



## SOPAS Communication Interface Description



**ML20**

Version: 1.110  
Copyright SICK AG, 2015-2-26



## Version History

Version	Description	Date
1.108	Initial version TCP/IP.	2012-11-15
1.110	NEW parameter udiFrameResolution.	2015-26-02
	NEW parameters uiVerticalBlankingTop, uiVerticalBlankingBottom for vertical blanking.	
	CHANGED parameter/return format of getPatchData, setPatchData. See section 2.2 Compatibility and migrating client software for details.	

## Copyright

Copyright © 2015

SICK AG

Erwin-Sick-Str. 1

79183 Waldkirch

Germany

## Document Info

This document was generated with SOPAS Documentation Generator 3.0.1.0001R.

Generation Date: 26.02.2015 at 14:50:57



# Table Of Contents

<b>1. General</b>	1
1.1. Introduction	1
1.2. User Level	1
1.3. Methods	1
1.4. Events	1
1.5. Datatypes	1
<b>2. Interfaces</b>	3
2.1. TCP/IP Cola-B Dataframe	3
2.2. Compatibility and migrating client software	6
2.3. Group: V100	7
2.3.1. Variable: DeviceId	7
2.3.2. Variable: SOPASVersion	8
2.3.3. Variable: LocationName	9
2.3.4. Variable: SerialNumber	10
2.3.5. Variable: FirmwareVersion	11
2.3.6. Variable: SopasInfo	12
2.3.7. Method: SetAccessMode	15
2.3.8. Method: GetAccessMode	16
2.3.9. Method: Run	17
2.3.10. Method: GetDescription	18
2.4. ML20 device configuration	21
2.4.1. Variable: udiEncoderResolution	21
2.4.2. Variable: udiFrameResolution	23
2.4.3. Variable: udiUserConfig	24
2.4.4. Method: accessConfigMemory	26
2.4.5. Ethernet	28
2.4.5.1. Variable: udiIpAddress	28
2.4.5.2. Variable: udiSubnetMask	29
2.4.5.3. Variable: udiGatewayAddress	31
2.5. Teach	33
2.5.1. Teach-in and configuration	34
2.5.1.1. Method: startTeach	34
2.5.1.2. Method: stopTeach	35
2.5.1.3. Method: triggerTeach	36
2.5.1.4. Method: cancelTeach	37
2.5.1.5. Variable: eTeachDirectionSelect	38
2.5.1.6. Variable: udiTrigTeachLength	39



2.5.2. Teach result data .....	41
2.5.2.1. Variable: eTeachResult .....	42
2.5.2.2. Variable: udiDisplayTeachQuality .....	43
2.5.2.3. Variable: udiCurrentTeachLength .....	44
2.5.2.4. Variable: teCurrentTeachDirection .....	45
2.5.2.5. Method: getPatchData .....	46
2.5.2.6. Method: setPatchData .....	48
2.5.2.7. Method: readTeachData .....	50
2.5.2.8. Method: applyTeachData .....	51
2.5.3. Blanking .....	53
2.5.3.1. Method: recomputeTeach .....	54
2.5.3.2. Variable: sBlankingWindow1 .....	55
2.5.3.3. Variable: sBlankingWindow2 .....	56
2.5.3.4. Variable: uiVerticalBlankingTop .....	57
2.5.3.5. Variable: uiVerticalBlankingBottom .....	59
2.6. Run .....	61
2.6.1. Variable: eDeviceOperatingState .....	61
2.6.2. Variable: udiDisplayRunQuality .....	62
2.6.3. Variable: udiActualFormatLength .....	63
2.6.4. Variable: diQOffset .....	64
2.7. Diagnostics .....	66
2.7.1. Variable: eErrorCode .....	66
2.7.2. Method: getEncoderPosition .....	67
2.7.3. Image handling .....	69
2.7.3.1. Method: getImage .....	70
2.7.3.2. Method: acquireRunImage .....	71
2.7.3.3. Variable: udiImageSize .....	72
2.7.3.4. Variable: sPixelFormat .....	73
2.7.3.5. Variable: bHasTeachImage .....	74
2.7.3.6. Variable: bHasRunImage .....	75
<b>3. User types .....</b>	<b>77</b>
3.1. Type: teEncoderDirection .....	77
3.2. Type: ImageDataStruct .....	77
3.3. Type: BlankingWinStruct .....	78
<b>Index .....</b>	<b>79</b>



# 1. General

## 1.1. Introduction

This document describes the functional interfaces of the ML20 device, 1.110. The ML20 device is a SOPAS device. SOPAS devices have Variables, Methods and Events. Variables can always be read and in some cases be written by user.

## 1.2. User Level

Whether a Variable can be written by user depends on the least user level. Defined user levels are:

ID	Name
0	Always (Run)
1	Operator
2	Maintenance
3	Authorized Client
4	Service
5	SICK Service
6	Production
7	Developer

Table 1: User Levels

## 1.3. Methods

Methods can be invoked by using certain parameters. The method will return with a structure of one or more return values. If a Method can be invoked depends as well on the least user level (see above).

## 1.4. Events

Events can be registered and will then be fired by the device to the registered client. Most Events have parameters which are the data coming with the Event.

## 1.5. Datatypes

All items of the interface have certain data elements. These are the Variables itself, the parameters of Methods and Events and the return values of the Methods.

The structure of the data elements can be one of the following BasicType(s), Structures or Arrays.

Basic Type

Name	Description	Range of values
Bool	boolean	True(1), False(0)
USInt	unsigned short (8 bit)	(0..255)
UInt	unsigned int (16 bit)	(0..65535)
UDInt	unsigned double int (32 bit)	(0..4294967295)
ULInt	unsigned long int (64 bit)	(0..18446744073709551616)
SInt	signed short (8 bit)	(-128..127)
Int	signed int (16 bit)	(-32768..32767)



Name	Description	Range of values
DInt	signed double int (32 bit)	(-2147483648..2147483647)
LInt	signed long int (64 bit)	(-9223372036854775808..9223372036854775807)
Real	IEEE-754 single precision (32 bit) (float)	See specification in IEEE-754
LReal	IEEE-754 single precision (64 bit) (double)	See specification in IEEE-754
Enum8	short enumeration (8 bit)	certain values defined in a list of choices (0-255)
Enum16	short enumeration (16 bit)	certain values defined in a list of choices (0-65535)
String	array of visible characters (array of 8 bit)	a character = an USInt with values between 0x20..0xFF
FlexString	array of visible characters with preceding current length (UInt length) (array of 8 bit)	See description of String and FlexArray
Byte	bitset definition (8 bit). Detailed specification of bits UInt1..UInt16 = UInt (1..16 bit) Int1..Int16 = Int (1..16 bit) Enum1..Enum16 = Enum16 (1..16 bit) Bool = Bool (1 bit)	value is transferred as an array of USInt. See "XByte Serialisation" document for further details on bit ordering
Word	bitset definition (16 bit), see description of Byte	value is transferred as an array of USInt see "XByte Serialisation" document for further details on bit ordering.
DWord	bitset definition (32 bit), see description of Byte	value is transferred as an array of USInt see "XByte Serialisation" document for further details on bit ordering.
LWord	bitset definition (64 bit), see description of Byte	value is transferred as an array of USInt see "XByte Serialisation" document for further details on bit ordering.
XByte	bitset definition (8,16,24,32,... bit) see description of Byte	value is transferred as an array of USInt see "XByte Serialisation" document for further details on bit ordering.
SCont	bitset definition (8 bit). Detailed specification of bits UInt1..UInt16 = UInt (1..16 bit) Int1..Int16 = Int (1..16 bit) Enum1..Enum16 = Enum16 (1..16 bit) Bool = Bool (1 bit)	value is transferred as USInt.
Cont	bitset definition (16 bit), see description of SCont	value is transferred as UInt.
DCont	bitset definition (32 bit), see description of SCont	value is transferred as UInt.
LCont	bitset definition (64 bit), see description of SCont	value is transferred as UInt.

Table 2: Basic Datatypes

## Struct

A structure is a sequence of further types. These types can be of a BasicType, Structs again or an Array.

## Array

An Array is a repetition of a type. The length of the array is defined with each Array. The types can be of a BasicType, a Struct or an Array again (n- dimensional).

## Flex Array

A FlexArray is a repetition of a type with a variable length. The maximum length of the array is defined with each FlexArray. The current length of the FlexArray is transferred as a UInt preceding the Array itself. The types can be of a BasicType, a Struct or an Array again (n- dimensional).

## 2. Interfaces

### 2.1. TCP/IP Cola-B Dataframe

The ML20 from firmwareversion V6.01.009.xxxx supports the SICK CoLa-B (Command Language B) protocol via TCP/IP at port 2112. This protocol follows a strict request-response scheme, ie. the client PC sends a single request telegram to the host ML20 which always acknowledges the request with a single telegram.

#### Description rules

Named values (characters, sequences) are printed in sharp brackets: <> .

Numbers (decimal or hexadecimal (0x...)) in sharp brackets represent one single byte with the respective Value,e.g. <0x10>.

Names in curly brackets are placeholder for parts of the telegram with different lengths, e.g. {SVIdx} .

Numbers are always decimal number unless they have a prefix 0x or are declared otherwise.

Numbers starting with 0x are hexadecimal numbers.

#### CoLa-B commands

Following CoLa-B commands are implemented:

Request by client (PC)	Response by server (ML20)	Command
sRI{SVIdx}	sRA{SVIdx}{variable value}	Read variable identified by index <i>SVIdx</i>
sWI{SVIdx}	sWA{SVIdx}	Write variable identified by index <i>SVIdx</i>
sMI{MIdx}{parameters}	sMA{MIdx}{return values}	Invoke method identified by index <i>MIdx</i>
	sFA{error code}	Alternatively, the device can respond to every command with an error code.

Figure 1. CoLa-B command and respond format.

The request and acknowledge command are ascii encoded while the indices are encoded as unsigned 16-Bit MSB first values. Encoding of values and parameters is described in the previous chapter General. Variable and method indices and the variable and parameter types are listed in the following chapters of this document.

#### CoLa-B Errorcodes

Following errorcode may appear after requesting a a CoLa-B command in an sFA-response:

Error code	Description
0	No error
1	Method: access denied
2	Method: unknown index
3	Variable: unknown index
4	Local condition failed (temporarily not available)
5	Invalid data
10	Variable: write access denied
others >0	Other error

Figure 2. CoLa-B errorcodes.

## Telegram structure by example

This section describes an example TCP/IP-frame for requesting the content of variable *FirmwareVersion* (index 4) of the ML20.

Byte Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Content (as hex-Value)	02	02	02	02	00	00	00	05	73	52	49	00	04	6c	
Description	STx framing start				Length = 5 bytes Length of CoLa-B data (without checksum byte)				„sRI“ ASCII-command read Variable			Var.-Idx = 4		Checksum = 6c	
Layer	CoLa framing start								CoLa-B command Valid indices see following chapters.						CoLa framing end

Figure 3. TCP/IP request frame. Example: query variable firmware version.

The telegram begins with the CoLa framing start block (yellow), which is constructed in the following way:

The STx-Sequence is always 0x02020202. Data length is the 32-bit length value (encoded as MSB first) of the CoLa-B content in bytes. The checksum field is not counted. The length of the CoLa-B content in this example is 5 bytes.

After the CoLa framing start block the CoLa-B command block (white) follows. This example shows the CoLa-B command *Read Variable* (ASCII-encoded „sRI“) of variable *FirmwareVersion* at index 4.

The telegram ends with the CoLa framing end block (yellow) containing only the checksum byte. The checksum byte is computed by XOR'ing all bytes in the CoLa-B command block, here Bytes 9 to 13. In this example the checksum is 0x6c.



The ML20 could answer with the following response frame:

Byte Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
Content (as hex-Value)	02	02	02	02	00	00	00	15	73	52	41	00	04	00	0e	44	35	
Description	STx Framing Start				Length = 21 bytes Length of CoLa-B data (without checksum byte)				„sRA“ ASCII-answer read acknowledge				Var.-Idx = 4		Flexstring-length = 14			
Layer	CoLa framing start								CoLa-B command Valid indices see following chapters.						CoLa-B value			

Byte Index	18	19	20	21	22	23	24	25	26	27	28	29	30				
Content (as hex-Value)	2e	31	33	2e	30	30	38	2e	32	37	32	32	0a				
Description	Variable content, here: Flexstring, ascii: „D5.13.008.2722“												Checksum = 0a				
Layer	CoLa-B value (continued)												CoLa framing end				

Figure 4. TCP/IP response frame. Example: response to query variable firmware version.

The CoLa framing start and end fields are constructed as in the request frame. The value field of the response must be interpreted according to the datatype. In this example the datatype of variable *FirmwareVersion* is a *Flexstring*.

## 2.2. Compatibility and migrating client software

In order to handle future interface versions of the sensor software it is **highly recommended** to evaluate the variable *DeviceIdent.Version* (see section 2.3.1) which contains the interface version. This should be done as a first step immediately after establishing ethernet connection to the sensor. Only parameters and communication format defined by the appropriate interface version document should be used by the communicating client.

The following table shows all released ML20 software versions and the corresponding supported interface versions:

ML20 software version	SOPAS communication interface version
V6.01.009.xxxx	1.108
V6.05.010.3xxx	1.110

Figure 5. Software and interface versions

### Migrating from Interface version V1.108 to V1.110

In this interface version V1.110 the parameters *udiFrameResolution*, *uiVerticalBlankingTop* and *uiVerticalBlankingBottom* have been added. The default settings of these parameters correspond to the configuration in device software with interface version V1.108. Thus, when these parameters are not changed beyond their defaults the sensor behaves compatible in relation to the function of these parameters.

Due to an improvement of the teach algorithm, the teach data format has changed.

Therefore parameter 5 (threshold) has been added to the methods *getPatchData* and *setPatchData* for reading and writing teach data. Both methods have been moved to new method indices 22 and 23.

**For backward compatibility the old method calls from interface version V1.108 have been preserved** at the indices 14 and 15 in order to support communication software written for V1.108 devices.

Yet, using the V1.108 method calls will not make use of some new algorithms to deliver improved detection performance with the ML20. They are not recommended for use in new ML20 communication interface development. Find detailed information about the V1.108 methods in the corresponding version of the SOPAS communication interface description. For immediate recommendation on converting teach data between both interface versions see section 2.5.2 .

In order to handle features of future interface versions please consider the recommendation at the head of this chapter to consider the variable *DeviceIdent.Version* (see section 2.3.1) which contains the interface version and extend your client software to support interface features depending this variable.



## 2.3. Group: V100

### 2.3.1. Variable: DeviceId

The following section contains a detailed description of the variable DeviceId.

#### Variable Overview

Variable Name	Description
DeviceId	Unique Identification of device

Sopas Synchronisation	Variable is not relevant for synchronisation with SOPAS ET.
Sopas Index	0 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

Struct	
Name	
FlexString	
Length	0..4
Initialisation	ML20
Version	
FlexString	
Length	0..5
Initialisation	1.110

#### Variable Telegram Syntax

Read Variable:				
sRI 0				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	0	String	1	Unique Identification of device

Read Variable Response:				
sRA 0 <Name> <Version>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	0	String	1	Unique Identification of device
Variable Data 1	Name	FlexString	4	
Variable Data 2	Version	FlexString	5	



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 00 68	.....sRI..h
Read Variable Response:	02 02 02 02 00 00 00 12 73 52 41 00 00 04 4D 4C 32 30 00 05 31 2E 31 31 30 4D	.....sRA....M L20..1.110M

### 2.3.2. Variable: SOPASVersion

The following section contains a detailed description of the variable SOPASVersion.

#### Variable Overview

Variable Name	Description
SOPASVersion	Version of SOPAS runtime (SOPAS DCD)

Sopas Synchronisation	Variable is not relevant for synchronisation with SOPAS ET.
Sopas Index	1 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

Struct	
Version	
USInt	
Value Range	0..255
Initialisation	2
Release	
USInt	
Value Range	0..255
Initialisation	48
Build	
UInt	
Value Range	0..65535
Initialisation	9



## Variable Telegram Syntax

Read Variable:				
sRI 1				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	1	String	1	Version of SOPAS runtime (SOPAS DCD)

Read Variable Response:				
sRA 1 <Version> <Release> <Build>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	1	String	1	Version of SOPAS runtime (SOPAS DCD)
Variable Data 1	Version	USInt	1	
Variable Data 2	Release	USInt	1	
Variable Data 3	Build	UInt	2	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 01 69	.....sRI..i
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 01 02 30 00 09 5A	.....sRA...0. ·Z

### 2.3.3. Variable: LocationName

The following section contains a detailed description of the variable LocationName.

#### Variable Overview

Variable Name	Description
LocationName	Location of Device (set by user)

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	2 (fixed)
Storage	Variable is stored in EEPROM
Read-Access	Always
Write-Access	Maintenance, AuthorizedClient, Service, SickService, Production, Developer

FlexString	
Length	0..16
Initialisation	No location



## Variable Telegram Syntax

Read Variable:				
sRI 2				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	2	String	1	Location of Device (set by user)

Read Variable Response:				
sRA 2 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	2	String	1	Location of Device (set by user)
Variable Data	data	FlexString	16	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 02 6A	.....sRI..j
Read Variable Response:	02 02 02 02 00 00 00 12 73 52 41 00 02 00 0B 4E 6F 20 6C 6F 63 61 74 69 6F 6E 75	.....sRA....N o locationu

### 2.3.4. Variable: SerialNumber

The following section contains a detailed description of the variable SerialNumber.

#### Variable Overview

Variable Name	Description
SerialNumber	serial number of device

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	3 (fixed)
Storage	Variable is stored in EEPROM
Read-Access	Always
Write-Access	Production, Developer

FlexString	
Length	0..12
Initialisation	1234567890AB



## Variable Telegram Syntax

Read Variable:				
sRI 3				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	3	String	1	serial number of device

Read Variable Response:				
sRA 3 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	3	String	1	serial number of device
Variable Data	data	FlexString	12	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 03 6B	.....sRI..k
Read Variable Response:	02 02 02 02 00 00 00 13 73 52 41 00 03 00 0C 31 32 33 34 35 36 37 38 39 30 41 42 6D	.....sRA....1 234567890ABm

### 2.3.5. Variable: FirmwareVersion

The following section contains a detailed description of the variable FirmwareVersion.

#### Variable Overview

Variable Name	Description
FirmwareVersion	Version of the application software

Sopas Synchronisation	Variable is not relevant for synchronisation with SOPAS ET.
Sopas Index	4 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

FlexString	
Length	0..15
Initialisation	6.03.009.xxxxxx



## Variable Telegram Syntax

Read Variable:				
sRI 4				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	4	String	1	Version of the application software

Read Variable Response:				
sRA 4 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	4	String	1	Version of the application software
Variable Data	data	FlexString	15	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 04 6C	.....sRI..1
Read Variable Response:	02 02 02 02 00 00 00 16 73 52 41 00 04 00 0F 36 2E 30 33 2E 30 30 39 2E 78 78 78 78 78 78 49	.....sRA....6 .03.009.xxxxxxI

### 2.3.6. Variable: SopasInfo

The following section contains a detailed description of the variable SopasInfo.

#### Variable Overview

Variable Name	
SopasInfo	
Sopas Synchronisation	Variable is not relevant for synchronisation with SOPAS ET.
Sopas Index	6 (fixed)
Read-Access	Always
Write-Access	No! (readonly)





DWord				
Bit Length		32		
CIDUploadSupported				
0.0	Bool			
	Value Range		False, True	
	Initialisation		True	
ShortUDDUploadSupported				
0.1	Bool			
	Value Range		False, True	
	Initialisation		True	
PMDUploadSupported				
0.2	Bool			
	Value Range		False, True	
	Initialisation		True	
LocationNameExists				
0.3	Bool			
	Value Range		False, True	
	Initialisation		True	
SegmentSize				
0.4	...	Enum4		
		Value	Name	Description
0.7		0	SegSize4	
		1	SegSize8	
		2	SegSize16	
		3	SegSize64	
		4	SegSize256	
		5	SegSize1024	
		6	SegSize4096	
		7	SegSize16384	
SupportsEventPolling				
1.0	Bool			
	Value Range		False, True	
	Initialisation		True	
hasProclIndex				
1.1	Bool			
	Value Range		False, True	
	Initialisation		False	
CIDChecksumProvided				
1.2	Bool			
	Value Range		False, True	
	Initialisation		True	
CheckPasswordProvided				
1.3	Bool			
	Value Range		False, True	
	Initialisation		True	



DWord				
hubFunctionality				
1.4	Enum2			
...	Default Value		0	
1.5		Value	Name	Description
		0	noSubDevices	
		1	subDevicesWithDetails	
		2	subDevicesWithMaxAddr	
		3	subDevicesExtended	
JarUploadSupported				
1.6	Bool			
	Value Range		False, True	
	Initialisation		False	
hasFirmwareDownloadAlgorithm				
1.7	Bool			
	Value Range		False, True	
	Initialisation		False	
SimultaneousMethodsSupport				
2.0	Bool			
	Value Range		False, True	
	Initialisation		False	
HashValueSupport				
2.1	Bool			
	Value Range		False, True	
	Initialisation		False	
HasAdditionalTimeout				
2.2	Bool			
	Value Range		False, True	
	Initialisation		False	
BulkTransferSupported				
2.3	Bool			
	Value Range		False, True	
	Initialisation		False	

## Variable Telegram Syntax

Read Variable:				
sRI 6				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	6	String	1	

Read Variable Response:				
sRA 6 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	6	String	1	
Variable Data	data	DWord	4	



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 06 6E	.....sRI..n
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 06 08 0D 00 00 63	.....sRA..... ..c

### 2.3.7. Method: SetAccessMode

The following section contains a detailed description of the method SetAccessMode.

#### Method Overview

Method Name	Description
SetAccessMode	Change operation mode returns True, if successful

Sopas Index	0 (fixed)
Invocation Access	Always

Parameters	
NewMode	
SInt	
Value Range	-128..127
Password	hash value
UDInt	
Value Range	0..4294967295

Return Values	
success	
Bool	
Value Range	False, True
Initialisation	False



## Method Telegram Syntax

Method Invocation:				
sMN SetAccessMode <NewMode> <Password>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMN	String	3	Request (SOPAS Method by Name)
Command	SetAccessMode	String	13	Change operation mode returns True, if successful
Parameter 1	NewMode	SInt	1	
Parameter 2	Password	UDInt	4	hash value

Method Return Value:				
sAN SetAccessMode <success>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAN	String	3	Result (SOPAS Method Result)
Command	SetAccessMode	String	13	Change operation mode returns True, if successful
Return Value 1	success	Bool	1	

## Method Telegram Examples

Example: Default Values	
Method telegram examples with parameter data and return value data set to default values.	
Method Invocation:	[STX]sMN SetAccessMode 0 0[ETX]
Method Return Value:	[STX]sAN SetAccessMode 0[ETX]

## 2.3.8. Method: GetAccessMode

The following section contains a detailed description of the method GetAccessMode.

### Method Overview

Method Name	Description
GetAccessMode	returns actual operation mode

Sopas Index	1 (fixed)
Invocation Access	Always

Return Values	
opmode	
SInt	
Value Range	-128..127



## Method Telegram Syntax

Method Invocation:				
sMN GetAccessMode				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMN	String	3	Request (SOPAS Method by Name)
Command	GetAccessMode	String	13	returns actual operation mode

Method Return Value:				
sAN GetAccessMode <opmode>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAN	String	3	Result (SOPAS Method Result)
Command	GetAccessMode	String	13	returns actual operation mode
Return Value 1	opmode	SInt	1	

## Method Telegram Examples

Example: Default Values	
Method telegram examples with parameter data and return value data set to default values.	
Method Invocation:	[STX]sMN GetAccessMode[ETX]
Method Return Value:	[STX]sAN GetAccessMode 0[ETX]

### 2.3.9. Method: Run

The following section contains a detailed description of the method Run.

#### Method Overview

Method Name	Description
Run	Change operation mode to "Run"

Sopas Index	2 (fixed)
Invocation Access	Always

Return Values	
success	
<b>Bool</b>	
Value Range	False, True
Initialisation	False



## Method Telegram Syntax

Method Invocation:				
sMN Run				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMN	String	3	Request (SOPAS Method by Name)
Command	Run	String	3	Change operation mode to "Run"

Method Return Value:				
sAN Run <success>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAN	String	3	Result (SOPAS Method Result)
Command	Run	String	3	Change operation mode to "Run"
Return Value 1	success	Bool	1	

## Method Telegram Examples

Example: Default Values	
Method telegram examples with parameter data and return value data set to default values.	
Method Invocation:	[STX]sMN Run[ETX]
Method Return Value:	[STX]sAN Run 0[ETX]

### 2.3.10. Method: GetDescription

The following section contains a detailed description of the method GetDescription.

#### Method Overview

Method Name	
GetDescription	
Sopas Index	4 (fixed)
Invocation Access	Always



Parameters			
eType			
Enum8			
Default Value		CID	
	Value	Name	Description
	1	CID	
	2	ShortUDD	
	3	PMD	
	4	Jar	
	5	CidPMD	
	6	Eip2PMD	
	7	ChInfo	
	8	AVC	
	9	Profibus	
	10	Profibus2	
	11	CanOpen	
uiSegmentNumber			
UInt			
Value Range		0..65535	

Return Values					
eState					
		Enum8			
			Value	Name	Description
		0		TypeNotSupported	
		1		SegmentOutOfRange	
		2		FirstSegment	
		3		NormalSegment	
		4		LastSegment	
uiSegmentNumber					
		UInt			
		Value Range	0..65535		
aByteStream					
		Array			
		Length	0..1		
		USInt			
		Value Range	0..255		



## Method Telegram Syntax

Method Invocation:				
sMN GetDescription <eType> <uiSegmentNumber>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMN	String	3	Request (SOPAS Method by Name)
Command	GetDescription	String	14	
Parameter 1	eType	Enum8	1	
Parameter 2	uiSegmentNumber	UInt	2	

Method Return Value:				
sAN GetDescription <eState> <uiSegmentNumber> <aByteArray>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAN	String	3	Result (SOPAS Method Result)
Command	GetDescription	String	14	
Return Value 1	eState	Enum8	1	
Return Value 2	uiSegmentNumber	UInt	2	
Return Value 3	aByteArray	Array	1	

## Method Telegram Examples

Example: Default Values	
Method telegram examples with parameter data and return value data set to default values.	
Method Invocation:	[STX]sMN GetDescription 1 0[ETX]
Method Return Value:	[STX]sAN GetDescription 0 0 0[ETX]





## 2.4. ML20 device configuration

The following variables must be configured properly for basic operation

### *udiEncoderResolution*

Set this to the encoder resolution of your system.

### *udiFrameResolution*

Set this to the required switching point resolution of your system.

### *udiUserConfig*

Set individual bits appropriately.

To permanently save these and other settings call the method *accessConfigMemory*.

### 2.4.1. Variable: udiEncoderResolution

The following section contains a detailed description of the variable *udiEncoderResolution*.

#### Variable Overview

Variable Name	Description
udiEncoderResolution	Encoder line distance in [µm].

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	29 (fixed)
Read-Access	Always
Write-Access	Always

UDInt	
Value Range	100..400
Initialisation	100



## Variable Telegram Syntax

Read Variable:				
sRI 29				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	29	String	2	Encoder line distance in [µm].

Read Variable Response:				
sRA 29 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	29	String	2	Encoder line distance in [µm].
Variable Data	data	UDInt	4	

Write Variable:				
sWI 29 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	29	String	2	Encoder line distance in [µm].
Variable Data	data	UDInt	4	

Write Variable Response:				
sWA 29				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	29	String	2	Encoder line distance in [µm].

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 1D 75	.....sRI..u
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 1D 00 00 00 64 19	.....sRA.....d.
Write Variable:	02 02 02 02 00 00 00 09 73 57 49 00 1D 00 00 00 64 14	.....sWI.....d.
Write Variable Response:	02 02 02 02 00 00 00 05 73 57 41 00 1D 78	.....sWA..x

## 2.4.2. Variable: udiFrameResolution

This parameter configures the switching point resolution ie. the frame resolution of the sensor.

By default this is 600 µm. For a higher resolution this value may be configured down to 100µm. Below this value there will be no reasonable improvement due to optical limitations.

### Note

1. The applied internal frame resolution is rounded towards a multiple of a quarter of the encoder resolution. E.g. with an encoder resolution of 400µm the frame resolution will be rounded as a multiple of 100µm. When set to 150µm this results in rounding to 200µm as internally applied frame resolution.
2. The internally applied frame resolution after rounding may always be read back from the variable *sPixelFormat.x*.
2. The maximum teachlength reduces linear with resolution, ie. 1m at 600µm, 0.5m at 300µm, etc.
3. The minimum teachlength reduces linear with resolution, ie. 24mm at 600µm, 12mm at 300µm, etc.
4. The maximum velocity reduces linear with resolution, ie. 7m/s at 600µm, 3.5m/s at 300µm, etc.

### Variable Overview

Variable Name	Description
udiFrameResolution	Frame resolution in [µm].

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	97 (fixed)
Read-Access	Always
Write-Access	Always

UDInt	
Value Range	1..1000
Initialisation	600

### Variable Telegram Syntax

Read Variable:				
sRI 97				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	97	String	2	Frame resolution in [µm].

Read Variable Response:				
sRA 97 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	97	String	2	Frame resolution in [µm].
Variable Data	data	UDInt	4	



Write Variable:				
sWI 97 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	97	String	2	Frame resolution in [µm].
Variable Data	data	UDInt	4	

Write Variable Response:				
sWA 97				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	97	String	2	Frame resolution in [µm].

## Variable Telegram Examples

Example: Default Values				
Variable telegram examples with data set to default values.				
Read Variable:	02 02 02 02 00 00 00 05	73 52 49 00 61 09	.....sRI·a·	
Read Variable Response:	02 02 02 02 00 00 00 09 58 5B	73 52 41 00 61 00 00 02	.....sRA·a··· X[	
Write Variable:	02 02 02 02 00 00 00 09 58 56	73 57 49 00 61 00 00 02	.....sWI·a··· XV	
Write Variable Response:	02 02 02 02 00 00 00 05	73 57 41 00 61 04	.....sWA·a·	

### 2.4.3. Variable: udiUserConfig

The following section contains a detailed description of the variable udiUserConfig.

#### Variable Overview

Variable Name	Description
udiUserConfig	bit 2 error handling on/off bit 1 trigger teach mode bit 0 encoder type: TTL(differential) =0, HTL (single ended) = 1

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	20 (fixed)
Read-Access	Always
Write-Access	Always

UDInt	
Value Range	0..4294967295
Initialisation	0x0



## Variable Telegram Syntax

Read Variable:				
sRI 20				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	20	String	2	bit 2 error handling on/off bit 1 trigger teach mode bit 0 encoder type: TTL(differential) =0, HTL (single ended) = 1

Read Variable Response:				
sRA 20 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	20	String	2	bit 2 error handling on/off bit 1 trigger teach mode bit 0 encoder type: TTL(differential) =0, HTL (single ended) = 1
Variable Data	data	UDInt	4	

Write Variable:				
sWI 20 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	20	String	2	bit 2 error handling on/off bit 1 trigger teach mode bit 0 encoder type: TTL(differential) =0, HTL (single ended) = 1
Variable Data	data	UDInt	4	

Write Variable Response:				
sWA 20				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	20	String	2	bit 2 error handling on/off bit 1 trigger teach mode bit 0 encoder type: TTL(differential) =0, HTL (single ended) = 1

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 14 7C	.....sRI..
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 14 00 00 00 00 74	.....sRA..... ..t
Write Variable:	02 02 02 02 00 00 00 09 73 57 49 00 14 00 00 00 00 79	.....sWI..... ..y
Write Variable Response:	02 02 02 02 00 00 00 05 73 57 41 00 14 71	.....sWA..q

## 2.4.4. Method: accessConfigMemory

Use this method to

- a. save current settings in non-volatile memory
- b. restore settings from non-volatile memory
- c. reset your sensor to factory default settings

### Note

For permanently resetting the ML20 to factory defaults the following two method calls are required:

1. accessConfigMemory(tCMO\_RestoreDefaultConfiguration) to reset current settings to factory defaults
2. accessConfigMemory(tCMO\_SaveCurrentSettings) to save settings permanently

### Method Overview

Method Name	Description
accessConfigMemory	Execute configuration memory operation.

Sopas Index	3 (fixed)
Invocation Access	Always

Parameters			
operation	<b>Enum8</b>		
	Value	Name	Description
	0	tCMO_SaveCurrentSettings	Save current settings permanently to flash.
	1	tCMO_RestoreConfiguration	Restore settings from flash.
	2	tCMO_RestoreDefaultConfiguration	Restore factory firmware settings, will keep calibration values.

Return Values	
result	Result code for flash operation: an error occurred if this is not zero.
<b>Int</b>	
Value Range	-32768..32767
Initialisation	0



## Method Telegram Syntax

Method Invocation:				
sMI 3 <operation>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	3	String	1	Execute configuration memory operation.
Parameter 1	operation	Enum8	1	

Method Return Value:				
sAI 3 <result>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	3	String	1	Execute configuration memory operation.
Return Value 1	result	Int	2	Result code for flash operation: an error occurred if this is not zero.

## Method Telegram Examples

Example: Default Values				
Method telegram examples with parameter data and return value data set to default values.				
Method Invocation:	02 02 02 02 00 00 00 06	73 4D 49 00 03 00 74	.....sMI...t	
Method Return Value:	02 02 02 02 00 00 00 07	73 41 49 00 03 00 00 78	.....sAI....x	

## 2.4.5. Ethernet

Use the variables *udilpAddress*, *udiSubnetMask* and *udiGatewayAdress* to read or change current TCP/IP connection settings.

*Note:*

1. After changing TCP/IP settings call the method *accessConfigMemory* to permanently store the settings.
2. Changes of the TCP/IP settings will take effect after power-cycle of the sensor.

### 2.4.5.1. Variable: udilpAddress

The following section contains a detailed description of the variable *udilpAddress*.

#### Variable Overview

Variable Name	Description
udilpAddress	IP address

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	12 (fixed)
Read-Access	Always
Write-Access	Always

Array	
Length	4
Default Value	{192,168,100,100}
	<b>USInt</b>
	Value Range
	0..255

#### Variable Telegram Syntax

Read Variable:				
sRI 12				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	12	String	2	IP address

Read Variable Response:				
sRA 12 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	12	String	2	IP address
Variable Data	data	Array	4	





Write Variable:				
sWI 12 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	12	String	2	IP address
Variable Data	data	Array	4	

Write Variable Response:				
sWA 12				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	12	String	2	IP address

## Variable Telegram Examples

Example: Default Values				
Variable telegram examples with data set to default values.				
Read Variable:	02 02 02 02 00 00 00 05	73 52 49 00 0C 64	.....sRI...d	
Read Variable Response:	02 02 02 02 00 00 00 09 64 04	73 52 41 00 0C C0 A8 64	.....sRA...d d.	
Write Variable:	02 02 02 02 00 00 00 09 64 09	73 57 49 00 0C C0 A8 64	.....sWI...d d.	
Write Variable Response:	02 02 02 02 00 00 00 05	73 57 41 00 0C 69	.....sWA...i	

### 2.4.5.2. Variable: udiSubnetMask

The following section contains a detailed description of the variable udiSubnetMask.

#### Variable Overview

Variable Name	Description
udiSubnetMask	Subnet mask

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	15 (fixed)
Read-Access	Always
Write-Access	Always

Array	
Length	4
Default Value	{255,255,255,0}
	<b>USInt</b>
	Value Range
	0..255



## Variable Telegram Syntax

Read Variable:				
sRI 15				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	15	String	2	Subnet mask

Read Variable Response:				
sRA 15 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	15	String	2	Subnet mask
Variable Data	data	Array	4	

Write Variable:				
sWI 15 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	15	String	2	Subnet mask
Variable Data	data	Array	4	

Write Variable Response:				
sWA 15				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	15	String	2	Subnet mask

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 0F 67	.....sRI..g
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 0F FF FF FF 00 90	.....sRA.. .
Write Variable:	02 02 02 02 00 00 00 09 73 57 49 00 0F FF FF FF 00 9D	.....sWI.. .
Write Variable Response:	02 02 02 02 00 00 00 05 73 57 41 00 0F 6A	.....sWA..j

### 2.4.5.3. Variable: udiGatewayAddress

The following section contains a detailed description of the variable udiGatewayAddress.

#### Variable Overview

Variable Name	Description
udiGatewayAddress	Gateway adress

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	28 (fixed)
Read-Access	Always
Write-Access	Always

Array	
Length	4
Default Value	{0,0,0,0}
<b>USInt</b>	
Value Range	0..255

#### Variable Telegram Syntax

Read Variable:				
sRI 28				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	28	String	2	Gateway adress

Read Variable Response:				
sRA 28 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	28	String	2	Gateway adress
Variable Data	data	Array	4	

Write Variable:				
sWI 28 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	28	String	2	Gateway adress
Variable Data	data	Array	4	

Write Variable Response:				
sWA 28				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	28	String	2	Gateway adress



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 1C 74	.....sRI..t
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 1C 00 00 00 00 7C	.....sRA..... .
Write Variable:	02 02 02 02 00 00 00 09 73 57 49 00 1C 00 00 00 00 71	.....sWI..... .g
Write Variable Response:	02 02 02 02 00 00 00 05 73 57 41 00 1C 79	.....sWA..y



## 2.5. Teach

## 2.5.1. Teach-in and configuration

This chapter describes methods for configuring and executing the teach-in via the SOPAS TCP/IP interface.

### 2.5.1.1. Method: startTeach

This method starts a manual teach-in. The manual teach-in must be stopped using the stopTeach method.

#### Method Overview

Method Name	Description
startTeach	Immediately starts a teach. Teach must be stopped with the stopTeach method.

Sopas Index	9 (fixed)
Invocation Access	Always

Return Values			
result	Result code.		
	Enum8		
	Value	Name	Description
	0	eNoError	
	1	eErrorTeachBusy	

#### Method Telegram Syntax

Method Invocation:				
sMI 9				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	9	String	1	Immediately starts a teach. Teach must be stopped with the stopTeach method.

Method Return Value:				
sAI 9 <result>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	9	String	1	Immediately starts a teach. Teach must be stopped with the stopTeach method.
Return Value 1	result	Enum8	1	Result code.



## Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 05 73 4D 49 00 09 7E	.....sMI...~
Method Return Value:	02 02 02 02 00 00 00 06 73 41 49 00 09 00 72	.....sAI...r

### 2.5.1.2. Method: stopTeach

This method stops a manual teach-in.

#### Method Overview

Method Name	Description
stopTeach	Stops a teach procedure started previously with startTeach.
Sopas Index	10 (fixed)
Invocation Access	Always

#### Method Telegram Syntax

Method Invocation:				
sMI 10				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	10	String	2	Stops a teach procedure started previously with startTeach.

Method Return Value:				
sAI 10				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	10	String	2	Stops a teach procedure started previously with startTeach.

## Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 05 73 4D 49 00 0A 7D	.....sMI...}
Method Return Value:	02 02 02 02 00 00 00 05 73 41 49 00 0A 71	.....sAI...g

### 2.5.1.3. Method: triggerTeach

Calling this method immediately triggers a teach-in of the length specified in the variable *udiTrigTeachLength* and considering the teach direction specified in variable *eTeachDirectionSelect*. The content of these variables must therefore be defined prior to calling this function.

#### Method Overview

Method Name	Description
triggerTeach	Immediately triggers a teach at specified trigger teach length.

Sopas Index	8 (fixed)
Invocation Access	Always

Return Values			
result	Result code.		
	Enum8		
	Value	Name	Description
	0	eNoError	
	1	eErrorTeachBusy	

#### Method Telegram Syntax

Method Invocation:				
sMI 8				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	8	String	1	Immediately triggers a teach at specified trigger teach length.

Method Return Value:				
sAI 8 <result>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	8	String	1	Immediately triggers a teach at specified trigger teach length.
Return Value 1	result	Enum8	1	Result code.

#### Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 05 73 4D 49 00 08 7F	.....sMI...
Method Return Value:	02 02 02 02 00 00 00 06 73 41 49 00 08 00 73	.....sAI...s





#### 2.5.1.4. Method: cancelTeach

The method cancels any ongoing teach-in and makes the sensor return to RUN mode, see variable *eDeviceOperatingState*.

##### Method Overview

Method Name	Description
cancelTeach	Cancels any teach procedure started.
Sopas Index	18 (fixed)
Invocation Access	Always

##### Method Telegram Syntax

Method Invocation:				
sMI 18				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	18	String	2	Cancels any teach procedure started.

Method Return Value:				
sAI 18				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	18	String	2	Cancels any teach procedure started.

##### Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 05 73 4D 49 00 12 65	.....sMI..e
Method Return Value:	02 02 02 02 00 00 00 05 73 41 49 00 12 69	.....sAI..i

### 2.5.1.5. Variable: eTeachDirectionSelect

This variable selects the encoder moving direction during teach-in.

#### Note

If the encoder moving direction is known in advance, it is recommended to use the corresponding setting (CW or CCW) instead of automatic detection mode (Auto).

#### Variable Overview

Variable Name	
eTeachDirectionSelect	

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	38 (fixed)
Read-Access	Always
Write-Access	Always

Enum16			
Default Value		Auto	
	Value	Name	Description
	0	Auto	direction determined automatically
	1	CW	direction clock wise
	2	CCW	direction counter clock wise

#### Variable Telegram Syntax

Read Variable:				
sRI 38				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	38	String	2	

Read Variable Response:				
sRA 38 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	38	String	2	
Variable Data	data	Enum16	2	

Write Variable:				
sWI 38 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	38	String	2	
Variable Data	data	Enum16	2	



Write Variable Response:				
sWA 38				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	38	String	2	

## Variable Telegram Examples

Example: Default Values				
Variable telegram examples with data set to default values.				
Read Variable:	02 02 02 02 00 00 00 05	73 52 49 00 26 4E	.....sRI.&N	
Read Variable Response:	02 02 02 02 00 00 00 07	73 52 41 00 26 00 00 46	.....sRA.&..F	
Write Variable:	02 02 02 02 00 00 00 07	73 57 49 00 26 00 00 4B	.....sWI.&..K	
Write Variable Response:	02 02 02 02 00 00 00 05	73 57 41 00 26 43	.....sWA.&C	

### 2.5.1.6. Variable: udiTrigTeachLength

The following section contains a detailed description of the variable udiTrigTeachLength.

#### Variable Overview

Variable Name	Description
udiTrigTeachLength	Trigger teach length in [mm].

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	36 (fixed)
Read-Access	Always
Write-Access	Always

UDInt	
Value Range	0..4294967295
Initialisation	240



## Variable Telegram Syntax

Read Variable:				
sRI 36				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	36	String	2	Trigger teach length in [mm].

Read Variable Response:				
sRA 36 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	36	String	2	Trigger teach length in [mm].
Variable Data	data	UDInt	4	

Write Variable:				
sWI 36 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	36	String	2	Trigger teach length in [mm].
Variable Data	data	UDInt	4	

Write Variable Response:				
sWA 36				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	36	String	2	Trigger teach length in [mm].

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 24 4C	.....sRI.\$L
Read Variable Response:	02 02 02 02 00 00 00 09 F0 B4 73 52 41 00 24 00 00 00	.....sRA.\$...
Write Variable:	02 02 02 02 00 00 00 09 F0 B9 73 57 49 00 24 00 00 00	.....sWI.\$...
Write Variable Response:	02 02 02 02 00 00 00 05 73 57 41 00 24 41	.....sWA.\$A

### 2.5.2. Teach result data

After a teach-in all teach data except for the teach image is automatically stored to non-volatile memory and therefore immediately available after a restart. For advanced usage of the ML20 this paragraph describes methods for storing and restoring teach data via the TCP/IP interface.

#### Reading teach data for storing as external data set

A complete teach data set consists of eight patch sets and associated teach detail data.

To create a self-containing teach data set which may be saved to a data file one has to do the following steps:

1. Read and check variable *eDeviceOperatingState*. If *eDeviceOperatingState* = 2 (Run) continue with the following steps. Otherwise wait until teach is complete.

*Note:* Do not start a teach until read teach data is complete, otherwise the teach data read may be corrupt.

2. Call method *getPatchData* eight times, with parameter *Index* ranging from 0 to 7. Store the returned member fields *px*, *py*, *data*, *threshold* for each index.

3. Call method *readTeachData* and store all returned member fields *teachLength*, *teachDirection*, *teachQuality*, *refLabelLength*.

#### Restore teach data from an external data set

To restore a teach data set from previously read and stored teach data one has to do the following steps.

1. Read and check variable *eDeviceOperatingState*. If *eDeviceOperatingState* = 2 (Run) continue with the following steps. Otherwise wait until teach is complete.

*Note:* Do not start a new teach while restoring teach data. Starting a new teach while restoring teach data could lead to a corrupt a teach data set in the ML20 and detection failure.

2. Call method *setPatchData* eight times, with parameter *Index* ranging from 0 to 7. Fill in the data *px*, *py*, *data*, *threshold* for each index from the stored data set.

3. Call method *applyTeachData* and fill in data *teachLength*, *teachDirection*, *teachQuality*, *refLabelLength* from the stored data set. The new teach data will be used by the ML20 immediately.

#### Importing teach data stored with communication interface prior to V1.110

In prior versions of the communication interface (ie. V1.108) the threshold parameter of the methods *getPatchData* and *setPatchData* does not exist. When importing a teach data set stored with an earlier version of the ML20 communication interface to the current version V1.110, this parameter should be set to threshold =

32767 (0x7fff in hexadecimal code) for all patch indices when calling the method *setPatchData* to maintain functional compatibility.

### 2.5.2.1. Variable: eTeachResult

This variable contains the result of the last teach-in.

#### Variable Overview

Variable Name	Description
eTeachResult	Result of last teach.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	31 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

Enum16			
Default Value		UNTEACHED	
	Value	Name	Description
	0	UNTEACHED	Device not taught yet.
	1	SUCCESSFUL	Teach was successful or teachdata restored successfully.
	2	FAILED	Error occurred during teach.

#### Variable Telegram Syntax

Read Variable:				
sRI 31				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	31	String	2	Result of last teach.

Read Variable Response:				
sRA 31 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	31	String	2	Result of last teach.
Variable Data	data	Enum16	2	



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 1F 77	.....sRI..w
Read Variable Response:	02 02 02 02 00 00 00 07 73 52 41 00 1F 00 00 7F	.....sRA.....

### 2.5.2.2. Variable: udiDisplayTeachQuality

The value of teach quality ranges from 0 to 8, where a teach quality below 3 indicates low quality. In this case a new teach-in is recommended.

#### Variable Overview

Variable Name	Description
udiDisplayTeachQuality	Quality of teach, range 0 to 5

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	40 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

UDInt	
Value Range	0..5
Initialisation	0

#### Variable Telegram Syntax

Read Variable:				
sRI 40				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	40	String	2	Quality of teach, range 0 to 5

Read Variable Response:				
sRA 40 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	40	String	2	Quality of teach, range 0 to 5
Variable Data	data	UDInt	4	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 28 40	.....sRI· (@
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 28 00 00 00 00 48	.....sRA· ( ... ·H

### 2.5.2.3. Variable: udiCurrentTeachLength

The following section contains a detailed description of the variable udiCurrentTeachLength.

#### Variable Overview

Variable Name	Description
udiCurrentTeachLength	Length of last teach procedure in [mm/10].

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	18 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

UDInt	
Value Range	0..4294967295
Initialisation	0

#### Variable Telegram Syntax

Read Variable:				
sRI 18				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	18	String	2	Length of last teach procedure in [mm/10].

Read Variable Response:				
sRA 18 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	18	String	2	Length of last teach procedure in [mm/10].
Variable Data	data	UDInt	4	





## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 12 7A	.....sRI..z
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 12 00 00 00 00 72	.....sRA..... ·r

### 2.5.2.4. Variable: teCurrentTeachDirection

The following section contains a detailed description of the variable teCurrentTeachDirection.

#### Variable Overview

Variable Name	Description
teCurrentTeachDirection	Encoder direction during last teach procedure.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	48 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

UserType	
teEncoderDirection	See the chapter "User Types" for details.

#### Variable Telegram Syntax

Read Variable:				
sRI 48				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	48	String	2	Encoder direction during last teach procedure.

Read Variable Response:				
sRA 48 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	48	String	2	Encoder direction during last teach procedure.
Variable Data	data	teEncoderDirection	1	



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 30 58	.....sRI·0X
Read Variable Response:	02 02 02 02 00 00 00 06 73 52 41 00 30 00 50	.....sRA·0·P

### 2.5.2.5. Method: getPatchData

The following section contains a detailed description of the method getPatchData.

#### Method Overview

Method Name	Description
getPatchData	Query patch data from teach.

Sopas Index	22 (fixed)
Invocation Access	Always

Parameters	
index	Index of teach patch data.
Int	
Value Range	0..7

Return Values	
px	Position x.
UInt	
Value Range	0..65535
Initialisation	0
py	Position y.
UInt	
Value Range	0..65535
Initialisation	0
data	Patchdata array 8x32.
Array	
Length	256
USInt	
Value Range	0..255
threshold	Detection threshold.
UInt	
Value Range	0..65535



Method Invocation:				
sMI 22 <index>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	22	String	2	Query patch data from teach.
Parameter 1	index	Int	2	Index of teach patch data.

## Method Telegram Examples

---

ML20	8018469/2015-02-26	© SICK AG - Germany - All rights reserved	47
------	--------------------	---	----

### 2.5.2.6. Method: setPatchData

The following section contains a detailed description of the method setPatchData.

#### Method Overview

Method Name	Description
setPatchData	Set teach patch data.

Sopas Index	23 (fixed)
Invocation Access	Always

Parameters			
index	Index of teach patch data.		
	<b>Int</b>		
	Value Range	0..7	
px	Position x.		
	<b>UInt</b>		
	Value Range	0..65535	
	Initialisation	0	
py	Position y.		
	<b>UInt</b>		
	Value Range	0..65535	
	Initialisation	0	
	data	Patchdata array 8x32.	
<b>Array</b>			
Length		256	
		<b>UInt</b>	
		Value Range	0..255
threshold	Detection threshold		
	<b>UInt</b>		
	Value Range	0..65535	

Return Values			
result	Result code.		
	<b>Enum8</b>		
	Value	Name	Description
	0	eNoError	Operation successful.
	1	eErrorTeachBusy	Operation not allowed during teach.



## Method Telegram Syntax

Method Invocation:				
sMI 23 <index> <px> <py> <data> <threshold>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	23	String	2	Set teach patch data.
Parameter 1	index	Int	2	Index of teach patch data.
Parameter 2	px	UInt	2	Position x.
Parameter 3	py	UInt	2	Position y.
Parameter 4	data	Array	256	Patchdata array 8x32.
Parameter 5	threshold	UInt	2	Detection threshold

Method Return Value:				
sAI 23 <result>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	23	String	2	Set teach patch data.
Return Value 1	result	Enum8	1	Result code.

## Method Telegram Examples

Example: Default Values				
Method telegram examples with parameter data and return value data set to default values.				
Method Invocation:	02 02 02 02 00 00 01 0D	73 4D 49 00 17 00 00 00	.....sMI.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	
	00 00 00 00 00 00 60		.....`	
Method Return Value:	02 02 02 02 00 00 00 06	73 41 49 00 17 00 6C	.....sAI...1	

### 2.5.2.7. Method: readTeachData

The following section contains a detailed description of the method readTeachData.

#### Method Overview

Method Name	Description
readTeachData	Read teach data.

Sopas Index	17 (fixed)
Invocation Access	Always

Return Values	
teachLength	Teach length according to teach patch data in [frames].
UDInt	
Value Range	0..4294967295
teachDirection	Teach encoder direction.
UserType	
teEncoderDirection	See the chapter "User Types" for details.
teachQuality	Quality of teach, range 0 to 5
UDInt	
Value Range	0..5
refLabelLength	Reference label length in [frames].
UDInt	
Value Range	0..4294967295

#### Method Telegram Syntax

Method Invocation:				
sMI 17				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	17	String	2	Read teach data.

Method Return Value:				
sAI 17 <teachLength> <teachDirection> <teachQuality> <refLabelLength>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	17	String	2	Read teach data.
Return Value 1	teachLength	UDInt	4	Teach length according to teach patch data in [frames].
Return Value 2	teachDirection	teEncoderDirection	1	Teach encoder direction.
Return Value 3	teachQuality	UDInt	4	Quality of teach, range 0 to 5
Return Value 4	refLabelLength	UDInt	4	Reference label length in [frames].



## Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 05 73 4D 49 00 11 66	.....sMI..f
Method Return Value:	02 02 02 02 00 00 00 12 73 41 49 00 11 00 00 00 00 00 00 00 00 00 00 00 00 00 6A	.....sAI..... .....j

### 2.5.2.8. Method: applyTeachData

The following section contains a detailed description of the method applyTeachData.

#### Method Overview

Method Name	Description
applyTeachData	Apply teach and patch data.

Sopas Index	16 (fixed)
Invocation Access	Always

Parameters	
teachLength	Teach length according to teach patch data in [frames].
UDInt	
Value Range	0..4294967295
teachDirection	Teach encoder direction.
UserType	
teEncoderDirection	See the chapter "User Types" for details.
teachQuality	Quality of teach, range 0 to 5
UDInt	
Value Range	0..5
refLabelLength	Reference label length in [frames].
UDInt	
Value Range	0..4294967295

Return Values			
result	Result code.		
Enum8			
	Value	Name	Description
	0	eNoError	Operation successful.
	1	eErrorTeachBusy	Operation not allowed during teach.



## Method Telegram Syntax

Method Invocation:				
sMI 16 <teachLength> <teachDirection> <teachQuality> <refLabelLength>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	16	String	2	Apply teach and patch data.
Parameter 1	teachLength	UDInt	4	Teach length according to teach patch data in [frames].
Parameter 2	teachDirection	teEncoderDirection	1	Teach encoder direction.
Parameter 3	teachQuality	UDInt	4	Quality of teach, range 0 to 5
Parameter 4	refLabelLength	UDInt	4	Reference label length in [frames].

Method Return Value:				
sAI 16 <result>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	16	String	2	Apply teach and patch data.
Return Value 1	result	Enum8	1	Result code.

## Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 12 73 4D 49 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 67	.....sMI..... .....g
Method Return Value:	02 02 02 02 00 00 00 06 73 41 49 00 10 00 6B	.....sAI...k



### 2.5.3. Blanking

Blanking windows may be specified to restrict the teach-in to specific regions in the teach image.

During teach-in using the external blanking input "AT" may be used to define the blanking windows, see *ML20 Operating Instructions*. The results may be readout from the variables *sBlankingWindow1* and *sBlankingWindow2* after teach-in.

Blanking may also be invoked using the SOPAS TCP/IP interface. To apply blanking via TCP/IP one has to do the following steps:

1. Read and check variable *eDeviceOperatingState*. If *eDeviceOperatingState* = 2 (Run) continue with the following steps. Otherwise wait until teach is complete.
2. Read and check if variable *bHasTeachImage* = true . If no teach image is available first execute a new teach-in.
3. Set blanking windows in the variables *sBlankingWindow1* and if required *sBlankingWindow2*.
4. Call method *recomputeTeach*.
5. Complete the teach-in by moving labels in front of the ML20 in teach direction until the variable *eDeviceOperatingState* = 2 (Run). Check this variable approximately once or twice per second.

#### Vertical blanking

Vertical blanking is useful when teaching formats with a width lesser than the sensors field of view, ie. <34 mm. For vertical restriction of the teach-in region in the teach image the variables *uiVerticalBlankingTop* and *uiVerticalBlankingBottom* may be set during step 3 as described in the previous section *Horizontal blanking windows*.

The vertical blanking setting is independent of the horizontal blanking and may be configured in addition or by itself.

Vertical blanking from top and bottom must have a minimum value of 5 mm in order to compensate for lateral motion of the format.

*Note:* If the vertical blanking region is configured too large it might occur that no features can be found during teach-in which results in error code *ERR016*.

### 2.5.3.1. Method: recomputeTeach

The following section contains a detailed description of the method recomputeTeach.

#### Method Overview

Method Name	Description
recomputeTeach	Recompute teach data from current teach image in memory.

Sopas Index	12 (fixed)
Invocation Access	Always

Return Values			
result	Result code.		
	Enum8		
	Value	Name	Description
	0	eNoError	
	1	eErrorTeachBusy	
	2	eErrorNoTeachImage	

#### Method Telegram Syntax

Method Invocation:				
sMI 12				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	12	String	2	Recompute teach data from current teach image in memory.

Method Return Value:				
sAI 12 <result>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	12	String	2	Recompute teach data from current teach image in memory.
Return Value 1	result	Enum8	1	Result code.

#### Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 05 73 4D 49 00 0C 7B	.....sMI..{
Method Return Value:	02 02 02 02 00 00 00 06 73 41 49 00 0C 00 77	.....sAI...w

### 2.5.3.2. Variable: sBlankingWindow1

The following section contains a detailed description of the variable sBlankingWindow1.

#### Variable Overview

Variable Name	Description
sBlankingWindow1	Teach blanking window 1 in [mm].

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	55 (fixed)
Read-Access	Always
Write-Access	Always

UserType	
BlankingWinStruct	See the chapter "User Types" for details.

#### Variable Telegram Syntax

Read Variable:				
sRI 55				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	55	String	2	Teach blanking window 1 in [mm].

Read Variable Response:				
sRA 55 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	55	String	2	Teach blanking window 1 in [mm].
Variable Data	data	BlankingWinStruct	4	

Write Variable:				
sWI 55 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	55	String	2	Teach blanking window 1 in [mm].
Variable Data	data	BlankingWinStruct	4	

Write Variable Response:				
sWA 55				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	55	String	2	Teach blanking window 1 in [mm].



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
<b>Read Variable:</b>	02 02 02 02 00 00 00 05 73 52 49 00 37 5F	.....sRI·7_
<b>Read Variable Response:</b>	02 02 02 02 00 00 00 09 73 52 41 00 37 00 00 00 00 57	.....sRA·7... ·W
<b>Write Variable:</b>	02 02 02 02 00 00 00 09 73 57 49 00 37 00 00 00 00 5A	.....sWI·7... ·Z
<b>Write Variable Response:</b>	02 02 02 02 00 00 00 05 73 57 41 00 37 52	.....sWA·7R

### 2.5.3.3. Variable: sBlankingWindow2

The following section contains a detailed description of the variable sBlankingWindow2.

#### Variable Overview

Variable Name	Description
sBlankingWindow2	Teach blanking window 2 in [mm].

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	56 (fixed)
Read-Access	Always
Write-Access	Always

UserType	
BlankingWinStruct	See the chapter "User Types" for details.

#### Variable Telegram Syntax

Read Variable:				
sRI 56				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	56	String	2	Teach blanking window 2 in [mm].

Read Variable Response:				
sRA 56 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	56	String	2	Teach blanking window 2 in [mm].
Variable Data	data	BlankingWinStruct	4	



Write Variable:				
sWI 56 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	56	String	2	Teach blanking window 2 in [mm].
Variable Data	data	BlankingWin Struct	4	

Write Variable Response:				
sWA 56				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	56	String	2	Teach blanking window 2 in [mm].

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
<b>Read Variable:</b>	02 02 02 02 00 00 00 05 73 52 49 00 38 50	.....sRI·8P
<b>Read Variable Response:</b>	02 02 02 02 00 00 00 09 73 52 41 00 38 00 00 00 00 58	.....sRA·8... ·X
<b>Write Variable:</b>	02 02 02 02 00 00 00 09 73 57 49 00 38 00 00 00 00 55	.....sWI·8... ·U
<b>Write Variable Response:</b>	02 02 02 02 00 00 00 05 73 57 41 00 38 5D	.....sWA·8]

### 2.5.3.4. Variable: uiVerticalBlankingTop

The following section contains a detailed description of the variable uiVerticalBlankingTop.

#### Variable Overview

Variable Name	Description
uiVerticalBlankingTop	End position of vertical blanking area in [mm] relative to upper border of image.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	94 (fixed)
Read-Access	Always
Write-Access	Always

UInt	
Value Range	0..28
Initialisation	5



## Variable Telegram Syntax

Read Variable:				
sRI 94				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	94	String	2	End position of vertical blanking area in [mm] relative to upper border of image.

Read Variable Response:				
sRA 94 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	94	String	2	End position of vertical blanking area in [mm] relative to upper border of image.
Variable Data	data	UInt	2	

Write Variable:				
sWI 94 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	94	String	2	End position of vertical blanking area in [mm] relative to upper border of image.
Variable Data	data	UInt	2	

Write Variable Response:				
sWA 94				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	94	String	2	End position of vertical blanking area in [mm] relative to upper border of image.

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 5E 36	.....sRI.^6
Read Variable Response:	02 02 02 02 00 00 00 07 73 52 41 00 5E 00 05 3B	.....sRA.^...;
Write Variable:	02 02 02 02 00 00 00 07 73 57 49 00 5E 00 05 36	.....sWI.^...6
Write Variable Response:	02 02 02 02 00 00 00 05 73 57 41 00 5E 3B	.....sWA.^;

### 2.5.3.5. Variable: uiVerticalBlankingBottom

The following section contains a detailed description of the variable uiVerticalBlankingBottom.

#### Variable Overview

Variable Name	Description
uiVerticalBlankingBottom	End position of vertical blanking area in [mm] relative to lower border of image.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	95 (fixed)
Read-Access	Always
Write-Access	Always

UInt	
Value Range	0..28
Initialisation	5

#### Variable Telegram Syntax

Read Variable:				
sRI 95				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	95	String	2	End position of vertical blanking area in [mm] relative to lower border of image.

Read Variable Response:				
sRA 95 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	95	String	2	End position of vertical blanking area in [mm] relative to lower border of image.
Variable Data	data	UInt	2	

Write Variable:				
sWI 95 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	95	String	2	End position of vertical blanking area in [mm] relative to lower border of image.
Variable Data	data	UInt	2	

Write Variable Response:				
sWA 95				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	95	String	2	End position of vertical blanking area in [mm] relative to lower border of image.



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 5F 37	.....sRI._7
Read Variable Response:	02 02 02 02 00 00 00 07 73 52 41 00 5F 00 05 3A	.....sRA._...:
Write Variable:	02 02 02 02 00 00 00 07 73 57 49 00 5F 00 05 37	.....sWI._...7
Write Variable Response:	02 02 02 02 00 00 00 05 73 57 41 00 5F 3A	.....sWA._:



## 2.6. Run

This paragraph describes variables important during run mode of the ML20. The following read-only variables return process information during run:

*eDeviceOperatingState* informs whether device is in teach or in run mode.

*udiDisplayRunQuality* contains the current detection quality.

*udiActualFormatLength* contains the label length according to the last detected switching events.

Use the variable *diQOffset* to shift the switching point during run.

### 2.6.1. Variable: eDeviceOperatingState

The following section contains a detailed description of the variable *eDeviceOperatingState*.

#### Variable Overview

Variable Name	Description
eDeviceOperatingState	Device operating state.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	30 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

Enum16			
Default Value		RUN	
Value	Name	Description	
1	RUN	Device is running.	
2	TEACH	Device is in teach mode.	

#### Variable Telegram Syntax

Read Variable:				
sRI 30				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	30	String	2	Device operating state.

Read Variable Response:				
sRA 30 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	30	String	2	Device operating state.
Variable Data	data	Enum16	2	



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 1E 76	.....sRI..v
Read Variable Response:	02 02 02 02 00 00 00 07 73 52 41 00 1E 00 01 7F	.....sRA.....

### 2.6.2. Variable: udiDisplayRunQuality

The quality value ranges from 0 to 6. A value below 3 indicates bad quality, ie. missing or additional detections. A value equal to or above 3 indicates correct detection.

#### Variable Overview

Variable Name	Description
udiDisplayRunQuality	Quality of run, range 0 to 6

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	39 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

UDInt	
Value Range	0..6
Initialisation	0

#### Variable Telegram Syntax

Read Variable:				
sRI 39				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	39	String	2	Quality of run, range 0 to 6

Read Variable Response:				
sRA 39 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	39	String	2	Quality of run, range 0 to 6
Variable Data	data	UDInt	4	



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 27 4F	.....sRI.'O
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 27 00 00 00 00 47	.....sRA.'.... •G

### 2.6.3. Variable: udiActualFormatLength

The following section contains a detailed description of the variable udiActualFormatLength.

#### Variable Overview

Variable Name	Description
udiActualFormatLength	Current format length in [mm/10].

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	43 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

UDInt	
Value Range	0..4294967295
Initialisation	0

#### Variable Telegram Syntax

Read Variable:				
sRI 43				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	43	String	2	Current format length in [mm/10].

Read Variable Response:				
sRA 43 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	43	String	2	Current format length in [mm/10].
Variable Data	data	UDInt	4	



## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 2B 43	.....sRI.+C
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 2B 00 00 00 00 4B	.....sRA+.... •K

### 2.6.4. Variable: diQOffset

This variable shifts the detection point by the given length value (offset). Only positive offset values are permitted.

The default offset is zero, corresponding to label start. The maximum offset value must correspond to the label length.

#### Note

The offset value in *diQOffset* is reset to zero with each new teach-in. If desired the offset value should be configured after teach-in.

## Variable Overview

Variable Name	Description
diQOffset	Switching Q offset in [mm] relative to label start.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	45 (fixed)
Read-Access	Always
Write-Access	Always

DInt	
Value Range	0..999
Initialisation	0



## Variable Telegram Syntax

Read Variable:				
sRI 45				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	45	String	2	Switching Q offset in [mm] relative to label start.

Read Variable Response:				
sRA 45 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	45	String	2	Switching Q offset in [mm] relative to label start.
Variable Data	data	DInt	4	

Write Variable:				
sWI 45 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWI	String	3	Write SOPAS Variable by Name
Command	45	String	2	Switching Q offset in [mm] relative to label start.
Variable Data	data	DInt	4	

Write Variable Response:				
sWA 45				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sWA	String	3	SOPAS Variable Write Acknowledge
Command	45	String	2	Switching Q offset in [mm] relative to label start.

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 2D 45	.....sRI--E
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 2D 00 00 00 00 4D	.....sRA----- •M
Write Variable:	02 02 02 02 00 00 00 09 73 57 49 00 2D 00 00 00 00 40	.....sWI----- •@
Write Variable Response:	02 02 02 02 00 00 00 05 73 57 41 00 2D 48	.....sWA--H



## 2.7. Diagnostics

### 2.7.1. Variable: eErrorCode

This variable returns the current error state of the device. Most errors and warnings may occur during teach or immediately after sensor start up. Any error or warning may be overwritten by a subsequent error or warning.

Warnings may be issued during teach-in only. They indicate problems during teach-in, yet the teach-in could be completed.

In the list below the names of warning codes start with prefix *WRN* while names of error codes start with prefix *ERR*.

#### Variable Overview

Variable Name	Description
eErrorCode	Device error or warning code.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	44 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

Enum16			
	Value	Name	Description
	0	NoErr	No error has occurred.
	1	ERR001	Shortcircuit at Q or status output detected.
	5	ERR005	Primary boot image corrupt.
	10	ERR010	Teachlength below 40mm.
	11	WRN011	Teachlength above 1000mm.
	12	WRN012	Sensor signal is overdriven due to overexposure.
	13	WRN013	Low contrast in teach image.
	14	ERR014	Teach velocity too fast.
	15	WRN015	Teachlength is longer than labellength.
	16	ERR016	No features found during teach.
	17	ERR017	Can not recompute without teach image (only when invoked via display).
	18	ERR018	Label not found after teach.



## Variable Telegram Syntax

Read Variable:				
sRI 44				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	44	String	2	Device error or warning code.

Read Variable Response:				
sRA 44 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	44	String	2	Device error or warning code.
Variable Data	data	Enum16	2	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 2C 44	.....sRI.,D
Read Variable Response:	02 02 02 02 00 00 00 07 73 52 41 00 2C 00 00 4C	.....sRA.,..L

### 2.7.2. Method: getEncoderPosition

This method returns current encoder position (in unit encoder counts) and direction for diagnostic purposes.

#### Method Overview

Method Name	Description
getEncoderPosition	Query encoder position and direction.

Sopas Index	6 (fixed)
Invocation Access	Always

Return Values	
position	Current encoder position (quadrature decoded).
<b>UInt</b>	
Value Range	0..65535
direction	Current rotation direction of encoder.
<b>UserType</b>	
teEncoderDirection	See the chapter "User Types" for details.



## Method Telegram Syntax

Method Invocation:				
sMI 6				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	6	String	1	Query encoder position and direction.

Method Return Value:				
sAI 6 <position> <direction>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	6	String	1	Query encoder position and direction.
Return Value 1	position	UInt	2	Current encoder position (quadrature decoded).
Return Value 2	direction	teEncoderDirection	1	Current rotation direction of encoder.

## Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 05 73 4D 49 00 06 71	.....sMI..q
Method Return Value:	02 02 02 02 00 00 00 08 73 41 49 00 06 00 00 00 7D	.....sAI.....}



### 2.7.3. Image handling

The ML20 provides means to read out image data for diagnostic purposes. The image data is organized in lines of 128 pixels with an 8-Bit grey scale value, see user type *ImageDataStruct*.

For displaying the image at a correct aspect ratio in a viewer consider the variable *sPixelFormat*, which gives the extension of a pixel in x-y-direction in the object plane.

#### Reading a teach image

After teach execution a teach image is available and may be read from the ML20 using the following steps:

1. Read and check variable *bHasTeachImage* . If *bHasTeachImage* = false no teach image is available for reading. Otherwise continue with the following steps.
2. Read variable *udlImageSize* to find out the image size of the teach image, Unit of the variable content is the number of image lines.
3. Read all image lines by repetitive calling of method *getImage* . This method returns the image in line blocks of up to four lines per call. With each call an ML20-internal linecounter is incremented by the number of returned lines. The return value of the *getImage*-method is of the structuretype *ImageDataStruct*. It contains the current linecounter in member *lineId* and up to four lines in the 2-D byte array *frameData* [1..4][128]. The number of actually transferred lines may be read from the length-field of the first dimension of *frameData*.

Except for the last lineblock always four lines are transferred. The last line block may contain less lines. Continue reading beyond the last lineblock will return zero lines, so the length-field of the first dimension of *frameData* will be zero and the linecounter *lineId* will not increment anymore.

The following cases have to be implemented for reading image data

- 3a. At the first call the method *getImage* must be called with parameter *first* = true. This will reset the linecounter to zero in order to transfer the first line block of the image.
- 3b. In subsequent calls the method *getImage* is called with Parameter *first* = false until all lines have been retrieved.
4. The image must be assembled from the returned lineblocks.

## Acquiring and reading a process image

A process image may be acquired and read out during run by using the following steps:

1. Start image recording by calling method *acquireRunImage*. The ML20 will trigger recording at the next switching event of the Q-output and stop automatically at the subsequent switching event.

*Note:* a possibly stored teach image or an earlier process image will be overwritten.

2. Read and check variable *bHasRunImage* repetitively until *bHasRunImage* = true . Then the process image is available and it may be proceeded to step 3.

3. Continue with step 2 of prior paragraph *Reading a teach image*.

### 2.7.3.1. Method: getImage

The following section contains a detailed description of the method getImage.

#### Method Overview

Method Name	Description
getImage	Query teach or run image.

Sopas Index	13 (fixed)
Invocation Access	Always

Parameters	
first	Set true for querying first line of image.
<b>Bool</b>	
Value Range	False, True
Initialisation	False

UserType	
ImageDataStruct	See the chapter "User Types" for details.



## Method Telegram Syntax

Method Invocation:				
sMI 13 <first>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	13	String	2	Query teach or run image.
Parameter 1	first	Bool	1	Set true for querying first line of image.

Method Return Value:				
sAI 13 <lineId> <frameData>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	13	String	2	Query teach or run image.
Return Value 1	lineId	UInt	2	Index in image of first line in struct
Return Value 2	frameData	Array	512	Array of up to 4 lines with 128 pixel each

## Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 06 73 4D 49 00 0D 00 7A	.....sMI...z
Method Return Value:	02 02 02 02 00 00 00 09 73 41 49 00 0D 00 00 00 00 76	.....sAI..... ..v

### 2.7.3.2. Method: acquireRunImage

The following section contains a detailed description of the method acquireRunImage.

#### Method Overview

Method Name	Description
acquireRunImage	Schedules acquisition of next run image.

Sopas Index	11 (fixed)
Invocation Access	Always

Return Values			
result	Result code.		
	Enum8		
	Value	Name	Description
	0	eNoError	
	1	eErrorTeachBusy	
	2	eErrorAcqRunImageBusy	



## Method Telegram Syntax

Method Invocation:				
sMI 11				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sMI	String	3	Request (SOPAS Method by Name)
Command	11	String	2	Schedules acquisition of next run image.

Method Return Value:				
sAI 11 <result>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sAI	String	3	Result (SOPAS Method Result)
Command	11	String	2	Schedules acquisition of next run image.
Return Value 1	result	Enum8	1	Result code.

## Method Telegram Examples

Example: Default Values		
Method telegram examples with parameter data and return value data set to default values.		
Method Invocation:	02 02 02 02 00 00 00 05 73 4D 49 00 0B 7C	.....sMI..
Method Return Value:	02 02 02 02 00 00 00 06 73 41 49 00 0B 00 70	.....sAI...p

### 2.7.3.3. Variable: udilImageSize

The following section contains a detailed description of the variable udilImageSize.

#### Variable Overview

Variable Name	Description
udilImageSize	Length of current image in memory in [frames].

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	54 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

UDInt	
Value Range	0..4294967295
Initialisation	0



## Variable Telegram Syntax

Read Variable:				
sRI 54				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	54	String	2	Length of current image in memory in [frames].

Read Variable Response:				
sRA 54 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	54	String	2	Length of current image in memory in [frames].
Variable Data	data	UDInt	4	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 36 5E	.....sRI.6^
Read Variable Response:	02 02 02 02 00 00 00 09 73 52 41 00 36 00 00 00 00 56	.....sRA.6... ·V

### 2.7.3.4. Variable: sPixelFormat

The following section contains a detailed description of the variable sPixelFormat.

#### Variable Overview

Variable Name	Description
sPixelFormat	Dimensions represented by a pixel in the focal plane

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	87 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

Struct	
x	Extension in x direction in [mm]
<div>LReal</div> <div>Value Range</div> <div>Initialisation</div>	See specification IEEE 754
	0.6
y	Extension in y direction in [mm]
<div>LReal</div> <div>Value Range</div> <div>Initialisation</div>	See specification IEEE 754
	0.24

## Variable Telegram Syntax

Read Variable:				
sRN sPixelFormat				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRN	String	3	Read SOPAS Variable by Name
Command	sPixelFormat	String	12	Dimensions represented by a pixel in the focal plane

Read Variable Response:				
sRA sPixelFormat <x> <y>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	sPixelFormat	String	12	Dimensions represented by a pixel in the focal plane
Variable Data 1	x	LReal	8	Extension in x direction in [mm]
Variable Data 2	y	LReal	8	Extension in y direction in [mm]

## Variable Telegram Examples

Example: Default Values	
Variable telegram examples with data set to default values.	
Read Variable:	sRN sPixelFormat
Read Variable Response:	sRA sPixelFormat 3FE3333333333333 3FCEB851EB851EB8

### 2.7.3.5. Variable: bHasTeachImage

The following section contains a detailed description of the variable bHasTeachImage.

#### Variable Overview

Variable Name	Description
bHasTeachImage	Teach image available for transfer.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	32 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

Bool	
Value Range	False, True
Initialisation	False



## Variable Telegram Syntax

Read Variable:				
sRI 32				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	32	String	2	Teach image available for transfer.

Read Variable Response:				
sRA 32 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	32	String	2	Teach image available for transfer.
Variable Data	data	Bool	1	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 20 48	.....sRI· H
Read Variable Response:	02 02 02 02 00 00 00 06 73 52 41 00 20 00 40	.....sRA· ·@

### 2.7.3.6. Variable: bHasRunImage

The following section contains a detailed description of the variable bHasRunImage.

#### Variable Overview

Variable Name	Description
bHasRunImage	Run image available for transfer.

Sopas Synchronisation	Variable is relevant for synchronisation with SOPAS ET.
Sopas Index	33 (fixed)
Read-Access	Always
Write-Access	No! (readonly)

Bool	
Value Range	False, True
Initialisation	False



## Variable Telegram Syntax

Read Variable:				
sRI 33				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRI	String	3	Read SOPAS Variable by Name
Command	33	String	2	Run image available for transfer.

Read Variable Response:				
sRA 33 <data>				
Telegram Part	Telegram	Type	Length [Byte]	Description
Command Type	sRA	String	3	SOPAS Variable Read Acknowledge
Command	33	String	2	Run image available for transfer.
Variable Data	data	Bool	1	

## Variable Telegram Examples

Example: Default Values		
Variable telegram examples with data set to default values.		
Read Variable:	02 02 02 02 00 00 00 05 73 52 49 00 21 49	.....sRI·!·I
Read Variable Response:	02 02 02 02 00 00 00 06 73 52 41 00 21 00 41	.....sRA·!·A



## 3. User types

### 3.1. Type: teEncoderDirection

The following section contains a detailed description of the user type teEncoderDirection.

Type	Description
teEncoderDirection	Direction of encoder rotation

Struct			
value			
	<b>Enum8</b>		
		Value	Description
		0	eCW direction clock wise
		1	eCCW direction counter clock wise

### 3.2. Type: ImageDataStruct

The following section contains a detailed description of the user type ImageDataStruct.

Type	Description
ImageDataStruct	Struct containing image frame data of 1 to 4 lines.

Struct			
linelid	Index in image of first line in struct		
	<b>UInt</b>		
		Value Range	0..65535
frameData	Array of up to 4 lines with 128 pixel each		
	<b>Array</b>		
		Length	0..4
		<b>Array</b>	
		Length	128
		<b>UInt</b>	
		Value Range	0..255

### 3.3. Type: BlankingWinStruct

The following section contains a detailed description of the user type BlankingWinStruct.

Type	Description
BlankingWinStruct	Start and stop index of blanking window

Struct	
start	Start position of blanking window in [mm], must be less than stop index
UInt	
Value Range	0..1000
stop	Stop position of blanking window in [mm]
UInt	
Value Range	0..1000



# Index

## A

accessConfigMemory 26  
acquireRunImage 71  
applyTeachData 51

## B

bHasRunImage 75  
bHasTeachImage 74  
BlankingWinStruct 78

## C

cancelTeach 37

## D

DeviceIdent 7  
diQOffset 64

## E

eDeviceOperatingState 61  
eErrorCode 66  
Encoder type; *udiUserConfig* 24  
Errorhandling; *udiUserConfig* 24  
eTeachDirectionSelect 38  
eTeachResult 42

## F

FirmwareVersion 11

## G

GetAccessMode 16  
GetDescription 18  
getEncoderPosition 67  
getImage 70  
getPatchData 46

## I

ImageDataStruct 77

## L

Line teach mode; *udiUserConfig* 24  
LocationName 9

## M

Method: accessConfigMemory 26  
Method: acquireRunImage 71  
Method: applyTeachData 51  
Method: cancelTeach 37  
Method: GetAccessMode 16

Method: GetDescription 18  
Method: getEncoderPosition 67  
Method: getImage 70  
Method: getPatchData 46  
Method: readTeachData 50  
Method: recomputeTeach 54  
Method: Run 17  
Method: SetAccessMode 15  
Method: setPatchData 48  
Method: startTeach 34  
Method: stopTeach 35  
Method: triggerTeach 36

## R

readTeachData 50  
recomputeTeach 54  
Run 17

## S

sBlankingWindow1 55  
sBlankingWindow2 56  
SerialNumber 10  
SetAccessMode 15  
setPatchData 48  
SopasInfo 12  
SOPASVersion 8  
sPixelFormat 73  
startTeach 34  
stopTeach 35

## T

teCurrentTeachDirection 45  
teEncoderDirection 77  
triggerTeach 36  
Type: BlankingWinStruct 78  
Type: ImageDataStruct 77  
Type: teEncoderDirection 77

## U

udiActualFormatLength 63  
udiCurrentTeachLength 44  
udiDisplayRunQuality 62  
udiDisplayTeachQuality 43  
udiEncoderResolution 21  
udiFrameResolution 23  
udiGatewayAddress 31  
udiImageSize 72  
udiIpAddress 28  
udiSubnetMask 29  
udiTrigTeachLength 39  
udiUserConfig 24  
uiVerticalBlankingBottom 59  
uiVerticalBlankingTop 57



## V

Variable: bHasRunImage 75  
Variable: bHasTeachImage 74  
Variable: DeviceIdent 7  
Variable: diQOffset 64  
Variable: eDeviceOperatingState 61  
Variable: eErrorCode 66  
Variable: eTeachDirectionSelect 38  
Variable: eTeachResult 42  
Variable: FirmwareVersion 11  
Variable: LocationName 9  
Variable: sBlankingWindow1 55  
Variable: sBlankingWindow2 56  
Variable: SerialNumber 10  
Variable: SopasInfo 12  
Variable: SOPASVersion 8  
Variable: sPixelFormat 73  
Variable: teCurrentTeachDirection 45  
Variable: udiActualFormatLength 63  
Variable: udiCurrentTeachLength 44  
Variable: udiDisplayRunQuality 62  
Variable: udiDisplayTeachQuality 43  
Variable: udiEncoderResolution 21  
Variable: udiFrameResolution 23  
Variable: udiGatewayAddress 31  
Variable: udiImageSize 72  
Variable: udiIpAddress 28  
Variable: udiSubnetMask 29  
Variable: udiTrigTeachLength 39  
Variable: udiUserConfig 24  
Variable: uiVerticalBlankingBottom 59  
Variable: uiVerticalBlankingTop 57



Worldwide presence with subsidiaries  
in the following countries:

Australia  
Belgium/Luxembourg  
Brasil  
Česká Republika  
China  
Danmark  
Deutschland  
España  
France  
Great Britain  
India  
Israel  
Italia  
Japan

Nederlands  
Norge  
Österreich  
Polska  
Republic of Korea  
Republika Slovenija  
România  
Russia  
Schweiz  
Singapore  
Suomi  
Sverige  
Taiwan  
Türkiye  
United Arab Emirates  
USA/Canada/México

Please find detailed addresses and additional  
representatives and agencies in all major  
industrial nations at  
**[www.sick.com](http://www.sick.com)**