Serial protocol documentation of the Luigs & Neumann manipulator control systems

Version 1.8 from 01.06.2011

Description of the serial protocol for the manipulator control systems starting from the SSK version 2.8.3..... 4
Instructions:..... 6
Connection establishment:..... 6
Connection clearing: 6
Connection kept upright: 6
Procedure:..... 7
Single step: 7
Setting the incrementation for the "Stepp instruction“..... 8
Approaching a position: 8
Approaching a stored position:..... 8
Storing a position:..... 9
Positioning velocity stop: 9
Switchind axis on/off: 10
Setting the positioning velocity:.....10
Setting homing velocity stop: 10
Setting homing direction: 11
Setting single step resolution: 11
Setting the ramp length: 11
Resetting the position counter 2: 12
StepIncrement: 12
StepDecrement: 12
Home:..... 12
HomeReturn: 13
SetPositionNull: 13
GotoPositionNull:..... 13
Stop: 13
Keypad off: 14
Keypad on: 14
SlowMove ramp off: 14
SlowMove ramp on: 14
Inquiries: 15
Position inquiries:..... 15
Position inquiry of counter 2:.....15
Positioning velocity inquiry for fast travel: 15
Positioning velocity inquiry for slow travel:..... 16
Inquiry for step velocity:..... 16
Inquiry for home velocity: 16
Inquiry for traversing velocity: 17
Home direction inquiry: 17
Inquiry for the axis status: 18
Inquiry for the manipulator gradient: 18
Inquiry for the type of motor: 19
Version inquiry..: 19
Status inquiry for an output stage: 20
Inquiry for SlowMove ramp 21

Appendix:..... 22
Allocation of ramp values - > stages 1-16..... 22
Allocation of velocities applies for stages 1-16 (Control SM5 / SM6)..... 22
Allocation of velocities applies for stages 1-16 (Control SM7 / SM8)..... 23
Allocation of the unit numbers (Control SM5 / SM6)..... 24
Allocation of the unit numbers (Control SM7 / SM8)..... 25
Calculation of the CRC16-Checksum..... 27
History: 28

Description of the serial protocol for the manipulator control systems starting from the SSK version 2.8.3

This protocol enables to completely control all manipulators via the V24-interface or USB.

In principle the communication must be initiated by the PC.

For this the following data format is to be kept:

- Byte 1: has to be 0x16 < syn >
- Byte 2+3: ID of the desired action, whereas byte 2 represents the MSB and byte 3 the LSB
- Byte 4: number of following data bytes (0<=n<=20)?
- Byte 5..n+4: data
- Byte n+5: MSB of the CRC-16 check sum
- Byte n+6: LSB of the CRC-16 check sum

The Baud rate is 38400 Baud, 8 Data bit, NoParity and 1 Stopbit after switch power on.

Each Frame is answered by the manipulator control system.
Here one differentiates towards the control between an inquiry and an.
An instruction implements a certain action, however no data is returned.
If the instruction is recognized, the answer < Ack > is given.
Given an unknown instruction or an incorrect syntax the answer is < NAck >.

An inquiry is developed like an instruction, returns however besides < Ack > or < NAck > the requested data.

If the inquiry could not be answered, < NAck > is returned. In this case the length of the data field is 0.

In order to receive a high accuracy, the inquired data is not converted into ASCII-characters, but are binary displayed.
Here principally the LSB is transferred first.

The format is as follows:

- float - > 4 byte in the standard IEEE format
  bit 31=Characters, 30-23=Exponent, 22-0=Mantisse
  transmission: < LSB><Byte2><Byte3><MSB >
- Uint - > 2 byte < LSB > < MSB >

Before instructions or inquiries can be answered it is necessary to set up a data link with the interface card. For this the instruction with the ID 0x0400 has to be sent.
In the following there is a timeout of 3000 ms.
If within this time no instruction or no inquiry is received, the interface card terminates the connection and changes into the normal operating mode.
A regular connection termination is initiated by means of the ID 0x0401 whilst if the break between the actual instructions is longer than 3000 ms, then the connection can be kept upright by means of the ID 0x0402.

In general most settings are stored permanently and are made available after again switching on the equipment. If any deviations to this behaviour arise then they will be referred to separately.

The settings are axis-related, i.e. it is possible to give different values each axis.

During an inquiry extension the length of the data field is changed and new data is attached to the end of the data field, so that it does not result with an incompatibility with the existing code.
This presumes however that the existing code can handle the changed values in the "length" field of the data frames.

Instructions:

Connection establishment:

After successfully making the connection the message appears:
"Connection to PC is established" in the display of the system.

- establishConnection ID = 0x0400

Syntax: < syn><ID><0><crc >

Answer of the system:
< ack><0x040b><0><crc >

Connection clearing:

- releaseConnection ID = 0x0401

Syntax: < syn><ID><0><crc >

Answer of the system:
< ack><0x040b><0><crc >

Connection kept upright:

In order to keep the connection upright an instruction or an inquiry must be sent to the interface card every 3000 ms.
This instruction effectively only maintains the connection, implements however no other action.

- ConnectionKeepAlive ID = 0x0402

Syntax: < syn><ID><0><crc >

Answer of the system:
< ack><0x0402><0><crc >

Important: The ID on this instruction is only given back once by the system.
No other instruction or inquiry answers with this ID.

Procedure

All procedure instructions expect the unit number of the axis to be moved within the data frame.
The procedure instruction remains active until stop is sent, or a limit switch is reached.

Important: No instruction other than stop may be sent during the procedure.

It is possible to set the engine velocity in 16 stages.

- fast procedure in positive direction ID=0x0012
- fast procedure in negative direction ID=0x0013
- slow procedure in positive direction ID=0x0014
- slow procedure in negative direction ID=0x0015

Syntax: <syn><ID><1><unit number [byte]><crc >

Answer of the system:
<ack><XXXX><0><crc >

Single step:

A certain number of single steps can be preceded by means of this instruction.
The way of procedure depends on the engine in operation and the upward gradient of the manipulator.

Note: This instruction is directly in connection with the single step resolution.

- GoSingleSteps ID = 0x0147

Syntax: < syn><ID><2>< unit number [byte] > < steps [byte] > < (crc) >

Where: -127 < steps < 128

Answer of the system:
<ack><XXXX><0><crc>

Setting the incrementation for the "step instruction":

- StepSlowDistance ID=0x013a

Syntax: <syn><ID><5>< unit number [byte] >< incrementation in µm [float]><crc >

The system answers with:
<ack><XXXX><0>crc>

Approaching a position:

Any position can be approached. The values that are found under positioning velocities are used.
A position can be approached absolutely or relative to the current position.

Example:
Current position: +100 µm
In order proceed to the goal position -500 µm one can use the instruction [ID=0x0048/49] with the parameter -500, or however the instruction [ID=0x004a/4b] with the parameter -600.

- GoVariableFastToAbsolutePosition ID=0x0048
- GoVariableSlowToAbsolutePos ID=0x0049
- GoVariableFastToRelativePos ID=0x004a
- GoVariableSlowToRelativePos ID=0x004b

Syntax: <syn><ID><5>< unit number [byte] > < goal position µm [float]><crc >

The system answers with:
<ack><XXXX><0>crc >

Approaching a stored position:

One of the 16 previously stored positions is to be approached. This position is approached with the velocity set in "positioning velocity".

- GotoPosition ID=0x0110

Inquiry with:
<syn><ID><2>< unit number [byte] >< Number[byte]><crc>

where: 0 < number < = 16

The system answers with:
<ack><XXXX><0>crc>

Storing a position:

There is the possibility of storing up to 16 positions in each output stage and to approach these when required. As soon as the instruction is received by the output stage the current position is stored.

- SavePosition ID=0x010a

Syntax: <syn><ID><2><unit number [byte] >< number [byte] > < crc >

Where: 0 < number < = 16

Answer of the system:
<ack><XXXX><0><crc>

Positioning velocity stop:

The positioning velocity can be adjusted in 16 stages to enable fast positioning. Additionally it is possible to set the positioning velocity in full steps per second for the fast positioning and/or in µ-steps per second for the slow positioning.
Important: Although 2 separate possibilities exist, these are however dependent from each other.
I.E. previous settings done with SetPositioningVelocityFast are overwriten with SetPositioningVelocityFastLinear.

- SetPositioningVelocityFast ID=0x0144
- SetPositioningVelocitySlowLinear ID=0x003c
- SetPositioningVelocityFastLinear ID=0x003d

To apply SetPositioningVelocityFast:
Syntax: <syn><ID><2><unit number [byte] >< velocity [byte] >< (crc) >

where: 0< velocity < = 16

To apply SetPositioningVelocitySlowLinear and SetPositioningVelocityFastLinear:
Syntax: <syn><ID><3><unit number[byte] >< velocity [uInt] >< (crc) >

Where:
0 < SetPositioningVelocitySlowLinear < 18000
0 < SetPositioningVelocityFastLinear < 3000

Answer of the system:
<ack><XXXX><0><crc>

Switching axis on/off:

- Switching the axis off (dead) ID=0x0034
- Switching the axis on ID=0x0035

Syntax: <syn><ID><1><unit number [byte] ><crc>

Answer of the system:
<ack><XXXX><0><crc>

Setting the positioning velocity:

The positioning velocity can be adjusted both separately for slow and fast approaches in 16 stages.

- velocity for slow procedure ID=0x0135
- velocity for fast procedure ID=0x0134

Syntax: <syn><ID><2><unit number[byte] >< velocity [byte]><crc>

Where: 0< velocity < = 16

Answer of the system:
<ack><XXXX><0><crc>

Setting homing velocity stop:

The Home position is approached with a variable speed.
Here 16 different speeds are available.

- SetHomeVelocity ID = 0x0139

Syntax: <syn><ID><2><unit number[byte] >< velocity [byte]><crc>

Where: 0< velocity < = 16

Answer of the system:
<ack><XXXX><0><crc>

Setting homing direction:

- SetHomeDirection ID = 0x013c

Syntax: <syn><ID><2><unit number[byte] >< direction [byte]><crc>

Where:
Direction = 0 - > direction of positiv approach
Direction = 1 - > direction of negativ approach

Answer of the system:
<ack><XXXX><0><crc>

Setting single step resolution:

Here the number of µ-steps is set, that are to be completed per single step command

- SetHandwheelResolution ID = 0x0146

Syntax: <syn><ID><2><unit number [byte]><resolution [byte] ><crc>

Where:
0 < dissolution < = 255

Answer of the system:
<ack><XXXX><0><crc>

Setting the ramp length:

The length of the acceleration and of the braking ramp can be set in 16 stages.

Important: A too short ramp can lead to step losses.

- Setting the starting - stop ramp length ID=0x003a

Syntax: <syn><ID><2><unit number [byte]><lenght [byte]><crc>

Where:
0<length<=16

Answer of the system:
<ack><XXXX><0><crc>

Resetting the position counter 2:

Besides the main position counter a further independent counter is available. This can be separately set to zero. The first data byte after the unit number indicates the counter and must be 2 .

- Zeroising the 2nd counter ID=0x0132

Syntax: <syn><ID><2><unit number[byte]><2><crc>

Answer of the system:
<ack><XXXX><0><crc>

StepIncrement:

The StepIncrement instruction traverses the indicated axis with the preset distance in positive direction.

- StepIncrement ID = 0x0140

Syntax: <syn><ID><1><unit number [byte]><crc>

Answer of the system:
<ack><XXXX><0><crc>

StepDecrement:

The StepDecrement instruction traverses the indicated axis with the preset distance in negative direction

- StepDecrement ID = 0x0141

Syntax: <syn><ID><1><unit number [byte]><crc>

Answer of the system:
<ack><XXXX><0><crc>

Home:

The Home instruction stores the current position of the manipulator and then traverses it with the set velocity and the set direction to the limit switch.

- Home ID = 0x0104

Syntax: <syn><ID><1><unit number [byte]><crc>

Answer of the system:
<ack><XXXX><0><crc>

HomeReturn:

The HomeReturn instruction can only be implemented after a Home instruction.
It traverses the manipulator from the limit switch back to the position stored with the Home instruction.

- HomeReturn ID = 0x0022

Syntax: <syn><ID><1><unit number [byte]><crc>

Answer of the system:
<ack><XXXX><0><crc>

Set position zero:

This instruction sets the location counter 1 back to 0.

- SetPositionNull ID = 0x00f0

Syntax: <syn><ID><1>< unit number [byte]><crc>

Answer of the system:
<ack><XXXX><0><crc>

GotoPositionNull:

Proceeding to zero-position.

- GotoPositionNull ID = 0x0024

Syntax: <syn><ID><1>< unit number [byte]><crc>

Answer of the system:
<ack><XXXX><0><crc>

Stop:

The "stop instruction " interrupts the current instruction and stops a probable traversing axis.

- Stop ID = 0x00ff

Syntax: <syn><ID><1>< unit number [byte]><crc>

Answer of the system:
<ack><XXXX><0><crc>

Switch off the keypad:

This instruction switches off all the external connected appliances and ends the communication between them.
A corresponding message is shown on the display of the actual appliance.

- KeypadOff ID = 0x042d

Syntax: <syn><ID><0><crc>

Answer of the system:
<ack><0x042d><0><crc>

Switch on the keypad:

This instruction switches on all the external appliances and initiates new communication between them.

- KeypadOn ID = 0x042c

Syntax: <syn><ID><0><crc>

Answer of the system:
<ack><0x042c><0><crc>

Switch off the Slow Move ramp:

Switch off the ramp when proceeding in SlowMove. The motor starts immediately with the preset speed.

- SlowMoveRampOff ID = 0x042f

Syntax: <syn><ID><1><unit number><crc>

Answer of the system:
<ack><0x042d><0><crc>

Switch on the SlowMove ramps:

Switch on the ramp when proceeding in SLowMove.
The motor starts with one ramp.

- SlowMoveRampOn ID = 0x0430

Syntax: <syn><ID><1><unit number><crc>

Answer of the system:
<ack><XXXX><0><crc>

Inquiries:

Position inquiry:

The current position of the inquired output stage is returned in µm.

- position inquiry of a certain output stage ID=0x0101

Inquiry with:
<syn><ID><1><unit number[byte]><crc>

The system answers with:
<ack><XXXX><4><position in µm [float]><crc>

Position inquiry of counter 2:

Additionally to the position an independent 2nd position can be queried.
This can be zeroed separately.

- Inquiry of 2nd position ID=0x0131

Inquiry with:
<syn><ID><1>< unit number [byte] ><crc>

The system answers with:
<ack><XXXX><4><position in µm [float]><crc>

Positioning velocity inquiry for fast travel:

This velocity is used to proceed to a position.

- QueryPositioningVelocityFastLinear ID=0x0160

Inquiry with:
<syn><ID><1>< unit number [byte] ><crc>

Answer to: "QueryPositioningVelocityFastLinear":
<ack><XXXX><2><velocity[uInt]><crc>
whereby velocity = full steps per second.

Positioning velocity inquiry for slow travel:

This velocity is used to proceed to a position slowly.

- QueryPositioningVelocitySlowLinear ID=0x0161

Inquiry with:
<syn><ID><1>< unit number [byte] ><crc>

Answer to: "QueryPositioningVelocitySlowLinear" with:
<ack><XXXX><2>< velocity [uInt]><crc>
whereby velocity = µ-steps per second.

Inquiry for step velocity:

This speed is used in order to slowly traverse a pre-programmed step.
Note: Step instructions cannot be carried out using fast velocity

- QueryStepSlowVelocity ID=0x0159

Inquiry with:
<syn><ID><1><unit number [byte] ><crc>

The system answers with:
<ack><XXXX><1><velocity [byte]><crc>
whereby 0 < velocity < = 16

Inquiry for home velocity:

- QueryHomeVelocity ID = 0x0138

Inquiry with:
<syn><ID><1><unit number [byte] ><crc>

The system answers with:
<ack><XXXX><1><velocity [byte]><crc>
whereby 0 < velocity < = 16

Inquiry for traversing velocity:

The traversing velocity can be inquired separately for slow and fast proceedings.

- QueryFastMoveVelocity ID=0x012f
- QuerySlowMoveVelocity ID=0x0130

Inquiry with:
<syn><ID><1><unit number [byte] ><crc>

The system answers with:
<ack><XXXX><1><velocity [byte]><crc>
whereby 0 < velocity < = 16

Home direction inquiry:

- QueryHomeDirection ID = 0x013d

Inquiry with:
<syn><ID><1><unit number [byte] ><crc>

The system answers with:
<ack><XXXX><1><direction [byte]><crc>

Whereby :
Positive traversing direction - > direction=0
Negatives traversing direction - > direction=1

Inquiry for whether the output stage is physically present:

- QueryOutputstagePresent ID=0x011f

Inquiry with:
<syn><ID><1><unit number[byte]><crc>

The system answers with:
<ack><XXXX><1><status[byte]><crc>

Whereby < status > has the following meaning:

Output stage available: Status=1
Output stage missing: Status=0

Inquiry for the axis status:

The current power status of the axis is returned.
Take into consideration that while the axis is powerless all instructions other than "switch on axis" and the "version inquiry" are ignored.

- GetPowerStatusFromOutputstage ID = 0x011e

Inquiry with:
<syn><ID><1><unit number [byte]><crc>

The system answers with:
<ack><XXXX><1><status[byte]><crc>

whereby < status > has the following meaning:

Axis switched on: status=1
Axis without power: status=0

Inquiry for the manipulator gradient:

Shows the pitch gradient for the selected manipulator axis.

- QueryManipulatorPitch ID = 0x014d

Inquiry with:
<syn><ID><1><unit number[byte]><crc>

The system answers with:
<ack><XXXX><1><pitch[byte]><crc>

Whereby < pitch > has the following meaning:

pitch=0 - > pitch gradient =0,02 mm
pitch=1 - > pitch gradient =0,05mm
pitch=2 - > pitch gradient =0,1 mm
pitch=3 - > pitch gradient =0,125 mm
pitch=4 - > pitch gradient =0,175 mm
pitch=5 - > pitch gradient =0,35 mm
pitch=6 - > pitch gradient =0,4 mm
pitch=7 - > pitch gradient =0,5 mm
pitch=8 - > pitch gradient =1,0 mm
pitch=9 - > pitch gradient =2,0 mm
pitch=a - > pitch gradient =0,297 mm

Inquiry for the type of motor:

Shows the motor used in the queried axis.

- QueryMotortype ID = 0x014b

Inquiry with:
<syn><ID><1><unit number[byte]><crc>

The system answers with:
<ack><XXXX><1><type[byte]><crc>

whereby < type > has the following meaning:

Type=0 - > motor used: P310
Type=1 - > motor used: P430
Type=2 - > motor used: P530
Type=3 - > motor used: PK223
Type=4 - > motor used: PK233
Type=5 - > motor used: ST2018
Type=6 - > motor used: PK244P
Type=7 - > motor used: PK244M

The motors P430 and P530 complete 100 full steps per revolution, the motor P310 complete per revolution 60 full steps, the motors PK223, PK233, PK244P and ST2018 complete 200 full steps per revolution.
The motor PK244M complete per revolution 400 full steps.

Version inquiry:

The manipulator system contains - depending on the configuration - over 50 microprocessors with partially different firmware versions.
It is possibility to inquire the version number for every firmware. A version number always consists of a combination of 3 numbers.
Major version, minor version and sub minor version.
4 IDs are available for the inquiry:

- Version inquiry key field ID = 0x015a
- Version inquiry interface card ID = 0x015b
- Version inquiry output stage Main controller ID = 0x015c
- Version inquiry output stage motor controller ID = 0x015d

Inquiry with:
<syn><ID><1><unit number [byte] ><crc>

The system answers with:
<ack><XXXX><3 >< majorVersion[byte]>< minorVersion[byte] ><subMinorVersion[byte] ><crc>

Status inquiry for an output stage:

This inquiry enables the possibility to have the most important axes data in one Date frame.

- GetMainStatusFromOutputstage ID = 0x0120

Inquiry with:
<syn><ID><1><unit number [byte] ><crc>

The system answers with:
<ack><XXXX><6 >
<status limit switch[byte]>
<status power to axes [byte]>
<status home [byte]>
<reserved [byte]>
<reserved [byte]>
<resolution for single step [byte]>
<motor status [byte]>
<crc>

Whereby:

- Status limit switches:
  - no limit switch operates 0
  - limit switch operates in negative direction 1
  - limit switch operates in positive direction 2

- Status axis power
  - axis without power 0
  - axis switched on 1

- Status Home:
  - home inactive 0
  - motor proceeds towards negative limit switch 1
  - motor proceeds toward positive limit switch 2
  - motor is located at the limit switch, and/or the Homing was interrupted 3

- Resolution single step:
  - number per "GoSingleStep instruction" committed µ-steps 0<x<=255

- Motor status:
  - motor is standing 0
  - motor is runing 1

Data indicated with < reserved > may contain undefined values.

Inquiry SlowMove Ramps:

Queries whether the ramp for the slow are process on or off.

- QuerySlowMoveRampState ID = 0x0431

Inquiry with:
<syn><ID><1><unit number[byte]><crc>

The system answers with:
<ack><XXXX><1><status[byte]><crc>

whereby < status > has the following meaning:

0 -> The ramp is switch off
All other value -> ramp is switch on

Appendix:

Allocation of ramp values - > stages 1-16

Ramp values	
stage	Duration in ms
1	150
2	180
3	210
4	240
5	270
6	300
7	330
8	360
9	390
10	420
11	450
12	480
13	510
14	530
15	570
16	600

Allocation of velocities applies for stages 1-16

This allocation is for all adjustable velocities:

Control SM5 with Keypad SM5

Slow velocity		Fast velocity	
stage	revolution rps	stage	revolution rps
1	0,00464	1	0,8
2	0,00668	2	1,0
3	0,01	3	1,4
4	0,01334	4	2,0
5	0,01668	5	2,8
6	0,02	6	3,8
7	0,03	7	5,0
8	0,05	8	6,5
9	0,1	9	8,5
10	0,15	10	10,5
11	0,25	11	13,5
12	0,35	12	16,5
13	0,45	13	19,5
14	0,55	14	22,5
15	0,65	15	26,0
16	0,75	16	30,0

Control SM7 / SM8 with Keypad SM7
Motor with 200 fullsteps

Slow velocity		Fast velocity	
stage	revolution rps	stage	revolution rps
1	0,000017	1	0,66
2	0,00004	2	1,73
3	0,000067	3	2,63
4	0,000141	4	3,79
5	0,00026	5	4,67
6	0,000528	6	5,68
7	0,00263	7	6,33
8	0,00507	8	7,81
9	0,0102	9	8,47
10	0,0251	10	9,52
11	0,046	11	10,42
12	0,0601	12	11,36
13	0,0905	13	12,32
14	0,122	14	13,23
15	0,173	15	14,29
16	0,332	16	15,15

Control SM7 / SM8 with Keypad SM7
Motor with 100 fullsteps

Slow velocity		Fast velocity	
stage	revolution rps	stage	revolution rps
1	0,000034	1	1,32
2	0,00008	2	3,46
3	0,000134	3	5,26
4	0,000282	4	7,58
5	0,00052	5	9,34
6	0,001056	6	11,36
7	0,00526	7	12,66
8	0,01014	8	15,62
9	0,0204	9	16,94
10	0,0502	10	19,04
11	0,092	11	20,84
12	0,1202	12	22,72
13	0,181	13	24,64
14	0,244	14	26,46
15	0,346	15	28,58
16	0,664	16	30,3

Allocation of the unit numbers:

Control SM5 / SM6 with Keypad SM5 1/2

Address distribution			
Equipment	I2C-Address within the bus	bus	General unit number
Unit 1	1	Rack1	1
Unit 2	2	Rack1	2
Unit 3	3	Rack1	3
Unit 4	4	Rack1	4
Unit 5	5	Rack1	5
Unit 6	6	Rack1	6
Unit 1	1	Rack2	7
Unit 2	2	Rack2	8
Unit 3	3	Rack2	9
Unit 4	4	Rack2	10
Unit 5	5	Rack2	11
Unit 6	6	Rack2	12
Unit 1	1	Rack3	13
Unit 2	2	Rack3	14
Unit 3	3	Rack3	15
Unit 4	4	Rack3	16
Unit 5	5	Rack3	17
Unit 6	6	Rack3	18
Unit 1	1	Rack4	19
Unit 2	2	Rack4	20
Unit 3	3	Rack4	21
Unit 4	4	Rack4	22
Unit 5	5	Rack4	23
Unit 6	6	Rack4	24
Unit 1	1	Rack5	25
Unit 2	2	Rack5	26
Unit 3	3	Rack5	27
Unit 4	4	Rack5	28
Unit 5	5	Rack5	29
Unit 6	6	Rack5	30
Unit 1	1	Rack6	31
Unit 2	2	Rack6	32
Unit 3	3	Rack6	33
Unit 4	4	Rack6	34
Unit 5	5	Rack6	35
Unit 6	6	Rack6	36

Control SM5 / SM6 with Keypad SM5 2/2

Address distribution			
Equipment	I2C-Address within the bus	bus	General unit number
Unit 1	1	Rack7	37
Unit 2	2	Rack7	38
Unit 3	3	Rack7	39
Unit 4	4	Rack7	40
Unit 5	5	Rack7	41
Unit 6	6	Rack7	42
Unit 1	1	Rack8	43
Unit 2	2	Rack8	44
Unit 3	3	Rack8	45
Unit 4	4	Rack8	46
Unit 5	5	Rack8	47
Unit 6	6	Rack8	48
SSK1	10	SSK	100
SSK2	20	SSK	101
SSK3	30	SSK	102
SSK4	40	SSK	103
SSK5	50	SSK	104
SSK6	60	SSK	105
SSK7	70	SSK	106
SSK8	80	SSK	107
Keypad	9	SSK	9

Allocation of the unit numbers:

Control SM7 / SM8 with Keypad SM7 1/3

Address distribution			
Equipment	I2C-Address within the bus	bus	General unit number
Unit 1	1	Rack1	1
Unit 2	2	Rack1	2
Unit 3	3	Rack1	3
Unit 4	4	Rack1	4
Unit 5	5	Rack1	5
Unit 6	6	Rack1	6
Unit 7	7	Rack1	7
Unit 8	8	Rack1	8
Unit 9	9	Rack1	9
Unit 1	1	Rack2	10
Unit 2	2	Rack2	11
Unit 3	3	Rack2	12
Unit 4	4	Rack2	13
Unit 5	5	Rack2	14
Unit 6	6	Rack2	15
Unit 7	7	Rack2	16
Unit 8	8	Rack2	17
Unit 9	9	Rack2	18
Unit 1	1	Rack3	19
Unit 2	2	Rack3	20
Unit 3	3	Rack3	21
Unit 4	4	Rack3	22
Unit 5	5	Rack3	23
Unit 6	6	Rack3	24
Unit 7	7	Rack3	25
Unit 8	8	Rack3	26
Unit 9	9	Rack3	27
Unit 1	1	Rack4	28
Unit 2	2	Rack4	29
Unit 3	3	Rack4	30
Unit 4	4	Rack4	31
Unit 5	5	Rack4	32
Unit 6	6	Rack4	33
Unit 7	7	Rack4	34
Unit 8	8	Rack4	35
Unit 9	9	Rack4	36

Control SM7 / SM8 with Keypad SM7 2/3

Address distribution			
Equipment	I2C-Address within the bus	bus	General unit number
Unit 1	1	Rack5	37
Unit 2	2	Rack5	38
Unit 3	3	Rack5	39
Unit 4	4	Rack5	40
Unit 5	5	Rack5	41
Unit 6	6	Rack5	42
Unit 7	7	Rack5	43
Unit 8	8	Rack5	44
Unit 9	9	Rack5	45
Unit 1	1	Rack6	46
Unit 2	2	Rack6	47
Unit 3	3	Rack6	48
Unit 4	4	Rack6	49
Unit 5	5	Rack6	50
Unit 6	6	Rack6	51
Unit 7	7	Rack6	52
Unit 8	8	Rack6	53
Unit 9	9	Rack6	54
Unit 1	1	Rack7	55
Unit 2	2	Rack7	56
Unit 3	3	Rack7	57
Unit 4	4	Rack7	58
Unit 5	5	Rack7	59
Unit 6	6	Rack7	60
Unit 7	7	Rack7	61
Unit 8	8	Rack7	62
Unit 9	9	Rack7	63
Unit 1	1	Rack8	64
Unit 2	2	Rack8	65
Unit 3	3	Rack8	66
Unit 4	4	Rack8	67
Unit 5	5	Rack8	68
Unit 6	6	Rack8	69
Unit 7	7	Rack8	70
Unit 8	8	Rack8	71
Unit 9	9	Rack8	72

Control SM7 / SM8 with Keypad SM7 3/3

Address distribution			
Equipment	I2C-Address within the bus	bus	General unit number
SSK1	113	SSK	113
SSK2	114	SSK	114
SSK3	115	SSK	115
SSK4	116	SSK	116
SSK5	117	SSK	117
SSK6	118	SSK	118
SSK7	119	SSK	119
SSK8	120	SSK	120
Keypad	9	SSK	9

Calculation of the CRC16-checksum

The checksum uses the generator polynomial 0x1021 (`CrcPolynom`).

// ptr shows the calculated checksum on the buffer
// len shows the number of signs on the buffer
// in the checksum is returned in crcHigh and low

```
void serialCalculateCrc
(unsigned char *ptr,int len,unsigned char *crcHigh,unsigned
char *crcLow)
{
unsigned int crc;
unsigned char i;
crc = 0;
while(--len >= 0)
{
crc = crc ^ (unsigned int)*ptr++ << 8;
for (i = 0; i < 8; ++i)
if (crc & 0x8000)
crc = crc << 1 ^ CrcPolynom;
else
crc = crc << 1;
}
if(crcHigh!=NULL)
*crcHigh=(unsigned char)(crc>>8);
if(crcLow!=NULL)
*crcLow=(unsigned char)crc;
}
```

History:

Version 1.3:

- the ID for proceeding to a stored position was wrong, 0x0110 is correct.
- Setting positioning velocity:
  The instruction "SetPositioningVelocitySlow" does not apply
  The instruction "SetPositioningVelocity" was renamed "SetPositioningVelocityFast".
  The ID for "SetPositioningVelocityFastLinear" is now 0x003d and for "SetPositioningVelocitySlowLinear" 0x003c
- the instruction "QueryPositioningVelocity" does not apply
- the instruction "QueryPositioningVelocitySlow" does not apply
- when using the instruction "GetMainStatusFromOutputStage" the field <reserved > was missing
- the return ID for establishConnection was wrong, correct is 0x040b
- the return ID for releaseConnection was wrong, correct is 0x040b
- the return ID for ConnectionKeepAlive was wrong, correct is 0x0402
- the return ID's for all other instructions and inquiries are indefinite
- the instructions QueryFastMoveVelocity and QuerySlowMoveVelecity were added.
- from **SSK version 2.5.1** and onwards, the following instructions were added:
  StepIncrement, Stepdecrement, Home, HomeReturn, set position zero, GotoPositionNull

Version 1.5

- from the mainain control version 2.1.2 and onwards, 3 new upward gradients were added. Thereby the allocation of the upward gradient and the characteristic number changes.

Version 1.6

- from **SSK version 2.8.2** and onwards, the following instructions were added:
  Keypad on and off
  Switch on / off the ramps for the Slow Move
  Inquiry the ramp for the slow are process
  Added the source code to the production of the CRC16 checksum
  Parameter of the serial interface added

Version 1.7

- An incorrect General unit number was specified for the keypad
- From MaCo 2.4.5, changed the speed for the slow velocity
- From MaCo 2.4.5, following engine types have been added:
  - PK224P and PK224M

Version 1.8

- table for Control SM7 velocities applies and unit numbers added