

**LOCON 100-MB**

**LOCON 200-MB**

**Communication with ModScan32 (Modbus Master Simulator)**

## Inhalt

1. LOCON 100/200-MB.....	2
2. ModScan32 .....	4
3. Programming cams .....	15
4. Idle time compensation (ITC).....	21
5. Error quit.....	25
6. Teach-In Zero .....	28
7. Changing active program .....	29
8. Programming in program.....	30
9. Addresses .....	31

## 1. LOCON 100/200-MB

### Short overview LOCON 100/200-MB

LOCON 100/200-MB example settings:

Interface RS232-Modbus: Rotary coding switch „High“ = **C**  
Modbus-Slave-ID: Rotary coding switch „Low“ = **1**

**Note:** Only the Modbus-addresses 1... 15 are supported. The broadcast address “0” is not supported.

LOCON 100/200-MB supported function codes:

**FC3** (read holding register)

**FC16** (preset multiple registers)

LOCON 100/200-MB serial settings (not changeable): Baud rate = **19200**

Parity = **no parity**

LOCON 100/200-MB configuration (default):

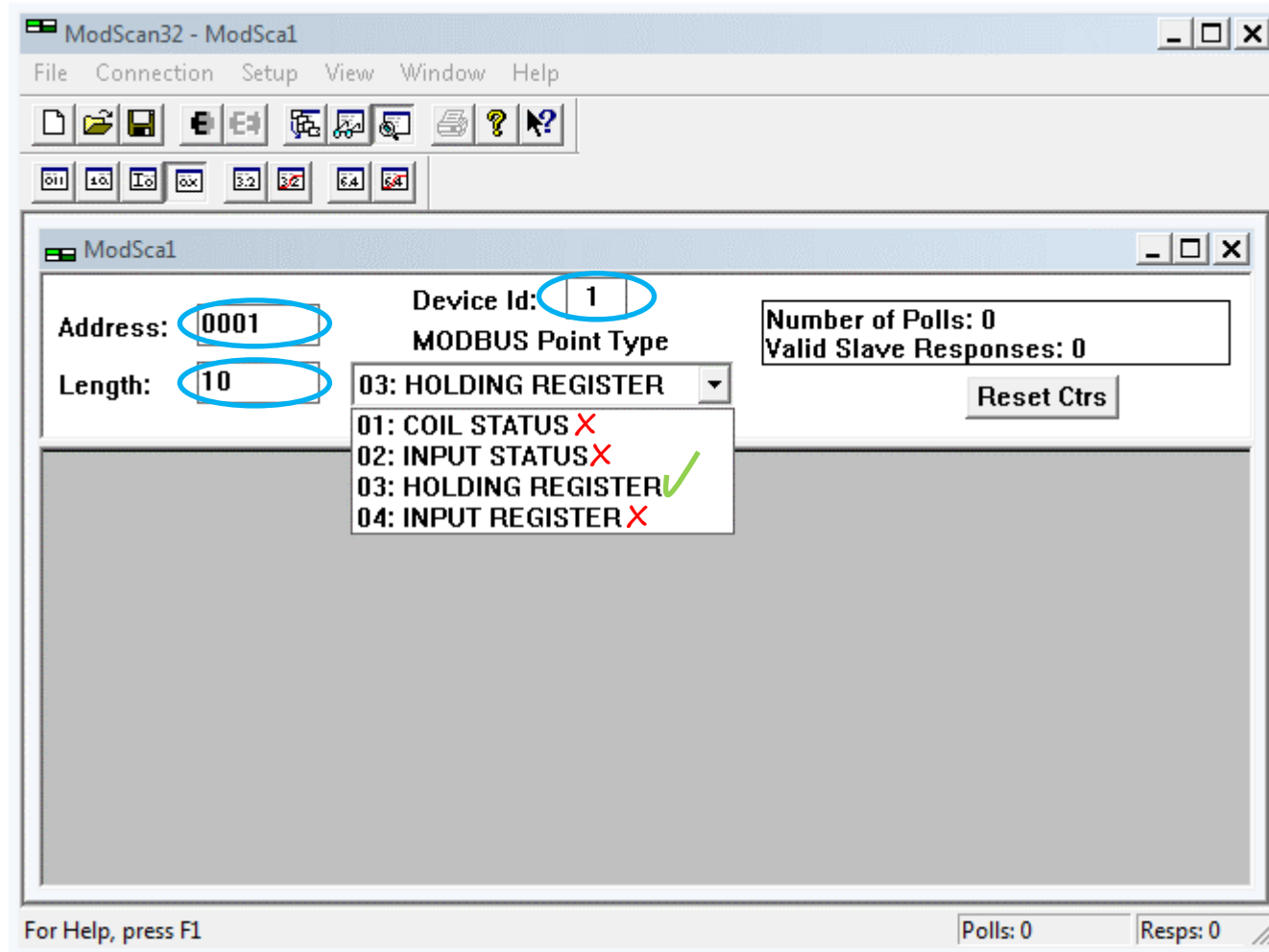
The virtual encoder resolution is **4096** (default). Cams can be programmed from **0** to **4095**.

The DTC (dead time compensation) is **bitwise** (default).

Encoder type	SSI singleturn
Encoder resolution	4096
Virtual encoder resolution	4096
Turn direction	clockwise
Encoder control	no
Zero shift	0
ITC Type	bitwise
ITC Function	standard
Interface Operationmode	RS232
DataInRamOnly	yes
Timebase DTC (µs)	100 µs
Timebase angle-time-cams (µs)	1000
Display factor	60
Option-X	0
Output Enable	not mapped
Program Enable	not mapped
Program Start	not mapped
Program 1	not mapped
Program 2	not mapped
Program 4	not mapped
Program 8	not mapped
Program 16	not mapped
Program 32	not mapped
Run Control Output	not mapped
Run-Control-Type	static
Dyn. zero shift mapping	not mapped

## 2. ModScan32

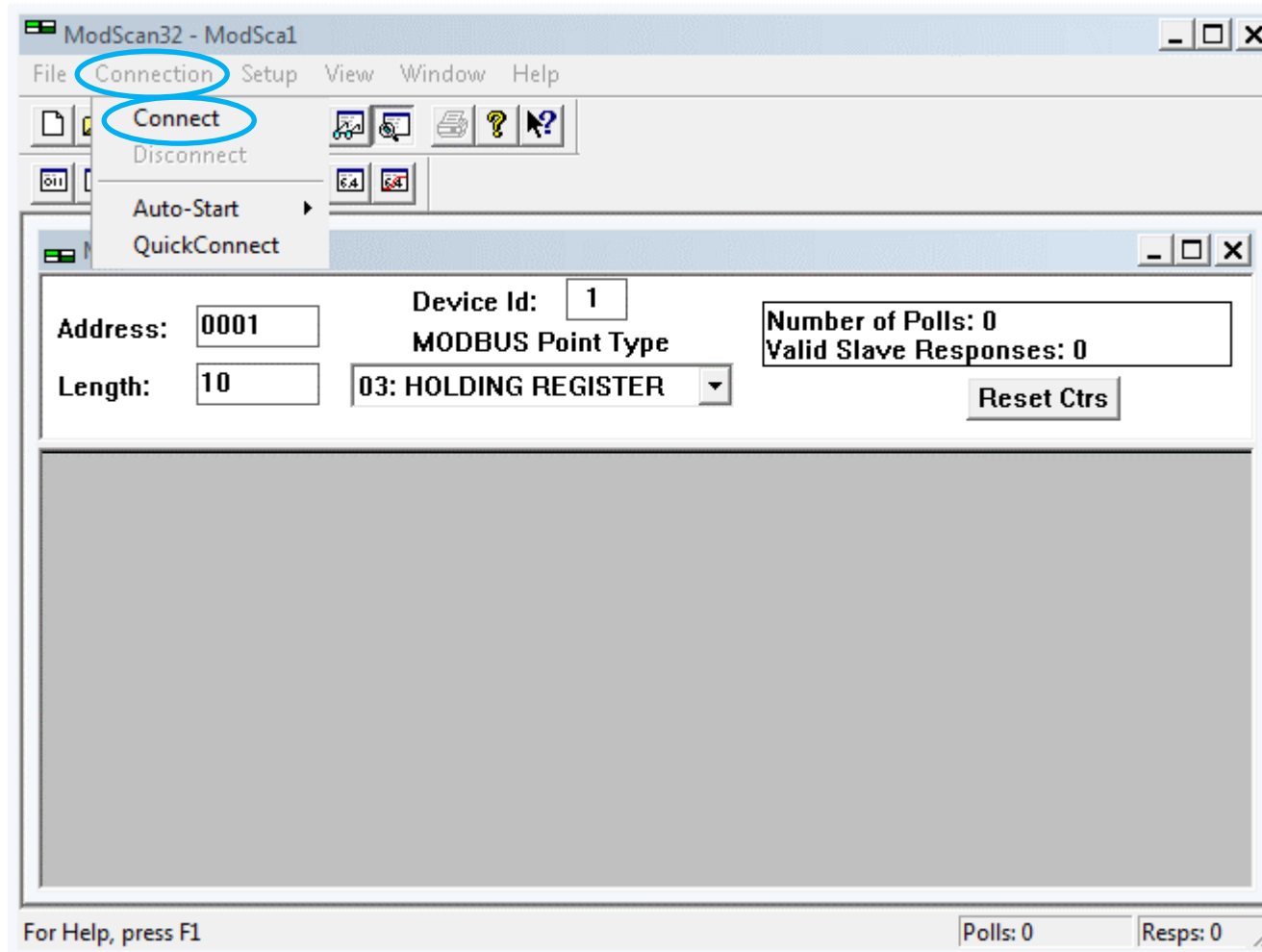
### ModScan32 setting



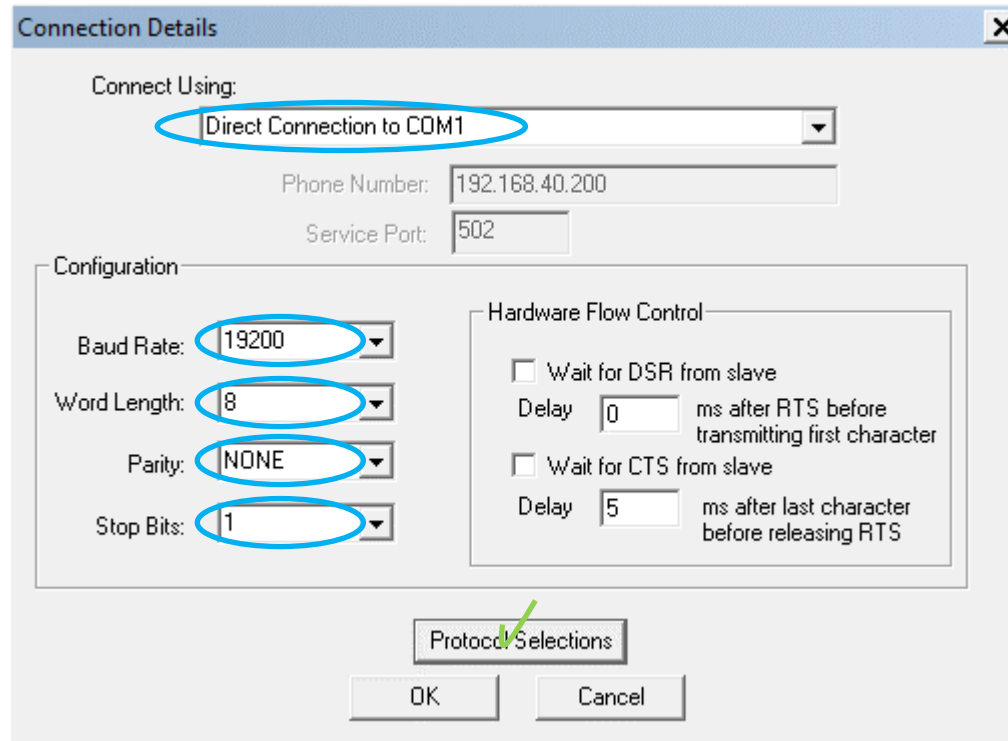
✓ Supported from LOCON 100/200-MB

✗ Not supported from LOCON 100/200-MB

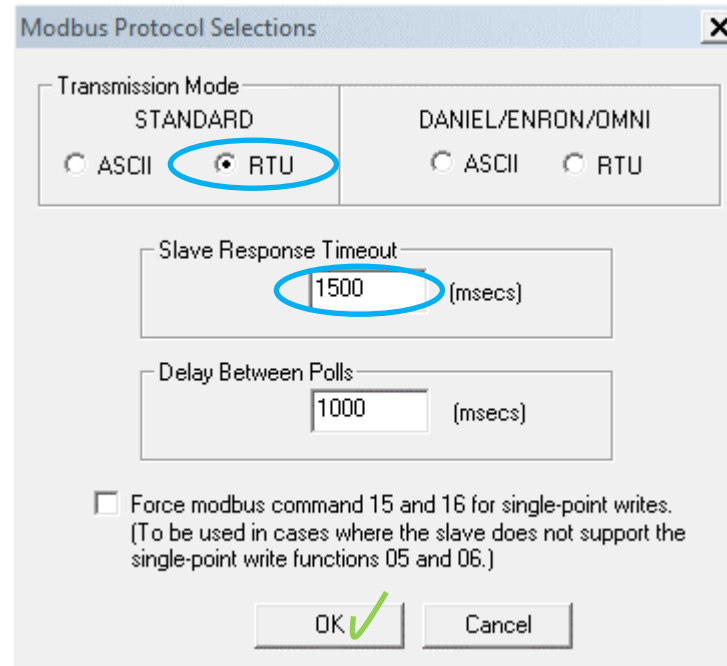
ModScan32 connection



Connect Using: Select the COM port, which you would like to use.

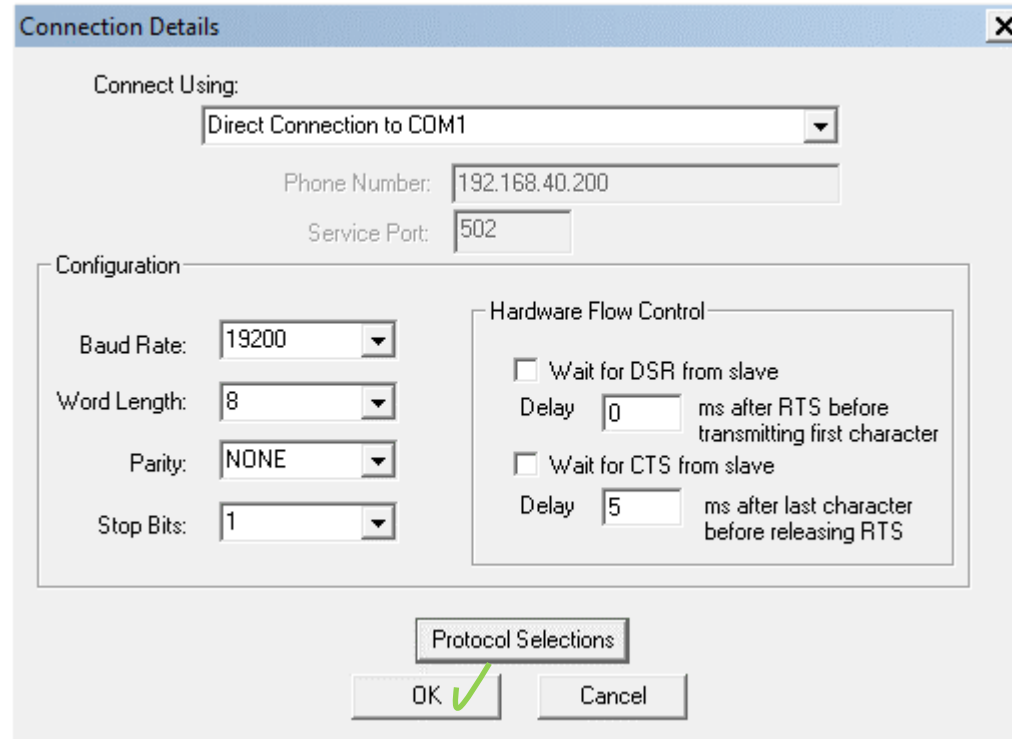


The Modbus protocol settings will be approved via the button “OK”.





The connection settings will be approved via the button “OK”.



Connection Details

Connect Using:  
 Direct Connection to COM1

Phone Number: 192.168.40.200  
 Service Port: 502

Configuration

Baud Rate: 19200  
 Word Length: 8  
 Parity: NONE  
 Stop Bits: 1

Hardware Flow Control

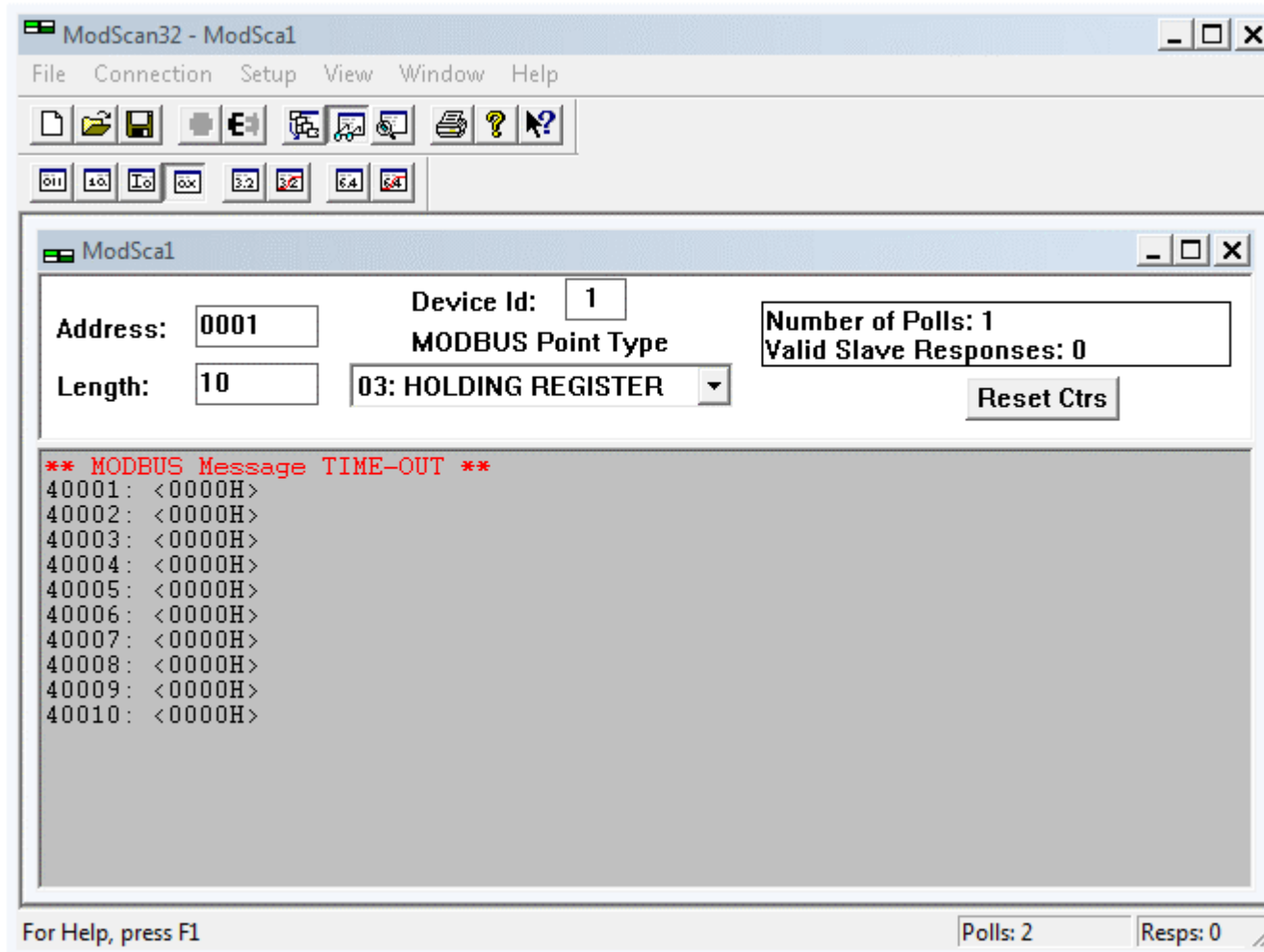
Wait for DSR from slave  
 Delay 0 ms after RTS before transmitting first character

Wait for CTS from slave  
 Delay 5 ms after last character before releasing RTS

Protocol Selections

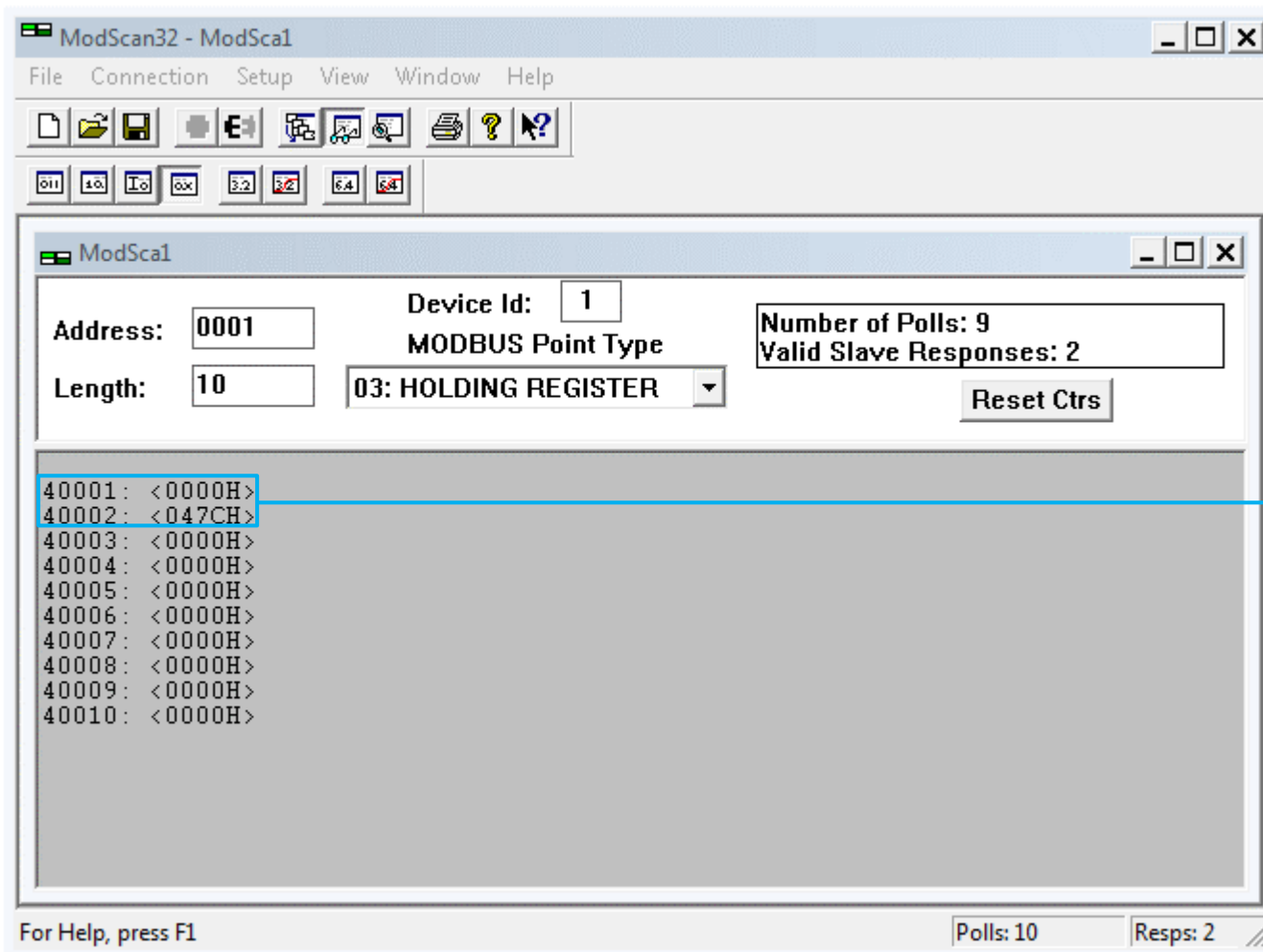
OK ✓ Cancel

ModScan32 (Modbus Master Simulator) has started the communication. ModScan32 sends requests cyclic. In this case requests with function code 3 (read holding register), but the Modbus Slave (LOCON 100/200-MB) hasn't send a response to ModScan32.



LOCON 100/200-MB switch ON. ModScan32 (Modbus Master Simulator) gets response from the Modbus Slave (LOCON 100/200-MB).

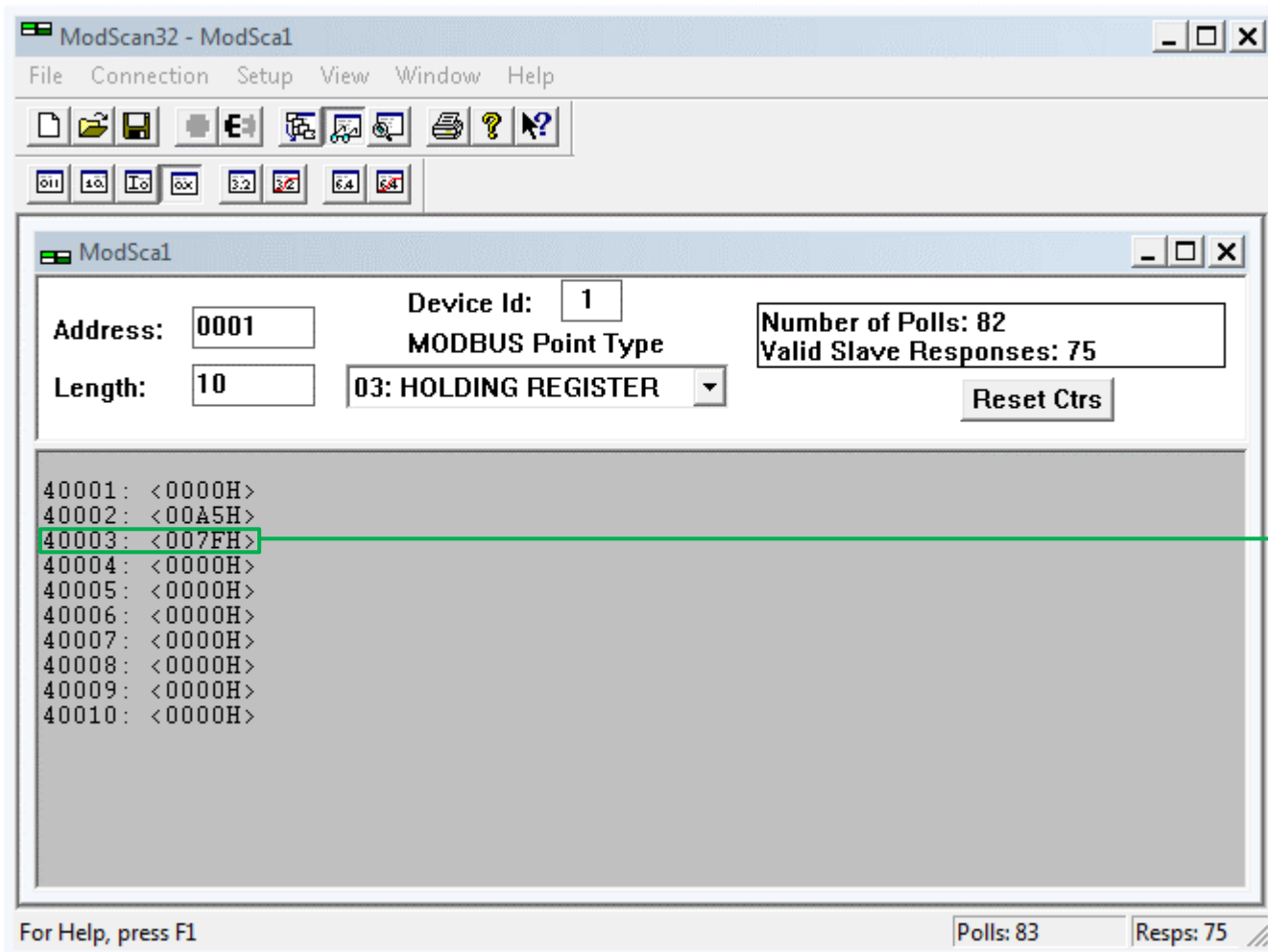
In this example, the encoder position is **47C** (hex) -> **1148** (dec).



Process data (read only):

4_0001..	4_0002	Position
	4_0003	Speed
	4_0004	Actual program
	4_0005	Error number
4_0006..	4_000D	Output state 1..128

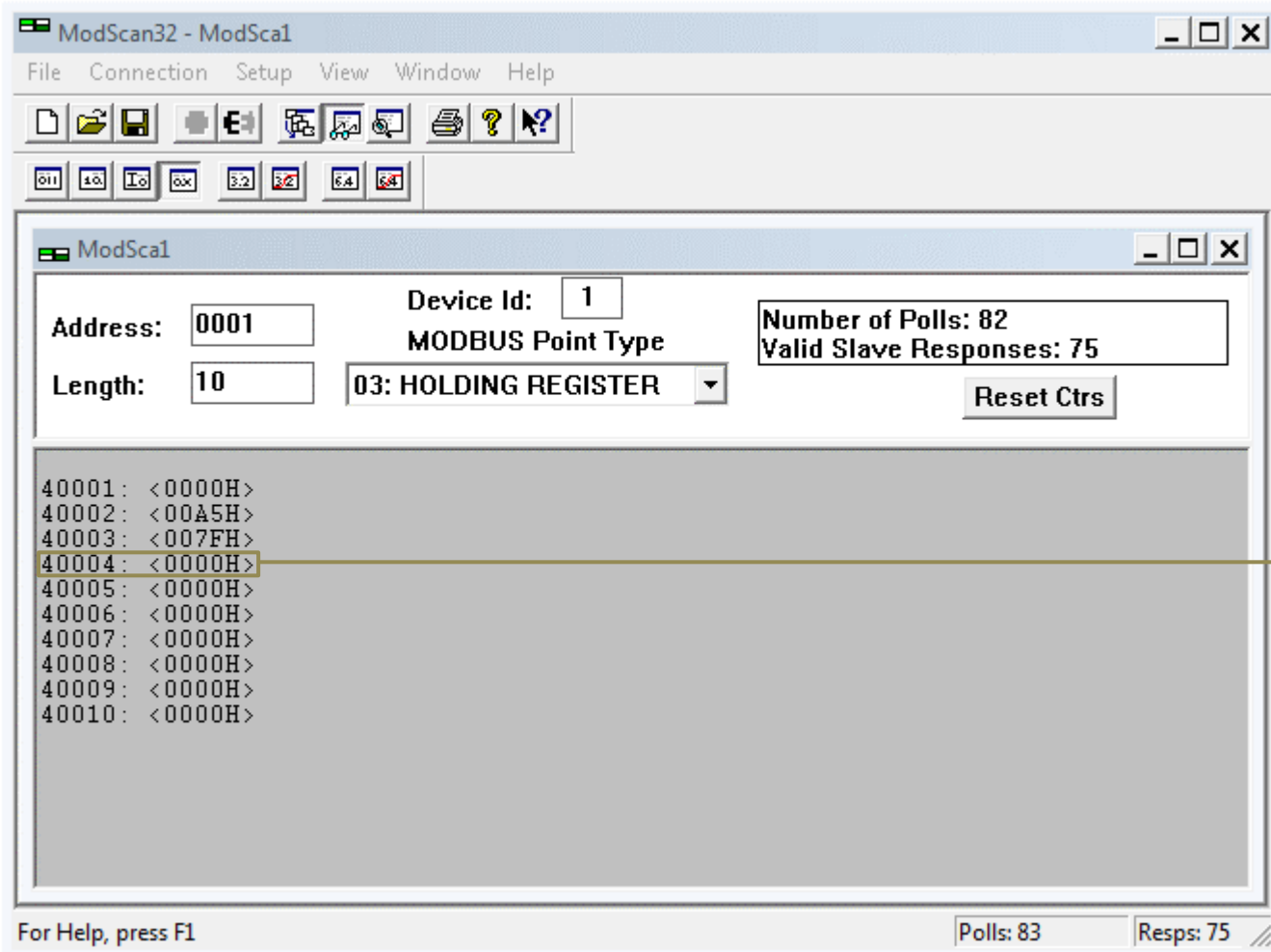
This example shows the encoder speed 7F (hex) -> 127 (dec).



Process data (read only):

4_0001.. 4_0002	Position
4_0003	Speed
4_0004	Actual program
4_0005	Error number
4_0006.. 4_000D	Output state 1..128

This example shows the actual program 00 (hex) -> 00 (dec).

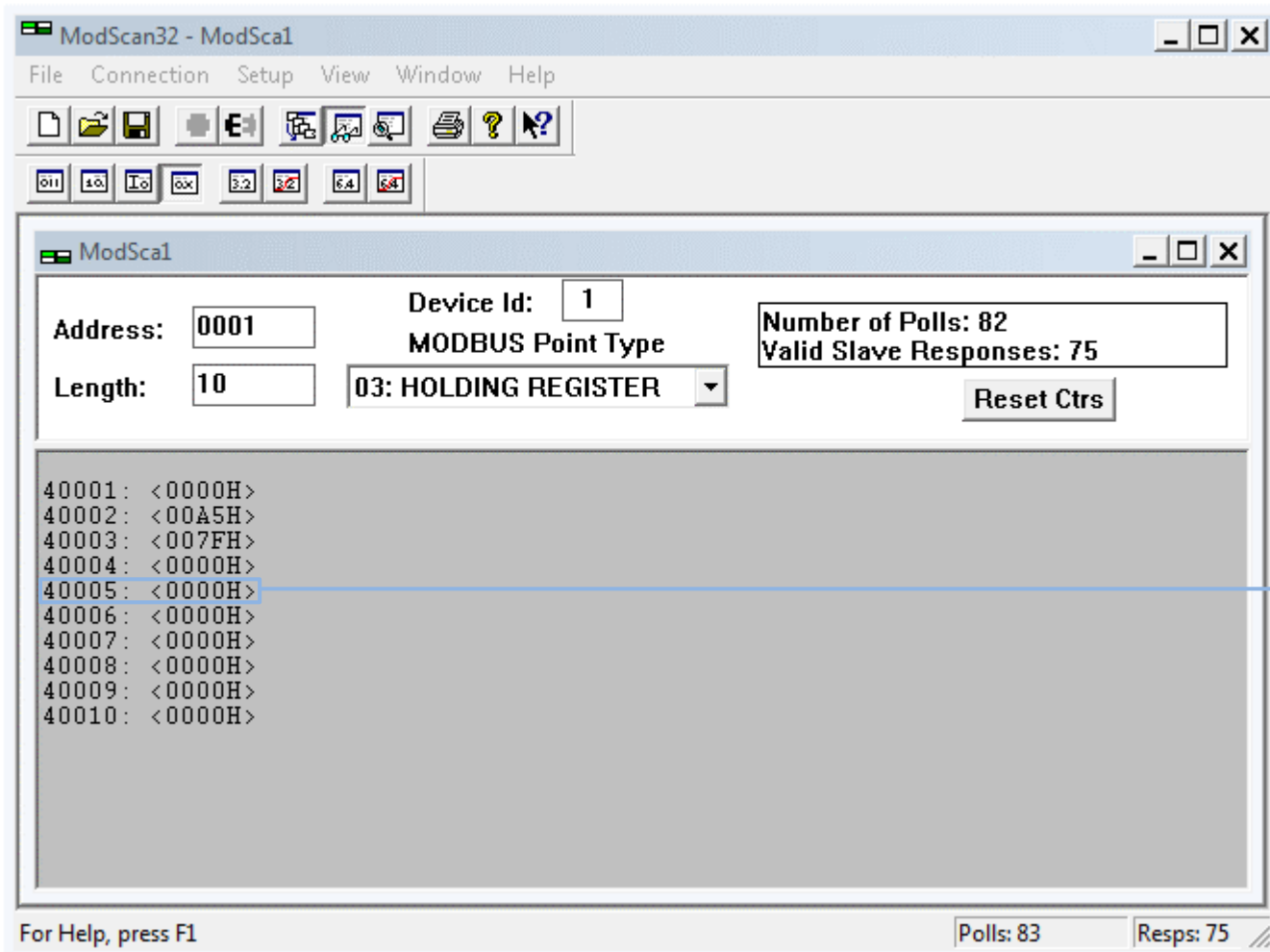


Process data (read only):

---

4_0001.. 4_0002	Position
4_0003	Speed
4_0004	Actual program
4_0005	Error number
4_0006.. 4_000D	Output state 1..128

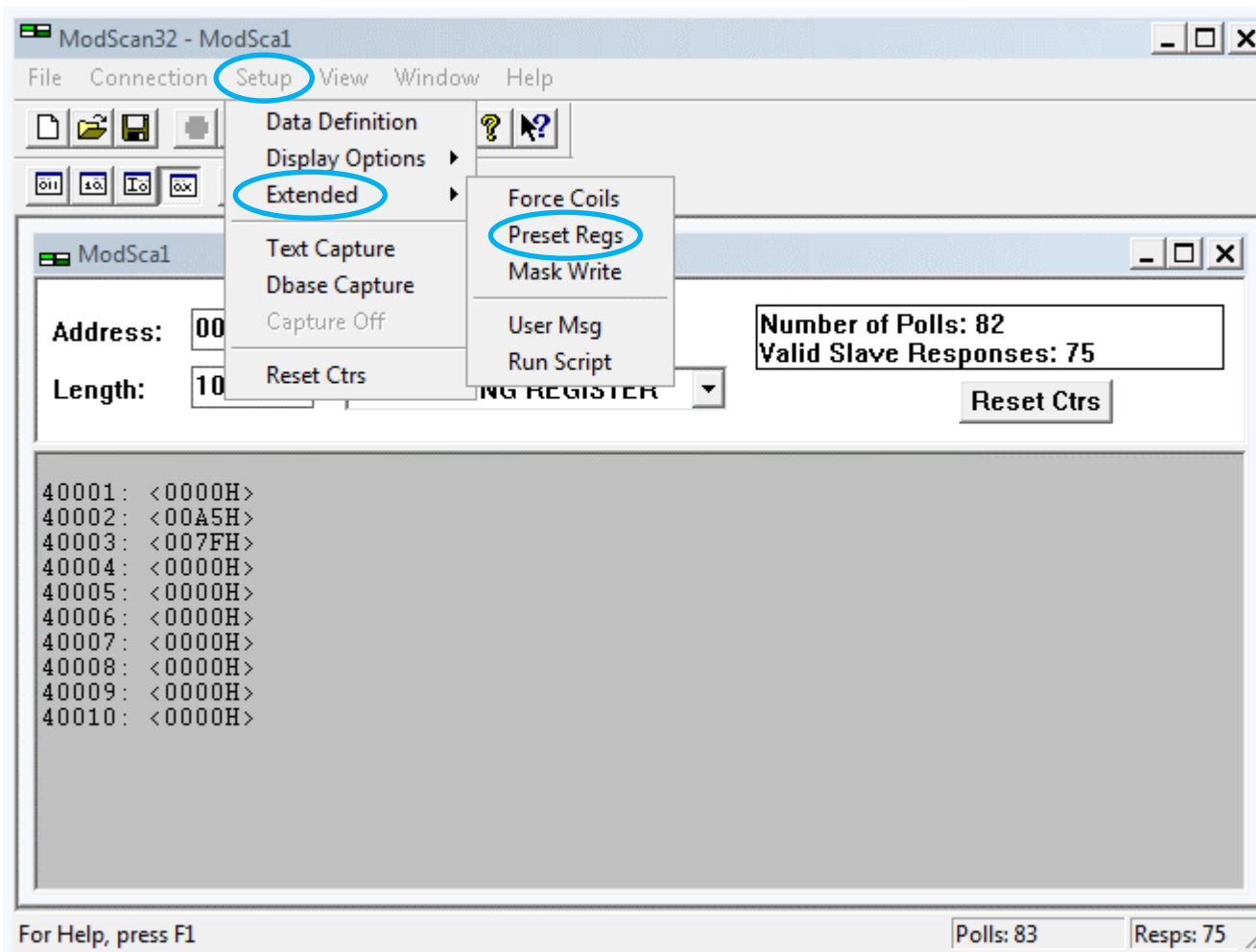
This example shows the error number 00 (hex) -> 00 (dec).



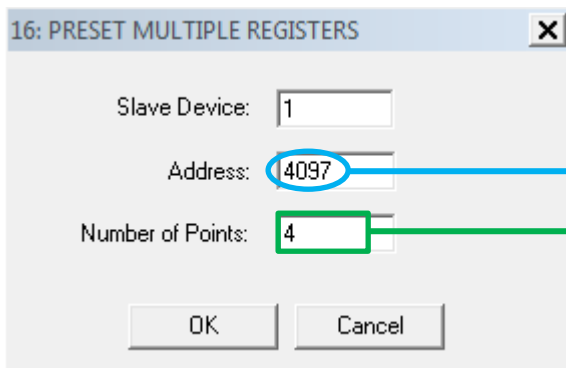
Process data (read only):

- 4\_0001.. 4\_0002 Position
- 4\_0003 Speed
- 4\_0004 Actual program
- 4\_0005 Error number
- 4\_0006.. 4\_000D Output state 1..128

The LOCON 100/200-MB supports the function code 16 (preset multiple registers). Cams, DTC and ERROR quit can be set with this function code.



### 3. Programming cams



Cams (read-write):

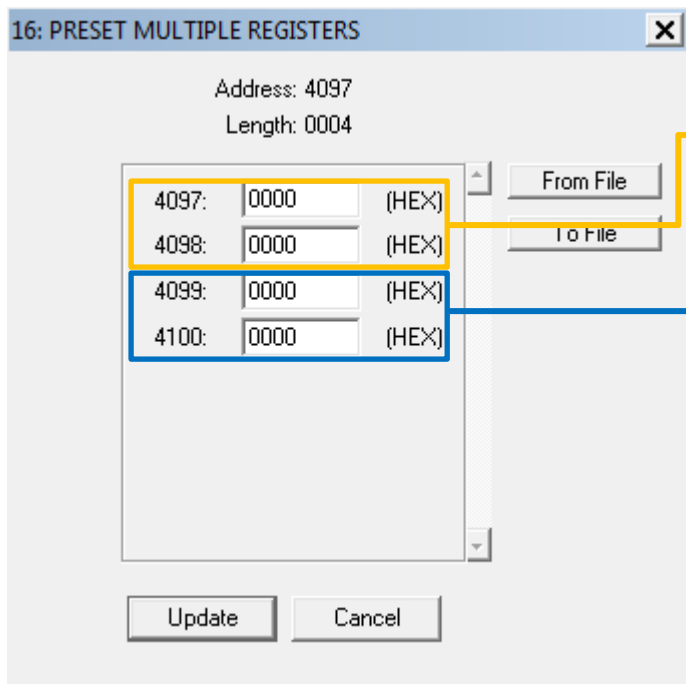
4_1001..4_1002	Cam 1 switch on output1
4_1003..4_1004	Cam 1 switch off output1
4_1005..4_1006	Cam 2 switch on output1
4_1007..4_1008	Cam 2 switch off output1
...	
4_107D..4_107E	Cam 32 switch on output1
4_107F..4_1080	Cam 32 switch off output1
...	
4_1081..4_1082	Cam 1 switch on output2
4_1083..4_1084	Cam 1 switch off output2
...	
4_10FD..4_10FE	Cam 32 switch on output2
4_10FF..4_110	Cam 32 switch off output2
...	
...	
4_4F01..4_4F02	Cam 1 switch on output128
4_4F03..4_4F04	Cam 1 switch off output128
...	
4_4FFD..4_4FFE	Cam 32 switch on output128
4_4FFF..4_5000	Cam 32 switch off output128

$$\text{Modbus-start-address from cam } c \text{ on output } o = 4\_1001H + ((o-1) * 80H) + ((c-1) * 4)$$



**ATTENTION:**  
A maximum of 32 cams each output can be programmed!





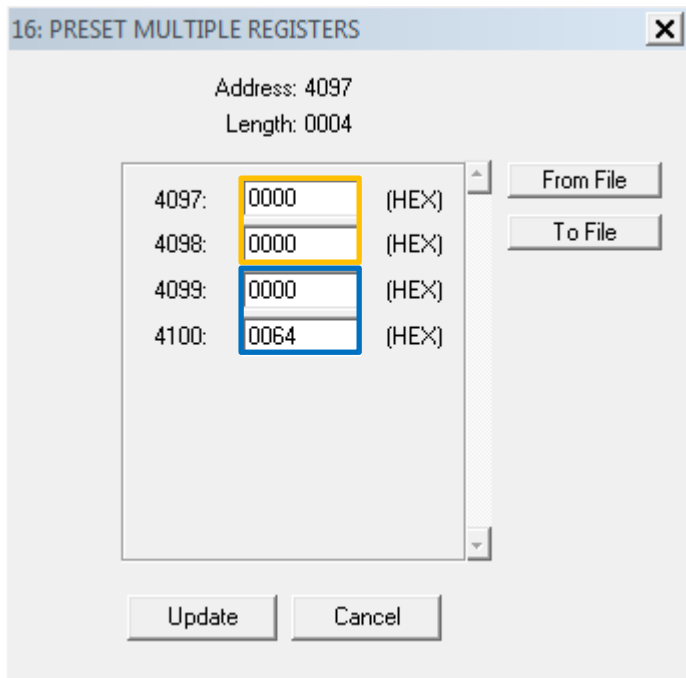
Cams (read-write):

4_1001..4_1002	Cam 1 switch on output1
4_1003..4_1004	Cam 1 switch off output1
4_1005..4_1006	Cam 2 switch on output1
4_1007..4_1008	Cam 2 switch off output1
...	
4_107D..4_107E	Cam 32 switch on output1
4_107F..4_1080	Cam 32 switch off output1
...	
4_1081..4_1082	Cam 1 switch on output2
4_1083..4_1084	Cam 1 switch off output2
...	
4_10FD..4_10FE	Cam 32 switch on output2
4_10FF..4_1100	Cam 32 switch off output2
...	
...	
4_4F01..4_4F02	Cam 1 switch on output128
4_4F03..4_4F04	Cam 1 switch off output128
...	
4_4FFD..4_4FFE	Cam 32 switch on output128
4_4FFF..4_5000	Cam 32 switch off output128

Modbus-start-address from cam c on output o =  $4\_1001H + ((o-1) * 80H) + ((c-1) * 4)$



**ATTENTION:**  
A maximum of 32 cams each output can be programmed!



Cams (read-write):

-----	
4_1001..4_1002	Cam 1 switch on output1
4_1003..4_1004	Cam 1 switch off output1
4_1005..4_1006	Cam 2 switch on output1
4_1007..4_100	Cam 2 switch off output1
...	
4_107D..4_107E	Cam 32 switch on output1
4_107F..4_1080	Cam 32 switch off output1
...	
4_1081..4_1082	Cam 1 switch on output2
4_1083..4_1084	Cam 1 switch off output2
...	
4_10FD..4_10FE	Cam 32 switch on output2
4_10FF..4_110	Cam 32 switch off output2
...	
...	
4_4F01..4_4F0	Cam 1 switch on output128
4_4F03..4_4F04	Cam 1 switch off output128
...	
4_4FFD..4_4FFE	Cam 32 switch on output128
4_4FFF..4_5000	Cam 32 switch off output128

Example cam 1, output 1:

Cam 1 switch on output 1 = 0 (dec) -> 0 (hex)

Cam 1 switch off output 1 = 100 (dec) -> 64 (hex)

$$\text{Modbus-start-address from cam } c \text{ on output } o = 4\_1001H + ((o-1) * 80H) + ((c-1) * 4)$$

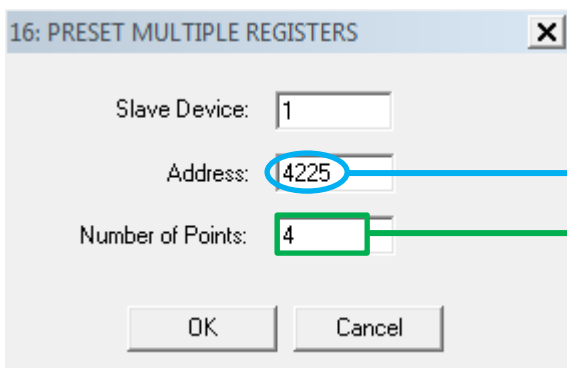


**ATTENTION:**  
A maximum of 32 cams each output can be programmed!

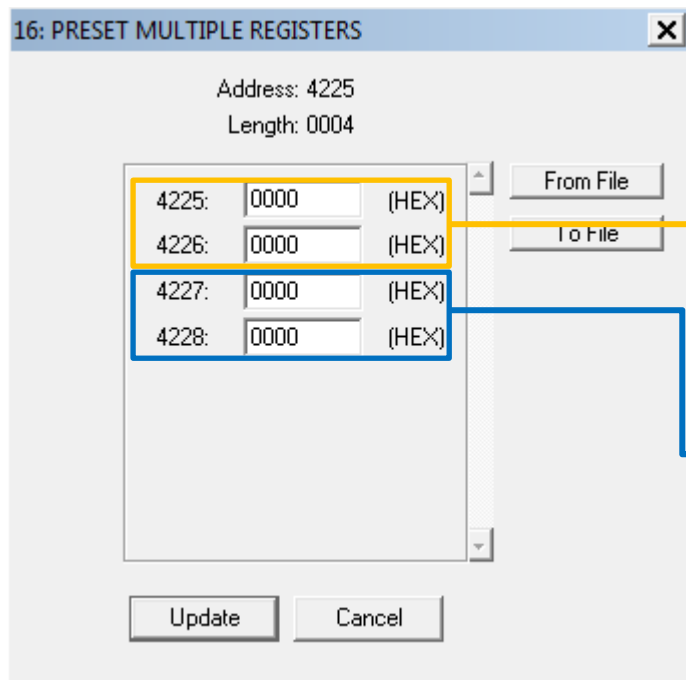
Cams (read-write):

4_1001..4_1002	Cam 1 switch on output1
4_1003..4_1004	Cam 1 switch off output1
4_1005..4_1006	Cam 2 switch on output1
4_1007..4_100	Cam 2 switch off output1
...	
4_107D..4_107E	Cam 32 switch on output1
4_107F..4_1080	Cam 32 switch off output1
...	
4_1081..4_1082	Cam 1 switch on output2
4_1083..4_1084	Cam 1 switch off output2
...	
4_10FD..4_10FE	Cam 32 switch on output2
4_10FF..4_110	Cam 32 switch off output2
...	
...	
4_4F01..4_4F0	Cam 1 switch on output128
4_4F03..4_4F04	Cam 1 switch off output128
...	
4_4FFD..4_4FFE	Cam 32 switch on output128
4_4FFF..4_5000	Cam 32 switch off output128

Modbus-start-address from cam c on output o =  $4\_1001H + ((o-1) * 80H) + ((c-1) * 4)$



**ATTENTION:**  
A maximum of 32 cams each output can be programmed!



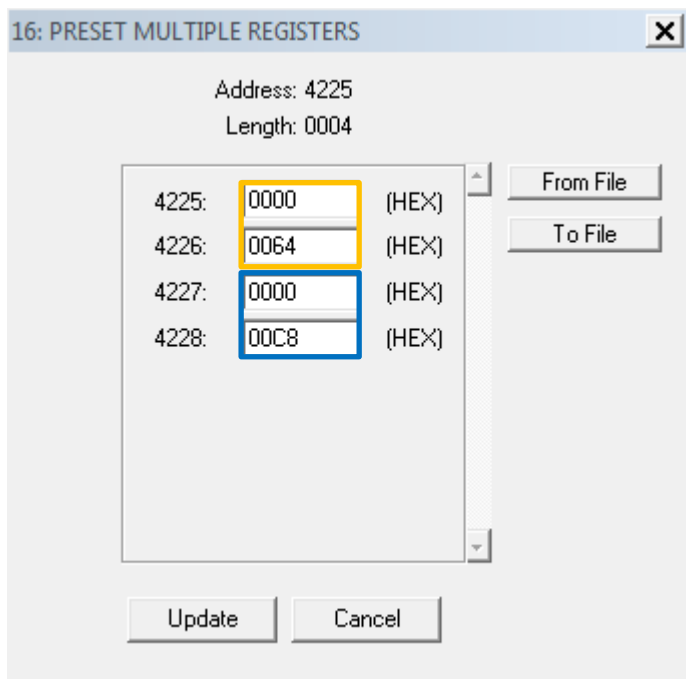
Cams (read-write):

4_1001..4_1002	Cam 1 switch on output1
4_1003..4_1004	Cam 1 switch off output1
4_1005..4_1006	Cam 2 switch on output1
4_1007..4_100	Cam 2 switch off output1
...	
4_107D..4_107E	Cam 32 switch on output1
4_107F..4_1080	Cam 32 switch off output1
...	
4_1081..4_1082	Cam 1 switch on output2
4_1083..4_1084	Cam 1 switch off output2
...	
4_10FD..4_10FE	Cam 32 switch on output2
4_10FF..4_110	Cam 32 switch off output2
...	
...	
4_4F01..4_4F0	Cam 1 switch on output128
4_4F03..4_4F04	Cam 1 switch off output128
...	
4_4FFD..4_4FFE	Cam 32 switch on output128
4_4FFF..4_5000	Cam 32 switch off output128

Modbus-start-address from cam c on output o =  $4\_1001H + ((o-1) * 80H) + ((c-1) * 4)$



**ATTENTION:**  
A maximum of 32 cams each output can be programmed!



Cams (read-write):

-----	
4_1001..4_1002	Cam 1 switch on output1
4_1003..4_1004	Cam 1 switch off output1
4_1005..4_1006	Cam 2 switch on output1
4_1007..4_100	Cam 2 switch off output1
...	
4_107D..4_107E	Cam 32 switch on output1
4_107F..4_1080	Cam 32 switch off output1
...	
4_1081..4_1082	Cam 1 switch on output2
4_1083..4_1084	Cam 1 switch off output2
...	
4_10FD..4_10FE	Cam 32 switch on output2
4_10FF..4_110	Cam 32 switch off output2
...	
...	
4_4F01..4_4F0	Cam 1 switch on output128
4_4F03..4_4F04	Cam 1 switch off output128
...	
4_4FFD..4_4FFE	Cam 32 switch on output128
4_4FFF..4_5000	Cam 32 switch off output128

Example cam 1, output 2:

Cam 1 switch on output 1 = **100** (dec) -> **64** (hex)

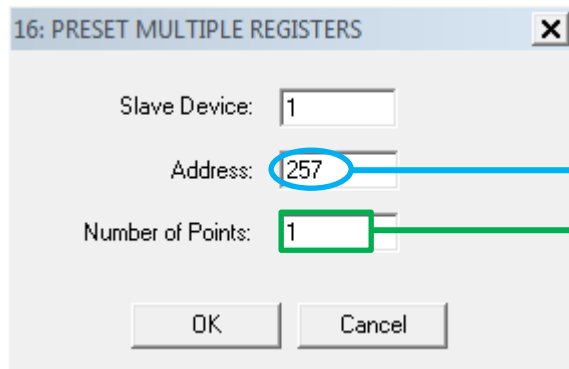
Cam 1 switch off output 1 = **200** (dec) -> **C8** (hex)

$$\text{Modbus-start-address from cam } c \text{ on output } o = 4\_1001H + ((o-1) * 80H) + ((c-1) * 4)$$



**ATTENTION:**  
A maximum of 32 cams each output can be programmed!

#### 4. Idle time compensation (ITC)

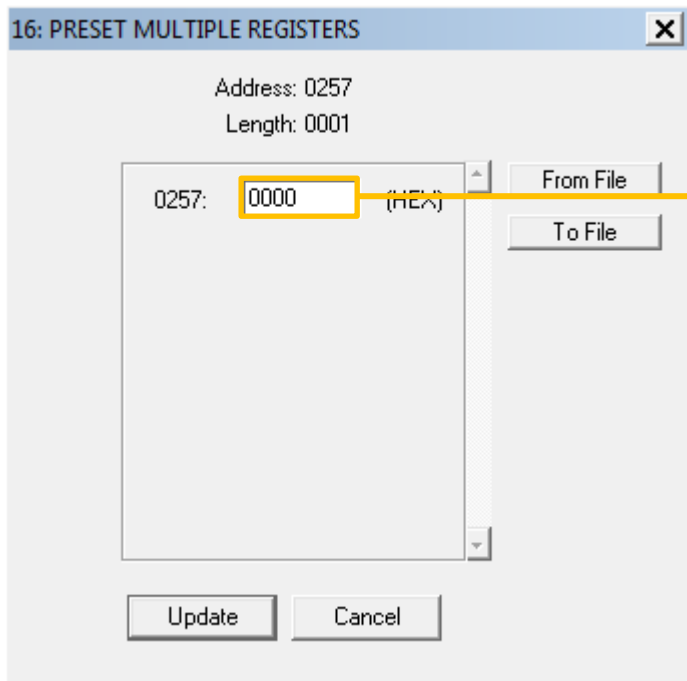


DTC (read-write):

---

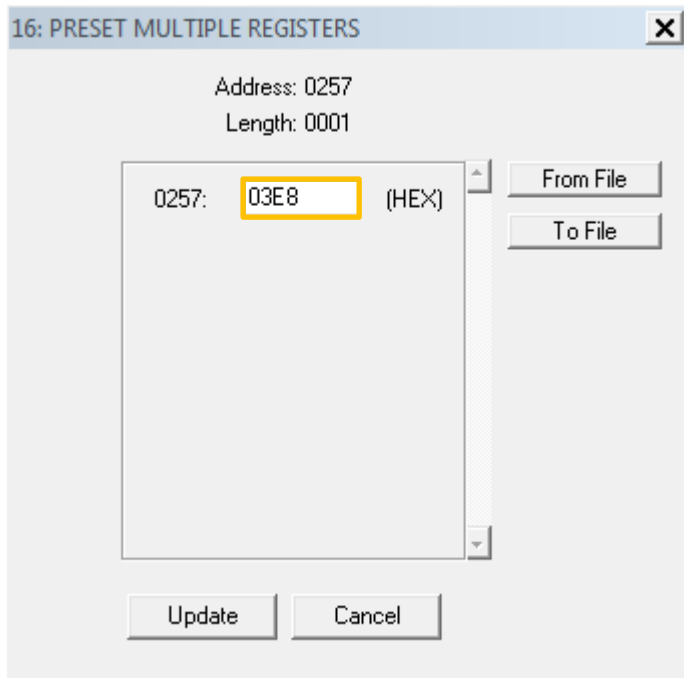
4_0101	DTC on output 1
4_0102	DTC off output 1
4_0103	DTC on output 2
4_0104	DTC off output 2
...	
...	
4_01FF	DTC on output 128
4_0200	DTC off output 128

**Note:** ITC = DTC



DTC (read-write):

4_0101	DTC on output 1
4_0102	DTC off output 1
4_0103	DTC on output 2
4_0104	DTC off output 2
...	
...	
4_01FF	DTC on output 128
4_0200	DTC off output 128



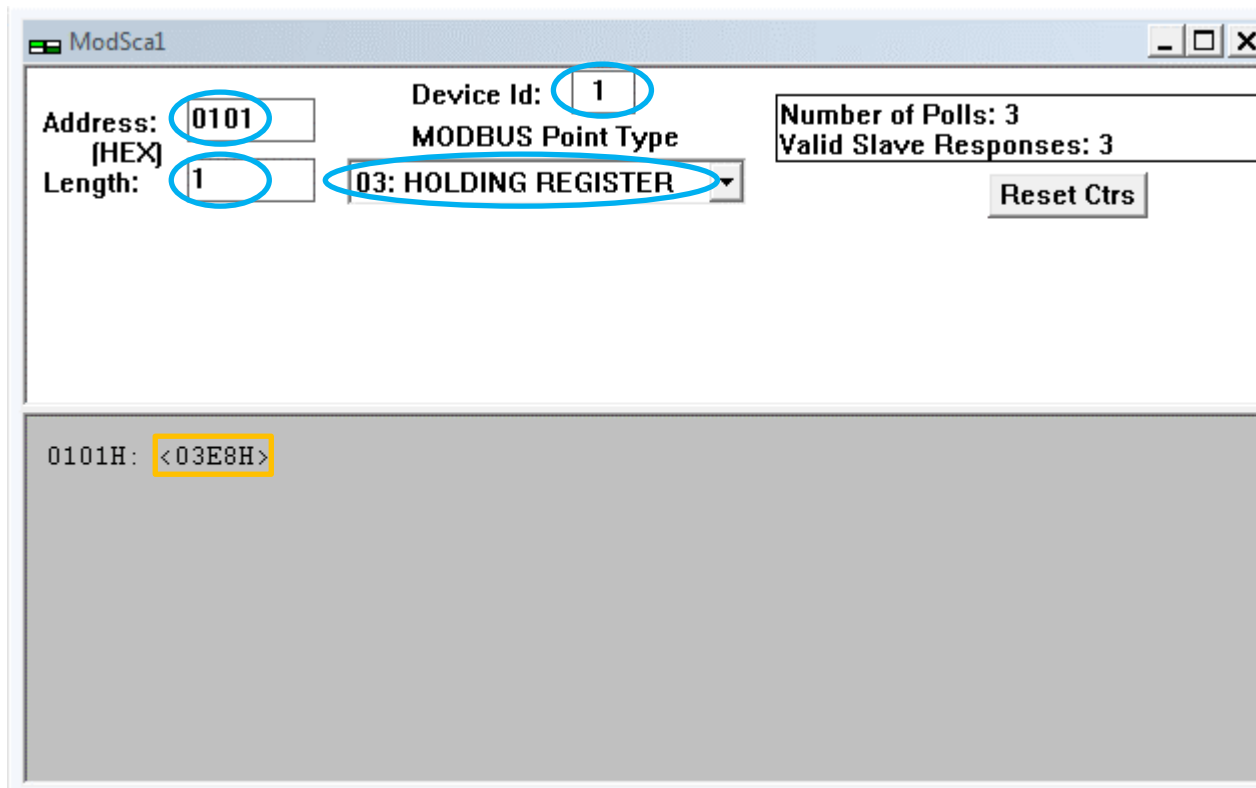
DTC (read-write):

-----	
4_0101	DTC on output 1
4_0102	DTC off output 1
4_0103	DTC on output 2
4_0104	DTC off output 2
...	
...	
4_01FF	DTC on output 128
4_0200	DTC off output 128

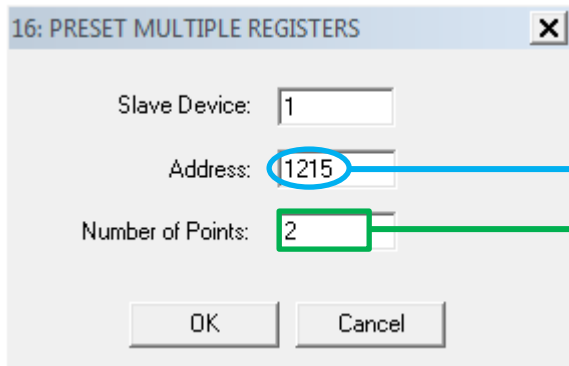
Example ITC at output 1:

ITC on = 100,0 ms (dec) -> 3E8 (hex)





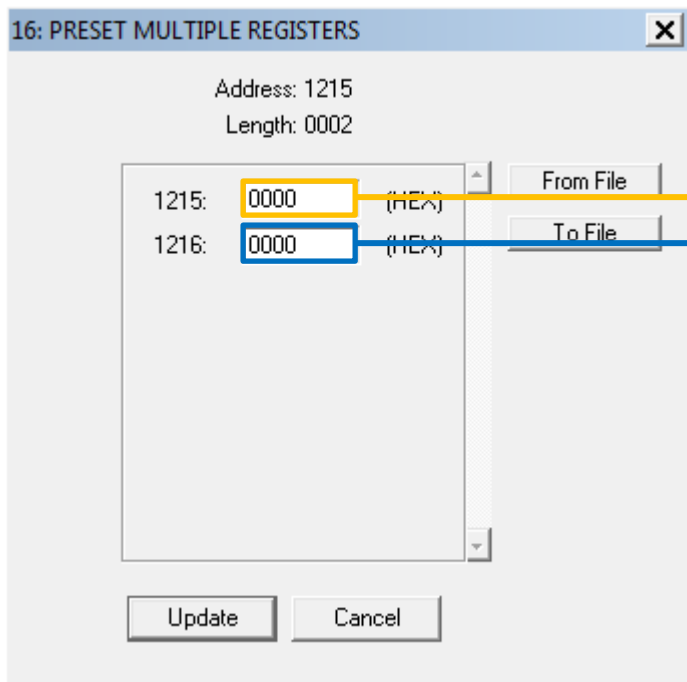
### 5. Error quit



-----  
 Parameter - Example: Acknowledging a pending error through Modbus.

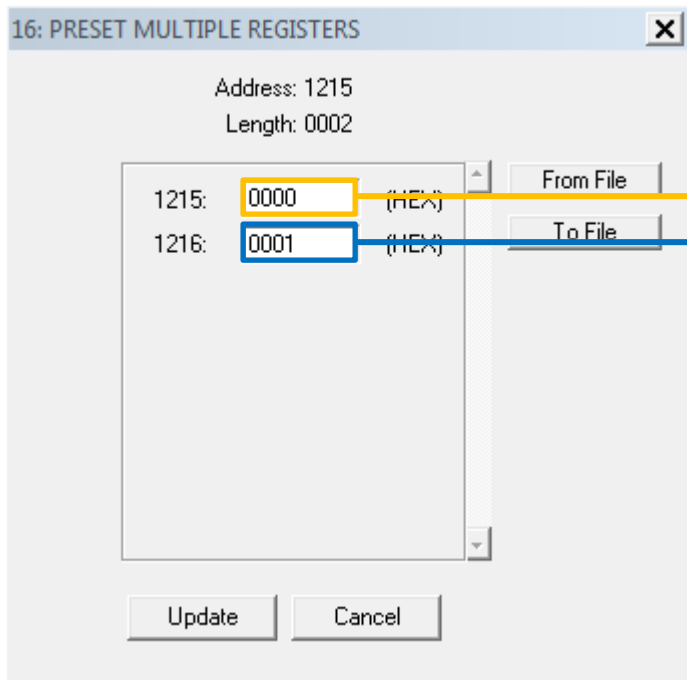
Transmission:	Receipt:
01 Slave address	01 Slave address
10 Function	10 Function
04 Starting address H	04 Starting address H
BE Starting address L	BE Starting address L
00 Length in words H	00 Length in words H
02 Length in words L	02 Length in words L
04 Length in bytes	xx CRC
00 Parameter H 1. byte	
00 Parameter H 2. byte	
00 Parameter L 1. byte	
01 Parameter L 2. byte	
xx CRC	

-----



Parameter - Example: Acknowledging a pending error through Modbus.

Transmission:	Receipt:
01 Slave address	01 Slave address
10 Function	10 Function
04 Starting address H	04 Starting address H
BE Starting address L	BE Starting address L
00 Length in words H	00 Length in words H
02 Length in words L	02 Length in words L
04 Length in bytes	xx CRC
00 Parameter H 1. byte	
00 Parameter H 2. byte	
00 Parameter L 1. byte	
01 Parameter L 2. byte	
xx CRC	



Parameter - Example: Acknowledging a pending error through Modbus.

Transmission:	Receipt:
01 Slave address	01 Slave address
10 Function	10 Function
04 Starting address H	04 Starting address H
BE Starting address L	BE Starting address L
00 Length in words H	00 Length in words H
02 Length in words L	02 Length in words L
04 Length in bytes	xx CRC
00 Parameter H 1. byte	
00 Parameter H 2. byte	
00 Parameter L 1. byte	
01 Parameter L 2. byte	
xx CRC	

### 6. Teach-In Zero

16: PRESET MULTIPLE REGISTERS

Slave Device:

Address:

Number of Points:

OK Cancel

Parameter in accordance with parameter table in the instruction manual (read-write):

4\_0401..4\_0402      Parameter 1  
 .....  
 4\_04B3..4\_04B4      Parameter 90  
 .....  
 4\_05FF..4\_0600      Parameter 256

! Parameter 256 = ModbusProgNr (default = 0) !  
 Parameter address = 0401H+((p-1)\*2)

16: PRESET MULTIPLE REGISTERS

Address: 1203  
 Length: 0002

1203:  (HEX)

1204:  (HEX)

From File  
 To File

Update Cancel

Command's name	Command's value	Parameter number in config-menu of TERM 6	Meaning	Explanation
PNR_SOFT_REV	0x0001	1	s_PNR_HARD_REV	
PNR_HARD_REV	0x0002	2	ASCII z. B. "3"1"2"1" = V3.12t - gives back the soft- or hardware version	
PNR_UNIT_NAME	0x0003	3	ASCII i. e. "L4"8"" = L48	
PNR_UNIT_TYP	0x0004	4	Device type	
PNR_VNUMBER	0x0005	5	Article number	
PNR_SN	0x0006	6	Serial number	
PNR_OPTION_X	0x0007	7	Option X	
PNR_ENCODER_TYP	0x0010	16	Encoder type	chapter 6.5.1
PNR_RESOLUTION_PER_TURN	0x0011	17	Real-resolution per revolution	chapter 6.5.2
PNR_NUMBER_OF_TURNS	0x0012	18	Real-number revolution	
PNR_SCALED_ENCODER_RES	0x0013	19	Virtual encoder value	
PNR_ENCODER_INVERT	0x0014	20	Reversal of rotational direction	chapter 6.5.3
PNR_SCALED_COUNT_RANGE	0x0017	23	Virtual count range	
PNR_COUNT_RANGE	0x0018	24	Counting area at incremental encoders	
PNR_COUNT_RESTORE_VALUE	0x0019	25	At X 16: = brake point	
PNR_TIMEBASE	0x001C	28	Time basis at timer	
PNR_DEADTIME_BASE_US	0x001D	29	Time unit for idle time compensation in µs (if not defined -> 1000µs)	
PNR_NUMBER_OUTPUTS	0x0020	32	Number of outputs	
PNR_NUMBER_LOCK_OUTPUTS	0x0021	33	Number of locked outputs	
PNR_NUMBER_DATA_RECORDS	0x0022	34	Number of data records	
PNR_NUMBER_LOGIC_INPUTS	0x0023	35	Number of Logic inputs	
PNR_NUMBER_ANGLE_TIME	0x0024	36	Number of angle/time outputs from output 1	
PNR_NUMBER_OUTNAME_CHAR	0x0025	37	Output names	
PNR_NUMBER_PROGRAMS	0x0026	38	Number of programs	
PNR_NUMBER_AXIS	0x0027	39	Number of axes	
PNR_NUMBER_ANALOGOUTPUT	0x0028	40	Number of analog outputs	
PNR_NUMBER_COUNTERCAM	0x0029	41	Number of counter cams	
PNR_FIRST_OUTPUT_NR	0x002A	42	Counting starts at 1	
PNR_SPEED_SCALE	0x0030	48	With reference to rev./ms =>60000 = rev./min 0...9999 (rev./s)	
PNR_LANGUAGE	0x0031	49	Language	chapter 6.5.4
PNR_DEADTIME_TYP	0x0032	50	ITC-type	chapter 6.5.5
PNR_ZEROPOINT_OFFSET	0x0033	51	Scaled preset value at inc.	
PNR_ACTIV_PROGMR	0x0034	52	Active program	0..max program -1
PNR_ACTIV_AXIS	0x0035	53	Active axis	1..max AxisNo.
PNR_CALC_SPEED_START	0x0036	54	IdleStart scaled	
PNR_CALC_SPEED_STOP	0x0037	55	IdleStop scaled	
PNR_DICNET_ID	0x0038	56	Actual value (PLS = 80..95), RS232 = 232	
PNR_CLEAR_LENGTH	0x0039	57	Length clear pulse	
PNR_BREAK_PARA	0x003A	58	(BrakeA*0x10000) + BrakeB	
PNR_OUTPUT_OFF_SPEED	0x003B	59	Speed-threshold value below which the outputs are switched off	
PNR_WZ_MAXTIME	0x003C	60	Time in ms	
PNR_WZ_TIMEBASE	0x003D	61	Time in µs	
PNR_V_LIMIT	0x003E	62	M13 = 1, if V_LIMIT is exceeded	
PNR_DREHSCHALTER	0x003F	63	Read switch position	
PNR_RESTART	0x004E	78	Warmstart with value 1:0x1234 -> 2:0xEDCB	
PNR_CLEAR_EEROM	0x004F	79	General deletion: 1: 0x1234 -> 2:0xEDCB	
PNR_STATUS_FLAGS	0x0050	80		
PNR_PROC_OUT_MAPPING	0x0051	81	Mapping of the process data in the Fieldbus	
PNR_PROC_IN_MAPPING	0x0052	82	Mapping of the process data in the Fieldbus	
PNR_USED_EEROM_LEN	0x0053	83	Actual used EEROM length	
PNR_S7_MODE	0x0054	84	1 = S7 do not copy data into the EEROM	
PNR_RESET_EEROM	0x0055	85	Set to set in factory 1:0x1234 -> 2:0xEDCB	
PNR_CYCLETIME	0x0056	86	Read cycle time	
PNR_AKTIV_STATUS	0x0057	87		
PNR_PROC_LOAD	0x0058	88	Processor utilization	
PNR_ENABLE_OPTION	0x0059	89	Release of options	
PNR_TEACH_IN_ZEROPOINT	0x0060	90	Teach-in zero offset	
PNR_ENABLE_TESTMODE	0x005B	91	With 0x1234 -> Switch to testmode	
PNR_ERROR_QUIT	0x0060		Error quit through Modbus 0 -> 1	

Example: Change the encoder resolution to "zero".

### 7. Changing active program

16: PRESET MULTIPLE REGISTERS

Slave Device: 1

Address: 1127

Number of Points: 2

OK Cancel

---

16: PRESET MULTIPLE REGISTERS

Address: 1127

Length: 0002

1127: 0000 (HEX)

1128: 0001 (HEX)

From File

To File

Update Cancel

Parameter in accordance with parameter table in the instruction manual (read-write):

4\_0401..4\_0402 Parameter 1

4\_0467..4\_0468 Parameter 52

4\_05FF..4\_0600 Parameter 256

! Parameter 256 = ModbusProgNr (default = 0) !  
 Parameter address = 0401H+((p-1\*2))

Command's name	Command's value	Parameter number in config-menu of TERM 6	Meaning	Explanation
PNR_SOFT_REV	0x0001	1	s_PNR_HARD_REV	
PNR_HARD_REV	0x0002	2	ASCII Z, B: "P" "V" = V3.12t - gives back the soft- or hardware version	
PNR_UNIT_NAME	0x0003	3	ASCII L: e: "L" "4" "8" " " = L48	
PNR_UNIT_TYP	0x0004	4	Device type	
PNR_VNUMBER	0x0005	5	Article number	
PNR_SN	0x0006	6	Serial number	
PNR_OPTION_X	0x0007	7	Option X	
PNR_ENCODER_TYP	0x0010	16	Encoder type	chapter 6.5.1
PNR_RESOLUTION_PER_TURN	0x0011	17	Real-resolution per revolution	chapter 6.5.2
PNR_NUMBER_OF_TURNS	0x0012	18	Real-number revolution	
PNR_SCALED_ENCODER_RES	0x0013	19	Virtual encoder value	
PNR_ENCODER_INVERT	0x0014	20	Reversal of rotational direction	chapter 6.5.3
PNR_SCALED_COUNT_RANGE	0x0017	23	Virtual count range	
PNR_COUNT_RANGE	0x0018	24	Counting area at incremental encoders	
PNR_COUNT_RESTORE_VALUE	0x0019	25	At X 16: = brake point	
PNR_TIMEBASE	0x001C	28	Time basis at Timer	
PNR_DEADTIME_BASE_US	0x001D	29	Time unit for idle time compensation in µs (if not defined -> 1000µs)	
PNR_NUMBER_OUTPUTS	0x0020	32	Number of outputs	
PNR_NUMBER_LOCK_OUTPUTS	0x0021	33	Number of locked outputs	
PNR_NUMBER_DATA_RECORDS	0x0022	34	Number of data records	
PNR_NUMBER_LOGIC_INPUTS	0x0023	35	Number of Logic inputs	
PNR_NUMBER_ANGLE_TIME	0x0024	36	Number of angle/time outputs from output 1	
PNR_NUMBER_OUTNAME_CHAR	0x0025	37	Output names	
PNR_NUMBER_PROGRAMS	0x0026	38	Number of programs	
PNR_NUMBER_AXIS	0x0027	39	Number of axes	
PNR_NUMBER_ANALOGOUTPUT	0x0028	40	Number of analog outputs	
PNR_NUMBER_COUNTERCAM	0x0029	41	Number of counter cams	
PNR_FIRST_OUTPUT_NR	0x002A	42	Counting starts at 1	
PNR_SPEED_SCALE	0x0030	48	With reference to rev./ms => 60000 = rev./min 0...9999 (rev./s)	
PNR_LANGUAGE	0x0031	49	Language	chapter 6.5.4
PNR_DEADTIME_TYP	0x0032	50	ITC-type	chapter 6.5.5
PNR_ZEROPOINT_OFFSET	0x0033	51	Scaled preset value at inc.	
PNR_ACTIV_PROGNR	0x0034	52	Active program	0..max program -1
PNR_ACTIV_AXIS	0x0035	53	Active axis	1..max AxisNo.
PNR_CALC_SPEED_START	0x0036	54	IdleStart scaled	
PNR_CALC_SPEED_STOP	0x0037	55	IdleStop scaled	
PNR_DICNET_ID	0x0038	56	Actual value (PLS = 80..95), RS232 = 232	
PNR_CLEAR_LENGTH	0x0039	57	Length clear pulse	
PNR_BREAK_PARA	0x003A	58	(BrakeA*0x10000) + BrakeB	
PNR_OUTPUT_OFF_SPEED	0x003B	59	Speed-threshold value below which the outputs are switched off	
PNR_WZ_MAXTIME	0x003C	60	Time in ms	
PNR_WZ_TIMEBASE	0x003D	61	Time in µs	
PNR_V_LIMIT	0x003E	62	M13 = 1, if V_LIMIT is exceeded	
PNR_DREHSCHALTER	0x003F	63	Read switch position	
PNR_RESTART	0x004E	78	Warmstart with value 1:0x1234 -> 2:0xEDCB	
PNR_CLEAR_EEROM	0x004F	79	General deletion: 1: 0x1234 -> 2:0xEDCB	
PNR_STATUS_FLAGS	0x0050	80		
PNR_PROC_OUT_MAPPING	0x0051	81	Mapping of the process data in the Fieldbus	
PNR_PROC_IN_MAPPING	0x0052	82	Mapping of the process data in the Fieldbus	
PNR_USED_EEROM_LEN	0x0053	83	Actual used EEROM length	
PNR_S7_MODE	0x0054	84	1 = S7 do not copy data into the EEROM	
PNR_RESET_EEROM	0x0055	85	Set to set in factory 1:0x1234 -> 2:0xEDCB	
PNR_CYCLETIME	0x0056	86	Read cycle time	
PNR_AKTIV_STATUS	0x0057	87		
PNR_PROC_LOAD	0x0058	88	Processor utilization	
PNR_ENABLE_OPTION	0x0059	89	Release of options	
PNR_TEACH_IN_ZEROPOINT	0x0060	90	Teach-in zero offset	
PNR_ENABLE_TESTMODE	0x005B	91	With 0x1234 -> Switch to testmode	
PNR_ERROR_QUIT	0x0080		Error quit through Modbus 0 -> 1	

Example: Change the active program to program 1.

## 8. Programming in program

**16: PRESET MULTIPLE REGISTERS** [X]

Slave Device:

Address:

Number of Points:

Parameter in accordance with parameter table in the instruction manual (read-write):

---

4_0401..4_0402	Parameter 1
.....	
4_05FF..4_0600	Parameter 256

**! Parameter 256 = ModbusProgNr (default = 0) !**  
 Parameter address = 0401H+((p-1\*2))

**16: PRESET MULTIPLE REGISTERS** [X]

Address: 1535  
Length: 0002

1535:	<input style="border: 1px solid gray;" type="text" value="0000"/>	(HEX)
1536:	<input style="border: 2px solid orange;" type="text" value="0001"/>	(HEX)

Example: Change the programming program to program number 1.

## 9. Available Modbus addresses

The available Modbus addresses for LOCON 100-MB and LOCON 200-MB are listed in the document [‘LOCON100-200-Modbus-RTU-addresses.xlsx’](#).