Serial protocol documentation of the Luigs & Neumann manipulator control systems

SM-10

Description of the serial protocol for the manipulator controller SM-10

This protocol makes the complete control of all manipulators possible via
the RS232-interface or USB(COM-Interface).

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 115200 Baud, 8 Data bit, NoParity and 1 Stopbit after switching
the power on.

The manipulator control differs between single commands, collection commands and group commands, as well as between individual queries and group questions.
Each frame is answered by the manipulator control, except for collecting commands and group commands.
A distinction is made between a request or a command to the controller.
A command performs a specific action, but returns no data. If a single command is recognized, this is confirmed with <ACK>.
Regarding collecting commands and group commands there is no response, so that data transmission is not delayed
Similarly an unknown command or faulty syntax is not answered.
A single request is structured as a single command, but, next to <ACK> the requested data is returned.
In order to acquire a high accuracy, the inquired data is not converted into ASCII- characters, but is displayed in binary.
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 >

Information about the reaction times of the output stage to commands, status queries or status inquiries can be provided upon request.

Single commands:

Movement:

All move commands in the dataframe must have the unit number (01-72) of the moving axis.
The move command remains active until stop command is sent, or a limit switch is reached.

Important: No other command than stop command may be sent during the movement process.
Following the stop command the motor needs 160ms until standstill at the standad ramp.
A change of direction may only follow in standstill.

It is possible to set the speed of the motor in 16 steps.

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

Syntax:
<syn><ID><01><unit number [byte]><crc >

Answer of the system:
<ack><ID><00><crc >

Setting of the movement speed:

The movement speed can be set separately for slow and fast movement each with 16 stages.

- slow speed movement ID=0x0135
- fast speed movement ID=0x0134

Syntax:
<syn><ID><02><unit number[byte] >< velocity [byte]><crc>

Value: 0…15

Answer of the system:
<ack><ID><00><crc>

Single step:

These commands make it possible to move a certain number of single steps.
The way of movement is dependent on the motor and the manipulator pitch.

Note: This command stands in direct combination with the single step resolution.
Each number of single steps is executed over a time period of at least 500ms.
Further single step commands can be sent in this time period.
All single steps are executed after each other and extend the time of movement.

- GoSingleSteps ID = 0x0147

Syntax:
< syn><ID><02>< unit number [byte] > < steps [byte] > < (crc) >

Value: 0 <steps < 255
(1 byte in ZKK = -127 < steps < +127)

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

Setting single step resolution:

Here the number of µ-steps is set to be executed per single step command

- SetHandwheelResolution ID = 0x0146

Syntax:
<syn><ID><02><unit number [byte]><resolution [byte] ><crc>

Value: 0 < resolution < 255

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

Trackball command:

Following this command, a certain number of individual steps can be moved.
The traversed path is dependent on the motor and the manipulator pitch.

Note: This command is directly related to the proportional factor. Each number of single steps is executed over a period of at least 100ms. Within this period more trackball commands can be sent. After the last trackball command, has been sent, the motor stops after 100 ms and all non-processed pulses are deleted.

- GoTrackballMode ID= 0x01E8

Syntax:
<syn><ID><02><Gerätenummer [byte] ><Steps [byte] ><crc>

Value: 2 Byte Steps in ZKK = -32768 <Steps < +32767

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

Trackball proportional factor:

Here the proportional factor is set, which determines the speed and direction of performance.

- SetProportionalFactor ID= 0x019F

Syntax:
<syn><ID><02><Gerätenummer [byte]><Faktor [byte] ><crc>

Value: 1 Byte in ZKK = -127 < Steps < +127

Answer of the system
<ack><ID><0><crc>

StepIncrement:

The step increment command moves the specified axis to the preset distance in the positive direction. Several commands may be successively sent so that there is a continuous movement of the motor.

- StepIncrement ID = 0x0140

Syntax: <syn><ID><01><unit number [byte]><crc> Answer of the system: <ack><ID><0><crc>

StepDecrement:

The StepDecrement command moves the specified axis with the preset distance in negative direction. Several commands may be successively sent, resulting in a continuous movement.

- StepDecrement ID = 0x0141

Syntax: <syn><ID><01><unit number [byte]><crc> Answer of the system: <ack><ID><0><crc>

Setting the step distance for the "step command":

- StepSlowDistance ID=0x044F

Syntax: <syn><ID><05>< unit number [byte] >< incrementation in µm [float]><crc > The system answers with: <ack><ID><0>crc>

Setting the velocity for the ”step-command“:

- SetStepSpeed ID=0x0158

Syntax: <syn><ID><02>< Gerätenummer [byte] ><Geschwindigkeit [byte]><crc> Answer of the system: <ack><ID><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 to 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

Syntax: <syn><ID><05>< unit number [byte] > < goal position µm [float]><crc > The system answers with: <ack><ID><0>crc >

- GoVariableFastToRelativePos ID=0x004A
- GoVariableSlowToRelativePos ID=0x004B

Syntax: <syn><ID><05>< unit number [byte] ><distance µm [float]><crc> The system answers with: <ack><ID><0>crc>

Setting the positioning velocity:

The positioning velocity can be adjusted in 16 stages to enable fast and slow 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 alternatives are possible, these are, however, dependent on each other.
I.E. previous settings done with SetPositioningVelocityFast are replaced by SetPositioningVelocityFastLinear. The same applies to SetPositioningVelocitySlow because the value of SetPositioningVelocitySlowLinear is replaced.

- SetPositioningSpeedMode ID=0x0191
- SetPositioningVelocityFast ID=0x0144
- SetPositioningVelocitySlow ID=0x018F
- SetPositioningVelocitySlowLinear ID=0x003C
- SetPositioningVelocityFastLinear ID=0x003D

For SetPositioningSpeedMode

Syntax:
<syn><ID><02><Gerätenummer [byte] ><selection [byte] ><crc>

Value: 0 = speed range Slow
1 = speed range Fast

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

To apply SetPositioningVelocityFast and SetPositionVelocitySlow

Syntax:
<syn><ID><02><unit number [byte] >< velocity [byte] >< (crc) >

Value: 0< velocity < 16

To apply SetPositioningVelocitySlowLinear and SetPositioningVelocityFastLinear

Syntax:
<syn><ID><03><unit number[byte] >< velocity [byte] >< (crc) >

Value: 0 < SetPositioningVelocitySlowLinear < 18000
0 < SetPositioningVelocityFastLinear < 3000

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

Approaching a stored position:

One of the 5 previously stored positions can be approached. This position is approached with the velocity set in "positioning velocity".

- GotoPosition ID=0x0110

Inquiry with: <syn><ID><02>< unit number [byte] >< Number[byte]><crc> Value: 0 < number < = 5 The system answers with: <ack><ID><0>crc>

Storing a position:

There is the possibility of storing up to 5 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><02><unit number [byte] >< number [byte] > < crc > Value: 0 < number < = 5 Answer of the system: <ack><ID><0><crc>

Switching axis on/off:

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

Syntax: <syn><ID><01><unit number [byte] ><crc> Answer of the system: <ack><ID><0><crc>

Home:

The Home command stores the actual position of the manipulator and then drives it with the set velocity and the set direction to the Endswitch. In this position the home function can only be ended over home return or over home Abort.

- Home ID = 0x0104

Syntax: <syn><ID><01><unit number [byte]><crc> Answer of the system: <ack><ID><0><crc>

Setting homing velocity stop:

The Home position is approached at variable speeds.
16 different speeds are available.

- SetHomeVelocity ID = 0x0139

Syntax: <syn><ID><02><unit number[byte] >< velocity [byte]><crc> Value: 0< velocity < = 15 Answer of the system: <ack><ID><0><crc>

Setting homing direction:

- SetHomeDirection ID = 0x013C

Syntax: <syn><ID><02><unit number[byte] >< direction [byte]><crc> value: Direction = 0 - > direction of positiv approach Direction = 1 - > direction of negativ approach Answer of the system: <ack><ID><0><crc>

HomeReturn:

The Home Return command can only be executed after a Home command.
The manipulator returns back to the position stored at Home command.

- HomeReturn ID = 0x0022

Syntax: <syn><ID><01><unit number [byte]><crc> Answer of the system: <ack><ID><0><crc>

Home Abort:

The Home Abort - command can only be executed when the manipulator stops and the Home function ends.

- HomeAbort ID= 0x013F

Syntax: <syn><ID><01><unit number [byte]><crc> Answer of the system: <ack><ID><0><crc>

Set position zero:

This command sets the location counter 1 back to 0.

- SetPositionZero ID = 0x00F0

Syntax: <syn><ID><01>< unit number [byte]><crc> Answer of the system: <ack><ID><0><crc>

Go to position zero:

Proceeding to zero-position.

- GotoPositionZero ID = 0x0024

Syntax: <syn><ID><1>< unit number [byte]><crc> Answer of the system: <ack><ID><0><crc>

Resetting the position counter 2:

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

- Set the 2nd counter to zero ID=0x0132

Syntax: <syn><ID><02><unit number[byte]><02><crc> Answer of the system: <ack><ID><0><crc>

Stop:

The "stop command" interrupts the current command and stops the moving axis.

- Stop ID = 0x00FF

Syntax: <syn><ID><01>< unit number [byte]><crc> Answer of the system: <ack><ID><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><01><unit number><crc>

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

Switch on the SlowMove ramp:

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

- SlowMoveRampOn ID = 0x0430

Syntax:
<syn><ID><01><unit number><crc>

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

Setting the ramp length:

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

Important: Too short ramps can lead to step losses.

- Setting the starting - stop ramp length ID=0x003A

Syntax:
<syn><ID><02><unit number [byte]><lenght [byte]><crc>

Value: 0<length<=15

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

Individual queries:

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><01><unit number[byte]><crc> The system answers with: <ack><ID><04><position in µm [float]><crc>

Position inquiry of counter 2:

In addition 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><01>< unit number [byte] ><crc> The system answers with: <ack><ID><04><position in µm [float]><crc>

Positioning selection query:

This Selected velocity range is used to set the velocity (fast/slow) of movement to a position.

- QueryPositioningSpeedMode ID=0x0192

Inquiry with: <syn><ID><01><Gerätenummer [byte] ><crc The system answer with: „QueryPositioningSpeedMode“: <ack><ID><01><Geschwindigkeitsbereich [byte]><crc> Velocity range:00 = Slow 01 = Fast

Positioning velocity inquiry for fast speed linear:

This velocity is used to proceed to a position.

- QueryPositioningVelocityFastLinear ID=0x0160

Inquiry with:
<syn><ID><01>< unit number [byte] ><crc>

Answer to: "QueryPositioningVelocityFastLinear":
<ack><ID><02><velocity[uInt]><crc>

Value: velocity = full steps per second

Positioning velocity inquiry for slow speed linear:

This velocity is used to proceed slowly to a position.

- QueryPositioningVelocitySlowLinear ID=0x0161

Inquiry with:
<syn><ID><01>< unit number [byte] ><crc>

Answer to: "QueryPositioningVelocitySlowLinear" with:
<ack><ID><02>< velocity [uInt]><crc>

Value: velocity = µ-steps per second.

Positioning velocity inquiry for fast speed:

This velocity is used to proceed quickly to a position.

- QueryPositioningVelocityFast ID=0x0143

Inquiry with:
<syn><ID><01>< unit number [byte] ><crc>

Answer to: "QueryPositioningVelocityFast":
<ack><ID><01><velocity[byte]><crc>

Value: 0 < velocity < = 16

Positioning velocity inquiry about slow speed:

This velocity is used to proceed slowly to a position.

- QueryPositioningVelocitySlow ID=0x0190

Inquiry with: <syn><ID><01>< unit number [byte] ><crc> Answer to: "QueryPositioningVelocitySlowLinear" with: <ack><ID><02>< velocity [byte]><crc> Value: 0 < velocity < = 16

Inquiry about driving velocity:

The driving velocity can be inquired separately for slow and fast speeds.

- QueryFastMoveVelocity ID=0x012F
- QuerySlowMoveVelocity ID=0x0130

Inquiry with: <syn><ID><01><unit number [byte] ><crc> The system answers with: <ack><ID><01><velocity [byte]><crc> Value: 0 < velocity < = 16

Inquiry about home velocity:

- QueryHomeVelocity ID = 0x0138

Inquiry with: <syn><ID><01><unit number [byte] ><crc> The system answers with: <ack><ID><01><velocity [byte]><crc> Value: 0 < velocity < = 16

Inquiry about home direction:

- QueryHomeDirection ID =0x013D

Inquiry with: <syn><ID><01><unit number [byte] ><crc> The system answers with: <ack><ID><01><direction [byte]><crc> Value: Positive move direction - > direction=0 Negative move direction - > direction=1

Inquiry about step velocity:

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

- QueryStepSlowVelocity ID=0x0159

Inquiry with: <syn><ID><01><unit number [byte] ><crc> The system answers with: <ack><ID><01><velocity [byte]><crc> Value: 0 < velocity < = 15

Inquiry about proportional mode:

- QueryProportionalMode ID=01A2

Inquiry with: <syn><ID><01><unit number [byte] ><crc> The system answers with: <ack><ID><01><factor [byte]><crc>

Inquiry about whether the output stage is physically present:

- QueryOutputstagePresent ID=0x011F

Inquiry with: <syn><ID><01><unit number[byte]><crc> The system answers with: <ack><ID><01<status[byte]><crc> Value: < status > has the following meaning: - Output stage available : Status=1 - Output stage missing : Status=0

Inquiry about the axis status:

The current power status of the axis is shown.
Important is that while the axis is powerless, all commands other than "switch on axis" and the "version inquiry" are ignored.

- GetPowerStatusFromOutputstage ID = 0x011E

Inquiry with: <syn><ID><01><unit number [byte]><crc> The system answers with: <ack><ID><01><status[byte]><crc> Value: < status > has the following meaning: - Axis switched on : Status=1 - Axis without power : Status=0

Inquiry slow move ramps:

Queries about whether the ramp for the slow movements are on or off.

- QuerySlowMoveRampState ID = 0x0431

Inquiry with:
<syn><ID><01><unit number[byte]><crc>

The system answers with:
<ack><ID><01><status[byte]><crc>

Value: < status > has the following meaning:

The ramp is switched off : 0
Ramp is switched on : All other values

Inquiry about the manipulator pitch:

Shows the pitch for the selected manipulator axis.

- QueryManipulatorPitch ID =0x014D

Inquiry with:
<syn><ID><01><unit number[byte]><crc>

The system answers with:
<ack><ID><01><pitch[byte]><crc>

Value: < pitch > has the following meaning:

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

Inquiry about the type of motor:

Shows the motor used in the queried axis.

- QueryMotortype ID = 0x014B

Inquiry with:
<syn><ID><01><unit number[byte]><crc>

The system answers with:
<ack><ID><01><type[byte]><crc>

Value: < type > has the following meaning:

Type=0 - > motor used: P310	60 full steps per revolution
Type=1 - > motor used: P430	100 full steps per revolution
Type=2 - > motor used: P530	100 full steps per revolution
Type=3 - > motor used: PK223	200 full steps per revolution
Type=4 - > motor used: PK233	200 full steps per revolution
Type=5 - > motor used: ST2018	200 full steps per revolution
Type=6 - > motor used: PK244P	200 full steps per revolution
Type=7 - > motor used: PK244M	400 full steps per revolution

Status inquiry about an output stage:

This inquiry makes it possible to have the most important axes data in one Date frame.

- GetMainStatusFromOutputstage ID0x0120

Inquiry with:
<syn><ID><01><unit number[byte]><crc>

The system answers with:
<ack><ID><08>
<status limit switch[byte]>
<status power to axes[byte]>
<status home[byte]>
<reserved[byte]>
<reolution for single step[byte]>
<motor status[byte]>
<reserved[byte]>
<crc>

Data indicated with < reserved > may contain undefined values.

Status:

Status limit switch	
- No limit switch operates	00
- Limit switch operates in negative direction	01
- Limit switch operates in positive direction	02

Staus axis power	
- axis without power	00
- axis switched on	01

Motor status	
- motor is standing	00
- motor is running	01

Resolution single step	
- number per “GoSingelStepCommand”	0<x<255
- Committed µ-steps	

Home status	
Home non-active	00
Motor driven in direction negative limit-switch	01
Motor driven in direction positive limit-switch	02
Motor stay by limit-switch Drive to the Homeposition was aborted	03

Collection command:

When collecting commands all selected axes can be addressed with a single command. This "Group address" always has 9 byte length. The corresponding “Group address” is determined by a binary matrix. The MSB is transmitted first.
(Exell table: Group instruction address or example in the Appendix)

Switching axis on/off:

- Switching the axis off (dead) BC_OutputStageOff ID=0xA034
- Switching the axis on BC_outputStageOn ID=0xA035

Syntax: <syn><ID><0A><A0><Group address[byte]><crc>

Set position Zero:

This instruction sets the location counter 1 back to 0.

- BC_SetPositionZero ID = 0xA0F0

Syntax: <syn><ID><0A><A0><Group adress[byte]><crc>

Resetting the position counter 2:

Besides the main position counter, a further independent counter is available. This can be set separately to zero.

- BC_SetCounterZero ID=0xA132

Syntax: <syn><ID><0A><A0><Group address[byte]><crc>

Procedure+ucVelocity:

All axes start simultaneously at the same speed. The move command remains active until the stop command is sent, or a limit switch is reached.
Important: No other command either than stop command may be sent during this process.
After stop the motors need 150ms to stop during rapid method (Fast) on the standard ramp.
A change of direction may only be made at standstill.

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

- fast procedure in positive direction BC_FastRunCW ID=0xA012
- fast procedure in negative direction BC_FastRunCCW ID=0xA013
- slow procedure in positive direction BC_SlowRunCW ID=0xA014
- slow procedure in negative direction BC_SlowRunCCW ID=0xA015

Syntax:
<syn><ID><0B><A0><Group address [byte]><Velocity[byte]<crc>

Value: 0<Velocity>=15

Stop:

The "stop command" interrupts the current command and stops the moving axes.

- BC_Abort ID= 0xA0FF

Syntax:
<syn><ID><0A><A0><Group address [byte]><crc>

Go to position zero + ucVelocity:

Proceeding to zero-position.

- BC_GotoPositionNull ID= 0xA024

Syntax:
<syn><ID><0B><A0><Group address [byte]><Velocity[byte]<crc>

Value: 0<Velocuty<=15

Approaching a position + ucVelocity:

One of the 5 previously stored positions is to be approached. This position is approached with the velocity set in "positioning velocity".

- BC_GotoPosition ID=0xA110

Inquiry with:

Syntax:
<syn><ID><0C><A0><Group address[byte]><Position[byte]><Velocity><crc>

Value: 0 < Position < = 5
0 < Velocity < = 15

Storing a position + ucSavePosition+ucVelocity:

It is possible to store and to drive up to 6 positions in each output stage. Once the command has been received by the output stage, the current position is stored.

- BC_SavePosition ID=0xA10A

Syntax:
<syn><ID><0B><A0><Group address [byte]><Position[byte]<crc>

Value: 0 <Position <= 5

StepIncrement + usSpeed + flDistance:

The step increment command moves the selected axes simultaneously at the same speed and distance in the positive direction.

- BC_StepSlowIncr ID=0xA140

Syntax:
<syn><ID><0F><A0><Group address [byte]><Velocity[byte]>
<Distance µm [float]><crc>

StepDecrement + usSpeed + flDistance:

The step increment command moves the selected axes simultaneously at the same speed and distance in the negative direction.

- BC_StepSlowDecr ID=0xA141

Syntax:
<syn><ID><0F><A0><Group address [byte]><Velocity[byte]>
<Distance µm[float]<crc>

Home + ucVelocity:

The Home command stores the actual position of the manipulator and then drives it with the set velocity and the set direction to the limit switch. In this position, the Home function can only be terminated over Home return or Home abort.

- BC_Home ID= 0xA104

Syntax:
<syn><ID><0B><A0><Group address [byte]><Velocity[byte]<crc>

HomeReturn + ucVelocity:

The Home return command can only be executed after a Home command.
The manipulator returns back to the position stored at Home command.

- HomeReturn ID= 0xA022

Syntax:
<syn><ID><0B><A0>><Group address [byte]><Velocity[byte]<crc>

Home Abort:

The Home Abort - command can only be executed when the manipulator stops and the Home function ends.

- HomeAbort ID= 0xA13F

Syntax:
<syn><ID><0A><A0><Group address [byte]><crc>

Group command (4 axes):

Command groups up to 4 axes can be started simultaneously in absolute or relative mode with different positions or distances. The speed can be set individually for each axes.
The device numbers should always be in the range of one unit (master or slave).

- GoVariableFastToAbsolutePosition ID= 0xA048
- GoVariableSlowToAbsolutePosition ID= 0xA049

Syntax:
<syn><ID><15><A0>><ucAdr1><ucAdr2><ucAdr3><ucAdr4><flPos1><flPos2><flPos3><flPos4><crc>

Value: ucAdr 1<unit number[byte]<72
flPos = Position µm[float] 4byte

- GoVariableFastToRelativePosition ID= 0xA04A
- GoVariableSlowToRelativePosition ID= 0xA04B

Syntax:
<syn><ID><15><A0>><ucAdr1><ucAdr2><ucAdr3><ucAdr4><flPos1><flPos2><flPos3><flPos4><crc>

Value: ucAdr 1<unit number[byte]<72
flPos = Distanz µm[float] 4byte

Group questions (4 axes):

For group inquiries position values of up to 4 axes can be scanned simultaneously.
The device numbers should always be in the range of one unit (master or slave).

QueryPosition

- QueryPosition ID= 0xA101

Inquiry with:

Syntax: <syn><ID><05><A0><ucAdr1><ucAdr2><ucAdr3><ucAdr4 [byte]><crc>

Answer of the system:

Syntax: <syn><ID><14><ucAdr1><ucAdr2><ucAdr3><ucAdr4><flPos1><flPos2><flPos3><flPos4><crc> Value: ucAdr 1<Unit number[byte]<72 flPos = Distanz µm[float] 4byte

QueryCounter2

- QueryCounter2 ID= 0xA131

Inquiry with:

Syntax: <syn><ID><05><A0>>< ucAdr1><ucAdr2><ucAdr3><ucAdr4 [byte]><crc>

Answer of the system:

Syntax: <syn><ID><14><ucAdr1><ucAdr2><ucAdr3><ucAdr4><flPos1><flPos2><flPos3><flPos4><crc> Value: ucAdr 1<Unit number[byte]<72 flPos = Distanz µm[float] 4byte

QueryMainState

- QueryMainState ID= 0xA120

Inquiry with:

Syntax:
<syn><ID><05><A0>>< ucAdr1><ucAdr2><ucAdr3><ucAdr4 [byte]><crc>

Answer of the system:

Syntax:
<syn><ID><14><ucAdr1><ucAdr2><ucAdr3><ucAdr4><ucStatus1><ucStatus2><ucStatus3> <ucStatus4><crc> Value: ucAdr 1<Gerätenummer[byte]<72 ucStatus = <status Endschalter [byte] ><status Achsenbestromung [byte] > <statusMotor [byte] ><Auflösung für Einzelschritt [byte] >

Status:

Status limit switch	
- No limit switch operates	00
- Limit switch operates in negative direction	01
- Limit switch operates in positive direction	02

Staus axis power	
- axis without power	00
- axis switched on	01

Motor status	
- motor is standing	00
- motor is running	01

Resolution single step	
- number per “GoSingelStepCommand”	0<x<255
- Committed µ-steps	

Appendix:

Allocation of velocities applies for stages 1-16

This assignment is valid for all adjustable speeds
Motor type 200Vollschritte
Slow velocity for slow procedure; Stepspeed, slow relatively and slow absolutely for position
Fast velocity for fast procedure, fast relatively and fast absolutely for position

Control SM10V1.0 with Keypad TAS7
Motor with 200 fullsteps

Slow velocity		Fast velocity	
stage	revolution rps	stage	revolution rps
1	0,000017	1	0,66
2	0,000040	2	1,73
3	0,000141	3	2,63
4	0,000260	4	3,79
5	0,001280	5	4,67
6	0,002630	6	5,68
7	0,005070	7	6,33
8	0,010200	8	7,81
9	0,025100	9	8,47
10	0,060100	10	9,52
11	0,173000	11	10,42
12	0,332000	12	11,36
13	0,498000	13	12,32
14	0,066400	14	13,23
15	0,996000	15	14,29
16	1,328000	16	15,15

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

Assignment unit number Master / Slave

Physical		Virtual	unit number
Master	1	1	01
Master	2	2	02
Master	3	3	03
Master	4	4	04
Master	5	5	05
Master	6	6	06
Master	7	7	07
Master	8	8	08
Master	9	9	09
Master	10	10	0A
Master	11	11	0B
Master	12	12	0C
Master	13	13	0D
Master	14	14	0E
Master	15	15	0F
Master	16	16	10
Master	17	17	11
Master	18	18	12

Physical		Virtual	unit number
Slave 1	1	19	13
Slave 1	2	20	14
Slave 1	3	21	15
Slave 1	4	22	16
Slave 1	5	23	17
Slave 1	6	24	18
Slave 1	7	25	19
Slave 1	8	26	1A
Slave 1	9	27	1B
Slave 1	10	28	1C
Slave 1	11	29	1D
Slave 1	12	30	1E
Slave 1	13	31	1F
Slave 1	14	32	20
Slave 1	15	33	21
Slave 1	16	34	22
Slave 1	17	35	23
Slave 1	18	36	24

Physical		Virtual	unit number
Slave 2	1	37	25
Slave 2	2	38	26
Slave 2	3	39	27
Slave 2	4	40	28
Slave 2	5	41	29
Slave 2	6	42	2A
Slave 2	7	43	2B
Slave 2	8	44	2C
Slave 2	9	45	2D
Slave 2	10	46	2E
Slave 2	11	47	2F
Slave 2	12	48	30
Slave 2	13	49	31
Slave 2	14	50	32
Slave 2	15	51	33
Slave 2	16	52	34
Slave 2	17	53	35
Slave 2	18	54	36

Physical		Virtual	unit number
Slave 3	1	55	37
Slave 3	2	56	38
Slave 3	3	57	39
Slave 3	4	58	3A
Slave 3	5	59	3B
Slave 3	6	60	3C
Slave 3	7	61	3D
Slave 3	8	62	3E
Slave 3	9	63	3F
Slave 3	10	64	40
Slave 3	11	65	41
Slave 3	12	66	42
Slave 3	13	67	43
Slave 3	14	68	44
Slave 3	15	69	45
Slave 3	16	70	46
Slave 3	17	71	47
Slave 3	18	72	48

Examples Group address by collection command

Group address Dev1 X-axis

			Master																	
Dev7			Dev6			Dev5			Dev4			Dev3			Dev2			Dev1		
z	y	x	z	y	x	z	y	x	z	y	x	z	y	x	z	y	x	z	y	x
16	17	18	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
	0				0				0				0				1			
	0				0				0				0				1			

Group address 000000000000000001

Group address Z-axis Dev 1 to Dev 6

			Master																	
Dev7			Dev6			Dev5			Dev4			Dev3			Dev2			Dev1		
z	y	x	z	y	x	z	y	x	z	y	x	z	y	x	z	y	x	z	y	x
16	17	18	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
	2				4				9				2				4			
	2				4				9				2				4			

Group address 000000000000024924

Group address complete Dev 2 + Dev 3 + Dev 4

			Master																	
Dev7			Dev6			Dev5			Dev4			Dev3			Dev2			Dev1		
z	y	x	z	y	x	z	y	x	z	y	x	z	y	x	z	y	x	z	y	x
16	17	18	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0
1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
	0				0				15				15				8			
	0				0				F				F				8			

Group address 0000000000000000FF8

ID numbers

ID	description	page
0x0012	Fast procedur in positive direction	4
0x0013	Fast procedur in negative direction	4
0x0014	Slow procedur in positive direction	4
0x0015	Slow procedur in negative direction	4
0x0022	Home Return	12
0x0024	Goto Position Null	13
0x0034	Switching the axis off	10
0x0035	Switching the axis on	10
0x003A	setting the starting - stop ramp length	14
0x003C	Set Positioning velocity slow linear	9
0x003D	Set Positioning velocity fast linear	9
0x0048	Go Variable Fast To Absolute Position	8
0x0049	Go Variable Slow To Absolute Position	8
0x004A	Go Variable Fast To Relative Position	8
0x004B	Go Variable Slow To Relative Position	8
0x00F0	Set position to Zero	12
0x00FF	Stop	13
0x0101	Inquiry of Position	15
0x0104	Home	11
0x0110	Goto Position	10
0x010A	Save Position	10
0x011E	Get power status from output stage	19
0x011F	Query output stage present	19
0x0120	Get Main status from output stage	21
0x012F	Query fast move velocity	17
0x0130	Query slow move velocity	17
0x0131	Inquiry of 2nd counter	15
0x0132	Set the 2nd counter to Zero	13
0x0134	Set Velocity for fast procedure	4
0x0135	Set Velocity for slow procedure	4
0x0138	Query Home velocity	17
0x0139	Set Home Velocity	11
0x013C	Set Home Direction	11
0x013D	Query home direction	18
0x013F	Home-Abort	12
0x0140	StepIncrement	7
0x0141	StepDecrement	7
0x0143	QueryPositioningVelocityFast	16
0x0144	SetPositioningVelocityFast	9

0x0146	SetHandwheelResolution	5
0x0147	Go Single Step	5
0x014B	QueryMotorType	21
0x014D	Query manipulator pitch	20
0x0158	SetStepSpeed	8
0x0159	QuereyStepSlowVelocity	18
0x0160	QueryPositioningVelocityFastLinear	16
0x0161	QueryPositioningVelocitySlowLinear	16
0x018F	SetPositioningVelocitySlow	9
0x0190	QueryPositioningVelocitySlow	17
0x0191	SetPositioningSpeedMode	9
0x0192	QueryPositioningSpeedMode	15
0x019F	SetProportionalFaktor	6
0x01A2	QueryProportionalMode	18
0x01E8	GoTrackballMode	6
0x042F	SlowMoveRampOff	14
0x0430	SlowMoveRampOn	14
0x0431	QuerySlowMoveRampState	20
0x044F	Step Slow Distance	7
0xA012	BC_FastRunCW	24
0xA013	BC_FastRunCCW	24
0xA014	BC_SlowRunCW	24
0xA015	BC_SlowRunCCW	24
0xA022	BC_HomeReturn	26
0xA024	BC_GotoPositionZero	24
0xA034	BC_OutputStageOff	23
0xA035	BC_OutputStageOn	23
0xA048	BC_GoVariableFastToAbsolutePosition	27
0xA049	BC_GoVariableSlowToAbsolutePosition	27
0xA04A	BC_GoVariableFastToRelativePosition	27
0xA04B	BC_GoVariableSlowToRelativePosition	27
0xA0F0	BC_SetPositionZero	23
0xA0FF	BC_Abort	24
0xA101	BC_QueryPositionFromOutputStage	28
0xA104	BC_Home	26
0xA110	BC_GotoPosition	25
0xA10A	BC_SavePosition	25
0xA120	BC_QueryMainStateFromOutputStage	29
0xA131	BC_QueryCounter2FromOutputStage	28
0xA132	BC_SetCounterZero	23
0xA13F	BC_HomeAbort	26
0xA140	BC_StepSlowIncr	25
0xA141	BC_StepSlowDecr	26

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  crcHigh and low the checksum is returned.

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!=zero)
*crcHigh=(unsigned char)(crc>>8);
if(crcLow!=zero)
*crcLow=(unsigned char)crc;
}
```

History:

Luigs & Neumann
Feinmechanik + Elektrotechnik GmbH
Boschstrasse 19 / D-40880 Ratingen / Germany
+49 2102 94700-0 / info@luigs-neumann.com
www.luigs-neumann.com