INTELEKTRONIK

Wejście binarne 8 kanałów 12-35 V DC z izolacją wejść

Podstawowe funkcje:

- Niezależne przełączanie 8 wejść,
- Niezależna reakcja wejść na zbocze rosnące, opadające,
- Opóźnienie wejścia,
- Dwa niezależne obiekty dla każdego wejścia,
- Funkcja ściemniania ,
- Funkcja rolet,
- Funkcja nadawania temperatury, jasności,
- Funkcja licznika impulsów (tylko wejścia 1,2) z możliwością zewnętrznej synchronizacji lub kasowania licznika (wejścia 3,4),
- Funkcja licznika impulsów (tylko wejścia 1,2) z możliwością automatycznego wewnętrznego kasowania (wejścia 3,4),
- Blokada wejścia,
- Reakcja na początek i koniec blokady dla funkcji przełączania, ściemniania , rolet,
- Opóźnienie po powrocie KNX BUS, reakcja po powrocie KNX BUS
- Cykliczne wysyłanie stanu wejścia.

Zasada działania:

Wejście 1:

Object	Name	Function	Type	Flag
Function: Switching (for all inputs)				
0 – 7	Input1 – 8	Switching	1 Bit	C, W, T, (R)
Function: Dimming (for all inputs)				
0 – 7	Input 1 – 8	Switching	1 Bit	C, W, T, (R)
8 – 15	Input 1 – 8	Dimming	4 Bit	C, T, (R)
Function: Blind/shutter control (for all inputs)				
0 – 7	Input 1 – 8	Short time operation	1 Bit	C, T, (R)
8 – 15	Input 1 – 8	Long time operation	1 Bit	C, T, (R)
Function: Dimming value transmitter (for all inputs)				
0 – 7	Input 1 – 8	Value	1 Byte	C, T, (R)
Function: Light scene extension (for all inputs)				
0 – 7	Input 1 – 8	Light scene extension	1 Byte	C, T, (R)
Function: Temperature value transmitter (for all inputs)				
8 – 15	Input 1 – 8	Temperature value	2 Byte	C, T, (R)
Function: value transmitter (for all inputs)				

8 – 15	Input 1 – 8	Brightness value	2 Byte	C, T, (R)
Function: Impulse counter (for inputs 1 and 2 only)				
2	Input 3	Synch signal counter 1	1 Bit	C, W, T, (R)
3	Input 4	Synch signal counter 2	1 Bit	C, W, T, (R)
8	Input 1	Meter reading counter 1	2 Byte	C, T, (R)
9	Input 2	Meter reading counter 2	2 Byte	C, T, (R)
Function: Switching counter (for inputs 1 and 2 only)				
0	Input 1	Switching counter	1 Bit	C, W, T, (R)
1	Input 2	Switching counter	1 Bit	C, W, T, (R)
8	Input 1	Switching counter	2 Byte	C, T, (R)
9	Input 2	Switching counter	2 Byte	C, T, (R)
Function: Blocking (for all inputs)				
16 – 23(*)	Input 1 – 8	Blocking	1 Bit	C, T, (R)

Liczba adresów możliwych do zaprogramowania – 26
Liczba obiektów komunikacyjnych – 24

Objects marked with (R):
Object value can be read out (set R-flag!)

Objects marked with (*):
If the inputs have been parameterised to "no function", "impulse counter" or "switching counter", the blocking function is not active.

Impulse counter:

The pulse counters can only be parameterised to input 1 or 2. In this case, inputs 3 (for pulse counter 1) or 4 (for pulse counter 2) are the sync signal inputs and cannot be assigned to any other function. Pulse counters 1 and 2 run independently of each other and have a resolution of 16 bit so that counts between 0 and 65535 are possible. You can set the R-flag to read out the current count at object 8 or 9. The counting pulse is applied to input 1 or 2. After the interval time specified as parameter has elapsed, the count will be taken over and sent as object value of the 2 byte "count" object (object 8 or 9). Then the 2 byte pulse counter will be internally reset during the next time interval.
Only upon the appearance of a new edge at the input, or after the newly started interval time has elapsed, the current count can be read out from the count objects (set R-flag).
In addition, the count and the interval time can be reset by a sync signal applied to input 3 or 4, respectively. Moreover, switch telegrams (no telegram, ON, OFF, TOGGLE) can be sent in dependence of the sync signal edge. The output value can be assigned to the edge.
The edge assignment for resetting the count can be parameterised independently of the output value.
For pulse counting, the mark-to-space time of a signal applied to input 1 or 2 must not fall below 100 ms.

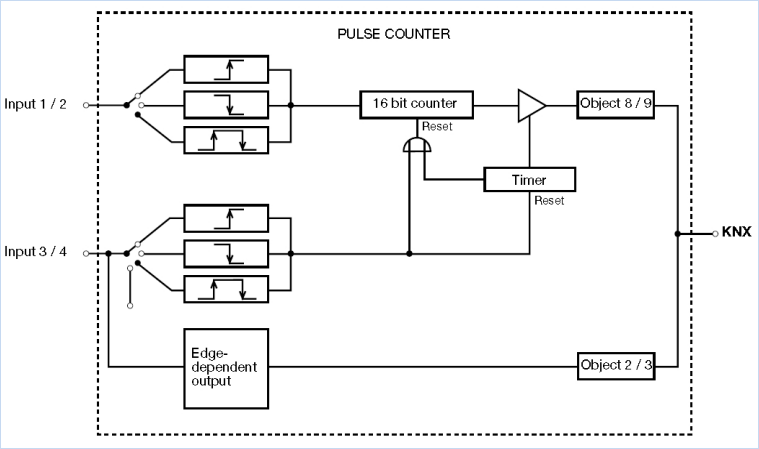

Fig.1 Impulse counter

Switching counter:

The switching counters can only be parameterised to input 1 or 2. Switching counters 1 and 2 run independently of each other and have a resolution of 16 bit so that counts between 1 and 65535 are possible. You can set the R-flag to read out the current count at object 8 or 9. The counting pulse is applied to input 1 or 2. After the count has reached the parameterised set value, it will be taken over into 2 byte object 8 or 9 and transmitted. Simultannously, it is possible to output a signal value (1 bit object "0" or "1") which can be parameterised. After the transmission, the 16 bit counter will be automatically and internally reset. Only upon the appearance of a new edge at the input, the current count can be read out from the count objects (set R-flag).
Moreover, the count will be sent in cycles after a pre-defined number of counting pulses (1 … 255), which is used to get an automatic update on any display for instance.
For switch counting, the mark-to-space time of a signal applied to input 1 or 2 must not fall below 100 ms.

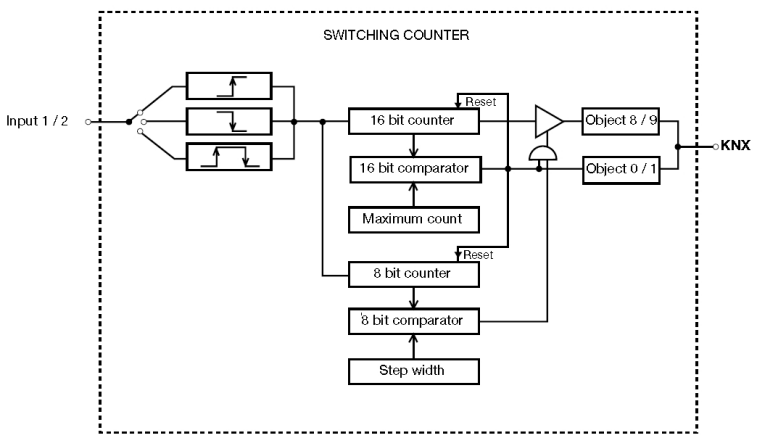

Fig.2 Switching counter

Blocking function:

At the beginning or at the end of the blocking, an independent response can be made to each input. In this case, you can set the parameter to "no response". Only in such case, any dimming or blind/shutter control or value changing events running until the action is completed during an active blocking. In any other cases, the parameterised command will be sent immediately at the start of blocking. Moreover, any edges or signals at the corresponding inputs will not be accepted during an active blocking.
Updates on blocking objects (disable or enable) each time cause the corresponding parameterised command to be sent "at start or end of the blocking".
During an active blocking, there will be no cyclic sending through the disabled input.
If cyclic sending was taking place prior to an activation of the blocking function, no more cyclic sending will be performed at the end of the blocking, provided that "no telegram" has been parameterised. In this case, the cyclic transmission of the object value will only be effected again after an update on the switching object. In any other cases, the object value will be sent in cycles again after the end of blocking.
