

PSI5 Sensor Programming -Seskion GmbH-

Content:

- Preparation for the diagnostic mode
- Settings for the programming mode of the sensor
- Opening the diagnostic mode
- Bidirectional communication

Version:	(1.0) 05.11.2021 – Creation
	(1.1) 21.12.2021 – Small improvements



Sensors have a diagnostic mode. This mode is defined by the sensor manufacturers and is not standardized. If a sensor is set to diagnostic mode, it provides data from defined addressing ranges of the manufacturer. The diagnostic mode is activated by specific data sequence that the ECU sends to the sensor.

The diagnostic mode of the sensors is a very specific section and should only be used by users with appropriate knowledge.

The Simulyzer diagnostic mode allows:

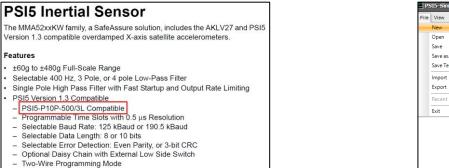
- put a sensor into dignosticmode,
- send commands to the sensor
- read out the memory cells of the sensor,
- overwrite the memory cells.

The following range of functions is available:

- Opening the diagnostic mode
- Bidirectional communication
- Memory data
- Configuration of the diagnostic memory.

Preparation for the diagnostic mode

In order to be able to work with the Simulyzer software, a new project must be created at the beginning after connecting the Simulyzer and the sensor. To do this, use the menu group *File* and the command *New* and *New Project*. Which version your PSI5 sensor has you can read in the features from the manual of the respective sensor. In this example a NXP sensor of the MMA52xxKW family with a PSI5-P10P-500-3L compatibility is used.

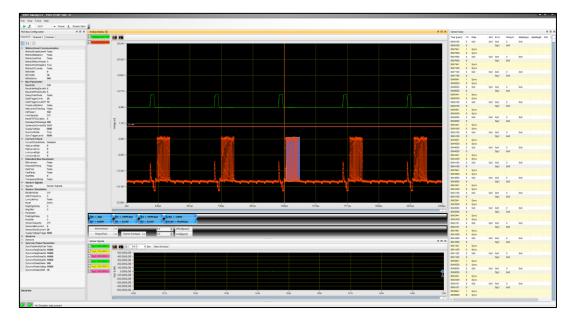


PSI5-Simulyzer		
File View Tools H	elp	
New 🕨	New Project	PSI5-S10P-500_4H
Open		PSI5-S10P-500_3L
Save		PSI5-P20CRC-500_3H
Save as		PSI5-P20CRC-500_2L
Save Template		PSI5-P16CRC-500_3H
Import		PSI5-P16CRC-500_2L
Export +		PSI5-P10P-500_4H
Recent File		PSI5-P10P-500_3L
		PSI5-P10P-500_2L
Exit		PSI5-P10P-250_1L
		PSI5-P10CRC-500_2L
		PSI5-A8P-250_1L
		PSI5-A16CRC-500_1L
		PSI5-A10P-250_1L
		PSI5-A10CRC-250_1L

Next, start a measurement by first selecting the ECU mode and then selecting the green arrow and Power.



You will now see that on the right side the individual data with time stamp are listed in tabular form. In the middle this is graphically represented.





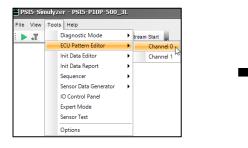
To stop the measurement, press the *red symbol* and then the *power* button to end the measurement completely.

PSI	5-Sim	ulyzer - PSI	5-P1	OP-500	_31	
File	View	Tools Help				
: .	-	ECU	•	Power	*	Stream Start

Settings for the programming mode of the sensor

To enter the programming mode of the sensor it is necessary to select the settings correctly. The correct settings for your sensor can be found in the respective data sheet.

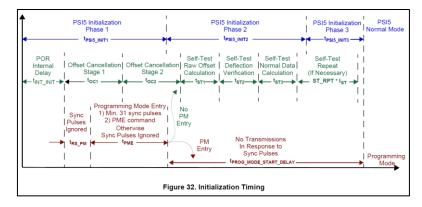
To set the correct settings in the program, select in the menu group **Tools** the command **ECU Pattern Editor**. Then select **Channel 0** or **Channel 1** depending on which channel your sensor is connected to.



ommand	Delay	Value	BitCount	Add
attern	0x2	0x2F8F	0xF	Delete
				Save
				Load
				Send
				Calculate
				Copy to List
2F8F				
2 0x7	0x0	0x7		

In the ECU Pattern Editor you will find a predefined example, which you have to update with your data. First check the **Use** *for Sensor Init* checkbox at the top.

To get into the programming mode of the sensor, you first have to set the Delay and Value correctly. You can find the Delay as shown here for example in a graphic in the sensor datasheet. This indicates that the delay must be at least 31 sync-pulses long. Therefore "0x1f" is entered at Delay, because 31 = 1f in hexadecimal.



You also set the correct value with data from the data sheet. The values are listed as hexadecimal and must be converted to decimal values. Here the SAdr has the decimal value = 1 and FC has the decimal value = 7. You can set these values in the ECU Pattern Editor at the bottom left and then copy the resulting value into the correct cell.

#	CMD				FC Command	Regis-	Data		Response (OK)	Response (Error)		
#	Туре	SAdr	FC	Command	ter Address	Field	RC	RD1	RD0	RC	RD1	RD0	
S0	Short		100	Execute Programming of NVM	N/A	N/A	ОК	0x2AA	N/A	Error	ErrN	N/A	
S1	Short		101	Invalid Command	N/A	N/A	No Response		No Response				
S2	Short		110	Invalid Command	N/A	N/A	No Response		No Response				
S3	Short		111	Enter Programming Mode	N/A	N/A	ОК	0x0CA	N/A		No Response		
LR	Long	001	010	Read nibble located at address RA5:RA0	Varies	Varies	ок	RData	RData+1	Error	ErrN	0x00	
LW	Long		011	Write nibble to register RA5:RA0	Varies	Varies	ОК	WData	RA5:RA0	Error	ErrN	0x00	
XLR	XLong		000	Invalid Command	Any	Any	No Response		nse		No Respor	nse	
KLW	XLong		001	Invalid Command	Any	Any	No Response		nse	No Response			

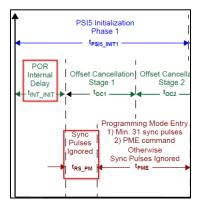


Make sure that the number above the respective description (SAdr / FC) must have the same decimal value as previously determined. The hexadecimal number cannot be entered 1:1 below it. For this example, the ECU Pattern Editor looks like this:

Command	Delay	Value	BitCount	Add
Pattern	0x1f	0x2CF9	0xF	Delete
		/		Save
	/	/		Load
				Send
				Calculate
/				Copy to List
x2CF9				
0x2 0x1	0x7	0x1		

To close the editor, click Close in the upper right corner.

To enter the programming mode, the values "POR Internal Delay" and "Sync Pulses Ignored" in the data sheet must be read out and added together to determine the advance.



The synchronization pulse is specified with at least 58 milliseconds for this sensor. The Internal Oscillator Frequency is 4 MHz = 0.00025 ms and is calculated into the Internal Delay, so the Internal Delay = $\frac{16.000ms}{1/_{0,00025 ms}}$ = 4 ms. Together, this is now 62 milliseconds.

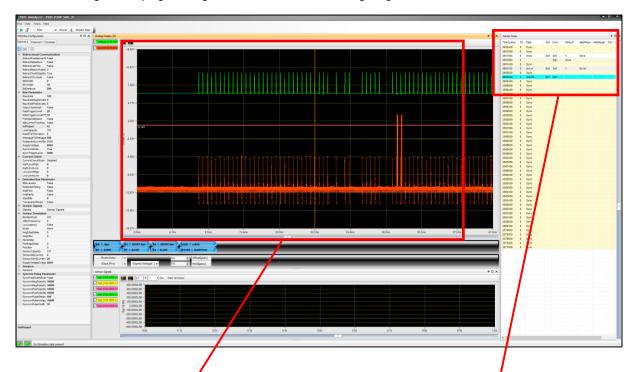
2.6 V _L ≤	2.6 Dynamic Electrical Characteristics - PSI5 $I_L \le (V_{CC} - V_{SS}) \le V_H$, $T_L \le T_A \le T_H$, $\Delta T \le 25$ K/min, unless otherwise specified										
#	Characteristic	Symbol	Min	1	Гур	Мах	Units				
104	Synchronization Pulse (Figure 5, Figure 28 and Figure 32) Reset to first sync pulse (Program Mode Entry)	trs_pm	58		_	_	ms				
2.7 ∨ _L ≤	2.7 Dynamic Electrical Characteristics - Signal Chain $V_L \leq (V_{CC} - V_{SS}) \leq V_H, T_L \leq T_A \leq T_H, \Delta T \leq 25$ K/min, unless otherwise specified										
#	Characteristic		Symbol	Min	Тур	Max	Units				
138	Internal Oscillator Frequency	*	f _{OSC}	3.80	4	4.20	MHz				
2.8 / _L ≤	$\label{eq:product} \begin{array}{l} \textbf{Dynamic Electrical Characteristics} \\ (V_{CC} - V_{SS}) \leq V_{H}, \ T_{L} \leq T_{A} \leq T_{H}, \ \Delta T \leq 25 \ \text{K/min, unless} \end{array}$										
#	Characteristic	s	Symbol	Min	Тур	Мах	Units				
177	Reset Recovery Internal Delay (After internal POR)	t		—	16000 / f _{OS}	- c	s				

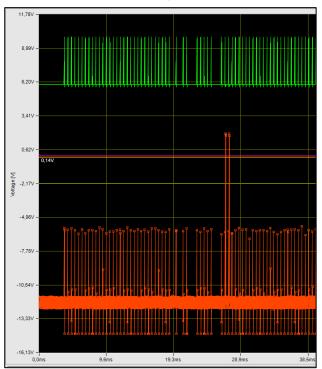


The 62 milliseconds are entered on the left in the software under Bus Parameter at "InitPhase1".

~	Bus Parameter	
	Baudrate	125
	BaudrateNegDeviatio	5
	BaudratePosDeviatic	5
	DaisyChainMode	False
	DataTriggerLevel	25
	DataTriggerLevelOff	10
	FrameLostDetect	False
	IdleCurrentTracking	False
	InitPhase1	62
	LineCapacity	Off
	MaxMTMTDeviation	5
	MessageToMessage	500
	QuiescentCurrentSe	5000
	SupplyVoltage	6000
	SyncronMode	True
	SyncTriggerLevel	6500

Now start the measurement again with the *green arrow* and the *power* button. The sensor now goes into programming mode and returns the following image:





Sensor Data								4 (
Time [µsec]	Ch	Data	Slot	Error	Parity/C	dataRegA	dataRegB	fctrl
2362968	0	Sync						
2362468	0	Sync						
2361968	0	Sync						
2361630	0	0xca	0x0	0x0	0	0xca		
2361630	0			Sig1	0xca			
2361468	0	Sync						
2361129	0	0x1e1	0x0	0x0	1	0x1e1		
2360968	0	Sync						
2360468	0	0x2cf9	0xf	0x0				
2360468	0	Sync						
2358968	0	Sync						
2358468	0	Sync						



Opening the diagnostic mode

Select the menu group *Tools* and the command *Diagnostic Mode* and click in the subgroup on the desired *channel*, which you want to program.

PSI5-Si	nulyzer		
File View	Tools Help		
	Diagnostic Mode	•	Channel 0
	ECU Pattern Editor	►	Channel 1
	Init Data Editor	→	
	Init Data Report	→	
	Sequencer	•	
	Sensor Data Generator	→	
	IO Control Panel		
	Expert Mode		
	Sensor Test		
	Options		

Bidirectional communication

The communication between ECU and sensor is displayed as logging. Here the menu item *Bidir Logging* shows the same time stamp and ECU pattern as in the right table.

Diagnostic		ration					
Timestamp[µs] 2360468	ECU pattem 0x2cf9	Type short	RC[0] 0x1e1	RD2[0] 0x0	RC[1] 0x0	RD1[1] 0x0	Clear
<							

To find out the programmability of your sensor, you must first find out the CMD type, whether this is a short or long type. It is also important with which FC can be read or written.

5.3.6	Pro	gram	min	g Mode Via PSI5 Commai	nd and	Resp	onse	Summa	ary					
Table	Table 17. Programming Mode Via PSI5 Commands and Responses													
#	CMD	SAdr	FC	Command	Regis- ter	Data		Response (OK)	Response (Error)				
#	Туре	SAdr		Command	Address	Field	RC	RD1	RD0	RC	RD1	RD0		
S0	Short		100	Execute Programming of NVM	N/A	N/A	OK	0x2AA	N/A	Error	ErrN	N/A		
S1	Short		101	Invalid Command	N/A	N/A	No Response		No Response		No Response		ise	
S2	Short		110	Invalid Command	N/A	N/A	No Response		nse	No Response		No Response		
S3	Short		111	Enter Programming Mode	N/A	N/A	OK	0x0CA	N/A		No Response			
LR	Long	001	010	Read nibble located at address RA5:RA0	Varies	Varies	ок	RData	RData+1	Error	ErrN	0x000		
LW	Long		011	Write nibble to register RA5:RA0	Varies	Varies	ок	WData	RA5:RA0	Error	ErrN	0x000		
XLR	XLong		000	Invalid Command	Any	Any		No Respo	nse		No Respor	ise		
XLW	XLong		001	Invalid Command	Any	Any		No Respo	nse		No Respor	ise		
Note:	When I	reading	the la	ast address in the data array, RD	ata+1 wi	ll alway	s returr	ם 0x00.						



Here is a small explanation how the short or long frame commands look like at least for this sensor:

υ.		.2.1																mat				t a al				6					Orthur	
	st	ructi	ons	are	e pe	rforn	ned	in r	resp	ons	e to	sho	ort fra	ame	cor	nm	ands	s. The														pecific rt Fram
CC	on	nma	nds	and	d re	spon	ses	are	e de	fine	d in	Sec	tion	5.3	.6, T	abl	e 18															
								\$	Start E	Bits			Senso Addre			Fu	nction	Code			CRC			F	lesp	ons	•					
								S2	2 S1	S0	Sy	SA0	SA1	SA2	Sy	FC	FC1	FC2	Sy	C2	C1	C0	1	R	2	RI)1					
								0	1	0	1	1	0	0	1	0	0	1	1	0	0	0	1	\$1	2	\$3	FF					
5.	3	.2.2			Lor	ng F	ran	ne	Cor	nm	and	lan	nd R	es	oon	se	For	mat														
		ong	frai			·														сT	he (dev	ice	can	nr	ovi	le r	iner	ste	r data	in res	ponse t
			na																													
3			or w	rite	rea	uest	Th	eL	ona	Fra	me	torm	าสบาร	s sn	own	In	-101	re 44	- 10													fined in
	re				req	uest	. Th	e L	ong	Fra	me	form	atis	s sn	own	IN	-igu	re 44	. LC	'ny	Fra	me	COI				nu	163	por			fined in
	re	ead			req	uest	. Th	e L	ong	Fra	me	form	atis	s sn	own	IN	-igu	re 44	. LC	, ng	Fra	me	COI				nu	103	-			fined in
	re eo	ead	5.3			uest Sensor	r	e L		Fra		form		_	egiste	_		re 44			Fra	Data	_			CF	_				Respons	
	re ec	ead of the start	5.3 Bits	.6.	Å	Sensor	r is		Func	tion	Code			R	egiste	er Ad	dress	RA4 R				Data	1			CF	c			RC		

Figure 44. Programming Mode Via PSI5 Long Command and Response Format

If you now look again in the manual, then you can find a table with data, which shows you bit functions read and write. With the *Nibble Addr* you can get the respective functions in the Configuration Mode. The *Type* indicates whether you have *read or write rights* in the function.

3 Functional Description

3.1 User Accessible Data Array

.....

.

A user accessible data array allows for each device to be customized. The array consists of an OTP factory programmable block, an OTP user programmable block, and read only registers for device status. The OTP blocks incorporate independent CRC circuitry for fault detection (reference Section 3.2). Portions of the factory programmable array are reserved for factory-programmed trim values. The user accessible data is shown in Table 2.

Byte Addr		Nibble Addr		Bit Fu	nction		Nibble Addr	Bit Function					
(XLong Msg)	Register	(Long Msg)	7	6	5	4	(Long Msg)	3	2	1	0	Ту	
\$00	SN0	\$01	SN[7]	SN[6]	SN[5]	SN[4]	\$00	SN[3]	SN[2]	SN[1]	SN[0]		
\$01	SN1	\$03	SN[15]	SN[14]	SN[13]	SN[12]	\$02	SN[11]	SN[10]	SN[9]	SN[8]		
\$02	SN2	\$05	SN[23]	SN[22]	SN[21]	SN[20]	\$04	SN[19]	SN[18]	SN[17]	SN[16]	F	
\$03	SN3	\$07	SN[31]	SN[30]	SN[29]	SN[28]	\$06	SN[27]	SN[26]	SN[25]	SN[24]		
\$04	DEVCFG1	\$09	0	0	1	0	\$08	0	RNG[2]	RNG[1]	RNG[0]		
\$05	DEVCFG2	\$0B	LOCK_U	PCM	SYNC_PD	LATENCY	\$0A	DATASIZE	BLANKTIME	P_CRC	BAUD		
\$06	DEVCFG3	\$0D	TRANS_MD[1]	TRANS_MD[0]	LPF[1]	LPF[0]	\$0C	TIMESLOTB[9]	TIMESLOTB[8]	TIMESLOTA[9]	TIMESLOTA[8]		
\$07	DEVCFG4	\$0F	TIMESLOTA[7]	TIMESLOTA[6]	TIMESLOTA[5]	TIMESLOTA[4]	\$0E	TIMESLOTA[3]	TIMESLOTA[2]	TIMESLOTA[1]	TIMESLOTA[0]		
\$08	DEVCFG5	\$11	TIMESLOTB[7]	TIMESLOTB[6]	TIMESLOTB[5]	TIMESLOTB[4]	\$10	TIMESLOTB[3]	TIMESLOTB[2]	TIMESLOTB[1]	TIMESLOTB[0]	U	
\$09	DEVCFG6	\$13	INIT2_EXT	ASYNC	U_DIR[1]	U_DIR[0]	\$12	U_REV[3]	U_REV[2]	U_REV[1]	U_REV[0]		
\$0A	DEVCFG7	\$15	MONTH[3]	MONTH[2]	MONTH[1]	MONTH[0]	\$14	YEAR[3]	YEAR[2]	YEAR[1]	YEAR[0]		
\$0B	DEVCFG8	\$17	CRC_U[2]	CRC_U[1]	CRC_U[0]	DAY[4]	\$16	DAY[3]	DAY[2]	DAY[1]	DAY[0]		
\$0C	sc	\$19	0	TM_B	RESERVED	IDEN_B	\$18	OC_INIT_B	IDEF_B	OFF_B	TEMPF_B		

U: User programmable OTP location via PSI5

R: Readable register via PSI5



In *Diagnostic Mode* the Long Frame Command can be set in the *Configuration*. It works exactly the same as in the ECU Pattern Editor with Copy/Paste. With *Add* you can add a new command or you can delete it with *Delete*. If you want to save a command, you can name it where "Read_0" is now and save it with *Save Command*.

Pattern 0x1f 0xB3AB3F7 0x1D Delete Save Load Remove Command Save Command Read_0 ▼ Copy to List	Command	Delay	Value	BitCount	Add
KB3AB3F7 KB 3AB3F7 K2 0x1 0x3 0xA 0xF 0x3 Sat Saf FC RAdr Data CRC					Delete
kB3AB3F7 k2 Dot1 0x3 0xA 0xF 0x3 Start SAr FC RAdr Data CRC					
Remove Command Save Command Read_0 Copy to List X2 Dx1 0x3 0x4 Data CRC			7		
Save Command Read_0 Copy to List 0x83AB3F7 0x2 0x1 0x3 0x4 0x6 0x5 Start SAdr FC RAdr Data CRC		1			
Read_0 Copy to List 0x2 0x1 0x3 0xA 0x4 0xF 0x3 0xA					Remove Command
Copy to List 0x83A83F7 0x2 0x1 0x3 0xA 0xF 0x3 Start SAdr FC RAdr Data CRC					Save Command
0x83A83F7 0x2 0x1 0x3 0xA 0xF 0x3 Start SAdr FC RAdr Data CRC		/			Read_0 👻
					Copy to List
0x2 0x1 0x3 0xA 0xF 0x3 Start SAdr FC RAdr Data CRC	/				
Start SAdr FC RAdr Data CRC	B3AB3F7				
0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1					
	0 1 0 1 1 0	0 1 1 1 0	101011001	1 1 1 1 1 0 1 1	1

To start it, you have to first selecting the *ECU mode* and then selecting the *green arrow* and *Power* again. Now you can find the individual commands in the *Bidir Logging* in the right column. By clicking on them, these commands will be run. With *Clear* you can delete the left logging.

Clear Clear Clear 22730 0x2a49 short 0x1e1 0x2a 0x0 0x0	imestamp[µs]	ECU pattern	Туре	RC[0]	RD1[0]	RD2[0]	RC[1]	RD1[1]	
51146 0xb2a3222 long 0x1e1 0x2 0x0 0x0 0x0 0x0 82646 0xb2ab22e long 0x1e1 0x0 0x0 0x0 0x0 82646 0xb2ab22e long 0x1e1 0x7 0xa 0x0 0x0 45646 0xb2ab22e long 0x1e1 0x7 0xa 0x0 0x0 29231 0xb2ab22e long 0x1e1 0x7 0x0 0x0 0x0 2021 0xb2ab22e long 0x1e1 0x7 0x0 0x0 0x0 829231 0xb2ab22e long 0x1e1 0x7 0x0 0x0 0x0									Clear
B2646 Oxb2ab22e Iong Ox1e1 0x0 0x0 0x0 0x0 0x0 Read_0 14116 Oxb2ab327 Iong Ox1e1 Oxf No 0x0 0x0 Read_1 5456 Oxb2ab22e Iong Ox1e1 Oxf 0x0 0x0 0x0 Read_1 28231 Oxb2ab22e Iong Ox1e1 0x7 0x0 0x0 Read_2 28231 Oxb2a3222 Iong Ox1e1 0x7 0x0 0x0 Read_2 28231 Oxb2a3222 Iong Ox1e1 0x7 0x0 0x0 Read_4 28231 Oxb2a3577 Iong Ox1e1 0x7 0x0 0x0 Read_4 22010 Oxb3a3677 Iong 0x1e1 0x7 0x0 0x0 0x0									
14116 0xb3ab3r7 long 0x1e1 0xf 0xa 0x0 0x0 Read_1 45546 0xb2ab22e long 0x1e1 0xf 0x0 0x0 0x0 Read_2 29231 0xb2ab22e long 0x1e1 0x7 0x6 0x0 0x0 Read_2 0731 0xb2ab22e long 0x1e1 0x7 0x0 0x0 Read_4 92201 0xb3ab3r7 long 0x1e1 0x7 0x0 0x0 Read_4									Read_0
45646 0xb2ab22e long 0x1e1 0xf 0x0 0x0 0x0 Read_2 29231 0xb2ab22e long 0x1e1 0xf 0x0 0x0 0x0 Read_2 20731 0xb2ab22e long 0x1e1 0xf 0x0 0x0 0x0 Read_4 22011 0xb3ab3f7 long 0x1e1 0xf 0x0 0x0 0x0			-						Read 1
29231 Oxb2a3222 long Ox1e1 Ox2 Oxf Oxf Ox0 Ox0 Read_4 50731 Oxb2ab22e long Ox1e1 Oxf Ox0 Ox0 Ox0 Read_4 92201 Oxb3ab3f7 long Ox1e1 Oxf Ox0 Ox0 Ox0			-						
S0731 Oxfo2ab22e long Ox1e1 Oxf Ox0 Ox0 Read_4 92201 0x63ab3f7 long 0x1e1 0xf 0xa 0x0 0x0									Read_2
92201 0xb3ab3f7 long 0x1e1 0xf 0xa 0x0 0x0			-						Read_4
			-						
			-						
	3323/31	0xb2ab22e	long	0x1e1	Uxt	0x0	0x0	0×0	
								>	

For a better overview in the Sensor Data you can set the *SyncronMode* in the Bus Configuration on the left on *False*. As a result, the Syncron pulses are not displayed in the table.

PSI	5 Bus Configuration		4 0
Cha	nnel 0 Channel 1 (Common	
•	2↓ 🖾		
~	Bus Parameter		
	Baudrate	125	
	BaudrateNegDeviatio	5	
	BaudratePosDeviatic	5	
	DaisyChainMode	False	
	DataTriggerLevel	25	
	DataTriggerLevelOff	13	
	FrameLostDetect	False	
	IdleCurrentTracking	False	
	InitPhase1	65	
	LineCapacity	Off	
	MaxMTMTDeviation	5	
	MessageToMessage	500	
	QuiescentCurrentSe	5000	
	SupplyVoltage	6000	
	SyncronMode	False	×
	SyncTriggerLevel	6500	3



Sensor Data								,	
Time [µsec]	Ch	Data	Slot	Error	Parity/C	dataRegA	dataRegB	fctrl	ľ
19883822	0	0x43	0x0	0x0	1	0x43			J
19883822	0			Sig1	0x43				
19883322	0	0xa2	0x0	0x0	1	0xa2			
19883322	0			Sig1	0xa2				
19882822	0	0x1e1	0x0	0x0	1	0x1e1			
19882191	0	0xb2aa226	0	0x0					
16958037	0	0x0	0x0	0x0	0	0x0			
16958037	0			Sig1	0x0				
16957537	0	0xf	0x0	0x0	0	0xf			
16957537	0			Sig1	0xf				
16957036	0	0x1e1	0x0	0x0	1	0x1e1			
16956405	0	0xb2ab22e	0	0x0					
16926537	0	0xa	0×0	0x0	0	0xa			
16926537	0			Sig1	0xa				
16926037	0	0xf	0×0	0x0	0	0xf			
16926037	0			Sig1	0xf				
16925536	0	0x1e1	0×0	0x0	1	0x1e1			