Manual AP90
Display Controller

- **8 Digit display in DIN-enclosure 144 X 72 mm**
- **Supply voltage 10 – 35V DC**
- **All in- and outputs optically isolated**
- **2x Input for absolute SSI encoders and counter input for incremental encoders**
- **CAN-bus, RS232, RS422/RS485**
- **8 Digital inputs and 16 digital outputs**
- **40 Programmable cams / 16 outputs**
- **80 Programmable nominal values**
- **Dynamic cam-adjustment (cycletime 500μS)**
- **Programmable analog output (16 bit)**

Table of contents

1 Introduction

1.1 General

With the microcontroller based AP90 it is possible to read 2 encoders with two 90° shifted incremental signals, one channel incremental encoders with a direction signal or absolute encoders with SSI-interface (Synchronous Serial Interface). It is standard equipped with 8 digital inputs and 16 digital outputs, all free programmable. The AP90 features a CAN-bus, RS232 communication, RS422/485 communication and an analog output.
All in- and outputs, including the communication-ports, are electrically isolated.

With the 40 programmable cams several functions can be realized like limit detection.

Another feature of the AP90 is the memory for the nominal values. The 80 programmable values can be used to determine the positions of the cams.

The microcontroller reads the actual sensor value of both inputs 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 AP90 is configurable for numerous applications by adjusting its parameters.

The AP90 can be programmable with the DST90 PC-based software.

1.2 Important information

- ✓ The AP90 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 AP90.
- ✓ If through a failure or fault the AP90 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 AP90.
- ✓ Necessary repairs to the AP90 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 AP90:

- ✓ 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

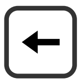
L→H : rising edge (low to high)
H →L : falling edge (high to low)

2 Operation

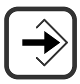
2.1 Key functions

[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)

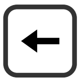


[Enter] key
- view status of inputs and outputs

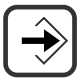
2.2 Key functions in programming mode

[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)

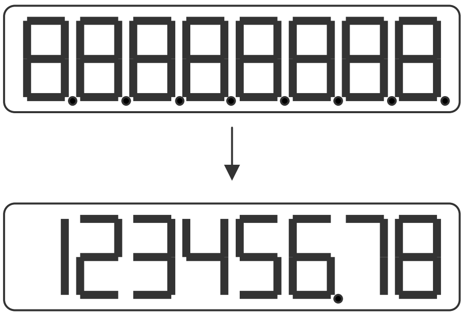


[Enter] key
- enter a submenu or parameter
- increase nominal value-/parameter number
- store a changed value
- clear value, hold down [Cursor] button (edit mode)

2.3 Display functions

2.3.1 Status functions

Power on



Appr. 1sec.

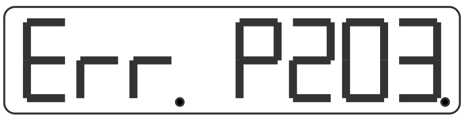
2.3.2 Error messages

There are two groups of errors:

- Parameter errors (error numbers 0…511, preceeded by a “P”)
- General error (error numbers from 512 and up)

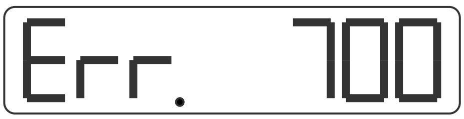
Example:

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*

Error no:

000...511 Parameter error is displayed as PXXX on the display.

700 = Reference value P[003] >= Counting range P[004]
701 = Zero point adjustment value SSI P[005] >= Counting range
of adjustment value SSI P005] < Nulpunt P[002] (only if justagetype = 1,2 en 4)
703 = Number of active SSI databits P[228] > number of SSI Clock pulses P[227]

705 = Transmitting AP-Link (Obj1…3) not possible with the same adress
710 = Reference value counter P[013] >= Counting range P[014]
711 = Zero-point adjustment value SSI P[015] >= Counting range
of adjustment value SSI P015] < Nulpunt P[012] (only if justagetype = 1,2 en 4)
713 = Number of active SSI databits P[248] > number of SSI Clock pulses P[247]

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)
737 = Function input-7 not valid (equal to input-1...input-6)
738 = Function input-8 not valid (equal to input-1...input-7)

800 = SSI error delta-s 1
801 = SSI error cable failure 1
802 = SSI error delta-s 2
803 = SSI error cable failure 2

Cam errors (last 2 digits = cam number).

Error no:

10xx

Counting range active

Length cam = 0 (if counting range is active)

No counting range active or source for cam is velocity

Length cam <= 0

11xx Length cam <= Hysteresis

12xx Hysteresis too large or length cam too large (2 * Hysteresis) + Length >= Counting range

13xx cam begin and/or cam end outside counting range (incl. hysteresis)

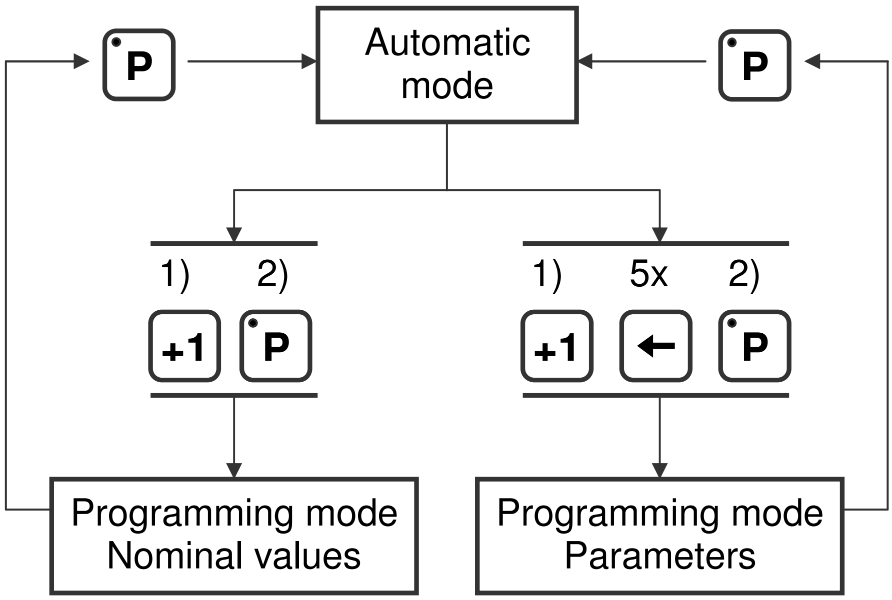
Error messages ASCII

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

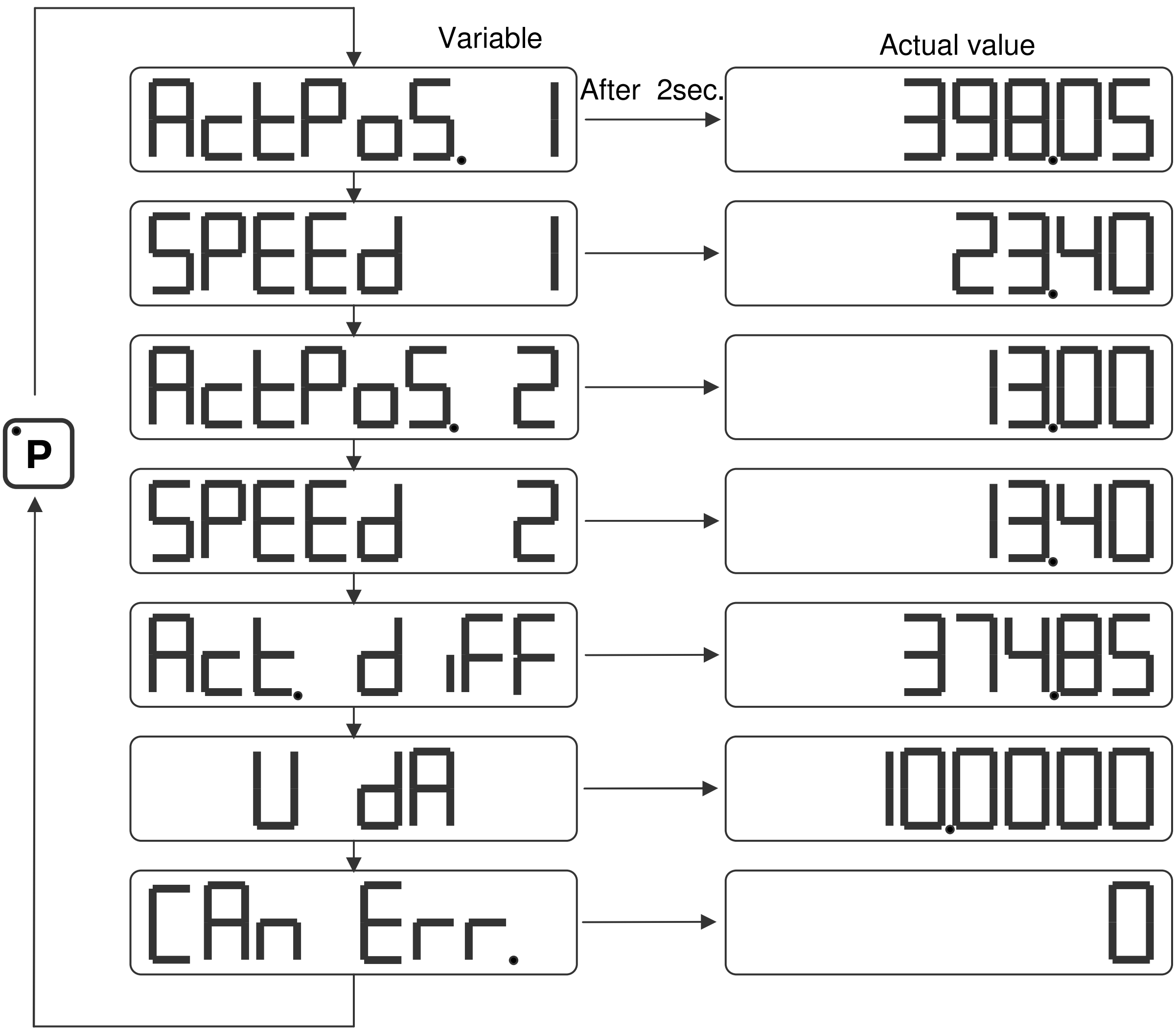
3.1 Automatic mode

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

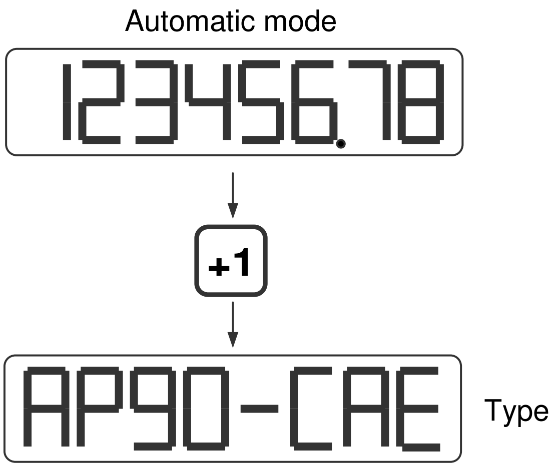
In automatic mode different variables can be displayed. By using the [P] key one can cycle through the different pages:

- Actual position 1
- Actual velocity 1
- Actual position 2
- Actual velocity 2
- Actual difference
- Voltage or current of the D/A-converter
- CANbus error

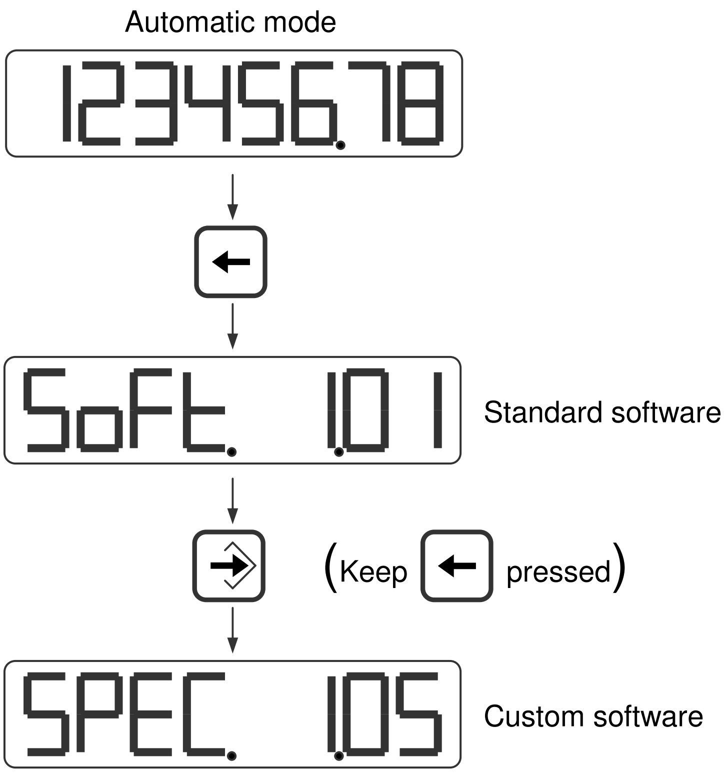


Parameter P[208] determines which option is visible after start-up.

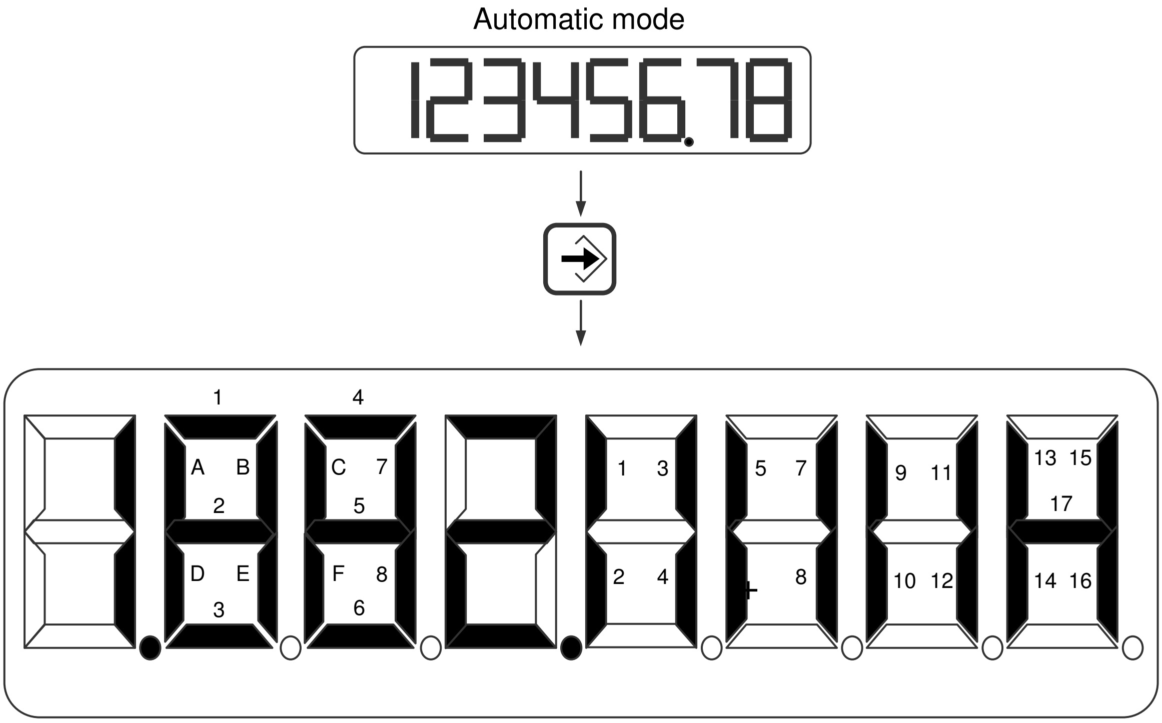
3.1.2 *Displaying the type number*



3.1.3 *Displaying the software-version*

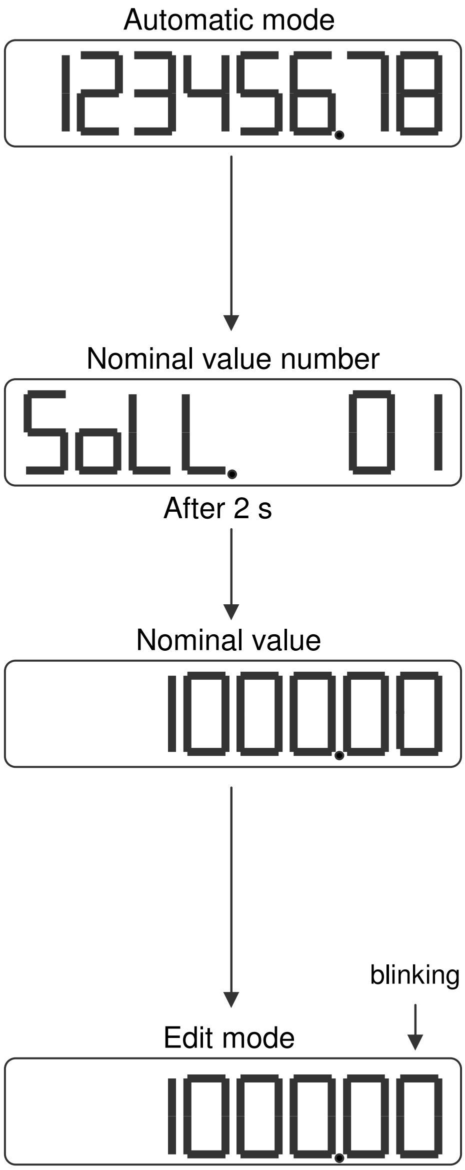


3.1.4 Status in- and outputs



Inputs	Outputs	Outputs
1 = input-1	1 = output-1	17 = Cams active
2 = input-2	2 = output -2	
3 = input-3	3 = output -3	
4 = input-4	4 = output -4	
5 = input-5	5 = output -5	
6 = input-6	6 = output -6	
6 = input-6	7 = output -7	
6 = input-6	8 = output -8	
A = input K1-1	9 = output -9	
B = input K2-1	10 = output -10	
C = input K0-1	11 = output -11	
D = input K1-2	12 = output -12	
E = input K2-2	13 = output -13	
F = input K0-2	14 = output -14	
	15 = output -15	
	16 = output -16	

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-value
- [←] Move one digit left
- [→] Confirm change
- *) [←] Then [→] Clear input

*) keep pressed

3.3 Changing parameters

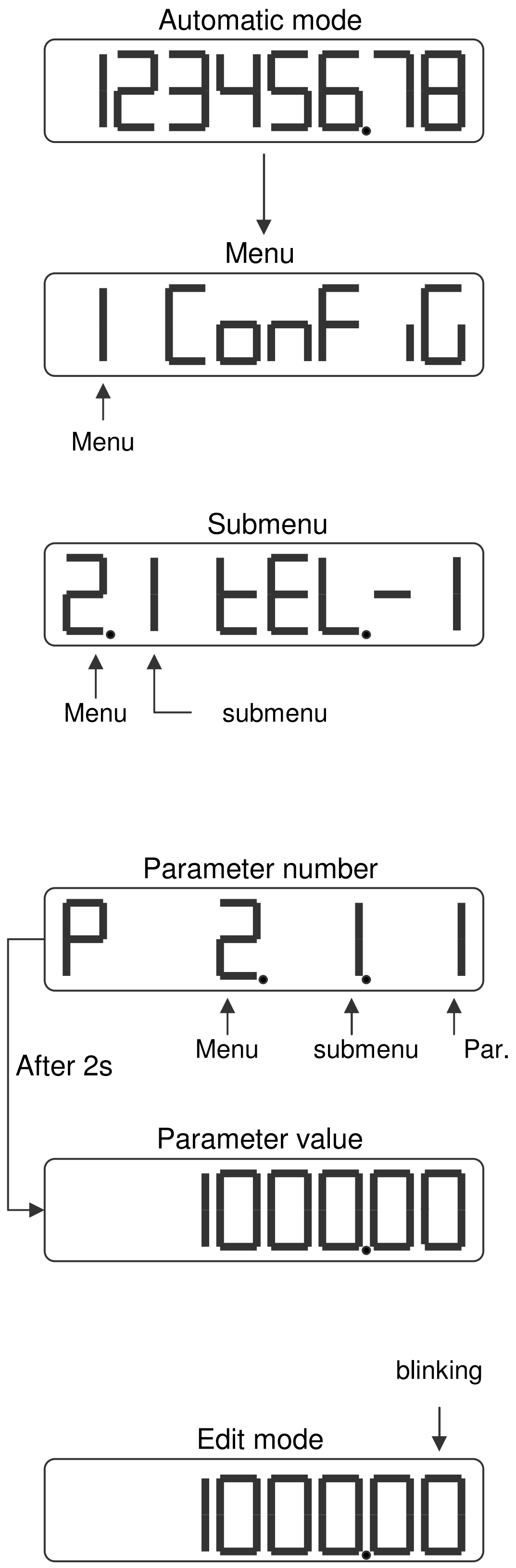
3.3.1 Menus

The parameters are displayed in different menus and submenus:

```
1 ConFiG
2 ActuAL
    2.1 teL-1
    2.2 SSI-1
    2.3 tel-2
    2.4 SSI-2
3 CAnbus
    3.1 ConFG.
    3.2 Out-1
    3.3 Out-2
    3.4 Out-3
4 SEriAL
    4.1 ConFG.
    4.2 SEr-1
    4.3 SEr-2
5 InPut
6 OutPut
    6.1 OP1
    ... ...
    ... ...
    6.9 OP16
7 AnALoG
    7.1 ConFG.
    7.2 dA-U
    7.3 dA-I
8 Cam
    8.1  CA1
    ...  ...
    ...  ...
    8.40 CA40
```

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

3.3.2 *Input parameters*



*) keep pressed down

<u>Access parameters</u>

*) [+1] 5x [←] Then [P]

<u>Menu selection</u>

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

<u>Submenu selection</u>

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

<u>Selecting parameters</u>

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

<u>Changing parameters</u>

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

4 Functions

4.1 Actual position

The actual position shown on the display is depending on the selected sensor input (P[200] input-1 and P[201] input-2) and several parameters.

Counter input:

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

SSI input:

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

FL	=	edgemultiplier	P[221], P[241]
Mt	=	multiplicator numerator	P[000], P[011]
Mn	=	multiplicator denominator	P[001], P[012]
N	=	offset	P[002], P[012]
dir	=	direction (x1 or x –1)	P[222], P[242]

4.2 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[047] = 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.*

4.3 Multiplicator

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

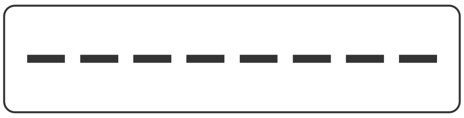
Example for counting input 1:
Encoder with 90º shifted signal and 1024 increments/revolution will have 4096 edges/revolution. If the desired value in the display is 360,0 (= 3600 AWE), the multiplicator will be 3600/4096 = 0,87890625.
It is possible to program the exact values (3600 and 4096) instead of the fraction.

Multiplicator (numerator)	*P[000], P[010]*	*=*	*3600*
Multiplicator (denominator)	*P[001], P[011]*	*=*	*4096*

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

4.4 Power failure protection

If P[206] = 1 then the actual position of the AP90 will be stored in EEPROM when power is shut down. After power up this value will be restored. When the values are stored the following display occurs.

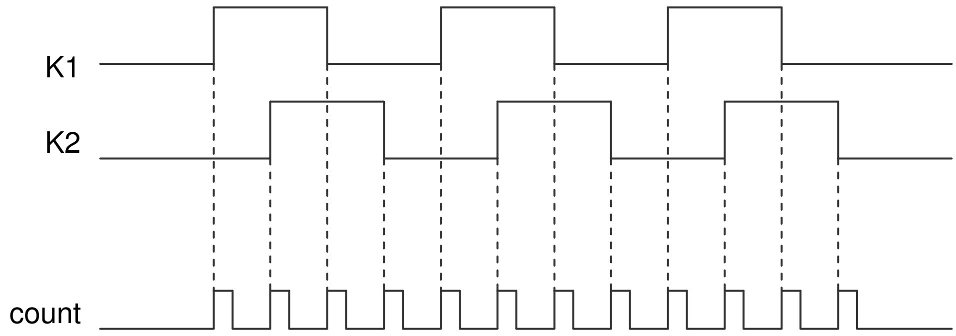


The power failure protection has no meaning with an absolute encoder (SSI).

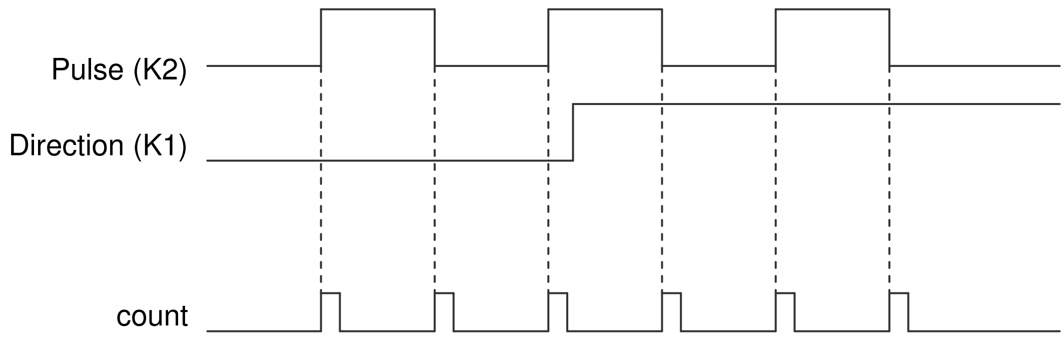
4.5 Edge multiplication (counter input)

There are two possibilities for the counter input:

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



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



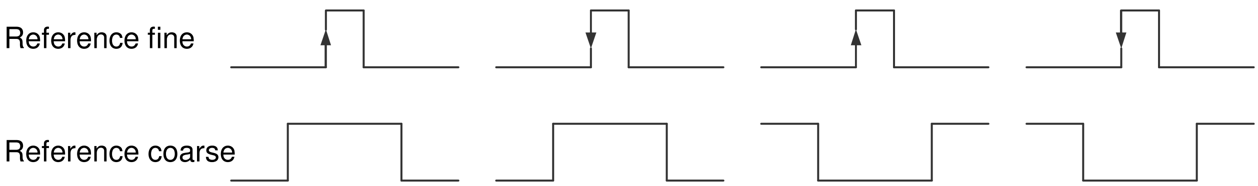
4.6 Preset (counter input)

The function preset is used to set the actual value to a programmed value, stored in P[003], P[013].
The value will be set in case of an active edge from the reference fine signal K0. (Interrupt)

If reference coarse has been activated (P[224] <> 0, P[044] <> 0) one of the inputs 1…8 (P[500] … [507]) has to be programmed to "**`1 – Reference Coarse`**".

When reference course is active, the actual value can only be preset when referenced input has the right level.

With parameter P[220], P[240] can be selected if the preset function is working direction independent.



Actual Input-1	P[223] = 1 P[224] = 1	P[223] = 0 P[224] = 1	P[223] = 1 P[224] = 0	P[223] = 0 P[224] = 0
Actual Input-2	P[243] = 1 P[244] = 1	P[243] = 1 P[244] = 1	P[243] = 1 P[244] = 1	P[243] = 1 P[244] = 1

4.7 Counting range (counter input)

The counting range used by the counter can be limited when using the counter input (P[200] = 0, P[201] = 0).
The number of increments is programmed, ignoring the decimal point.

<u>Counting range P[004], P[014]</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], P[010]*	*= 3600*
Multiplicator (denominator)	*P[001], P[011]*	*= 6000*
Counting range	*P[004], P014]*	*= 6000 increments*

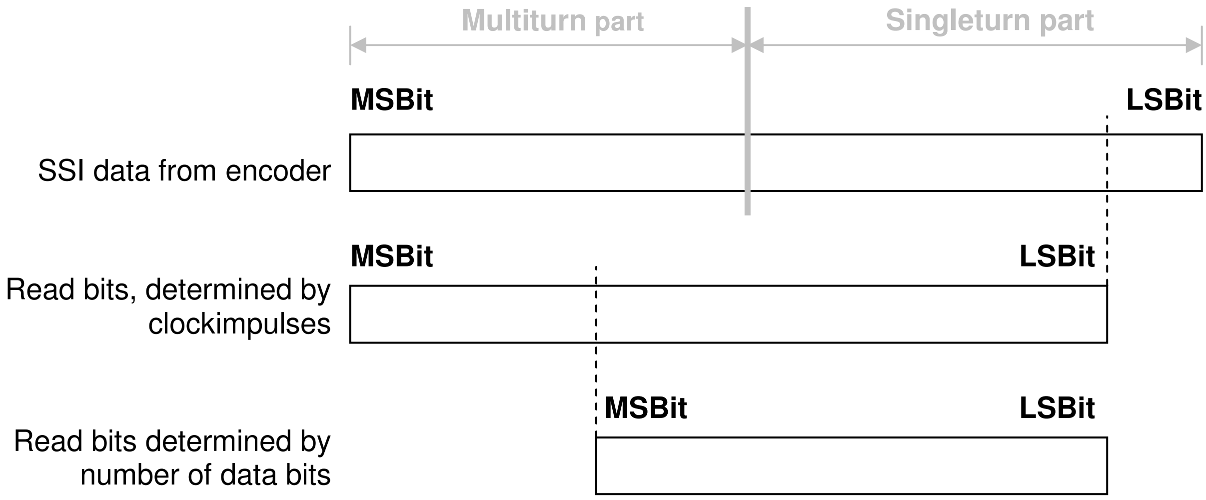
At P[203] it is possible to program the use of a decimal point.

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

4.8 Number of bits SSI

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

Number of clock impulses	P[227], P[247]
Number of data bits	P[228], P[248]



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 AP90. 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 single turn bits (65535 positions/rev.) and 14 multi turn 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 clock pulses will be:	*P[227], P[047]*	*=*	*27*	*(13+14)*
Number of data bits will be:	*P[228], P[048]*	*=*	*27*	*(13+14)*

4.9 SSI monitoring

The SSI-value is being read and processed every single cycle (500µs).
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], P[016]). It is possible to set the number of times that this value may be overrun (P[231], P[251]).

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.

4.9.1 Output "SSI error"

It is possible to create a SSI-error signal using one of the outputs 1…8:

<u>Output SSI error</u>
Output-x = option **`"2 SSI error"`** (Low = SSI error)

4.9.2 Reset "SSI error"

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

- Activate and leave programming mode parameters
- Reset with PC-program DST80
- Serial communication (ASCII)
- Set one of the inputs 1…8 to **`"4 Reset SSI error"`** and apply a high signal

4.10 SSI – zero point adjustment

The zero-point adjustment of a SSI-encoder can be done in two ways:

- Using the parameter "Adjustment absolute value" P[005], P[015]
- Using the input K0
- Using input 1…8

4.10.1 Using the parameter "Adjustment absolute value"

This function is applicable for rotary SSI encoders and is active when parameter "Justage" P[229], P[249] 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 SSI value and compensated with the display counting range.

$$\textbf{Display counting range} = \frac{\textbf{max SSI value} * \textbf{Multiplicator (numerator) P[000], P[010]}}{\textbf{Multiplicator (denominator) P[001], P[011]}}$$

4.10.2 Using the input K0

In this function the actual position can be preset to a certain value. The function is active when parameter "zero-point adjustment" P[229], P[249] 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)

Parameter "Zero-point adjustment value" P[005], P[015] can be set to a value to which the actual position will be adjusted when K0 detects a rising edge, possibly combined with a reference coarse signal ([P224], P[244]).
If through parameter [P224], P[244] reference coarse has been activated ([P224] <> 0], [P224] <> 0]), the option **"1 – Reference coarse"** should be selected for one of the inputs 1…8 (P[500]...P[507]).
The calculated offset will be stored in either RAM or EEPROM depending on the option chosen in parameter "zero-point adjustment" P[229], P[249]. When storing into RAM-memory (option 1 and 2) 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.

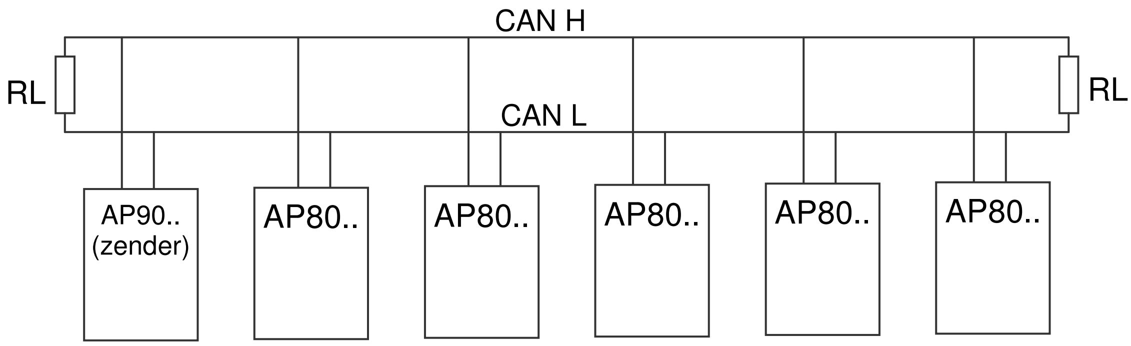
4.10.3 Justage using input 1…8

The function preset the absolute position can be executed with a rising edge on input 1…8. Then the actual position is preset to the value of P[005], P[015]. This function is active when:

<u>Sensor type SSI</u>
P[229], P[249] = variant **"5 rising edge input (1…8)"**
P[500] … [507]) = variant **"11 – SetRef/Adjustment 1"**
variant **"12 – SetRef/Adjustment 2"**

4.11 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.

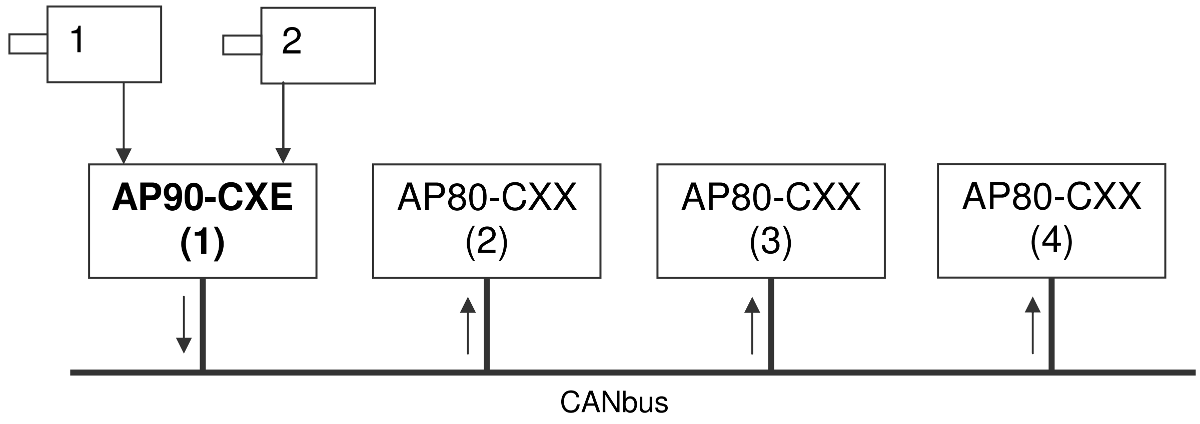
4.11.1 AP-Link

Using the CAN-bus (Obj1…3 out) it is possible to sent 3 separate CAN messages to one or more other AP90/80/40’s. For every CAN message can be adjusted which data has to be send

- Actual position 1 + velocity 1
- Actual position 2 + velocity 2
- Actual position difference + velocity = 0

Example:

AP90-CXE (1) sending data on CANbus
AP80-CXX (2) receiving data from AP90-CXE (1)
AP80-CXX (3) receiving data from AP90-CXE (1)
AP80-CXX (4) receiving data from AP90-CXE (1)



Settings for example as shown above **AP90**

	PAR	AP90 (1)
Baudrate	P[260]	5
Obj1/PDO1 Out adress	P[069]	1
Obj1/PDO1 Out function	P[261]	1
Obj1/PDO1 Out source	P[217]	0
Obj2/PDO2 Out address	P[070]	1
Obj2/PDO2 Out function	P[262]	1
Obj2/PDO2 Out source	P[218]	1
Obj3/PDO3 Out address	P[071]	3
Obj3/PDO3 Out function	P[263]	1
Obj3/PDO3 Out source	P[219]	2

Settings for example as shown above **AP80**

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

In this example the AP90-CXE (1) is sending:

- the actual position 1 + velocity 1 (adress 1)
- the actual position 2 + velocity 2 (adress 2)
- the actual difference + velocity = 0 (adress)3

AP80 (2), (3) and (4) receive this data as actual position and velocity.
If the time-out value (AP80) 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 received, or by performing a reset through one of the inputs 1..6 (choose option 14: “reset time-out error AP-link”).

4.12 ASCII protocol

The serial ports of the AP90, 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.

4.12.1 Overview functions

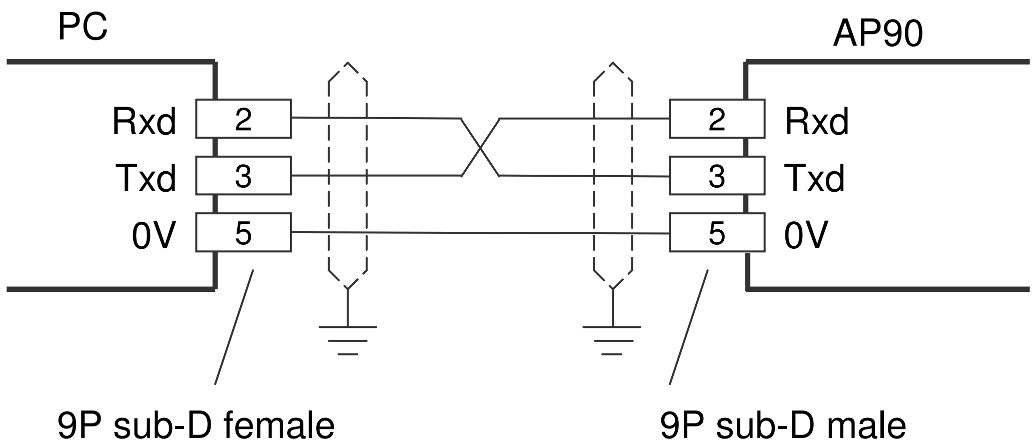
`sc` **select AP90**
`r0` **Read actual position (AWE)**
`r1` **Read actual velocity (AWE/s)**
`r2` **Read actual voltage output (0,1mV units)**
`r3` **Read actual current output (0,1 mA units)**
`ri` **Read status inputs**
`ru` **Read status outputs**
`rk` **Read status input K1, K2 and K0**
`wu` **Write 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**
`bp` **Load and activate parameters**

4.12.2 *General*

Through the ASCII protocol it is possible to communicate with the AP90.

<u>Send:</u> Data from PC, PLC → AP90

<u>Receive:</u> Data from AP90 → PC, PLC



(5m RS232 Kabel: KBL006-005)

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

<u>Receive structure:</u>
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)*

4.12.3 Functions

`sc` Select AP90

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

The AP90 with the unit Id number is selected, all consecutive commands are relevant for this unit.
An AP90 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 1 (AWE)

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

`r1` Read actual velocity 1 (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

`r4` Read actual position 2 (AWE)

send: `r4`
receive: `r4 xxxxxxxx`

transmitting parameter: none

`r5` Read actual velocity 2 (AWE/s)

send: **`r5`**
receive: **`r5 xxxxxxxx`**
transmitting parameter: none

`ri` Read status inputs

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

B0 = input1-K0	B5 = input-4
B1 = input2-K0	B6 = input-5
B2 = input-1	B7 = input-6
B3 = input-2	B8 = input-7
B4 = input-3	B8 = input-8

*Example: **ri 23 gives the following answer:***
23 → 17H, 00 0001 0111 B
input1- K0 = "1"
input2-K0 = "1"
input-1 = "1"
input-2 = "0"
input-3 = "1"
input-4 = "0"
input-5 = "0"
input-6 = "0"
input-7 = "0"
input-8 = "0"

`ru` Read status outputs

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

B0 = output-1	B8 = output-9
B1 = output-2	B9 = output-10
B2 = output-3	B10 = output-11
B3 = output-4	B11 = output-12

B4 = output-5	B12 = output-13
B5 = output-6	B13 = output-14
B6 = output-7	B14 = output-15
B7 = output-8	B15 = output-16

`rk` Read status input K1, K2 and K0

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

B0 = K0 (input-1)
B1 = K1 (input-1) or counting direction
B2 = K2 (input-1) or counting pulse
B3 = K0 (input-2)
B4 = K1 (input-2) or counting direction
B5 = K2 (input-2) 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	B8 = output-9
B1 = output-2	B9 = output-10
B2 = output-3	B10 = output-11
B3 = output-4	B11 = output-12
B4 = output-5	B12 = output-13
B5 = output-6	B13 = output-14
B6 = output-7	B14 = output-15
B7 = output-8	B15 = output-16

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

`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 AP90`
transmitting parameter: none

Example:
send: ***`rt`***
answer: ***`rt AP90-CAE`***

`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 1)*
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/zero-point adjustment 1 set
B2 = reference/zero-point adjustment 2 set

Example:
`rn 3 gives the following answer:`
3 → 3H, 0011 B
cams are active and ref/zero-point adjustment 1 have been set

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)*

4.12.4 Error messages

In case of an error the AP90 will sent an error message followed by an error number.

overview error messages
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

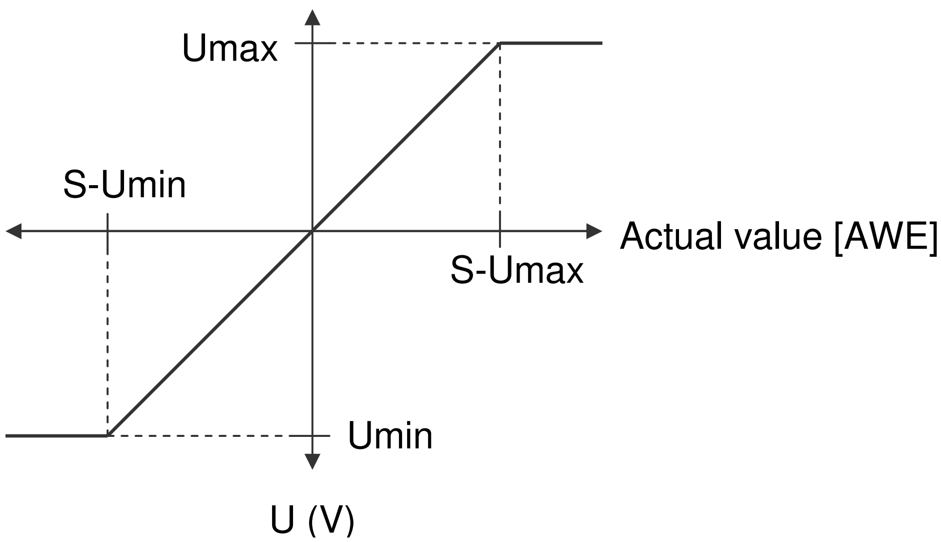
4.13 Analog output

The AP90 has an optional, galvanic isolated analog output (updated every 5ms). Using parameter P[280] 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 1,2 or velocity 1,2 or actual position difference (see parameter P[281]).

4.14 Voltage output

The voltage output has a resolution of 305 µV and is programmable through P[050] … P[053].

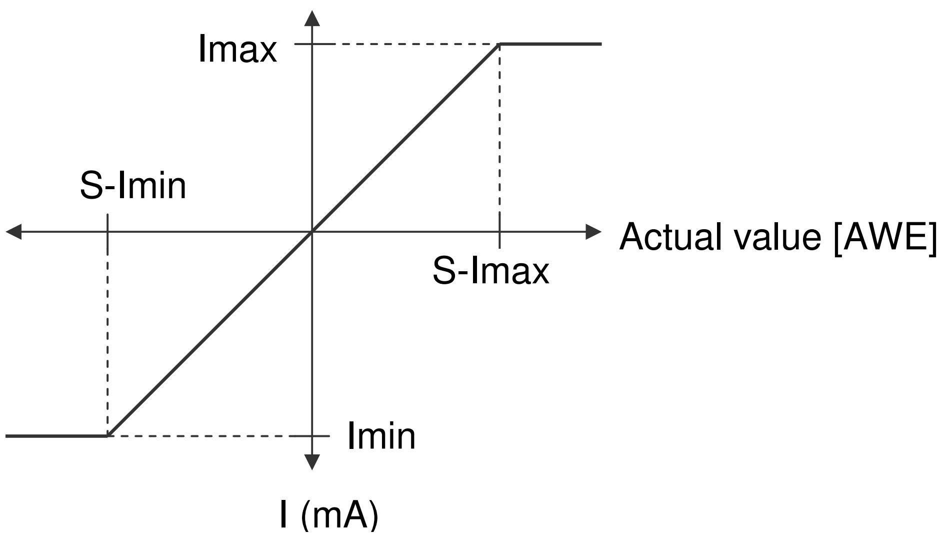
P[050] = Umin [V] (input in 0,0001V units)
P[051] = Umax [V] (input in 0,0001V units)
P[052] = S-Umin [AWE] (actual value at Umin)
P[053] = S-Umax [AWE] (actual value at Umax)



4.15 Current output

The current output has a resolution of 610 µA and is programmable through P[054] … P[057].

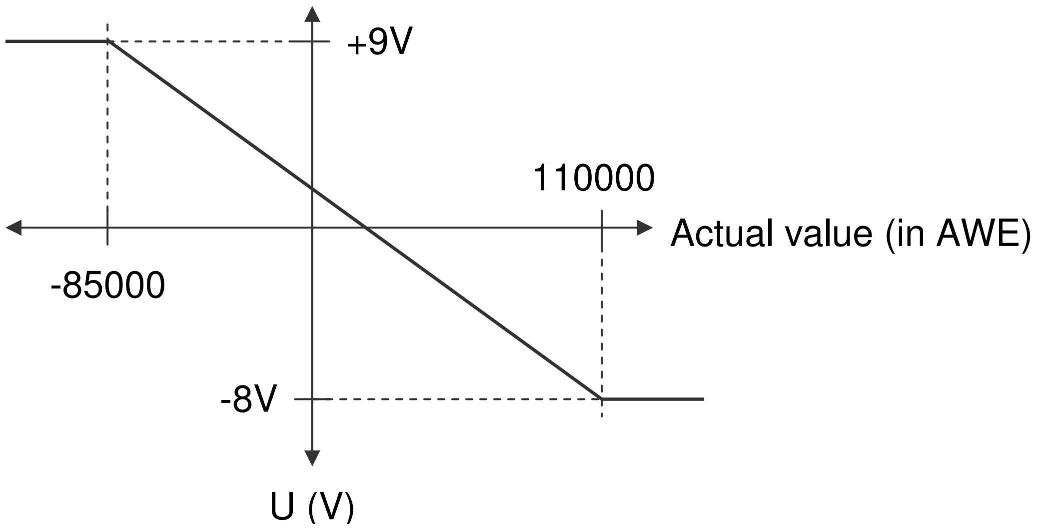
P[054] = Imin [A] (input in 0,0001mA units)
P[055] = Imax [A] (input in 0,0001mA units)
P[056] = S-Imin [AWE] (actual value at Imin)
P[057] = S-Imax [AWE] (actual value at Imax)



4.16 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[050]	=	Umin	=	-8,0000 [V]
P[051]	=	Umax	=	+9,0000 [V]
P[052]	=	S-Umin	=	110000 [AWE]
P[053]	=	S-Umax	=	-85000 [AWE]



4.17 Cams

4.17.1 General

The AP90 has a maximum of 40 programmable cams divided over a maximum of 16 outputs.

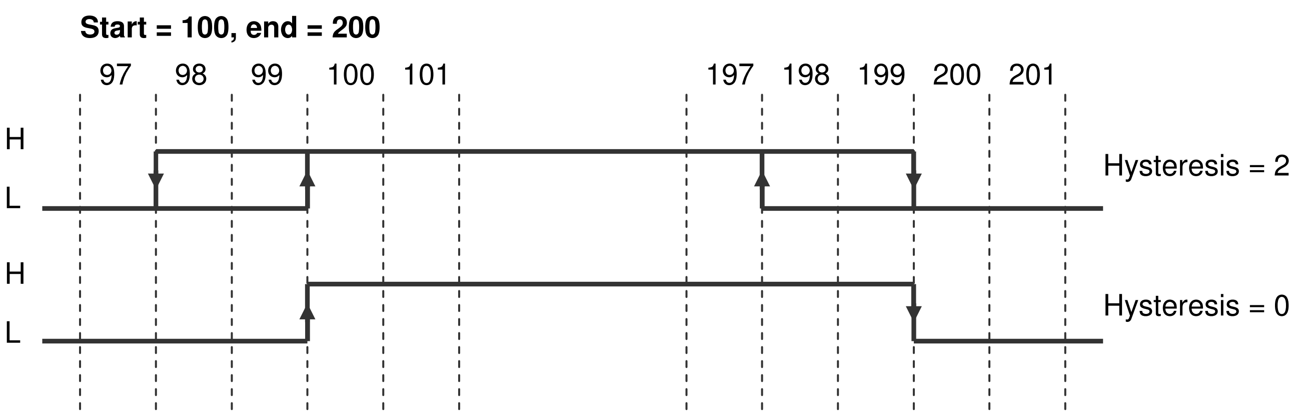
<u>Programmable functions:</u>

- 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 position difference)
- 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.

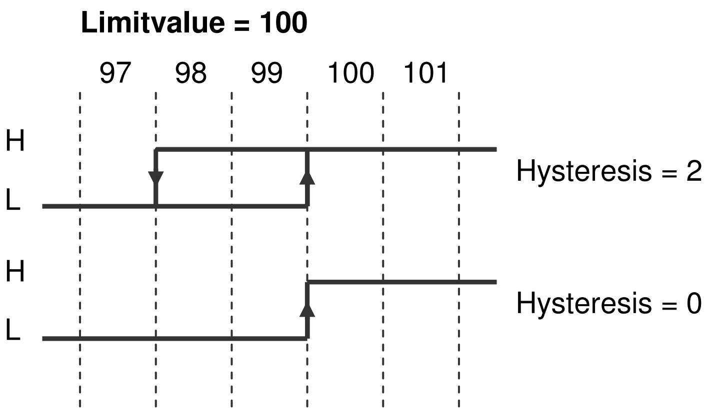
4.17.2 Cam with start- and end-value

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



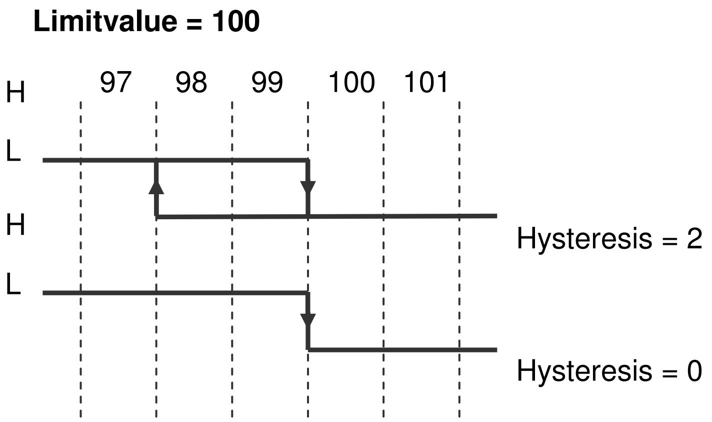
4.17.3 Greater than or equal to limit value

Only one value needs to be programmed.



4.17.4 Smaller than or equal to limit value

Only one value needs to be programmed.



4.17.5 Dynamic cams

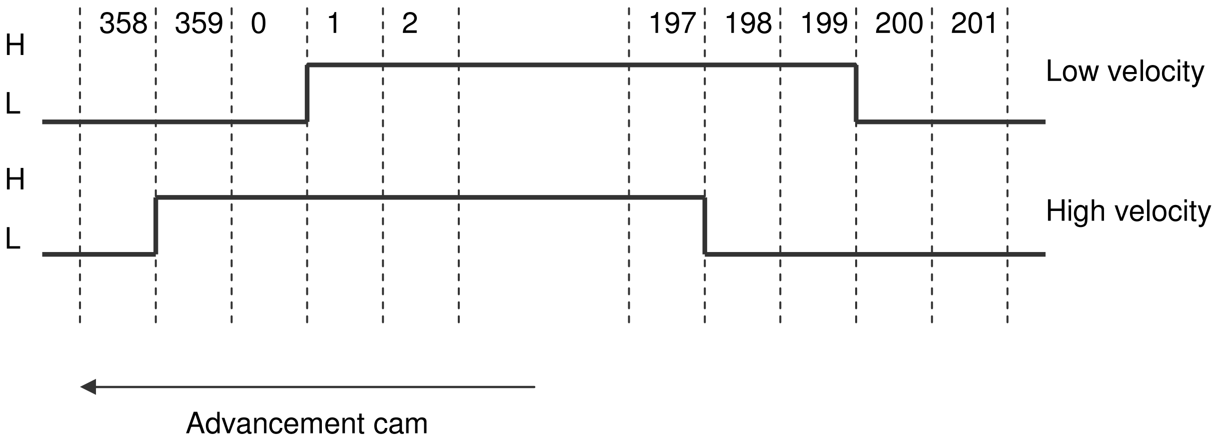
To compensate for actions with a static time, for example the switch time of a valve, it is possible to program a time for each output individually. The cams will be shifted according to this programmed time.

This function is only available for cams with a start- and end-value (P[300] … P[339] = 1) and when the source of this cam is the actual position(P[340] … P[379] = 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*



4.17.6 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)

4.17.7 Output "Cams active"

On one of the outputs 1…16 the signal "cams active" can be generated by choosing option **"5 cams active"** (high = cams enabled).

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** … 1
Input for actual position 1		
0 = Counter 1 = SSI		

PAR: 1.0.2	P[201]	**0** … 1
Input for actual position 2		
0 = Counter 1 = SSI		

PAR: 1.0.3	P[047]	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		
X		

PAR: 1.0.6	P[204]	**0** … 1
Store function		
0 = no function 1 = display		

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		
0 = not active 1 = active		

PAR: 1.0.9	P[207]	**0** … 123
Service functions Only possible to activate through keyboard AP90		
0 = not active 123 = set default parameters (automatically reset to 0)		

PAR: 1.0.10	P[208]	**0** … 5
Default monitor function Determines the default which is visible after start-up.		
0 = Actual position 1 1 = Actual velocity 1 2 = Actual position 2 3 = Actual velocity 2 4 = Position difference 5 = Actual value DA		

5.2 Menu 2 Actual

5.2.1 Submenu 2.1 Counter 1

PAR: 2.1.1	P[221]	**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[222]	**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[223]	**0** … 2
Reference fine (input K0)		
0 = no function 1 = rising edge 2 = falling edge		

PAR: 2.1.6	P[224]	**0** … 2
Reference coarse		
0 = no function 1 = high signal 2 = low signal		

PAR: 2.1.7	P[220]	**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		
-XXXXXXXX (AWE)		

PAR: 2.1.9	P[004]	**0** … 99999999
Counting range		
XXXXXXXX (AWE)		

5.2.2 Submenu 2.2 SSI 1

PAR: 2.2.1	P[225]	**0** … 1
SSI code		
0 = gray 1 = binary		

PAR: 2.2.2	P[222]	**0** … 1
Counting direction		
0 = no reverse 1 = reverse		

PAR: 2.2.3	P[227]	0 … **24** … 30
Number of SSI clockpulses		
XX		

PAR: 2.2.4	P[228]	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[229]	**0** … 6
Zero point 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 Zero-point 4 = rising edge (K0) permanent in EEPROM (not on interrupt) 5 = rising edge input 1…8 permanent in EEPROM (not on interrupt)		

PAR: 2.2.8	P[224]	**0** … 2
Reference coarse		
0 = no function 1 = high signal 2 = low signal		

PAR: 2.2.9	P[220]	**0** … 2
Counting direction zero-point 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
Zero-point adjustment value		
-XXXXXXXX (AWE) input 0 means function inactive		

PAR: 2.2.12	P[006]	1 … **50** … 99999
Monitoring delta-SSI per cycletime (500 µs) Unprocessed SSI-value, only depending on number of active SSI-databits P[228]		
XXXXX		

PAR: 2.2.13	P[231]	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[232]	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 Counter 2

PAR: 2.3.1	P[241]	**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: 2.3.2	P[242]	**0** … 1
Counting direction		
0 = no reverse 1 = reverse		

PAR: 2.3.3	P[010]	0 … **10000** … 16777215
Multiplicator numerator		
XXXXXXXX		

PAR: 2.3.4	P[011]	0 … **10000** … 16777215
Multiplicator denominator		
XXXXXXXX		

PAR: 2.3.5	P[243]	**0** … 2
Reference fine (input K0)		
0 = no function 1 = rising edge 2 = falling edge		

PAR: 2.3.6	P[244]	**0** … 2
Reference coarse		
0 = no function 1 = high signal 2 = low signal		

PAR: 2.3.7	P[240]	**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: 2.3.8	P[013]	-9999999 … **0** … 99999999
Reference value		
-XXXXXXXX (AWE)		

PAR: 2.3.9	P[014]	**0** … 99999999
Counting range		
XXXXXXXX (AWE)		

5.2.4 Submenu 2.4 SSI 2

PAR: 2.4.1	P[245]	**0** … 1
SSI code		
0 = gray 1 = binary		

PAR: 2.4.2	P[242]	**0** … 1
Counting direction		
0 = no reverse 1 = reverse		

PAR: 2.4.3	P[247]	0 … **24** … 30
Number of SSI clockpulses		
XX		

PAR: 2.4.4	P[248]	0 … **24** … 30
Number of SSI databits		
XX		

PAR: 2.4.5	P[010]	0 … **10000** … 16777215
Multiplicator numerator		
XXXXXXXX		

PAR: 2.4.6	P[011]	0 … **10000** … 16777215
Multiplicator denominator		
XXXXXXXX		

PAR: 2.4.7	P[249]	**0** … 5
Zero point 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 Zero-point 4 = rising edge (K0) permanent in EEPROM (not on interrupt) 5 = rising edge input 1…8 permanent in EEPROM (not on interrupt)		

PAR: 2.4.8	P[244]	**0** … 2
Reference coarse		
0 = no function 1 = high signal 2 = low signal		

PAR: 2.4.9	P[240]	**0** … 2
Counting direction zero-point adjustment		
0 = independant from direction 1 = when counting in positive direction 2 = when counting in negative direction		

PAR: 2.4.10	P[012]	-9999999 … **0** … 99999999
Offset		
-XXXXXXXX (AWE)		

PAR: 2.4.11	P[015]	-9999999 … **0** … 99999999
Zero-point adjustment value		
-XXXXXXXX (AWE) input 0 means function inactive		

PAR: 2.4.12	P[016]	1 … **50** … 99999
Monitoring delta-SSI per cycletime (500 µs) Unprocessed SSI-value, only depending on number of active SSI-databits P[248]		
XXXXX		

PAR: 2.4.13	P[251]	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: 2.4.14	P[252]	0 … **3**
SSI monitoring		
0 = not active 1 = only wiring 2 = only Delta SSI monitoring 3 = wiring + Delta SSI monitoring		

5.3 Menu 3 CAN-bus

5.4 Submenu 3.1 Config

PAR: 3.1.1	P[260]	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 Out

PAR: 3.2.1	P[069]	0 … **1** … 127
CAN adress Obj/PDO1 Out		
XXX		

PAR: 3.2.2	P[261]	**0** … 4
Function Obj/PDO1 Out		
0 = not active 1 = AP-Link (sending actual position and velocity)		

PAR: 3.2.3	P[217]	**0** … 2
Data Obj/PDO1 Out		
0 = Actual position 1 1 = Actual position 2 2 = Position difference		

5.6 Submenu 3.3 Obj2/PDO2 Out

PAR: 3.3.1	P[070]	0 … **1** … 127
CAN adress Obj/PDO2 Out		
XXX		

PAR: 3.3.2	P[262]	**0** … 1
Function Obj/PDO2 Out		
0 = not active 1 = AP-Link (sending actual position and velocity)		

PAR: 3.3.3	P[218]	**0** … 2
Data Obj/PDO2 Out		
0 = Actual position 1 1 = Actual position 2 2 = Position difference		

5.7 Submenu 3.4 Obj3/PDO3 Out

PAR: 3.4.1	P[071]	0 … **1** … 127
CAN adress Obj/PDO2 Out		
XXX		

PAR: 3.4.2	P[263]	**0** … 4
Function Obj/PDO2 Out		
0 = not active 1 = AP-Link (sending actual position and velocity)		

PAR: 3.4.3	P[219]	**0** … 2
Data Obj/PDO2 Out		
0 = Actual position 1 1 = Actual position 2 2 = Position difference		

5.8 Menu 4 Serial

5.8.1 Submenu 4.1 Config

PAR: 4.1.1	P[270]	**0** … 31
Unit address		
XX		

5.8.2 Submenu 4.2 Ser-1 (RS232)

PAR: 4.2.1	P[271]	0 … **1** … 4
Baudrate		
0 = 9600 1 = 19200 2 = 28800 3 = 38400 4 = 57600		

PAR: 4.2.2	P[272]	**0** … 1
Number of stopbits		
0 = 1 Stopbit 1 = 2 Stopbits		

PAR: 4.2.3	P[273]	0 … **2**
Parity		
0 = none 1 = odd 2 = even		

PAR: 4.2.4	P[274]	0 … **1**
Protocol		
0 = no function 1 = ASCII		

5.8.3 Submenu 4.3 Ser-2 (RS422/485)

PAR: 4.3.1	P[275]	0 …**1** … 4
Baudrate		
0 = 9600 1 = 19200 2 = 28800 3 = 38400 4 = 57600		

PAR: 4.3.2	P[276]	**0** … 1
Number of stopbits		
0 = 1 Stopbit 1 = 2 Stopbits		

PAR: 4.3.3	P[277]	0 … **2**
Parity		
0 = none 1 = odd 2 = even		

PAR: 4.3.4	P[278]	**0** … 1
Protocol		
0 = no function 1 = ASCII		

5.9 Menu 5 Input

INPUT-1

PAR: 5.0.1	P[500]	**0** … 10
Function input-1		
0 = no function 1 = reference coarse 2 = reference coarse 3 = store 4 = reset SSI error 5 = start/stop cams 6 = start cams 7 = stop cams 8 = lock input nominal values 9 = lock input parameters 10 = lock input nominal values + parameters 11 = SetRef/Justage 1 12 = SetRef/Justage 2		

INPUT-2

PAR: 5.0.2	P[501]	**0** … 10
Function input-2		
XX (see input-1)		

INPUT-3

PAR: 5.0.3	P[502]	**0** … 10
Function input-3		
XX (see input-1)		

INPUT-4

PAR: 5.0.4	P[503]	**0** … 10
Function input-4		
XX (see input-1)		

INPUT-5

PAR: 5.0.5	P[504]	**0** … 10
Function input-5		
XX (see input-1)		

INPUT-6

PAR: 5.0.6	P[505]	**0** … 10
Function input-6		
XX (see input-1)		

INPUT-7

PAR: 5.0.7	P[506]	**0** … 10
Function input-6		
XX (see input-1)		

INPUT-8

PAR: 5.0.8	P[507]	**0** … 10
Function input-6		
XX (see input-1)		

5.10 Menu 6 Output

5.10.1 Submenu 6.1 – 6.16 Op1…16

OUTPUT 1…16

PAR: 6.x.1	P[282]…P[297]	**0** … 8
Function output-1		
0 = cam 1 = cam inverted 2 = SSI error (high = no error) 3 = reference/zero-point set 1 4 = reference/zero-point set 2 5 = cams active 6 = Counting direction 1 (high = downwards counting) 7 = Counting direction 2 (high = downwards counting) 8 = ASCII protocol		

PAR: 6.x.2	P[020]…P[035]	**0** … 5000
Dynamic cams (only when output = cam and source = actual position)		
X.XXX (sec) input 0 = no function		

5.11 Menu 8 Analog

5.11.1 Submenu 7.1 Config

PAR: 7.1.1	P[280]	**0** … 2
Selection DA output		
0 = inactive 1 = voltage 2 = current		

PAR: 7.1.2	P[281]	**0** … 1
Selection DA source		
0 = actual position 1 = actual velocity		

5.11.2 Submenu 8.2 DA-U (voltage)

DA PAR 7.2.1…7.2.4 = 0: DA not active

PAR: 7.2.1	P[050]	**-100000** … 99999
Umin DA		
-XX.XXXX (V)		

PAR: 7.2.2	P[051]	-99999 … **100000**
Umax DA		
-XX.XXXX (V)		

PAR: 7.2.3	P[052]	-9999999… **-100000** … 99999999
S-Umin DA		
-XXXXXXXX (AWE)		

PAR: 7.2.4	P[053]	-9999999 … **100000** … 99999999
S-Umax DA		
-XXXXXXXX (AWE)		

5.11.3 Submenu 8.3 DA-I (current)

DA PAR 7.3.1…7.3.4 = 0: DA not active

PAR: 7.3.1	P[054]	**-200000** … 199999
Imin DA		
-XX.XXXX (mA)		

PAR: 7.3.2	P[055]	-199999 … **200000**
Imax DA		
-XX.XXXX (mA)		

PAR: 7.3.3	P[056]	-9999999 … **-200000** …99999999
S-Imin DA		
-XXXXXXXX (AWE)		

PAR: 7.3.4	P[057]	-9999999 … **200000** … 99999999
S-Imax DA		
-XXXXXXXX (AWE)		

5.12 Menu 8 Cam

5.12.1 Submenu 8.1 … 8.40 CA1…40

CAM-1…40

PAR: 8.x.1	P[300]...P[339]	**0** … 3
Cam function		
0 = no function 1 = range 2 = actual position >= limit value 3 = actual position <= limit value		

PAR: 8.x.2	P[340]…P[379]	**0** … 2
Source		
0 = actual position 1 1 = actual velocity 1 2 = actual position 2 3 = actual velocity 2 4 = position difference		

PAR: 8.x.3	P[380]…P[419]	**0** … 48
Source cam begin / limit value (limit value if cam function = 2 or 3)		
0 = parameters cam begin 1…80 = Nominal value 1…80		

PAR: 8.x.4	P[420]…P[459]	**0** … 48
Source cam end		
0 = parameters cam end 1…80 = Nominal value 1…80		

PAR: 8.x.5	P[160]…P[199]	-9999999 … **1000** … 99999999
Cam begin / limit value (limit value if cam function = 2 or 3)		
-XXXXXXX		

PAR: 8.x.6	P[120]…P[159]	-9999999 … **2000** …99999999
Cam end		
-XXXXXXX		

PAR: 8.x.7	P[080]…P[119]	**0** … 999999
Hysteresis cam		
XXXXXX		

PAR: 8.x.8	P[460]…P[499]	**0** … 16
Assign cam to output		
0 = no output 1…16 = output 1-16		

5.13 Overview parameters

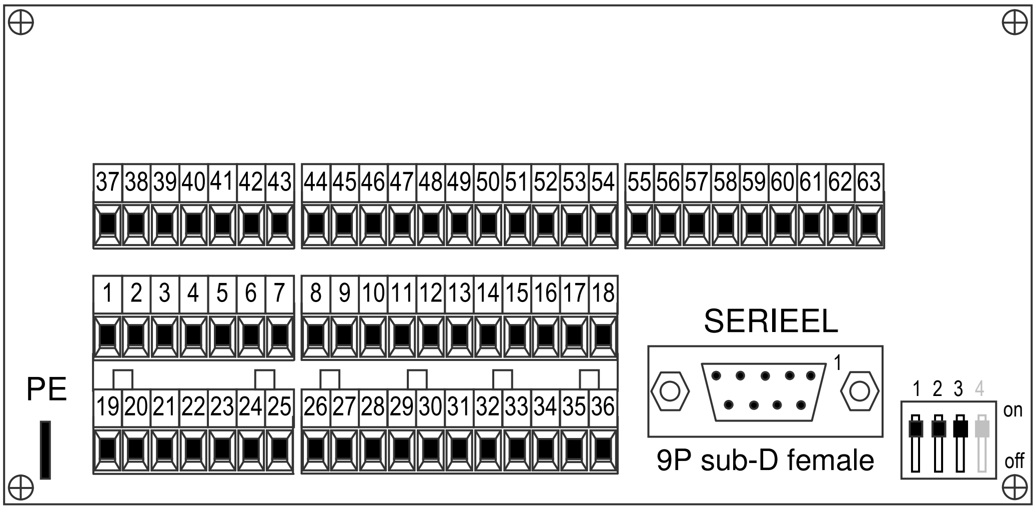
No	Description	Menu
[000] =	Multiplicator numerator 1	2.1.3/2.2.5
[001] =	Multiplicator denominator 1	2.1.4/2.2.6
[002] =	Offset 1	2.2.10
[003] =	Reference value 1	2.1.8
[004] =	Counting range 1	2.1.9
[005] =	Zero-point value 1	2.2.11
[006] =	Delta-SSI monitoring per cycle 1	2.2.12
[007]…[009] = no function		
[010] =	Multiplicator numerator 2	2.3.3/2.4.5
[011] =	Multiplicator denominator 2	2.3.4/2.4.6
[012] =	Offset 2	2.4.10
[013] =	Reference value 2	2.3.8
[014] =	Counting range 2	2.3.9
[015] =	Zero-point value 2	2.4.11
[016] =	Delta-SSI monitoring per cycle 2	2.4.12
[017]…[019] = no function		
[020]…[035] = Dynamic cam output 1…16		6.1.2…6.16.2
[036]…[046] = no function		
[047] =	Measuring time velocity	1.0.3
[048]…[049] = no function		
[050] =	Umin DA	8.2.1
[051] =	Umax DA	8.2.2
[052] =	S-Umin DA	8.2.3
[053] =	S-Umax DA	8.2.4
[054] =	Imin DA	8.3.1
[055] =	Imax DA	8.3.2
[056] =	S-Imin DA	8.3.3
[057] =	S-Imax DA	8.3.4
[058]…[068] = no function		
[069]…[071] = CAN adress Obj/PDO1…3 Out		3.2.1…3.4.1
[072]…[079] = no function		
[080]…[119] = Hysteresis cam		8.1.7…8.40.7
[120]…[159] = Cam end		8.1.6…8.40.6
[160]…[199] = Cam begin / limit value		8.1.5…8.40.5
[200] =	Input for actual position 1	1.0.1
[201] =	Input for actual position 2	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

No	Description	Menu
[206] =	Power failure protection	1.0.8
[207] =	Service functions	1.0.9
[208] =	Default monitor function	1.0.10
[209]…[216] = no function		
[217]…[219] = Function Obj/PDO1…3 Out		3.2.3…3.4.3
[220] =	Counting direction ref. / zero-p. adjustment 1	2.1.7/2.2.9
[221] =	Input type and edge multiplication 1	2.1.1
[222] =	Counting direction 1	2.1.2/2.2.2
[223] =	Reference fine (input K0) 1	2.1.5
[224] =	Reference coarse 1	2.1.6/2.2.8
[225] =	SSI code 1	2.2.1
[226] =	no function	
[227] =	Number of SSI clockpulses 1	2.2.3
[228] =	Number of SSI databits 1	2.2.4
[229] =	Zero-point adjustment (SSI) 1	2.2.7
[230] =	no function	
[231] =	Maximum number of SSI errors 1	2.2.13
[232] =	SSI monitoring 1	2.2.14
[233]…[239] = no function		
[240] =	Counting direction ref. / zero-p. adjustment 2	2.3.7/2.4.9
[241] =	Input type and edge multiplication 2	2.3.1
[242] =	Counting direction 2	2.3.2/2.4.2
[243] =	Reference fine (input K0) 2	2.3.5
[244] =	Reference coarse 2	2.3.6/2.4.8
[245] =	SSI code 2	2.4.1
[246] =	no function	
[247] =	Number of SSI clockpulses 2	2.4.3
[248] =	Number of SSI databits 2	2.4.4
[249] =	Zero-point adjustment (SSI) 2	2.4.7
[250] =	no function	
[251] =	Maximum number of SSI errors 2	2.4.13
[252] =	SSI monitoring 2	2.4.14
[253]…[259] = no function		
[260] =	Baudrate (Canbus)	3.1.1
[261]…[263] = Data Obj/PDO1…3 Out		3.2.2…3.4.2
[264]…[269] = no function		
[270] =	Unit adress	4.1.1
[271] =	Baudrate (RS232)	4.2.1
[272] =	Stopbits (RS232)	4.2.2
[273] =	Parity (RS232)	4.2.3
[274] =	Protocol (RS232)	4.2.4
[275] =	Baudrate (RS422/485)	4.3.1

No	Description	Menu
[276]	= Number of stopbits (RS422/485)	4.3.2
[277]	= Parity (RS422/485)	4.3.3
[278]	= Protocol (R422/485)	4.3.4
[279]	= no function	
[280]	= Selection DA output	7.1.1
[281]	= Selection DA source	7.1.2
[282]…[297]	= Function output 1…16	6.1.1…6.16.1
[298],[299]	= no function	
[300]…[339]	= Camfunction	8.1.1…8.40.1
[340]…[379]	= Source cam	8.1.2…8.40.2
[380]…[419]	= Source for cam begin/limit value	8.1.3…8.40.3
[420]…[459]	= Source for cam end	8.1.4…8.40.4
[460]…[499]	= assign cam to output	8.1.8…8.40.8
[500]…[507]	= Function input 1…8	5.0.1…5.0.8
[508]…[511]	= no function	

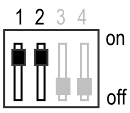
6 Connections

Connections on the rear



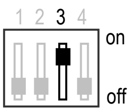
<u>PE</u>
To be connected with a 6,3mm faston

<u>RS 422/485</u>



If the AP90 is the last device, the DIP-switches 1 and 2 should be set to on.

<u>CANbus</u>



If the AP90 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

Encoder 1:

8. +10…35V DC supply output for encoder
9. +5V DC supply output for encoder
10. 0V supply for encoder
11. SSI-Clock+
12. SSI-Clock-
13. K1 or counting direction or SSI-Data+
14. /K1 or counting direction or SSI-Data-
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 1…6

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

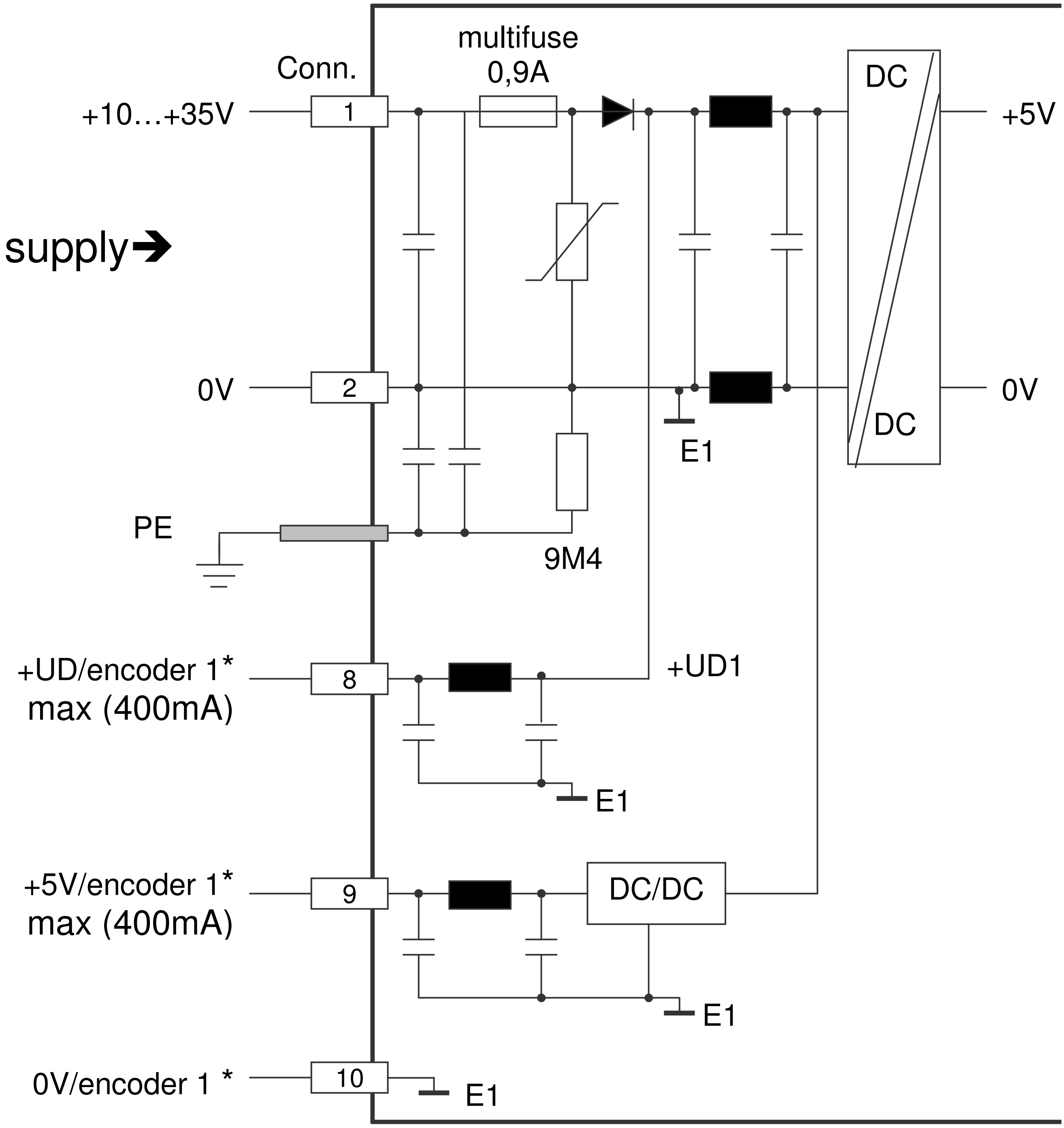
37. +10…+35V encoder input-2
38. 0V encoder input-2
39. +10…+35V (connected with clamp 37)
40. 0V (connected with clamp 38)
41. Input-7
42. Input-8
43. common for inputs 7 and 8

Encoder 2:

44. +10…35V DC supply output for encoder
45. +5V DC supply output for encoder
46. 0V supply for encoder
47. SSI-Clock+
48. SSI-Clock-
49. K1 or counting direction or SSI-Data+
50. /K1 or counting direction or SSI-Data-
51. K2 or counting pulse
52. /K2 or counting pulse
53. K0
54. /K0

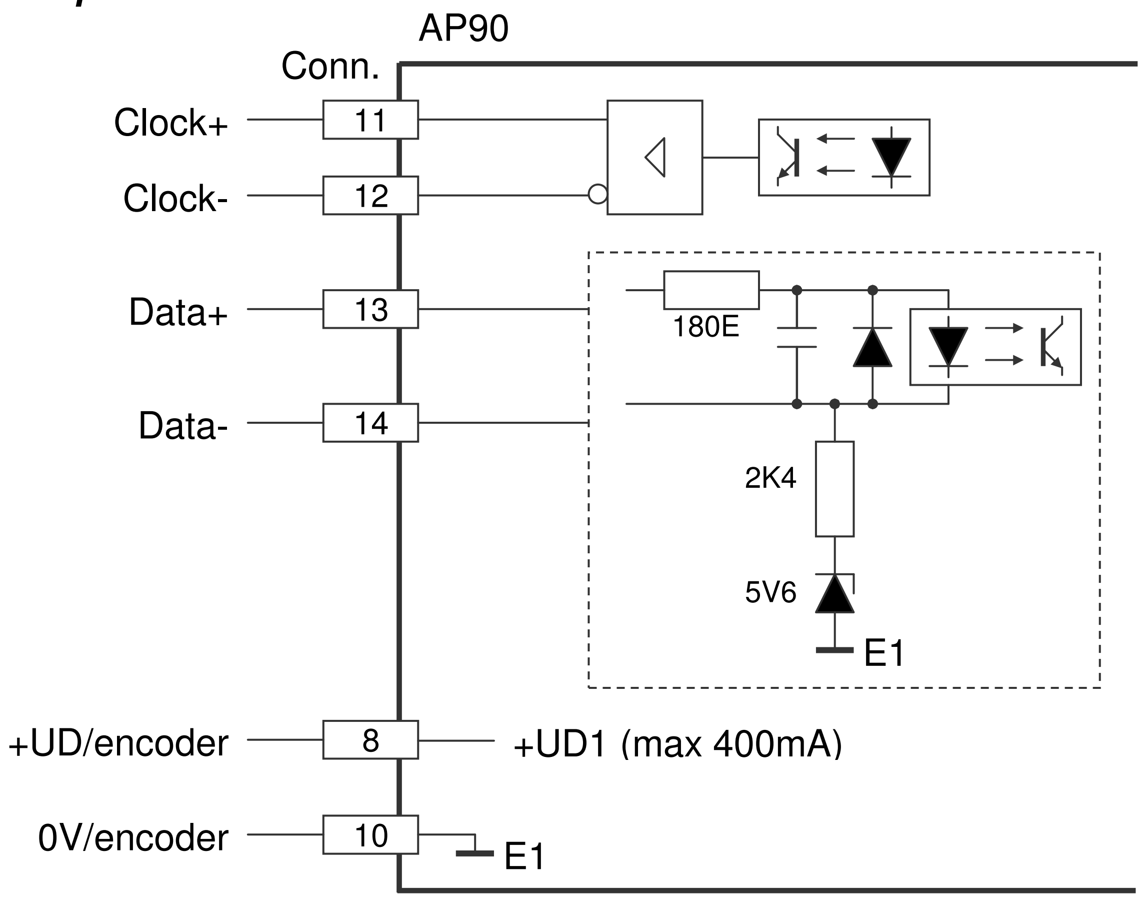
55. +U for outputs
56. 0V for outputs
57. Output -10
58. Output -11
59. Output -12
60. Output -13
61. Output -14
62. Output -15
63. Output -16

6.2 Supply

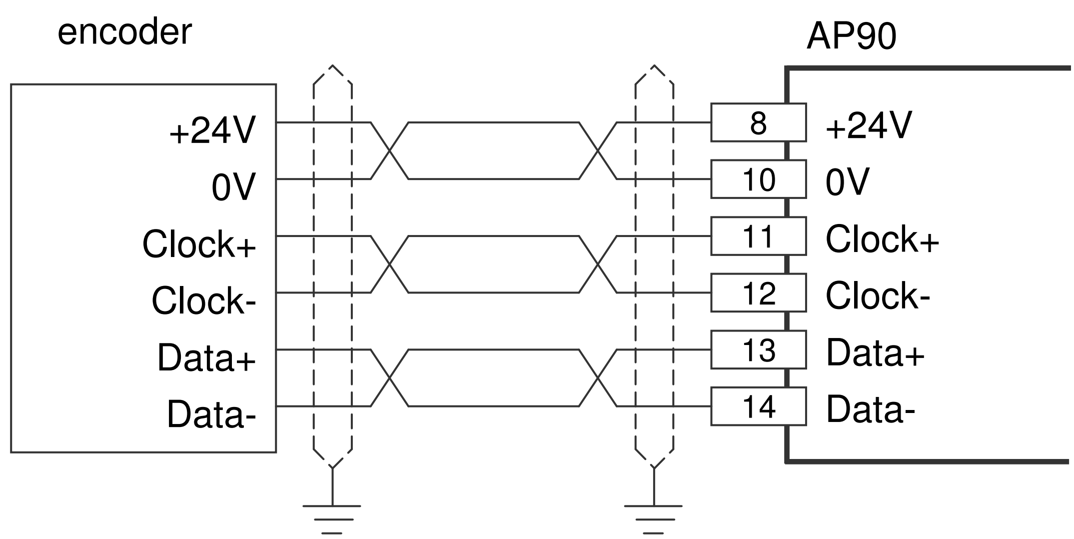


* supply output for encoder 1

6.3 SSI input 1

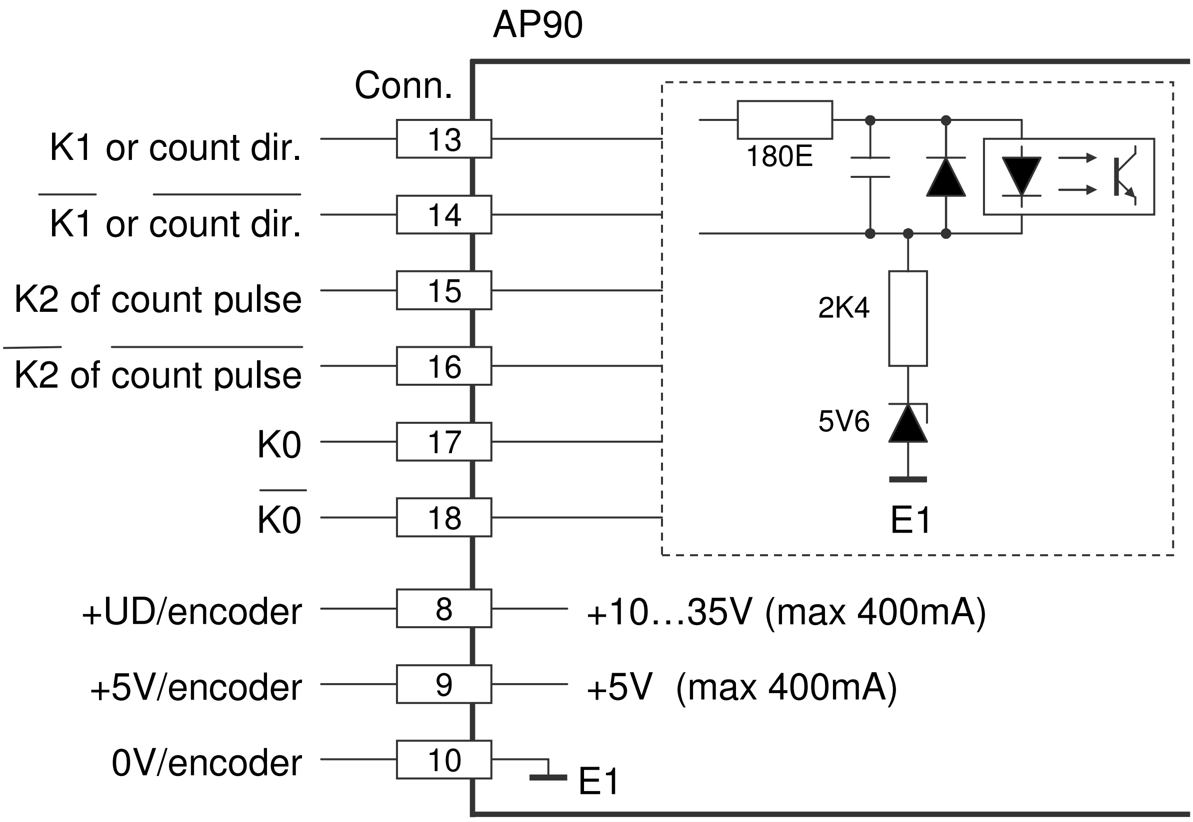


6.3.1 SSI encoder 24V



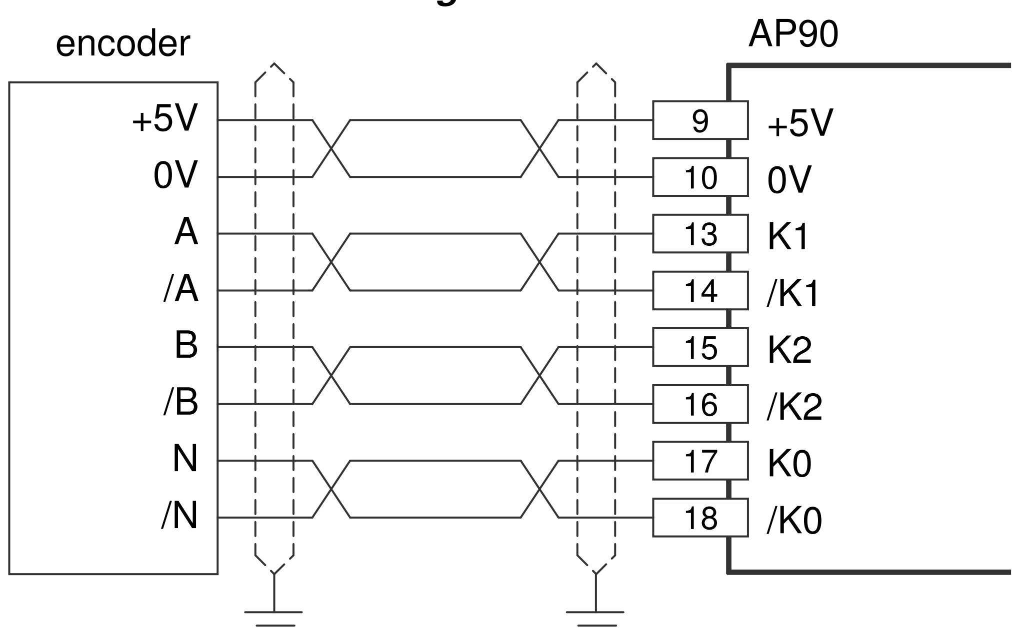
Supply voltage AP90 clamp 1 and 2 is 24V DC

6.4 Counting input 1

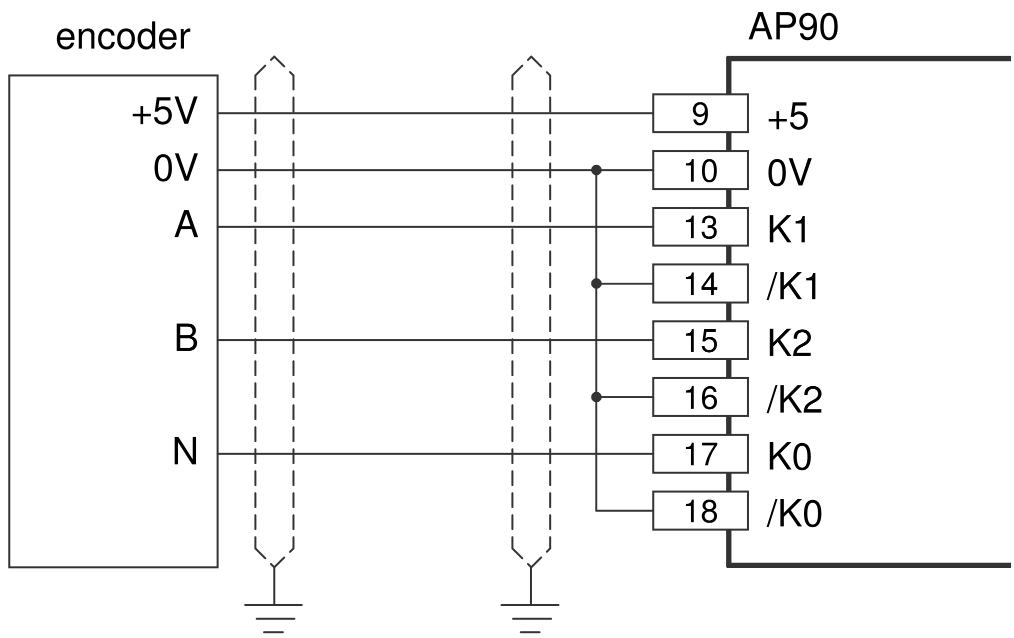


It is possible to use different voltage signals for K0 and K1, K2. For example: encoder signals (K1/K2) with a level of 5V and a reference fine (K0) with 24V level.

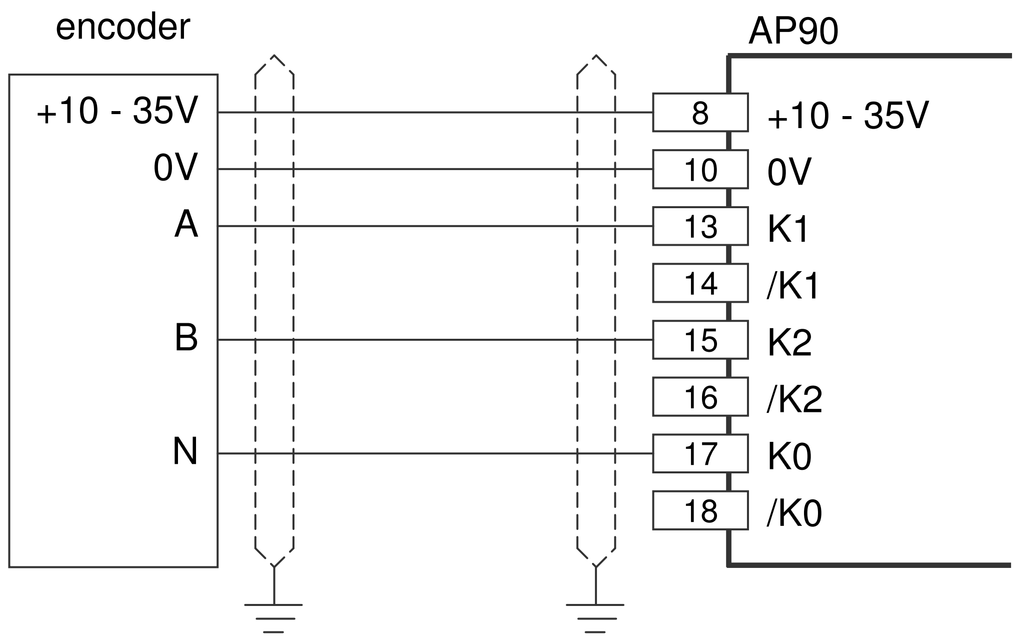
6.4.1 Encoder 5V with inverted signals



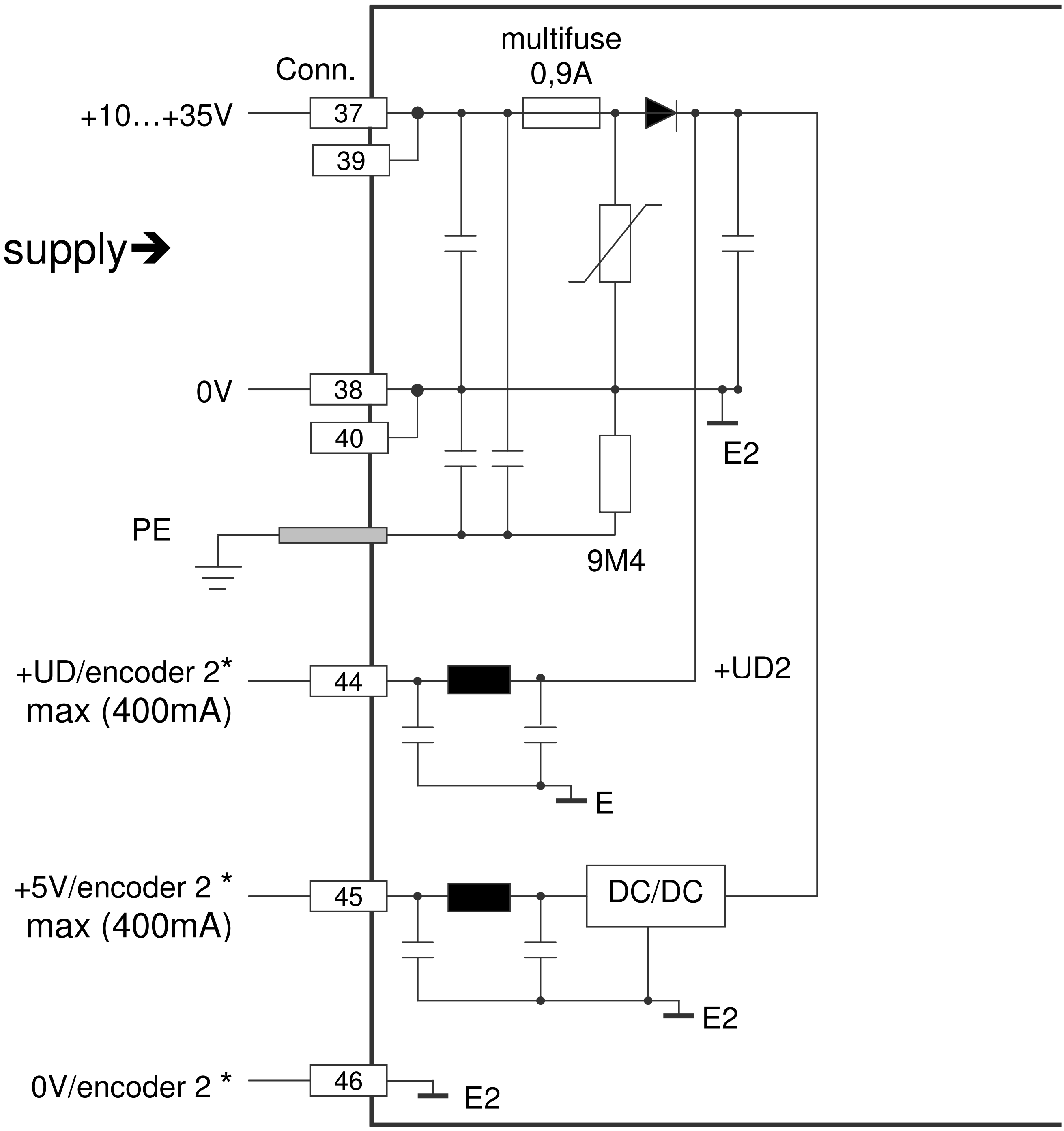
6.4.2 Encoder 5V without inverted signals



6.4.3 Encoder 10 – 30V

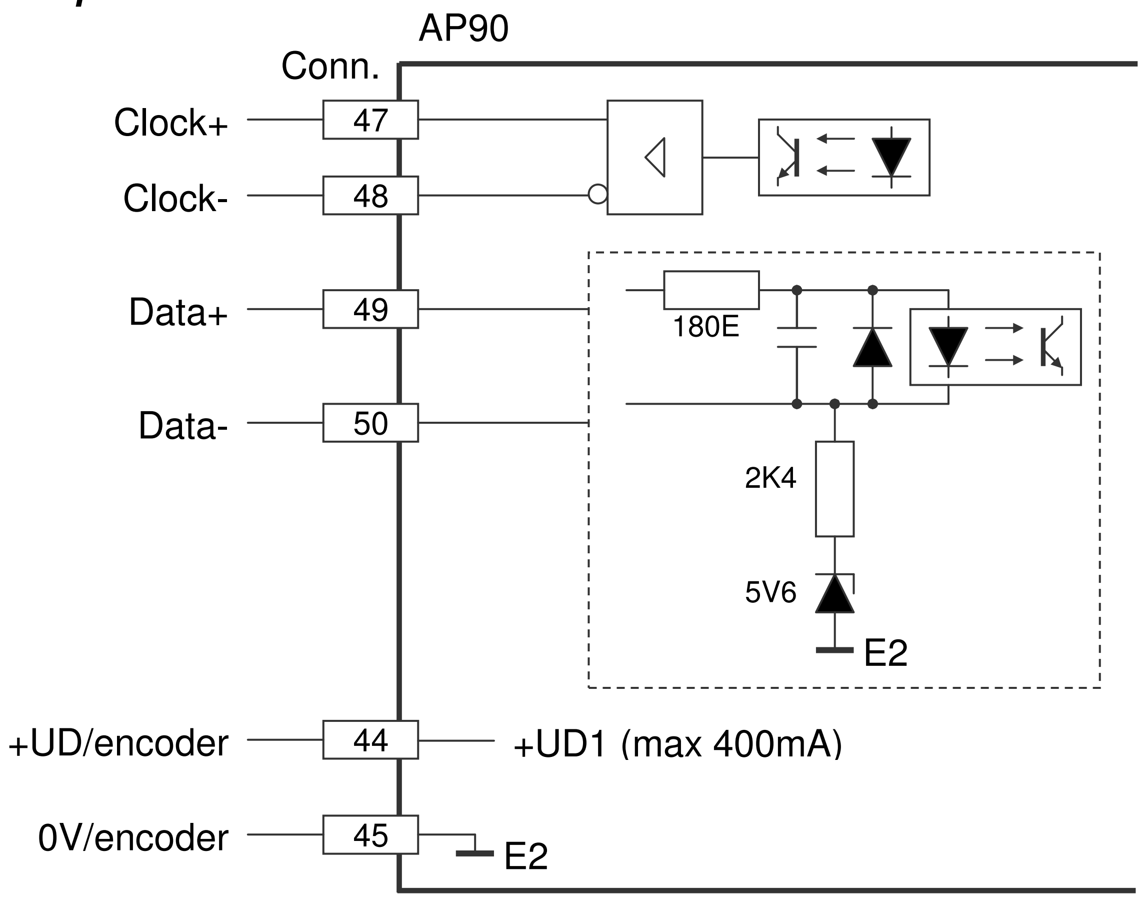


6.5 Supply (encoder 2)

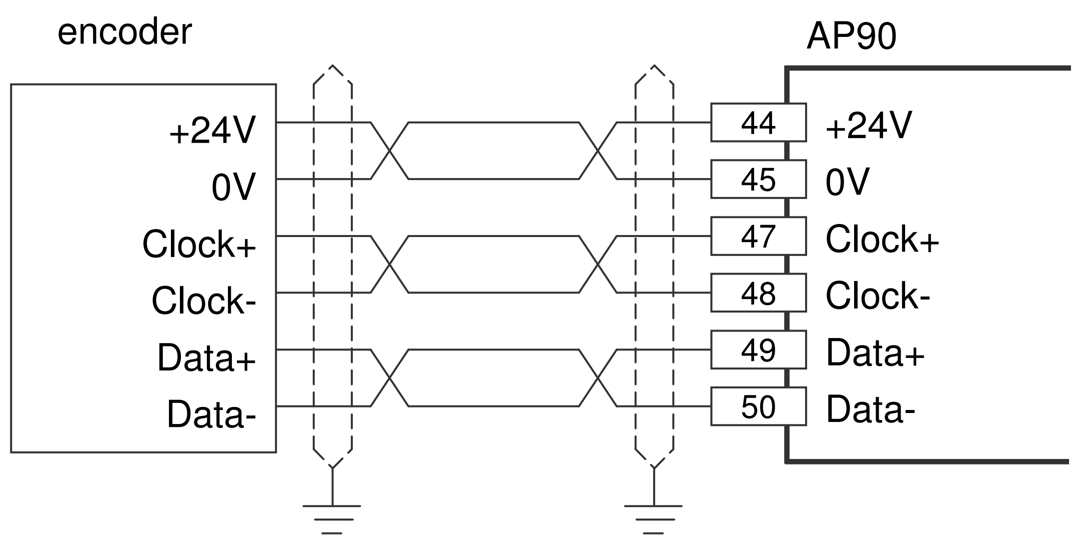


* supply output for encoder 2

6.6 SSI input 2

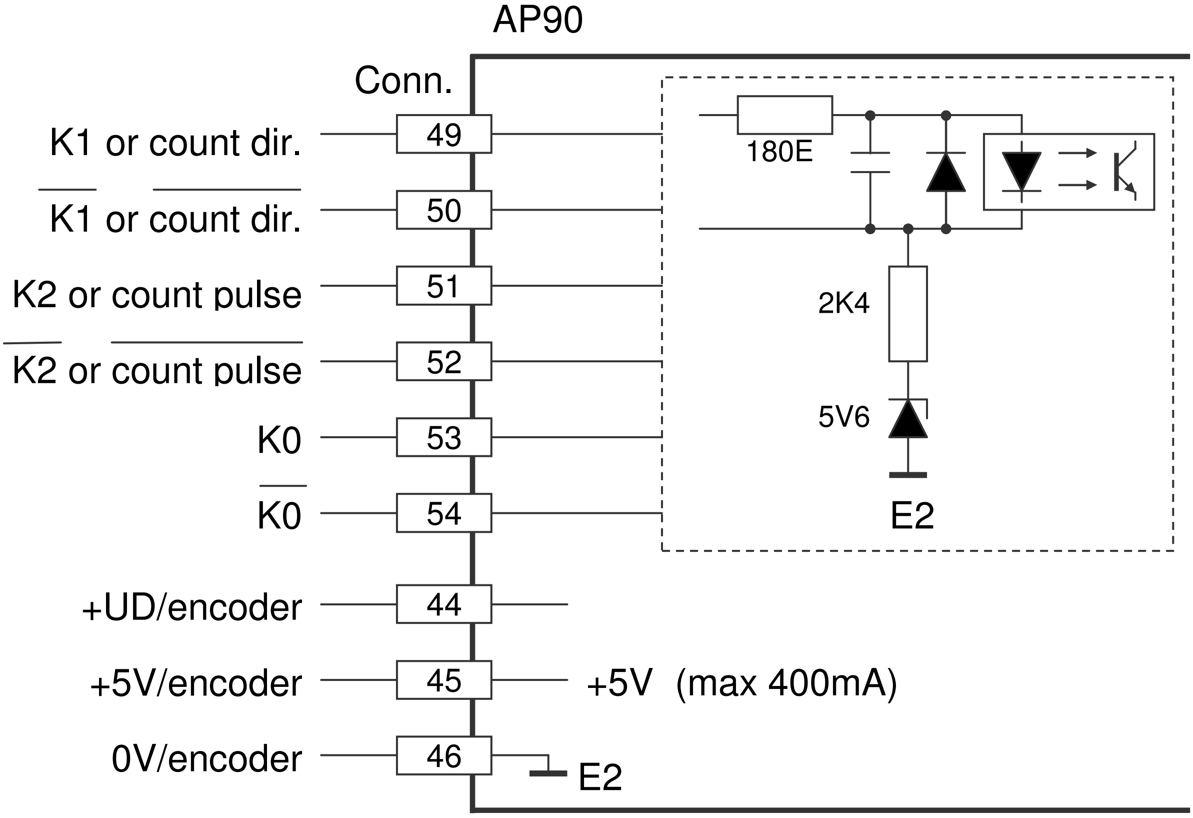


6.6.1 SSI encoder 24V



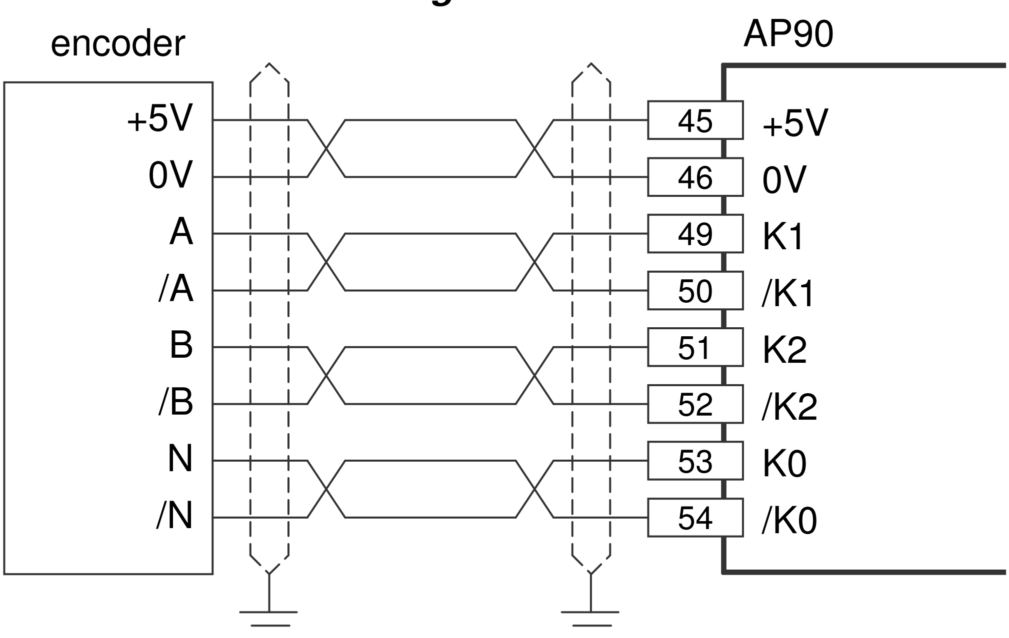
Supply voltage AP90 clamp 37 and 38 is 24V DC

6.7 Counting input 2

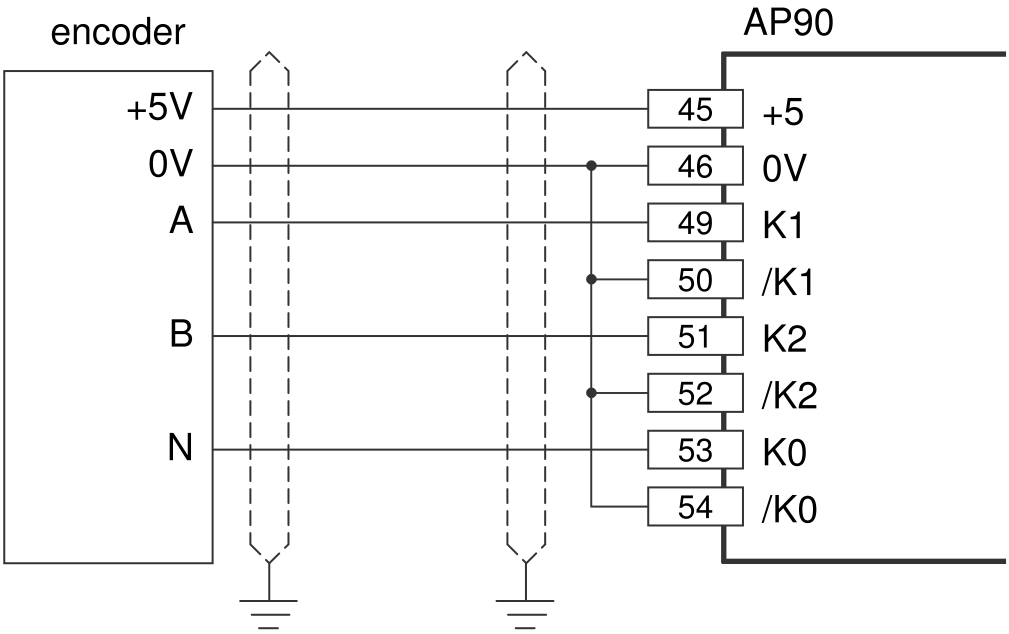


It is possible to use different voltage signals for K0 and K1, K2. For example: encoder signals (K1/K2) with a level of 5V and a reference fine (K0) with 24V level.

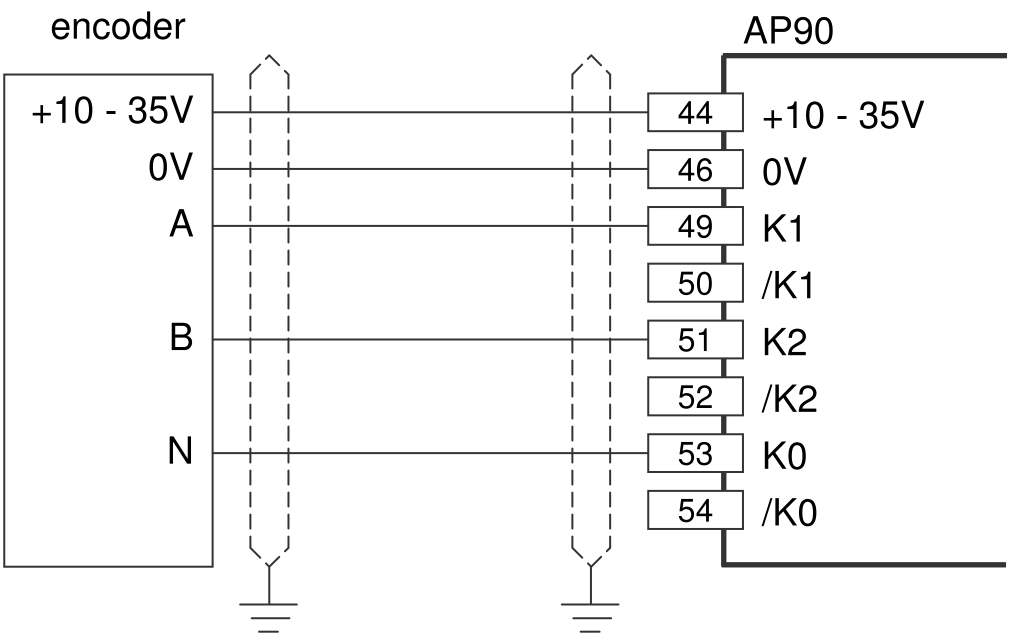
6.7.1 Encoder 5V with inverted signals



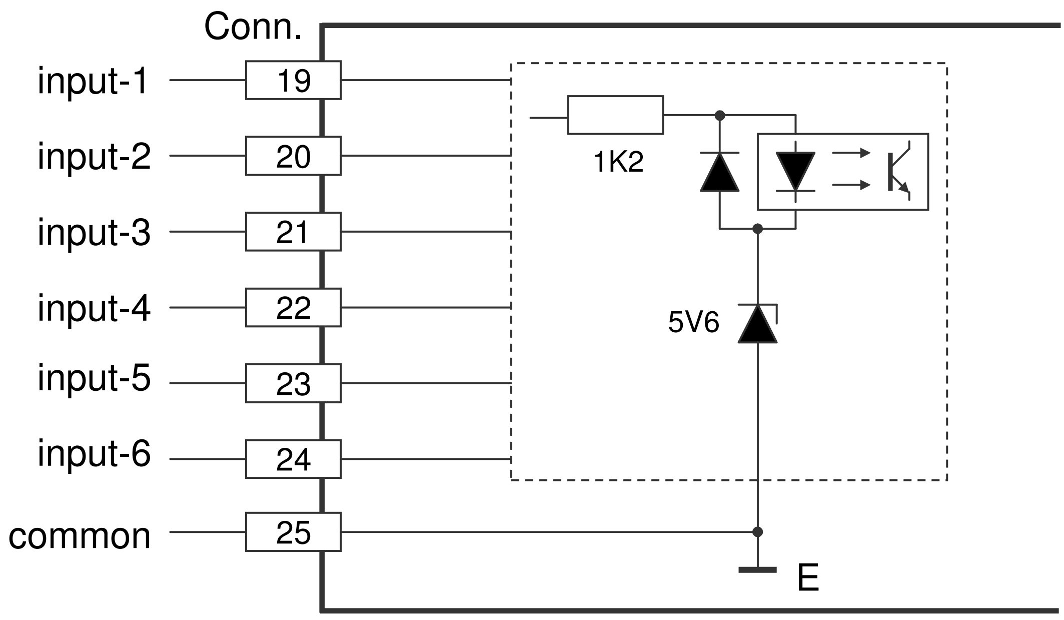
6.7.2 Encoder 5V without inverted signals

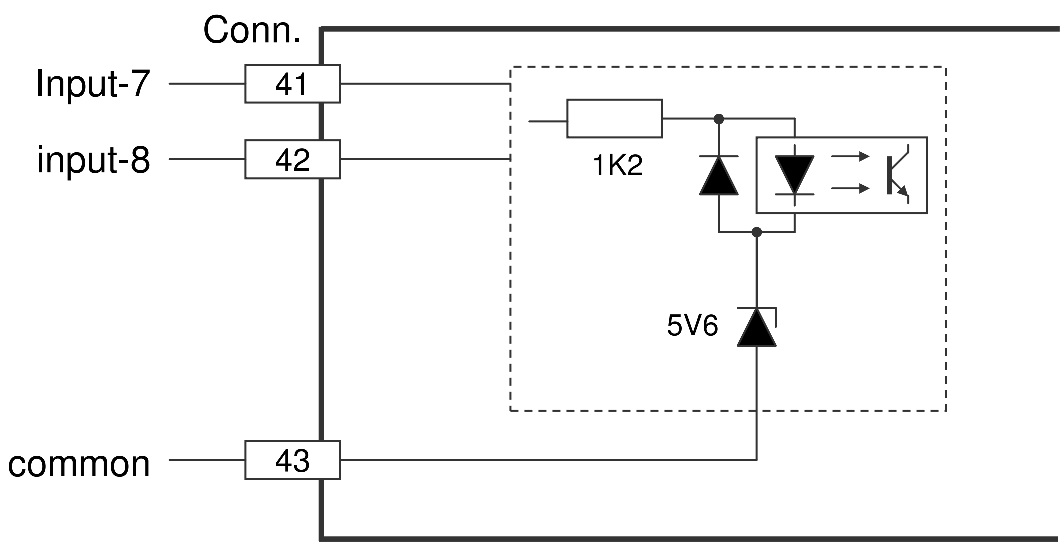


6.7.3 Encoder 10 – 30V

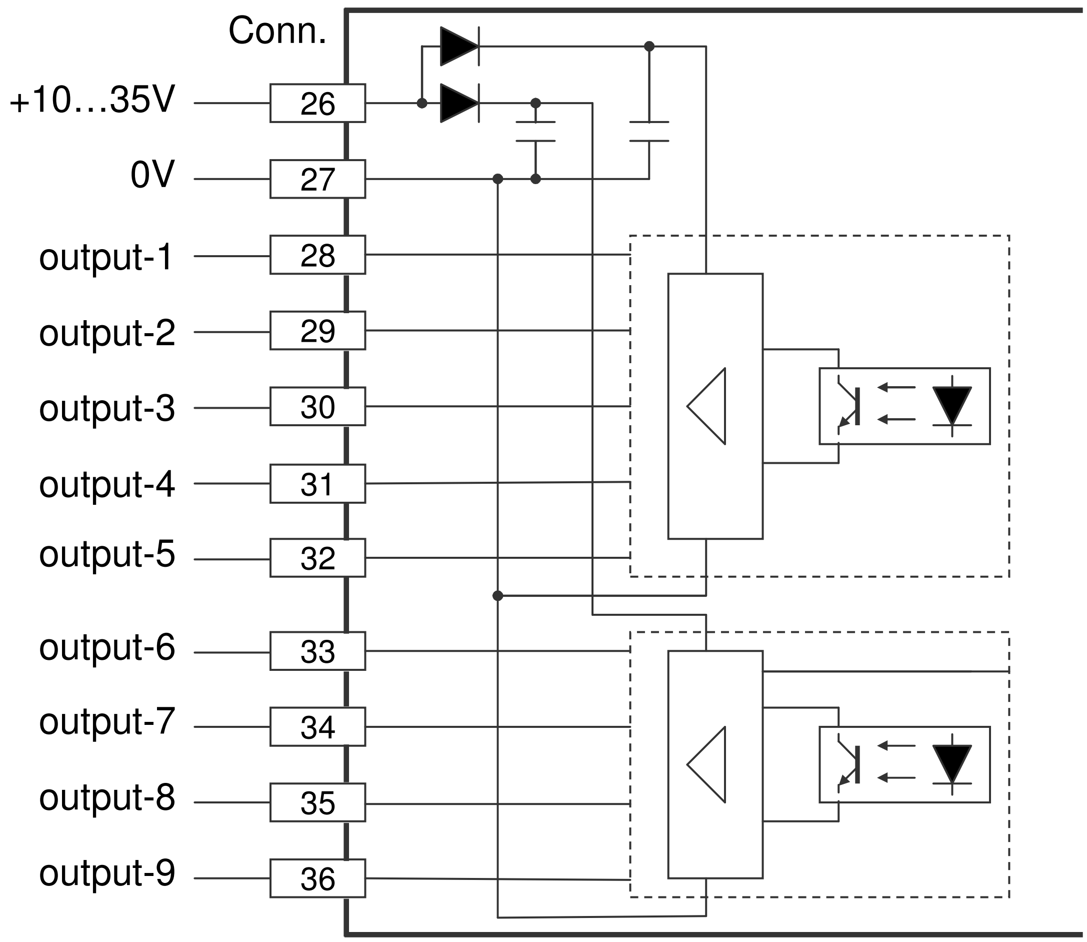


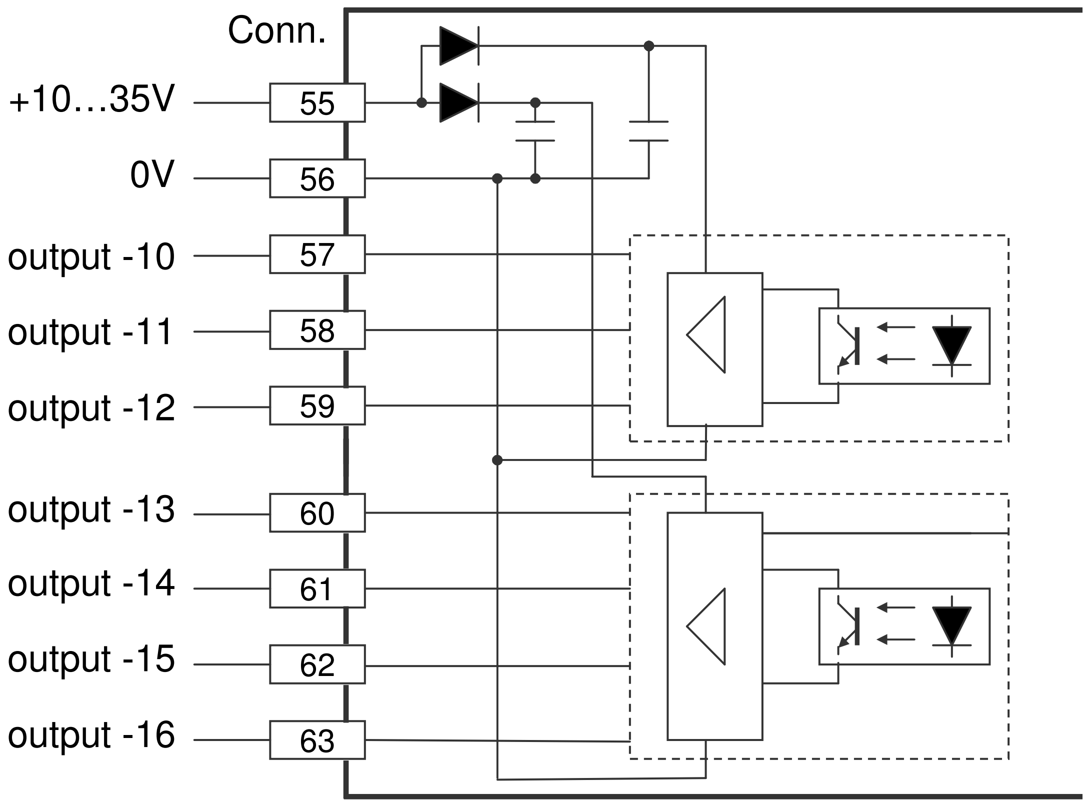
6.8 Digital inputs



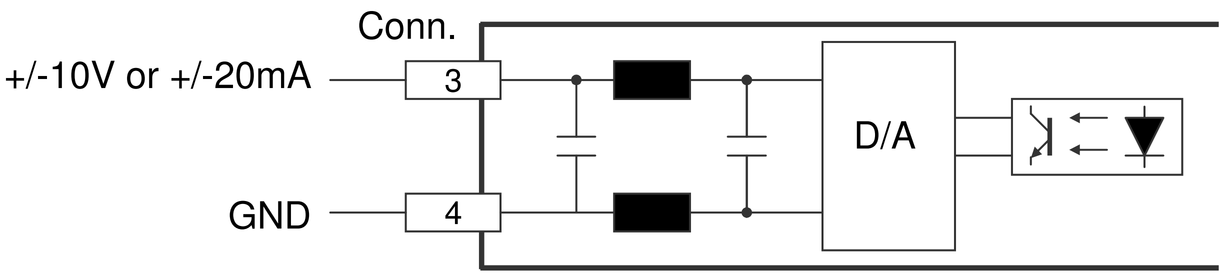


6.9 Digital outputs

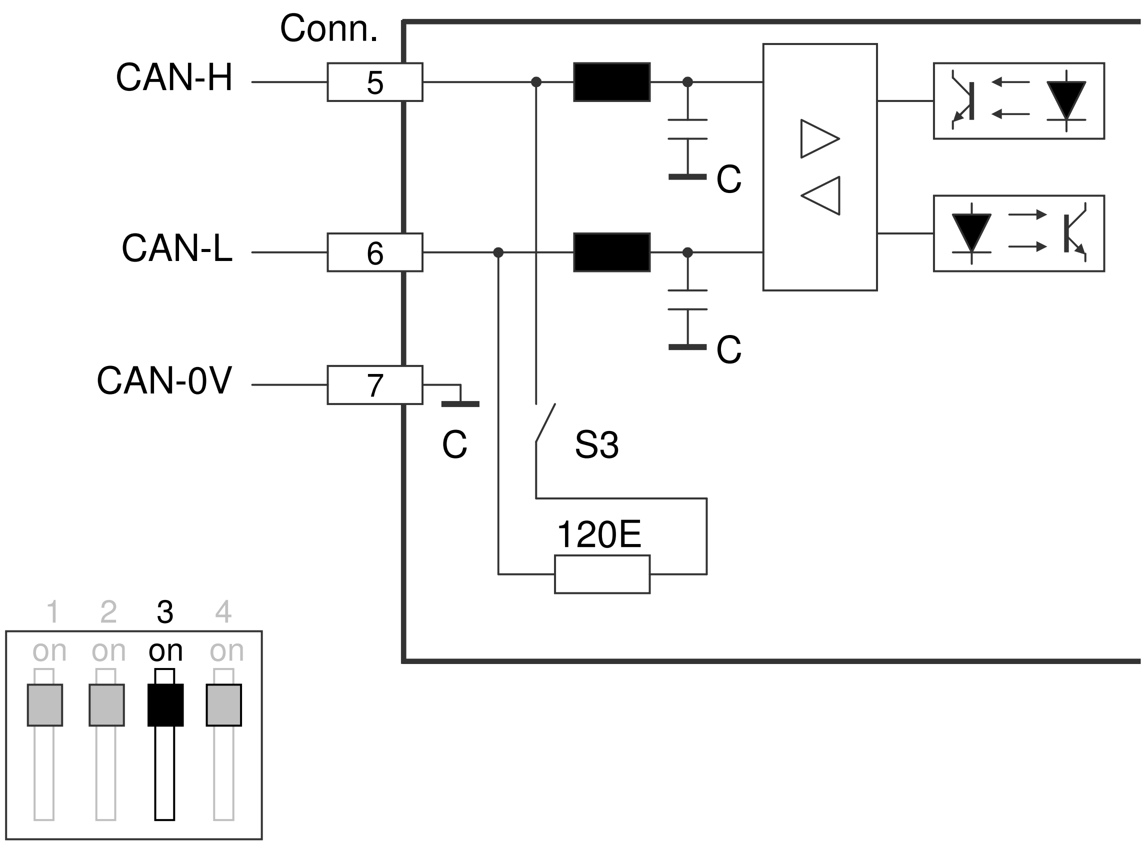




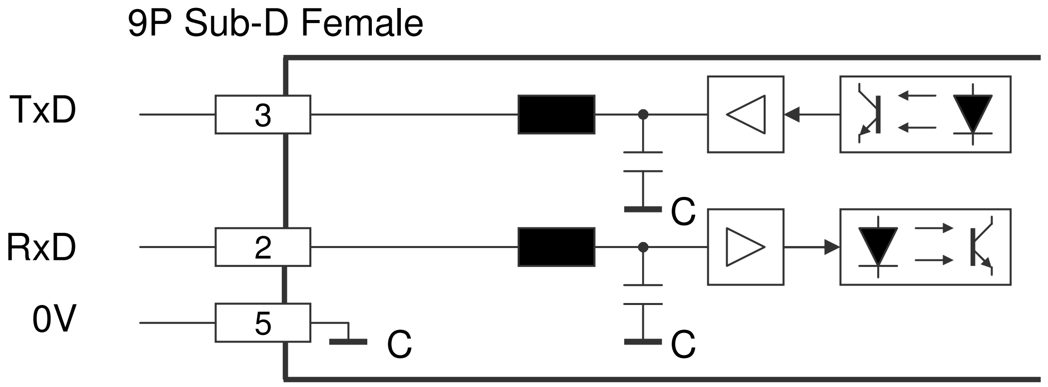
6.10 Analog output



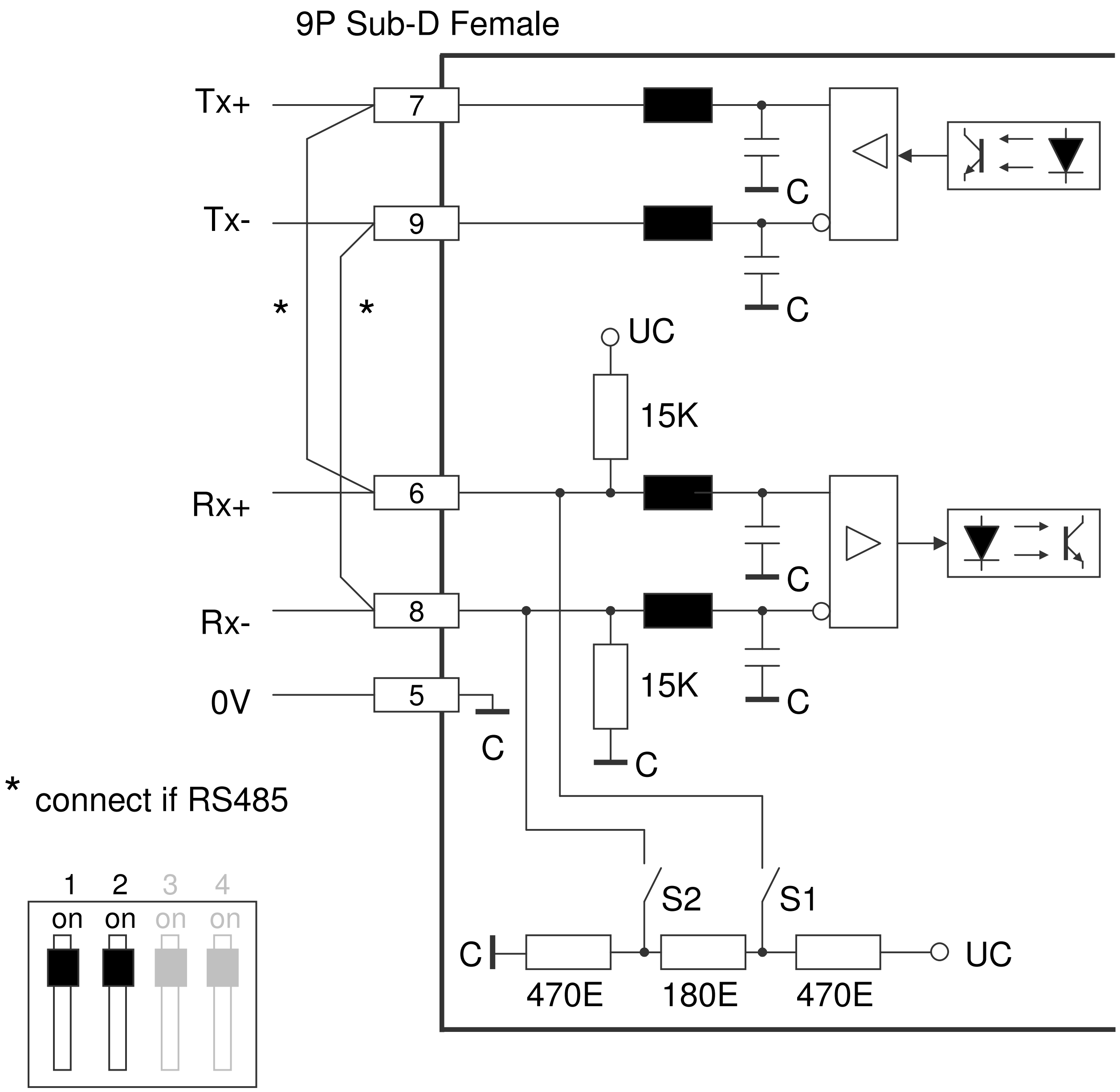
6.11 CAN-bus



6.12 RS232 Ser-1



6.13 RS422/485 Ser-2



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	500µS (fixed)
Counting range	-9999999…+99999999
- Counting input 1,2	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 1,2	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)
- Digital inputs 1…8	optically isolated
	low: 0…+5V
	high: +10V…+35V
input resistor	appr. 1.8kOhm at 24V

- Digital outputs 1…16	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
Update	5ms
- Current output	galvanically isolated
range	max. –20mA … +20mA
resolution	610 µA
offset-temp. coeff.	< 20 ppm/ °C
Rmax	550 Ohm
Update	5ms
- 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°C
- Connection diameter	1,6 mm^2
- 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

AP90- C X E

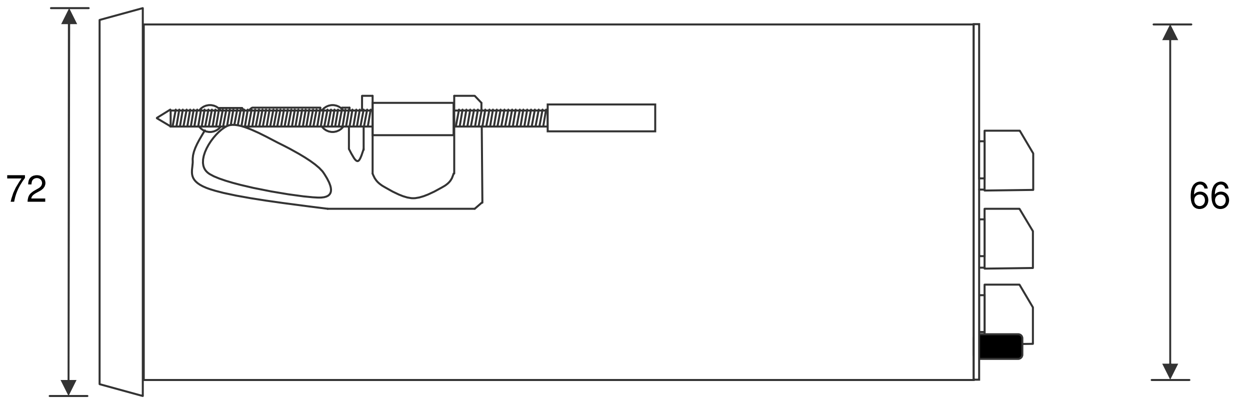
Second encoder input
E = Encoder 2

Analog output
0 = no analog output
A = analog output

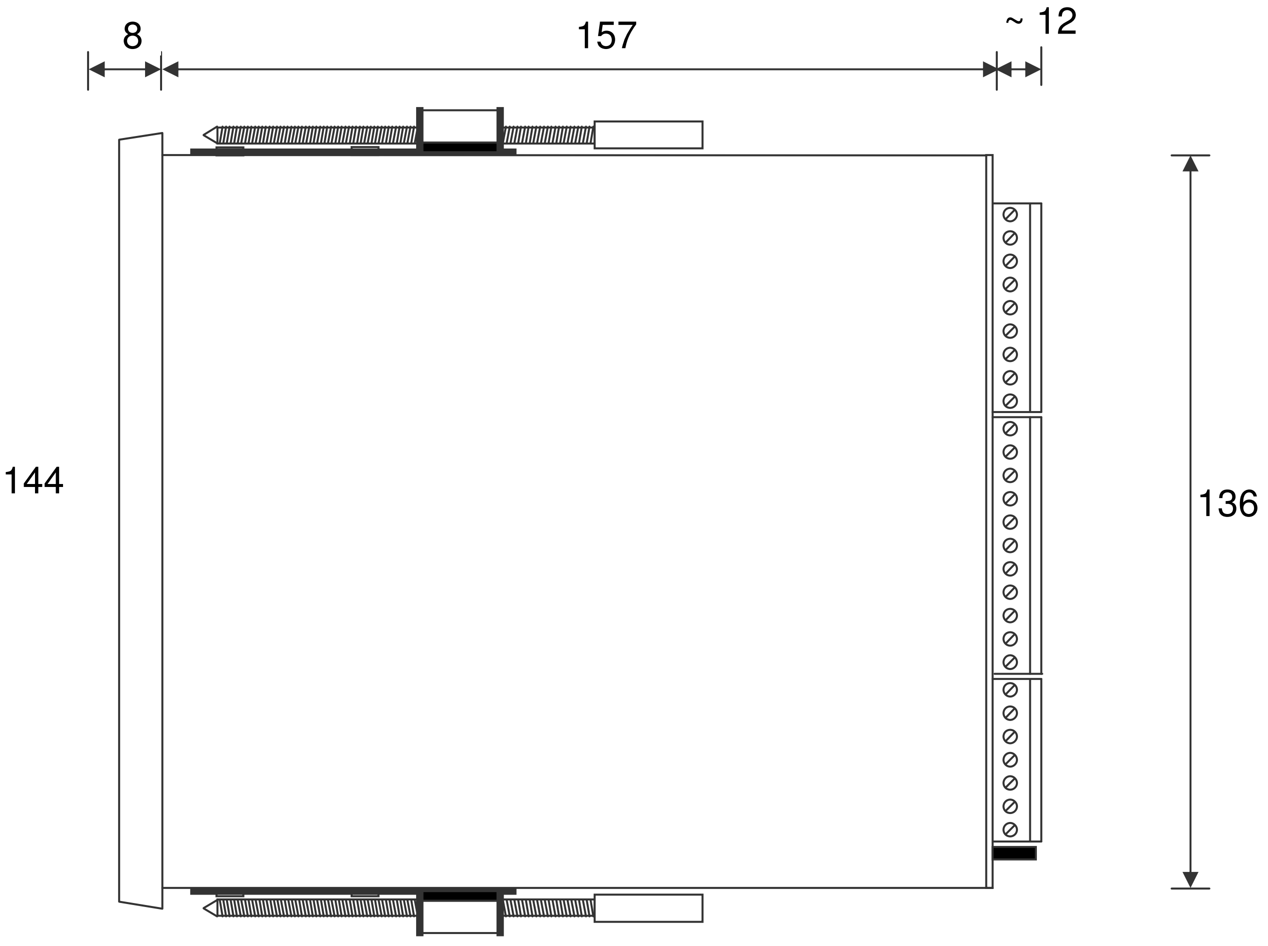
Serial communication
C = RS232, RS422/485 and CANbus

7.3 Dimensions AP90

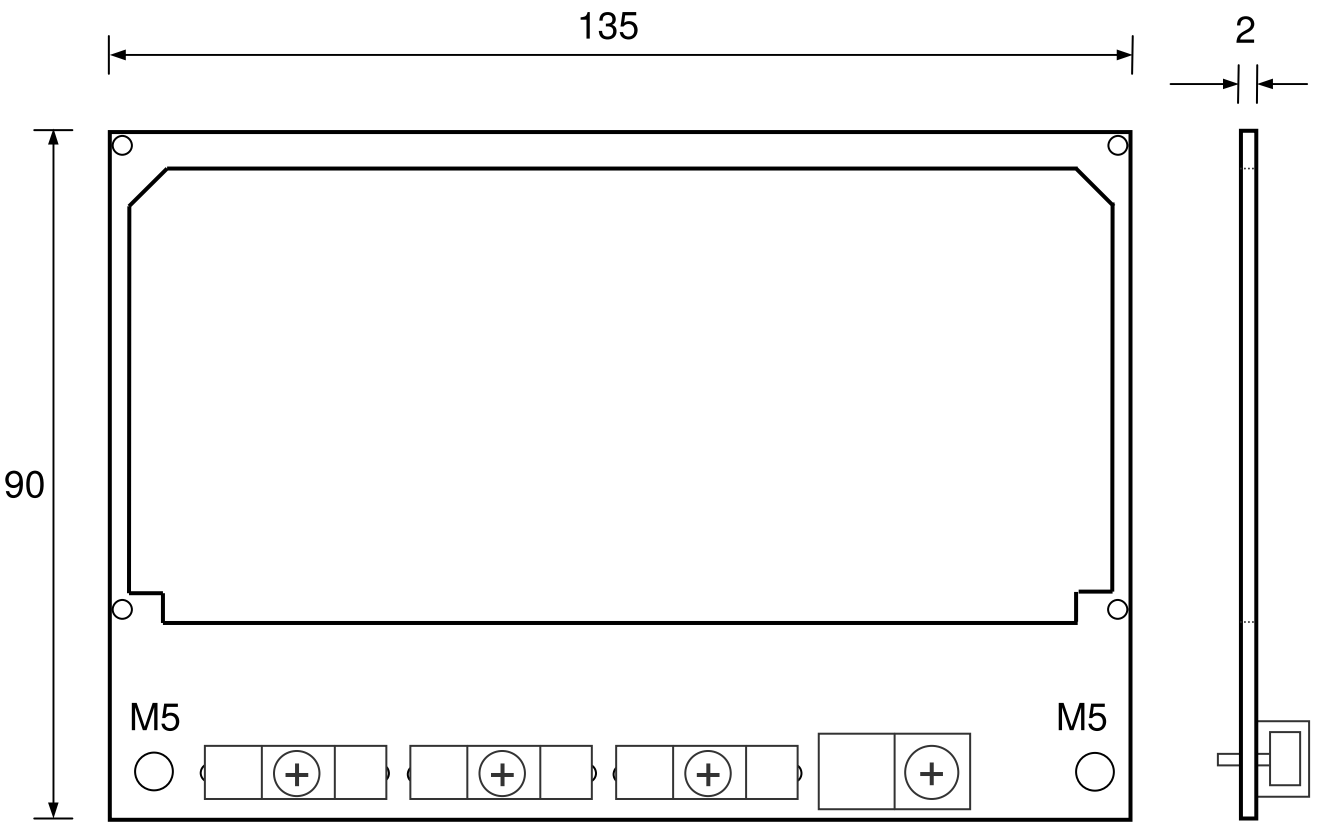
Side view



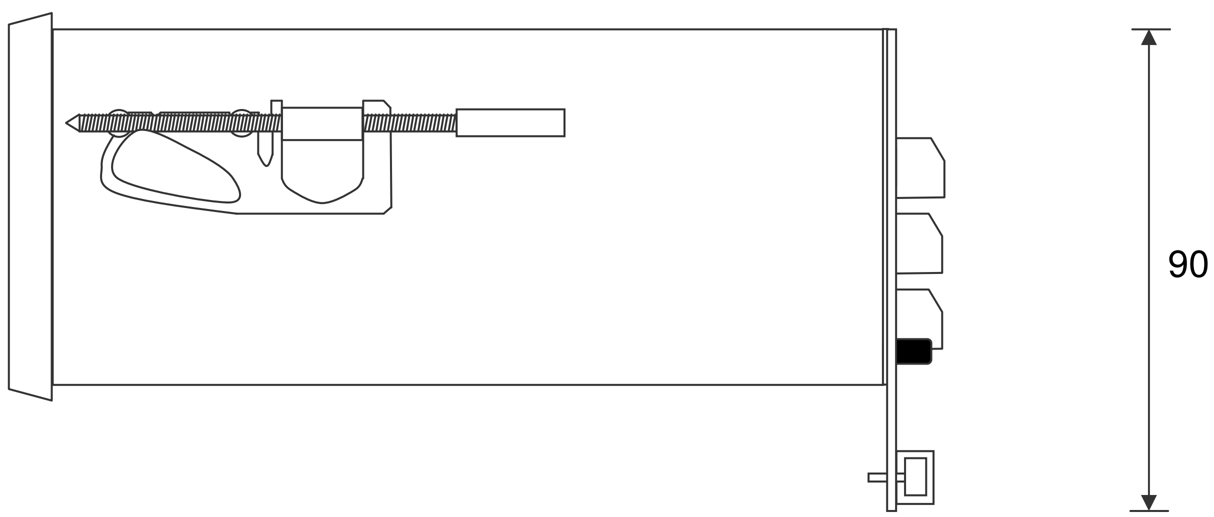
Top view



7.4 Dimensions EMC bracket type EMC-B01



Zijaanzicht met EMC beugel



7.5 Dimensions protective hood type CDS-B01