

Quick Start

Logic function for electronic cam controls

Deutschmann Automation GmbH & Co. KG www.deutschmann.com | wiki.deutschmann.de

Logic

The logic function can only be activated with the **WINLOC32** software. More information on the WINLOC32 software can be found in the manual **"WINLOC32 Configuration & Programming**" (V3969E).

The logic function for a corresponding cam controller can be activated via the menu item **"Extras" -> "Options" -> "Optional product activation"** if a corresponding license key has been purchased for this cam controller.

Hinloc/32 V3.43											_ 🗆 🗡
File Edit Extras C	Online View V	Window He	elp								
产 🖥 🖸 Opti	ons			🔜 💻		5			C1	C2 C3	
Net	Observer										
	aaro data										
	Jai e uata										
Data	a transfer										
S7 T	ool										
Or Prod	luct Option Activ	vation									
Upload 1			,								
- opioud 1			46/4000								
program ju		iemory	10/1000	english							
L	<-> name	scams	ITC turn. dir.	Comment	' b	00 '	400 '	600 '	800	' 🔳	
<u>G</u> raphic	Out 1	0-50;	0,0 both								
	Out 2	50-100;	0,0 both								
Configuration	Out 3	100-150;	0,0 both								
Comment	Out 4	150-200;	0,0 both		_						
Comment	Out 5	200-250;	0,0 both								
- <u>e</u> r Louie	Out 6	250-300;	0,0 both								
Eogle	Out 7	300-350;	0,0 both								
Analon	Out 8	350-400;	0,0 both								
Sumply	Out 9	400-450;	0,0 both								
	Out 10	450-500;	0,0 both				-				
	Out 11	500-550;	0,0 both				- -				
	Out 12	550-600;	0,0 both				- 7			_	
B 100 V3.7 P:64 A	16 F:A 1000 bit	twise	unch	ange 132			_	-			1
			juner								T
COM 1 dicnet Die	Adap3 V1.3		Winloc/32 V	3.43							

The serial number of the cam controller for which the logic function is to be enabled can be read out using the "**read serial number**" button. The license key, which is linked to the respective serial number, must be entered and loaded with the **"Run"** button. The logic function is then enabled. If the cam controller is restarted, the addition **"LG"** appears in the switch-on message.

Note: In the case of an RS485 (DICNET) connection via a DICNET adapter from the cam controller to WINLOC32, the device ID of the device concerned must be entered.

4



R100-16-04096-LG-232 V3.73

After another upload from the cam controller, the logic function can be selected and programmed.



The following window is initially empty. A basic device appears with, if necessary, the corresponding attached expansion modules. The basic device and the extension modules can be selected individually.

By clicking with the right mouse button in the empty window, a selection window appears. A new logic block can be selected in this.

In the following window an output or a marker can be selected.

Editor	
M1.I/O17	OK
M1.I/O18	
M1.I/O19	Cancol
M1.I/O20	Cancer
M1.M1	
M1.M2	
M1.M3	
M1.M4	
M1.M5	
M1.M6	
M1.M7	
M1.M8	
	6

Programm 0	▲ Speicher 1/0/392 Deutsch	
<u> </u>	M1.I/O17 : M1.M1 Steigend 0ms	
<u>G</u> rafik	C Input O Mark	
K <u>o</u> nfiguration	C Shift C Cam M1.I/017	
<u>K</u> ommentar	M1.I/O19	
:E-	M1.I/O20	
<u>L</u> ogik	M1.I/O21	
	M1.//022 M1.//023	
Analog	M1.I/024	
Modu X		7

You can then select when the flag or the output should switch.

A logic function can be added by right-clicking in the logic column. Then, by clicking the left mouse button, e. g. another selection window can be selected on **AND**.

Note: The left mouse button must be held down when the window opens. Only when the mouse pointer is on the window can the left mouse button be released. This applies to all values that can be selected in the column.



In the example, 4 outputs and 4 inputs were configured for the expansion module. Therefore, in addition to 8 flags, the 4 outputs now appear.

Example:

M1.I / O17: M1.I / O21 AND M1.N17

M1 stands for the 1st expansion module. **I / O17** has been configured as an output. **I / O21** has been configured as an input. **N17** stands for the cam / s that have been programmed at output 17.

Output 17 only switches when input 21 is switched and the cam (s) at output 17 come.



42.0	Logic_Table.Logic[1].ProgNo	BYTE	B#16#00	B#16#00
43.0	Logic_Table.Logic[1].DestNo	BYTE	B#16#00	B#16#01
44.0	Logic_Table.Logic[1].DestType	BYTE	B#16#00	B#16#00
45.0	Logic_Table.Logic[1].OpNo1	BYTE	B#16#00	B#16#05
46.0	Logic_Table.Logic[1].OpType1	BYTE	B#16#00	B#16#01
47.0	Logic_Table.Logic[1].LogicFct1_2	BYTE	B#16#00	B#16#02
48.0	Logic_Table.Logic[1].OpNo2	BYTE	B#16#00	B#16#01
49.0	Logic_Table.Logic[1].OpType2	BYTE	B#16#00	B#16#00
50.0	Logic_Table.Logic[1].LogicFct2_3	BYTE	B#16#00	B#16#00
51.0	Logic_Table.Logic[1].OpNo3	BYTE	B#16#00	B#16#01
52.0	Logic_Table.Logic[1].OpType3	BYTE	B#16#00	B#16#00
53.0	Logic_Table.Logic[1].LogicFct3_4	BYTE	B#16#00	B#16#00
54.0	Logic_Table.Logic[1].OpNo4	BYTE	B#16#00	B#16#01
55.0	Logic_Table.Logic[1].OpType4	BYTE	B#16#00	B#16#00
56.0	Logic_Table.Logic[1].OutputDelay	WORD	W#16#0000	W#16#0000
58.0	Logic_Table.Logic[1].OutputTrigger	BYTE	B#16#00	B#16#00
59.0	Logic_Table.Logic[1].Modul_No	BYTE	B#16#00	B#16#01 10

The logic function can also be programmed in the data block. The following is an excerpt from a data block with regard to the logic function described in the example.

ProgNofrom 0 to MAX_PROGBYTEDestNofrom 1 to 160 deletes complete logic functionBYTEDestType0 = hardware output 1 = flag 2 = hardware output inverted 3 = flag invertedByteByteOpNo11 - 32BYTEOpType10 = internal cam control output 1 = lnput: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType1see OpType1BYTEDopType3see CogicFct1-2BYTEOpNo31 - 32BYTEOpType3see LogicFct1-2BYTEOpType3see LogicFct1-2BYTEOpType41 - 32BYTEOpNo41 - 32BYTEOpType3see LogicFct1-2BYTEOpType3see LogicFct1-2BYTEOpType41 - 32BYTEOpType3see LogicFct1-2BYTEOpType41 - 32BYTEOpType3see LogicFct1-2BYTEOpType4bes CogicFct1-2BYTEOpType4bes CogicFct1-2BYTEOpType4bes CogicFct1-2BYTEOpType3see CogicFct1-2BYTEOpType4bes CogicFct1-2BYTEOpType4bes CogicFct1-2BYTE	Structure	Value	Function	Length
DestNofrom 1 to 160 deletes complete logic functionBYTEDestType0 = hardware output 1 = flag 2 = hardware output inverted 3 = flag invertedByteByteOpNo11 - 320 = internal cam control output 1 = lnput: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType1See OpType1BYTEDopType2see OpType1BYTEDopType3see LogicFct1-2BYTEOpType3see LogicFct1-2BYTEOpType3see LogicFct1-2BYTEOpType3see LogicFct1-2BYTEOpType41 - 32BYTEDopType3see LogicFct1-2BYTEOpType3see LogicFct1-2BYTEOpType41 - 32BYTEDopType3see LogicFct1-2BYTEDopType3see LogicFct1-2BYTEDopType41 - 32BYTEDopType4see LogicFct1-2BYTEDopType3see LogicFct1-2BYTEDopType41 - 32BYTEDopType4see LogicFct1-2BYTEDopType4see LogicFct1-2BYTEDopType4see CopType1BYTE	ProgNo	from 0 to MAX_PROG		BYTE
DestType0 = hardware output 1 = flag 2 = hardware output inverted 3 = flag invertedByteOpNo11 - 32BYTEOpType10 = internal cam control output 1 = Input: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpType2see OpType1BYTEOpType3see CopType1BYTEDeformationBYTEDeformationBYTEOpto31 - 32Opto41 -	DestNo	from 1 to 16	0 deletes complete logic function	BYTE
1 = flag 2 = hardware output inverted 3 = flag invertedBYTEOpNo11 - 32BYTEOpType10 = internal cam control output 1 = Input: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpNo31 - 32BYTEOpType3see OpType1BYTEOpType3see LogicFct1-2BYTEOpType41 - 32BYTEOpType3see LogicFct1-2BYTEOpType41 - 32BYTEOpType3see ContinedBYTEOpType41 - 32BYTEOpType41 - 32BYTEOpType3see ContinedBYTEOpType41 - 32BYTEOpType41 - 32BYTEOpType41 - 32BYTEOpType41 - 32BYTEOpType41 - 32BYTE	DestType	0 = hardware output		Byte
2 = hardware output inverted 3 = flag invertedBYTEOpNo11 - 32BYTEOpType10 = internal cam control output 1 = Input: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpType2see OpType1BYTELogicFct2-3see LogicFct1-2BYTEOpType3see OpType1BYTEDpNo41 - 32BYTEDpNo41 - 32BYTEDpNo41 - 32BYTEDpType3see LogicFct1-2BYTEDpNo41 - 32BYTEDpType3see LogicFct1-2BYTEDpType3see LogicFct1-2BYTE		1 = flag		
3 = flag invertedOpNo11 - 32BYTEOpType10 = internal cam control output 1 = Input: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpType21 - 32BYTEOpType3see OpType1BYTELogicFct3-4see LogicFct1-2BYTEOpType41 - 32BYTEOpType3see LogicFct1-2BYTEOpType41 - 32BYTEOpType41 - 32BYTEOpType5see LogicFct1-2BYTEOpType5see LogicFct1-2BYTEOpType41 - 32BYTE		2 = hardware output inverted		
OpNo11 - 32BYTEOpType10 = internal cam control output 1 = Input: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType2see OpType1BYTELogicFct2-3see CogicFct1-2BYTEOpNo31 - 32BYTEOpNo41 - 32BYTEOpType3see LogicFct1-2BYTEOpNo41 - 32BYTEOpNo41 - 32BYTEOpType3see CogicFct1-2BYTEOpNo41 - 32BYTE		3 = flag inverted		
OpType10 = internal cam control output 1 = Input: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType2see OpType1 See LogicFct1-2BYTEDyNo31 - 32BYTEOpType3see OpType1 See OpType1BYTEDopNo41 - 32BYTEDopNo41 - 32BYTEDopNo41 - 32BYTEDopType3see LogicFct1-2BYTEDopNo41 - 32BYTEDopType4See LogicFct1-2BYTEDopType3see LogicFct1-2BYTEDopType41 - 32BYTEDopType41 - 32BYTEDopType41 - 32BYTEDopType41 - 32BYTE	OpNo1	1 - 32		BYTE
1 = Input: hard-/software 2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType2see OpType1 1 - 32BYTELogicFct2-3see LogicFct1-2BYTEOpNo31 - 32BYTEDpNo41 - 32BYTEDpTuredBYTEBYTEDpNo41 - 32BYTEDpNo41 - 32BYTE	ОрТуре1	0 = internal cam control output		BYTE
2 = flag 3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType2see CopType1BYTELogicFct2-3see LogicFct1-2BYTEOpNo31 - 32BYTEOpType3see CopType1BYTELogicFct3-4see LogicFct1-2BYTEOpNo41 - 32BYTEOpNo41 - 32BYTE		1 = Input: hard-/software		
3 = SR (shift register) 4 = PB-Input (only LOCON 200)BYTELogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType2see OpType1BYTELogicFct2-3see LogicFct1-2BYTEOpNo31 - 32BYTEOpType3see CopType1BYTELogicFct3-4see LogicFct1-2BYTEOpNo41 - 32BYTEOpNo41 - 32BYTEOpNo51 - 32BYTEOpType3see CopType1BYTEDopNo41 - 32BYTEOpNo41 - 32BYTEOpTupe1BYTEOpTupe1BYTE		2 = flag		
4 = PB-Input (only LOCON 200)LogicFct1-20 = none 1 = or 2 = and 3 = or not 4 = and notOpNo21 - 32OpType2see OpType1LogicFct2-3see LogicFct1-2OpNo31 - 32OpType3see CopType1LogicFct3-4see LogicFct1-2OpNo41 - 32OpNo41 - 32OpNo41 - 32OpType3see LogicFct1-2OpNo41 - 32OpNo41 - 32OpType3see LogicFct1-2OpNo41 - 32OpType3see LogicFct1-2OpType3see LogicFct1-2OpType3see LogicFct1-2OpType3see LogicFct1-2OpType3see LogicFct1-2OpType3see LogicFct1-2OpType3see LogicFct3-4OpType3see LogicFct3-4OpType3see LogicFct3-4OpType3see LogicFct3-2OpType3see LogicFct3-2OpType3 <td></td> <td>3 = SR (shift register)</td> <td></td> <td></td>		3 = SR (shift register)		
LogicFct1-20 = noneBYTE1 = or2 = and3 = or not3 = or not4 = and notOpNo21 - 32OpType2see OpType1LogicFct2-3see LogicFct1-2OpNo31 - 32OpType3see CopType1LogicFct3-4see LogicFct1-2OpNo41 - 32OpNo41 - 32OpType1BYTEOpNo41 - 32OpType1BYTEOpNo41 - 32OpType1BYTE		4 = PB-Input (only LOCON 200)		
1 = or 2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType2see OpType1BYTELogicFct2-3see LogicFct1-2BYTEOpNo31 - 32BYTEOpType3see CopType1BYTELogicFct3-4see LogicFct1-2BYTEOpNo41 - 32BYTEOpNo41 - 32BYTE	LogicFct1-2	0 = none		BYTE
2 = and 3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType2see OpType1BYTELogicFct2-3see LogicFct1-2BYTEOpNo31 - 32BYTEOpType3see CopType1BYTELogicFct3-4see LogicFct1-2BYTEOpNo41 - 32BYTEOpNo41 - 32BYTE		1 = or		
3 = or not 4 = and notBYTEOpNo21 - 32BYTEOpType2see OpType1BYTELogicFct2-3see LogicFct1-2BYTEOpNo31 - 32BYTEOpType3see CopType1BYTELogicFct3-4see LogicFct1-2BYTEOpNo41 - 32BYTEOpNo41 - 32BYTE		2 = and		
4 = and notOpNo21 - 32BYTEOpType2see OpType1BYTELogicFct2-3see LogicFct1-2BYTEOpNo31 - 32BYTEOpType3see OpType1BYTELogicFct3-4see LogicFct1-2BYTEOpNo41 - 32BYTEOpNo41 - 32BYTE		3 = or not		
OpNo2 1 - 32 BYTE OpType2 see OpType1 BYTE LogicFct2-3 see LogicFct1-2 BYTE OpNo3 1 - 32 BYTE OpType3 see OpType1 BYTE LogicFct3-4 see LogicFct1-2 BYTE OpNo4 1 - 32 BYTE OpType3 see LogicFct1-2 BYTE		4 = and not		
OpType2 see OpType1 BYTE LogicFct2-3 see LogicFct1-2 BYTE OpNo3 1 - 32 BYTE OpType3 see OpType1 BYTE LogicFct3-4 see LogicFct1-2 BYTE OpNo4 1 - 32 BYTE OpType3 see LogicFct1-2 BYTE	OpNo2	1 - 32		BYTE
LogicFct2-3 see LogicFct1-2 BYTE OpNo3 1 - 32 BYTE OpType3 see OpType1 BYTE LogicFct3-4 see LogicFct1-2 BYTE OpNo4 1 - 32 BYTE	ОрТуре2	see OpType1		BYTE
OpNo3 1 - 32 BYTE OpType3 see OpType1 BYTE LogicFct3-4 see LogicFct1-2 BYTE OpNo4 1 - 32 BYTE	LogicFct2-3	see LogicFct1-2		BYTE
OpType3 see OpType1 BYTE LogicFct3-4 see LogicFct1-2 BYTE OpNo4 1 - 32 BYTE	OpNo3	1 - 32		BYTE
LogicFct3-4 see LogicFct1-2 BYTE OpNo4 1 - 32 BYTE	ОрТуре3	see OpType1		BYTE
OpNo4 1 - 32 BYTE	LogicFct3-4	see LogicFct1-2		BYTE
	OpNo4	1 - 32		BYTE
Op Type4 [see Op Type1] BYTE	ОрТуре4	see OpType1		BYTE
OutputDelay ms at present max. 255 WORD	OutputDelay	ms	at present max. 255	WORD
OutputTrigger 0 = leading edge BYTE	OutputTrigger	0 = leading edge		BYTE
1 = trailing edge		1 = trailing edge		
Module number 0 basis BYTE	Module number	0 basis		BYTE
(only LOCON 200) x I/O-module number	(only LOCON 200)	x I/O-module number		

A description of the parameters can be found in this table.