: none

Example:
`ra 1320 gives the following answer:`
1320 → 528H, 0101 0010 1000 B

`rb` Read status data output

send: **`rb`**
receive: **`rb xxxxxxxx`**
transmitting parameter: none

Example:
`rb 1320 gives the following answer:`
1320 → 528H, 0101 0010 1000

`wd` Write data output

“Source” for data output (P[387]) = 3 "via ASCII protocol". If not, err 8 will be answered.
Following parameters have no function:

P[384] "Data output code"
P[385] "Minus sign for data output"
P[386] "Data ready for data output"

send: **`wd xxxxxxxx`**
receive: **`wd xxxxxxxx`**
transmitting parameter: data for output

Example writing data output with value 280D
send: ***`wd 280`***
answer: ***`wd 280`***

The output will be: 0000 0000 0000 0001 0001 1000B

`rp` Read parameter

send: **`rp xxx`**
receive: **`rp xxxxxxxx`**
transmitting parameter : parameter number

Example reading parameter P[004]
send: ***`rp 4`***
answer: ***`rp 4 10000`***

`wp` write parameter (only EEPROM)

send: **`wp xxx xxxxxxxx`**
receive: **`wp xxx xxxxxxxx`**
transmitting parameter: parameter number and parameter value

Example writing parameter P[004] with value 185000
send: ***`wp 4 185000`***
answer: ***`wp 4 185000`***

Parameter will be stored to EEPROM but is not yet active.

`rs` Read nominal value

send: **`rs xx`**
receive: **`rs xx xxxxxxxx`**
transmitting parameter: nominal value

Example reading nominal value 22
send: ***`rs 22`***
answer: ***`rs 22 72500`***

`ws` **Write nominal value** (RAM + EEPROM)

send: **`ws xx xxxxxxxx`**
receive: **`ws xx xxxxxxxx`**
transmitting parameter: nominal value number and nominal value

Example writing nominal value 22 with value 195200
send: ***`wp 22 195200`***
answer: ***`wp 22 195200`***

`rx` **Read software version**

send: **`rx`**
receive: **`rx SW Vxx.xx SSW xx.xx`**
transmitting parameter: none

SW = standard software version
SSW = special software version

Example:
send: ***`rx`***
answer: ***`rx SW 4.02 SSW 1.00`***

`rt` **Read type number**

send: **`rt`**
receive: **`rt AP80`**
transmitting parameter: none

Example:
send: ***`rt`***
answer: ***`rt AP80-CAP`***

`rh` **Read hardware version**

send: **`rh`**
receive: **`rh HW x RV x`**
transmitting parameter: none

`rf` Read error number

send: `rf`
receive: `rf xxxx`
transmitting parameter: none

When -1 returns no error is present.

Example:
send: `rf`
answer: `rf 800` *(SSI error)*
or
answer: `rf -1` *(no error)*

`wf` Reset SSI error

send: `wf`
receive: `wf`
transmitting parameter: none

`rn` Read status bits

send: `rn`
receive: `rn xxx`
transmitting parameter: none

B0 = cams active (started)
B1 = reference/adjustment set

Example:
`rn 3 gives the following answer:`
3 → 3H, 0011 B
cams are active and ref/adjustment have been set

`rd` Read actual values thumbwheel sets

send: `rd 1`
receive: `rd 1 xxxxxxx`
transmitting parameter: thumbwheel number

bp Load and activate

send:	**bp**
receive:	**bp xxx**
transmitting parameter:	none

In case of an error the error number will be returned (-1 is no errors).

Example:	*answer:*	***bp -1***	*(no errors)*
	answer:	***bp 20***	*(error parameter 20)*

4.13.4 Error messages

In case of an error the AP80 will sent an error message followed by an error number.

<u>overview error messages</u>
er 1 = parity error
er 2 = frame error
er 3 = overflow error
er 4 = buffer overrun
er 5 = number invalid
er 6 = data invalid (for example outside min/max range)
er 7 = programming mode parameters/nominal values still active
er 8 = function impossible

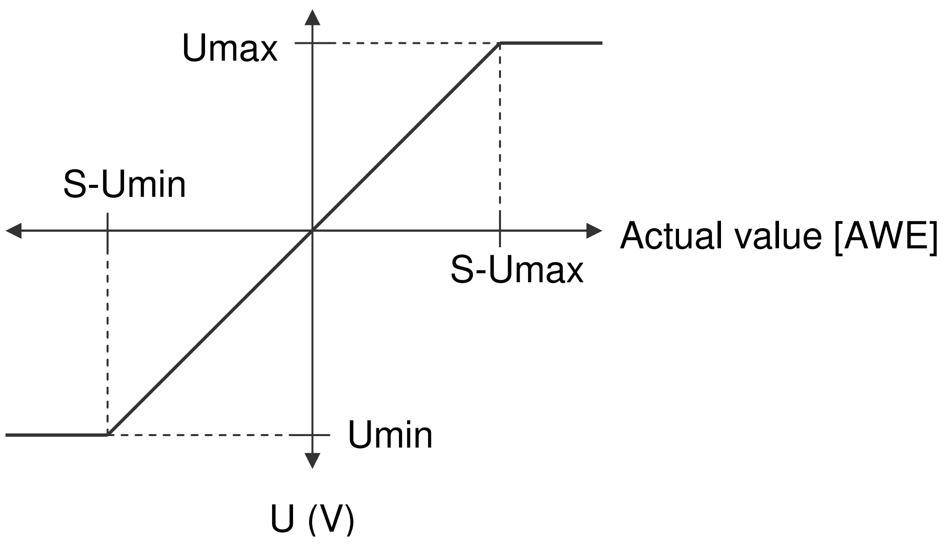
4.14 Analog output

The AP80 has an optional, galvanic isolated analog output. Using parameter P[388] it is possible to choose between a current output or a voltage output.The analog output can be used to give out the actual position or velocity (see parameter P[383]).

4.15 Voltage output

The voltage output has a resolution of 305 µV and is programmable through P[080] … P[083].

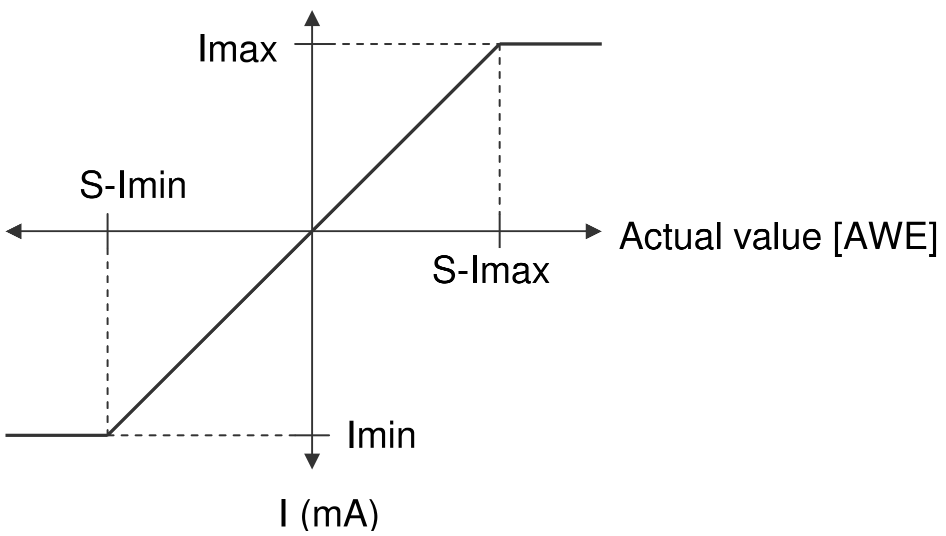
P[080] = Umin [V] (input in 0,0001V units)
P[081] = Umax [V] (input in 0,0001V units)
P[082] = S-Umin [AWE] (actual value at Umin)
P[083] = S-Umax [AWE] (actual value at Umax)



4.16 Current output

The current output has a resolution of 610 µA and is programmable through P[084] … P[087].

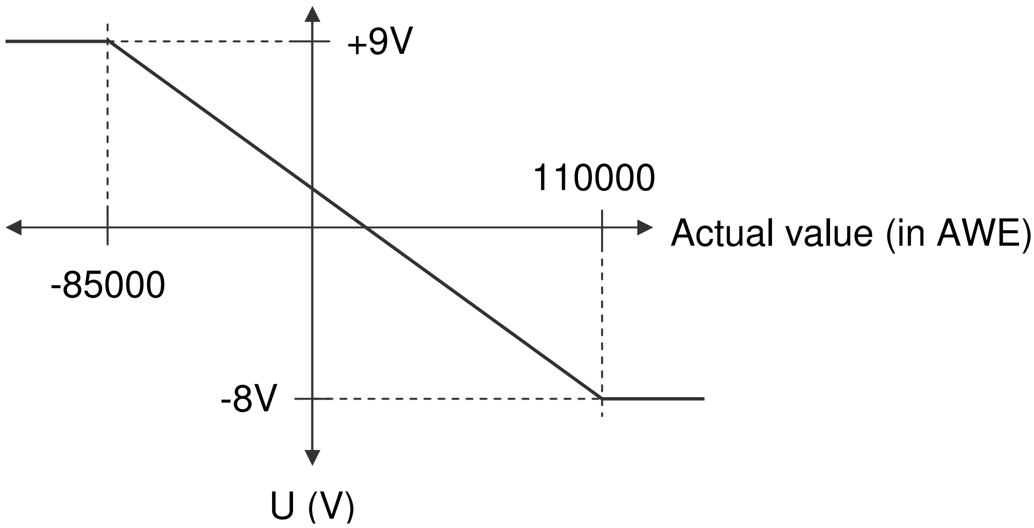
P[084] = Imin [mA] (input in 0,0001mA units)
P[085] = Imax [mA] (input in 0,0001mA units)
P[086] = S-Imin [AWE] (actual value at Imin)
P[087] = S-Imax [AWE] (actual value at Imax)



4.17 Example programming voltage output

> 0,01 mm units
> actual position at +9V should be -850,00 mm
> actual position at -8V should be 1100,00 mm

P[080]	=	Umin	=	-8,0000 [V]
P[081]	=	Umax	=	+9,0000 [V]
P[082]	=	S-Umin	=	110000 [AWE]
P[083]	=	S-Umax	=	-85000 [AWE]



4.18 Cams

4.18.1 General

The AP80 has a maximum of 24 programmable cams divided over a maximum of 9 outputs.

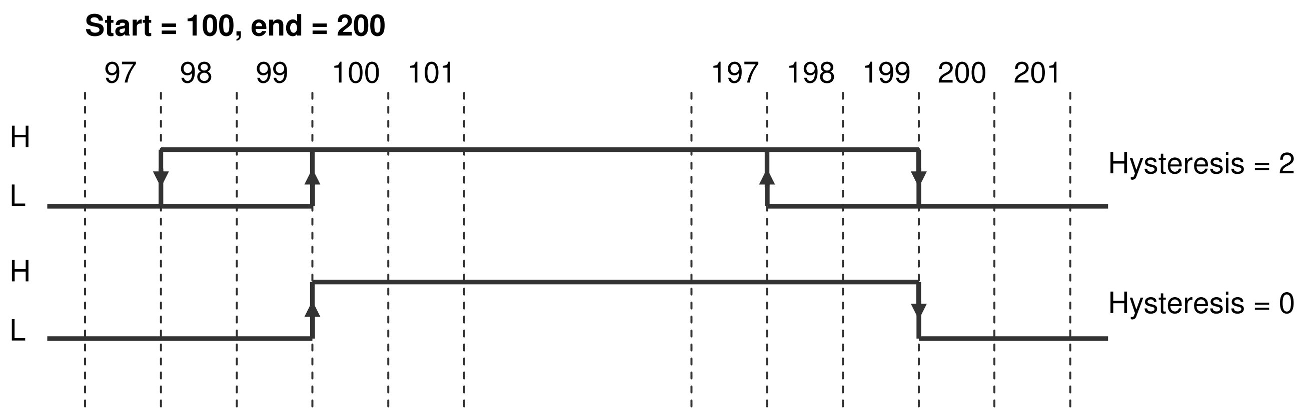
Programmable functions:

- Type
  1. Cam with start- and end-value
  2. Greater than or equal to limit value
  3. Smaller than or equal to limit value
- Source (actual position, actual velocity or external data input minus actual position etc.)
- Nominal value location number for start-, end-, or limit value
- Hysteresis
- Output for cam

Per cam one can choose whether to program the values directly into the parameters or to use a nominal value location number where the values are programmed. In addition the external thumbwheel sets can be used as start and end value.

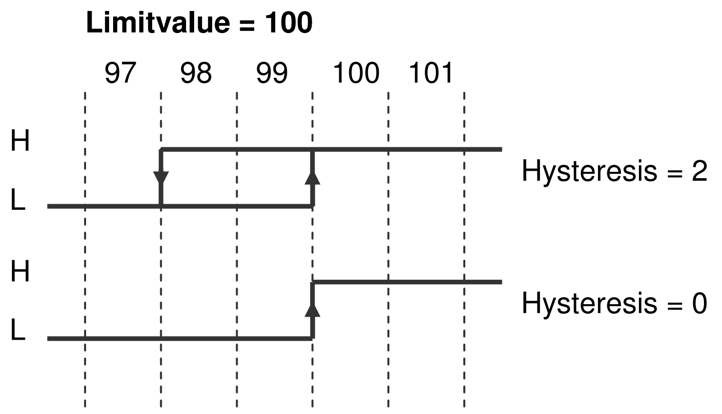
4.18.2 Cam with start- and end-value

Two values are programmed, a start value and an end-value.



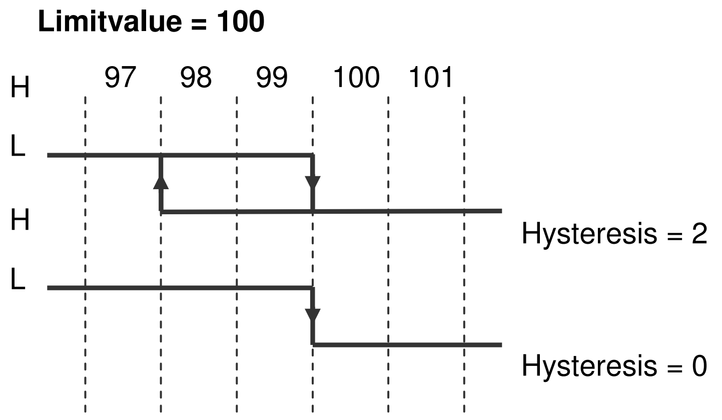
4.18.3 Greater than or equal to limit value

Only one limit value needs to be programmed.



4.18.4 Smaller than or equal to limit value

Only one limit value needs to be programmed.



4.18.5 Dynamic cams

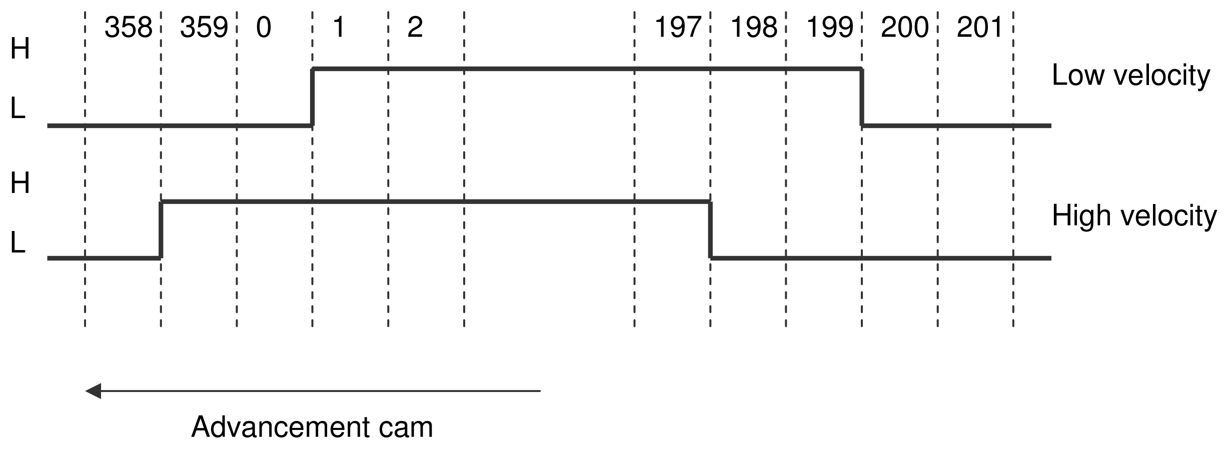
To compensate for actions with a static time, for example the switch time of a valve, it is possible to program a time for each output individually. The cams will be shifted according to this programmed time.

This function is only available for cams with a start- and end-value (P[256] … P[279] = 1) and when the source of this cam is the actual position(P[280] … P[303] = 1).

When working with a counting range the possibility of cams shifting over the zero-point is taken into account.

Example:

startvalue:	*1*
endvalue:	*200*
hysteresis:	*0*
counting range display:	*360*



4.18.6 Start/stop cam

The outputs for the cams can be enabled or disabled, if for one of the inputs 1..6 the function start/stop cams has been chosen.

Start/stop cams with one signal
Input-x = option **"5 start/stop cams"** (high = cams enabled)

Start/stop cams with double signal
Input-x = option **"6 start cams"** (rising edge = enable cams)
Input-x = option **"7 stop cams"** (rising edge = disable cams)

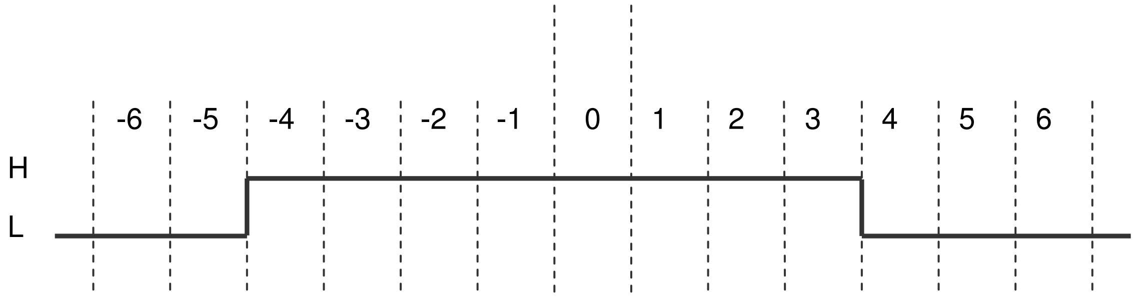
4.18.7 Output "Cams active"

On one of the outputs 1…9 the signal “cams active” can be generated by choosing option**"4 cams active"** (high = cams enabled).

4.18.8 Relative cams

When using relative cams it is possible to configure simple positioning systems. The cams will switch based on the difference between the actual position and an external value. This value is provided through the data input module.

Difference (External data – actual position)



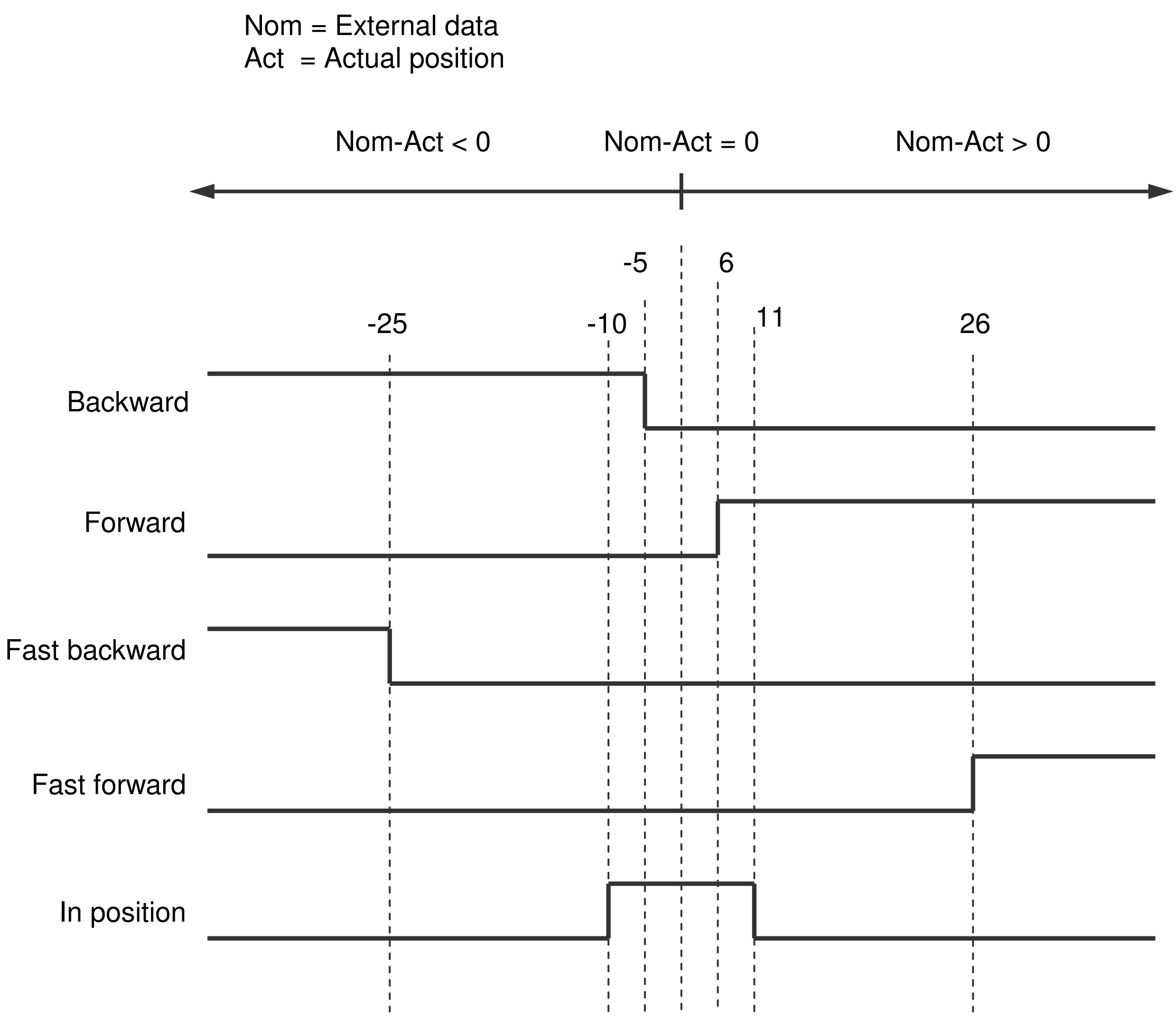
Cam start = -4

Cam end = 4

Example:
A 2-speed motor should be positioned towards an externally provided position. The signals to be used for the drive are:

forward:	*output-1*
backward:	*output-2*
fast forward:	*output-3*
fast backward:	*output-4*
in-position:	*output-5*
start input:	*input-1*
stop input:	*input-2*

<u>Start stop cams</u>
Input-1 = option **"6 start cams"**
Input-2 = option **"7 stop cams"**



Signal	Outp. Nr.	Cam Nr.	Cam Function	Cam Source	Cam start	Cam end
Backward	1	1	3	2	-5	---
Forward	2	2	2	2	6	---
Fast backward	3	3	3	2	-25	---
Fast forward	4	4	2	2	26	---
In position	5	5	1	2	-10	11

To keep the switch distances in positive and negative direction the same, for all the positive switch distances the value should be raised by 1.

If in the example above there is only need for one signal to determine the fast/slow-function, it is possible to appoint output-3 to cam 4. Both cams will then be set on output-3.

4.19 Data input

Using the data input module (AP80-CXP) it is possible to read a maximum of 24 bits. This data input can be used for reading the actual position data from bit parallel encoders or as input for the nominal value in case of relative cams. It is also possible to read a minus sign through pin 12 or 25. Pin 25 can also be used to read a “data valid” signal.

4.20 Data output

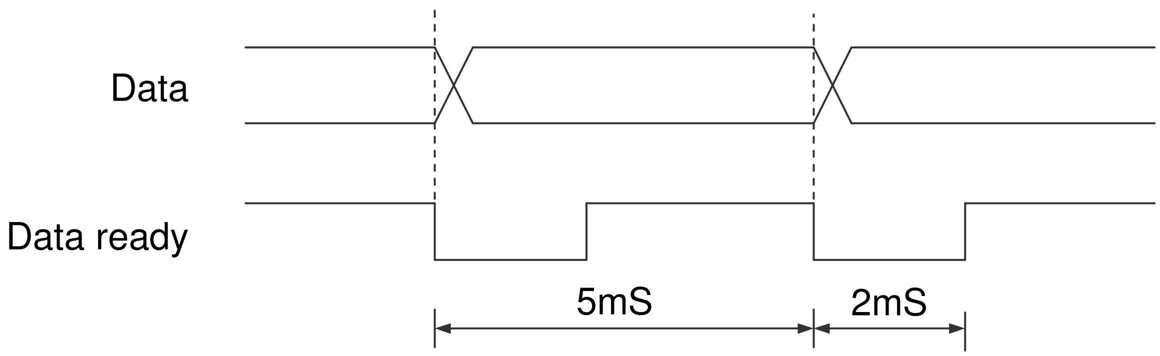
4.20.1 General

Using the data output module it is possible to give out a maximum of 32 bits
AP80-CXP 24 open emitter outputs.
AP80-CXF 32 tri-state outputs.

Possible output data is the actual position or actual velocity. The data of multiple AP80’s can be set to busmode and therefore all be read by a PLC. The store-function will lock the data during reading and the enable-function will enable the data.

4.20.2 Output data ready

The signal data ready ensures that the data on the outputs is stable. A high signal means that the data is stable.

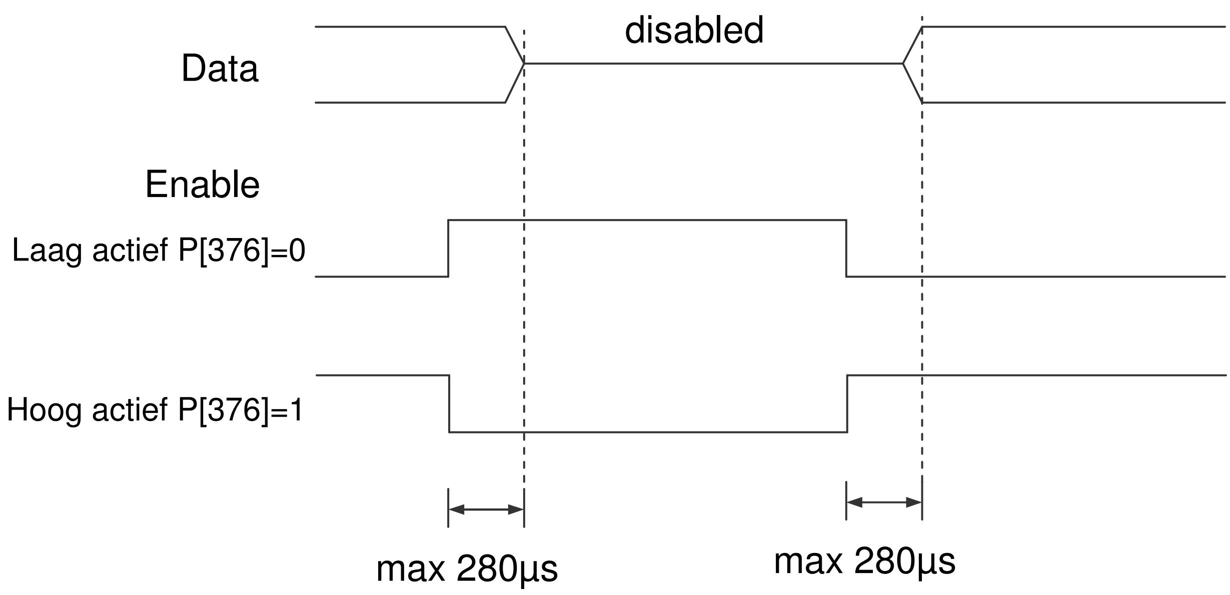


4.20.3 Input enable

Through the inputs one can activate the “enable”- signal. The signal activates or deactivates the data output.

If none of the inputs has been assigned, the data output is always active.

If there has been an output programmed with the enable-function, the signal will work as follows:

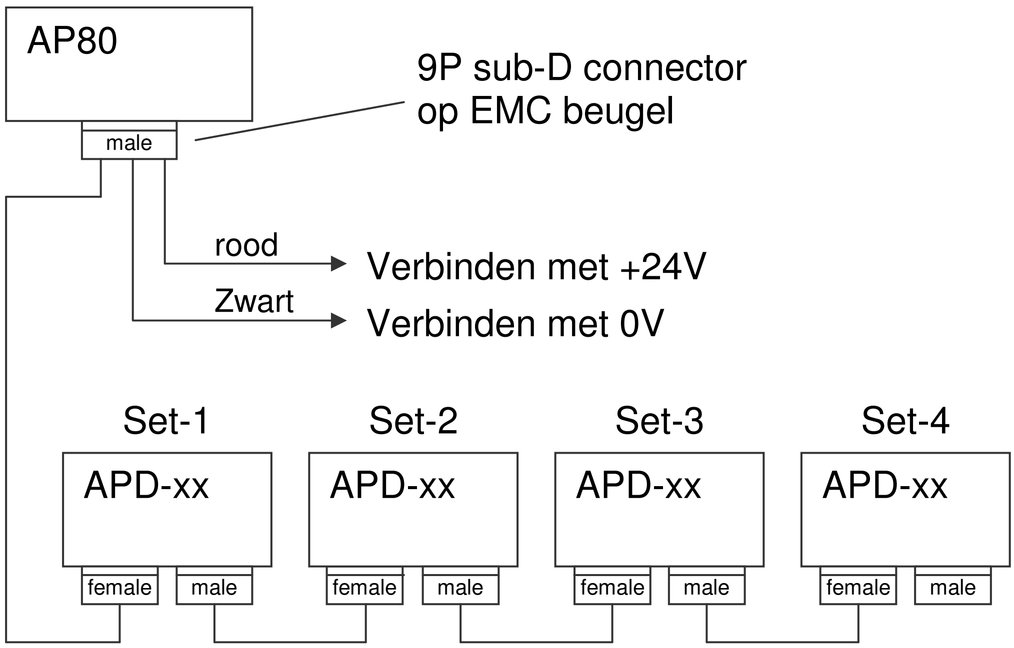


4.21 External thumbwheel sets

4.21.1 General

Up to 4 external thumbwheel sets can be connected to the AP80.
The data is read by the RS422 serial interface (ser-2) en can be configured separate. The thumbwheel value can also be read by the monitor function in auto mode. They can be used for different functions such as begin and end values for the cams. Thumbwheel-1 can also be used for justage/presetting of the actual position.

4.21.2 Connecting thumbwheel sets



Cable KBL101-0003 (30cm): connection AP40 to first thumbwheel
Cable KBL102-0003 (30cm): connection between thumbwheel sets

Available types:

APD-04	4 decades
APD-05	5 decades
APD-06	6 decades
APD-07	7 decades

APD-T4	4 decades with sign
APD-T5	5 decades with sign
APD-T6	6 decades with sign

4.21.3 Parameter settings

P[245] must be set to 0 (if RS422/RS485 ser-2 is not active)

Parameter P[222] determines the number of thumbwheel sets that will be read.

P[223] determines the method of reading the thumbwheel value:
option: **"0 auto"**
option: **"1 on ext. input"** (rising edge)

When P[223] = 1 then one of the inputs must be programmed with option 18 **"read thumbwheel switches"**

For each thumbwheel set there are 4 parameters.

Parameter "Format" P[110] … P[113]
option: **"0 7654321"**
option: **"1 x654321"**
option: **"2 xx54321"**
option: **"3 xxx4321"**
option: "4 xxxx321"
option: "5 xxxxx21"
option: "6 xxxxxx1"
option: **"7 -654321"**
option: **"8 x-54321"**
option: **"9 xx-4321"**
option: "10 xxx-321"
option: "11 xxxx-21"
option: "12 xxxxx-1"

Parameter "Min" P[120] … P[123]
Definition of minimum input value

Parameter "Max" P[124] … P[127]
Definition of maximum input value

Parameter "Multiplicator" P[128] … P[131]
Multiplies the value of the thumbwheel set
option: **"0 x1"**
option: **"1 x10"**
option: **"2 x100"**

4.22 Linearization

The linearization function allows to display and process nonlinear motions. The actual display position (sensor) position or actual velocity is converted into an additional value “Actual linearization”. This additional value can serve as a source for the cams function and analog output.

Parameter “Source” P[246]
optie **"0 Inactive"**
optie **"1 Actual position"**
optie **"2 Actual speed"**

The linearization function works with a tabel of max. 30 points (P1…P30). Every point has a X (Pn-X) and Y (Pn-Y) value. The X value represents the actual (sensor) position or velocity and the Y value represents the desired actual (sensor) position or velocity called “Actual linearization”. Interpolation takes place between these points

Parameter P[247] determines the number of active points en can be programmed with a value of 2 …30 points. The value of the actual position linearization can be displayed in the monitor function. P[248] determines the number of decimals shown.

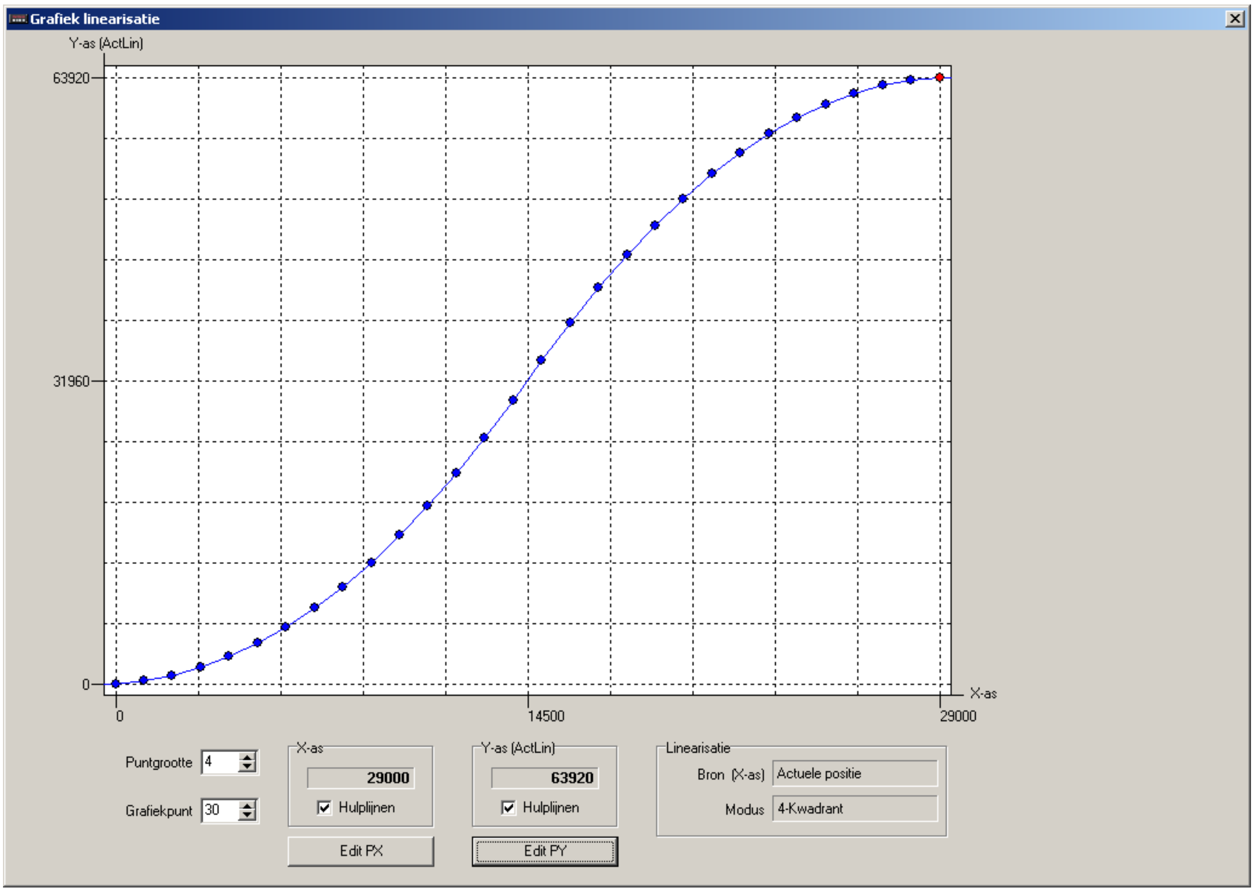
For linearization 3 modes are available.

Parameter “mode” P[241]
optie **"0 4-Quadrant"**
optie **"1 Mirror Y-axis"**
optie **"2 Mirror XY-axis"**

4.22.1 mode 0 “4-Quadrant”

This is the standard mode wich can build every possible curve. Both the X-axis and the Y-axis may contain negative values.

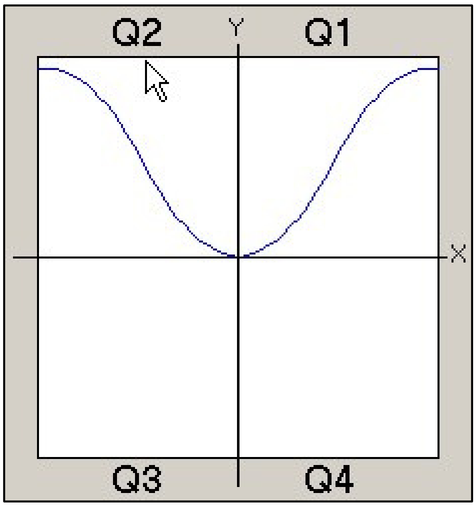
<u>Condition:</u>
P1-X < P2-X < P3X < … < P30-X)



Screenshot from PC programm DST80

4.22.2 mode 1 "Mirror Y-axis"

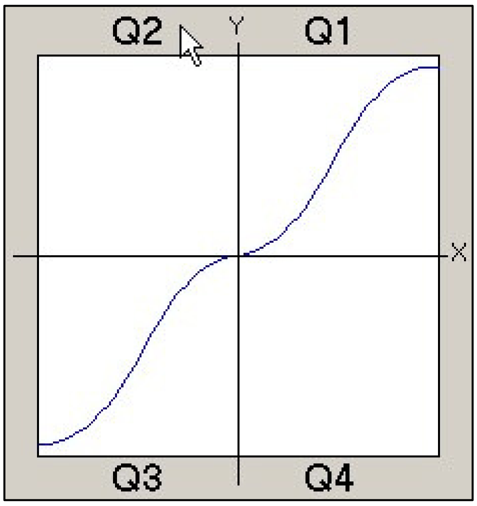
In this mode, the entered curve is mirrored and copied over the Y-axis, so that this curve is the same for the negative X-axis values. Q1 is the entered curve.



Conditions:
P1-X < P2-X < P3X < … < P30-X)
P1-X = 0

4.22.3 mode 2 "Mirror XY-axis"

In this mode, the entered curve is mirrored and copied over the X-axis and Y-axis, so that this curve is the same for the negative X-axis and Y-axis values. Q1 is the entered curve.



Conditions:
P1-X < P2-X < P3X < … < P30-X)
P1-X = 0
P1-Y = 0

5 Parameters

General lay-out:

PAR.	**PAR Nr:**	Possible values (bold is the standard value)
Basic description		
Description of possible values		

5.1 Menu 1 Config

PAR: 1.0.1	P[200]	**0** … 6
Basic function		
0 = standard 1 = Hold/Reset K0 L →H (rising edge) 2 = Hold/Reset K0 H →L (falling edge) 3 = Hold/Reset start/stop H 4 = Hold/Reset start/stop L 5 = Hold/Reset start/stop L →H (rising edge) 6 = Hold/Reset start/stop H →L (falling edge) (1 … 6 only possible for counter and frequency)		

PAR: 1.0.2	P[201]	**0** … 4
Input for actual position		
0 = Counter 1 = SSI 2 = Internal frequency 3 = Parallel (through data module) 4 = CAN-bus AP-Link 5 = Start/Stop		

PAR: 1.0.3	P[088]	0 … **40** … 2500
Measuring time velocity [AWE/s] (equal to refreshment time display)		
X.XXX (sec) input 0 .. 1.000s		

PAR: 1.0.4	P[202]	0 … **10** … 20
Integrator velocity Actual velocity is the average from the number of measurements		
0 = not active 1…20 number of measurements		

PAR: 1.0.5	P[203]	**0** … 6
Number of decimals		
0 = none 1 = X.X 2 = X.XX 3 = X.XXX 4 = X.XXXX 5 = X.XXXXX 6 = X.XXXXXX		

PAR: 1.0.6	P[204]	**0** … 3
Store function		
0 = no function 1 = only display 2 = only data output 3 = display + data output		

PAR: 1.0.7	P[205]	**0** ... 2
Store signal		
0 = high active 1 = low active		

PAR: 1.0.8	P[206]	**0** … 1
Power failure protection (no function when Hold/Reset is active (P[200])		
0 = not active 1 = active		

PAR: 1.0.9	P[207]	**0** … 123
Service functions Only possible to activate through keyboard AP80		
0 = not active 123 = set default parameters (automatically reset to 0)		

PAR: 1.0.10	P[208]	**0** … 2
Default monitor function Determines the default which is visible after start-up.		
0 = Actual position 1 = Actual velocity 2 = Actual value DA 3 = Actual position linearization		

5.2 Menu 2 Actual

5.2.1 Submenu 2.1 Counter

PAR: 2.1.1	P[210]	**0** … 1
Signal type and edge multiplication "S-signal X2": K2 is counter and K1 is direction		
0 = V-signal X4 1 = S-signal X2		

PAR: 2.1.2	P[211]	**0** … 1
Counting direction		
0 = no reverse 1 = reverse		

PAR: 2.1.3	P[000]	0 … **10000** … 16777215
Multiplicator numerator		
XXXXXXXX		

PAR: 2.1.4	P[001]	0 … **10000** … 16777215
Multiplicator denominator		
XXXXXXXX		

PAR: 2.1.5	P[212]	**0** … 5
Reference fine (input K0)		
0 = no function 1 = rising edge 2 = falling edge 3 = front keys 4 = rising edge (referece value thumbwheel-1) 5 = falling edge (referece value thumbwheel-1) 6 = rising edge input (1…6)		

PAR: 2.1.6	P[213]	**0** … 2
Reference coarse		
0 = no function 1 = high signal 2 = low signal		

PAR: 2.1.7	P[209]	**0** … 2
Counting direction for setting reference value		
0 = independant from direction 1 = when counting in positive direction 2 = when counting in negative direction		

PAR: 2.1.8	P[003]	-9999999 … **0** … 99999999
Reference value 1		
-XXXXXXXX (AWE)		

PAR: 2.1.9	P[004]	**0** … 99999999
Counting range (no function when Hold/Reset is active (P[200])		
XXXXXXXX (AWE)		

PAR: 2.1.10	P[104]	-9999999 … **0** … 99999999
Reference value 2		
-XXXXXXXX (AWE)		

PAR: 2.1.11	P[233]	**0** … 48
Source for counting range		
0 = parameters counting range P[004] 1…48 = nominal value 1…48		

5.2.2 Submenu 2.2 SSI

PAR: 2.2.1	P[214]	**0** … 1
SSI code		
0 = gray 1 = binary		

PAR: 2.2.2	P[211]	**0** … 1
Counting direction		
0 = no reverse 1 = reverse		

PAR: 2.2.3	P[216]	0 … **24** … 30
Number of SSI clockpulses		
XX		

PAR: 2.2.4	P[217]	0 … **24** … 30
Number of SSI databits		
XX		

PAR: 2.2.5	P[000]	0 … **10000** … 16777215
Multiplicator numerator		
XXXXXXXX		

PAR: 2.2.6	P[001]	0 … **10000** … 16777215
Multiplicator denominator		
XXXXXXXX		

PAR: 2.2.7	P[218]	**0** … 9
Adjustment		
0 = no function 1 = rising edge (K0) only temporary in RAM (on interrupt) 2 = falling edge (K0) only temporary in RAM (on interrupt) 3 = set with parameter adjustment value 4 = rising edge (K0) permanent in EEPROM (not on interrupt) 5 = rising edge (K0) only temporary in RAM (on interrupt) (adjustment value on external data input) 6 = falling edge (K0) only temporary in RAM (on interrupt) (adjustment value on external data input) 7 = front keys 8 = rising edge (K0) reference value thumbwheel-1 9 = faling edge (K0) reference value thumbwheel-1 10 = rising edge input (1…6)		

PAR: 2.2.8	P[213]	**0** … 2
Reference coarse		
0 = no function 1 = high signal 2 = low signal		

PAR: 2.2.9	P[209]	**0** … 2
Counting direction adjustment		
0 = independant from direction 1 = when counting in positive direction 2 = when counting in negative direction		

PAR: 2.2.10	P[002]	-9999999 … **0** … 99999999
Offset		
-XXXXXXXX (AWE)		
PAR: 2.2.11	P[005]	-9999999 … **0** … 99999999
Adjustment value		
-XXXXXXXX (AWE) input 0 means function inactive		

PAR: 2.2.12	P[006]	1 … **50** … 99999
Monitoring delta-SSI per cycletime (250 µs) Unprocessed SSI-value, only depending on number of active SSI-databits P[217]		
XXXXX		

PAR: 2.2.13	P[220]	0 … **2** … 9
Maximum number of SSI errors to set external SSI-error. At every SSI-error the actual value is determined by interpolation of the last valid value		
X		

PAR: 2.2.14	P[221]	0 … **3**
SSI monitoring		
0 = not active 1 = only wiring 2 = only Delta SSI monitoring 3 = wiring + Delta SSI monitoring		

5.2.3 Submenu 2.3 Internal frequency

PAR: 2.3.1	P[224]	**0** … 3
Timebasis		
0 = 78,125 kHz 1 = 625 kHz 2 = 5 MHz 3 = 10 MHz		

PAR: 2.3.2	P[211]	**0** … 1
Counting direction		
0 = positive 1 = negative		

PAR: 2.3.3	P[000]	0 … **10000** … 16777215
Multiplicator numerator		
XXXXXXXX		

PAR: 2.3.4	P[001]	0 … **10000** … 16777215
Multiplicator denominator		
XXXXXXXX		

PAR: 2.3.5	P[212]	**0** … 5
Reference fine (input K0)		
0 = no function 1 = rising edge 2 = falling edge 3 = front keys 4 = rising edge (K0) reference value thumbwheel-1 5 = faling edge (K0) reference value thumbwheel-1 6 = rising edge input (1…6)		

PAR: 2.3.6	P[213]	**0** … 2
Reference coarse		
0 = no function 1 = high signal 2 = low signal		

PAR: 2.3.7	P[209]	**0** … 2
Counting direction for setting reference value		
0 = independant from direction 1 = upwards counting 2 = downwards counting		

PAR: 2.3.8	P[003]	-9999999 … **0** … 99999999
Reference value 1		
-XXXXXXXX (AWE)		

PAR: 2.3.9	P[004]	**0** … 99999999
Counting range (no function when Hold/Reset is active (P[200])		
XXXXXXXX (AWE)		

PAR: 2.3.10	P[233]	**0** … 48
Source for counting range		
0 = parameters counting range P[004] 1…48 = nominal value 1…48		

5.2.4 Submenu 2.4 Parallel

PAR: 2.4.1	P[380]	**0** … 5
Parallel code		
0 = Binary 1 = BCD 2 = Gray 3 = Binary inverse 4 = BCD inverse 5 = Gray inverse		

PAR: 2.4.2	P[211]	**0** … 1
Counting direction		
0 = no reverse 1 = reverse		

PAR: 2.4.3	P[379]	0 … **24**
Number of active bits		
XX		

PAR: 2.4.4	P[000]	0 … **10000** … 16777215
Multiplicator numerator		
XXXXXXXX		

PAR: 2.4.5	P[001]	0 … **10000** … 16777215
Multiplicator denominator		
XXXXXXXX		

PAR: 2.4.6	P[002]	-9999999 … **0** … 99999999
Offset		
-XXXXXXXX (AWE)		

PAR: 2.4.7	P[005]	-9999999 … **0** … 99999999
Adjustment value		
-XXXXXXXX (AWE) input 0: function is inactive		

5.2.5 Submenu 2.5 CAN AP-Link

PAR: 2.5.1	P[103]	**0** … 99999999
Display counting range		
XXXXXXXX (AWE)		

PAR: 2.5.2	P[079]	**0** … 9999
Time-out in 0,01 s units		
XX.XX (sec) input 0: inactive		

PAR: 2.5.3	P[215]	**0** … 1
Time-out reset		
0 = auto reset (reset when new value has been stored) 1 = reset by using an input		

5.2.6 Submenu 2.6 Start/Stop

PAR: 2.6.1	P[115]	0 … **280000** .. 999999
Gradient Actual signalspeed sensor internal in m/s (0 = 2800.00 m/s)		
XXXX.XX (m/s)		

PAR: 2.6.2	P[118]	0 … **6000** … 29999
Measuring length sensor		
XXXXX mm		

PAR: 2.6.3	P[211]	**0** … 1
Counting direction		
0 = no reverse 1 = reverse		

PAR: 2.6.4	P[000]	0 … **10000** … 16777215
Multiplicator numerator		
XXXXXXXX		

PAR: 2.6.5	P[001]	0 … **10000** … 16777215
Multiplicator denominator		
XXXXXXXX		

PAR: 2.6.6	P[219]	**0** … 7
Adjustment		
0 = no function 1 = rising edge (K0) only temporary in RAM (on interrupt) 2 = falling edge (K0) only temporary in RAM (on interrupt) 3 = set with parameter adjustment value 4 = rising edge (K0) permanent in EEPROM (not on interrupt) 5 = front keys 6 = rising edge (K0) reference value thumbwheel-1 7 = faling edge (K0) reference value thumbwheel-1 8 = rising edge input (1…6)		

PAR: 2.6.7	P[213]	**0** … 2
Reference coarse		
0 = no function 1 = high signal 2 = low signal		

PAR: 2.6.8	P[209]	**0** … 2
Counting direction adjustment		
0 = independant from direction 1 = when counting in positive direction 2 = when counting in negative direction		

PAR: 2.6.10	P[002]	-9999999 … **0** … 99999999
Offset		
-XXXXXXXX (AWE)		

PAR: 2.6.11	P[005]	-9999999 … **0** … 99999999
Adjustment value		
-XXXXXXXX (AWE) input 0 means function inactive		

5.3 Menu 3 CAN-bus

5.4 Submenu 3.1 Config

PAR: 3.1.1	P[228]	0 … **5** … 7
Baudrate		
0 = 20 kbit/s 1 = 50 kbit/s 2 = 100 kbit/s 3 = 125 kbit/s 4 = 250 kbit/s 5 = 500 kbit/s 6 = 800 kbit/s 7 = 1 Mbit/s		

5.5 Submenu 3.2 Obj1/PDO1 In

PAR: 3.2.1	P[089]	0 … **1** … 127
CAN adress Obj/PDO1 In		
XXX		

PAR: 3.2.2	P[229]	**0** … 1
Function Obj/PDO1 In		
0 = not active 1 = AP-Link (reading actual position and velocity)		

5.6 Submenu 3.3 Obj1/PDO1 Out

PAR: 3.3.1	P[090]	0 … **1** … 127
CAN adress Obj/PDO1 Out		
XXX		

PAR: 3.3.2	P[230]	**0** … 4
Function Obj/PDO1 Out		
0 = not active 1 = AP-Link (sending actual position and velocity)		

5.7 Menu 4 Serial

5.7.1 Submenu 4.1 Config

PAR: 4.1.1	P[236]	**0** … 31
Unit adress		
XX		

5.7.2 Submenu 4.2 Ser-1 (RS232)

PAR: 4.2.1	P[237]	0 … **1** … 4
Baudrate		
0 = 9600 1 = 19200 2 = 28800 3 = 38400 4 = 57600		

PAR: 4.2.2	P[238]	**0** … 1
Number of stopbits		
0 = 1 Stopbit 1 = 2 Stopbits		

PAR: 4.2.3	P[239]	0 … **2**
Parity		
0 = none 1 = odd 2 = even		

PAR: 4.2.4	P[240]	0 … **1**
Protocol		
0 = no function 1 = ASCII		

5.7.3 Submenu 4.3 Ser-2 (RS422/485)

PAR: 4.3.1	P[242]	0 …**1** … 4
Baudrate		
0 = 9600 1 = 19200 2 = 28800 3 = 38400 4 = 57600		

PAR: 4.3.2	P[243]	**0** … 1
Number of stopbits		
0 = 1 Stopbit 1 = 2 Stopbits		

PAR: 4.3.3	P[244]	0 … **2**
Parity		
0 = none 1 = odd 2 = even		

PAR: 4.3.4	P[245]	**0** … 1
Protocol		
0 = no function 1 = ASCII		

5.8 Menu 5 Input

INPUT-1

PAR: 5.0.1	P[249]	**0** … 16
Function input-1		
0 = no function 1 = coarse 2 = store 3 = enable 4 = error reset SSI 5 = start/stop cams 6 = start cams 7 = stop cams 8 = Hold/Reset start/stop 9 = Hold/Reset start 10 = Hold/Reset stop 11 = lock input nominal values 12 = lock input parameters 13 = lock input nominal values + parameters 14 = reset time-out AP-Link 15 = select reference value 1 or 2 (low = 1) 16 = read external thumbwheel sets 17 = SetRef/Adjustment		

INPUT-2

PAR: 5.0.2	P[250]	**0** … 16
Function input-2		
XX (see input-1)		

INPUT-3

PAR: 5.0.3	P[251]	**0** … 16
Function input-3		
XX (see input-1)		

INPUT-4

PAR: 5.0.4	P[252]	**0** … 16
Function input-4		
XX (see input-1)		

INPUT-5

PAR: 5.0.5	P[253]	**0** … 16
Function input-5		
XX (see input-1)		

INPUT-6

PAR: 5.0.6	P[254]	**0** … 16
Function input-6		
XX (see input-1)		

5.9 Menu 6 Output

5.9.1 Submenu 6.1 – 6.9 Op1…9

OUTPUT 1…9

PAR: 6.x.1	P[389]…P[397]	**0** … 8
Function output-1		
0 = cam 1 = cam inverted 2 = SSI error (high = no error) 3 = reference/adjustment set 4 = cams active 5 = Hold/Reset started (only when P[200] = 3 …6) 6 = time-out AP-Link (high = no time-out) 7 = ASCII protocol 8 = Counting direction (high = downwards counting) 9 = Start/Stop error (no magnet or time-out)		

PAR: 6.x.2	P[094]…P[102]	**0** … 5000
Dynamic cams (only when output = cam and source = actual position)		
X.XXX (sec) input 0 = no function		

5.10 Menu 7 Data

5.10.1 Submenu 7.1 Data in

PAR: 7.1.1	P[379]	**0** … 24
Number of active bits		
XX		

PAR: 7.1.2	P[380]	**0** … 5
Data input code		
0 = Binary 1 = BCD 2 = Gray 3 = Binary inverse 4 = BCD inverse 5 = Gray inverse		

PAR: 7.1.3	P[381]	**0** … 1
Minus-sign data output		
0 = no function 1 = minus-sign on pin 25 2 = minus-sign on pin 12		

PAR: 7.1.4	P[382]	**0** … 2
Data valid data input		
0 = no function 1 = data valid on pin 25		

PAR: 7.1.5	P[378]	**0** … 1
Function data input When set to inactive (0) the data input can be used as input for the actual position (parallel encoders)		
0 = Inactive 1 = Active		

5.10.2 Submenu 7.2 Data out

PAR: 7.2.1	P[384]	**0** … 5
Data output code		
0 = Binary 1 = BCD 2 = Gray 3 = Binary inverse 4 = BCD inverse 5 = Gray inverse		

PAR: 7.2.2	P[385]	**0** … 1
Minus-sign data output AP80-XXP minus-sign on pin 25 AP80-XXF minus-sign on pin 35		
0 = no function 1 = minus-sign on pin 25/35 H = sign active 2 = minus-sign on pin 25/35 L = sign active		

PAR: 7.2.3	P[386]	**0** … 2
Data ready output AP80-XXP data valid on pin 25 or 12 AP80-XXF data valid on pin 35 or 16		
0 = no function 1 = data valid on pin 25/35 2 = data valid on pin 12/16		

PAR: 7.2.4	P[387]	**0** … 3
Source for data output When set to 3 the parameters [384], [385] and [386] have no function		
0 = no data output 1 = actual position 2 = actual velocity 3 = ASCII protocol 4 = actual position linearization		

PAR: 7.2.5	P[376]	**0** ... 1
Enable data output		
0 = low activ 1 = high activ		

5.11 Menu 8 Analog

5.11.1 Submenu 8.1 Config

PAR: 8.1.1	P[388]	**0** … 2
Selection DA output		
0 = inactive 1 = voltage 2 = current		

PAR: 8.1.2	P[383]	**0** … 1
Selection DA source		
0 = actual position 1 = actual velocity 2 = actual position linearization 3 = actual |velocity| abs		

5.11.2 Submenu 8.2 DA-U (voltage)

DA PAR 8.2.1…8.2.4 = 0: DA not active

PAR: 8.2.1	P[080]	**-100000** … 99999
Umin DA		
-XX.XXXX (V)		

PAR: 8.2.2	P[081]	-99999 … **100000**
Umax DA		
-XX.XXXX (V)		

PAR: 8.2.3	P[082]	-9999999… **-100000** … 99999999
S-Umin DA		
-XXXXXXXX (AWE)		

PAR: 8.2.4	P[083]	-9999999 … **100000** … 99999999
S-Umax DA		
-XXXXXXXX (AWE)		

5.11.3 Submenu 8.3 DA-I (current)

DA PAR 8.2.1…8.2.4 = 0: DA not active

PAR: 8.3.1	P[084]	**-200000** … 199999
Imin DA		
-XX.XXXX (mA)		

PAR: 8.3.2	P[085]	-199999 … **200000**
Imax DA		
-XX.XXXX (mA)		

PAR: 8.3.3	P[086]	-9999999 … **-200000** …99999999
S-Imin DA		
-XXXXXXXX (AWE)		

PAR: 8.3.4	P[087]	-9999999 … **200000** … 99999999
S-Imax DA		
-XXXXXXXX (AWE)		

5.12 Menu 9 Cam

5.12.1 Submenu 9.1 … 9.24 CA1…24

CAM-1…24

PAR: 9.x.1	P[256]...P[279]	**0** … 3
Cam function		
0 = no function 1 = range 2 = actual position >= limit value 3 = actual position <= limit value		

PAR: 9.x.2	P[280]…P[303]	**0** … 4
Source		
0 = actual position 1 = actual velocity 2 = external data input – actual position 3 = Thumbwheel-1 actual position (relative cams) 4 = Thumbwheel-2 actual position (relative cams) 5 = actual linearization		

PAR: 9.x.3	P[304]…P[327]	**0** … 52
Source cam begin / limit value (limit value if cam function = 2 or 3)		
0 = parameters cam begin 1…48 = Nominal value 1…48 49…52 = Thumbwheel sets 1…4		

PAR: 9.x.4	P[328]…P[351]	**0** … 52
Source cam end		
0 = parameters cam end 1…48 = Nominal value 1…48 49…52 = Thumbwheel sets 1…4		

PAR: 9.x.5	P[007]…P[030]	-9999999 … **1000** … 99999999
Cam begin / limit value (limit value if cam function = 2 or 3)		
-XXXXXXX		

PAR: 9.x.6	P[031]…P[054]	-9999999 … **2000** …99999999
Cam end		
-XXXXXXX		

PAR: 9.x.7	P[055]…P[078]	**0** … 999999
Hysteresis cam		
XXXXXX		

PAR: 9.x.8	P[352]…P[375]	**0** … 9
Assign cam to output		
0 = no output 1…9 = output 1-9		

5.13 External thumbwheel sets

5.13.1 Submenu 10.1 Config

PAR: 10.1.1	P[222]	**0** … 4
Number of thumbwheel sets		
0 = inactive 1 = DW1 2 = DW1+DW2 3 = DW1+DW2+DW3 4 = DW1+DW2+DW3+DW4		

PAR: 10.1.2	P[223]	**0** … 1
Reading		
0 = auto (iedere 100ms) 1 = via ext. Ingang		

5.13.2 Submenu 10.2 …10.4 Set-1...4

PAR: 10.x.1	P[110]…P[113]	**0** … 12
Format		
0 = 7654321 (without sign) 1 = x654321 (without sign) 2 = xx54321 (without sign) 3 = xxx4321 (without sign) 4 = xxxx321 (without sign) 5 = xxxxx21 (without sign) 6 = xxxxxx1 (without sign) 7 = -654321 (with sign) 8 = x-54321 (with sign) 9 = xx-4321 (with sign) 10= xxx-321 (with sign) 11= xxxx-21 (with sign) 12= xxxxx-1 (with sign)		

PAR: 10.x.2	P[120]…P[123]	**-999999** … 9999999
min Definition of minimum value.		
-XXXXXXX		

PAR: 10.x.3	P[124]…P[127]	-999999 … **9999999**
Max Definition of maximum value.		
-XXXXXXX		

PAR: 10.x.4	P[128]…P[131]	**0** … 2
Factor		
0 = x1 1 = x10 2 = x100		

5.14 Menu 11 Linearization

5.14.1 Submenu 11.1 Config

PAR: 11.1.1	P[246]	**0** … 1
Source		
0 = inactive 1 = actual position		

PAR: 11.1.2	P[247]	2 … **10**… 30
Number of points		
XX		

PAR: 11.1.3	P[248]	**0** … 6
Number of decimals for actual position linearization		
0 = geen 1 = X.X 2 = X.XX 3 = X.XXX 4 = X.XXXX 5 = X.XXXXX 6 = X.XXXXXX		

PAR: 11.1.4	P[241]	**0** … 2
Mode		
0 = 4-Quadrant 1 = Mirror X-as 2 = Mirror XY-as		

5.14.2 Submenu 11.2 … 11.31 P1…P30

P1…30 ➔ x = 2…31

PAR: 11.x.1	P[140]…P[169]	-9999999 … **0** … 99999999
P1-X		
-XXXXXXX		

PAR: 11.x.2	P[170]…P[199]	-9999999 … **0** … 99999999
P1-Y		
-XXXXXXX		

5.15 Overview parameters

No	Description	Menu
[000]	= Multiplicator numerator	2.1.3/2.2.5/2.3.3/2.4.4/2.6.4
[001]	= Multiplicator denominator	2.1.4/2.2.6/2.3.4/2.4.5/2.6.5
[002]	= Offset	2.2.10/2.4.6/2.6.9
[003]	= Reference value 1	2.1.8/2.3.8
[004]	= Counting range	2.1.9/2.3.9
[005]	= Adjustment value	2.2.11/2.4.7/2.6.10
[006]	= Delta-SSI monitoring per cycle	2.2.12
[007]…[030]	= Cam begin / limit value	9.1.5…9.24.5
[031]…[054]	= Cam end	9.1.6…9.24.6
[055]…[078]	= Hysteresis cam	9.1.7…9.24.7
[079]	= Time-out in 0,01s units (APLink)	2.5.2
[080]	= Umin DA	8.2.1
[081]	= Umax DA	8.2.2
[082]	= S-Umin DA	8.2.3
[083]	= S-Umax DA	8.2.4
[084]	= Imin DA	8.3.1
[085]	= Imax DA	8.3.2
[086]	= S-Imin DA	8.3.3
[087]	= S-Imax DA	8.3.4
[088]	= Measuring time velocity	1.0.3
[089]	= CAN adress Obj/PDO1 In	3.2.1
[090]	= CAN adress Obj/PDO1 Out	3.3.1
[091]…[093]	= no function	
[094]…[102]	= Dynamic cam output 1…9	6.1.2…6.9.2
[103]	= Display counting range (APLink)	2.5.1
[104]	= Reference value 2	2.1.10
[105]…[109]	= no function	
[110]…[113]	= formaat duimwiel sets	10.2.1…10.5.1
[114]	= no function	
[115]	= gradient Start/Stop sensor	2.6.1
[116],[117]	= no function	
[118]	= measuring length Start/Stop sensor	2.6.2
[119]	= geen functie	
[120]…[123]	= min value thumbwheel sets	10.2.2…10.5.2
[124]…[127]	= max value thumbwheel sets	10.2.3…10.5.3
[128]…[131]	= factor thumbwheel sets	10.2.4…10.5.4
[132]…[139]	= no function	
[140]…[169]	= Pn-X (linearization)	11.2.1…11.31.1
[170]…[199]	= Pn-Y (linearization)	11.2.2…11.31.2

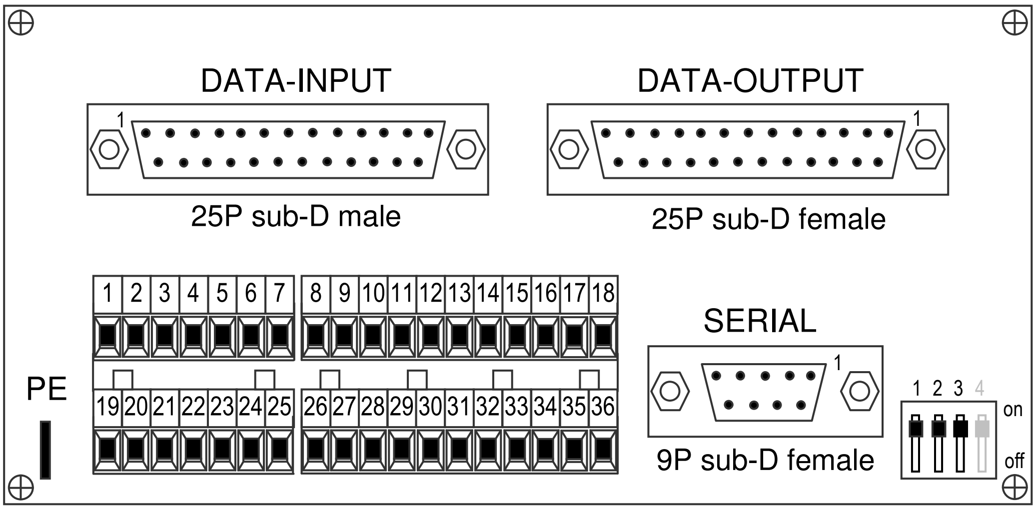
No	Description	Menu
[200] =	Basic function	1.0.1
[201] =	Input for actual position	1.0.2
[202] =	Integrator velocity	1.0.4
[203] =	Number of decimals	1.0.5
[204] =	Store function	1.0.6
[205] =	Store signal	1.0.7
[206] =	Power failure protection	1.0.8
[207] =	Service functions	1.0.9
[208] =	Default monitor function	1.0.10
[209] =	Counting direction ref. / zero-p. adjustment	2.1.7/2.2.9/2.3.7/2.6.8
[210] =	Input type and edge multiplication	2.1.1
[211] =	Counting direction	2.1.2/2.2.2/2.3.2/2.4.2/2.6.3
[212] =	Reference fine (input K0)	2.1.5/2.3.5
[213] =	Reference coarse	2.1.6/2.2.8/2.3.6/2.6.7
[214] =	SSI code	2.2.1
[215] =	Time-out reset (APLink)	2.5.3
[216] =	Number of SSI clockpulses	2.2.3
[217] =	Number of SSI databits	2.2.4
[218] =	Adjustment (SSI)	2.2.7
[219] =	Adjustment (Start/Stop)	2.6.6
[220] =	Maximum number of SSI errors	2.2.13
[221] =	SSI monitoring	2.2.14
[222] =	number of thumbwheel sets	10.1.1
[223] =	reading thumbwheel sets	10.1.2
[224] =	Time basis (internal frequency)	2.3.1
[225]…[227] =	no function	
[228] =	Baudrate (Canbus)	3.1.1
[229] =	Function Obj/PDO1 In	3.2.2
[230] =	Function Obj/PDO1 Out	3.3.2
[231],[232] =	no function	
[233] =	Source for counting range	2.1.11/2.3.10
[234],[235] =	no function	
[236] =	Unit adress	4.1.1
[237] =	Baudrate (RS232)	4.2.1
[238] =	Stopbits (RS232)	4.2.2
[239] =	Parity (RS232)	4.2.3
[240] =	Protocol (RS232)	4.2.4
[241] =	Modus linearization	11.1.4
[242] =	Baudrate (RS422/485)	4.3.1
[243] =	Number of stopbits (RS422/485)	4.3.2
[244] =	Parity (RS422/485)	4.3.3

No	Description	Menu
[245]	= Protocol (R422/485)	4.3.4
[246]	= Source (linearization)	11.1.1
[247]	= Number of points (linearization)	11.1.2
[248]	= Number of decimals (linearization)	11.1.3
[249]…[254]	= Function input 1…6	5.0.1…5.0.6
[255]	= no function	
[256]…[279]	= Cam function	9.1.1…9.24.1
[280]…[303]	= Source cam	9.1.2…9.24.2
[304]…[327]	= Source for cam begin/limit value	9.1.3…9.24.3
[328]…[351]	= Source for cam end	9.1.4…9.24.4
[352]…[375]	= assign cam to output	9.1.8…9.24.8
[376]	= Enable data output	7.2.5
[377]	= no function	
[378]	= Function for data input	7.1.5
[379]	= Number of active bits data input	2.4.3/7.1.1
[380]	= Code for data input	2.4.1/7.1.2
[381]	= minus-sign data input	7.1.3
[382]	= Data valid data input	7.1.4
[383]	= Selection DA source	8.1.2
[384]	= Code for data output	7.2.1
[385]	= minus-sign for data output	7.2.2
[386]	= Data ready data output	7.2.3
[387]	= Source data output	7.2.4
[388]	= Selection DA output	8.1.1
[389]…[397]	= Function output 1…9	6.1.1…6.9.1
[398],[399]	= no function	

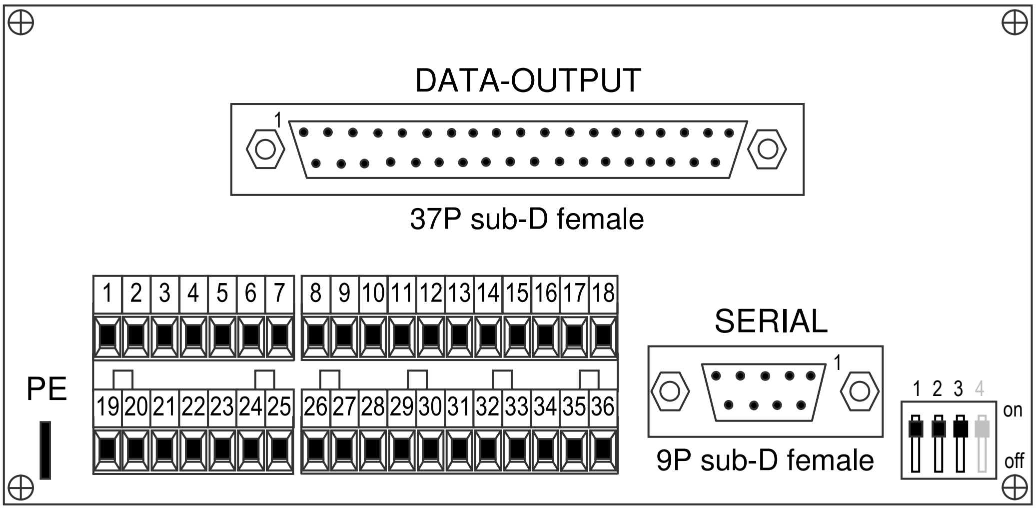
6 Connections

Connections on the rear

AP80-CXP



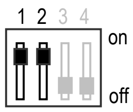
AP80-CXF



PE

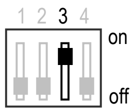
To be connected with a 6,3mm faston

RS 422/485



If the AP80 is the last device, the DIP-switches 1 and 2 should be set to on.

CANbus



If the AP80 is the last device in a CANbus network, DIP-switch 3 should be set to on.

(DIP-switch 4 has no function)

6.1 Overview clamp connections

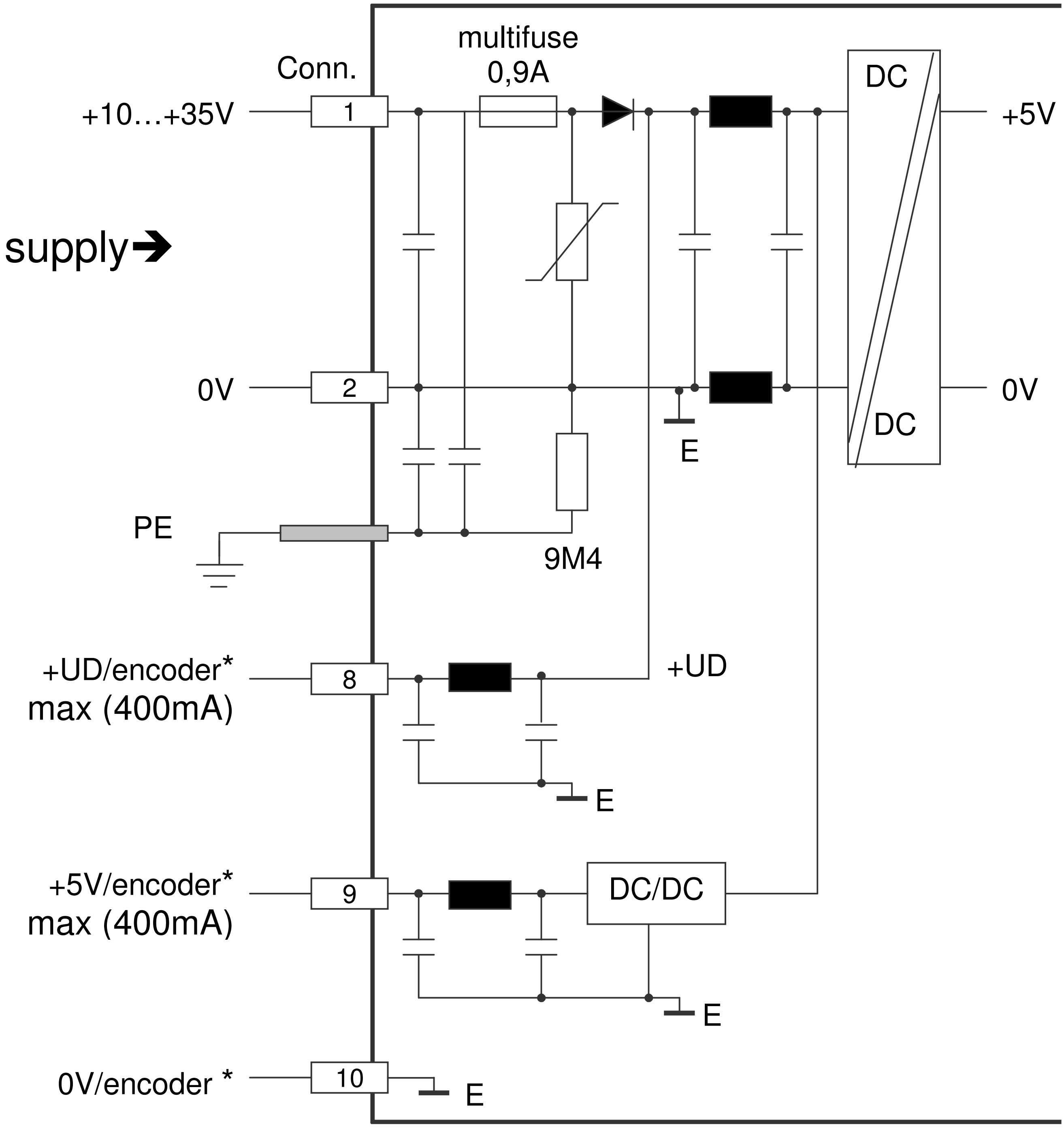
1. +10…+35V supply
2. 0V supply
3. +/-10V or +/-20mA analog output
4. common analog output
5. CAN-H
6. CAN-L
7. CAN 0V

8. +10…35V DC supply output for encoder
9. +5V DC supply output for encoder
10. 0V supply for encoder
11. SSI-Clock+ or start+ signal Start/Stop sensor
12. SSI-Clock- or start- signal Start/Stop sensor
13. K1 or counting dir. or SSI-Data+ or stop+ signal Start/Stop sensor
14. /K1 or counting dir. or SSI-Data- or stop- signal Start/Stop sensor
15. K2 or counting pulse
16. /K2 or counting pulse
17. K0
18. /K0

19. Input-1
20. Input-2
21. Input-3
22. Input-4
23. Input-5
24. Input-6
25. common for inputs

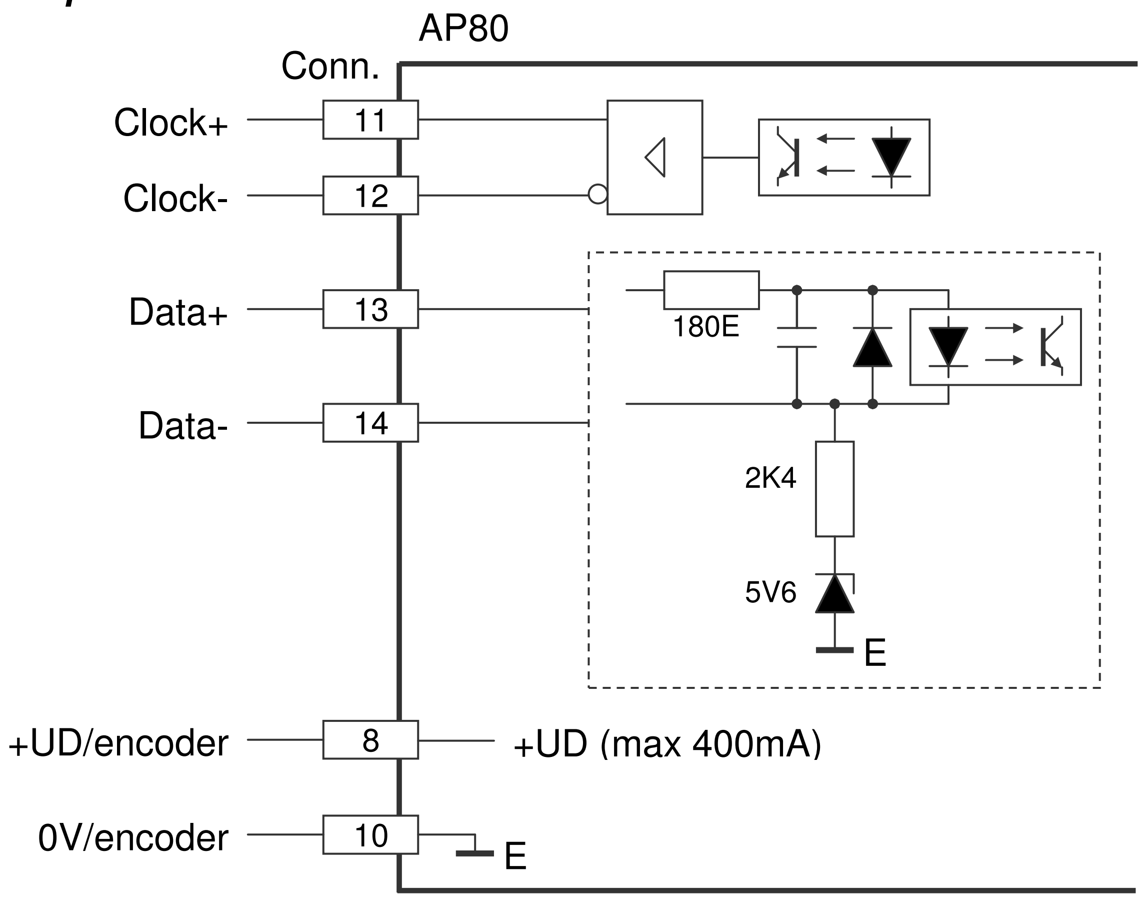
26. +U for outputs
27. 0V for outputs
28. Output -1
29. Output -2
30. Output -3
31. Output -4
32. Output -5
33. Output -6
34. Output -7
35. Output -8
36. Output -9

6.2 Supply

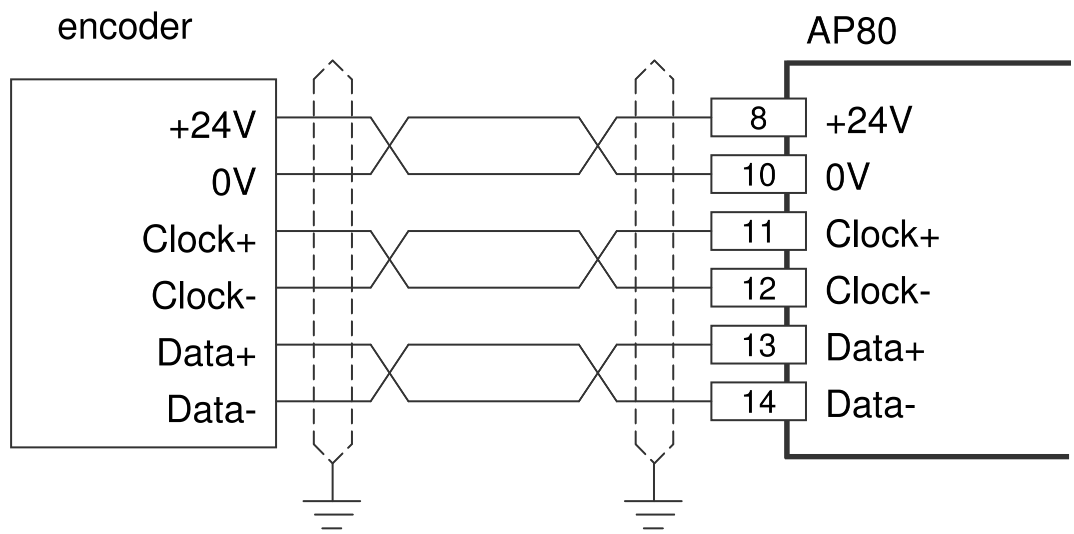


* supply output for encoder

6.3 SSI input

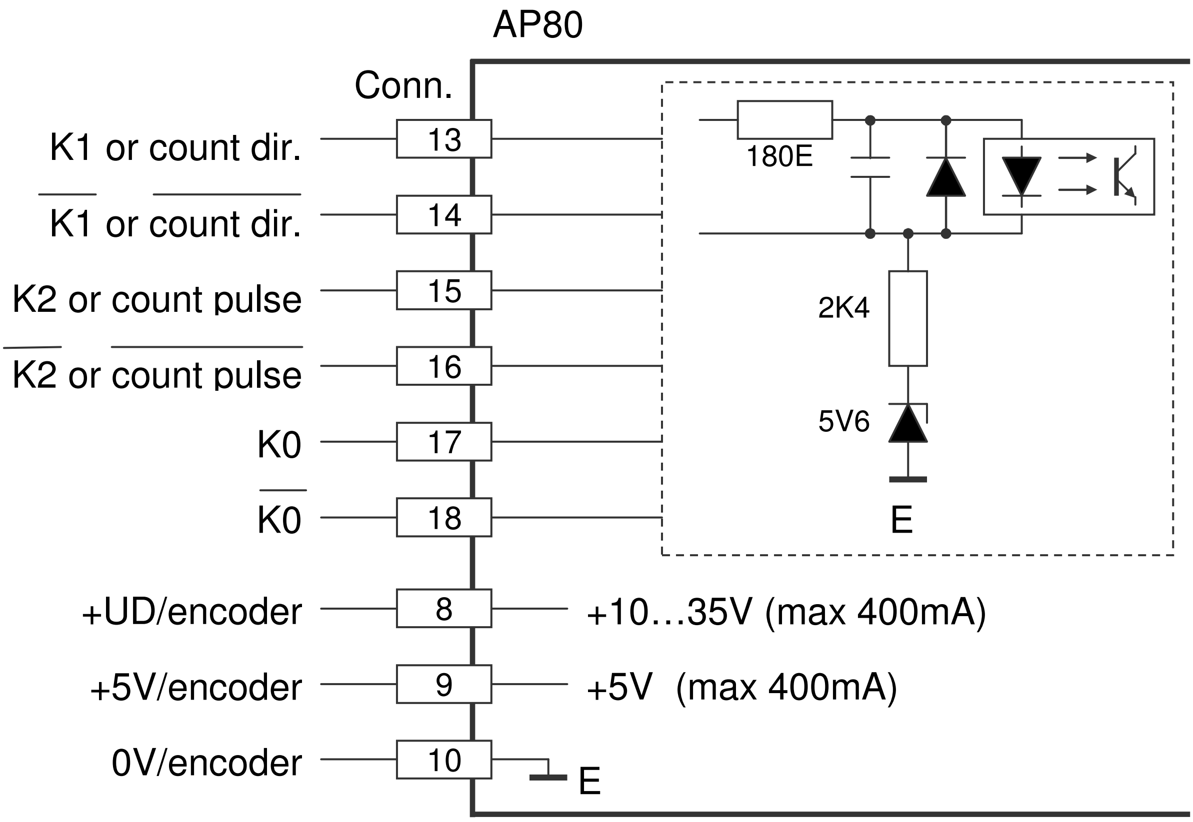


6.4 SSI encoder 24V



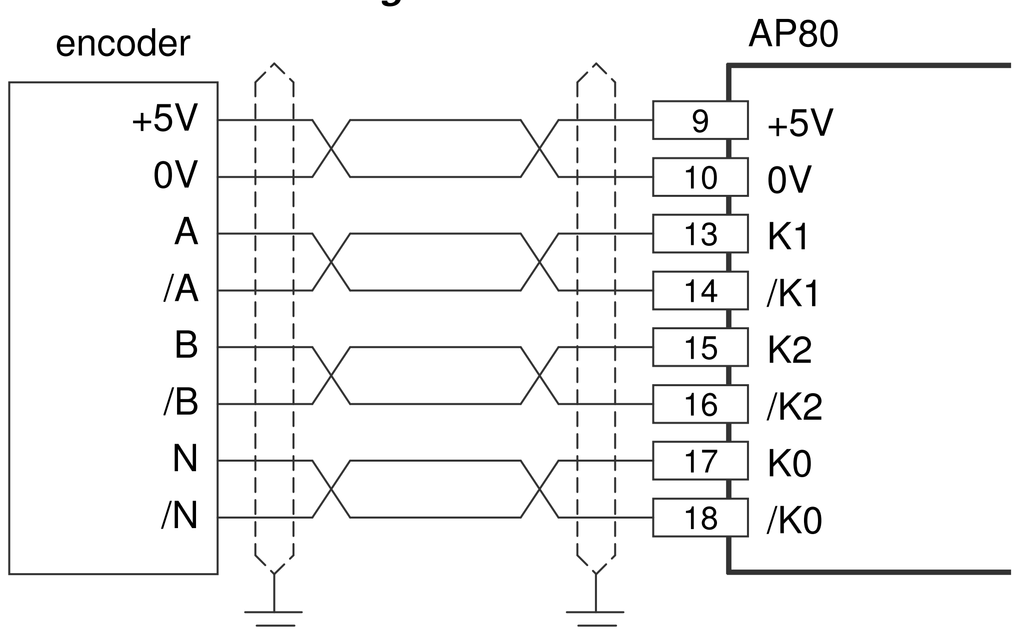
Supply voltage AP80 clamp 1 and 2 is 24V DC

6.5 Counting input

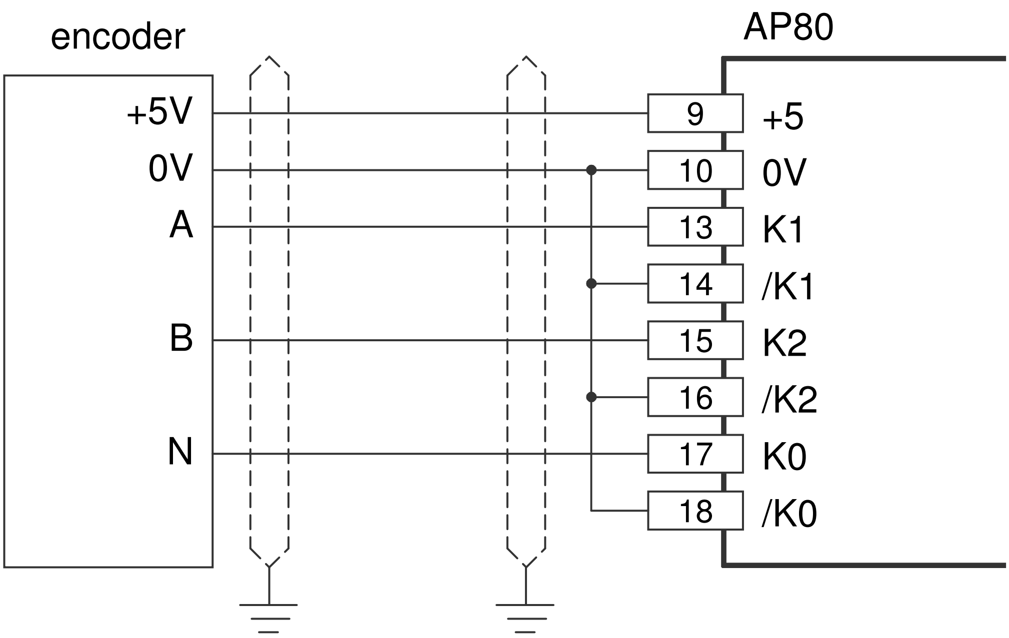


It is possible to use different voltage signals for K0 and K1, K2. For example: encoder signals (K1/K2) with a level of 5V and a reference fine (K0) with 24V level.

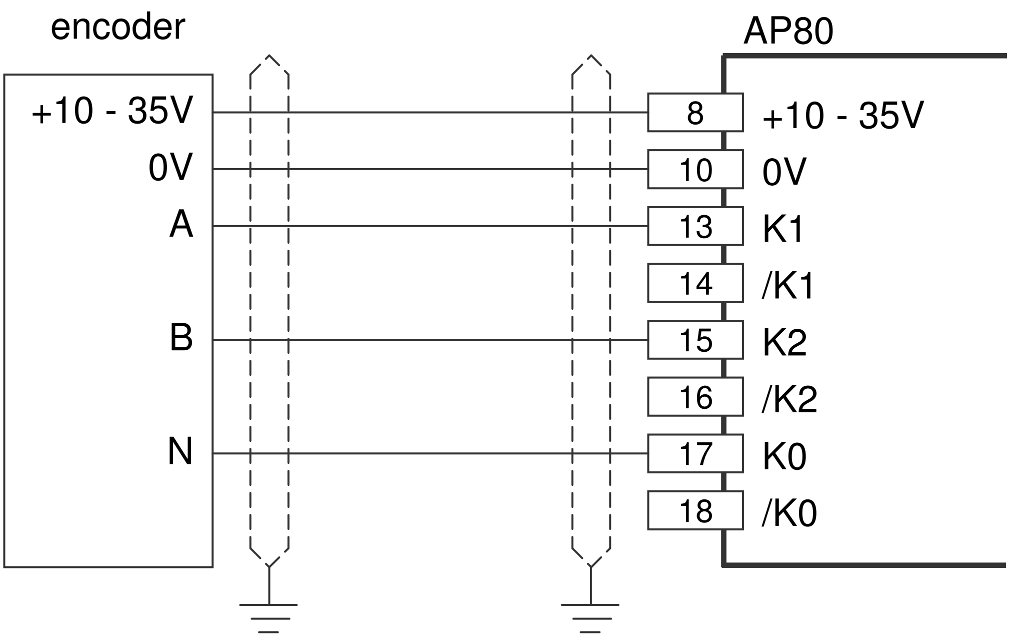
6.6 Encoder 5V with inverted signals



6.7 Encoder 5V without inverted signals

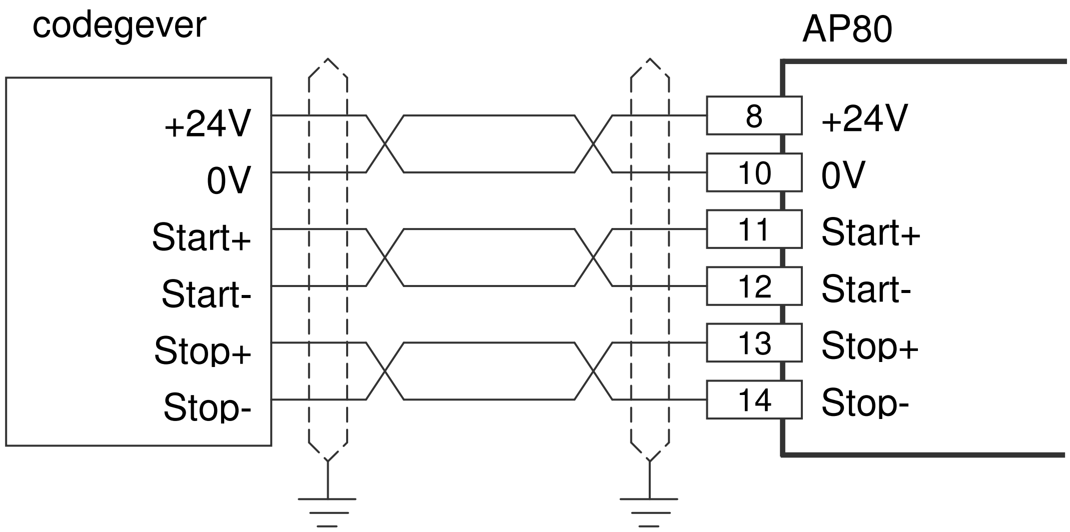


6.8 Encoder 10 – 30V



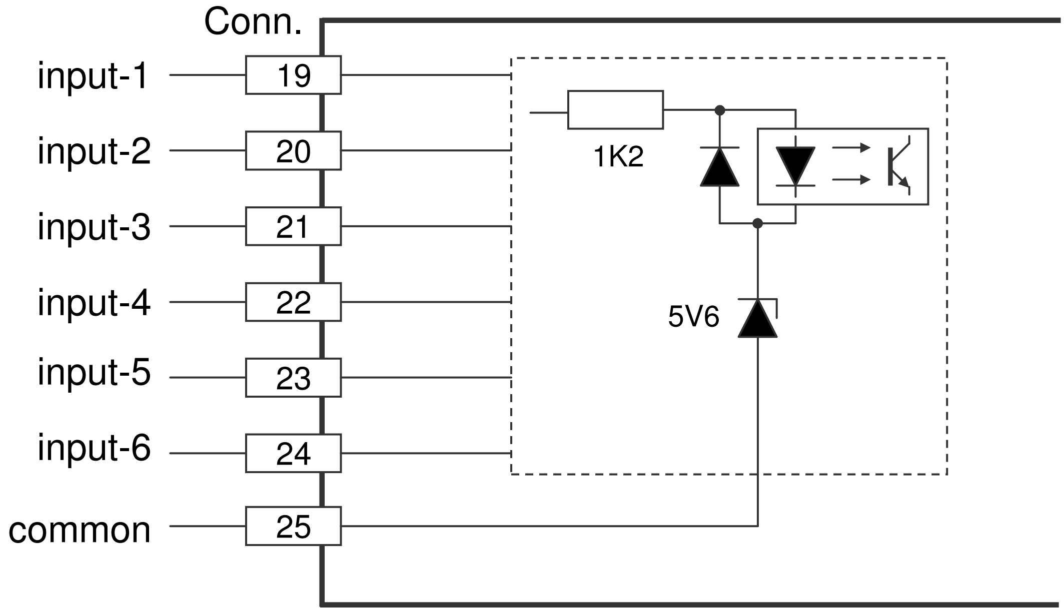
CAUTION!
When using 24V encoders don't connect terminal 14,16 and 18.

6.9 Start/Stop sensor

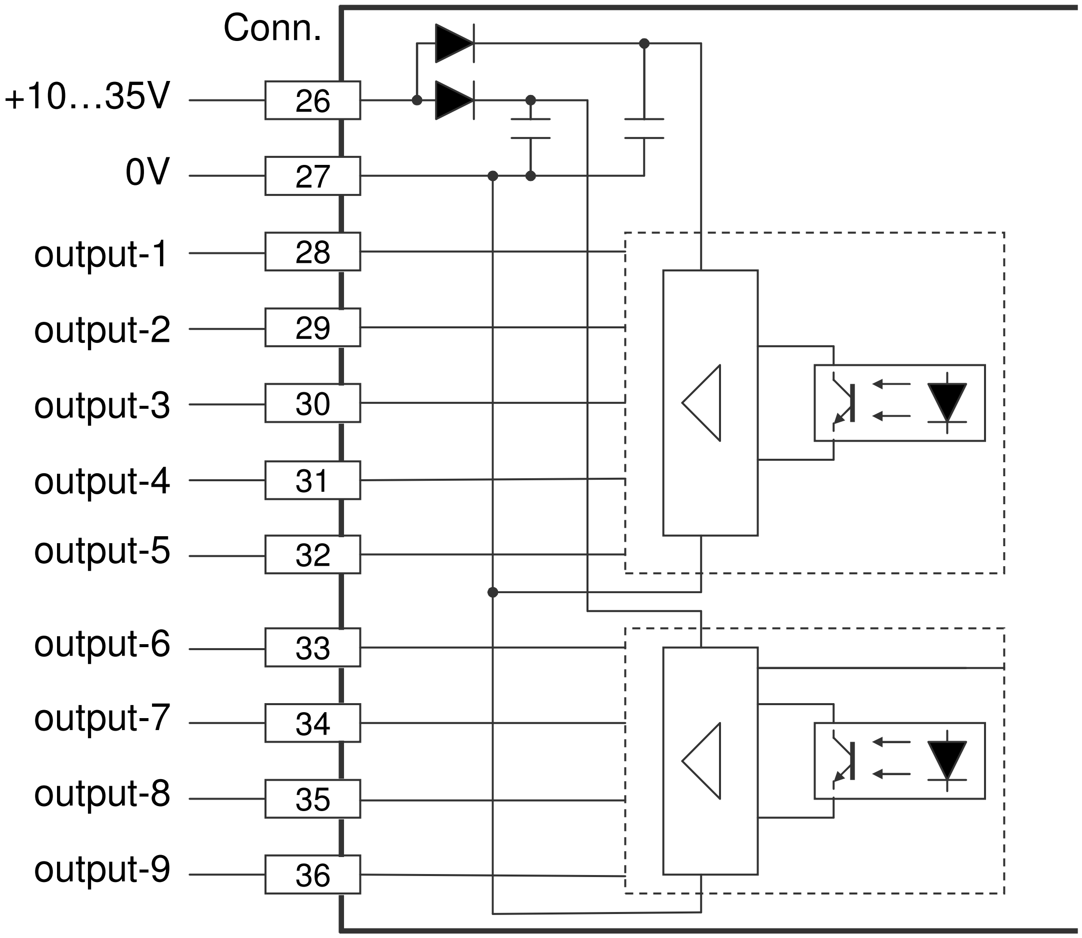


Supply voltage AP80 clamp 1 en 2 is 24V DC

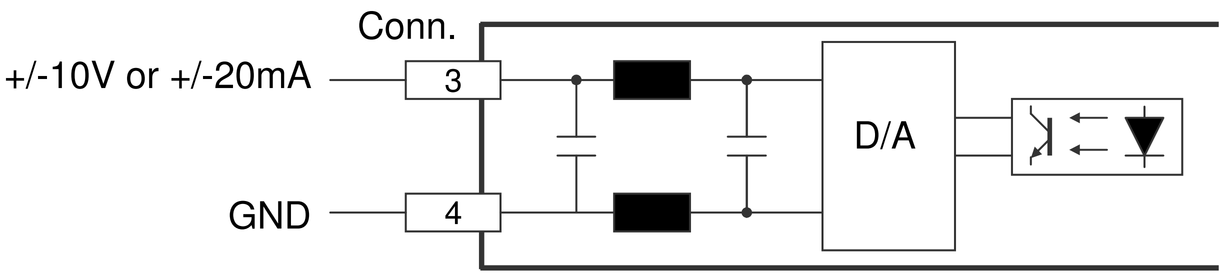
6.10 Digital inputs



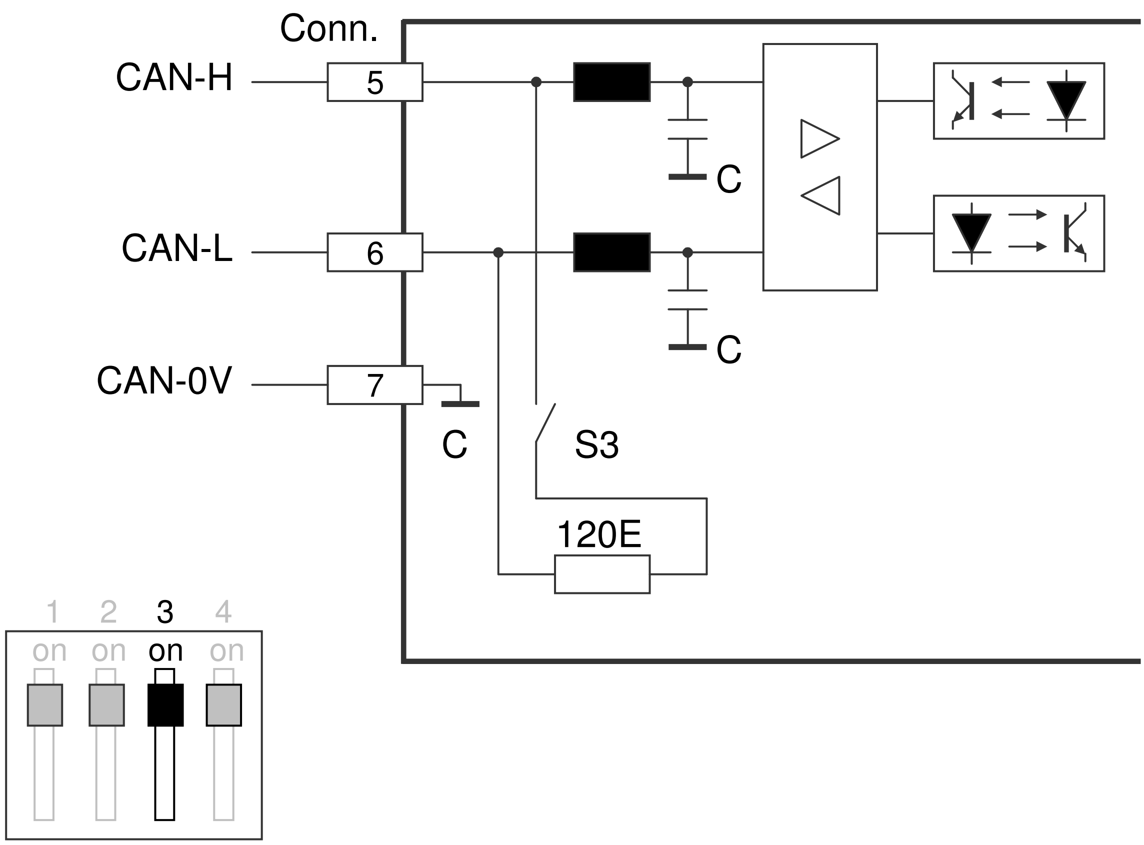
6.11 Digital outputs



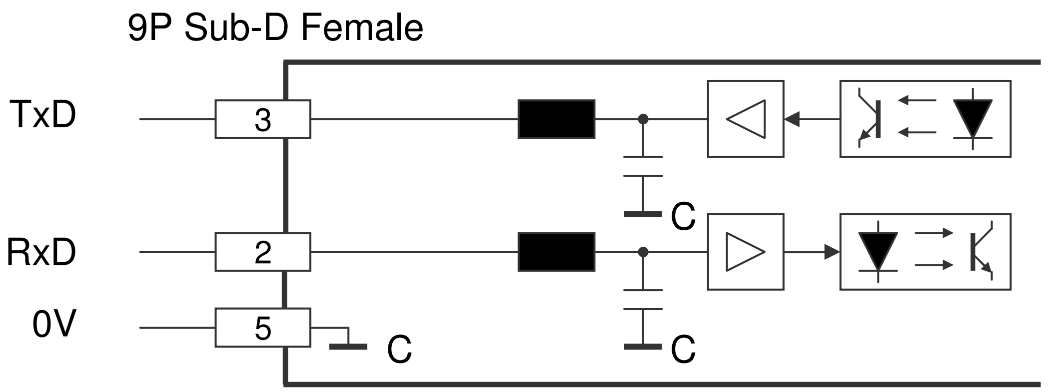
6.12 Analog output



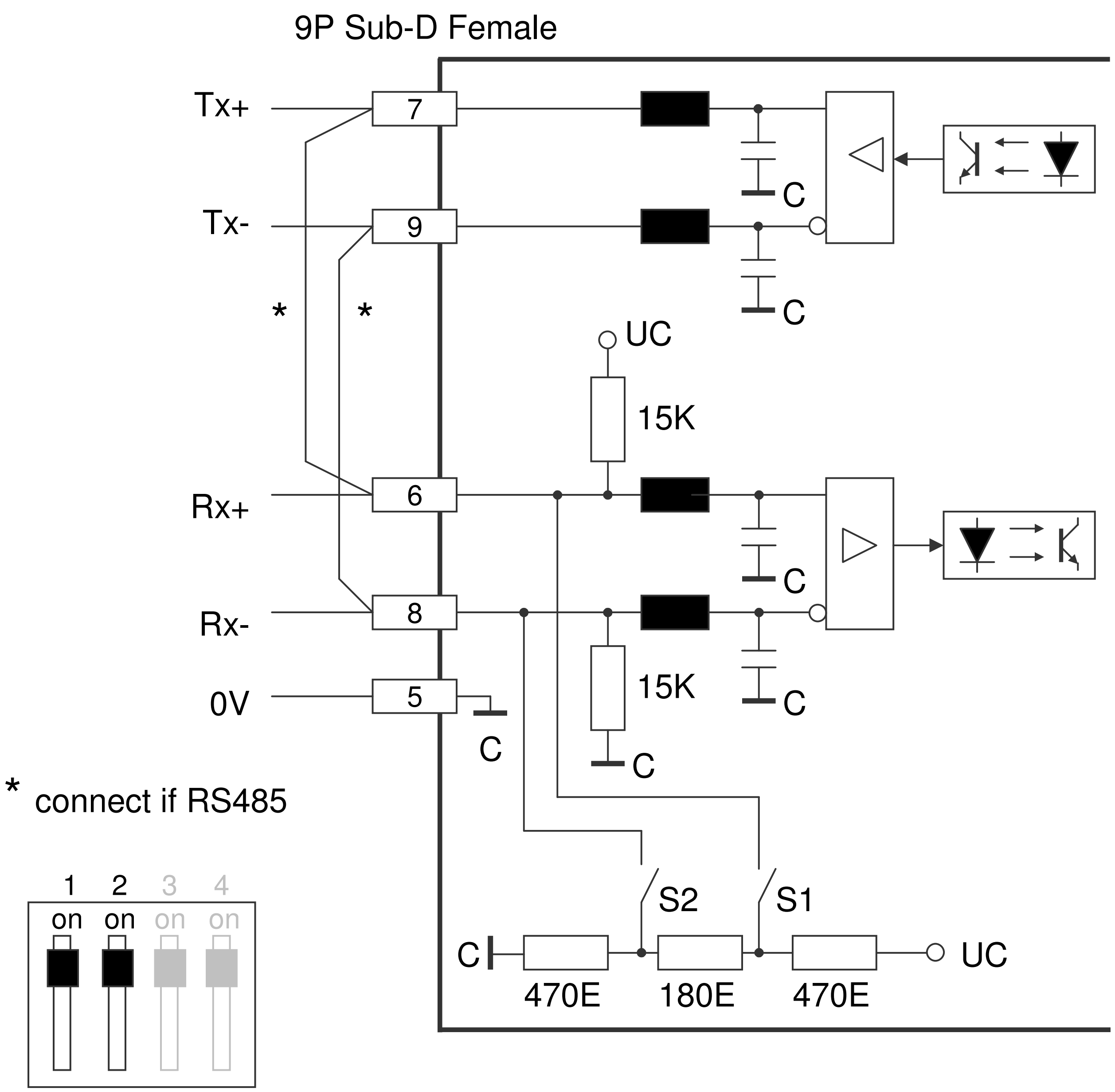
6.13 CAN-bus



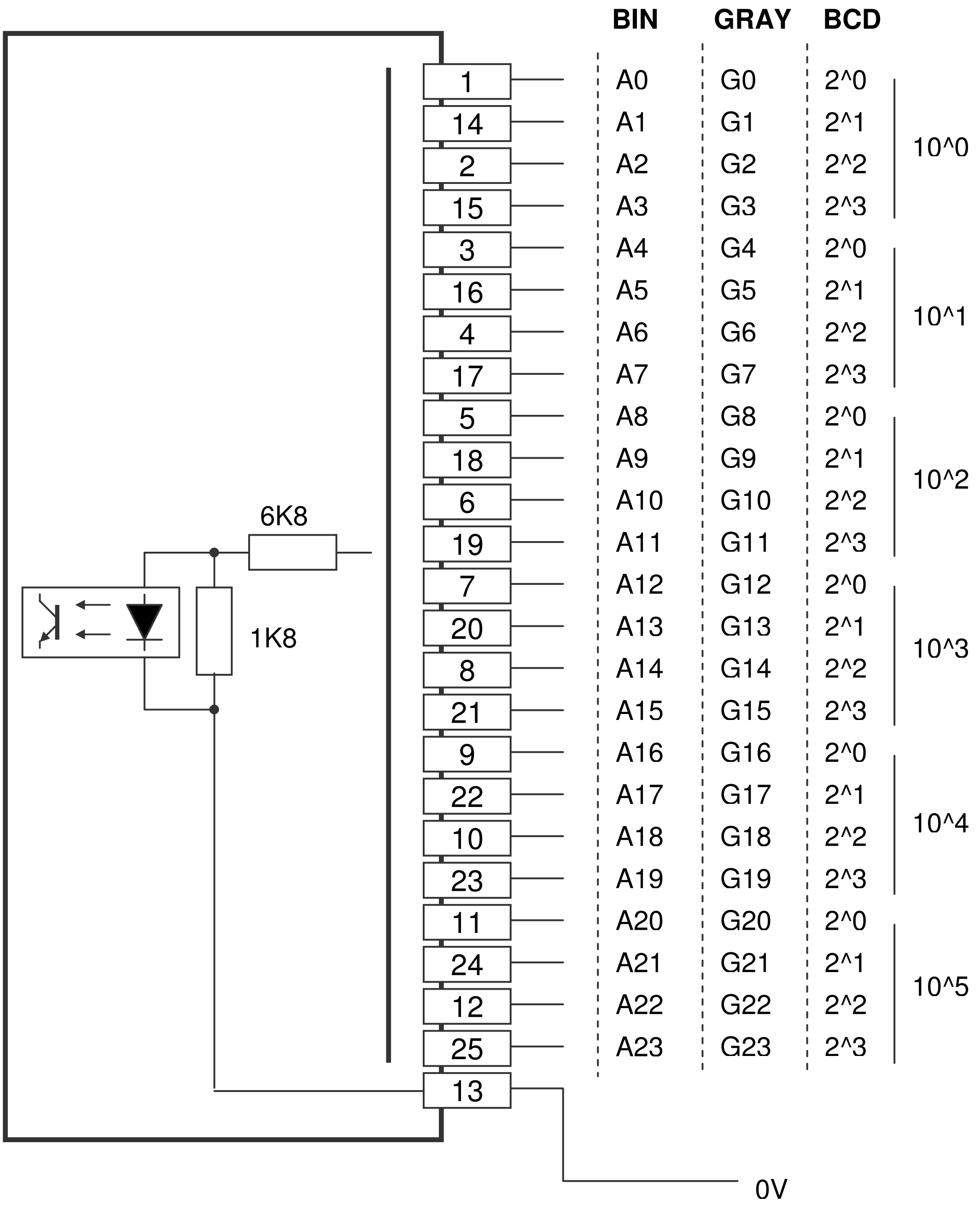
6.14 RS232 Ser-1



6.15 RS422/485 Ser-2



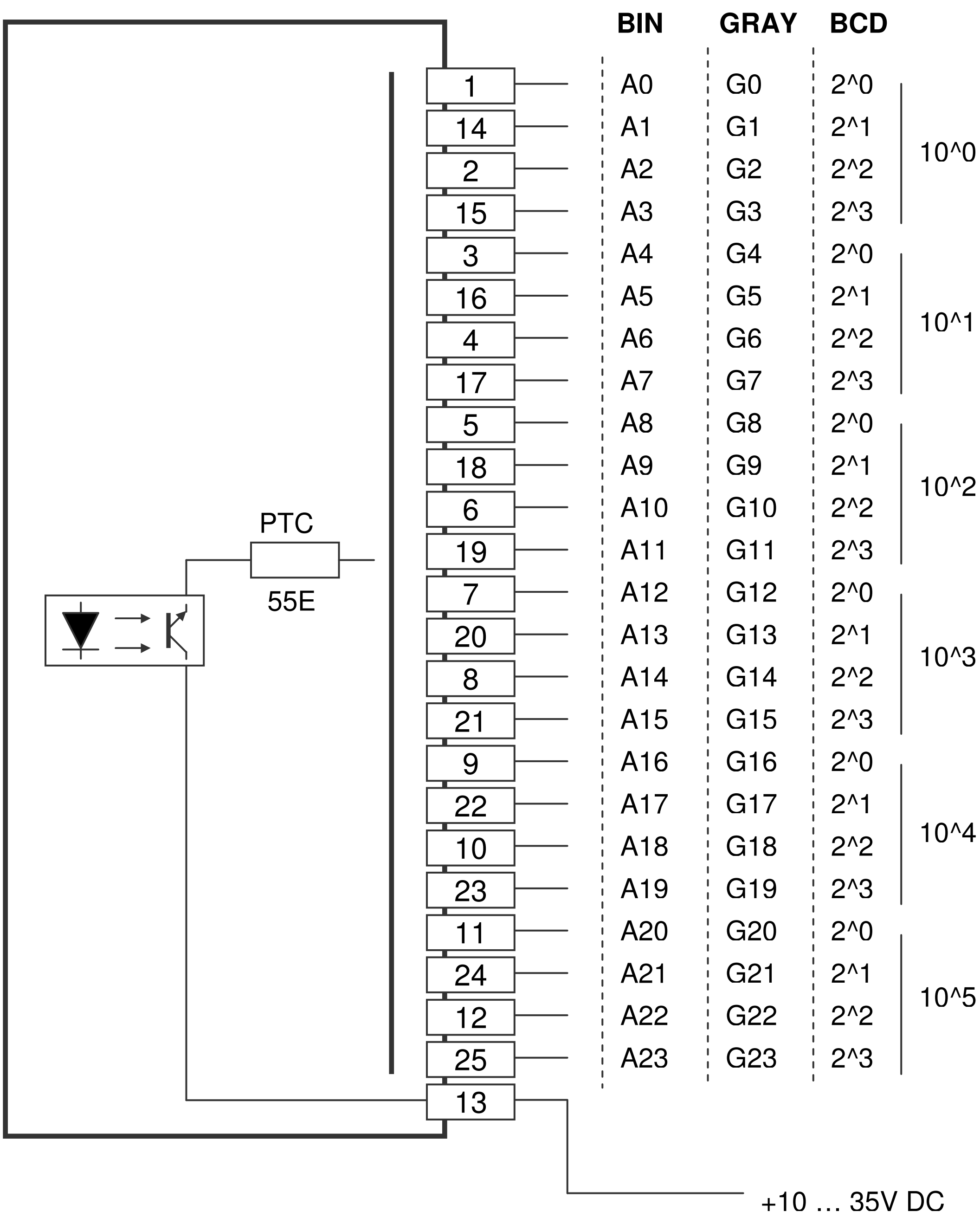
6.16 Data input (25P sub-D male)



The signals "minus sign" and "data-valid" can be appointed to different inputs (see parameters for data input).

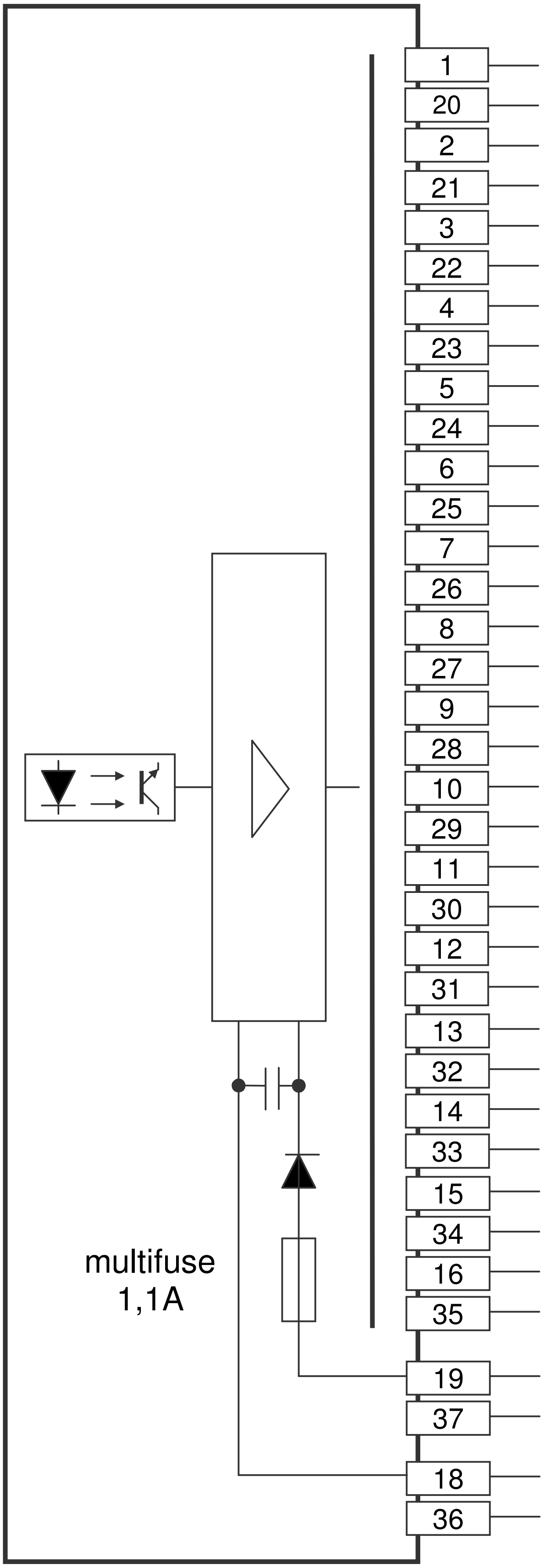
When using the minus sign: Logical high = active.

6.17 Data output (25P sub-D female)



The signals "minus sign" and "data-valid" can be appointed to different outputs (see parameters for data output).

6.18 Data output (37P sub-D female)



Pin	BIN	GRAY	BCD	
1	A0	G0	2^0	10^0
20	A1	G1	2^1	
2	A2	G2	2^2	
21	A3	G3	2^3	
3	A4	G4	2^0	10^1
22	A5	G5	2^1	
4	A6	G6	2^2	
23	A7	G7	2^3	
5	A8	G8	2^0	10^2
24	A9	G9	2^1	
6	A10	G10	2^2	
25	A11	G11	2^3	
7	A12	G12	2^0	10^3
26	A13	G13	2^1	
8	A14	G14	2^2	
27	A15	G15	2^3	
9	A16	G16	2^0	10^4
28	A17	G17	2^1	
10	A18	G18	2^2	
29	A19	G19	2^3	
11	A20	G20	2^0	10^5
30	A21	G21	2^1	
12	A22	G22	2^2	
31	A23	G23	2^3	
13	A24	G24	2^0	10^6
32	A25	G25	2^1	
14	A26	G26	2^2	
33	A27	G27	2^3	
15	A28	G28	2^0	10^7
34	A29	G29	2^1	
16	A30	G30	2^2	
35	A31	G31	2^3	
19, 37	+10 … 35V DC			
18, 36	0V			

7 Technical specifications

7.1 Specifications

- Supply voltage	10…35V DC (power failure not active)
	16…35V DC (power failure active)
current consumption	< 150mA
- Output voltage	for external encoder
+UD	max 400mA depending on supply voltage
+5V	max 400mA
- Processor	
µController	XC167
Data memory	EEPROM
Cycle time	250µS (fixed)
Counting range	-9999999…+99999999
- Counting input	optically isolated
signal level	low (5V): 0…+0.8V
	high (5V): +2.8V…+5V
	low (24V): 0…+5V
	high (24V): +15V…+35V
voltage output	5,3V max. 350mA
input resistor	appr. 3kOhm at 24V
	appr. 0.35kOhm at 5V
input frequency	max. 150 kHz
impulswidth K0	min. 2µ S
- SSI	optically isolated
data-input	low 0…+0,8V
	high +2,8V…+5V
clock-output	driver (RS422)
clock-frequency	125 kHz (138,9 kHz if > 26 bit encoder signal)
- Start/Stop	optically isolated
stop-input	low 0…+0,8V
	high +2,8V…+5V
start-output	driver (RS422)

- Digital inputs 1…6	optically isolated
	low: 0…+5V
	high: +10V…+35V
input resistor	appr. 1.8kOhm at 24V
- Digital outputs 1…9	optically isolated, N FET, short-circuit proof
Imax	500 mA (min load 200 µA)
Supply voltage	35V max.
- Voltage output	galvanically isolated
range	max. –10V … +10V
resolution	305 µV
offset-temp. coeff.	< 20 ppm/ °C
Imax	+/-12mA
- Current output	galvanically isolated
range	max. –20mA … +20mA
resolution	610 µA
offset-temp. coeff.	< 20 ppm/ °C
Rmax	550 Ohm
- Data-inputs (24bit)	optically isolated
(AP80-CXP)	low: 0…+5V
	high: +10…+35V
input resistor	appr. 6,8K Ohm at 24V
- Data-output (24bit)	optically isolated, NPN transistor,
(AP80-CXP)	open emitter with PTC
Imax	50 mA
Supply voltage	35V DC max.
Output voltage	supply voltage -3,5V at 50mA
	-2V at 20mA
- Data-output (32bit)	optically isolated, Push-Pull with tri-state
(AP80-CXF)	control en current limiting on each output
Imax	50 mA
Supply voltage	35V DC max.
UOutput voltage	supply voltage -1V max
- Serial communication	
Ser-1	RS232 C
Ser-2	RS422/485

- CAN-bus	
protocol	AP-Link
input objects (PDOs)	1 (each 64 bit data width)
output objects (PDOs)	1 (each 64 bit data width)
- Display	8 decades 7-segments LED
digit height	14 mm
- Temperature range	0…50℃
- Connection diameter	1,6 mm²
- Electromagnetic compatibility	in accordance with guideline 89/366/EEC
emission	EN 50081-1
immunity	EN 50082-2
- Weight	< 0.7 kg
- Sealing	front IP50, with protective hood IP54 rear IP20

7.2 Typekey

AP80- C X X

Data I/O

0 = no data I/O
P = 24 data inputs + 24 data outputs
F = 32 data outputs (tri-state)

Analog output

0 = no analog output
A = analog output

Serial communication

C = RS232, RS422/485 and CANbus

7.3 Typekey Thumbwheel sets

APD- X X

Number of decades

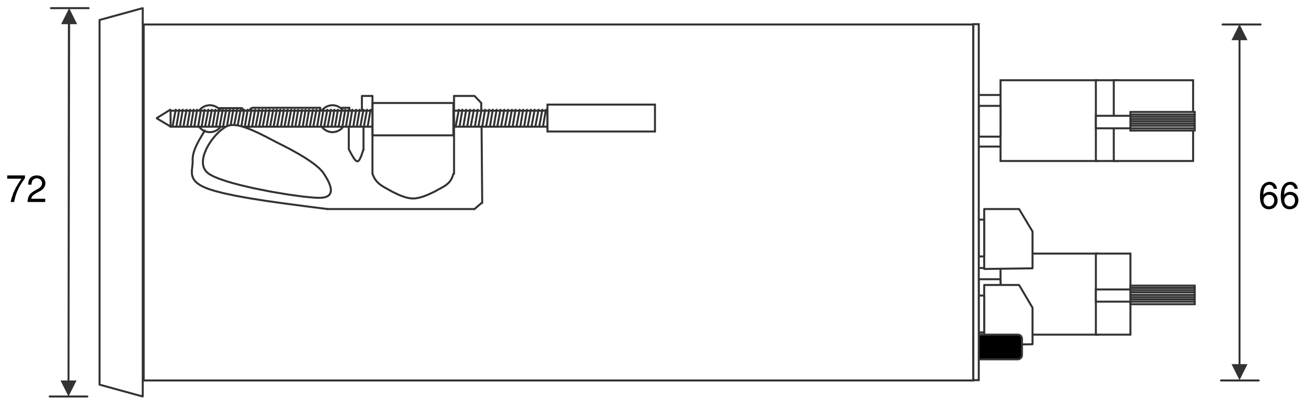
4 = 4 decade
5 = 5 decade
6 = 6 decade
7 = 7 decade (no sign possible)

Sign

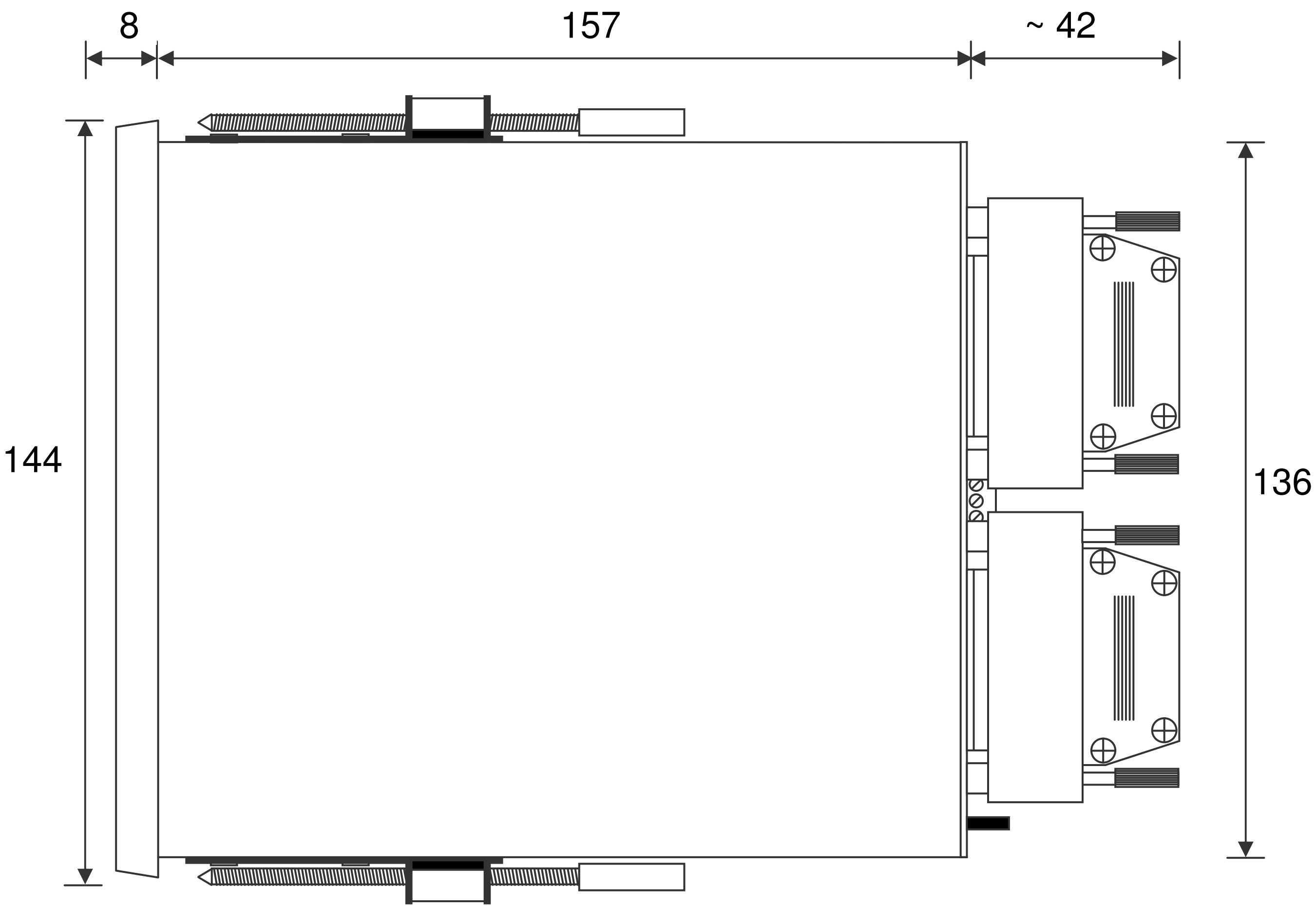
0 = no sign
T = with sign

7.4 Dimensions AP80

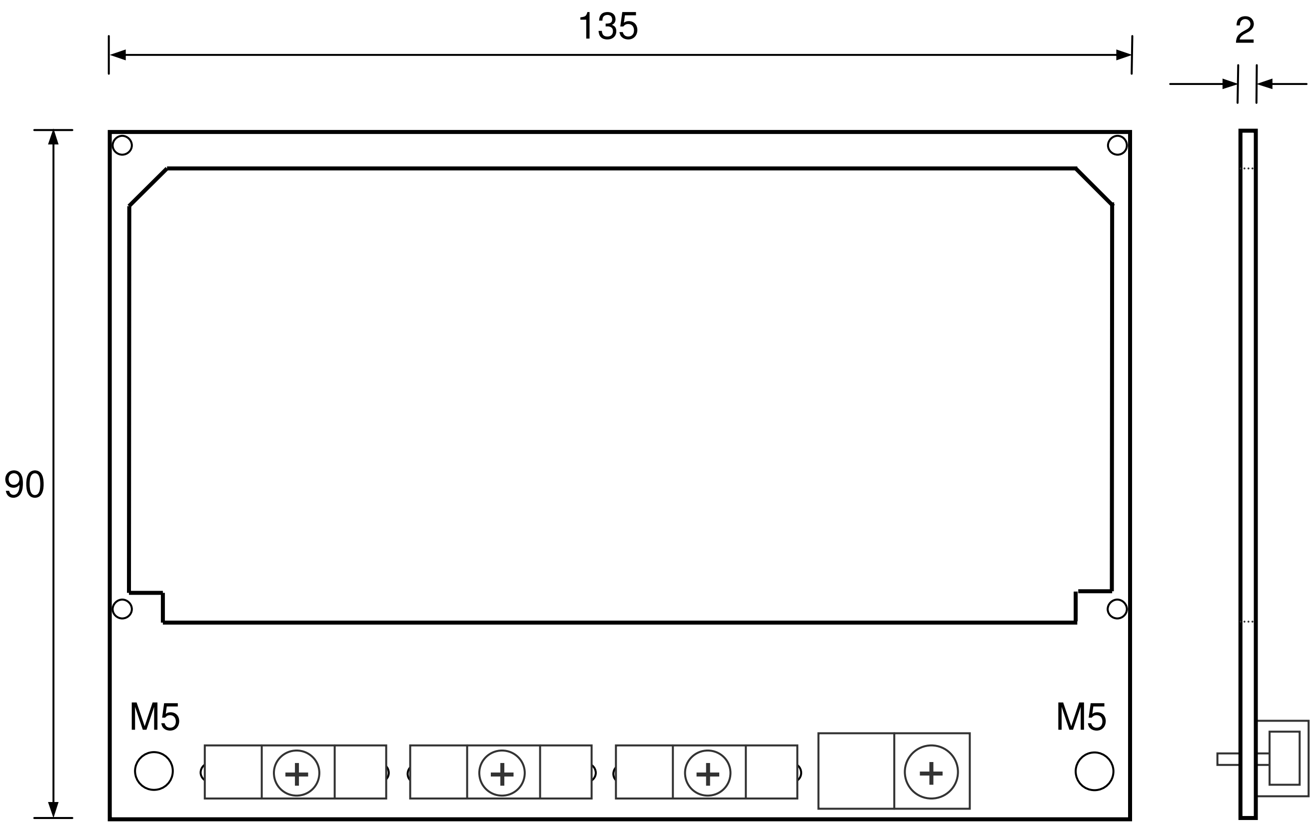
Side view



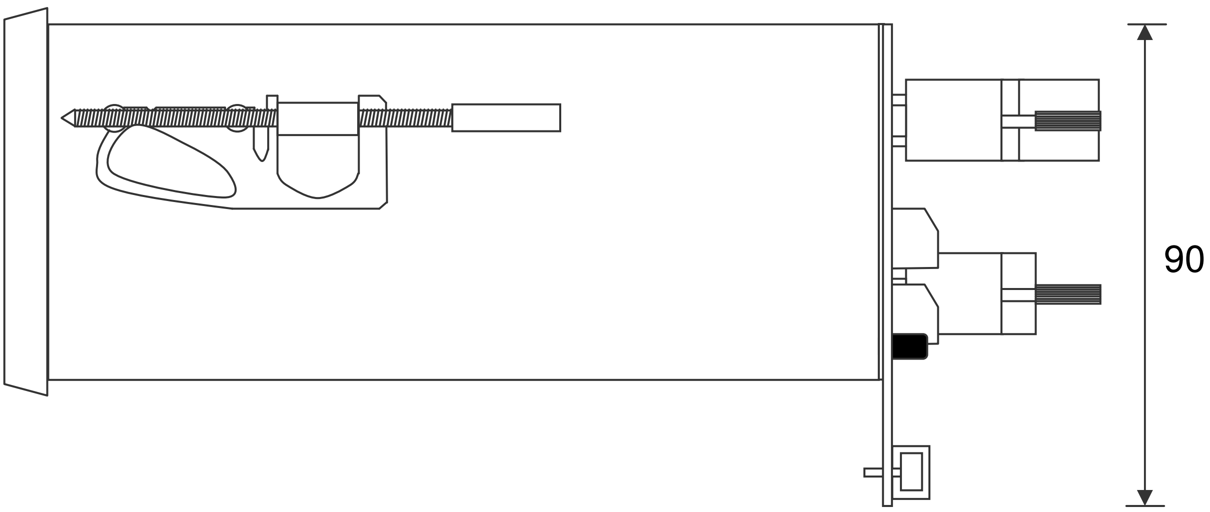
Top view



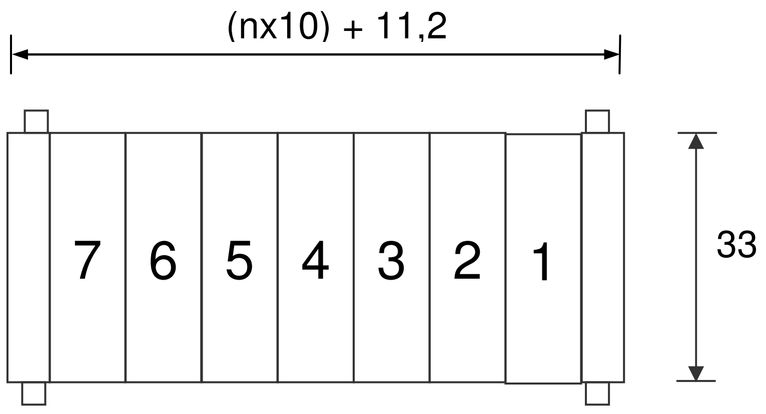
7.5 Dimensions EMC bracket type EMC-B01



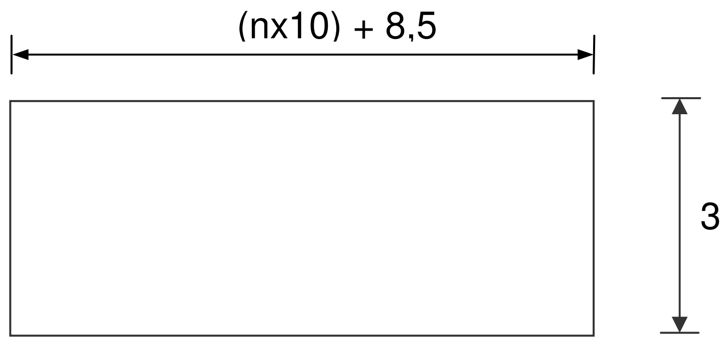
Side view with EMC bracket



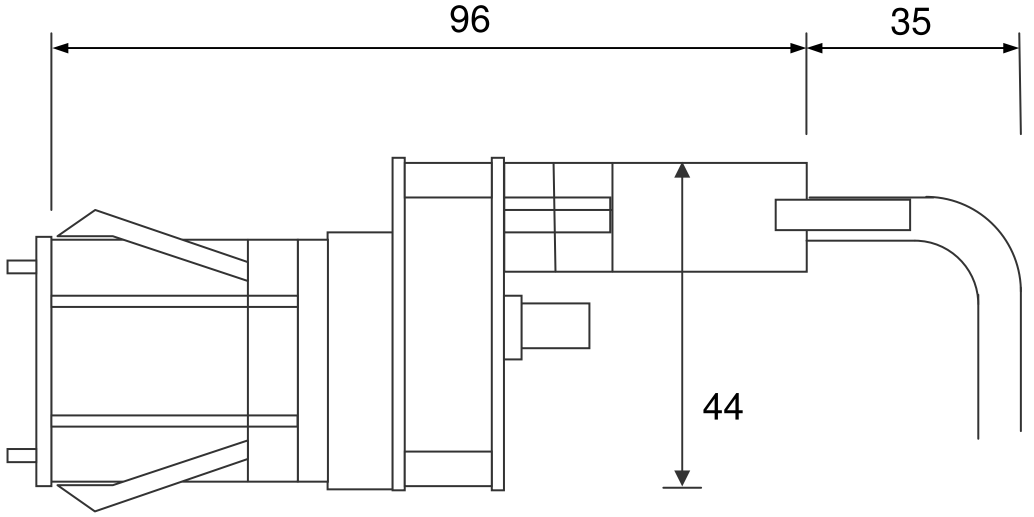
7.6 Dimensions thumbwheel switches APD-xx



Cut-out



Side view



7.7 Dimensions protective hood type CDS-B01