SharePoint Jhohan Ng

TKB Technical Knowledge Base

LMS_TiM Scan data telegram: Content and parameters

Products: LMS1xx, LMS5xx, TiM3xx, TiM5xx
Contact person: Markus Wagner

Contents [Hide]

Data output [Edit]

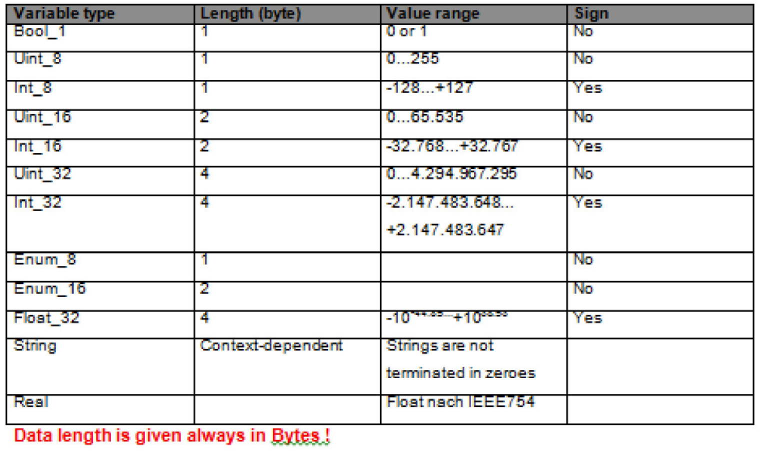
If a measurement data telegram is requested (sRN LMDscandata), the last valid scan will be given out. This can also be old values from a former measurement if the measurement was stopped in the mean time for example.
I the scandata event is subscribed (sEN LMDscandata 1), than always the acutal measurement data will be given out permanently. If the measurement is stopped, than no data will be given out any more.

Data format [Edit]

The measuring values you receive from the scanner are 16 bit integer values (word), the 8-bit values are not implemented. With the binary complement operation you can convert it into decimal values. When you convert FFC7 you'll get the decimal value -57. The negative value means that your object is too close to the scanner and it cannot measure the distance (Minimum distance between scanner and object is 500 mm). Now the scale factor describes that your measuring data is in mm so: 57mm

Data output is possible in CoLa-A (ASCII formag) or in CoLa-B (binary format).
Settings have to be made in the ethernet settings of the device.

Data types [Edit]



Variable type	Length (byte)	Value range	Sign
Bool_1	1	0 or 1	No
Uint_8	1	0...255	No
Int_8	1	-128...+127	Yes
Uint_16	2	0...65.535	No
Int_16	2	-32.768...+32.767	Yes
Uint_32	4	0...4.294.967.295	No
Int_32	4	-2.147.483.648... +2.147.483.647	Yes
Enum_8	1		No
Enum_16	2		No
Float_32	4	$-10^{-44.85}$...$+10^{38.53}$	Yes
String	Context-dependent	Strings are not terminated in zeroes	
Real		Float nach IEEE754	

Data length is given always in Bytes !

binary protocol [Edit]

The binary protocol is the basic protocol of the scanner. It has always a fix length and the content and byte length of the string fit to that document.
The binary protocol has a special framing so that the scanner is able to recognize it as the start of a binary telegram.
The string has to start with 4 STX symbols (for example: 02 02 02 02), that is followed by the length of the telegram in HEX (for example: 00 00 00 17).
Example:

Binary	02 02 02 02 00 00 00 17 73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 03 F4 72 47 44 B3 Header: 02 02 02 02; Length: 00 00 00 17; Checksum: B3

The length could be created by counting every single letter (Hex value) of the command (without checksum and framing but with blanks) and convert the value into HEX.
After the length the command itself starts. All letters of the command convertet to HEX and That the parameters (mostly numbers) written directly behind the command in pairs of two.
All parameters of the command has to be in hex (for example: scan frequency 25Hz is 00009C4h (It is a 4 byte value).
Checksum is calculated with XOR.
Between the command and the parameters, there has to be a blank, but not between the parameters itself.

Example string:
sMN SetAccessMode 03 F4724744

Binary string:
02 02 02 02 00 00 00 17 73 4D 4E 20 53 65 74 41 63 63 65 73 73 4D 6F 64 65 20 03 F4 72 47 44 B3
In the scandata telegram from the scanner, the range values could be used as the are, they don’t have to be convertet. Every value is 2 byte long.
The binary protocol could only be used at the host port of the scanner, and at the moment only with the LMS1xxx.

ASCII Protocol [Edit]

The ASCII telegram is an additional format and because of the ASCII signs a little better to understand.
The framing of the telegram is a STX at the start and an ETX at the end of each telegram.
The command is written in ASCII letters, followed by the parameters like defined in that document. Parameters could be transferred in hex or decimal format, but in decimal format they need a sign (for example: scan frequency 25Hz: 09C4h/+2500d)
Attention: leading zeros of each parameter and value were deleted, so the byte length of a parameter may not fit to what is standing in that document. That also causes different string length in the scan data telegram.
For using with PLC’s the binary protocol is recommended.

Motorola Format of the string [Edit]

The telegram is based on the motorola format. That means that if a values coming in two parts, the LSB comes first and then the MSB.
For example the digital outputs in the measurement telegram will be given out as: 07 00
MSB = 00
LSB = 07

So finally the real value is 00 07

Example of the data string [Edit]

This Example is done with a LMS5xx, because it may be the longest telegram because of it's 5 pulses.
The LMS1xx and the TiM just have less measurement data (only one or two pulses DIST1 or DIST1+2), all the rest is the same.
sRN LMDscandatasRA LMDscandata 0 1 9E14CE 0 0 ECF8 ED6E C267B795 C268A7A8 0 0 3F 0 0 9C4 21C 1 3AD 0 5 **DIST1** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 890B 8927 8945 8922 892F 8936 8975 0 0 0 0 0 8A4A 8A54 8A7D 0 0 8456 848E 84AF 8506 8560 85E9 8697 86DE 7E16 7DF5 **DIST2** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8720 874D **DIST3** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **DIST4** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **DIST5** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 **RSSI1** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 D 19 26 23 29 21 15 0 0 0 0 0 10 12 7 0 0 15 2E 36 38 37 33 33 29 14 28 **RSSI2** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1E 14 **RSSI3** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **RSSI4** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **RSSI5** *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 10 not defined 0 1 7B2 1 F F 27 1F D59F8 0

Black:	Request
Dark grey:	Front Header
Yellow:	Measurement and RSSI Values from Echo 1
Green:	Measurement and RSSI Values from Echo 2
Turquoise:	Measurement and RSSI Values from Echo 3
Red:	Measurement and RSSI Values from Echo 4
Pink:	Measurement and RSSI Values from Echo 5
Light grey:	End Header

Meanings of single Parts (Everything is in HEX):

Front Header [Edit]

sRA LMDscandata 0 1 9E14CE 0 0 ECF8 ED6E C267B795 C268A7A8 0 0 3F 0 0 9C4 21C 1 3AD 0 5

sRA	=	Command Type
LMD Scandata	=	Command Name
0	=	Version Number
1	=	Device Number
9E14CE	=	Serial Number (=1036 0014)
0 0	=	Device Status (The value "Device Status" in the scan telegram exists of two 8-bit values. So it is not one 16-bit value, but two 8-bit values. Also it is Low byte first, so the sequence is LSB, MSB.)
ECF8	=	Telegram Counter
ED6E	=	Scan Counter
C267B795	=	Time since Startupp in µsec
C268A7A8	=	Time of transmission in µsec
0 0	=	Status of the digital inputs (two 8-bit values)
3F 0	=	Status of the digital outputs (3Fh = 111111b = all outputs high)
0	=	Reserved
9C4	=	Scan frequency (9C4h = 2500d = 25Hz)
21C	=	Measurement frequency

1	=	Amount of encoder data
3AD	=	Encoder Position
0	=	Encoder Speed
5	=	Amount of 16bit Channels (means how many echoes are evaluated)

Measurement Values (example: Echo 1) [Edit]

DIST1 *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 890B 8927 8945 8922 892F 8936 8975 0 0 0 0 0 8A4A 8A54 8A7D 0 0 8456 848E 84AF 8506 8560 85E9 8697 86DE 7E16 7DF5

DIST1	=	now coming the radial distance values of the first pulse
3F800000	=	Scale Factor (3F800000 = factor 1; 40000000 = factor 2)
00000000	=	Scale Factor Offset
DBBA0	=	Start angle (DBBA0h = 900.000d = 90°)
683	=	Angular resolution (683h = 1.667d = 0,1667°)
1F	=	Amount of Data (1Fh = 31d = 31 measurement values are following)
0 0 0 0 890B 892......	=	measurement values in HEX (example: 890Bh = 35.083 = 35083mm = 35,083m)

This is the same for every other DIST string.

RSSI Values (Example: Echo 1) [Edit]

RSSI1 *3F800000 00000000 DBBA0 683 1F* 0 0 0 0 D 19 26 23 29 21 15 0 0 0 0 0 10 12 7 0 0 15 2E 36 38 37 33 33 29 14 28

RSSI1	=	now coming the radial distance values of the first pulse
3F800000	=	Scale Factor (3F800000 = factor 1; 40000000 = factor 2)
00000000	=	Scale Factor Offset
DBBA0	=	Start angle (DBBA0h = 900.000d = 90°)
683	=	Angular resolution (683h = 1.667d = 0,1667°)
1F	=	Amount of Data (1Fh = 31d = 31 measurement values are following)
0 0 0 0 D 19 26....	=	RSSI Values

This is the same for every other RSSI String.

End Header [Edit]

0 1 10 not defined 0 1 7B2 1 F F 27 1F D59F8 0

0	=	Position data (0 = no position data)
1	=	Name (1 = Name available)
10	=	Length of the name
not defined	=	Name
0	=	Comment (0 = no Command)
1	=	Time

7B2	=	Year
1	=	Month
F	=	Day
F	=	Hours
27	=	Minutes
1F	=	Seconds
D59F8	=	µsec
0	=	Event INfo (0 = no event info)

Find below the meaning and value range of several values from the data string.

Measurement range [Edit]

LMS1xx: Measuring the distance is only possible in the range from 0.5m to 20m. An obstacle which is more distant than 20m will result in a 0 in the measurement data.
Measurement closer than 0,5m to the scanner will result in floating values.

Up from Firmware V1.23, measurement is also possible below 0.5m

LMS5xx: Measuring the distance is only possible in a range up to 80m.

TiM31x: Measuring the distance is only possible in a range up to 4m.

Number of measurement values [Edit]

For 0.25 deg. resolution, there are 1082 measurement points.
The LMS is also giving out the angle for 225,25° .Therefore it's one shot more. That means our opening angle for 0,25° mode in LMS100 is 270,25°.
For 0.5 deg. resolution, there are 541 measurement points per a scan.

Timer [Edit]

Time since startup [Edit]

This timer starts running at the power up of the device, starting with 0; The actual value is put into the data telegram stream always when the mirror is at the zero index (-14° LMS5xx / -50° LMS1xx)
Time is in µs and can be used as a relative time to the scan before.
It is a temporary operation time counter.
The time stamp is taken depending on the zero-index of the code plate, that means it depends on the scan frequency of the scanner, so there is no possiblity to have an influence to the tolerance (±300µs)

Time of transmission [Edit]

This timer starts running at the power up of the device, starting with 0; The actual value is put into the data telegram when the complete scan data is transmitted to the buffer for data output. After that the data transmission to the host is started.
Time is in µs and can be used as a relative time to the scan before.

It is a timer that shows, that the actual scan is finished (LMS5xx: -5..185°; LMS1xx: -45..225°), the scan telegram is completed and the data is ready to be picked up from the internal data buffer (ready to be sent by the interface).

Time [Edit]

Is the time stamp from the internal time of the scanner but it is not a real time clock.

It is the time which can be set via command (LSPsetdatetime), in the LMS5xx also via NTP. It is also resetted with every reboot of the time, starting again with 01.01.1970, so the scanner does not keep the time.
It is not related to any scan telegram or other events.

This time information is added to the telegram, after all distance values are received.
After the complete scan is done (the scanner always scans 270° even if the data output range is smaller, so the data output range has no influence on that time stamp and the point in time where the telegram is sent), the generation of the measurement is finished and the all the values are sent to the internal application module. Whe the complete scan is received there, the time stamp is created and added to the scan.
There is no special angular degree when this timestamp is generated, it is done somewhere after the last measurement of the scan and the first measurement of the next scan.

During 24h, this time looses 4 sec.

It's unit is µs and the values are:

- uiYear
- usiMonth
- usiDay
- usiHour
- usiMinute
- usiSec
- udiUSec

In the telegram, the sequence is then:
Year:Month:Day:Hour:Minute:Second:Microsecond

To get the time stamp in the telegram, it has to be activated in the data processing page of the scanner.

Timeset with command: LSPsetdatetime
The Format in the telegram is +2009 +7 +22 +12 +0 +0 +0. It represents (year month day hour minute second microsecond) always with blank in between. If plus is used up-front the data its interpreted as an integer decimal number, without the plus it's the scanner reads the data as hex format. Answers come always in hex format.
The customer should use:

- sMN LSPsetdatetime 7D9 2 11 10 22 0 0 <--- (hex-format w/o sign)
- activate time stamp on page "data processing"

Counter [Edit]

Scan Counter [Edit]

The Scan Counter gives the number or scans the sensor made over all, doesn't matter if they are given out or not.

Telegram Counter [Edit]

Countes the scans that have been send from the scanner to a host. The number of sent scans can be parametrized by the "output Interval" in the data processing.

Status of the digital outputs [Edit]

The status is a HEX value which represents, converted into binary, the state of all the outputs.
Examples for LMS1xx:

HEX value	binary value	OUT3 state	OUT2 state	OUT1 state
0 0	000	0	0	0
1 0	001	0	0	1
2 0	010	0	1	0
3 0	011	0	1	1
4 0	100	1	0	0
5 0	101	1	0	1
6 0	110	1	1	0
7 0	111	1	1	1

Device Status [Edit]

describes the bit position and its meaning is:
00 00 – OK
00 01 – Error
00 02 – Pollution Warning
00 04 – Pollution Error
00 05 – Pollution and Device Error

in binary it is:

0000 - 0 ok
0001 - 1 Error (general device error (e.g. Motor Error) you will get the 0001 (1))
0010 - 2 Pollution Warning
0100 - 4 Pollution Error
0101 - 5 Pollution and Device Error (occurs like the pollution Error. You will get 0101 (5) because Pollution Error and the Device itself is in Error (Device not ready))

reserved values [Edit]

Valid measurement values are values starting from 16 upwards, everything below has the following meaning:

Value	RSSI	Description
0	0	no meas value detected; means that in the angle, there was no valid measurement value. Probably the object to measure was out of the range of the or the object was reflecting too less light back (black objects)
1	0xFFFF (16Bit output) 0xFF (8Bit output)	dazzled, geblendet
2	0	implausible meas values
3	0	value was set to invalid by a filter (never by Echo Fitler, only Particle Filter and old firmware)
4 – 15	0	reserved, at the moment not given out, if there occurs a value in that range anyway → perform a Softwareupdate
>16	>0	valid measurement values

Valid for LMS1xx/5xx, TiM5xx

max. measurement value LMS1xx: Dez: 20.000mm --> Hex: 4E20
max. measurement value LMS15x: Dez: 50.000mm --> Hex: C350
max. measurement value LMS5xx: Dez: 80.000mm --> Hex: 13880

Higher measurement values will be given out with a zero, that means no measurement value detected.

RSSI [Edit]

Amount of energy of the reflected laser pulse. (That are no remission values, because the remission is a property of the measured object and indicates how much % of the incoming laserlight is reflected back.)
The maximung value does not show a dazzling, also not a reflection by a reflector. The maximum value will never be reached, every value will ever be in the range
The value has no unit, its just digits.

8bit à max. RSSI value: 255
16 bit à max. RSSI value: 4095 (in reality there are only 12 bit values used, but as there are no 12 bit data types available in programming, just not the whole variable is used).
0x00 Mean there was no Echo from the Target at all.
0xff and distance 1 means the sensor was dazzeled
512 is a good value for white objects, a value of 3600 has already been seen from a reflector.

The range of values is the same in both resolutions, but with 16bit resolution you have a finer granularity, means you have more values in the same scan range, which makes your scan result more accurate.
The dependency of the value is strictly monotone, the more light comes back, the higher is the value, but it is not linear.

The value is made for internal measurement corrections but it can also be used by any other user, but there will be no further informations and not "how to use" about it.
The values are not adjusted, that means you can get different values from the same object with two different devices, we do not guarantee for that values. That are internal values but usable by the customer

if necessary.
A reference target measurement is not available for the RSSI values.

RSSI data in the LMS1xx are 8bit (0-255) or 16bit (0-65535)
RSSI data in the LMS5xx are 8bit (0-255)
RSSI data in the TiM5xx are 8bit (0-255)

Bugfixes:
There was a bug of displaying the 8-bit RSSI-values in the V1.23 of the LMS1xx but it was fixed in the V1.31

Remark LMS5xx:
The RSSI datas are available in the LMS5xx since the FW V1.10 (March 2011 / T7 Status of the device)

8bit/16bit [Edit]

MS1xx:
Measurement values are always given out in 16bit.
RSSI values can be chosen as 16 or 8 bit values.

LMS5xx:
Measurement values are always given out in 16bit.
RSSI values are always given out in 8 bit values.

General:
Only the RSSI values are possible as 16 or 8 bit values (but not in the LMS500) Measurement is always 16bit.
That means that 16 bit gves a more detailed resolution for the RSSI values. For 16bit the highest value ist FFFFh and for 8bit it is FF
Finally the range stays the same ---> 16bit FFFFh = 8bit FFh
But as the range is wider, the resolution is higher.

Output Channels [Edit]

Channel 1 = Pulse 1
Channel 2 = Pulse 2
Channel 3 = Remission 1
Channel 4 = Remission 2

Scale Factor [Edit]

Scale factor [Edit]

The scale factor describes you how to interpret the measuring data. It is a fixed value which can't be changed and you will receive it in the telegramm with the distance data. In this case it says that one increment of the measuremet data is equal to 1 mm.

The value in the measurement string is a 32-Bit Floating Point Value, the shown hex value is exactly 1,000000 if it is converted.

Scale factor offset [Edit]

The scale facto offset is to use like the the scalefactor of the distance. (just called "Scale factor"). In distances >65m, the are scaled with the factor 2,0. The offset has to be added to the distance values.The data type is "real" and has to be converted (see SOPAS Protocol Specification CoLaA, chapter Data Types)

Encoder information [Edit]

If „fixed speed" is parametrized for the encoder speed, there will be only the speed in the telegram, not the increments. If there is a Encoder connected, the increments will also be there.

The encoder data are given out when it is parametrized on page „data processing" in Sopas.
Voltage is similar to the normal inputs, 24V should work.
Encoder datas are generated at the end of every scan between the 225° and 270° shot and added to the actual telegram.
Encoders to be used are HTL types because the LMS devices are working with 24V at the encoder input (The actual typ series from SICK is the DFS60)

Encoder recommendation:
2058475 Encoder DFV60A-W2EZ0-S01 M12-5Pol. Resolution 10mm
2058476 Encoder DFV60A-W2EZ0-S02 M12-5Pol. Resolution 0,5mm

Event Info, Comment, Position [Edit]

The Event Info part is empty for the LMS1xx, it is not filled with data by the scanner. The LMS5xx gives an event info.
The Comment part is empty, it is not filled with data by the scanner and nothing can be written into it by the user.
The position information (X,Y,Z) will be empty in TiM, LMS1xx, LMS5xx.

Buffer [Edit]

There is no buffer that buffers scans for later output or similar. The scans are given out immediately and the following scan deletes the one before.

Point of time for data output [Edit]

After the scan is complete (270° scan is done), the measurement data is delivered from the measurement module to the application module. There are another 3ms go by until the compensatin and filters are done. Than additionally the time for sending the telegram has to be added. After that the field evaluation and the output of the evaluation result is done. So all in all about 10ms after the last measurement point, the data telegram reaches the ethernet port of the client.
Everything has to be finished before the next measurement data from the measurement module is coming in.

If a single scan telegram is asked, always the values from the last completed scans were sent.

Triggered scan data output:
The timing is the same as described above but in this case the ditigal input determines if the telegram is given out. That is checked every time a new complete scan is available, that means, the most important factor for the data output in this case is the availability of a finished scan, so timing is the same as above.

If the digital input is set right in that momentthe actual scan is given out, than the maximum time of one scan goes by until the next scan is given out belonging to the digital input.

Some measurement results:

The measurements were done for the worst case.
- capturing of measurement values: fog filter active (incl. SPS-Filter)
- 1 Field 270°, 10 evaluation cases "Pixel", Distance dependent
- Scan data output interval: 1; all channels active
- output via ethernet (data output via serial interface all times are shorter.)
- 0.5°@50Hz

Times:
- receive the measurement data after the scan from the measurement module: 3,5 ms
- data output via Ethernet: 5,0 ms
- field evaluation and output of the result: 7,5 ms
Total time: ca. 16 ms

Notes:
All the times are no fix values, in other conditions (less evaluation cases, no filter ....) the times are lower.
The worst case is the case when he has fewest processing time.
If 25Hz and 0,25° is used, the times for receiving the data from the measurement module will grow (by using the same settings) because the double amout of data has to be handled. On the other side because of the cut in a half frequency, there is the double of time for the processing.

Format of 2 byte values (motorola format) [Edit]

If there are values coming in two parts (for example the outputs in the measurement telegram documented as: 00 07, output will be 07 00; LSB first, than MSB)

Data amount per telegram [Edit]

Example how to calculate the amount of data for a measurement telegram.

Size of one measurement value:	5 Byte	(4 Byte for the value itself, 1 Byte for the blank after the value)
Size of the telegram header:	81 Byte	
Size of the telegram end:	12 Byte	

Calculation of number of Measurement values depends always on the resolution:
0.5° = 2 measurements per degree
0.25° = 4 measurements per degree
Always one additional measurement for the last measurement

Number of measurement Values = Number of degrees × measurements per degree + 1

Example for measuremet of 56° in 0.5° resolution: $56 \times 2 + 1 = 113$ Measurement values

Amount of Data for this measurement values: 113 × 5 Byte = 565 Byte

Calculation of amount of data per telegram:

Data of one Telegram = Header + Measurements + end of telegram

81 Byte + 113 Measurements + 12 Byte

81 Byte + (113 × 5Byte) + 12 Byte = 658 Byte per Telegram (= 5264 Bit (658 × 8 Bit))

Possible amout for delivery with special Speed:

Number of telegrams per second = Speed ÷ telegram size

Speed Example: 115200 Bit/s = 11520 Byte/s = 11,52 Byte/s

11520 (Byte/s) ÷ 658 Byte = 17,5 Telegrams/s

Telegram size with **0,25°** resolution:
Degrees: 270°
Resolution: 0.25°

-» Measurement Values = 270 × 4 + 1 = 1081

Data per Telegram = 81 Byte + (1081 × 5 Byte) + 12 Byte = **5498 Byte** (= 43984 Bit)

Telegram size with **0,5°** resolution:
Degrees: 270°
Resolution: 0.5°

-» Measurement Values = 270 × 2 + 1 = 541

Data per Telegram = 81 Byte + (541 × 5 Byte) + 12 Byte = **2798 Byte** (= 22384 Bit)

Usage of PLC's [Edit]

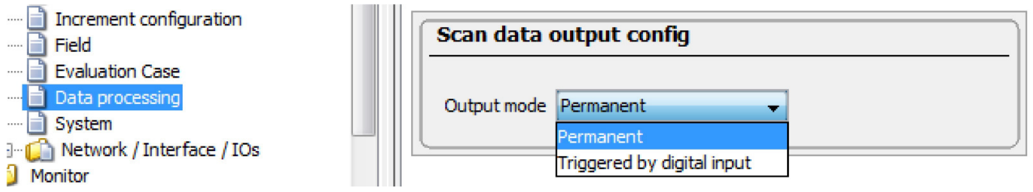
Using a PLC for scan data handling is not recommended, because for using full telegrams at maximum scan speed, the PLC normally is too slow. If you want to use it anyway, the complete handling has to be done by the customer. We only deliver the telegram by ethernet.

Data Output via CAN [Edit]

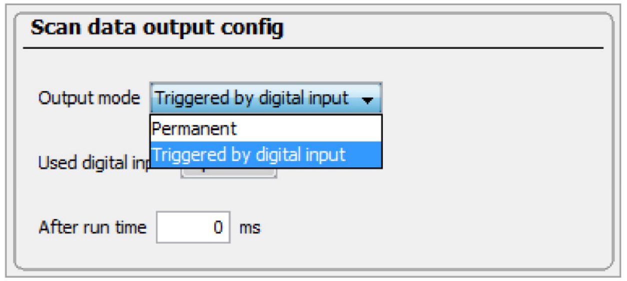
The data output via CAN is NOT possible. The CAN of the LMS is only for the external I/O module.

Data Output Trigger [Edit]

In newer Firmware Versions in the LMS1xx and LMS5xx is it possible to set trigger conditions for the dataflow.
The dataflow will only flow when an input trigger is active.

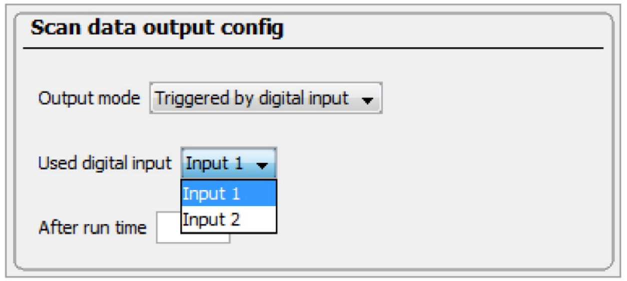


Output mode Permanent



Output mode Triggered by digital input

After run time 0 ms



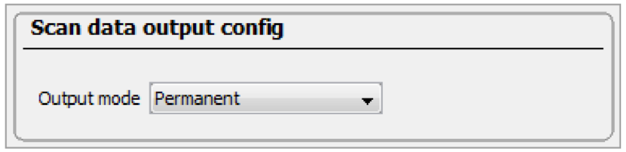
Output mode Triggered by digital input

Used digital input Input 1

But keep in mind:
the "sEN LMDscandata 1" telegram has once to be send to the scanner. After that the function will run.

The scanner will never automaticly after the boot-up process and activation of the trigger sending datas to the evaluation unit like PC/Laptop etc. (without the starting telegram (sEN LMDscandata 1)) !
Always once after the boot-up, the telegram has to be send!

If you don't one any trigger conditions, choose "permanent" to get the datas directly after sending the telegram "sEN LMDscandata 1"


Scan data output config

Output mode: Permanent

Created at 08.03.2012 14:45 by Markus Wagner (UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
Last modified at 02.07.2015 12:11 by Mirko Vlasic (UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna Total Views: 1359

Rating : Ratings & Comments (Click the stars to rate the page)

Tags: data format;time since startup;Time;time of transmission;measurement values;RSSI;Output channels;scale factor;encoder informations;Event info;comment;position;ASCII;Binary;data output;Number of measurement values;1081;1082;541;Telegram Size;Data Amount;Example Scan telegram;point of time;PLC

Recent Discussions
There are no items to show in this view.