



# Specification

STX/ETX Protocol

4.73

Project name <b>STX/ETX Protocol</b>		Version <b>4.73</b>	
	Responsible <b>Harald Lenz</b>	Last revision <b>06.05.15</b>	Editor <b>Jürgen Kalbitzer</b>
Printout		No. of pages incl. cover sheet <b>295</b>	
Path C:\projekte_svn\stxetx\trunk\public\StxEtx.odt			
Document number <b>0000.000.000</b>			
Signature Project Manager	Development Manager		Managing Director

## Contents

<b>1 Basic Protocol Format.....</b>	<b>17</b>
1.1 Notational Remarks.....	17
1.2 General Transmission Structure.....	17
Data Framing.....	17
Control Character Definition .....	18
1.3 Host Request to RF System.....	18
Function Calls.....	18
Interruption.....	19
1.4 RF System Response to Host.....	19
Positive Acknowledgement.....	19
Negative Acknowledgement.....	20
No Acknowledge.....	20
Multiple Responses.....	21
<b>2 Command Description.....</b>	<b>22</b>
2.1 1xxx: System Control.....	22
1000: Reset Request .....	22
1001: Request Version Number.....	22
1002: Interface Test.....	23
1003: Change Baud Rate.....	23
1004: Set Byte Timeout.....	24
1006: Request Log Data.....	24
1008: Switch RF System to Low-current Mode.....	25
100A: Request Supported Transponder Types.....	25
100B: Select Transponder Types for General Functions.....	26
100E: Request Supported Function Numbers.....	27
1010: Request System Setting.....	27
1011: Edit System Setting.....	28
1014: Field Analysis.....	28
1015: Tune Antenna.....	29
1016: Get Reader ID.....	30
1017: Set Reader ID.....	30
1018: Flush Buffers.....	30
1019: Get Device Serial Number.....	31
101A: Sync Listen.....	31
101B: Ping Sync Slave.....	32
101C: Pingback Sync Slave.....	32
101D: Get Reader ID List.....	33
101E: Complete Sync Test.....	33
1020: Get Command Buffer.....	34
1021: Set Command Buffer.....	35
1022: Trigger Command Buffer.....	36
1023: Enable Command Buffer.....	36
1026: Get Port Configuration.....	36

1028: Get Processor Identification String.....	37
1029: Get Hardware Feature.....	37
102A: Request Interface Board Setting.....	38
102B: Edit Interface Board Setting.....	40
102C: Forward Command.....	40
102D: Request Interface Board Version Number.....	41
102E: Get Local Device Name.....	41
102F: Set Local Device Name.....	42
1068: Get SmartRead Configuration.....	42
1069: Set SmartRead Configuration.....	42
106A: Clear SmartRead Configuration.....	43
1070: Get Tag Buffer Item.....	43
1071: Add Tag Buffer Item.....	44
1072: Get Tag Buffer Item Count.....	44
1073: Clear Tag Buffer.....	45
1074: SmartRead Resume.....	45
1075: Set Tag Buffer Filter.....	46
1076: Query SmartRead Configuration.....	46
1077: Switch SmartRead Configuration.....	46
1078: Get FIFO Buffer Total Size.....	47
1079: Get FIFO Buffer Used Bytes.....	47
107A: Read from FIFO Buffer.....	48
107B: Write to FIFO Buffer.....	48
107C: Peek into FIFO Buffer.....	49
107D: Discard FIFO Buffer.....	49
F003: Switch off Device.....	49
2.2 2xxx: Periphery Control.....	50
2000: Determine Pushbutton Status.....	50
2003: LED Control.....	50
2004: Read Digital Input Ports.....	51
2005: Write Digital Output Ports.....	51
2008: Digital Output Control.....	51
2009: Relay Control.....	52
200A: Read Realtime Clock.....	52
200B: Write Realtime Clock.....	53
200C: Get antenna MUX status.....	53
200D: Set antenna MUX.....	54
200E: Read Temperature Sensor.....	55
200F: Activate Buzzer.....	55
2010: Set LCD Text.....	56
2020: Request Port Sequence.....	56
2021: Edit Port Sequence.....	57
2022: Get Port Sequencer Step.....	58
2023: Set Port Sequencer Step.....	58
2024: Iterate Port Sequencer.....	58
2030: Start GSM Subsystem.....	59

2031: Shutdown GSM Subsystem.....	59
2032: Submit AT Command.....	60
2034: Read SMS Buffer.....	60
2035: Add Message To SMS Buffer.....	60
2036: Get Sender Number From SMS Buffer.....	61
2037: Initialize SMS Buffer.....	61
203A: Receive Messages To SMS Buffer.....	62
203B: Transmit Messages From SMS Buffer.....	62
203C: Delete All Messages From SIM.....	62
2040: Get GSM/GPS Version.....	63
2048: Get GPS Fix.....	63
2.3 3xxx: Transponder Recognition.....	64
3000: General Transponder Recognition.....	64
2.4 4xxx: Transponder Read Functions.....	64
4000: Read Transponder (All Types).....	64
4F00: Read Barcode.....	65
2.5 5xxx: Transponder Write Functions.....	65
5000: Write to Transponder (All Types).....	65
2.6 6xxx: Transponder Specific Functions.....	66
6000: Activate Password Mode (General Write/Read Functions).....	66
6001: Deactivate Password Mode (General Write/Read Functions).....	67
6002: Write New Password (General).....	67
6010: Set Password Buffer.....	68
2.7 Exxx: Non-volatile memory commands.....	68
E000: Recover Factory Settings.....	68
E001: Recover User Settings.....	69
E002: Freeze Current Settings.....	69
E008: Recover Tag Buffer.....	70
E009: Freeze Current Tag Buffer.....	70
E080: Read From EEPROM User Space.....	70
E081: Write To EEPROM User Space.....	71
E082: Get EEPROM User Space Size.....	71
E083: Erase EEPROM User Space.....	72
2.8 Fxxx: Service Commands.....	72
F000: Switch on/off Oscillator .....	72
F001: HF Reset.....	73
<b>3 Reader Device Specifics.....</b>	<b>74</b>
3.1 125KHz Systems.....	74
Default Settings after Switching on or Resetting.....	74
3.2 13.56MHz Systems.....	75
General.....	75
System Settings.....	75
Supplementary general command specification.....	78
Get Version.....	78

Flush Buffers.....	78
Common Status Values and Data Flag.....	78
<b>4 Philips PCF7930 Transponder.....</b>	<b>80</b>
4.1 Features.....	80
4.2 Transponder Recognition .....	80
3100: Philips PCF7930 Transponder Recognition.....	80
4.3 Transponder Read Functions.....	80
4100: Read Philips PCF7930 Transponder.....	80
4.4 Transponder Write Functions.....	81
5100: Write to Philips PCF7930 Transponder Block by Block.....	81
5101: Write to Philips PCF7930 Transponder Byte by Byte.....	81
5102: Write to Philips PCF7935 Transponder Block by Block.....	82
5103: Write to Philips PCF7935 Transponder Byte by Byte.....	82
4.5 Transponder-specific Functions.....	83
6100: Set Philips PCF7930 Read Pointer.....	83
6105: Set Philips PCF7935 Read Pointer.....	83
<b>5 Philips Hitag 1 / Hitag S Transponder.....</b>	<b>85</b>
5.1 Features.....	85
5.2 Transponder Recognition .....	85
3200: Philips Hitag 1 /Hitag S Transponder Recognition.....	85
5.3 Transponder Read Functions.....	85
4200: Read Philips Hitag 1 / Hitag S Transponder (Block by Block).....	85
4201: Read Philips Hitag 1 / Hitag S Transponder (Page by Page).....	86
5.4 Transponder Write Functions.....	87
5200: Write to Philips Hitag 1 / Hitag S Transponder Block by Block.....	87
5201: Write to Philips Hitag 1 / Hitag S Transponder Page by Page.....	87
5.5 Transponder-specific Functions.....	88
6200: Philips Hitag 1 / Hitag S ID Request.....	88
6201: Hitag S Set to 4001 Compatible Mode.....	88
6220: Hitag S Anticollision ID Request.....	89
<b>6 EM 4001/2 Sokymat Unique Transponder.....</b>	<b>90</b>
6.1 Features.....	90
6.2 Transponder Recognition.....	90
3300: EM 4001 & 4002 / SID Unique Transponder Recognition.....	90
6.3 Transponder Read Functions.....	90
4300: Read EM 4001 & 4002 or SID Unique Transponder.....	90
4301: Read EM 4001 & 4002 or SID Unique Serial Number.....	91
<b>7 EM V4050 Sokymat Titan Transponder.....</b>	<b>92</b>
7.1 Features.....	92
7.2 Transponder Recognition .....	92
3400: EM 4050 / SID Titan Transponder Recognition.....	92
7.3 Transponder Read Functions.....	92

4400: Read EM 4050 or SID Titan.....	92
<b>7.4 Transponder Write Functions.....</b>	<b>93</b>
5400: Write to EM V4050 / SID Titan Transponder Block by Block.....	93
<b>7.5 Transponder-specific Functions.....</b>	<b>94</b>
6400: Write to EM V4050 / SID Titan Protection Word.....	94
6401: Write to EM V4050 / SID Titan Control Word.....	94
6402: Switch on EM V4050 / SID Password Mode (Specific).....	95
6403: Deactivate EM V4050 / SID Titan Transponder Password Mode (Specific).....	95
6404: Write New Password to EM V4050 / SID Titan Transponder (Specific).....	96
<b>8 EM 4205/4305 Transponder.....</b>	<b>97</b>
8.1 Features.....	97
8.2 Transponder Read Functions.....	97
4410: Read EM 4205/4305 Transponder.....	97
8.3 Transponder Write Functions.....	97
4410: Write EM 4205/4305 Transponder.....	97
8.4 Transponder-specific Functions.....	98
6412: EM 4205/4305 Login.....	98
6413: EM 4205/4305 Protect Command.....	98
6416: Configure Reader for EM 4205/4305.....	99
6417: Get Reader Configuration for EM 4205/4305.....	99
6418: Configure Reader and/or Tag for EM 4205/4305.....	100
<b>9 Temic e5530 Transponder.....</b>	<b>101</b>
9.1 Features.....	101
9.2 Transponder Recognition.....	101
3500: Temic e5530 Transponder Recognition.....	101
9.3 Transponder Read Functions.....	101
4500: Read Temic e5530 Transponder.....	102
9.4 Transponder-specific Functions.....	102
6500: Temic e5530 Reader Configuration.....	102
6501: Temic e5530 Header.....	103
<b>10 Temic e555x / Q5 Transponder.....</b>	<b>104</b>
10.1 Features.....	104
10.2 Transponder Recognition .....	104
3600: Temic e5550 Transponder Recognition.....	104
10.3 Transponder Read Functions.....	105
4600: Read Temic e5550 Transponder.....	105
4601: Read Temic Q5 Transponder via Direct Access.....	105
10.4 Transponder Write Functions.....	106
5600: Write to Temic e5550 Transponder without Lock Bit.....	106
5601: Write to Temic e5550 Transponder with Lock Bit.....	107
5602: Write to Temic Q5 Transponder without Lock Bit.....	107
5603: Write to Temic Q5 Transponder with Lock Bit.....	107

10.5 Transponder-specific Functions.....	108
6600: Temic e5550 Reader Configuration.....	108
6601: Configure Temic e5550 Transponder.....	109
6602: Temic Q5 Reader Configuration.....	111
6603: Configure Temic Q5 Transponder.....	111
6604: Q5 Transponder Set to 4001 Compatible Mode.....	112
<b>11 Sokymat NOVA Transponder.....</b>	<b>113</b>
11.1 Features.....	113
11.2 Transponder Recognition.....	113
3000: General Transponder Recognition.....	113
3700: NOVA Transponder Recognition.....	113
11.3 Transponder Read Functions.....	113
4000: Read Transponder (All Types).....	113
4700: Read NOVA Transponder.....	114
4701: Read NOVA Transponder in EM4001 Mode.....	114
11.4 Transponder Write Functions.....	115
5000: Write to Transponder (All Types).....	115
5700: Write to NOVA Transponder Block by Block.....	115
5701: Write to NOVA Transponder Block by Block in EM4001 Mode.....	115
11.5 Transponder-specific Functions.....	116
6700: Configure NOVA Transponder.....	116
6702: Configure Reader for NOVA Transponder.....	117
6703: Request Reader Configuration for NOVA.....	118
<b>12 Philips PCF-7936 / Hitag 2 Transponder.....</b>	<b>119</b>
12.1 Features.....	119
12.2 Transponder Recognition.....	119
3000: General Transponder Recognition.....	119
3800: Philips PCF-7936/Hitag2 - Transponder Recognition.....	119
12.3 Transponder Read Functions.....	120
4000: Read Transponder (All Types).....	120
4800: Read Philips PCF7936/Hitag2 Transponder.....	120
12.4 Transponder Write Functions.....	120
5000: Write to Transponder (All Types).....	121
5800: Write Philips PCF7936/Hitag2 Transponder.....	121
12.5 Transponder specific Functions.....	121
6800: Philips PCF7936/Hitag2 ID (serial number) Identification.....	121
6801: Philips PCF7936/Hitag2 - Password Transmission.....	122
6802: Set Mode of Philips PCF7936/Hitag2.....	122
6803: Set Crypto-Keys and Password-TAG.....	123
6804: Crypto Processor Version Request.....	123
68FF: Hitag Family Generic Custom Command.....	124
<b>13 Philips PCF-7937 / Hitag 2 extended Transponder.....</b>	<b>125</b>
13.1 Features.....	125

13.2 Transponder Read Functions.....	125
4830: Read.....	125
13.3 Transponder Write Functions.....	126
5830: Write.....	126
13.4 Transponder specific Functions.....	126
6830: Read XMA-configuration.....	126
6831: Write XMA-configuration.....	127
6832: Read multiple XMA-configuration.....	127
6833: Write multiple XMA-configuration.....	127
<b>14 ICode 1.....</b>	<b>128</b>
14.1 Features.....	128
14.2 Low Level Functions.....	128
6A10: Anticollision / Select.....	128
4A10: Selected Read.....	128
4A14: Unselected Read.....	129
5A10: Write.....	129
6A18: Halt.....	130
6A1A: Reset QUIET Bit.....	131
6A1C: EAS.....	131
14.3 High Level Functions.....	131
6A20: Create Inventory.....	132
6A21: Get Inventory.....	132
6A22: Get ID Range from Inventory.....	133
6A23: Realtime Inventory.....	133
6A24: Create/Get Inventory.....	134
6A26: Select Individual.....	134
4A28: Looped Read.....	135
5A24: Direct Write.....	135
6A28: Direct Halt.....	136
6A29: EAS Alarm.....	136
6A2A: Request Write Protect State.....	137
6A2B: Set Write Protect.....	137
6A2C: Change EAS Bit.....	137
6A2D: Set QUIET Bit.....	138
14.4 Settings Functions.....	138
1A30: Request Protocol Specific Setting.....	138
1A31: Edit Protocol Specific Setting.....	139
<b>15 Tag-it (non-ISO).....</b>	<b>140</b>
15.1 Features.....	140
15.2 Low Level Functions.....	140
6B10: Single Anticollision Round.....	140
4B10: Get Block.....	141
4B16: Get Version.....	141
5B10: Put Block.....	142



5B14: Put Block Lock.....	142
6B14: Lock Block.....	142
6B18: Quiet.....	143
15.3 High Level Functions.....	143
6B20: Create Inventory.....	143
6B21: Get Inventory.....	144
6B22: Get ID Range from Inventory.....	144
6B23: Realtime Inventory.....	145
6B24: Create/Get Inventory.....	145
4B24: Get Multiple Blocks.....	146
5B24: Put Multiple Blocks.....	147
5B28: Put And Lock Multiple Blocks.....	147
6B28: Lock Multiple Blocks.....	148
4B2A: Looped Address Scan.....	148
15.4 Settings Functions.....	149
1B30: Request Protocol Specific Setting.....	149
1B31: Edit Protocol Specific Setting.....	149
<b>16 ISO 15693.....</b>	<b>151</b>
16.1 Features.....	151
16.2 Low Level Functions.....	151
6C10: Single Anticollision Round.....	151
6C12: Select.....	152
4C10: Read Single Block.....	152
4C12: Read Multiple Blocks.....	152
4C16: Get System Information.....	153
4C18: Get Security Status.....	154
5C10: Write Single Block.....	154
5C12: Write Multiple Blocks.....	154
5C16: Write AFI.....	155
5C17: Write DSFID.....	155
6C14: Lock Block.....	156
6C16: Lock AFI.....	156
6C17: Lock DSFID.....	157
6C18: Stay Quiet.....	157
6C1A: Reset To Ready.....	157
6C1C: Low Level Custom Command.....	158
6C1E: Custom Read Command.....	159
6C1F: Custom Write Command.....	159
1C1E: Set Custom Response Delay.....	161
16.3 High Level Functions.....	161
6C20: Create Inventory.....	161
6C21: Get Inventory.....	162
6C22: Get ID Range from Inventory.....	162
6C23: Realtime Inventory.....	163
6C24: Create/Get Inventory.....	164

4C20: Advanced Read Single Block.....	164
4C24: Advanced Read Multiple Blocks.....	165
5C20: Advanced Write Single Block.....	166
5C24: Advanced Write Multiple Blocks.....	166
6C26: Advanced Lock Single Block.....	167
6C28: Advanced Lock Multiple Blocks.....	167
5C26: Advanced Write AFI.....	168
5C27: Advanced Write DSFID.....	168
6C2A: Advanced Lock AFI.....	169
6C2B: Advanced Lock DSFID.....	169
4C2A: Looped Address Scan.....	170
16.4 Settings Functions.....	170
1C30: Request Protocol Specific Setting.....	170
1C31: Edit Protocol Specific Setting.....	172
1C34: Get ISO Tag Descriptor.....	172
1C35: Set ISO Tag Descriptor.....	173
1C36: Get ISO Tag Descriptor from ROM.....	174
1C38: Set Temporary Response Delay.....	175
<b>17 ISO 15693 Custom.....</b>	<b>176</b>
17.1 TI Tag-it ISO Custom Functions.....	176
5C82: Write Two Blocks.....	176
17.2 Infineon My-D Custom Functions.....	176
4C90: Read Block.....	176
5C90: Write Block.....	177
6C94: Lock Block.....	177
17.3 Philips ICode SLI Custom Functions.....	178
6CA0: Inventory Read.....	178
6CA4: Create IR Inventory.....	179
6CA5: Get IR Inventory.....	179
6CA6: Get ID Range from IR Inventory.....	180
6CA7: Create/Get IR Inventory.....	180
6CA8: Change EAS Flag.....	181
6CA9: Lock EAS Flag.....	181
6CAA: EAS.....	182
6CAB: EAS Alarm.....	182
17.4 KSW TempSens Custom Functions.....	183
6CB0: Start Log.....	183
6CB2: Stop Log.....	183
6CB4: Get Log Entry.....	184
6CB6: Get Status.....	184
6CB8: Lock.....	185
6CB9: Unlock.....	185
17.5 EM 4x3x Custom Functions.....	186
4CC8: Fast Read Multiple Blocks.....	186
4CCA: Read Signature.....	186

5CC8: Change Key.....	187
6CC0: Login.....	187
6CC1: Login using Key/Password Buffer.....	188
6CC2: Inventory Block Read.....	188
6CC3: Destroy.....	189
6CC4: Get Multiple Block Protection Status.....	189
6CC5: Protect Memory Page.....	190
6CC6: Set Privacy Mode.....	190
6CC7: Set Random ID Mode.....	191
6CC8: Change Authentication Level.....	191
6CC9: Enhanced Authentication.....	192
6CCA: Authentication Sequence.....	192
6CCB: Authentication Sequence using Key/Password Buffer.....	193
6CCC: Create IBR Inventory.....	194
6CCD: Get IBR Inventory.....	194
6CCE: Get ID Range from IBR Inventory.....	195
6CCF: Create/Get IBR Inventory.....	195
17.6 KSW VarioSens Custom Functions.....	196
6CD0: Init.....	196
6CD1: Set Log Mode.....	197
6CD2: Set Log Timer.....	197
6CD3: Set Custom.....	197
6CD4: Start Log.....	198
6CD5: Get Log State.....	198
6CD6: Set Passive.....	199
6CD7: Get Timer State.....	199
6CD8: Get State.....	200
6CD9: Get One Block.....	200
6CDA: Timer Sync.....	201
6CDB: Get Voltage.....	201
6CDC: Set Calibration.....	202
6CDD: Verify Password.....	202
6CDE: Set Password.....	203
6CDF: Verify Buffered Password (Schreiner e-Temp).....	203
17.7 VarioSens High Level Custom Functions.....	203
6CBA: Easy Init.....	204
6CBB: Get Config.....	204
6CBC: Get Progress.....	205
6CBE: Read Single Measurement.....	206
6CBF: Get Battery Voltage.....	207
17.8 ICode SLI-S Custom Functions.....	208
6CAC: Password Protect EAS.....	208
6CAD: Write EAS ID.....	208
6CAE: Read EPC.....	209
6CE0: Inventory Page Read.....	209
6CE2: Get Random Number.....	210

6CE3: Set Password.....	210
6CE4: Write Password.....	211
6CE5: Lock Password.....	211
6CE6: Protect Page.....	212
6CE7: Lock Page Protection.....	213
6CE8: Get Protection Status.....	213
6CE9: Destroy.....	214
6CEA: Enable Privacy.....	214
6CEB: 64 Bit Password Protection.....	215
17.9 ICode SLI-S High Level Custom Functions.....	215
6CAF: Login.....	215
6CEC: Create IPR Inventory.....	216
6CED: Get IPR Inventory.....	217
6CEE: Get ID Range from IPR Inventory.....	217
6CEF: Create/Get IPR Inventory.....	217
17.10 ScemTag Sensor Transponder Custom Functions.....	218
6CF0: Single Measurement.....	218
<b>18 ICode EPC/UID.....</b>	<b>220</b>
18.1 Features.....	220
18.2 EPC Low Level Functions.....	220
6A80: Single Round.....	220
5A80: Write Block.....	221
6A84: Destroy.....	221
18.3 UID Low Level Functions.....	222
6A90 Single Round.....	222
5A90: Write Block.....	223
6A94: Destroy.....	223
18.4 EPC High Level Functions.....	223
6AA0: Create Inventory.....	223
6AA1: Get Inventory.....	224
6AA2: Get ID Range from Inventory.....	225
6AA3: Realtime Inventory.....	225
6AA4: Create/Get Inventory.....	226
5AA0: Write User Data.....	226
5AA1: Write Destroy Code.....	227
4AA8: Looped Read.....	227
18.5 UID High Level Functions.....	228
6AB0: Create Inventory.....	228
6AB1: Get Inventory.....	228
6AB2: Get ID Range from Inventory.....	229
6AB3: Realtime Inventory.....	229
6AB4: Create/Get Inventory.....	230
5AB0: Write User Data.....	231
5AB1: Write Destroy Code.....	231
4AB8: Looped Read.....	232

18.6 Settings Functions.....	232
1A32: Request Protocol Specific Setting.....	232
1A33: Edit Protocol Specific Setting.....	233
1A34: Get ID Mask.....	234
1A35: Set ID Mask.....	234
<b>19 SC-433.....</b>	<b>235</b>
19.1 Features.....	235
19.2 Low Level Functions.....	235
6F10: Single Anticollision Round.....	235
6F12: Select.....	235
4F10: Read Single Block.....	236
4F12: Read Multiple Blocks.....	236
4F16: Get Version.....	237
4F17: Get Size.....	237
4F1A: Get Battery Status.....	237
5F10: Write Single Block.....	238
5F18: Set ID.....	238
5F19: Set Password.....	239
5F1A: Set Defaults.....	239
6F18: Quiet.....	239
6F19: Sleep.....	240
6F1A: Reset.....	240
6F1B: Erase Tag Data.....	241
19.3 High Level Functions.....	241
6F1C: Create Inventory.....	241
6F1D: Get Inventory.....	242
6F1E: Get ID Range from Inventory.....	242
6F1F: Create/Get Inventory.....	242
6F14: Create Inventory (Compatibility Mode).....	243
6F15: Get Inventory (Compatibility Mode).....	244
6F16: Get ID Range from Inventory (Compatibility Mode).....	244
6F17: Create/Get Inventory (Compatibility Mode).....	245
19.4 Settings Functions.....	245
1F10: Request Protocol Specific Setting.....	245
1F11: Edit Protocol Specific Setting.....	246
<b>20 ISO 11784 / ISO 11785 (FDX-B) Transponder.....</b>	<b>247</b>
20.1 Features.....	247
20.2 Transponder Recognition .....	247
3D00: ISO 11784 / 11785 (FDX-B) transponder Recognition.....	247
20.3 Transponder Read Functions.....	247
4D00: Read ISO 11784 / 11785 (FDX-B) transponder.....	247
<b>21 EPC Class 1 Gen.2.....</b>	<b>249</b>
21.1 Features.....	249

21.2 Low Level Functions.....	249
21.3 High Level Functions.....	249
4F38: Read Multiple Blocks.....	249
5F38: Write Single Block.....	249
6F3C: Create Inventory.....	250
6F3D: Get Inventory.....	251
6F3E: Get Single ID from Inventory.....	251
6F3F: Create/Get Inventory.....	251
21.4 Settings Functions.....	252
1F30: Request Protocol Specific Setting.....	252
1F31: Edit Protocol Specific Setting.....	253
<b>22 EPC Class 1 Gen.2 Custom.....</b>	<b>254</b>
22.1 ScemTag Sensor Transponder Custom Functions.....	254
6F60: Single Measurement.....	254
22.2 IDS SL900A Based Sensortransponder Custom Functions.....	254
6F70: Single Measurement.....	254
6F73: Configure Log.....	255
6F74: Initialize Log.....	255
6F75: Start Log.....	256
6F76: Stop Log.....	256
6F77: Read Log Data.....	257
6F7F: CoolLog Custom Command.....	257
<b>23 ISO 18000-3 Mode 3.....</b>	<b>259</b>
23.1 Features.....	259
23.2 Low Level Functions.....	259
6F80: Low Level Select.....	259
6F81: Single Round.....	260
4F80: Low Level Read.....	261
5F80: Low Level Write.....	262
6F82: Low Level Custom Command.....	262
23.3 High Level Functions.....	263
6F8C: Create Inventory.....	263
6F8D: Get Inventory.....	264
6F8E: Get ID Range from Inventory.....	264
6F8F: Create/Get Inventory.....	265
8F8C: Realtime Inventory.....	265
8F80: Select.....	266
8F81: Iterate Inventory.....	267
4F88: Read.....	268
5F88: Write.....	269
8F82: Kill.....	269
8F83: Lock.....	270
8F84: Access.....	271
8F85: Erase.....	271

8F86: Permalock.....	272
8F88: Custom Command.....	273
23.4 Settings Functions.....	273
1F80: Request Setting.....	273
1F81: Edit Setting.....	274
<b>24 ISO 18000-3 Mode 3 Custom.....</b>	<b>276</b>
24.1 NXP ICode ILT Custom Functions.....	276
6F84: ICode ILT Config.....	276
6F85: ICode ILT EAS.....	276
6F86: ICode ILT EAS Alarm.....	277
<b>25 Hitag AES.....</b>	<b>278</b>
25.1 Features.....	278
25.2 Transponder Read Functions.....	278
4840: Read Hitag AES Transponder.....	278
25.3 Transponder Write Functions.....	278
5840: Write Hitag AES Transponder.....	278
25.4 Transponder Specific Functions.....	279
6840: Get Hitag AES ID.....	279
6841: Set Hitag AES Key.....	279
<b>26 Hitag Pro.....</b>	<b>280</b>
26.1 Features.....	280
26.2 Transponder Read Functions.....	280
4880: Read Page.....	280
4881: Read Byte.....	280
4882: Read Multiple Pages.....	281
26.3 Transponder Write Functions.....	281
5880: Write Page.....	281
5881: Write Byte.....	282
5882: Write Multiple Pages.....	282
26.4 Transponder Specific Functions.....	283
6880: Get Hitag Pro ID.....	283
6881: Set Immobilizer Secret Key.....	283
<b>27 ISO 14443-A.....</b>	<b>284</b>
27.1 Low Level Functions.....	284
6F42: Select.....	284
4F40: Read Page.....	284
5F40: Write Block.....	284
5F41: Write4 Block.....	285
27.2 High Level Functions.....	285
6F4C: Create Inventory.....	285
6F4D: Get Inventory.....	286
6F4E: Get ID Range from Inventory.....	286

---

6F4F: Create/Get Inventory.....	287
<b>28 ISO 14443-A Custom.....</b>	<b>288</b>
28.1 NXP NTAG Custom Functions.....	288
8F40: Get Version.....	288
8F41: Fast Read.....	288
8F42: Read Counter.....	289
8F43: Password Authentication.....	289
8F44: Read Signature.....	289
<b>29 Document Version History.....</b>	<b>291</b>



## 1 Basic Protocol Format

### 1.1 Notational Remarks

Within the request/response syntax description all control characters are printed in **bold** letters, i. e. for an **ACK** the value 06h (6 Hex) is transmitted on the serial interface. Constant character strings are indicated by quotes (e. g. "1000"). Variable parameters are enclosed within pointy brackets ("<" and ">") with each character corresponding to one character to be actually transmitted. The meaning of the characters is explained in a table following the syntax description. Values which are optional and not necessarily present (e. g. response data which depends on request parameters) are enclosed in square brackets ("[" and "]"). Braces ("{" and "}") are used for special values like checksums.

### 1.2 General Transmission Structure

#### Data Framing

Transfer of request and response is generally embedded within the control characters **STX** and **ETX**. Between these control characters only ASCII characters are permitted. Numerical values (parameters) are transferred in an ASCII character string. For e. g. the numerical value 123 (= 7Bh), the ASCII character string "7B" is transmitted. For example, to transfer the parameter value 100, proceed as follows:

<i>Decimal parameter value</i>	<i>Hex value</i>	<i>ASCII character string to be transmitted</i>
100	64h	"64"

Thus, for each byte to be transferred two characters are used.

In some rare cases, parameters are expected as decimal character strings. If e. g. the value 123 (decimal) is to be transmitted, proceed as follows:

<i>Decimal parameter value</i>	<i>ASCII character string to be transmitted</i>
123	"123"

As you can see, in this case 3 characters are used for each byte to be transmitted.

The data transfer ends with a checksum behind the **ETX** control character. The checksum is calculated using a XOR operation from the first transferred control character up to (and including) the **ETX** control character. The start value for the XOR operation is 0. In case of a response from the RF system, the **STX** is preceded by another control character for acknowledgement (**ACK** for positive acknowledgement and **SYN** for an error message).

Alternatively to framed requests or responses, specific single-byte control characters are permitted

without any parameters (the control characters **ESC** and **NAK**).

### Control Character Definition

The control characters are defined with the following values:

<i>Acronym</i>	<i>Value</i>	<i>Description</i>
<b>STX</b>	02h	Start Of Transmission
<b>ETX</b>	03h	End Of Transmission
<b>ACK</b>	06h	Acknowledge
<b>NAK</b>	15h	No Acknowledge
<b>SYN</b>	16h	Error
<b>ESC</b>	1Bh	Interrupt

## 1.3 Host Request to RF System

### Function Calls

A request consists of the **STX** control character, a 4 hex character function number, the parameters and the **ETX** control character, followed by the checksum:

**STX** <Function number> [Parameters] **ETX** <Checksum>

In the following example the oscillator is being switched on for maintenance.

**STX** "F000" <vv> **ETX** {c}

In this case, "vv" is a parameter which can be "00" or "01" and describes the oscillator state to be switched to. The {c} at the end stands for the XOR checksum. Throughout this document, the following notation will be used for request format descriptions:

<i>Host to RF system</i>	
<b>STX</b>	"F000" <vv>
<b>ETX</b>	{c}
v	Oscillator state to switch to ("00"=off; "01"=on).

The checksum is calculated as follows:

<i>Character</i>	<i>Value</i>	<i>Operation</i>	<i>Checksum</i>		<i>Result</i>
			00h		
<b>STX</b>	02h	XOR	00h	=	02h
'F'	46h	XOR	02h	=	44h
'0'	30h	XOR	44h	=	74h
'0'	30h	XOR	74h	=	44h
'0'	30h	XOR	44h	=	74h

'0'	30h	XOR	74h	=	44h
'1'	31h	XOR	44h	=	75h
<b>ETX</b>	03h	XOR	75h	=	76h
			76h		

The transmission will be:

**STX** "F000" "01" **ETX** 76h

In the programming languages C/C++ and Java, the character sequence

0x02 "F00001" 0x03 0x76

would be transmitted, in Pascal/Delphi it would be the character string

#\$02 + 'F00001' + #\$03 + #\$76.

## Interruption

Certain states of the RF device can be interrupted by transmitting a single **ESC** character. These states can be, depending on the type of device connected, the following:

- A complex function (e. g. "Create Inventory") is running: interruption forces an immediate response which will probably contain an incomplete result.
- Automatic Execution (using the Command Buffer) is running: interruption disables Automatic Execution until next reset.
- An interface board with standalone capabilities (e. g. the SEC-1500 "Combi Board") is connected and active: interruption disables standalone capabilities until next reset.

## 1.4 RF System Response to Host

### Positive Acknowledgement

The response of the RF system is structured like the request; the only difference is that the response block is preceded by an acknowledge control character:

**ACK STX** <Function number> [Data] **ETX** <Checksum>

In case of a positive response, the **ACK** will be sent immediately, whereas processing the command may take some time until the rest of the response is transmitted.

Regarding our above example, the general (positive) response format will look like the following. Please note that some line breaks have been inserted for better readability; these are not actually part of the message!

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"F000"
<b>ETX</b>	{c}
-	No return data.

### Negative Acknowledgement

A specified error message (negative acknowledge) looks like this:

**SYN STX** <Function number> <Error number> **ETX** <Checksum>

The error number will be returned as a parameter. The error response is only returned if the request has been made according to the request syntax, but includes e. g. a parameter providing an incorrect value, or if the function is simply not supported by the RF system.

A negative acknowledgement has the following format:

<i>RF system to Host (when an error occurred)</i>	
<b>SYN</b>	
<b>STX</b>	<ffff> <ee>
<b>ETX</b>	{c}
f	Function number of the request which has caused the error message.
e	Error code; 2 characters are transmitted, making up a 1-byte error number.

The following table shows all theoretically possible error codes including a short description:

Error	Description
"00"	Unspecified error.
"01"	Common system error.
"02"	Invalid function number.
"03"	Function not supported.
"04"	Syntax error in parameter.
"05"	Invalid parameter value.
"06"	No log data available.
"08"	Device not ready.
"09"	Buffer overflow.
"10"	Tag write/read error.
"11"	Password error.
"12"	Error executing function.
"13"	Antenna error (bad SWR) – tag functions temporarily unavailable.
"14"	System overheat – tag functions temporarily unavailable.

### No Acknowledge

An unspecified error message (no acknowledge) consists of a **NAK** control character. It is

transmitted to the host when the request structure was invalid.

## Multiple Responses

Some commands (namely the “Realtime Inventory” commands) can by definition send multiple responses to one command. This may be desirable to quickly obtain interim results before the command has actually finished execution.

Example:

```
ACK STX <Function number> [ Partial Data] ETX <Checksum>
ACK STX <Function number> [ Partial Data] ETX <Checksum>
ACK STX <Function number> [ Partial Data] ETX <Checksum>
ACK STX <Function number> ETX <Checksum>
```

The last response, containing no data, always signals the host that no more responses are to be expected. This restores synchronization between host and device.

There are several settings that affect multiple responses:

- *STX/ETX Handshaking*: When enabled, each additional response must be triggered by sending a single **ACK** character to make sure the host is not flooded by more data than it can handle. When disabled, each additional response is transmitted as soon as it is available.
- *Overscan*: Depending on the device, Overscan (i. e. automatically adding additional scan loops to inventory commands) can be used as an optimization of responsiveness (often named \*\_RETRIES in the tag specific settings).
- *Per-cycle repeat filter*: Depending on the device, the per-cycle repeat filter, which is enabled by default to make sure no tag is repeated within one cycle, may be disabled to increase data redundancy.

## 2 Command Description

### 2.1 1xxx: System Control

#### 1000: Reset Request

The Reset request returns the RF system to its start-up condition by carrying out a restart. Depending on the system, it may take some time until acknowledgement. During restart, the serial interface is reset to its default values, the Background functions are deactivated, and the displays and outputs, if existing (system dependent), are reset to their default values (see “Default Settings”).

<i>Host to RF System Request</i>	
<b>STX</b>	"1000"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF system to Host</i>
See command “1001”, replace 1001 with 1000.

#### 1001: Request Version Number

This function is used to determine the (system) designations of the connected hardware as well as the index versions of hard- and software (see “Reset Request”).

A new extended format has been introduced, which, for backward compatibility, starts with the old format, using “X” as the main index. The version string may be up to 40 characters in total.

<i>Host to RF System Request</i>	
<b>STX</b>	"1001"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF system to Host (legacy format)</i>	
<b>ACK</b>	
<b>STX</b>	"1001" <sss> <i> "/" <hh> <vv>
<b>ETX</b>	{c}
s	System designation, e. g. „SIH“ or „STK“ or „DEM“, etc.
i	System main index (1 or 4 digit), e. g. „1“ or „2“ or “1000” or “2100”
h	Hardware version number, e. g. „01“ for hardware index 1.
v	Software version number, e. g. „10“ for software index 10.

<i>RF System to Host Response</i>
-----------------------------------

<b>ACK</b>	
<b>STX</b> "1001" <sss> "X/" <vv> <ww> "/" <m...> ["/Build:" <bb..bb>] [ "/" <s...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	System designation, e. g. „STK“ or „SHL“, etc.
v	Version number, major part.
w	Version number, minor part.
m	Complete module name.
b	Build Version and/or Date, e.g. "/Build: Rev:15 Date: 2007/10/10 14:13:22"
s	Software features.

## 1002: Interface Test

The interface test is only necessary in the case of a baud rate change (see “Change Baud Rate”).

<i>Host to RF System Request</i>	
<b>STX</b> "1002"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1002"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

## 1003: Change Baud Rate

With this function you can set the transmission baud rate of the RF system. The supported baud rates depend on the system used and can be found in the system documentation (technical data); invalid baud rate settings generate an error message. After successful transmission of a new baud rate, an interface test (command “1002”, see description above) must be carried out within 10 seconds on the basis of the new baud rate; otherwise, the system is restarted (Reset). Within this time (until the successful interface test) all other functions or interface transmissions also cause a restart of the RF system. The "1003" command changes the baud rate only temporarily; when powering off the system, it is lost. To change the baud rate permanently, you'll prefer the commands "1011"/"E002" or "1011", depending on which your system supports.

<i>Host to RF System Request</i>	
<b>STX</b> "1003" <bbbb>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	New baud rate, given in hex digits:

04B0: 1200 Baud.
12C0: 4800 Baud.
2580: 9600 Baud (factory default).
4B00: 19200 Baud.
7080: 28800 Baud.
9600: 38400 Baud.
E100: 57600 Baud.
FFFF: 115200 Baud.
FFFE: 230400 Baud.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1003"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

#### 1004: Set Byte Timeout

This function is used to determine the timeout interval of the RF system during receipt, i. e. the maximum time between transmission of two data bytes during a request, in order to be able to work with slower host systems, too, or to manually enter data. The maximum time that can be set is 10 seconds, whereas the default timeout setting is 500 ms. Please note that for devices which support the 1010/1011 (Global Settings) commands the use of the 1004 command is discouraged.

<i>Host to RF System Request</i>	
<b>STX</b> "1004" <tttt>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	Time value, will be multiplied by 10ms.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1004"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

#### 1006: Request Log Data

This function returns an 8 byte random number to be used for transmitting a password in encoded form to the RF system. The password encoding algorithm for the different transponder types can be found in separate data sheets.

<i>Host to RF system</i>	
<b>STX</b> "1006"	



<b>ETX</b>	{c}
-	No request parameters.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"1006" <zzzzzzzzzzzzzzzzzz>
<b>ETX</b>	{c}
z	ASCII character string of an 8 byte long random number.

### 1008: Switch RF System to Low-current Mode

This function puts the RF system into power saving mode; the system remains in this low-current mode until any character is sent via interface. Then the RF system displays a start reset message and returns to its default settings (see "Default Settings").

<i>Host to RF system</i>	
<b>STX</b>	"1008"
<b>ETX</b>	{c}
-	No request parameters.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"1008"
<b>ETX</b>	{c}
-	No return data.

### 100A: Request Supported Transponder Types

With this function you can query which transponder types are supported by the RF system used.

<i>Host to RF System Request</i>	
<b>STX</b>	"100A"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"100A" <ssss> [ <eeee> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Field of 16 bits.
e	Extended bit field, only present if bit #15 in s is set.

The supported transponder types are returned in binary code.

Bit No.	Transponder Type	Code Number
0	1 = Philips PCF7930	1
1	1 = Mikron Hitag HT1-DCx	2
2	1 = EM V4002 / SID Unique	3
3	1 = EM V4050 / SID Titan	4
4	1 = Temic e5530	5
5	1 = Temic e5550	6
6	1 = SID Nova	7
7	1 = Philips PCF7936 / Hitag2	8
8	1 = Temic e5560/ e5561	9
9	1 = Philips ICode	A
10	1 = TI Tag-it	B
11	1 = ISO 15693	C
12	1 = ISO 11784/11785	D0
13	<i>not used; reserved for future transponder types</i>	n/a
14	1 = Barcode / Philips SECT Special functions	F
15	1 = Use extended bit field	n/a

The extended bit field is encoded as follows.

Bit No.	Transponder Type	Code Number
0	1 = SC-433	F1
1	1 = Philips UCode EPC	F2
2	1 = EPC Gen. 2	F3
3	1 = Lyngsoe PosTag	F1
4	1 = ISO 14443 Type A	F4
5	1 = ISO 14443 Type B	F5
6	1 = ICode EPC (often only ICode=09 is used)	A
7	1 = ICode UID (often only ICode=09 is used)	A
8	1 = EM 4x05	4
9	1 = Hitag AES	84
10	1 = ISO 18000-3 Mode 3	F8
11	1 = Hitag Pro	88
12	1 = Megacoder	D4
13	1 = Hitag3	8C
14-15	<i>not used; reserved for future transponder types</i>	n/a

### 100B: Select Transponder Types for General Functions

For reasons of speed, this function can be used to release the transponder types that are used for the general transponder functions. In the case of, e. g., transponder Search function “3000”, all released transponder types must be searched for one after the other. This does take up a certain time for each type. If search has to be carried out for all transponders that are supported by the system, this may take too long for some applications. Since, on the other hand, most applications only make use of some of the available transponder types, this function can be used to speed up the Search, Write, and Read functions. Only transponder types that are supported by the system can be released

(request via command "100A", if necessary). After program start-up, all supported transponder types are released.

<i>Host to RF system</i>	
<b>STX</b>	"100B" <ssss>
<b>ETX</b>	{c}
s	ASCII character string of a word = value of 16 bits' size.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"100B"
<b>ETX</b>	{c}
-	No return data.

The selectable transponder types are transmitted in binary code as follows:

<i>Flag</i>	<i>Transponder Type</i>
Bit 00	1= Philips PCF7930
Bit 01	1= Mikron Hitag HT1-DCx
Bit 02	1= EM V4002 / SID Unique
Bit 03	1= EM V4050 / SID Titan
Bit 04	1= Temic e5530
Bit 05	1= Temic e5550
Bit 06..15	<i>not used; reserved for future transponder types</i>

### 100E: Request Supported Function Numbers

Returns a list of STX/ETX functions supported by the device.

<i>Host to RF System Request</i>	
<b>STX</b>	"100E"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"100E" [ <ffff> [ <ffff> [...] ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
f	Function number (4 digits hex).

### 1010: Request System Setting

Using a key/value combination, this function allows access to several system settings which are not related to a specific transponder type. Numeric values are unsigned 16 bit hex values; switch values

are "0000" for "false" or "0001" for "true".

Please see chapter "Reader Device Specifics" for key table.

<i>Host to RF System Request</i>	
<b>STX</b> "1010" <kk>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1010" <vvvv>	
<b>ETX</b> {c}	
<i>Response Data</i>	
v	Value.

### 1011: Edit System Setting

This function is used to change system constants. See "request system settings" ("1010") for further information.

<i>Host to RF System Request</i>	
<b>STX</b> "1011" <kk> <vvvv>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key.
v	Value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1011"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 1014: Field Analysis

This function measures the power sent to the antenna, the power reflected by the antenna, and then calculates the standing wave ratio (SWR). Precision is very limited, so take these values as a rough estimation only. On devices with an SWR related LED, this command will also update the LED according to the SWR value. The SWR should be as low as possible; 1000 would be ideal but is only possible in theory. Values below 1500 are good, values above 2000 rather bad and might be improved by moving/rotating the antenna or re-tuning it. Be aware that the antenna may also be detuned by a metallic environment!

This command supports the GLS\_AUTO\_NOREPEAT setting. If activated, repetitions of identical responses are suppressed in auto execution mode (running from command buffer).

<i>Host to RF System Request</i>	
<b>STX</b>	"1014"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1014" <ffff> <rrrr> <ssss>
<b>ETX</b>	{c}
<i>Response Data</i>	
f	Forward power in milliwatts.
r	Reflected power in milliwatts.
s	Standing wave ratio *1000.

### 1015: Tune Antenna

For devices with an Antenna Matching Unit (AMU), this command will perform the auto tuning procedure. The function can use several criteria – if more than one is specified, the ones with the higher bits will be used as a fallback if the lower ones have failed. In case of a complete failure, the previous relays setting will be restored. The current AMU relays configuration can also be read from the system settings (GLS\_ANT\_TUNE) at any time.

<i>Host to RF System Request</i>	
<b>STX</b>	"1015" <cc>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
c	Criteria flags, binary coded: 01: Find minimum of reflected power. 02: Find maximum of default tags (e. g. ISO 15693).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1015" <rr> <ss>
<b>ETX</b>	{c}
<i>Response Data</i>	
r	New AMU relays setting.
s	Status value. 00: Ok. 01: No tags found for specified tag maximum criterium. 02: All relay settings gave an equally good result. 03: More than one local minimum with equal values found. 04: Tuning not performed.

### 1016: Get Reader ID

To get the reader's ID and also its group ID, this function must be used. The reader ID is preset to the device serial number, whereas the group ID is preset to 0000.

<i>Host to RF System Request</i>	
<b>STX</b> "1016"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1016" <gggg> <rrrrrrrr>	
<b>ETX</b> {c}	
<i>Response Data</i>	
g	16 bit Reader group ID.
r	32 bit Reader ID.

### 1017: Set Reader ID

This command is used to set the reader's ID and also its group ID. The group ID is also used by the synchronization feature. To store both IDs permanently, the command "E002" must be used; otherwise they will be lost when the system is powered down. A value of all "F" means to leave either part unchanged.

<i>Host to RF System Request</i>	
<b>STX</b> "1017" <gggg> <rrrrrrrr>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
g	16 bit Reader group ID ("FFFF" to leave old GID).
r	32 bit Reader ID ("FFFFFFFF" to leave old RID).
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1017"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 1018: Flush Buffers

This function can have various effects, depending on the device it is applied to. For example, if the device supports inventory operation (anticollision), the inventory will be cleared. Please see chapter "Reader Device Specifics" for more information.

<i>Host to RF System Request</i>	
----------------------------------	--

<b>STX</b>	"1018"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1018"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 1019: Get Device Serial Number

This command can be used to obtain the reader's serial number.

<i>Host to RF System Request</i>	
<b>STX</b>	"1019"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1019" <ssssssss>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	32 bit serial number.

### 101A: Sync Listen

When you send this command to a sync slave, it will enter the listen mode. In that mode, it will receive commands from the sync master on the RS485 sync cable instead of the standard RS232 line. To take a reader out of listen mode again, simply send an ESC character on RS232. The reader will return into normal RS232 operation mode right away, issuing a notification on RS232.

<i>Host to RF System Request</i>	
<b>STX</b>	"101A"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"101A" <ss>
<b>ETX</b>	{c}

<i>Response Data</i>	
s	Status value: 00: failed to enter listen mode. 01: listen Mode has been entered. 02: listen mode has been terminated ( <b>ESC</b> character received).

### 101B: Ping Sync Slave

This is a rather simple command to test the sync connection from the master to a specific slave, as well as the BUSY connection. An exception is ID "00000000", which addresses all readers.

<i>Host to RF System Request</i>	
<b>STX</b> "101B" <iiiiiii> <mmmmmmmm> <b>ETX</b> {c}	
<i>Request Parameters</i>	
i	The slave's reader ID.
m	The broadcast mask; every bit which is set here will be compared.

<i>RF System to Host Response</i>	
<b>ACK</b> <b>STX</b> "101B" <ss> <b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value: 00: Ok: at least one slave has responded to the request. 01: Error: either the slave is not in listen mode, or there is a problem with the connection.

### 101C: Pingback Sync Slave

Like the "Ping Slave" command, this one also sends a ping request to a specific slave. If this was successful, it will afterwards swap roles temporarily: the master becomes slave and the slave becomes master. Then, the former slave will try a ping on the former master. Afterwards, the role exchange is switched back again. Thus, both directions are tested and it is ensured that the slave reader also works as a master.

<i>Host to RF System Request</i>	
<b>STX</b> "101C" <iiiiiii> <b>ETX</b> {c}	
<i>Request Parameters</i>	
i	The slave's reader ID.

<i>RF System to Host Response</i>	
<b>ACK</b> <b>STX</b> "101C" <ss> <b>ETX</b> {c}	
<i>Response Data</i>	



s	Status value: 00: No errors 01: Ping from master to slave failed 02: BUSY line not free 04: Pingback from slave to master failed 08: Unexpected command received from slave 10: Test aborted 20: Pingback timeout 40: General error 80: Slave not tested
---	---

### 101D: Get Reader ID List

This command scans for any attached slaves and returns a list of reader IDs.

<i>Host to RF System Request</i>	
<b>STX</b>	"101D"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"101D" <cccc> [ <iiiiiiii> [<iiiiiiii> [...] ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
c	Number of reader IDs that follow.
i	Reader ID.

### 101E: Complete Sync Test

This command basically sums up all sync test commands. First, it collects the IDs of all attached (and listening) slaves, and then it does a pingback test on each single slave. It returns a status value for each ID, as well as a cumulated status value for all slaves. If this value is 0, all detected slaves were tested without errors.

<i>Host to RF System Request</i>	
<b>STX</b>	"101E"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"101E" <cccc> <oo> [ <iiiiiiii> <ss> [ <iiiiiiii> <ss> [...] ] ]

<b>ETX</b> {c}	
<i>Response Data</i>	
c	Number of reader IDs that follow.
o	OR'ed cumulation of all status values as a quick overview.
i	Reader ID.
s	Status value (see "Pingback slave" command).

## 1020: Get Command Buffer

This function returns one single line of the current command buffer, which is used for the AutoExec feature of some devices. A command (or command sequence) which has been stored in this buffer can be executed continuously or as a response to a specified trigger event.

The operation mode of the command buffer is controlled via the system settings (GLS\_AUTOEXEC\_MODE), by default it is disabled. Please be aware that changes become active *after* the next reset, i. e. you'll have to store your changes using the E002 function for the changes to take effect at all. This is to make sure that the configuration progress does not become disturbed by automatic responses. For testing, the 1023 command can be used instead of a reset.

The following modes are available:

Mode	Type	Description
0000	Off.	No automatic execution (factory default).
0001	Power On	Command Buffer is triggered immediatley or delayed after power-on. The delay is controlled via GLS_AUTOEXEC_TIME; a delay of 0 means immediate execution.
0002	Timer	Command Buffer is triggered continuously or repeatedly. The time raster is controlled via GLS_AUTOEXEC_TIME; a delay of 0 means continuous execution.
0010	Input Level	Command Buffer is triggered continuously as long as Input #1 is high, but at least for the time specified in GLS_AUTOEXEC_TIME.
0011	Input Level	Command Buffer is triggered continuously as long as Input #2 is high, but at least for the time specified in GLS_AUTOEXEC_TIME.
0018	Input Level	Command Buffer is triggered continuously as long as Input #1 is low, but at least for the time specified in GLS_AUTOEXEC_TIME.
0019	Input Level	Command Buffer is triggered continuously as long as Input #2 is low, but at least for the time specified in GLS_AUTOEXEC_TIME.
0020	Input Edge	Command Buffer is triggered continuously after a positive edge on Input #1, but at least for the time specified in GLS_AUTOEXEC_TIME.
0021	Input Edge	Command Buffer is triggered continuously after a positive edge on Input #2, but at least for the time specified in GLS_AUTOEXEC_TIME.
0028	Input Edge	Command Buffer is triggered continuously after a negative edge on Input #1, but at least for the time specified in GLS_AUTOEXEC_TIME.
0029	Input Edge	Command Buffer is triggered continuously after a negative edge on Input #2, but at least for the time specified in GLS_AUTOEXEC_TIME.

<i>Host to RF System Request</i>	
<b>STX</b> "1020" [ <m> [ <l1> ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode: 's': Single line (default); a line number is not specified. 'm': Multi line; a line number is specified.
l	Line number to be returned; line counting start with 0.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1020" [ <cccc> [ <c> [ <...> ] ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Current command buffer contents (specified line).

### 1021: Set Command Buffer

This function is used to write a new command (or line of a command sequence) to the command buffer, which is used for the AutoExec feature of some devices. The command buffer line count in the system settings (GLS\_CMDBUF\_COUNT) will automatically be updated to match the new sequence length. See command description "1020" for more information.

Command sequences must be set in the appropriate order, starting with line 0. Trying to set a line which has no predecessor (except the first line, of course) leads to an error message. When an existing line is changed, all subsequent lines are deleted. To remove a line, this command can be submitted with no command line attached.

<i>Host to RF System Request</i>	
<b>STX</b> "1021" [ <m> [ <l1> ] ] [ <cccc> [ <c> [ <...> ] ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode: 's': Single line (default); a line number is not specified. 'm': Multi line; a line number is specified.
l	Line number to be set; line counting start with 0.
c	New command (with all required parameters) to store in the specified line of the command buffer.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1021"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 1022: Trigger Command Buffer

The trigger function can be used to trigger the current command buffer, which is used for the AutoExec feature of some devices. This is meant for test purposes only, to see if the command buffer performs as desired.

<i>Host to RF System Request</i>	
<b>STX</b>	"1022"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	<rrrrr...>
<b>ETX</b>	{c}
<i>Response Data</i>	
r	Response of the command currently stored in the command buffer.

### 1023: Enable Command Buffer

Instead of a reset, this command can be used to start the command buffer subsystem. All configured triggers and timers will become active immediately.

<i>Host to RF System Request</i>	
<b>STX</b>	"1023"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1023"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 1026: Get Port Configuration

This command returns the number of inputs, digital (e. g. open collector) outputs, relay outputs and antenna ports. In addition, it reveals all available RF power steps, sorted lowest to highest.

<i>Host to RF System Request</i>	
<b>STX</b>	"1026"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1026" <ii> <oo> <rr> <aa> <pp> [ <ssss> [ <ssss> [ ... ] ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
i	Number of input ports.
o	Number of TTL or open collector output ports.
r	Number of relay output ports.
a	Number of (additional) antenna ports, 0 means only one antenna port.
p	Number of supported power steps.
s	Power step in milliwatts; steps are sorted from lowest to highest.

### 1028: Get Processor Identification String

This function returns the CPU Identification String.

<i>Host to RF System Request</i>	
<b>STX</b> "1028"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1028" <s...>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Processor Identification String.

### 1029: Get Hardware Feature

This function returns information on the hardware.

<i>Host to RF System Request</i>	
<b>STX</b> "1029" <kk>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1029" <vvvv>	
<b>ETX</b> {c}	
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Numeric/Switch, Default, Description):

Key	C	Alias	N/S	Def.	Description
01		HW_FAMILY	n	–	Architecture identifier 0x01xx: TI DSP based 0x02xx: Atmel ARM based
02		HW_FLASH	n	–	Size of Flash memory (kB)
03		HW_RAM	n	–	Size of Ram (kB)

### 102A: Request Interface Board Setting

Using a key/value combination, this function allows access to several system settings for an SEC-1500 or similar interface board. Numeric values are unsigned 16 bit hex values; switch values are "0000" for "false" or "0001" for "true".

<i>Host to RF System Request</i>	
<b>STX</b> "102A" <kk>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "102A" <vvvv>	
<b>ETX</b> {c}	
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Signed/Unsigned, Default, Description):

Key	C	Alias	N/S	Def.	Description
01		GLS_UART0_BAUD	n	2580	UART 0 (host) baud rate.
02		GLS_UART0_BUFFER	n	0400	UART 0 (host) buffer size.
03		GLS_UART0_BLOCK_R	n	001E	UART 0 (host) read blocking time (milliseconds).
04		GLS_UART0_BLOCK_W	n	01F4	UART 0 (host) write blocking time (milliseconds).
05		GLS_UART1_BAUD	n	0000	UART 1 (target) baud rate (or 0 for automatic negotiation).
06		GLS_UART1_BUFFER	n	0400	UART 1 (target) buffer size.
07		GLS_UART1_BLOCK_R	n	001E	UART 1 (target) read blocking time (milliseconds).
08		GLS_UART1_BLOCK_W	n	0064	UART 1 (target) write blocking time (milliseconds).
09		GLS_USB_BUFFER	n	0400	USB buffer size.
0A		GLS_USB_BLOCK_R	n	0014	USB read blocking time (milliseconds).
0B		GLS_USB_BLOCK_W	n	0000	USB write blocking time (milliseconds).
0C		GLS_PROT_AWARE	s	0001	Protocol awareness: finish reception immediately after detecting ETX (and BCC).
0D		<reserved>	–	0000	
0E		GLS_IPADDR_MSW	n	0000	TCP/IP address (most significant word).

0F	GLS_IPADDR_LSW	n	0000	TCP/IP address (least significant word).
10	GLS_NETMASK_MSW	n	0000	TCP/IP netmask (most significant word).
11	GLS_NETMASK_LSW	n	0000	TCP/IP netmask (least significant word).
12	GLS_GATEWAY_MSW	n	0000	TCP/IP gateway (most significant word).
13	GLS_GATEWAY_LSW	n	0000	TCP/IP gateway (least significant word).
14	GLS_TCP_PORT	n	0835	TCP port to use for TCP/IP connections.
15	GLS_FIND_PORT	n	093A	Multicast port to use for device detection.
16	GLS_DEFAULT_HOST	n	0002	Host to send AutoExec responses to (2=UART, 4=USB, 5=TCP/IP).
17	GLS_CMD_TIMEOUT	n	1388	Time until internal command tracking automatically frees target access.
18	GLS_ABORT_TIME	n	01F4	Time between received ESC and forced reset of internal command tracking.
19	GLS_TERM_TIMEOUT	n	2710	Terminal menu activation timeout (milliseconds).
1A	GLS_TERM_CRITICAL	s	0001	Stop terminal activation timeout at first received byte other than the activation character (0=no; 1=yes).
1B	GLS_UART2_BAUD	n	2580	UART 2 (secondary reader) baud rate.
1C	GLS_UART2_BLOCK_R	n	001E	UART 2 (secondary reader) read blocking time (milliseconds).
1D	GLS_UART2_BLOCK_W	n	01F4	UART 2 (secondary reader) write blocking time (milliseconds).
1E	GLS_UART2_RX_CRLF	s	0001	UART 2 (secondary reader) reception stopped by CR/LF (0=no; 1=yes).
1F	GLS_COMBI_ESC	s	0001	Received ESC from host terminates all standalone routines (0=no; 1=yes).
20	GLS_RTC_OSC_ADJ	n	0000	Realtime clock oscillator adjustment.
21	GLS_GSM_TIMEOUT	n	0014	UART byte timeout of GSM interface.
22	GLS_DEBUG	s	0000	Global debug mode switch (0=off; 1=on).
23	GLS_LOGLEVEL	n	0000	Log level for log file (0=off; 1...3=on).
24	GLS_GSM_INIT_TIME	n	0014	Timeout for GSM module startup (seconds).
25	GLS_USB_TX_NO_ZLP	s	0000	Disable USB ZLP (zero length packet) after transmissions of packet boundary size (0=off; 1=on).
26	GLS_USB_BLOCK_KE	n	0064	USB HID write blocking time (milliseconds).
27	GLS_USB_HID	s	0000	Special output to USB HID interface instead of standard USB serial link (0=off; 1=on).
28	GLS_USB_MODE	n	0001	USB mode: 1=Serial; 2=Composite MSD; 3=Composite HID; 4=HID.
29	GLS_TCP_MAXRTX	n	0003	Limit TCP retransmissions to this count (0 to disable configurable limit).
2A	GLS_CUSTOM_FLAGS	n	0000	Proprietary custom flags, binary coded.
2B	GLS_GSM_TX_TIME	n	003C	GSM transmission timeout in seconds.
2C	GLS_SDC_SCBR	n	0004	SD card transfer speed.
2D	GLS_TCPCL_KEEPU	s	0001	Keep up connection in TCP/IP client mode (1) or close connection after each data packet (0).
2E	GLS_IPCLADDR_MSW	n	0000	Host address for TCP/IP client mode, most significant 16 bit word.
2F	GLS_IPCLADDR_LSW	n	0000	Host address for TCP/IP client mode, least significant 16 bit word.
30	GLS_TCPCLPORT	n	0000	TCP port for TCP/IP client mode.
31	GLS_USB_HID_CC	n	0000	USB HID keymap country code.
32	GLS_UART0_IDLE	n	003C	Automatically put host UART to sleep after this amount of seconds (0 to disable autosleep).

33	GLS_UART1_SLEEP	s	0000	Put reader UART to sleep while not used.
34	GLS_DHCP_TIMEOUT	n	001e	DHCP timeout (in minutes).
35	GLS_BT_MODE	n	0001	Bluetooth mode: 1 = Serial; 4 = HID.
36	GLS_FIFO_MODE	n	0000	FIFO buffer mode: 0 = disabled; 1 = auto; 2 = triggered; 3 = stxetx.
37	GLS_FIFO_DELAY	n	01f5	FIFO Tx delay for AUTO mode (in ms).
38	GLS_ASYNC_OUTPUT	s	0000	Enable (1) or disable (0) asynchronous output.
39	GLS_TB_DATA_SIZE	n	0010	Size of additional TagBuffer data (in bytes).
3a	GLS_FIFO_PURGE	n	0000	FIFO buffer overflow policy: 0 = block; 1 = purge old; 2 = discard new.
3b	GLS_UARTD_BAUD	n	2580	Debug UART: baud rate.
3c	GLS_UARTD_BUFFER	n	0000	R.F.U.
3d	GLS_UARTD_BLOCK_R	n	001e	Debug UART: read blocking time (milliseconds).
3e	GLS_UARTD_BLOCK_W	n	0000	Debug UART: write blocking time (milliseconds).
3f	GLS_LEDS_ENABLED	s	0001	Debug LEDs enabled (1) or disabled (0).

### 102B: Edit Interface Board Setting

This function is used to change the system constants of an SEC-1500 or similar interface board. See "request interface board settings" ("102A") for further information.

<i>Host to RF System Request</i>	
<b>STX</b>	"102B" <kk> <vvvv>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	Key.
v	Value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "102B"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 102C: Forward Command

This command is used only in combination with an interface board like the SEC-1500. Its purpose is to forward a command which would be intercepted (and executed) by the interface board directly to the target device. Simply spoken, a command which is sent via 102C will always be executed by the target device and in no way by the interface board.

<i>Host to RF System Request</i>	
<b>STX</b>	"102C" <cccc> [ <p...> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
c	Command, i. e. hex function number.



p	Parameters for command.
---	-------------------------

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "cccc" [ <p...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Command that was executed by target device.
p	Response parameters.

### 102D: Request Interface Board Version Number

This function provides some version information concerning the interface board.

<i>Host to RF System Request</i>	
<b>STX</b> "102D"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "102D" <m...> "/" <v...> "." <w...> "." <b...> [ "/" <s...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
m	Module name.
v	Version number, major part
w	Version number, minor part
b	Build number
s	Software features

### 102E: Get Local Device Name

This function will return the local device name.

<i>Host to RF System Request</i>	
<b>STX</b> "102E"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "102E" <n...>	
<b>ETX</b> {c}	
<i>Response Data</i>	

n	Device name.
---	--------------

### 102F: Set Local Device Name

Sets the local device name. The local device name is used by the Bluetooth environment.

<i>Host to RF System Request</i>	
<b>STX</b> "102F" <n...>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
n	Device name.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "102F"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 1068: Get SmartRead Configuration

This function applies to devices equipped with Scemtec's SmartRead feature. It offers a rather crude way to replicate SmartRead configurations between devices. It is imperative that those devices have the same firmware version, since there is no low-level integrity check. Loading a configuration dump into a device with a different version may lead to serious data corruption.

<i>Host to RF System Request</i>	
<b>STX</b> "1068"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1068" <dd...>	
<b>ETX</b> {c}	
<i>Response Data</i>	
d	Hex dump of current SmartRead configuration.

### 1069: Set SmartRead Configuration

This function applies to devices equipped with Scemtec's SmartRead feature. It can be used to load a previously generated SmartRead configuration hex dump into a different device (with the same firmware version). This is done in packets to make sure the device does not become overloaded.

<i>Host to RF System Request</i>	
----------------------------------	--

<b>STX</b>	"1069" <oooo> <1111> <dd...>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
o	Offset of packet within SmartRead Configuration in bytes.
l	Length of packet in bytes.
d	Hex dump packet of new SmartRead configuration.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1069" <ss>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status: 00: SMARTSET_OK - No error. 10: SMARTSET_PAD - Non-critical error (warning): Data is padded (too long). 11: SMARTSET_TRUNC - Non-critical error (warning): Data is truncated (too short). 20: SMARTSET_NO_DATA - Critical error: No data available at all. 21: SMARTSET_SIZE - Critical error: Invalid size in header. 22: SMARTSET_OVERFLOW - Critical error: Data does not fit in buffer. 30: SMARTSET_MAGIC - Critical error: Invalid magic number. 31: SMARTSET_VERSION - Critical error: Invalid header version. 32: SMARTSET_SUBCOUNT - Critical error: Invalid number of sub menus.

### 106A: Clear SmartRead Configuration

This function applies to devices equipped with Scemtec's SmartRead feature. It can be used to remove (i. e. disable) the current SmartRead configuration in order to return the device to a known state.

<i>Host to RF System Request</i>	
<b>STX</b>	"106A"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"106A"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 1070: Get Tag Buffer Item

This function applies to devices equipped with Scemtec's SmartRead feature and has an impact on the Tag Buffer in "Output Filter" mode. It can be used to retrieve one single item (specified by a

zero-based index) from the current tag buffer. The largest possible index is the result of the “Get Tag Buffer Item Count” function minus one, since we're zero-based.

<i>Host to RF System Request</i>	
<b>STX</b> "1070" <nnnn>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
n	Index of item to get (0 ≤ n ≤ count-1).

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1070" <tt> <dd...>	
<b>ETX</b> {c}	
<i>Response Data</i>	
t	Tag type ID (see SmartRead manual for tag type ID table).
d	Tag (ID) data.

### 1071: Add Tag Buffer Item

This function applies to devices equipped with Scemtec's SmartRead feature and has an impact on the Tag Buffer in “Output Filter” mode. It can be used to add a single item to the Tag Buffer. If the specified item is already contained in the Tag Buffer, the command has no effect. Once finished editing Tag Buffer contents, the command E009 can be used to store the changes to non-volatile memory.

<i>Host to RF System Request</i>	
<b>STX</b> "1071" <tt> <dd...>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	Tag type ID (see SmartRead manual for tag type ID table).
d	Tag (ID) data.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1071"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 1072: Get Tag Buffer Item Count

This function applies to devices equipped with Scemtec's SmartRead feature and has an impact on the Tag Buffer in “Output Filter” mode. It can be used to determine the current size of the Tag Buffer, i. e. how many items are currently stored.

<i>Host to RF System Request</i>	
----------------------------------	--

<b>STX</b>	"1072"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1072" <cccc>
<b>ETX</b>	{c}
<i>Response Data</i>	
c	Number of items currently present in the Tag Buffer (hex).

### 1073: Clear Tag Buffer

This function applies to devices equipped with Scemtec's SmartRead feature and has an impact on the Tag Buffer in "Output Filter" mode. It can be used to remove all items from the Tag Buffer. Once finished editing Tag Buffer contents, the command E009 can be used to store the changes to non-volatile memory.

<i>Host to RF System Request</i>	
<b>STX</b>	"1073"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1073"
<b>ETX</b>	{c}
<i>Response Data</i>	
–	None.

### 1074: SmartRead Resume

A device configured to SmartRead stand-alone operation, which has been switched to STX/ETX mode by sending an ESC character, can be returned to SmartRead operation using this command.

<i>Host to RF System Request</i>	
<b>STX</b>	"1074"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1074"

<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 1075: Set Tag Buffer Filter

This function applies a filter to the Tag Buffer functions 1070, 1072 and 1073. After executing this command, the operations apply only to entries matching the configured filter.

<i>Host to RF System Request</i>	
<b>STX</b>	"1075" <ff>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-f	Filter value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1075"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 1076: Query SmartRead Configuration

This function returns the name of the currently active SmartRead configuration.

<i>Host to RF System Request</i>	
<b>STX</b>	"1076"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1076" <n...>
<b>ETX</b>	{c}
<i>Response Data</i>	
n	Name of currently selected SmartRead configuration.

### 1077: Switch SmartRead Configuration

This function can be used to either switch to a specified SmartRead configuration or cycle to the next SmartRead configuration having its "active" flag set.

<i>Host to RF System Request</i>	
<b>STX</b>	"1077" [ <n...> ]

<b>ETX</b> {c}	
<i>Request Parameters</i>	
n	Name of SmartRead configuration to activate; if omitted, the next available (active) SmartRead configuration will be selected.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1077" <n...>	
<b>ETX</b> {c}	
<i>Response Data</i>	
n	Name of currently selected SmartRead configuration after performing this command.

### 1078: Get FIFO Buffer Total Size

For devices equipped with FIFO buffer memory, this function returns the total size of the FIFO buffer.

<i>Host to RF System Request</i>	
<b>STX</b> "1078"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1078" <bbbbbb>	
<b>ETX</b> {c}	
<i>Response Data</i>	
b	Total number of bytes available in the FIFO buffer.

### 1079: Get FIFO Buffer Used Bytes

For devices equipped with FIFO buffer memory, this function returns the number of occupied bytes in the FIFO buffer.

<i>Host to RF System Request</i>	
<b>STX</b> "1079"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1079" <bbbbbb>	
<b>ETX</b> {c}	
<i>Response Data</i>	

b	Number of bytes in use.
---	-------------------------

### 107A: Read from FIFO Buffer

This function can be used to read data sequentially from the FIFO buffer. A maximum is specified to avoid buffer overflows. If less data than the maximum is available in the FIFO buffer, all data is transmitted, leaving the FIFO buffer empty. If the FIFO buffer is already empty when calling this function, it returns an empty result.

Please note that this function is not packet-oriented (e. g. when using SmartRead), but only byte-oriented.

<i>Host to RF System Request</i>	
<b>STX</b>	"107A" <mmmm>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
m	Maximum number of bytes to read.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"107A" [ <d...> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 107B: Write to FIFO Buffer

Although it is currently not necessary to write the FIFO buffer during normal operation, it may be desirable for debugging (or for special applications) to also write data to the FIFO buffer. That's where this function comes in.

<i>Host to RF System Request</i>	
<b>STX</b>	"107B" <d...>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
d	Data to write to FIFO.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"107B"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.



### 107C: Peek into FIFO Buffer

This function works similar to the “Read from FIFO Buffer” function, with the exception that data is not removed from the FIFO buffer while reading.

<i>Host to RF System Request</i>	
<b>STX</b>	"107C" <mmmm>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
m	Maximum number of bytes to read.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"107C" [ <d...> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
–	None.

### 107D: Discard FIFO Buffer

This function can be used to reset the FIFO buffer to empty status at any time.

<i>Host to RF System Request</i>	
<b>STX</b>	"107D"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"107D"
<b>ETX</b>	{c}
<i>Response Data</i>	
–	None.

### F003: Switch off Device

This command switches the Device completely off.

<i>Host to RF System Request</i>	
<b>STX</b>	"F003"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	

<b>STX</b>	"F003"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

## 2.2 2xxx: Periphery Control

### 2000: Determine Pushbutton Status

The Pushbutton request feature is system-dependent. For RF systems without pushbutton this request is without effect. The function can be carried out in both Single and Background mode; in Background mode each change of status of the pushbutton is transmitted to the PC.

<i>Host to RF system</i>	
<b>STX</b>	"2000" <m>
<b>ETX</b>	{c}
m	Mode: „S“ = single mode, „B“ = background mode.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"2000" <s> <t>
<b>ETX</b>	{c}
s	Pushbutton #2 status; „0“ = not pressed, „1“ = pushbutton #2 pressed
t	Pushbutton #1 status; „0“ = not pressed, „1“ = pushbutton #1 pressed

### 2003: LED Control

The LED control is system dependent. For RF systems without LED this request is without effect (or even not supported).

<i>Host to RF system</i>	
<b>STX</b>	"2003" <l1>
<b>ETX</b>	{c}
l	LED bit pattern; „00“ = no LED; „01“ = LED1 on; „02“ = LED 2 on; „03“ = LED1+2 on ...

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"2003"
<b>ETX</b>	{c}
-	No return data.

## 2004: Read Digital Input Ports

Queries the state of the digital input lines. The result is returned as a binary value, where port #1 is the LSB, followed by all others.

This command supports the GLS\_AUTO\_NOREPEAT setting. If activated, repetitions of identical responses are suppressed in auto execution mode (running from command buffer).

<i>Host to RF System Request</i>	
<b>STX</b> "2004"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2004" <pp>	
<b>ETX</b> {c}	
<i>Response Data</i>	
p	State of ports, binary coded.

## 2005: Write Digital Output Ports

Sets the general purpose digital outputs and relay lines. The first port is the LSB, all others follow. It is advised to use the more flexible commands 2008 and/or 2009 instead of this command.

<i>Host to RF System Request</i>	
<b>STX</b> "2005" <rr> <pp>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
r	State of relay port(s), binary coded.
p	State of digital output ports, binary coded.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2005"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

## 2008: Digital Output Control

In comparison to the "Write Digital Outputs" command, this one allows for a more specific control, i. e. you can also control which bits are to be changed and which to be left untouched using a bit mask. Relay ports cannot be changed with this command.

<i>Host to RF System Request</i>	
----------------------------------	--

<b>STX</b> "2008" <mmmm> <pppp>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Bit mask as a hex value.
p	Bit values as a hex value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2008"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

## 2009: Relay Control

In comparison to the "Write Digital Outputs" command, this one allows for a more specific control, i. e. you can also control which bits are to be changed and which to be left untouched using a bit mask. Normal digital output ports cannot be changed with this command.

<i>Host to RF System Request</i>	
<b>STX</b> "2009" <mmmm> <pppp>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Bit mask as a hex value.
p	Bit values as a hex value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2009"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

## 200A: Read Realtime Clock

This command reads the current time from a connected realtime clock.

It supports the GLS\_AUTO\_NOREPEAT setting. If activated, repetitions of identical responses are suppressed in auto execution mode (running from command buffer). In fact, this will produce a "heartbeat" message every second.

<i>Host to RF System Request</i>	
<b>STX</b> "200A"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "200A" <ss> <mm> <hh> <ww> <dd> <mm> <yy> <l> <o>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Seconds.
m	Minutes.
h	Hours.
w	Day of week.
d	Day of month.
m	Month.
y	Year.
l	Low battery (0=ok; 1=low).
o	Oscillator was stopped (0=ok; 1=stopped; 2=power on reset).

## 200B: Write Realtime Clock

This command writes a new time to an optional realtime clock.

<i>Host to RF System Request</i>	
<b>STX</b> "200B" <ss> <mm> <hh> <ww> <dd> <mm> <yy>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
s	Seconds.
m	Minutes.
h	Hours.
w	Day of week.
d	Day of month.
m	Month.
y	Year.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "200B"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

## 200C: Get antenna MUX status

This command can be used to query configuration and state of the antenna multiplexing part of the reader.

Up to two multiplexers (identified via hardware as “primary” and “secondary”) can be addressed simultaneously, including power and phase configuration. If only a MUX hardwired as “secondary”

is attached, it is addressed as “primary”, so that a single attached MUX is always referenced as “primary”.

<i>Host to RF System Request</i>	
<b>STX</b>	"200C"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"200C" <a> <b> <c> <d> <e> <f> <g> <h>
<b>ETX</b>	{c}
<i>Response Data</i>	
a	Power step (HEX value starting at 1, maximum value supported by protocol is F).
b	Phase inversion status for devices supporting phase inversion (1=0°; 2=180°).
c	(Primary) MUX antenna port (0-4 or 0-8).
d	Secondary MUX antenna port (0-4 or 0-8).
e	Highest supported power step (maximum value supported by protocol is F).
f	Highest supported phase parameter (0 if no power splitter, 2 if power splitter is equipped).
g	(Primary) MUX antenna port count (4, 8 or 0 if no MUX present).
h	Secondary MUX antenna port count (4, 8 or 0 if no secondary MUX present).

## 200D: Set antenna MUX

This command can be used to set the antenna multiplexing part of the reader to a specific configuration.

Up to two multiplexers (identified via hardware as “primary” and “secondary”) can be addressed simultaneously, including power and phase configuration. If only a MUX hardwired as “secondary” is attached, it is addressed as “primary”, so that a single attached MUX is always referenced as “primary”.

It is also possible to switch only phase inversion and/or power step by specifying both primary and secondary antenna port as 0 to leave all antenna ports unchanged.

<i>Host to RF System Request</i>	
<b>STX</b>	"200D" <a> <b> <c> <d>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Power step (HEX value starting at 1, maximum value supported by protocol is F; 0 to keep currently active power step).
b	Phase inversion for devices supporting phase switching (1=0°; 2=180°; 0 to keep currently active phase configuration).
c	(Primary) MUX antenna port (1-4 or 1-8; 0 to deactivate all antennas on primary MUX).
d	Secondary MUX antenna port (1-4 or 1-8; 0 to deactivate all antennas on secondary MUX).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "200D"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

## 200E: Read Temperature Sensor

This command can be used to read the current value of a specific temperature sensor. If only one sensor is present in the system, use index 00. If the temperature exceeds a limit specified in the global settings, a warning level is issued. The warning level starts with 1, and increases every 10 degrees by which the temperature is above the limit.

This command supports the GLS\_AUTO\_NOREPEAT setting. If activated, repetitions of identical responses are suppressed in auto execution mode (running from command buffer).

<i>Host to RF System Request</i>	
<b>STX</b> "200E" <ii>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
I	Index of sensor to be read.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "200E" <tttt> <ww>	
<b>ETX</b> {c}	
<i>Response Data</i>	
t	Temperature in Kelvin (hex).
w	Warning level: 01: 1-10 degrees. 02: 11-20 degrees. 03: ...

## 200F: Activate Buzzer

This function activates a connected buzzer for a given period of time. It returns when the specified time has elapsed and the buzzer is turned off again.

<i>Host to RF System Request</i>	
<b>STX</b> "200F" <tttt>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	Time in milliseconds.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "200F"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

## 2010: Set LCD Text

To write any amount of characters to a connected LCD, this function can be used. Please make sure that the length of the specified text does not exceed the LCD's character capacity.

<i>Host to RF System Request</i>	
<b>STX</b> "2010" <xx> <yy> <t...>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
x	X position (column) to start at.
y	Y position (line) to start at.
t	Text to display.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2010"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

## 2020: Request Port Sequence

This command is used to obtain one specified step of the current port sequence which is stored in the reader. Usually, the port sequence consists of two or more steps. The port sequence is used by the reader's port sequencer, which can be used to e. g. autonomously control an external antenna multiplexer.

The port sequencer mode is configured via the system settings (GLS\_PORTSEQ\_MODE) and supports the following configurations:

<i>Mode</i>	<i>Description</i>
0000	Port Sequencer deactivated (factory default).
0001	Step mode: Commands which support the Port Sequencer (e. g. EAS, Inventory) automatically switch to the next sequence step <i>before</i> next execution. Use this mode for example if anticollision is to be combined with further tag processing like reading/writing block data.
0002	Auto mode: Commands which support the Port Sequencer (e. g. EAS, Inventory) automatically process the complete sequence and return a cumulated result. Use this mode for example if pure anticollision at maximum rate is desired.



Each single port sequence step consists of one 16-bit word and is compatible to the 200C/200D antenna MUX commands.

<i>Host to RF System Request</i>	
<b>STX</b> "2020" <iiii>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of the sequence step to retrieve, beginning with 0000.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2020" <a> <b> <c> <d>	
<b>ETX</b> {c}	
<i>Response Data</i>	
a	Power step (HEX value starting at 1, maximum value supported by protocol is F; 0 to keep currently active power step).
b	Phase inversion for devices supporting phase switching (1=0°; 2=180°; 0 to keep currently active phase configuration).
c	(Primary) MUX antenna port (1-4 or 1-8; 0 to deactivate all antennas on primary MUX).
d	Secondary MUX antenna port (1-4 or 1-8; 0 to deactivate all antennas on secondary MUX).

## 2021: Edit Port Sequence

This function is used to edit a single step of the reader's port sequence. Please see the "Request Port Sequence" command ("2020") for more information on the port sequence structure. The sequence length, which is kept in the system settings (GLS\_PORTSEQ\_LEN), will be automatically adapted to the given index, so it is necessary to submit the single port sequence steps in their natural order.

<i>Host to RF System Request</i>	
<b>STX</b> "2021" <iiii> <a> <b> <c> <d>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of the sequence step to set, beginning with 0000.
a	Power step (HEX value starting at 1, maximum value supported by protocol is F; 0 to keep currently active power step).
b	Phase inversion for devices supporting phase switching (1=0°; 2=180°; 0 to keep currently active phase configuration).
c	(Primary) MUX antenna port (1-4 or 1-8; 0 to deactivate all antennas on primary MUX).
d	Secondary MUX antenna port (1-4 or 1-8; 0 to deactivate all antennas on secondary MUX).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2021"	
<b>ETX</b> {c}	
<i>Response Data</i>	

-	None.
---	-------

## 2022: Get Port Sequencer Step

This command is used to obtain the current step number of the reader's port sequencer.

<i>Host to RF System Request</i>	
<b>STX</b> "2022"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2022" <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current step number of port sequencer.

## 2023: Set Port Sequencer Step

This command is used to set the reader's port sequencer to a specific step number.

<i>Host to RF System Request</i>	
<b>STX</b> "2023" <ssss>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
s	New step number for port sequencer.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2023"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

## 2024: Iterate Port Sequencer

This command is used to externally switch the reader's port sequencer to its next step.

<i>Host to RF System Request</i>	
<b>STX</b> "2024"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2024" <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current step number of port sequencer.

### 2030: Start GSM Subsystem

The reader's GSM module (if present) is started using either the current configuration or, if provided, new parameters.

<i>Host to RF System Request</i>	
<b>STX</b> "2030" [ <pppp> [ '+' <s...> [ '+' <n...> ] ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	SIM PIN.
s	SMSC number.
n	Target number.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2030" <ss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value: 00h: Ok. 01h: Error starting SMS system. 02h: Error registering to network. 03h: Error during basic GSM configuration.

### 2031: Shutdown GSM Subsystem

Shuts down the reader's GSM module (if present).

<i>Host to RF System Request</i>	
<b>STX</b> "2031"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2031"	
<b>ETX</b> {c}	
<i>Response Data</i>	

-	None.
---	-------

### 2032: Submit AT Command

This command is used to send a standard AT command to the reader's GSM module. Please note that an AT command may send multiple responses.

<i>Host to RF System Request</i>	
<b>STX</b> "2032" <c...>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
c	AT command (without AT prefix).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2032" [ <r...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
r	Response to AT command.

### 2034: Read SMS Buffer

This command is used to read a single message from the reader's SMS buffer.

<i>Host to RF System Request</i>	
<b>STX</b> "2034" <ii>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of message to read.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2034"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 2035: Add Message To SMS Buffer

This command is used to add a message to the reader's SMS buffer. An SMS may consist of up to 140 bytes (which makes 280 characters). It is strongly recommended to transmit the SMS in several chunks to make sure the reader's receive buffer is not overloaded.

<i>Host to RF System Request</i>	
<b>STX</b> "2035" <ii> <oo> <cc> <dd...>	
<b>ETX</b> {c}	

<i>Request Parameters</i>	
i	Index of message to write.
o	Offset in bytes (2 hex characters make up one byte).
c	Number of bytes minus one, e. g. '03' to write blocks of 4 bytes.
d	SMS data (2 hex characters each byte).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2035"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

### 2036: Get Sender Number From SMS Buffer

This command returns the sender number corresponding to a specified index within the SMS buffer. If the specified message was not received but written to the buffer, an empty response is generated.

<i>Host to RF System Request</i>	
<b>STX</b> "2036" <i i>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of message to which the corresponding sender number is requested.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2036" [ '+' <n...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
n	Sender number.

### 2037: Initialize SMS Buffer

This function deletes the reader's SMS buffer.

<i>Host to RF System Request</i>	
<b>STX</b> "2037"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2037"	
<b>ETX</b> {c}	
<i>Response Data</i>	

-	None.
---	-------

### 203A: Receive Messages To SMS Buffer

This command transfers all messages from the SIM card into the reader's SMS buffer. These messages are then immediately removed from the SIM!

<i>Host to RF System Request</i>	
<b>STX</b>	"203A"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"203A" <cc>
<b>ETX</b>	{c}
<i>Response Data</i>	
c	Number of messages received.

### 203B: Transmit Messages From SMS Buffer

This command transmits all messages currently stored in the reader's SMS buffer.

<i>Host to RF System Request</i>	
<b>STX</b>	"203B"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"203B" <cc>
<b>ETX</b>	{c}
<i>Response Data</i>	
c	Number of messages successfully transmitted.

### 203C: Delete All Messages From SIM

This function removes all messages from the currently inserted SIM card.

<i>Host to RF System Request</i>	
<b>STX</b>	"203C"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "203C"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

## 2040: Get GSM/GPS Version

This command can be used to read the version information from the GSM and/or GPS module.

<i>Host to RF System Request</i>	
<b>STX</b> "2040" <v>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
v	Which version to read: 0: GSM module. 1: GPS module.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2040" [ <d...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
d	Version data.

## 2048: Get GPS Fix

This command can be used to obtain a GPS fix.

<i>Host to RF System Request</i>	
<b>STX</b> "2048" <f> <tttt>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
f	Fix type: 1: Get GPS timestamp only (fastest). 2: Get 2D fix (slow). 3: Get 3D fix (slowest).
t	Timeout in seconds.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "2048" <yyyy>-<mm>-<dd>/<hh>:<mm>:<ss> [ /<aaa°aa'aa.aaa"a>/<ooo°oo'oo.ooo"o> [ /<llll.l>m ] ]	

<b>ETX</b> {c}	
<i>Response Data</i>	
y	Timestamp: year.
m	Timestamp: month.
d	Timestamp: day.
h	Timestamp: hour.
m	Timestamp: minute.
s	Timestamp: second.
a	2D fix: Latitude.
o	2D fix: Longitude.
l	3D fix: Altitude.

## 2.3 3xxx: Transponder Recognition

### 3000: General Transponder Recognition

With this function you can find out which transponder type is in reading distance to the antenna. In case of some transponder types a default configuration is assumed, since these transponders can also emulate other transponder types.

<i>Host to RF system</i>	
<b>STX</b> "3000"	
<b>ETX</b> {c}	
-	No request parameters.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "3000" <tt>	
<b>ETX</b> {c}	
T	Transponder type. See code number given for the "100A" command. A value of "00" means that no transponder was found.

## 2.4 4xxx: Transponder Read Functions

Note that for some transponders certain memory areas may be read-protected by password or status bits.

### 4000: Read Transponder (All Types)

With the „Read transponder“ function all supported transponder types can be read. In case of some transponder types, a default configuration is assumed, since these transponders can also emulate other transponder types. When writing via function "5000", these transponders are configured to default settings. You can select between different response formats.

<i>Host to RF system</i>	
--------------------------	--



<b>STX</b>	"4000" <aaaa> <nn> <f>
<b>ETX</b>	{c}
a	Start address from where to read.
n	Number of bytes to be read.
f	Response format: „A“ = ASCII, „H“ = HEX, „D“ = decimal, „B“ = binary.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4000" <d> [ <d> [ ... ] ]
<b>ETX</b>	{c}
d	Read data; the number of transmitted data characters depends on the number of requested bytes and on the data format used. In ASCII, the number of returned characters is identical to the number requested; in HEX format, the number is doubled, in decimal format it is tripled, and in binary format 8 times as many characters are returned.

If data areas are to be accessed that lie outside the transponder's address range, an error message is returned!

#### 4F00: Read Barcode

With this command the Laser Scanner is switched on. For a time of 5 sec the the engine searches for a barcode. The data then will be sent back.

<i>Host to RF system</i>	
<b>STX</b>	"4F00"
<b>ETX</b>	{c}
-	No request parameters.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4F00" <d> [ <d> [ ... ] ]
<b>ETX</b>	{c}
d	Barcode data.

## 2.5 5xxx: Transponder Write Functions

Note that for some transponders certain memory areas may be write-protected by password or status bits.

#### 5000: Write to Transponder (All Types)

With the „Write to transponder“ function all read/write transponders supported by the RF system can be written to. You can access a transponder byte by byte, independent of its organization and access type. You may have to read the transponder before doing so. This function configures the

transponders to their default configuration so that they can be read via the general Read function "4000". In case of writing beyond the transponder's writeable range, an error message is generated.

<i>Host to RF system</i>	
<b>STX</b>	"5000" <aaa> <f> <d> [ <d> [ ... ] ]
<b>ETX</b>	{c}
a	Start address from where to write. The lowest writeable address is 0; for this, „0000“ must be transmitted. If you want to write, e. g., from address 12 onwards, „000C“ must be transmitted.
f	Format character; „A“ = ASCII, „H“ = HEX, „D“ = decimal.
d	Characters to be transmitted; the number of characters to be transmitted depends on the format used. In ASCII, the characters are transmitted to the transponder 1:1, in HEX format it is 2 characters and in decimal format it is 3 characters that make up one byte to be written.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5000"
<b>ETX</b>	{c}
-	No return data.

## 2.6 6xxx: Transponder Specific Functions

### 6000: Activate Password Mode (General Write/Read Functions)

With this function the required password is transmitted to the RF system; in addition, Password mode is switched on in the RF system (not in the transponder!). To be able to use the password, some transponders require setting the corresponding status bits in the transponder control registers. After having called the function, you can write to password protected transponders or read them (exception: the Mikron Hitag HT1-DCx crypto transponder). You may use passwords of any length for the function; missing bytes are replaced by zeroes, excess bytes are cut off. The is recognized and the password applied to it. Password mode remains active until it is either deactivated or a new mode is activated with another password.

**CAUTION:** The password transmitted via interface can e. g. be intercepted by a „Line Listener“.

The Password function refers to both the general and the specific Write/Read functions. Some transponder types may also be used with a transponder-specific password. Such a password will then have priority over the general password only for the respective transponder.

<i>Host to RF system</i>	
<b>STX</b>	"6000" <f> <p> [ <p> [ ... ] ]
<b>ETX</b>	{c}

f	Format character for password transfer; „A“ = ASCII, „H“ = HEX.
p	Password data; the number of characters to be transmitted depends on the format used.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6000"
<b>ETX</b>	{c}
-	No return data.

The following password sizes have been determined:

<i>Transponder Type</i>	<i>Password Size</i>
Philips PCF7930	7 bytes
GEMPLUS	7 bytes
EM V4050	4 bytes
SID Titan	4 bytes
Temic e5550	4 bytes

### 6001: Deactivate Password Mode (General Write/Read Functions)

Switches off general Password mode. Afterwards, protected data can only be accessed via the activated transponder-specific passwords.

<i>Host to RF system</i>	
<b>STX</b>	"6001"
<b>ETX</b>	{c}
-	No request parameters.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6001"
<b>ETX</b>	{c}
-	No return data.

### 6002: Write New Password (General)

To overwrite a transponder's existing password, you must first have activated Password mode with the current general password, otherwise the function returns an error message. To be able to use the password, some transponders require setting the corresponding status bits in the transponder control registers. If the new password could be written, the current general password is replaced by the new entry, which will be used for all subsequent write/read accesses.

<i>Host to RF system</i>	
<b>STX</b>	"6002" <f> <p...>
<b>ETX</b>	{c}
f	Format character for password transfer; „A“ = ASCII, „H“ = HEX.

p	Password data; the number of characters to be transmitted depends on the format used.
---	---

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6002"
<b>ETX</b>	{c}
-	No return data.

Caution: The password transmitted via interface can e. g. be intercepted by a „Line Listener“.

The Password function refers to both the general and the specific Write/Read functions. Some transponder types may also be used with a transponder-specific password. Such a password will then have priority over the general password only for the respective transponder.

### 6010: Set Password Buffer

This function can be used to write a password to the reader's global password buffer. The password buffer can afterwards be saved using the “Store Settings” command. Transponder specific functions can be used to transmit a password from the password buffer to the transponder.

<i>Host to RF System Request</i>	
<b>STX</b>	"6010" <pp...>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
p	Password in Hex notation. Length is currently limited to 8 bytes, i. e. 16 hex characters.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6010"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

## 2.7 Exxx: Non-volatile memory commands

These commands, some of which are not revealed in this document (for they may cause permanent damage to the system if not used correctly), control data storage in non-volatile memory, i. e. flash or EEPROM.

### E000: Recover Factory Settings

This function is used to restore all system settings to the manufacturer defaults.

<i>Host to RF System Request</i>	
<b>STX</b>	"E000"
<b>ETX</b>	{c}

<i>Request Parameters</i>	
–	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "E000"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

### E001: Recover User Settings

This function is used to restore all system settings to the previously saved (“frozen”) user-specified values.

<i>Host to RF System Request</i>	
<b>STX</b> "E001"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "E001"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

### E002: Freeze Current Settings

This function is used to save all settings (tag-independent system settings as well as tag-related settings) to non-volatile memory. On reset or power on these settings will be automatically loaded.

<i>Host to RF System Request</i>	
<b>STX</b> "E002"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "E002"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

### E008: Recover Tag Buffer

This function applies to devices equipped with Scemtec's SmartRead feature and has an impact on the Tag Buffer in "Output Filter" mode. It is used to restore the Tag Buffer to the previously saved ("frozen") state.

<i>Host to RF System Request</i>	
<b>STX</b>	"E008"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"E008"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### E009: Freeze Current Tag Buffer

This function applies to devices equipped with Scemtec's SmartRead feature and has an impact on the Tag Buffer in "Output Filter" mode. It is used to store the current Tag Buffer to non-volatile memory; it will then be loaded automatically once SmartRead gets active in Output Filter mode.

<i>Host to RF System Request</i>	
<b>STX</b>	"E009"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"E009"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### E080: Read From EEPROM User Space

This function can be used to read raw data from an EEPROM (user space).

<i>Host to RF System Request</i>	
<b>STX</b>	"E080" <aaaa> <nn>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Address of first byte to be read.

n	Number of bytes to be read minus one (currently supported: 00 – 0f).
---	--

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "E080" <dd> [ <dd> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
d	EEPROM data.

### E081: Write To EEPROM User Space

This function can be used to write raw data to an EEPROM (user space).

<i>Host to RF System Request</i>	
<b>STX</b> "E081" <aaaa> <dd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Destination address of byte to be written.
d	Byte data.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "E081"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

### E082: Get EEPROM User Space Size

This function returns the size of the EEPROM user space in bytes.

<i>Host to RF System Request</i>	
<b>STX</b> "E082"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "E082" <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Size of EEPROM user space in bytes.

### E083: Erase EEPROM User Space

To completely clear the EEPROM user space to zero, use this function. This process may take some time.

<i>Host to RF System Request</i>	
<b>STX</b>	"E083"
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"E083"
<b>ETX</b>	{ c }
<i>Response Data</i>	
–	None.

## 2.8 Fxxx: Service Commands

The Service commands allow for direct access to the function of the RF system. They include hardware control functions that are automatically controlled by the RF system. These functions can be “manually“ accessed via the Service commands.

### F000: Switch on/off Oscillator

With this function the oscillator of the RF system (and thus the RF carrier) can be switched on or off respectively. Note that the Write/Read functions may change the status set via this function.

<i>Host to RF System Request</i>	
<b>STX</b>	"F000" <zz>
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
z	New oscillator state: 00: All carriers off 01: Primary carrier on. 02: Auxiliary carrier on.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"F000"
<b>ETX</b>	{ c }
<i>Response Data</i>	
–	None.



**F001: HF Reset**

Switches the carrier off long enough to allow all tags in the field to be initialized.

<i>Host to RF System Request</i>	
<b>STX</b>	"F001"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"F001"
<b>ETX</b>	{c}
<i>Response Data</i>	
–	None.

## 3 Reader Device Specifics

This chapter will point out several characteristics which are not specific to a single transponder type but rather to a certain reader device. This includes special behaviour of some general commands as well as key/value tables and factory defaults.

### 3.1 125KHz Systems

#### Default Settings after Switching on or Resetting

After having switched on or reset the RF system, it returns to its default settings. These include:

- Serial interface: 9600 baud, 1 stop bit, no parity bit, no hardware handshake
- Time-out of receipt routine: 500 ms
- All ports switched off (LEDs off)
- No active Background function
- Password mode off
- All supported transponder types released for general transponder functions

The default settings for General Transponder Functions are as follows:

- Philips PCF7930: Memory range: block 2 to block 7
- Mikron HT1-DCx: Memory range: block 8 to block 15 (page 32 to page 63)
- EM V4050 - Sokymat Titan: Memory range: block 3 to block 33  
Read-After-Write switched on
- Temic e5530: Memory: 8 bytes (2 blocks)  
Data frequency: RF/64  
Encoding: bi-phase  
Header: E6 (8 bits)
- Temic e5550: Memory range: block 1 to block 6  
Data frequency: RF/64  
Encoding: Manchester  
Terminators: sequential terminators
- Temic Q5: Data frequency: RF/64  
Encoding: Manchester

Default settings of other special transponder types can be found in the respective transponder type descriptions.

## 3.2 13.56MHz Systems

### General

The SHL-2xxx, STK-26xx, STK-27xx, STK-2020, and SHT-20xx devices provide full anticollision functionality combined with high performance long range operation on all currently available 13.56MHz transponders, including ICode-1, ICode EPC, ICode UID, Tag-it and the ISO 15693 standard.

### System Settings

System settings are queried with the “1010” command and altered with the “1011” command. The following settings are currently defined for the SHL-2xxx system (N/S=Numeric/Switch, Def.=Default):

Key	C	Alias	N/S	Def.	Description
01		GLS_ANTENNA	n	0000	Antenna port (for devices with on-board antenna mux). If supported, a value of 0 switches the multiplexer off.
02		GLS_POWER	n	0004	Approximate RF power on antenna (supported: 1-4, i. e. four different power steps).
03		GLS_SER_BAUD	n	2580	Baud rate (115200 baud = FFFFh; 230400 baud = FFFEh). Changes become active on reset and only if settings are stored.
04		GLS_SER_TIMEOUT	n	01F4	Serial port timeout (*1ms).
05		<reserved>	-	0000	
06		GLS_OSC_AUTO_OFF	n	01F4	Delay before auto carrier off (*1ms) – to make sure a tag gets enough power to process the first transaction.
07		<reserved>	-	0000	
08		<reserved>	-	0000	
09		GLS_SWR_THRES_HI	n	1F40	Upper SWR threshold. When antenna SWR exceeds this value, DETUNED LED is used to display an error and the RF carrier is blocked until SWR is below threshold again.
0A		GLS_CRC_TAG_LED	s	0000	Tag LED on when CRC error occurred.
0B		GLS_INT_ENABLE	s	0001	Interrupt mode: make some functions “breakable”. To abort a function, send an ESC character to the reader during the time between ACK and the rest of the response.
0C		GLS_VERBOSE_START	s	0000	Reset response on startup (i. e. “ACK STX 1000...”).
0D		GLS_SYNC	n	0000	Reader Sync (0=off; 1=Master; 2=Slave; 3=Auto).
0E		GLS_SA_INIT_TIME	n	000A	Standalone RS232 wait delay (*1s).
0F		GLS_SA_INTERVAL	n	0001	Default wakeup interval (minutes).
10		GLS_RELAY_USAGE	n	0000	Relay usage (0=User; 1=EAS Alarm; 2=Tag Detect).
11		GLS_RTC_DST	s	0000	Switch RTC DST (0=Winter; 1=Summer).
12		GLS_SYNC_GROUPS	n	0001	Number of groups used for synchronization.
13		GLS_INV_RESET	s	0001	Allow HF reset on Create Inventory Commands.
14		GLS_ERROR_REBOOT	s	0000	Instant reboot in case of a system error.
15		GLS_TAG_DET_LEN	n	01F4	Length of "Tag Detect" alarm (*1ms).
16		GLS_EAS_ALARM_LEN	n	01F4	Length of EAS alarm (*1ms).

17		GLS_ANT_MEASURE	n	000A	Interval for field measurement; 0=off (*1s).
18		GLS_TEMP_MEASURE	n	0005	Interval for temperature measurement; 0=off (*1s).
19		GLS_DISPLAY_WAIT	n	09C4	Message display time for LCD (*1ms).
1A		GLS_SWR_THRES_LO	n	07D0	Lower SWR threshold. When antenna SWR exceeds this value, DETUNED LED is used to display a warning.
1B		GLS_TEMP_THRES_LO	n	014D	Lower temperature threshold. When internal temperature exceeds this value, SYS LED is used to display a warning.
1C		GLS_AUTOEXEC_MODE	n	0000	Mode of AutoExec subsystem; 0=off (see "Command Buffer" functions for more info). Changes become active on reset and only if settings are stored.
1D		GLS_PORTSEQ_MODE	n	0000	Mode of Port Sequencer; 0=off (see "Port Sequence" functions for more info).
1E		GLS_PORTSEQ_LEN	n	0000	Length of current port sequence.
1F		GLS_CMDBUF_LEN	n	0000	Length of currently buffered command.
20		GLS_AUTOEXEC_TIME	n	0000	Timer value (*1s) for AutoExec subsystem.
21		GLS_ANT_TUNE	n	0000	Most recent antenna relays configuration.
22		GLS_AMU_BISTABLE	s	0001	AMU relays are bistable.
23		GLS_DEBUG	s	0000	Debug mode (can have various effects, should be 0).
24	X	GLS_STXETX_NO_CHK	s	0000	Set to 1 to disable STX/ETX checksum.
25	X	GLS_STXETX_NO_FRM	s	0000	Set to 1 to disable STX/ETX frame (control characters).
26	X	GLS_HEARTBEAT	s	0000	Heartbeat interval for certain functions; 0=off.
27		GLS_SER_RTSCTS	s	0000	RTS/CTS Hardware flow control disabled/enabled. Changes become active on reset and only if settings are stored.
28		GLS_AUTO_FORCE	s	0000	By default, AutoExec is terminated immediately after serial data has been received. If this switch is on (1), AutoExec will remain active until disabled explicitly.
29	X	GLS_NO_TAG_LEN	n	0000	Length of "no tag" signal for certain custom functions.
2A	X	GLS_CYCLE_TIME	n	06A4	Maximum cycle time in milliseconds for certain custom functions.
2B	X	GLS_BLOCK_OFFSET	n	0000	Block offset for certain custom functions.
2C	X	GLS_PORT_CARR_OFF	s	0001	Always switch carrier off before switching ports (0=no; 1=yes).
2D		GLS_INV_LIMIT	n	0000	Limit inventory RAM to given number of words (0=no artificial limitation).
2E		GLS_PORTS_TAG_DET	n	0000	Binary bit pattern to set on the output ports on TAG DETECT event.
2F		GLS_PORTS_NO_TAG	n	0000	Binary bit pattern to set on the output ports on NO TAG event.
30		GLS_PORTS_EAS_AL	n	0000	Binary bit pattern to set on the output ports on EAS ALARM event.
31		GLS_INPUT_OVSMP	s	0000	Oversampling for digital inputs (only used in special applications), given in milliseconds.
32		GLS_AUTO_QUIET	s	0000	Drop empty responses in AutoExec mode.
33		GLS_DELAYED_ACK	s	0000	Do not transmit ACK character before the actual response is ready for transmission during normal operation.
34		GLS_AUTOEXEC_ACK	s	0000	Begin AutoExec response with ACK character.
35		GLS_AUTO_INIT_DLY	n	0032	Initial delay for AutoExec (*0.1s). This makes sure that the reader is ready for external configuration for a specified period of time (5 seconds default).

36	GLS_AUTO_NOREPEAT	s	0000	Drop repeatedly equal responses in AutoExec mode. This is only implemented for a few commands!
37	GLS_CMDBUF_COUNT	n	0000	Number of lines In command buffer.
38	GLS_BULK_DELAY	n	0064	Number of milliseconds to insert between packages in bulk data transfers (e. g. inventory). This helps ensure correct interoperability with interface modules like the SEC1500.
39	GLS_BULK_PKGSIZE	n	0000	Maximum size of a single package in bulk data transfers (e. g. inventory). After each package, a short delay (see above) is inserted.
3A	GLS_SA_NOSLEEP	s	0000	Suppress going to sleep in standalone operation.
3B	GLS_SA_OFF_DELAY	n	0000	Standalone: Delay between last transaction and going to sleep, specified in milliseconds.
3C	GLS_LANGUAGE	n	0000	Language identifier (for devices with both LCD and i18n support); 0 is default.
3D	GLS_EN_WORKAROUND	s	0001	Enable/disable specific workarounds (internal use).
3E	GLS_STARTUP_BUSY	s	0001	Send BUSY error if commands are received during startup (or nothing if 0 is configured here).
3F	GLS_TEMP_THRES_HI	n	0161	Upper temperature threshold. When internal temperature exceeds this value, SYS LED is used to display an error and the RF carrier is blocked until temperature is below threshold again.
40	GLS_SPLIT_BYPASS	s	0000	On devices equipped with an internal power splitter, this option can be used to bypass the splitter and concentrate all RF power on the first antenna output. In some cases, RF power will be automatically limited to avoid damage to external components.
41	GLS_REGION	n	0000	Regulatory Region (0=ETSI; 1=FCC; 2=Japan).
42	GLS_LCD_BL_TIME	n	0064	LCD backlight timeout (in ms).
43	GLS_LCD_TITLE_TIME	n	01f4	LCD title show timeout (in ms).
44	GLS_LCD_SCROLL_DLY	n	012c	LCD scroll delay (in ms).
45	GLS_RTC_OFFSET	n	0080	RTC offset (time zone) +80h; 0080 means 0 hours, 007f means -1 hour, 0081 means 1 hour, and so on.
46	GLS_KEY_TIMEOUT	n	000a	Default timeout for key press (in seconds).
47	GLS_LCD_ANIM_SPEED	n	0064	LCD animation delay time (in ms).
48	GLS_VBAT_SHOW	n	0004	Voltage threshold below which to show battery symbol.
49	GLS_VBAT_BLINK	n	0002	Voltage threshold below which to blink battery symbol.
4a	GLS_NATIVE_MUX	n	0000	Native antenna multiplexer configuration. If supported by hardware, this option can be set to a bit pattern of antennas to use (e. g. 0003 for antennas 1+2 or 0005 for antennas 1+3).
4b	GLS_RT_INV_LIMIT	n	0010	Global limitation of Realtime Inventory commands (number of IDs per response).
4c	GLS_HANDSHAKE	s	0000	Set to 1 to enable STX/ETX handshaking (used by Realtime Inventory): each subsequent response must be triggered by sending a single ACK character.
4d	GLS_INV_NO_FILTER	s	0000	Bypass per-cycle repetition filter for some inventory functions.

## Supplementary general command specification

### *Get Version*

In general, only the software feature “K” is defined, which means that the KSW “TempSens” and “VarioSens” transponder are supported. These tags are not featured by default.

For systems with SC-433 support (requires special hardware), the feature “H” is issued.

### *Flush Buffers*

When sent to a 13.56MHz device, the “Flush Buffers” command simply clears the current inventory, regardless of tag type.

## Common Status Values and Data Flag

Many transponder specific functions return a one-character status value, giving the host software the possibility to react in various ways.

<i>Value</i>	<i>Alias</i>	<i>Description</i>
0h	STAT_OK	Ok, no error.
1h	STAT_NO_LABEL	No tag found.
2h	STAT_COLL_CRC	Collision or CRC error.
3h	STAT_COLL_WEAK	Weak collision.
4h	STAT_NO_ACT_OK	Tag not halted or written, as desired.
5h	STAT_WRONG_SNR	Unexpected serial number in this timeslot.
6h	STAT_PROTECTED	Tag is write protected.
7h	STAT_FAILED	Failed to write/lock due to lack of power.
8h	STAT_NO_BLOCK	The requested block does not exist.
9h	STAT_CMD_ERROR	Error in tag command/parameters.
Ah	STAT_GEN_ERROR	Unspecified error.
Bh	STAT_VER_ERROR	Verify error.
Ch	STAT_CUST_1	Tag manufacturer specific status code #1.
Dh	STAT_CUST_2	Tag manufacturer specific status code #2.
Eh	STAT_CUST_3	Tag manufacturer specific status code #3.
Fh	STAT_CUST_4	Tag manufacturer specific status code #4.

The tag manufacturer specific status codes are defined as follows.

<i>Value</i>	<i>Infineon My-D</i>
Ch	Access denied.
Dh	Authentication error.
Eh	-
Fh	-

<i>Value</i>	<i>KSW VarioSens</i>
Ch	Wrong password; the “verified” state was not changed.

Dh	Block is write protected.
Eh	VarioSens is running.
Fh	Access to blocks 00h..0Fh is not allowed for ISO15693 commands.

Additionally, a so-called "data flag" is used to tell the host software if data follows (e. g. when a block has been read in the current slot) or not. This way, no dummy ("don't care") data needs to be transferred, resulting in a significantly faster serial transmission.

<i>Value</i>	<i>Description</i>
"y"	Yes, data follows.
"n"	No, this slot is empty due to an error (see status value).

For Tag-it (non-ISO) there are two additional status values.

<i>Value</i>	<i>Description</i>
"u"	Tag-it data follows and is user locked.
"f"	Tag-it data follows and is factory locked.

## 4 Philips PCF7930 Transponder

### 4.1 Features

- Read/write transponder
- 8 blocks à 16 bytes each = 128 bytes
- Block 0 and block 1 are configuration blocks
- Continuous data output (range of data user definable)
- Write protection via status bits and password
- Data frequency = RF/64

### 4.2 Transponder Recognition

#### 3100: Philips PCF7930 Transponder Recognition

This function determines in Single mode if there is a Philips PCF7930 tag in reading distance to the antenna or not.

<i>Host to RF system</i>	
<b>STX</b>	"3100" <m>
<b>ETX</b>	{c}
m	Mode: at the moment only „S“ = single mode.

  

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"3100" <xx>
<b>ETX</b>	{c}
x	„00“ = no PCF7930 transponder in reading distance, „01“ = transponder was recognized.

### 4.3 Transponder Read Functions

#### 4100: Read Philips PCF7930 Transponder

Reads a Philips PCF7930 transponder. The Philips PCF7930 transponder can only be read block by block, i. e. in blocks of 16 bytes each. Since direct addressing for reading the blocks is not possible, block assignment must be performed by the user, e. g. by determining the output range via read pointers. If more blocks are requested than have been released in the transponder, blocks are returned repeatedly. At the moment this function can only be used in Single mode.

<i>Host to RF system</i>	
<b>STX</b>	"4100" <m> <nn>
<b>ETX</b>	{c}
m	Read mode: at the moment only „S“ = single mode.



n	Number of blocks to be read, e. g. „01“ for 1 block, „05“ for 5 blocks, etc.
---	--

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "4100" <dd> [ <dd> [ ... ] ]	
<b>ETX</b> {c}	
d	Read data; the number of transmitted characters depends on the number of blocks to be read. 32 characters are transmitted per block, with 2 characters being combined to one data byte.

## 4.4 Transponder Write Functions

The PCF7930 transponder can be write protected via password or status bits.

### 5100: Write to Philips PCF7930 Transponder Block by Block

With this function you can write to a Philips PCF7930 transponder block by block, i. e. by 16 bytes each. Blocks 0 and 1 are configuration blocks, i. e. writing to these blocks can (negatively) influence the transponder's functionality!

If the transponder is in „write protected mode“, an error message is generated unless the correct password has been transmitted! At the moment, only the general password can be used as password.

<i>Host to RF system</i>	
<b>STX</b> "5100" <bb> <dddddddddddddddddddddddddddddd>	
<b>ETX</b> {c}	
b	Block number to be written to, e. g. „02“ for second or „07“ for seventh block.
d	Data to be written. For each byte to be written 2 characters must be transmitted.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "5100"	
<b>ETX</b> {c}	
-	No return data.

### 5101: Write to Philips PCF7930 Transponder Byte by Byte

With this function you can write to a Philips PCF7930 transponder byte by byte within a block. Blocks 0 and 1 are configuration blocks, i. e. writing to these blocks can (negatively) influence the transponder's functionality!

If the transponder is in “write protected mode“, an error message is generated unless the correct password has been transmitted! At the moment, only the general password can be used as password.

<i>Host to RF system</i>	
--------------------------	--

<b>STX</b>	"5101" <bb> <aa> <nn> <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
b	Block number to be written to, e. g. „02“ for second or „07“ for seventh block.
a	Byte start address (0..15) within a block; if you want to write, e. g., from address 10 onwards, „0A“ must be transmitted as parameter.
n	Number of bytes to be written from start address onwards. You can only write within one block, otherwise an error message is returned.
d	Data to be written. The number of characters to be transmitted depends on the number of bytes to be written; for each byte to be written 2 characters must be transmitted.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5101"
<b>ETX</b>	{c}
-	No return data.

### 5102: Write to Philips PCF7935 Transponder Block by Block

With this function you can write to a Philips PCF7935 transponder block by block, i. e. by 16 bytes each. Blocks 0 and 1 are configuration blocks, i. e. writing to these blocks can (negatively) influence the transponder's functionality!

If the transponder is in „write protected mode“, an error message is generated unless the correct password has been transmitted! At the moment, only the general password can be used as password.

<i>Host to RF system</i>	
<b>STX</b>	"5102" <bb> <dddddddddddddddddddddddddddddd>
<b>ETX</b>	{c}
b	Block number to be written to, e. g. „02“ for second or „07“ for seventh block.
d	Data to be written. For each byte to be written 2 characters must be transmitted.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5102"
<b>ETX</b>	{c}
-	No return data.

### 5103: Write to Philips PCF7935 Transponder Byte by Byte

With this function you can write to a Philips PCF7935 transponder byte by byte within a block. Blocks 0 and 1 are configuration blocks, i. e. writing to these blocks can (negatively) influence the transponder's functionality!

If the transponder is in “write protected mode“, an error message is generated unless the correct password has been transmitted! At the moment, only the general password can be used as password.

<i>Host to RF system</i>	
<b>STX</b>	"5103" <bb> <aa> <nn> <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
b	Block number to be written to, e. g. „02“ for second or „07“ for seventh block.
a	Byte start address (0..15) within a block; if you want to write, e. g., from address 10 onwards, „0A“ must be transmitted as parameter.
n	Number of bytes to be written from start address onwards. You can only write within one block, otherwise an error message is returned.
d	Data to be written. The number of characters to be transmitted depends on the number of bytes to be written; for each byte to be written 2 characters must be transmitted.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5103"
<b>ETX</b>	{c}
-	No return data.

## 4.5 Transponder-specific Functions

### 6100: Set Philips PCF7930 Read Pointer

The PCF7930 transponder can be write protected via password or status bits.

With this function you can set the Philips PCF7930 read pointers, i. e. determine the range of blocks to be subsequently output while reading the tag.

<i>Host to RF system</i>	
<b>STX</b>	"6100" <aa> <ee>
<b>ETX</b>	{c}
a	First block to be read, e. g. „00“ for block 0 or „02“ for block 2, etc.
e	Last block to be read, e. g. „05“ for block 5 or „07“ for block 7, etc.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6100"
<b>ETX</b>	{c}
-	No return data.

### 6105: Set Philips PCF7935 Read Pointer

The PCF7935 transponder can be write protected via password or status bits.

With this function you can set the Philips PCF7935 read pointers, i. e. determine the range of blocks

to be subsequently output while reading the tag.

<i>Host to RF system</i>	
<b>STX</b>	"6105" <aa> <ee>
<b>ETX</b>	{c}
a	First block to be read, e. g. „00“ for block 0 or „02“ for block 2, etc.
e	Last block to be read, e. g. „05“ for block 5 or „07“ for block 7, etc.
<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6105"
<b>ETX</b>	{c}
-	No return data.

## 5 Philips Hitag 1 / Hitag S Transponder

### 5.1 Features

- Read/write transponder
- 16 blocks à 16 bytes each or 64 pages à 4 bytes each = 256 bytes
- Page 0 is the ID page (serial number), page 1 is the configuration page, block 1 is the log-data block (login password)
- Read and write protection via encryption (protection range user-definable)
- Data frequency = RF/64
- The Memory of the Hitag S transponder depends on the used type of Hitag S transponder:
  - HTS H32 (Hitag S 32) → ID only (32Bit read only)
  - HTS H56 (Hitag S 256) → page 0 to page 7 (256Bit)
  - HTS H48 (Hitag S 2048) → full memory range of the Hitag 1 (2048Bit)

### 5.2 Transponder Recognition

#### 3200: Philips Hitag 1 /Hitag S Transponder Recognition

This function determines, either in Single or in Background mode, if there is a Philips Hitag 1 or Hitag S transponder in reading distance to the antenna or not.

<i>Host to RF system</i>	
<b>STX</b>	"3200" <m>
<b>ETX</b>	{ c }
m	Mode: „S“ = single mode, „B“ = background mode.

  

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"3200" <xx>
<b>ETX</b>	{ c }
x	„00“ = no Hitag transponder in reading distance, „01“ = transponder was recognized.

### 5.3 Transponder Read Functions

#### 4200: Read Philips Hitag 1 / Hitag S Transponder (Block by Block)

Reads a Philips Hitag transponder. With this function the Hitag transponder is only read block by block, i. e. in blocks of 16 bytes each.

Hitag transponders contain memory areas that can only be accessed (reading/writing) via complex encryption routines. The memory areas to be addressed in such a way can be determined in the transponder control registers via status bits.

The encryption mode is not implemented at the moment.

<i>Host to RF system</i>	
<b>STX</b>	"4200" <m> <aa> <ee>
<b>ETX</b>	{c}
m	Read mode: „S“ = single mode, „B“ = background mode.
a	First block to be read, e. g. 08.
e	Last block to be read, e. g. 0A.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4200" <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
d	Read data; the number of transmitted characters depends on the number of blocks to be read. 32 characters are transmitted per block, with 2 characters being combined to one data byte.

#### 4201: Read Philips Hitag 1 / Hitag S Transponder (Page by Page)

Reads a Philips Hitag 1 / Hitag S transponder. With this function the Hitag transponder is only read page by page, i. e. in pages of 4 bytes each.

Hitag transponders contain memory areas that can only be accessed (reading/writing) via complex encryption routines. The memory areas to be addressed in such a way can be determined in the transponder control registers via status bits.

The encryption mode is not implemented at the moment.

<i>Host to RF system</i>	
<b>STX</b>	"4201" <m> <aa> <ee>
<b>ETX</b>	{c}
m	Read mode: „S“ = single mode, „B“ = background mode.
a	First page to be read, e. g. 20.
e	Last page to be read, e. g. 24.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4201" <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
d	Read data; the number of transmitted characters depends on the number of pages to be read. 8 characters are transmitted per page, with 2 characters being combined to one data byte

In Background mode, the number of bytes to be read is *restricted to 16 bytes*.

## 5.4 Transponder Write Functions

### 5200: Write to Philips Hitag 1 / Hitag S Transponder Block by Block

With this function you can write to a Philips Hitag transponder block by block, i. e. by 16 bytes each. Blocks 0 and 1 are configuration blocks, i. e. writing to these blocks can (negatively) influence the transponder's functionality!

Hitag transponders contain memory areas that can only be accessed (reading/writing) via complex encryption routines. The memory areas to be addressed in such a way can be determined in the transponder control registers via status bits.

*The encryption mode is not implemented at the moment.*

Host to RF system	
<b>STX</b>	"5200" <bb> <dddddddddddddddddddd>
<b>ETX</b>	{c}
b	Block number to be written to, e. g. „02“ for 2nd or „07“ for 7th block.
d	Data to be written. For each byte to be written 2 characters must be transmitted.

RF system to Host	
<b>ACK</b>	
<b>STX</b>	"5200"
<b>ETX</b>	{c}
-	No return data.

### 5201: Write to Philips Hitag 1 / Hitag S Transponder Page by Page

With this function you can write to a Philips Hitag transponder page by page, i. e. in blocks of 4 bytes each. Pages 0 to 7 are configuration pages, i. e. writing to these pages can (negatively) influence the transponder's functionality!

Hitag transponders contain memory areas that can only be accessed (reading/writing) via complex encryption routines. The memory areas to be addressed in such a way can be determined in the transponder control registers via status bits.

*The encryption mode is not implemented at the moment.*

Host to RF system	
<b>STX</b>	"5201" <aa> <nn> <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
a	First page to be written to. If you want to write, e. g., from page 34 onwards, „22“ must be transmitted as parameter.
n	Number of pages to be written from start address onwards. You can write to max. 4 pages.

d	Data to be written. The number of characters to be transmitted depends on the number of pages to be written; for each page 8 characters must be transmitted, with 2 characters representing one byte to be written.
---	---

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5201"
<b>ETX</b>	{c}
-	No return data.

## 5.5 Transponder-specific Functions

### 6200: Philips Hitag 1 / Hitag S ID Request

This function determines the ID (read-only page) of a Philips Hitag 1 / Hitag S transponder in reading distance to the antenna.

<i>Host to RF system</i>	
<b>STX</b>	"6200"
<b>ETX</b>	{c}
-	No request parameters.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6200" <iiiiiiii>
<b>ETX</b>	{c}
i	Philips Hitag ID; 8 characters are transmitted, with 2 characters being combined to one byte each.

### 6201: Hitag S Set to 4001 Compatible Mode

This function calculates the parity bits and write them together with the header into page 4 and 5 of the transponder. Then it's been set to 64 bit output with Manchester modulation, data rate 2kBit (rf/64) TTF Mode.

<i>Host to RF system</i>	
<b>STX</b>	"6201" <iiiiiiiiiii>
<b>ETX</b>	{c}
i	5 Bytes ID Without header and parity information. 2 characters being combined to one byte each.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6201"



<b>ETX</b>	{c}
-	No return data.

### 6220: Hitag S Anticollision ID Request

Calling this function, the reader will detect all Hitag S Transponder ion the Field and set them into quiet mode. The number of slots is to optimize the reading speed in dependence of the expected number of tags in the field.

<i>Host to RF system</i>	
<b>STX</b>	"6220i" <ss>
<b>ETX</b>	{c}
ss	Number of slots used for the Anti-Collision Recommend values: 01: If you expect 5 Tags or less in the field at the same time 04: If you expect more than 10 Tags in the field at the same time > 04: Not recommended

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6220" [<iiiiiiii>] [<iiiiiiii>]
<b>ETX</b>	{c}
i	Tag ID; 8 characters are transmitted per Tag, with 2 characters being combined to one byte each.

## 6 EM 4001/2 Sokymat Unique Transponder

### 6.1 Features

- Read-only transponder
- 64 bits
- 40 data bits
- Continuous data output
- Data frequency = RF/64

### 6.2 Transponder Recognition

#### 3300: EM 4001 & 4002 / SID Unique Transponder Recognition

This function determines in Single mode if there is an EM V4002 or SID Unique transponder respectively in reading distance to the antenna or not.

<i>Host to RF system</i>	
<b>STX</b>	"3300" <m>
<b>ETX</b>	{ c }
m	Mode: at the moment only „S“ = single mode.

  

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"3300" <xx>
<b>ETX</b>	{ c }
x	„00“ = no EM 400x transponder in reading distance, „01“ = transponder was recognized.

### 6.3 Transponder Read Functions

#### 4300: Read EM 4001 & 4002 or SID Unique Transponder

Reads an EM 4001 & 4002 or SID Unique transponder. All 64 bits of the transponder are returned, i. e. all data bits including line parity, column parity and header bits. A parity check is performed by the RF system. The 9 header bits are appended to the end of the transmitted data. At the moment, the function can only be used in Single mode.

<i>Host to RF system</i>	
<b>STX</b>	"4300" <m>
<b>ETX</b>	{ c }
m	Read mode: at the moment only „S“ = single mode.

  

<i>RF system to Host</i>	
--------------------------	--

<b>ACK</b>	
<b>STX</b>	"4300" <dddddddddddddd>
<b>ETX</b>	{ c }
d	Read data; 16 characters are returned, i. e. 2 characters per byte.

#### 4301: Read EM 4001 & 4002 or SID Unique Serial Number

Reads an EM 400x or SID Unique serial number. Only the data bits relevant for the serial number are returned, i. e. 5 bytes. The parity check is performed in the RF system. At the moment, the function can only be used in Single mode.

<i>Host to RF system</i>	
<b>STX</b>	"4301" <m>
<b>ETX</b>	{ c }
m	Read mode: at the moment only „S“ = single mode.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4301" <dddddddddd>
<b>ETX</b>	{ c }
d	Read data; 10 characters are returned, i. e. 2 characters per byte read.

## 7 EM V4050 Sokymat Titan Transponder

### 7.1 Features

- Read/write transponder
- 32 blocks à 4 bytes each = 128 bytes with read/write access
- Block 0 is the password block, blocks 1 and 2 are configuration blocks, blocks 32 and 33 are ID blocks (serial number)
- Read and write protection via status bits and password
- Data frequency = RF/64

### 7.2 Transponder Recognition

#### 3400: EM 4050 / SID Titan Transponder Recognition

This function determines, if there is an EM V4050 or SID Titan transponder respectively in reading distance to the antenna or not.

<i>Host to RF system</i>	
<b>STX</b>	"3400" <m>
<b>ETX</b>	{ c }
m	Mode: „S“ = single mode

  

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"3400" <xx>
<b>ETX</b>	{ c }
X	„00“ = no EM 4050 transponder in reading distance, „01“ = transponder was recognized.

### 7.3 Transponder Read Functions

#### 4400: Read EM 4050 or SID Titan

Reads an EM 4050 or SID Titan transponder. With this function the EM 4050 transponder is only read by double words, i. e. in blocks of 4 bytes each. The parity data are evaluated in the RF system and not output.

In case of an incorrect LOGIN password, the RF system returns a read error.

EM 4050 transponders can be partly read protected by a password; in case of a Read request of a protected block without password transmission, an error message is generated! Password mode can be set via both the general Password function „6000“ and the transponder-specific Password function „6402“. If both are in use, the specific password has priority for the V4050 and Titan transponders.

<i>Host to RF system</i>	
<b>STX</b>	"4400" <m> <aa> <ee>
<b>ETX</b>	{c}
m	Read mode: „S“ = single mode
a	First block to be read, e. g. „07“ for 7th block or „0A“ for 10th block.
e	Last block to be read, e. g. „09“ for 9th or „0C“ for 12th block.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4400" <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
d	Read data; the number of transmitted characters depends on the number of blocks to be read. 8 characters are transmitted per block, with 2 characters being combined to one data byte.

## 7.4 Transponder Write Functions

The EM V4050 and SID Titan transponders can be write protected via password or status bits.

### 5400: Write to EM V4050 / SID Titan Transponder Block by Block

With this function you can write to an EM V4050 or SID Titan transponder respectively block by block, i. e. in blocks of 4 bytes each. Note that this function cannot be used to write to the transponder configuration blocks 1 and 2; for this purpose separate requests are provided!

The Write function uses the transponder's Read-after-write capability. If this function has not been switched on at the transponder (by setting the corresponding status bit), it is possible to write to the transponder, the RF system will, however, return an error message.

EM 4050 transponders can be partly write protected via password or status bits; in the case of a Write request of a protected block, an error message is generated! Password mode can be set via both the general Password function „6000“ and the transponder-specific Password function „6402“. If both are in use, the specific password has priority for the V4050 and Titan transponders.

<i>Host to RF system</i>	
<b>STX</b>	"5400" <bb> <dddddddd>
<b>ETX</b>	{c}
b	Block number to be written to. If you want to write to, e. g., block 5, „05“ must be transmitted as parameter.
d	Data to be written. For each block 8 characters must be transmitted, with 2 characters representing one byte to be written

<i>RF system to Host</i>	
<b>ACK</b>	

<b>STX</b>	"5400"
<b>ETX</b>	{c}
-	No return data.

## 7.5 Transponder-specific Functions

### 6400: Write to EM V4050 / SID Titan Protection Word

With this function the protection word of an EM 4050 or SID Titan transponder respectively can be written.

Writing to the protection word can only be performed via password! If no password has been activated, the RF system outputs the default password 00000000h. Password mode can be set via both the general Password function „6000“ and the transponder-specific Password function „6402“. If both are in use, the specific password has priority for the V4050 and Titan transponders.

<i>Host to RF system</i>	
<b>STX</b>	"6400" <aa> <ee> <ff> <11>
<b>ETX</b>	{c}
a	First word (block) which is read-protected via password, e. g. „03“ for word address 03.
e	Last word (block) which is read-protected via password, e. g. „1C“ for word address 28.
f	First word address (block address) which is write-protected, e. g. „09“ for word address 09.
1	Last word address (block address) which is write-protected, e. g. „0A“ for word address 10.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6400"
<b>ETX</b>	{c}
-	No return data.

### 6401: Write to EM V4050 / SID Titan Control Word

With this function the control word of an EM 4050 or SID Titan transponder respectively can be written to.

Writing to the control word can only be performed via password! If no password has been activated, the RF system outputs the default password 00000000h. Password mode can be set via both the general Password function „6000“ and the transponder-specific Password function „6402“. If both are in use, the specific password has priority for the V4050 and Titan transponders.

<i>Host to RF system</i>	
<b>STX</b>	"6401" <aa> <ee> <nnnn>
<b>ETX</b>	{c}

a	First word (block) which is output permanently.
e	Last word (block) which is output permanently.
n	Control word: „0000“ = password check off, no read-after-write. „0100“ = password check on, no read-after-write. „0200“ = password check off, read-after-write. „0300“ = password check on, read-after-write.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6401"
<b>ETX</b>	{c}
–	No return data.

#### 6402: Switch on EM V4050 / SID Password Mode (Specific)

With this function the transponder-specific password of an EM 4050 or SID Titan transponder respectively can be transmitted to the RF system and Password mode can be switched on for the RF system (not for the transponder).

In order to use the password, the corresponding status bits must be set in the transponder control register. For this transponder, the password transmitted via this function has priority over the general password „6000“.

CAUTION: The password transmitted via interface can e. g. be intercepted by a „Line Listener“.

<i>Host to RF system</i>	
<b>STX</b>	"6402" <pppppppp>
<b>ETX</b>	{c}
p	Data. 8 characters must be transmitted, with 2 characters making up one byte to be written.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6402"
<b>ETX</b>	{c}
–	No return data.

#### 6403: Deactivate EM V4050 / SID Titan Transponder Password Mode (Specific)

With this function the transponder-specific password of the V4050 or Titan transponder is switched off. If the general password „6000“ is active, it will be used subsequently.

<i>Host to RF system</i>	
<b>STX</b>	"6403"
<b>ETX</b>	{c}

-	No request parameters.
---	------------------------

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6403"
<b>ETX</b>	{c}
-	No return data.

#### 6404: Write New Password to EM V4050 / SID Titan Transponder (Specific)

To overwrite an existing password you must first have activated Password mode with the current password, otherwise this function returns an error message. To use the password, the corresponding status bits must be set in the transponder control registers. If the new password could be written, the current password is replaced by the new one and used for all subsequent write/read accesses. For this transponder, the password transmitted via this function has priority over the general password „6000“.

CAUTION: The password transmitted via interface can e. g. be intercepted by a „Line Listener“.

<i>Host to RF system</i>	
<b>STX</b>	"6404" <pppppppp>
<b>ETX</b>	{c}
p	Password data. 8 characters must be transmitted, with 2 characters making up one byte to be written.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6404"
<b>ETX</b>	{c}
-	No return data.



## 8 EM 4205/4305 Transponder

### 8.1 Features

- Read/write transponder
- 16 blocks à 4 bytes each = 512 bit with read/write access
- 32 bit UID (Block 1)
- 32 bit Password (Block 2)
- 32 bit Configuration (Block 4)
- Protection Word (Block 14 / Block 15)
- Read and write protection via status bits and password
- Data frequency = RF/64 RF/32
- Modulation: Manchester, Biphase

### 8.2 Transponder Read Functions

#### 4410: Read EM 4205/4305 Transponder

<i>Host to RF system</i>	
<b>STX</b> "4410" <m> <aa> <ee>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Read mode: „S“ = single mode
a	First block to be read, e. g. „07“ for 7th block or „0A“ for 10th block.
e	Last block to be read, e. g. „09“ for 9th or „0C“ for 12th block.
<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "4410" <dd> [ <dd> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
d	Read data; the number of transmitted characters depends on the number of blocks to be read. 8 characters are transmitted per block, with 2 characters being combined to one data byte.

### 8.3 Transponder Write Functions

Attention: The transponders can be write protected via password or status bits.

#### 4410: Write EM 4205/4305 Transponder

<i>Host to RF system</i>	
<b>STX</b> "5410" <bb> <dddddddd>	

<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number to be written to. If you want to write to, e. g., block 5, „05“ must be transmitted as parameter.
d	Data to be written. For each block 8 characters must be transmitted, with 2 characters representing one byte to be written

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "5410"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	No return data.

## 8.4 Transponder-specific Functions

### 6412: EM 4205/4305 Login

<i>Host to RF system</i>	
<b>STX</b> "6412" <11111111>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
l	password

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "6412"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	No return data.

### 6413: EM 4205/4305 Protect Command

<i>Host to RF system</i>	
<b>STX</b> "6413" <pppp>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	protection bits

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "6413"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	No return data.

e.G.: 0080 will set the protection for word 7.

### 6416: Configure Reader for EM 4205/4305

This command is used to configure the reader for autoscan function.. For proper function in autoscan mode the tag type and output format has to be set. See function "F012".

<i>Host to RF system</i>	
<b>STX</b>	"6416" <m> <dd> <ll>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
m	modulation type: 0=manchester 1=biphase
dd	data rate: 40=rf/64 or 20=rf32
ll	last page to read (4 bytes / page) (hex value)
<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6416"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	No return data.

### 6417: Get Reader Configuration for EM 4205/4305

This command is used to configure the reader for autoscan function.. For proper function in autoscan mode the tag type and output format has to be set. See function "F012".

With the function the setting is stored in the non volatile memory of the reader.

<i>Host to RF system</i>	
<b>STX</b>	"6417"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6417" <m> <dd> <ll>
<b>ETX</b>	{c}
<i>Response Data</i>	
m	modulation type: 0=manchester 1=biphase
dd	data rate: 40=rf/64 or 20=rf32
ll	last page to read (hex value)

### 6418: Configure Reader and/or Tag for EM 4205/4305

<i>Host to RF system</i>	
--------------------------	--

<b>STX</b>	"6418" <t> <m> <dd> <ll>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
t	Target: 1= Reader; 2=Transponder; 3=Reader and Transponder
m	modulation type: 0=manchester 1=biphase
dd	data rate: 40=rf/64 or 20=rf32
ll	last page to read (hex value)

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6418"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	No return data.

**With this function the settings are not stored permanent in the reader. So it has to be transmitted after each reset or power on.**

Configuring the Transponder the bits for data rate, modulation type and last default read word (LRW) are set, all other bits in the configuration word has been set to 0.

## 9 Temic e5530 Transponder

### 9.1 Features

- Read-only transponder
- 128 bits
- Continuous data output
- Different data lengths
- Different data frequencies
- Different encoding and modulation types

### 9.2 Transponder Recognition

If no special transponder configuration is transmitted via function „6500“, the following default settings are used:

- Memory: 8 bytes (2 blocks)
- Header: E6 (8 bits)
- Data frequency: RF/64
- Encoding: bi-phase

#### 3500: Temic e5530 Transponder Recognition

This function determines in Single mode if there is a Temic e5530 transponder in reading distance to the antenna or not.

<i>Host to RF system</i>	
<b>STX</b>	"3500" <m>
<b>ETX</b>	{ c }
m	Mode: at the moment only „S“ = single mode.
<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"3500" <xx>
<b>ETX</b>	{ c }
x	„00“ = no Temic e5530 transponder in reading distance, „01“ = transponder was recognized.

### 9.3 Transponder Read Functions

If no special transponder configuration is transmitted via function „6500“, the following default settings are used:

- Memory: 8 bytes (2 blocks)
- Header: E6 (8 bits)

- Data frequency: RF/64
- Encoding: bi-phase

#### 4500: Read Temic e5530 Transponder

Reads a Temic e5530 transponder. Note that the RF system must know the transponder's current header size, the header itself, and the number of released blocks (of 4 bytes each). It must also know the bit rate and the encoding type of the transponder to be read. The RF system's setting is performed on the basis of the corresponding configuration functions. All requested bytes are returned, starting with the first byte behind the header. If the number of requested bytes is equal to or higher than the number of bytes released in the transponder (= blocks \* 4), the header is returned, too. At the moment, the function can only be used in Single mode.

<i>Host to RF system</i>	
<b>STX</b>	"4500" <m> <nn>
<b>ETX</b>	{ c }
m	Read mode: at the moment only „S“ = single mode.
n	Number of bytes to be read, e. g. „01“ for 1 byte or „02“ for 2 bytes, etc.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4500" <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{ c }
d	Read data; the number of transmitted characters depends on the number of blocks to be read. 8 characters are transmitted per block, with 2 characters being combined to one data byte

## 9.4 Transponder-specific Functions

#### 6500: Temic e5530 Reader Configuration

In order to be able to read the Temic e5530 transponder, the RF system must know the transponder configuration. This includes memory size, bit rate, and encoding of the transponder data. If this function is not called up, the RF system uses the following default settings:

- Memory size: 8 bytes (2 blocks)
- Data frequency: RF/64
- Encoding: bi-phase

<i>Host to RF system</i>	
<b>STX</b>	"6500" <kkkkkkkk>
<b>ETX</b>	{ c }
k	Configuration; the configuration consists of 32 bits, i. e. of 4 bytes, i. e. 8 characters are to be transmitted to the reader.

The following settings are possible:

Bit 20	Bit 19	Bit 18	Bit rate
0	1	0	RF/32
0	1	1	RF/40
1	0	0	RF/50
1	0	1	RF/64

Bit 16	Bit 15	Encoding
0	1	Manchester
1	0	Bi-phase

No other bits are evaluated at the moment!

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6500"
<b>ETX</b>	{c}
-	No return data.

### 6501: Temic e5530 Header

In order to be able to read the Temic e5530 transponder, the RF system must know the header size, the header itself, and the number of blocks released in the transponder. The following function transmits these data to the RF system:

<i>Host to RF system</i>	
<b>STX</b>	"6501" <gg> <hhhh> <nn>
<b>ETX</b>	{c}
g	Header size; „08“ = 8-bit header, „10“ = 16-bit header.
h	Header consisting of 4 characters.
n	Number of blocks released in the transponder.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6501"
<b>ETX</b>	{c}
-	No return data.

If this function is not called up, the RF system uses the 8-bit default header E6 and assumes 2 released blocks (8 bytes).

## 10 Temic e555x / Q5 Transponder

### 10.1 Features

- Read/write transponder
- 8 blocks à 4 bytes = 32 bytes
- Block 0 is the configuration block, block 7 is the password block
- Write-protection via status bits and password
- Continuous data output (range of data user-definable)
- Different data lengths
- Different data frequencies
- Different encoding and modulation types

Note: in some cases the problem may occur that you cannot read a Temic transponder because you simply don't know its current configuration. However, since data transmission from reader to tag is fixed, the tag will understand a reconfiguration command, no matter how it has previously been configured. Tests have shown that the most safe way to put a Temic tag into a known state is the following:

- Write some dummy data to a user page (e. g. page #1)
- Send the transponder configuration command, using the desired settings
- Send the reader configuration command, using the same settings as above

Afterwards, both the reader and the tag have been put into the new configuration. You may now perform any read/write operations.

### 10.2 Transponder Recognition

If no special transponder configuration is transmitted via function „6600“, the following default settings are used:

- Data frequency: RF/64
- Terminators: sequential terminators

#### 3600: Temic e5550 Transponder Recognition

This function determines in Single mode if there is a Temic e5550 transponder in reading distance to the antenna or not. This works for e5550 only!

<i>Host to RF system</i>	
<b>STX</b>	"3600" <m>
<b>ETX</b>	{c}
m	Mode: at the moment only „S“ = single mode.

<i>RF system to Host</i>
--------------------------



<b>ACK</b>	
<b>STX</b>	"3600" <xx>
<b>ETX</b>	{c}
x	„00“ = no Temic e5550 transponder in reading distance, „01“ = transponder was recognized.

### 10.3 Transponder Read Functions

If no special transponder configuration is transmitted via function „6600“, the following default settings are used:

- Data frequency: RF/64
- Encoding: Manchester
- Terminators: sequential terminators

At the moment, the Read function only supports Manchester-type encoding.

#### 4600: Read Temic e5550 Transponder

Reads a Temic e5550 transponder. Note that the RF system must know the transponder configuration. The RF system's setting is performed on the basis of the corresponding configuration functions. The requested number of blocks is returned, starting with the first block released on the transponder. In case more blocks are requested than are released on the transponder to be read, the RF system returns the error message „No TAG in field“. At the moment, the function can only be used in Single mode.

<i>Host to RF system</i>	
<b>STX</b>	"4600" <m> <nn>
<b>ETX</b>	{c}
m	Read mode: at the moment only „S“ = single mode.
n	Number of blocks to be read, e. g. „01“ for 1 block or „02“ for 2 blocks, etc.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4600" <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
d	Read data; the number of transmitted characters depends on the number of blocks to be read. 8 characters are transmitted per block, with 2 characters being combined to one data byte.

#### 4601: Read Temic Q5 Transponder via Direct Access

Reads a Temic e5551 or Q5 (a. k. a. e5555) transponder using Direct Access. Note that the RF system must know the transponder configuration. The RF system's setting is performed on the basis of the corresponding configuration functions. The requested number of blocks is returned, starting

with the first block released on the transponder. At the moment, the function can only be used in Single mode. Note for e5551: password functionality is not supported by the transponder.

<i>Host to RF system</i>	
<b>STX</b>	"4601" <m> <ff> <11>
<b>ETX</b>	{c}
m	Read mode: at the moment only „S“ = single mode.
f	First block to be read. Possible values are 0..7, others will cause an error.
l	Last block to be read. Possible values are 0..7, others will cause an error.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4601" <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
d	Read data; the number of transmitted characters depends on the number of blocks to be read. 8 characters are transmitted per block, with 2 characters being combined to one data byte.

## 10.4 Transponder Write Functions

### 5600: Write to Temic e5550 Transponder without Lock Bit

With this function you can write to a Temic e5550 transponder block by block (with 4 bytes per block). The lock bit is not set! Blocks 1 through 7 can be used. The configuration block can only be written to with a special configuration instruction.

If the transponder is in “write protected mode“, an error message is generated unless the correct password has been transmitted! To activate Password mode you can use the general Password function “6000”.

<i>Host to RF system</i>	
<b>STX</b>	"5600" <bb> <ddddddddd>
<b>ETX</b>	{c}
b	Block number, e. g. „02“ if block 2 is to be written to, or „05“ for block 5.
d	Data to be written. 8 characters are transmitted, with 2 characters representing one byte.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5600"
<b>ETX</b>	{c}
-	No return data.

### 5601: Write to Temic e5550 Transponder with Lock Bit

With this function you can write to a Temic e5550 transponder block by block (with 4 bytes per block), just as with function "5600". In addition, the lock bit of the block to be written to is set here, so that it cannot be changed subsequently. Blocks 1 to 7 can be used. Configuration block 0 can be written to with a special format instruction.

If the transponder is in "write protected mode", an error message is generated, unless the correct password has been transmitted! To activate Password mode you can use the general Password function "6000".

<i>Host to RF system</i>	
<b>STX</b>	"5601" <bb> <dddddddd>
<b>ETX</b>	{c}
b	Block number, e. g. „02“ if block 2 is to be written to, or „05“ for block 5.
d	Data to be written. 8 characters are transmitted, with 2 characters representing one byte to be written.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5601"
<b>ETX</b>	{c}
-	No return data.

### 5602: Write to Temic Q5 Transponder without Lock Bit

With this function you can write to a Temic Q5 transponder block by block (with 4 bytes per block). Blocks 1 through 7 can be used. The configuration block can only be written to with a special configuration instruction.

<i>Host to RF system</i>	
<b>STX</b>	"5602" <bb> <dddddddd>
<b>ETX</b>	{c}
b	Block number, e. g. „02“ if block 2 is to be written to, or „05“ for block 5.
d	Data to be written. 8 characters are transmitted, with 2 characters representing one byte.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5602"
<b>ETX</b>	{c}
-	No return data.

### 5603: Write to Temic Q5 Transponder with Lock Bit

With this function you can write to a Temic Q5 transponder block by block (with 4 bytes per block). In addition, the lock bit of the block to be written to is set here, so that it cannot be changed subsequently. Blocks 1 through 7 can be used. The configuration block can only be written to with a

special configuration instruction.

<i>Host to RF system</i>	
<b>STX</b>	"5603" <bb> <ddddddddd>
<b>ETX</b>	{c}
b	Block number, e. g. „02“ if block 2 is to be written to, or „05“ for block 5.
d	Data to be written. 8 characters are transmitted, with 2 characters representing one byte.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5603"
<b>ETX</b>	{c}
-	No return data.

## 10.5 Transponder-specific Functions

### 6600: Temic e5550 Reader Configuration

In order to be able to read the Temic e5550 transponder, the RF system must know the transponder configuration. If this function is not called up, the RF system uses the following default settings:

- Data frequency: RF/64
- Encoding: Manchester
- Terminators: sequential terminators

<i>Host to RF system</i>	
<b>STX</b>	"6600" <kkkkkkkkk>
<b>ETX</b>	{c}
k	Configuration; the configuration consists of 32 bits, i. e. of 4 bytes, i. e. 8 characters are to be transmitted to the reader, e. g. "00088008" for RF/32 with Manchester encoding and sequential terminators.

The following settings are possible:

<i>Bit3</i>	<i>Bit2</i>	<i>Terminator mode</i>
0	0	no terminator (e5530 mode)
0	1	only block terminators
1	0	only sequential terminators
1	1	sequential and block terminators

<i>Bit16</i>	<i>Bit15</i>	<i>Encoding</i>
0	1	Manchester
1	0	Biphase (not yet implemented)

<i>Bit 20</i>	<i>Bit 19</i>	<i>Bit 18</i>	<i>Bit rate</i>
---------------	---------------	---------------	-----------------

0	1	0	RF/32
0	1	1	RF/40
1	0	0	RF/50
1	0	1	RF/64

No other bits are evaluated at the moment!

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6600"
<b>ETX</b>	{c}
-	No return data.

### 6601: Configure Temic e5550 Transponder

With this function you can configure the Temic e5550 transponder (write to block 0). The bit values should be checked in the e5550 transponder specification. You may also define configurations that cannot be processed by the RF system itself, as, e. g., FSK modulation. In addition you may protect block 0 from being re-written by using the lock bit.

If the transponder is in "write protected mode", an error message is generated unless the correct password has been transmitted! To activate Password mode you can use the general Password function "6000".

**Note:** When changing the password flag, it is necessary to proceed with greatest care, since it is very easy to accidentally destroy the tag! Namely, a tag with deactivated password flag must never be addressed by a reader which is in password mode; otherwise the command will be misunderstood by the tag and cause arbitrary data to be written - even if only a read command has been submitted.

<i>Host to RF system</i>	
<b>STX</b>	"6601" <11> <kkkkkkkk>
<b>ETX</b>	{c}
l	Lock-bit flag: „00“ = do not set lock bit, „01“ = set lock bit.
k	Configuration; the configuration consists of 32 bits, i. e. of 4 bytes, i. e. 8 characters are to be transmitted to the RF system, e. g. "000880E8" for RF/32 with Manchester encoding and sequential terminators. All blocks can be transmitted, cause maxblock is set to 7.

<i>Bit3</i>	<i>Bit2</i>	<i>Terminator mode</i>
0	0	no terminator (e5530 mode)
0	1	only block terminators
1	0	only sequential terminators
1	1	sequential and block terminators

<i>Bit 7</i>	<i>Bit 6</i>	<i>Bit 5</i>	<i>MaxBlocks</i>
--------------	--------------	--------------	------------------

0	0	0	0
0	0	1	1
0	1	0	1 to 2
0	1	1	1 to 3
1	0	0	1 to 4
1	0	1	1 to 5
1	1	0	1 to 6
1	1	1	1 to 7

Bit16	Bit15	Encoding
0	1	Manchester
1	0	Biphase (not yet implemented)

Bit 20	Bit 19	Bit 18	Bit rate
0	1	0	RF/32
0	1	1	RF/40
1	0	0	RF/50
1	0	1	RF/64

RF system to Host	
<b>ACK</b>	
<b>STX</b>	"6601"
<b>ETX</b>	{c}
-	No return data.

#### Example:

The following example shows how to configure Tag and Reader in RF/32 mode:

#### Transponder configuration

Host to RF system : **STX** "660100000880E8" **ETX** {c}

RF system to Host : **SYN STX** "660110" **ETX** {c}

#### Configuration Bits:

```

00000000000 010 0 01 0000000 111 0 10 00
               |||      ||      |||      +-+-- Terminator mode
               |||      ||      |||      (sequential)
               |||      ||      |||      Max Block = 7
               |||      +-+----- Manchester Encoding
               +-+----- Bitrate RF/32
  
```

corresponding hex value: 000880E8

#### Reader configuration

Host to RF system : **STX** "660000088008" **ETX** {c}

RF system to Host : **ACK STX** "6600" **ETX** {c}

```

000000000000 010 0 01 000000000000 10 00
          |||      ||
          |||      ||
          |||      ||
          |||      ||
          |||      ++----- Manchester Encoding
          +++----- Bitrate RF/32
                                +-+--- Terminator mode
                                (sequential)

```

#### Read Transponder in RF/32 mode

Host to RF system : **STX** "4600S01" **ETX** {c}

RF system to Host : **ACK** **STX** "4600<ddddddd>" **ETX** {c}

### 6602: Temic Q5 Reader Configuration

In order to be able to read the Temic Q5 transponder, the RF system must know the transponder configuration.

<i>Host to RF system</i>	
<b>STX</b> "6602" <kkkkkkkk>	
<b>ETX</b> {c}	
k	Configuration; the configuration consists of 32 bits, i. e. of 4 bytes, i. e. 8 characters are to be transmitted to the reader. The following configurations are supported to change the bit rate: 0000F000 = RF/32 00013000 = RF/40 00018000 = RF/50 0001F000 = RF/64

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "6602"	
<b>ETX</b> {c}	
-	No return data.

### 6603: Configure Temic Q5 Transponder

With this function you can configure the Temic Q5 transponder (write to block 0). The bit values should be checked in the Q5 transponder specification.

If the transponder is in "password mode", data cannot be read or written unless the correct password has been transmitted! To activate Password mode you can use the general Password function "6000".

**Note 1:** When changing the password flag, it is necessary to proceed with greatest care, since it is very easy to accidentally destroy the tag! Namely, a tag with deactivated password flag must never be addressed by a reader which is in password mode; otherwise the command will be

misunderstood by the tag and cause arbitrary data to be written - even if only a read command has been submitted.

**Note 2:** The configuration page is not automatically verified, due to the problems mentioned above.

<i>Host to RF system</i>	
<b>STX</b>	"6603" <11> <kkkkkkkkk>
<b>ETX</b>	{c}
l	Lock-bit flag: „00“ = do not set lock bit, „01“ = set lock bit.
k	Configuration; the configuration consists of 32 bits, i. e. of 4 bytes, i. e. 8 characters are to be transmitted to the RF system. Examples: 0001F004 = (RF/64, maxblock = 2) 0001300E = (RF/40, maxblock = 7)

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6603"
<b>ETX</b>	{c}
-	No return data.

#### 6604: Q5 Transponder Set to 4001 Compatible Mode

This function calculates the parity bits and write them together with the header into page 1 and 2 of the transponder. Then it's been set to 64 bit output with Manchester modulation, data rate 2kBit (rf/64) TTF Mode.

<i>Host to RF system</i>	
<b>STX</b>	"6604" <iiiiiiiiiii>
<b>ETX</b>	{c}
i	5 Bytes ID Without header and parity information. 2 characters being combined to one byte each.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6604"
<b>ETX</b>	{c}
-	No return data.



## 11 Sokymat NOVA Transponder

### 11.1 Features

- Read/write transponder
- 10 blocks à 2 bytes = 20 bytes
- Block 8 is the configuration block, block 9 is the password block
- Write-protection via status bits and password
- Continuous data output
- Different data lengths
- Different data frequencies
- Different encoding and modulation types

### 11.2 Transponder Recognition

#### 3000: General Transponder Recognition

The NOVA transponder has **not** been implemented yet in general transponder recognition. Once it is implemented, it will get code number „07“.

#### 3700: NOVA Transponder Recognition

This function determines in Single mode if there is a NOVA transponder within the field or not. It requires the correct bit rate setting for NOVA transponders (see function “6702“).

<i>Host to RF system</i>	
<b>STX</b>	"3700" <m>
<b>ETX</b>	{ c }
m	Mode: at the moment only „S“ or „s“ = single mode.
<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"3700" <xx>
<b>ETX</b>	{ c }
x	“00“ = no NOVA transponder in reading distance or reader not configured for NOVA transponder bit rate; „01“ = NOVA transponder in reading distance.

### 11.3 Transponder Read Functions

#### 4000: Read Transponder (All Types)

At the moment, the NOVA transponder **cannot** be read via the general Read function.

#### 4700: Read NOVA Transponder

With this function, a NOVA transponder can be read. Note that before the read attempt, the reader must be set to the NOVA transponder configuration (see function „6702“). A NOVA transponder without termination impulses (PMC or LIW) is read once a header has been previously defined via function „6702“; otherwise, the system always searches for an LIW or PMC. In some cases, the transponder's data-encoding type can be determined automatically; otherwise, the reader settings are taken over. At the moment, the Background function is not supported.

The result is at least 8, maximally 16 data bytes long.

<i>Host to RF system</i>	
<b>STX</b>	"4700" <m>
<b>ETX</b>	{c}
m	Mode: at the moment only „S“ or „s“ = single mode.

  

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4700" <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
d	Received data in the form of a HEX character string, with two HEX characters having to be combined to one data byte. Example: „40“ = 40h = 64. Min. 16 and max. 32 characters are transmitted (corresponding to 8..16 data bytes).

#### 4701: Read NOVA Transponder in EM4001 Mode

With this function, NOVA transponders can be read in EM4001 or UNIQUE mode respectively. Except for the bit mirroring of the read data (LSB instead of MSB and mirrored bit sequence) this function is identical to function „4700“.

<i>Host to RF system</i>	
<b>STX</b>	"4701" <m>
<b>ETX</b>	{c}
m	Mode: at the moment only „S“ or „s“ = single mode.

  

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4701" <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
d	Received data in the form of a HEX character string, with two HEX characters having to be combined to one data byte. Example: „40“ = 40h = 64. Min. 16 and max. 32 characters are transmitted (corresponding to 8..16 data bytes).

## 11.4 Transponder Write Functions

### 5000: Write to Transponder (All Types)

At the moment, the NOVA transponder **cannot** be written to with the general Write function.

### 5700: Write to NOVA Transponder Block by Block

With this function you can write to a NOVA transponder block by block. Writing to the configuration block is not possible with this function. Note that for writing to the transponder, the reader must have been previously set to the respective NOVA transponder bit rate (see function "6702").

Writing to the NOVA transponder with password is not implemented at the moment.

<i>Host to RF system</i>	
<b>STX</b>	"5700" <bb> <dddd>
<b>ETX</b>	{c}
b	Block number to be written to, „00“ for block 0 to „07“ for block 7.
d	Data to be written, as HEX character string. 1 word is written per block.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5700"
<b>ETX</b>	{c}
-	No return data.

For reasons of performance, only the positive acknowledgement of the transponder is evaluated; it may be sensible to read the transponder's contents after the writing procedure and compare them with the written data.

### 5701: Write to NOVA Transponder Block by Block in EM4001 Mode

With this function it is possible to write to a NOVA transponder block by block in the EM4001 mode, i. e. the data are written to the transponder in „bit-mirrored“ form. Afterwards, the data can be correctly read via function „4701“. The function allows for directly copying the data obtained from an EM4001 or SID UNIQUE to a NOVA transponder without conversion. Otherwise, the function is identical to function „5700“.

Writing to the NOVA transponder with password is not implemented at the moment.

<i>Host to RF system</i>	
<b>STX</b>	"5701" <bb> <dddd>
<b>ETX</b>	{c}
b	Block number to be written to, „00“ for block 0 to „07“ for block 7.

d	Data to be written, as HEX character string. 1 word is written per block.
---	---

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5701"
<b>ETX</b>	{c}
-	No return data.

For reasons of performance, only the positive acknowledgement of the transponder is evaluated; it may be sensible to read the transponder's contents after the writing procedure and compare them with the written data.

## 11.5 Transponder-specific Functions

### 6700: Configure NOVA Transponder

With this function the configuration word of a NOVA transponder can be written to. The value 0000h is assumed as default password.

<i>Host to RF system</i>	
<b>STX</b>	"6700" <ffff>
<b>ETX</b>	{c}
f	Configuration word as HEX character string. This way, the NOVA transponder is, e. g., set to RF/32, Manchester encoding, no terminator, and 128 bits' memory size via configuration „A400h“.

Possible configurations:

<i>Bit 0 to bit 8</i>	<i>Write-protection for blocks 0 to 8</i>
0	Write-protection for corresponding block is off
1	Write-protection for corresponding block is on

<i>Bit 9</i>	<i>Password mode</i>
0	without password check
1	with password check

<i>Bit 10</i>	<i>Encoding</i>
0	bi-phase
1	Manchester

<i>Bit 11</i>	<i>Memory sizes</i>
0	128 bits
1	64 bits

<i>Bit 13</i>	<i>Bit 12</i>	<i>Bit rate</i>
---------------	---------------	-----------------

0	0	RF/64
0	1	RF/40
1	0	RF/32

Bit 15	Bit 14	Terminator
0	0	with LIW
0	1	with PMC
1	0	none

RF system to Host	
<b>ACK</b>	
<b>STX</b>	"6700"
<b>ETX</b>	{c}
-	No return data.

## 6702: Configure Reader for NOVA Transponder

With this function the transponder reader can be configured for NOVA transponders. Here, all settings concerning bit rate, encoding, and, if necessary, header are made.

Host to RF system	
<b>STX</b>	"6702" <bb> <x> <hhhh> <ss>
<b>ETX</b>	{c}
b	Bit rate to be set. Permitted values are „00“ for automatic bit-rate recognition, „20“ for RF/32, „28“ for RF/40, and „40“ for RF/64.
x	Modulation type. Permitted values are „X“ or „x“ for automatic modulation-type determination <sup>1</sup> , „B“ or „b“ for bi-phase, and „M“ or „m“ for Manchester modulation.
h	Header for not-terminated transponders (i. e. without LIW or PMC).
s	Bits relevant for the header. If a value > 0 is entered, the reader attempts to always read a transponder with header. In this case, terminated transponders cannot be read any more.

Example for an EM4001-equivalent reader configuration (header consisting of 9 one-bits):

**STX** "670240M01FF09" **ETX** <c>

RF system to Host	
<b>ACK</b>	
<b>STX</b>	"6702"
<b>ETX</b>	{c}
-	No return data.

<sup>1</sup> Automatic modulation-type recognition always presupposes Manchester modulation; if the transponder is encoded in bi-phase mode, the returned read data are invalid. Automatic modulation-type recognition does not work for transponders without terminator, i. e. without LIW or PMC.

### 6703: Request Reader Configuration for NOVA

With this function the reader settings for the NOVA transponder can be requested. The response is in the same syntax as for configuration "6702".

<i>Host to RF system</i>	
<b>STX</b>	"6703"
<b>ETX</b>	{c}
-	No request parameters.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "6703" <bb> <x> <hhhh> <ss>	
<b>ETX</b> {c}	
b	Bit rate to be set. Permitted values are „00“ for automatic bit-rate recognition, „20“ for RF/32, „28“ for RF/40, and „40“ for RF/64.
x	Modulation type. Permitted values are „X“ or „x“ for automatic modulation-type determination, „B“ or „b“ for bi-phase, and „M“ or „m“ for Manchester modulation.
h	Header for not-terminated transponders (i. e. without LIW or PMC).
s	Bits relevant for the header. If a value > 0 is entered, the reader attempts to always read a transponder with header. In this case, terminated transponders cannot be read any more.

## 12 Philips PCF-7936 / Hitag 2 Transponder

### 12.1 Features

- Read/write transponder
- 8 pages á 4 bytes = 32 bytes
- Page 0 is the serial number page, pages 1 and 2 are password pages, page 3 is a password and configuration page
- Continuous data output (mode dependent)
- Read and write protection by status bits, password and encryption
- Data frequency = RF/64 and RF/32 (mode depending)
- Different encoding types (mode depending)
- Emulation of EM4001/2 and Philips PCF7930

At the moment, only the Password Mode and the manchester encoding is supported!

### 12.2 Transponder Recognition

The Philips PCF7936/Hitag2 transponder can only be recognized if its configuration is known and the passwords and keys of the RF system and the Transponder do correspond. If no separate transponder configuration is transmitted via function "6802", the following standard settings are used:

- Hitag2-mode
- Password-mode with PWD = 4D494B52h

#### 3000: General Transponder Recognition

The PCF7936/Hitag2-transponder has **not** been implemented yet in general transponder recognition. Once it is implemented, it will get code number „08“.

#### 3800: Philips PCF-7936/Hitag2 - Transponder Recognition

This function determines in single mode if a Philips PCF7936/Hitag2 can be located in the reading distance of the antenna or not.

<i>Host to RF system</i>	
<b>STX</b>	"3800" <m>
<b>ETX</b>	{c}
m	Mode: at the moment only „S“ or „s“= single mode.
<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"3800" <xx>

<b>ETX</b>	{ c }
x	„00“= no PCF7936/Hitag2 transponder in reading distance, „01“=Transponder was recognized.

## 12.3 Transponder Read Functions

The Philips PCF7936/Hitag2 transponder can only be read if its configuration is known and the passwords and keys of the RF system and the Transponder do correspond. If no separate transponder configuration is transmitted via function “6802“, the following standard settings are used:

- Hitag2 mode
- Password mode with PWD = 4D494B52h

### 4000: Read Transponder (All Types)

At the moment, the PCF7936/Hitag2-transponder **cannot** be read via the general Read function.

### 4800: Read Philips PCF7936/Hitag2 Transponder

At the moment only the single mode is supported.

<i>Host to RF system</i>	
<b>STX</b>	"4800" <m> <nn>
<b>ETX</b>	{ c }
m	Read mode : at the moment only „S“ or „s“= single mode.
n	Number of the page to be read, e. g. „01“ for page 1, „05“ for page 5 etc.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4800" <dddddddd>
<b>ETX</b>	{ c }
d	Data read; 8 characters are transmitted per page, with 2 characters being combined to one data byte.

## 12.4 Transponder Write Functions

The Philips PCF7936/Hitag2-Transponder can only be written if its configuration is known and the passwords and keys of the RF system and the Transponder do correspond. If no separate transponder configuration is transmitted via function “6802“, the following standard settings are used:

- Hitag2 mode
- Password mode with PWD = 4D494B52h



## 5000: Write to Transponder (All Types)

At the moment, the PCF7936/Hitag2 transponder **cannot** be written to with the general Write function.

## 5800: Write Philips PCF7936/Hitag2 Transponder

With this function a Philips PCF7936/Hitag2 transponder can be written page by page, (i. e. 4 bytes). Page 3 is the configuration page i. e. writing to these page can (negatively) influence the transponder's functionality!

<i>Host to RF system</i>	
<b>STX</b>	"5800" <pp> <dddddddd>
<b>ETX</b>	{c}
p	Page number, e. g. „02“ for 2nd or „07“ for 7th page.
d	Data to be written. 2 characters per byte have to be transmitted.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"5800"
<b>ETX</b>	{c}
-	No return data.

To be sure that the page was successfully written, command "4800" should be used for a "read-after-write"!

## 12.5 Transponder specific Functions

### 6800: Philips PCF7936/Hitag2 ID (serial number) Identification

With this function the ID (Page 0) of a Philips PCF7936/Hitag2 can be determined. The ID can be determined in every transponder mode. First the RF system has to be set to Hitag2 mode using the "6802" command (pre-setting).

<i>Host to RF system</i>	
<b>STX</b>	"6800"
<b>ETX</b>	{c}
-	No request parameters.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6800" <iiiiiiii>
<b>ETX</b>	{c}
i	ID of transponder (Page 0).

### 6801: Philips PCF7936/Hitag2 - Password Transmission

With this function the password of the Philips PCF7936/Hitag2 can be transmitted to the RF system. The password is required for the password mode of the PCF7936/Hitag2 transponder (see command "6802").

<i>Host to RF system</i>	
<b>STX</b>	"6801" <pppppppp>
<b>ETX</b>	{c}
p	Password.

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6801"
<b>ETX</b>	{c}
-	No return data.

### 6802: Set Mode of Philips PCF7936/Hitag2

With this function the mode of PCF7936/Hitag2 can be transmitted to the RF system. Please note that writing to the PCF7936/Hitag2 in all other modes but the Hitag2 mode only works if the Hitag2 mode has been switched off in the RF system before (by using command "6802").

<i>Host to RF system</i>	
<b>STX</b>	"6802" <mm>
<b>ETX</b>	{c}
m	Mode byte.

Possible mode settings are as follows:

<i>Bit 0</i>	<i>Description</i>
Always 0	Reserved

<i>Bit 2</i>	<i>Bit 1</i>	<i>Description</i>
0	0	Public mode B (EM4001/2)
0	1	Public mode A
1	0	Public mode C (PCF7930)
1	1	Hitag2 mode

<i>Bit 3</i>	<i>Description</i>
0	Password mode
1	Crypto mode

<i>Bit 4</i>	<i>Description</i>
0	No password check in crypto mode
1	Crypto mode with password check

Example for Hitag2-mode + password-mode:

**STX** "6802" "06" **ETX** {c}

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "6802"	
<b>ETX</b> {c}	
-	No return data.

### 6803: Set Crypto-Keys and Password-TAG

With this function the keys and the password-TAG of the Philips PCF7936/ PCF7937 **crypto mode** are transmitted to the RF-System. Before that it should be verified with command „6804“, if a crypto processor is connected to the RF-System. This function only changes the keys and the password which is used for the crypto mode. It does not have any effect on the password stored in the Reader, which is used for the Password mode.

The crypto processor is delivered with the following setting:

**FFFFh (key high), FFFFFFFFh (key low), FFFFFFFh (password-TAG)**

<i>Host to RF system</i>	
<b>STX</b> "6803"<hhhh> <aaaaaaaa> <pppppp> <nnnn> <gggggggg> <dddddd>	
<b>ETX</b> {c}	
h	old key-high (16Bit Key = 2Byte)
a	old key-low (32Bit Key = 4Byte)
p	old password (24Bit = 3Byte)
n	new key-high (16Bit Key = 2Byte)
g	new key-low (32Bit Key = 4Byte)
d	new password (24Bit = 3Byte)

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "6803"	
<b>ETX</b> {c}	
-	No return data.

### 6804: Crypto Processor Version Request

With this function the version of the installed crypto processor can be read out. Also the crypto processor itself and the data exchange to the crypto processor can be checked with this function. If there is no crypto processor connected to the RF-system an error message is returned.

<i>Host to RF system</i>	
<b>STX</b> "6804"	
<b>ETX</b> {c}	

-	none
---	------

### 68FF: Hitag Family Generic Custom Command

This command can be used to send custom commands to any Hitag 2, Hitag 3, Hitag AES or Hitag Pro Transponder.

<i>Host to RF System Request</i>	
<b>STX</b>	"68FF" <tt> <rr> <dd...>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
t	Length of Command in bits.
r	Length of Response in bits
d	Command bits. Most significant bit is sent first. Incomplete Bytes has to be padded with zero bits at the end.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"68FF" <rr...>
<b>ETX</b>	{c}
<i>Response Data</i>	
r	Tag reponse.

## 13 Philips PCF-7937 / Hitag 2 extended Transponder

The Transponder PCF7937 is a derivative of the PCF7936 (Hitag2) and is downwardly compatible with it. To transfer the password and key, the functions for Hitag2 are used.

### 13.1 Features

- Read/write transponder
- downwardly compatible with PCF7936 (Hitag2)
- Up to 448 Byte User EEPROM size

### 13.2 Transponder Read Functions

#### 4830: Read

With this function, the Transponder memory as well as the extended Memory (XM) of the PCF7937 Transponder can be read out. The Transponder can only be read, when the password / crypto keys (depending on the used mode) match those in the Transponder.

The Reader must also use the same mode as the Transponder, otherwise the reading will not be successful.

<i>Host to RF system</i>	
<b>STX</b>	"4830"<m> <ss> <bb> <pp> <aa> <cc>
<b>ETX</b>	{ c }
m	Read mode : at the moment only „S“ or „s“= single mode.
s	Number of the segment where the page is in
b	Number of the block inside the segment
p	Number of the page to be read, e. g. „01“ for page 1, „05“ for page 5 etc.
a	Perform Authentication “00” = skip Authentication “01” = perform Authentication
c	Carrier status after command “00” = switch off carrier after this command “01” = leave carrier switched on after the command

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4830" <dddddddd>
<b>ETX</b>	{ c }
d	Data read; 8 characters are transmitted per page, with 2 characters being combined to one data byte.

## 13.3 Transponder Write Functions

### 5830: Write

With this function, the Transponder memory as well as the extended Memory (XM) of the PCF7937 Transponder can be written to. The Transponder can only be written, when the password / crypto keys (depending on the used mode) match those in the Transponder.

**Attention:** Since the PCF7937 does not support a verification of the written data, it is strongly recommended to verify the data using a read command after every write.

<i>Host to RF system</i>	
<b>STX</b> "5830" <ss> <bb> <pp> <aa> <cc> <dddddddd>	
<b>ETX</b> {c}	
s	Number of the segment where the page is in
b	Number of the block inside the segment
p	Number of the page to be read, e. g. „01“ for page 1, „05“ for page 5 etc.
a	Perform Authentification “00” = skip authentification “01” = perform Authentification
c	Carrier status after command “00” = switch off carrier after this command “01” = leave carrier switched on after the command
d	Data to be written to the Transponder

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "5830"	
<b>ETX</b> {c}	
	No parameters

## 13.4 Transponder specific Functions

### 6830: Read XMA-configuration

With this function, the Configuration of the extended Memory can read.

<i>Host to RF system</i>	
<b>STX</b> "6830" <ss>	
<b>ETX</b> {c}	
s	segment

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b> "6830" <mm> <ss>	
<b>ETX</b> {c}	

m	Configuration
s	size

### 6831: Write XMA-configuration

With this function, the Configuration of the extended Memory can be changed.

<i>Host to RF system</i>	
<b>STX</b>	"6831" <ss> <m> <dd>
<b>ETX</b>	{c}
s	segment
m	Mode (m=mode / s=size)
d	Data to be written

### 6832: Read multiple XMA-configuration

With this function, the Configuration of the extended Memory can be read.

<i>Host to RF system</i>	
<b>STX</b>	"6832" <ss> <ll>
<b>ETX</b>	{c}
f	First segment
l	Last segment

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"6832" <mm> <ss> [<mm> <ss>]
<b>ETX</b>	{c}
m	Configuration
s	size

### 6833: Write multiple XMA-configuration

With this function, the Configuration of the extended Memory can be changed.

<i>Host to RF system</i>	
<b>STX</b>	"6833" <ff> <ll> <m> <dd> [<dd>]
<b>ETX</b>	{c}
f	First segment to write
l	Last segment to write
m	Mode (m=mode / s=size)
d	Data to be written

## 14 ICode 1

### 14.1 Features

- Read/write transponder
- 16 blocks or pages with 4 bytes/page = 64 bytes
- Page 0 und 1 are the serial number pages  
Pages 2,3 (4 optionally) are configuration pages
- Write protection by status bits

Attention: in case of a set "QUIET Bit" in the Transponder, it will not answer to any request. In this case first the "QUIET Bit" must first be reset. See function "Reset QUIET Bit".

### 14.2 Low Level Functions

These commands are basically those offered by the transponder air protocol.

#### 6A10: Anticollision / Select

Selects all tags not becoming involved in a collision.

<i>Host to RF System Request</i>	
<b>STX</b> "6A10" <hh> <e>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
h	Hash value (00 – 1F).
e	Timeslot exponent (0 – 7).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A10" <s> <f> [ <iiiiiiiiiiiiiiii> ] [ <s> <f> [ <iiiiiiiiiiiiiiii> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('y': data follows; 'n': no data).
i	Tag ID (64 bit).

#### 4A10: Selected Read

Reads all selected tags.

<i>Host to RF System Request</i>	
<b>STX</b> "4A10" <a> <b>	
<b>ETX</b> {c}	



<i>Request Parameters</i>	
a	Start block (0 – F).
b	Number of blocks to be read minus one (0 – F).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4A10" <s> <f> [ <dddddddd> [ ... ] ] [ <s> <f> [ <dddddddd> [ ... ] ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('y': data follows; 'n': no data).
d	Tag data.

#### 4A14: Unselected Read

Reads all unselected tags not becoming involved in a collision.

<i>Host to RF System Request</i>	
<b>STX</b> "4A14" <hh> <e> <s> <c>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
h	Hash value (00 – 1F).
e	Timeslot exponent (0 – 7).
s	Start block (0 – F).
c	Number of blocks to be read minus one (0 – F).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4A14" <s> <f> [ <dddddddd> [ ... ] ] [ <s> <f> [ <dddddddd> [ ... ] ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('y': data follows; 'n': no data).
d	Tag data.

#### 5A10: Write

Writes one block of data to some or all selected tags. A bitmask defines which tags are to be written. This mask consists of 32 bytes, each containing eight binary timeslot flags, beginning with the LSB. This function always refers to the most recent anticollision/select command.

<i>Host to RF System Request</i>	
<b>STX</b> "5A10" <hh> <s> <dddddddd> <tt...>	
<b>ETX</b> {c}	

<i>Request Parameters</i>	
h	Hash value (0x00 – 0x1f).
s	Block address (0x0 – 0xf).
d	Data to be written.
t	Timeslot bitmask (32 bytes).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5A10" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

The timeslot bitmask is determined as follows:

<i>TS</i>	<i>Binary Bitmask</i>			<i>Hex BM</i>
1	1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	...	010000..00
2	0 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0	...	020000..00
3	0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0	...	040000..00
4	0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0	...	080000..00
5	0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0	...	100000..00
6	0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0	...	200000..00
7	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	...	400000..00
8	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	...	800000..00
9	0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	...	000100..00
10	0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0	...	000200..00
11	0 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0	...	000400..00
12	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0	...	000800..00
13	0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 0	...	001000..00
14	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0	...	002000..00
15	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	...	004000..00
16	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	...	000800..00
...				

## 6A18: Halt

Puts some or all selected tags into halt mode. Once halted, a tag won't respond until its next reset. This function always refers to the most recent anticollision/select command.

<i>Host to RF System Request</i>	
<b>STX</b> "6A18" <hh> <tt...>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
h	Hash value (0x00 – 0x1f).
t	Timeslot bitmask (32 bytes).

<i>RF System to Host Response</i>
-----------------------------------

<b>ACK</b>	
<b>STX</b> "6A18" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6A1A: Reset QUIET Bit

Removes the QUIET flag from all tags in the field.

<i>Host to RF System Request</i>	
<b>STX</b> "6A1A"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A1A"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

### 6A1C: EAS

Attempts to read an EAS code from all tags present in the field.

<i>Host to RF System Request</i>	
<b>STX</b> "6A1C"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A1C" <rr...>	
<b>ETX</b> {c}	
<i>Response Data</i>	
r	EAS response (a 256 bit hex constant).

## 14.3 High Level Functions

These commands combine and/or simplify some of the low level commands.

## 6A20: Create Inventory

Builds an inventory of all tags currently present in the field by their serial number. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "6A20" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A20" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 04: Collision warning - check your ICSS_DT_* service settings. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

## 6A21: Get Inventory

This function will return all serial numbers from the current inventory.

<i>Host to RF System Request</i>	
<b>STX</b> "6A21"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A21" <ssss> [ <iiiiiiiiiiiiiiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.

i	Tag ID (64 bit).
---	------------------

### 6A22: Get ID Range from Inventory

To read a range of IDs or one single ID from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b>	"6A22" <iiii> [ <l111> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Index of first ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6A22" [ <cccc> ] [ <iiiiiiiiiiiiiiii> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
i	Tag ID (64 bit).

### 6A23: Realtime Inventory

Builds an inventory of all tags currently present in the field by their serial number. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

In contrast to the “create” inventory functions, this command sends tag IDs “on the fly”, i. e. as soon as they are detected, in multiple responses. The last response is always empty (0 IDs) and contains accumulated status flags. When the last response has been transmitted, the device is ready for the next command.

<i>Host to RF System Request</i>	
<b>STX</b>	"6A23" <m>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6A23" <ee> <ssss> [ <iiiiiiiiiiiiiiii> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	

e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 04: Collision warning - check your ICSS_DT_* service settings. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit).

### 6A24: Create/Get Inventory

This command is a combination of the commands "6A20" and "6A21". An inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6A24" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A24" <ee> <ssss> [ <iiiiiiiiiiiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 04: Collision warning - check your ICSS_DT_* service settings. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit).

### 6A26: Select Individual

Selects a single tag by its serial number. If found, the tag's timeslot exponent and number are returned, so that also low level commands may be used for further processing. If you get an IC\_STAT\_VER\_ERROR, this means that probably not all unwanted tags could be halted. Anyway, you can rely on the "direct write/halt" functions as well as all read functions to address only the desired tag, as they internally mask out occasional surplus tags. This function is breakable, that

means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b>	"6A26" <iiiiiiiiiiiiiiii>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Tag ID (64 bit).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6A26" <s> <f> [ <e> <tt> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('y': specs follow; 'n': not found).
e	Timeslot exponent.
t	Timeslot number.

#### 4A28: Looped Read

This function continuously repeats a read attempt until a tag has been recognized. It does not support the anticollision feature. Due to completely internal processing, it is very fast (approx. 200 attempts/sec for one block, 165 for two blocks, ...). This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b>	"4A28" <s> <c>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
s	Start block (0 – F).
c	Number of blocks to be read minus one (0 – F).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4A28" <s> <f> [ <dddddddd> [ <dddddddd> [ ... ] ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('y': specs follow; 'n': not found).
d	Tag data.

#### 5A24: Direct Write

Writes one block of data to all selected tags.

<i>Host to RF System Request</i>
----------------------------------

<b>STX</b> "5A24" <s> <dddddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
s	Block address (0 – F).
d	Data to be written.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5A24" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

## 6A28: Direct Halt

Puts all selected tags into halt mode.

<i>Host to RF System Request</i>	
<b>STX</b> "6A28"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A28" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

## 6A29: EAS Alarm

Searches for an EAS code and on success activates the EAS alarm. In looped mode, this command will be internally repeated until an EAS code has been found. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "6A29" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	EAS Mode: 'n': Normal. 'l': Looped.

<i>RF System to Host Response</i>	
<b>ACK</b>	



<b>STX</b> "6A29" <mm> <a>	
<b>ETX</b> {c}	
<i>Response Data</i>	
m	Number of matching bits.
a	Alarm state ('y': alarm; 'n': no alarm).

### 6A2A: Request Write Protect State

Queries a single block write protection status for all selected tags.

<i>Host to RF System Request</i>	
<b>STX</b> "6A2A" <s>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
s	Block address (0 – F).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A2A" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6A2B: Set Write Protect

Activates one or more block write protection flags on all selected tags.

<i>Host to RF System Request</i>	
<b>STX</b> "6A2B" <s> <c>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
s	Start block (0 – F).
c	Number of blocks to be protected minus one (0 – F).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A2B" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6A2C: Change EAS Bit

Switches the EAS flag on all selected tags.

<i>Host to RF System Request</i>	
----------------------------------	--

<b>STX</b>	"6A2C" <s>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
s	New bit value (0: EAS off; 1: EAS on).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6A2C" <s> [ <s> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6A2D: Set QUIET Bit

Sets the QUIET flag on all selected tags.

<i>Host to RF System Request</i>	
<b>STX</b>	"6A2D"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6A2D" <s> [ <s> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

## 14.4 Settings Functions

These commands are used to view or edit ICode specific settings.

### 1A30: Request Protocol Specific Setting

This function is used to retrieve the value of a certain ICode setting. Numeric values are unsigned 16 bit hex values; switch values are "0000" for "false" or "0001" for "true".

<i>Host to RF System Request</i>	
<b>STX</b>	"1A30" <kk>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	

<b>STX</b> "1A30" <vvvv>	
<b>ETX</b> {c}	
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Numeric/Switch, Default, Description):

Key	C	Alias	N/S	Def.	Description
01		ICS_FAMCODE	n	0000	Family code.
02		ICS_APPID	n	0000	Application identifier.
03		ICS_EAS_LEVEL	n	00C0	EAS bit threshold (0-FF).
04		<reserved>	-	0000	
05		<reserved>	-	0000	
06		ICS_RETRIES	n	0000	Automatic retries on inventory functions.
07		ICS_INV_SAFETY	n	0000	Number of retries before "incomplete" warning.
08		ICS_INV_COL	n	0018	Number of retries before "collision" warning.
09		ICS_CRC_ONLY	s	0001	Correct CRC suffices, don't check signal level.
0A		ICS_INV_RESET_S	s	0000	Perform HF reset at start of inventory command.
0B		ICS_INV_RESET_E	s	0001	Perform HF reset at end of inventory command.
0C		<reserved>	-	0000	
0D		ICS_FAST_MODE	s	0001	Use fast mode when transmitting to tag.
0E		ICS_VERIFY	s	0000	Verify mode, i. e. automatic "read after write".

### 1A31: Edit Protocol Specific Setting

Used to change the value of a certain ICode setting. See "Request Setting" ("1A30") for more details.

<i>Host to RF System Request</i>	
<b>STX</b> "1A31" <kk> <vvvv>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key.
v	Value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1A31"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

## 15 Tag-it (non-ISO)

## 15.1 Features

- Read/write transponder 13,56 MHz
- Anticollision possibility
- 256 bits (32 byte) user data in 8 blocks
- 32 bit (4 byte) SID address and version information
- Write protection for each user block

## 15.2 Low Level Functions

These commands are basically those offered by the transponder air protocol.

## 6B10: Single Anticollision Round

This command performs a complete SID Poll cycle. The response always contains all 16 timeslots. It is strongly recommended that you use the high level command "Create Inventory" ("6B20") instead, unless you know exactly what you are doing.

Host to RF System Request	
<b>STX</b>	"6B10" <i><11> [ <mm...> ]
<b>ETX</b>	{c}
Request Parameters	
i	Info flag ('i': info, 'n': no info).
l	Mask length.
m	Mask (rounded up to byte boundary, max. 32 bits / 4 bytes).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6B10" <e> <f> [ <iiiiiiii> [ <mm> <vvvv> <ss> <bb> ] ] <e> <f> [ <iiiiiiii> [ <mm> <vvvv> <ss> <bb> ] ] ...
<b>ETX</b>	{c}
<i>Response Data</i>	
e	Error status.
f	Data flag ('n': no data; 'y': data follows).
i	Tag ID (32 bit).
m	Chip manufacturer code.
v	Chip version.
s	Block size (in bytes) minus one.
b	Number of blocks minus one.

#### 4B10: Get Block

This command is used to read a single block of data from one tag. This may occur either in addressed mode, when a specific tag is called via its SID address, or in non-addressed mode. When using the latter, more tags than one in the field will lead to a collision, which means an unusable result.

<i>Host to RF System Request</i>	
<b>STX</b>	"4B10" <b> <f> [ <iiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	Block number.
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4B10" <e> <f> [ <ddddddd> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
e	Error status.
f	Data flag ('n': no data; 'y': no lock; 'u': user lock; 'f': factory lock).
d	Block data.

#### 4B16: Get Version

This function is used to query an individual tag's version and properties.

<i>Host to RF System Request</i>	
<b>STX</b>	"4B16" <f> [ <iiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
a	Tag ID (32 bit), expected only when addressed mode flag is 'y'.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4B16" <e> <f> [ <iiiiiii> <mm> <vvvv> <ss> <bb> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
e	Error status.
f	Data flag ('n': no data; 'y': data follows).
i	Tag ID (32 bit).
m	Chip manufacturer code.
v	Chip version.
s	Block size (in bytes) minus one.
b	Number of blocks minus one.

### 5B10: Put Block

This function writes a block of data to one or more tags.

<i>Host to RF System Request</i>	
<b>STX</b>	"5B10" <b> <dddddddd> <f> [ <iiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	Block number.
d	32 bit block data.
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5B10" <e>
<b>ETX</b>	{c}
<i>Response Data</i>	
e	Error status.

### 5B14: Put Block Lock

This function writes a block of data to one or more tags and locks (i. e. write protects) it.

<i>Host to RF System Request</i>	
<b>STX</b>	"5B14" <b> <dddddddd> <f> [ <iiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	Block number.
d	32 bit block data.
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5B14" <e>
<b>ETX</b>	{c}
<i>Response Data</i>	
e	Error status.

### 6B14: Lock Block

To write protect a previously unprotected block, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b>	"6B14" <b> <f> [ <iiiiiii> ]
<b>ETX</b>	{c}

<i>Request Parameters</i>	
b	Block number.
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6B14" <e>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Error status.

### 6B18: Quiet

To put a tag into Quiet mode, use this command. When in Quiet mode, a tag will not respond to any further request until it leaves and re-enters the field or a HF reset has been performed. The error status is used for verify mode only, otherwise it is 0.

<i>Host to RF System Request</i>	
<b>STX</b> "6B18" <f> [ <iiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6B18" <e>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Error status.

## 15.3 High Level Functions

These commands combine and/or simplify some of the low level commands.

### 6B20: Create Inventory

Builds an inventory of all tags currently present in the field by their serial number. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "6B20" <m>	
<b>ETX</b> {c}	

<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6B20" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found.
s	Current inventory size, i. e. the number of tags found.

### 6B21: Get Inventory

This function will return all serial numbers from the current inventory.

<i>Host to RF System Request</i>	
<b>STX</b> "6B21"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6B21" <ssss> [ <iiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (32 bit).

### 6B22: Get ID Range from Inventory

To read a range of IDs or one single ID from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b> "6B22" <iiii> [ <1111> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	



i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6B22" [ <cccc> ] [ <iiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
i	Tag ID (32 bit).

### 6B23: Realtime Inventory

Builds an inventory of all tags currently present in the field by their serial number. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

In contrast to the “create” inventory functions, this command sends tag IDs “on the fly”, i. e. as soon as they are detected, in multiple responses. The last response is always empty (0 IDs) and contains accumulated status flags. When the last response has been transmitted, the device is ready for the next command.

<i>Host to RF System Request</i>	
<b>STX</b> "6B23" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6B23" <ee> <ssss> [ <iiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (32 bit).

### 6B24: Create/Get Inventory

This command is a combination of the commands "6B20" and "6B21". An inventory is built and

instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6B24" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6B24" <ee> <ssss> [ <iiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (32 bit).

#### 4B24: Get Multiple Blocks

This command is used to read a variable amount of blocks from a single tag. This may occur either in addressed mode, when a specific tag is called via its SID address, or in non-addressed mode. When using the latter, more tags than one will lead to a collision, which means an unusable result. In "looped" mode, the reader will internally repeat the read attempt until all specified blocks have been read. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "4B24" <b> <c> <m> <f> [ <iiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Start block (first block: 0).
c	Number of blocks to be read minus one.
m	Mode ('n': normal; 'l': looped).
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.

<i>RF System to Host Response</i>	
<b>ACK</b>	

<b>STX</b> "4B24" <e> <f> [ <dddddddd> ]	
[ <e> <f> [ <dddddddd> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Error status.
f	Data flag ('n': no data; 'y': no lock; 'u': user lock; 'f': factory lock).
d	Block data.

## 5B24: Put Multiple Blocks

To write several blocks at once to a single tag, use this command.

<i>Host to RF System Request</i>	
<b>STX</b> "5B24" <b> <c> <dddddddd>	
[ <dddddddd> [ ... ] ] <f> [ <iiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Start block (first block: 0).
c	Number of blocks to be written minus one.
d	32 bit * (c+1) block data.
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5B24" <e>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Error status.

## 5B28: Put And Lock Multiple Blocks

To write several blocks at once to a single tag and lock them, use this command.

<i>Host to RF System Request</i>	
<b>STX</b> "5B28" <b> <c> <dddddddd>	
[ <dddddddd> [ ... ] ] <f> [ <iiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Start block (first block: 0).
c	Number of blocks to be written minus one.
d	32 bit * (c+1) block data.
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.

<i>RF System to Host Response</i>	
-----------------------------------	--

<b>ACK</b>	
<b>STX</b> "5B28" <e> [ <e> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Error status.

## 6B28: Lock Multiple Blocks

To write protect several blocks at once on a single tag, use this command.

<i>Host to RF System Request</i>	
<b>STX</b> "6B28" <b> <c> <f> [ <iiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Start block (first block: 0).
c	Number of blocks to be written minus one.
f	Addressed mode flag ('a': addressed; 'n': non-addressed).
i	Tag ID (32 bit), expected only when addressed mode flag is 'y'.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6B28" <e> [ <e> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Error status.

## 4B2A: Looped Address Scan

Continuously repeats an unaddressed Get Version attempt until a tag has been detected. This command does not support the anticollision feature. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "4B2A"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4B2A" <e> <f> [ <iiiiiii> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Error status.
d	Data flag ('n': no data; 'y': data follows).
i	Tag ID (32 bit).

## 15.4 Settings Functions

These commands are used to view or edit Tag-it specific settings.

### 1B30: Request Protocol Specific Setting

This function is used to retrieve the value of a certain Tag-it setting. Numeric values are unsigned 16 bit hex values; switch values are "0000" for "false" or "0001" for "true".

<i>Host to RF System Request</i>	
<b>STX</b> "1B30" <kk>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key (see table).

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1B30" <vvvv>	
<b>ETX</b> {c}	
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Numeric/Switch, Default, Description):

Key	C	Alias	N/S	Def.	Description
01		TIS_RETRIES	n	0000	Automatic retries on read/verify/inventory
02		TIS_INV_RESET_S	s	0000	Perform HF reset at start of inventory command.
03		TIS_INV_RESET_E	s	0001	Perform HF reset at end of inventory command.
04		TIS_VERIFY	s	0000	Automatic "read after write".
05		TIS_COLL_DEPTH	n	0000	Anticollision depth limit (0=none).

### 1B31: Edit Protocol Specific Setting

Used to change the value of a certain Tag-it setting. See "Request Setting" ("1B30") for more details.

<i>Host to RF System Request</i>	
<b>STX</b> "1B31" <kk> <vvvv>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key.
v	Value.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1B31"	
<b>ETX</b> {c}	

<i>Response Data</i>	
–	None.

## 16 ISO 15693

### 16.1 Features

- Read/write transponder 13,56 MHz
- 64 bit (8 byte) UID address

Since there are various different flavours of ISO 15693 tags (e. g. different memory size/arrangement, special features etc.), it is strongly recommended to consult the transponder datasheet for further information.

Due to memory limitation, the Scemtec 13.56MHz Long Range Reader can process only up to 256 bytes of data at once, e. g. 64 blocks of 32 bits or 128 blocks of 16 bits. Tag responses larger than this won't be understood; the system will assume a collision due to a missing EOF.

### 16.2 Low Level Functions

These commands are basically those offered by the transponder air protocol.

#### 6C10: Single Anticollision Round

This command performs a complete Inventory cycle. Thre response always contains all 16 timeslots. It is strongly recommended that you use the high level command "Create Inventory" ("6C20") instead, unless you know exactly what you are doing. When the AFI value in the settings is non-zero, it will be used to perform an AFI specific request.

<i>Host to RF System Request</i>	
<b>STX</b>	"6C10" <t> <l1> [ <mm...> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
t	Timeslot flag ('m': multiple/16 timeslots; 's': single/1 timeslot).
l	Mask length in bits.
m	Mask (rounded up to byte boundary, max. 64 bits / 8 bytes).
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6C10" <s> <f> [ <dd> <iiiiiiiiiiiiiiiiiiii> ]
	<s> <f> [ <dd> <iiiiiiiiiiiiiiiiiiii> ]
	...
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	DSFID byte.
i	Tag ID (64 bit) in reverse byte order.

## 6C12: Select

To put a single tag into "selected" state, use this command. When selected, a tag can quickly be addressed via the "selected" address mode.

<i>Host to RF System Request</i>	
<b>STX</b>	"6C12" <iiiiiiiiiiiiiiii>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6C12" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

## 4C10: Read Single Block

This command allows for reading a single block from an ISO 15693 tag. When activated in the ISO settings, also the block security information will be returned.

<i>Host to RF System Request</i>	
<b>STX</b>	"4C10" <bb> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	Block number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4C10" <s> <f> [ [ <pp> ] <dd> [ <dd> [ ... ] ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	Protection/security information.
d	Tag data.

## 4C12: Read Multiple Blocks

This command allows for reading several blocks at once from an ISO 15693 tag. When activated in the ISO settings, also the security information will be returned for each block.

<i>Host to RF System Request</i>	
----------------------------------	--



<b>STX</b>	"4C12" <bb> <cc> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	First block.
c	Number of blocks to be read minus one.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4C12" <s> <f> [ [ <pp> ] <dd> [ <dd> [ ... ] ] ] [ [ <pp> ] <dd> [ <dd> [ ... ] ] ] [ ... ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	Protection/security information.
d	Tag data.

#### 4C16: Get System Information

This command returns the system information of an ISO 15693 tag. The last 5 fields are present depending on the info flag.

<i>Host to RF System Request</i>	
<b>STX</b>	"4C16" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4C16" <s> <f> [ <oo> <iiiiiiiiiiiiiiii> [ <dd> ] [ <aa> ] [ <nn> <bb> ] [ <rr> ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
o	Info flag byte.
i	Tag ID (64 bit) in reverse byte order.
d	DSFID byte.
a	AFI byte.
n	Number of available blocks minus one.
b	Bytes per block minus one.
r	IC reference byte.

#### 4C18: Get Security Status

To determine the block security status of an ISO 15693 tag, use this function. In contrast to the read functions, this one allows for examining multiple blocks with one single access.

<i>Host to RF System Request</i>	
<b>STX</b>	"4C18" <bb> <cc> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	First block.
c	Number of blocks to be read minus one.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4C18" <s> <f> [ <pp> [ <pp> [ ... ] ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	Protection/security information.

#### 5C10: Write Single Block

This function can be used to write a single block of data to an ISO 15693 tag. Please make sure that the provided data fits the tag's block size.

<i>Host to RF System Request</i>	
<b>STX</b>	"5C10" <bb> <a> [ <iiiiiiiiiiiiiiii> ] <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	Block number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
d	Data to be written.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5C10" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

#### 5C12: Write Multiple Blocks

This function can be used to write multiple blocks of data to an ISO 15693 tag at once. Please make

sure that the provided data fits the tag's block size.

<i>Host to RF System Request</i>	
<b>STX</b>	"5C12" <bb> <cc> <a> [ <iiiiiiiiiiiiiiii> ] <dd> [ <dd> [ ... ] ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	First block.
c	Number of blocks minus one.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
d	Data to be written.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5C12" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 5C16: Write AFI

This function can be used to modify the AFI byte of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"5C16" <bb> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	AFI byte.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5C16" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 5C17: Write DSFID

This function can be used to modify the DSFID byte of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"5C17" <bb> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}

<i>Request Parameters</i>	
b	DSFID byte.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5C17" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C14: Lock Block

Use this function to lock a single data block of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6C14" <bb> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block to be locked.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C14" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C16: Lock AFI

Use this function to lock the AFI byte of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6C16" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C16" <s>	
<b>ETX</b> {c}	

<i>Response Data</i>	
s	Status value.

### 6C17: Lock DSFID

Use this function to lock the DSFID byte of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6C17" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C17" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C18: Stay Quiet

Use this function to put an ISO 15693 tag into QUIET mode. The tag will not respond to any command except "Reset" or "Select" afterwards, until it receives an HF reset or one of the two commands mentioned before.

<i>Host to RF System Request</i>	
<b>STX</b> "6C18" <iiiiiiiiiiiiiiii>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C18"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 6C1A: Reset To Ready

To reset a tag into "ready" state, use this command.

<i>Host to RF System Request</i>	
<b>STX</b> "6C1A" <a> [ <iiiiiiiiiiiiiiii> ]	

<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C1A" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C1C: Low Level Custom Command

This function can be used to pass an arbitrary command to an ISO 15693 tag. Unlike the commands 6C1E and 6C1F, this one does neither append the CRC16, nor evaluate or remove the received CRC16. Also the flags are not evaluated; the user must take care of the response settings explicitly using command 1C31. Apart from the ISO framing that is still handled, it's pretty transparent.

The response delay time, i. e. the time between end of transmission and start of reception must be specified in units of  $16T_0$ . To achieve the nominal  $320.9\mu s$  specified in ISO 15693, a theoretical value of 110h would be correct, though actually a slightly higher value (e. g. 114h) should be more accurate due to a minor offset induced by hardware design.

Further, the maximum number of expected response bytes must be specified. If the actual number of bytes received is smaller, this is not really a problem. If the actual number of bytes received is higher, the EOF is missed and status 2 (STAT\_COLL\_CRC) is generated.

<i>Host to RF System Request</i>	
<b>STX</b> "6C1C" <tttt> <ll> <c...>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	Response time in units of $16T_0$ ( $1T_0=0.0737\mu s$ ; $16T_0=1.18\mu s$ ).
l	Number of expected response bytes (maximum).
c	Command beginning with flags and ending with CRC.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C1C" <s> <f> [ <r...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
r	Complete tag response with CRC.

### Examples:

a) Equivalent to the STX/ETX command 6C10s00 (Get ID with single slot):

**STX** "6C1C2701002A50" **ETX** {c}

*Response (in case of success):*

**ACK STX** "6C1C0y0000FE762615000104E0538C" **ETX** {c}

### **6C1E: Custom Read Command**

This function can be used to pass an arbitrary command to an ISO 15693 tag. The flags will be evaluated and temporarily used (like modulation type or data rate). You might even use this command to imitate any other ISO 15693 command. Use the "Custom Read Command" function for commands which do not perform any write functions, since the response timing is different here.

<i>Host to RF System Request</i>	
<b>STX</b> "6C1E" <ff> <cc> <mm> <p> [ <p> [ ... ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
f	Request flags.
c	Custom command code.
m	IC manufacturer code.
p	Custom request parameters.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C1E" <s> <f> [ <r...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
r	Complete tag response.

### Examples:

a) Read Infineon my-d tag in custom mode:

**STX** "6C1E" "00" "A0" "05" "10" <pp> "00" **ETX** {c}

p: Page number to be read

### **6C1F: Custom Write Command**

This function can be used to pass an arbitrary command to an ISO 15693 tag. The flags will be

evaluated and temporarily used (like modulation type or data rate). You might even use this command to imitate any other ISO 15693 command. Use the "Custom Write Command" function for commands which perform write functions, since the response timing is different here.

<i>Host to RF System Request</i>	
<b>STX</b> "6C1F" <ff> <cc> <mm> <p> [ <p> [ ... ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
f	Request flags.
c	Custom command code.
m	IC manufacturer code.
p	Custom request parameters.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C1F" <s> <f> [ <r...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
r	Complete tag response.

### **Examples:**

a) Write Infineon my-d tag in custom mode:

```
STX "6C1F" "00" "A0" "05" "30" <pp> "00" <dddddddddddddddd>
ETX <c>
```

p: Page number to be written  
 d: Data to be written (64 bits in this example)

a) Write two TI Tag-it ISO blocks:

```
STX "6C1F" "40" "A2" "07" <pp> <dddddddddddddddd> ETX <c>
```

p: First page number, must be even  
 d: Data to be written (64 bits in this example)

c) Write Philips I\*Code SLI tag with high data rate and FSK:

```
STX "6C1F" "03" "21" <pp> <ddddddd> ETX {c}
```

p: Page number to be written (32 bits in this example)  
 d: Data to be written

Please note that in the last example the "A0" custom command code is replaced by "21", which is



the command code for “Write Single Block” defined by the ISO 15693 specification. In fact, you can specify any command you want to be sent to the tag directly with the 6C1E/6C1F function. Only the tag’s framing and CRC is generated automatically.

### 1C1E: Set Custom Response Delay

Temporarily sets the response delay time for the next Custom Command to a specified value. This can be used for custom commands depending on a response delay time which doesn't comply to ISO15693, e. g. measurement functions which take a lot of time. The next 6C1E or 6C1F command will use this delay time and then automatically reset it to zero.

<i>Host to RF System Request</i>	
<b>STX</b> "1C1E" <dddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
d	Delay in units of 16T <sub>0</sub> (1.18µs).
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1C1E"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

## 16.3 High Level Functions

These commands combine and/or simplify some of the low level commands.

### 6C20: Create Inventory

Builds an inventory of all tags currently present in the field by their serial number. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "6C20" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.
<i>RF System to Host Response</i>	

<b>ACK</b>	
<b>STX</b> "6C20" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

### 6C21: Get Inventory

This function will return all serial numbers from the current inventory.

<i>Host to RF System Request</i>	
<b>STX</b> "6C21" [ <r> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
r	Response mode: 'i': Response consists of unique ID only (default if response mode is not specified). 'd': Response consists of DSFID only. 'a': Response consists of all available data (unique ID preceded by DSFID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C21" <ssss> [ [ <dd> ] [ <iiiiiiiiiiiiiiii> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
d	Tag DSFID byte.
i	Tag ID (64 bit) in reverse byte order.

### 6C22: Get ID Range from Inventory

To read a range of IDs or one single ID from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b> "6C22" <iiii> [ <l111> [ <r> ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).
r	Response mode:

	'i': Response consists of unique ID only (default if response mode is not specified). 'd': Response consists of DSFID only. 'a': Response consists of all available data (unique ID preceded by DSFID).
--	---

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C22" [ <cccc> ] [ [ <dd> ] [ <iiiiiiiiiiiiiiiiiii> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
d	Tag DSFID byte.
i	Tag ID (64 bit) in reverse byte order.

### 6C23: Realtime Inventory

Builds an inventory of all tags currently present in the field by their serial number. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

In contrast to the “create” inventory functions, this command sends tag IDs “on the fly”, i. e. as soon as they are detected, in multiple responses. The last response is always empty (0 IDs) and contains accumulated status flags. When the last response has been transmitted, the device is ready for the next command.

<i>Host to RF System Request</i>	
<b>STX</b> "6C23" <m> [ <r> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset.
r	Response mode: 'i': Response consists of unique ID only (default if response mode is not specified). 'd': Response consists of DSFID only. 'a': Response consists of all available data (unique ID preceded by DSFID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C23" <ee> <ssss> [ [ <dd> ] [ <iiiiiiiiiiiiiiiiiii> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow.

	08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
d	Tag DSFID byte.
i	Tag ID (64 bit) in reverse byte order.

#### 6C24: Create/Get Inventory

This command is a combination of the commands "6C20" and "6C21". An inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6C24" <m> [ <r> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.
r	Response mode: 'i': Response consists of unique ID only (default if response mode is not specified). 'd': Response consists of DSFID only. 'a': Response consists of all available data (unique ID preceded by DSFID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C24" <ee> <ssss> [ [ <dd> ] [ <iiiiiiiiiiiiiiii> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
d	Tag DSFID byte.
i	Tag ID (64 bit) in reverse byte order.

#### 4C20: Advanced Read Single Block

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate read

command. This may be necessary since some tags (e. g. the first My-D) don't support the default read command. In "looped" mode, the reader will internally repeat the read attempt until the specified block has been successfully read. In "looped ISO-only" mode, the above determination will be omitted to increase performance. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "4C20" <bb> <m> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
m	Mode ('n': normal; 'l': looped; 'i': looped ISO-only).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4C20" <s> <f> [ [ <pp> ] <dd> [ <dd> [ ... ] ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	Protection/security information.
d	Tag data.

#### 4C24: Advanced Read Multiple Blocks

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate read command (either "Read Multiple Blocks" or "Read Single Block"). In "looped" mode, the reader will internally repeat the read attempt until all specified blocks have been successfully read. In "looped ISO-only" mode, the above determination will be omitted to increase performance. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "4C24" <bb> <cc> <m> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
c	Number of blocks to be read minus one.
m	Mode ('n': normal; 'l': looped; 'i': looped ISO-only).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
-----------------------------------	--

<b>ACK</b>	
<b>STX</b> "4C24" <s> <f> [ [ <pp> ] <dd> [ <dd> [ ... ] ] ]	
[ <s> <f> [ [ <pp> ] <dd> [ <dd> [ ... ] ] ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	Protection/security information.
d	Tag data.

### 5C20: Advanced Write Single Block

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate write command. This may be necessary since some tags (e. g. the first My-D) don't support the default write command.

<i>Host to RF System Request</i>	
<b>STX</b> "5C20" <bb> <a> [ <iiiiiiiiiiiiiiii> ] <dd> [ <dd> [ ... ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
d	Block data.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5C20" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 5C24: Advanced Write Multiple Blocks

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate write command (either "Write Multiple Blocks" or "Write Single Block"). This may be necessary since some tags (e. g. the first My-D) don't support the default write command.

<i>Host to RF System Request</i>	
<b>STX</b> "5C24" <bb> <cc> <a> [ <iiiiiiiiiiiiiiii> ]	
<dd> [ <dd> [ ... ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.

c	Number of blocks to be written minus one.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
d	Block data.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5C24" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C26: Advanced Lock Single Block

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate lock command. This may be necessary since some tags (e. g. the first My-D) don't support the default lock command.

<i>Host to RF System Request</i>	
<b>STX</b> "6C26" <bb> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C26" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C28: Advanced Lock Multiple Blocks

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate lock command. This may be necessary since some tags (e. g. the first My-D) don't support the default lock command. In addition, this command can lock a whole range of blocks at once.

<i>Host to RF System Request</i>	
<b>STX</b> "6C28" <bb> <cc> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.

c	Number of blocks to be locked minus one.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C28" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 5C26: Advanced Write AFI

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate read command. This function can be used to modify the AFI byte of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "5C26" <bb> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	AFI byte.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5C26" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 5C27: Advanced Write DSFID

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate write command. This function can be used to modify the DSFID byte of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "5C27" <bb> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	DSFID byte.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.



<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5C27" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C2A: Advanced Lock AFI

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate lock command. Use this function to lock the AFI byte of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6C2A" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C2A" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C2B: Advanced Lock DSFID

The Advanced Read/Write/Lock functions are more powerful than the respective low-level commands, for they first determine the tag manufacturer code and then use the appropriate lock command. Use this function to lock the DSFID byte of an ISO 15693 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6C2B" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6C2B" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

#### 4C2A: Looped Address Scan

Continuously repeats a single-timeslot inventory attempt until a tag has been detected. This command does not support the anticollision feature. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b>	"4C2A"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4C2A" <s> <f> [ <dd> <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	DSFID byte.
i	Tag ID (64 bit) in reverse byte order.

## 16.4 Settings Functions

These commands are used to view or edit ISO 15693 specific settings.

#### 1C30: Request Protocol Specific Setting

This function is used to retrieve the value of a certain ISO-Tag setting. Numeric values are unsigned 16 bit hex values; switch values are "0000" for "false" or "0001" for "true".

<i>Host to RF System Request</i>	
<b>STX</b>	"1C30" <kk>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1C30" <vvvv>
<b>ETX</b>	{c}
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Numeric/Switch, Default, Description):

Key	C	Alias	N/S	Def.	Description
-----	---	-------	-----	------	-------------

01	ISOS_MOD_DEPTH	s	0001	Modulation depth (0=20%; 1=20/100%; 2=100%).
02	ISOS_SUB_CARRIER	s	0001	Response modulation type (0=ASK, 1=FSK).
03	ISOS_DATA_RATE	s	0001	Response data rate (0=low; 1=high).
04	ISOS_WRITE_TRIG	s	0000	Use trigger mode when writing.
05	ISOS_AFI	n	0000	AFI byte for Inventory (0: ignore).
06	ISOS_READ_SEC	s	0000	Get security information when reading.
07	ISOS_STRICT_FRAME	s	0000	Use most marginal frame bits of a response.
08	ISOS_RETRIES	n	0000	Retries on read and inventory functions.
09	ISOS_REVERT_BYTES	s	0000	Transmit ID and tag data in reverse byte order.
0A	ISOS_TEMPS_K1	n	010C	Calibration value 1 (in Kelvin).
0B	ISOS_TEMPS_K2	n	0120	Calibration value 2 (in Kelvin).
0C	ISOS_EAS_LEVEL	n	00C0	ICode SLI EAS alarm bit threshold (0-FF).
0D	<reserved>	-	0000	
0E	ISOS_INV_RESET_S	s	0000	Perform HF reset at start of inventory command.
0F	ISOS_INV_RESET_E	s	0001	Perform HF reset at end of inventory command.
10	ISOS_FAST_INV	s	0001	Start inventory with one timeslot.
11	ISOS_WRITE_PREC	n	0001	Write resp. pre-counter for low data rate.
12	ISOS_FAST_MODE	s	0001	Use high data rate from reader to tag.
13	ISOS_TI_FASTSID	s	0000	Use FastSID for TI tags. Currently not operational!
14	ISOS_VERIFY	s	0000	Verify mode, i. e. automatic “read after write”.
15	ISOS_COLL_DEPTH	n	0000	Anticollision depth limit (0=none).
16	ISOS_RESP_RESYNC	s	0000	Dynamic response resynchronization in case of lost response clocks (tag bug workaround).
17	ISOS_EM_QSTORAGE	s	0000	Use EM “Quiet Storage” command instead of Quiet command for EM tags.
18	ISOS_SLIS_INV_PG	n	0000	Page number to read on ICode SLI-S Custom Inventory.
19	ISOS_SLIS_INV_ID	s	0000	Each data block is preceded by its tag ID.
1A	ISOS_SLIS_INV_R	s	0000	Use double data rate for SLI-S Custom Inventory.
1B	X ISOS_DSFID_OUTPUT	s	0000	Output DSFID with tag ID.
1C	X ISOS_DSFID_VALUE	n	0000	DSFID Filter value (DSFID byte for comparison).
1D	X ISOS_DSFID_FILTER	s	0000	DSFID Filter switch (0=inactive; 1=active).
1E	ISOS_ID_VERIFY	S	0001	Drop all IDs which appear in a wrong timeslot (0=no; 1=yes) as collisions. This is a workaround for LRI64 bug resulting in false IDs.
1F	ISOS_VAR_ID_LEN	s	0000	By default, the ID length is fixed to 64 bits plus 8 bits DSFID. When 1 is configured here, length verification is disabled.
20	ISOS_BLKSIZE_DEF	n	0003	Default block size (in bytes minus one) to be assumed by “advanced” functions in case tag manufacturer and type are unknown.
21	ISOS_BLKCOUNT_DEF	n	00FF	Default block count (minus one) to be assumed by “advanced” functions in case tag manufacturer and type are unknown.
22	ISOS_AUTO_HALT	s	0001	When enabled, each inventory round is started by sending a “Quiet” command to all IDs in the inventory. This can be useful for cumulative or sequencer mode.
23	ISOS_INV_TIMEOUT	n	0000	This timeout (in milliseconds) is used for each inventory cycle (and for each sequencer step). Each cycle is forced to stop after this time. A value of 0 deactivates the timeout.

### 1C31: Edit Protocol Specific Setting

Used to change the value of a certain ISO-Tag setting. See "Request Setting" ("1C30") for more details.

<i>Host to RF System Request</i>	
<b>STX</b> "1C31" <kk> <vvvv>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key.
v	Value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1C31"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 1C34: Get ISO Tag Descriptor

This function reads an item specified via its index from the tag descriptor table. This table is used by the "Advanced" high level functions as an extension to the ROM descriptor table to determine the capabilities of an ISO 15693 transponder. There are 10 slots in the RAM part of the table; it can be stored to EEPROM using the E002 function. If an invalid index is specified, a parameter error is returned.

<i>Host to RF System Request</i>	
<b>STX</b> "1C34" <ii>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of slot to read.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1C34" <hh> <ff> <tt> <mm> <ss> <cc> <ffff>	
<b>ETX</b> {c}	
<i>Response Data</i>	
h	Header byte (first byte of UID); usually E0.
f	Manufacturer ID (second byte of UID).
t	Type ID (third byte of UID).
m	Type Mask (bit mask to "AND" to type ID byte).
s	Block size in bytes minus one.
c	(Minimal available) block count minus one.
f	Property flags: Bit #0: Tag supports 20% modulation index. Bit #1: Tag supports 100% modulation index.

Bit #2: Tag supports single subcarrier response.
Bit #3: Tag supports dual subcarrier response.
Bit #4: Tag supports 1 out of 256 mode.
Bit #5: Tag supports 1 out of 4 mode.
Bit #6: Tag supports low (1/4) data rate response.
Bit #7: Tag supports high data rate response.
Bit #8: Tag supports writing in timer mode (option flag clear).
Bit #9: Tag supports writing in trigger mode (option flag set).
Bit #10: Tag supports standard read single block command.
Bit #11: Tag supports standard write single block command.
Bit #12: Tag supports standard read multiple blocks command.
Bit #13: Tag supports standard write multiple blocks command.
Bit #14: <Reserved for future use>
Bit #15: <Reserved for future use>

### 1C35: Set ISO Tag Descriptor

This function writes an item specified via its index to the tag descriptor table.

<i>Host to RF System Request</i>	
<b>STX</b>	"1C35" <ii> <hh> <ff> <tt> <mm> <ss> <cc> <ffff>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Index of slot to write.
h	Header byte (first byte of UID); usually E0.
f	Manufacturer ID (second byte of UID).
t	Type ID (third byte of UID).
m	Type Mask (bit mask to "AND" to type ID byte).
s	Block size in bytes minus one.
c	(Minimal available) block count minus one.
f	Property flags: Bit #0: Tag supports 20% modulation index. Bit #1: Tag supports 100% modulation index. Bit #2: Tag supports single subcarrier response. Bit #3: Tag supports dual subcarrier response. Bit #4: Tag supports 1 out of 256 mode. Bit #5: Tag supports 1 out of 4 mode. Bit #6: Tag supports low (1/4) data rate response. Bit #7: Tag supports high data rate response. Bit #8: Tag supports writing in timer mode (option flag clear). Bit #9: Tag supports writing in trigger mode (option flag set). Bit #10: Tag supports standard read single block command. Bit #11: Tag supports standard write single block command. Bit #12: Tag supports standard read multiple blocks command. Bit #13: Tag supports standard write multiple blocks command. Bit #14: <Reserved for future use>

	Bit #15: <Reserved for future use>
--	------------------------------------

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1C35"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 1C36: Get ISO Tag Descriptor from ROM

This function reads an item specified via its index from the factory tag descriptor table located in ROM. This table is used by the "Advanced" high level functions to determine the capabilities of an ISO 15693 transponder. If an invalid index is specified, a parameter error is returned. User descriptors can extend this table (see "Set ISO Tag Descriptor" command), but they can also be used to overshadow a ROM table item in case of incorrect data. Usually, it is not necessary to read the ROM descriptor table.

<i>Host to RF System Request</i>	
<b>STX</b>	"1C36" <ii>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Index of slot to read.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1C36" <hh> <ff> <tt> <mm> <ss> <cc> <ffff>
<b>ETX</b>	{c}
<i>Response Data</i>	
h	Header byte (first byte of UID); usually E0.
f	Manufacturer ID (second byte of UID).
t	Type ID (third byte of UID).
m	Type Mask (bit mask to "AND" to type ID byte).
s	Block size in bytes minus one.
c	(Minimal available) block count minus one.
f	Property flags: Bit #0: Tag supports 20% modulation index. Bit #1: Tag supports 100% modulation index. Bit #2: Tag supports single subcarrier response. Bit #3: Tag supports dual subcarrier response. Bit #4: Tag supports 1 out of 256 mode. Bit #5: Tag supports 1 out of 4 mode. Bit #6: Tag supports low (1/4) data rate response. Bit #7: Tag supports high data rate response. Bit #8: Tag supports writing in timer mode (option flag clear). Bit #9: Tag supports writing in trigger mode (option flag set).

	Bit #10: Tag supports standard read single block command. Bit #11: Tag supports standard write single block command. Bit #12: Tag supports standard read multiple blocks command. Bit #13: Tag supports standard write multiple blocks command. Bit #14: <Reserved for future use> Bit #15: <Reserved for future use>
--	--

### 1C38: Set Temporary Response Delay

Temporarily sets the response wait time for the next ISO15693 command to a specified value. This can be used for commands depending on a response wait time which doesn't comply to ISO15693, e. g. measurement functions which take a lot of time. The next command will use this wait time and then automatically reset it to zero.

<i>Host to RF System Request</i>	
<b>STX</b>	"1C38" <dd>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
d	Delay in units of $512T_0$ (37.76µs); maximum is FF=9628µs.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1C38"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

## 17 ISO 15693 Custom

### 17.1 TI Tag-it ISO Custom Functions

Please consult the tag data sheet for more information.

#### 5C82: Write Two Blocks

This function uses a Tag-it specific custom command to access the tag in custom mode, writing two blocks with one command. It is faster to write two blocks at once, since the EEPROM programming time remains the same even for two blocks.

<i>Host to RF System Request</i>	
<b>STX</b> "5C82" <bb> <a> [ <iiiiiiiiiiiiiiii> ] <dddddddddddddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
d	Block data.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5C82" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 17.2 Infineon My-D Custom Functions

Please consult the tag data sheet for more information.

#### 4C90: Read Block

This function uses a My-D specific custom command to access the tag in custom mode. Note that the first four blocks are read-only system blocks in this mode, and all blocks have a width of 64 bit.

<i>Host to RF System Request</i>	
<b>STX</b> "4C90" <bb> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.



<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4C90" <s> <f> [ [ <pp> ] <dddddddddddddddd> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	Protection/security information.
d	Tag data.

### 5C90: Write Block

This function uses a My-D specific custom command to access the tag in custom mode. Note that the first four blocks are read-only system blocks in this mode, and all blocks have a width of 64 bit.

<i>Host to RF System Request</i>	
<b>STX</b> "5C90" <bb> <a> [ <iiiiiiiiiiiiiiii> ] <dddddddddddddddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
d	Block data.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5C90" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6C94: Lock Block

This function uses a My-D specific custom command to access the tag in custom mode. Note that the first four blocks are read-only system blocks in this mode, and all blocks have a width of 64 bit.

<i>Host to RF System Request</i>	
<b>STX</b> "6C94" <bb> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
-----------------------------------	--

<b>ACK</b>	
<b>STX</b> "6C94" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

## 17.3 Philips ICode SLI Custom Functions

Please consult the tag data sheet for more information.

### 6CA0: Inventory Read

This function reads the tag data blocks in a special mode which operates much the same as ISO 15693 anticollision (see also the command description for “Single Anticollision Round”). Instead of the tag ID, specified pages of tag data are transmitted to the host. Optionally, the tag ID can be transmitted together with the data, but in that mode the masked parts of the ID are *not* transmitted. Another option is the data rate used by the tag for transmission, which can be either ISO 15693 compliant or twice as fast (which is not ISO compliant). In Double data rate mode, the system will automatically use ASK; the tag does not support FSK in that mode.

<i>Host to RF System Request</i>	
<b>STX</b> "6CA0" <r> <i> <bb> <cc> <t> <ll> [ <mm...> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
r	Downlink data rate ('n': normal ISO 15693; 'h': double).
i	Add part of ID to data ('i': add ID; 'd': data only).
b	Start block to read.
c	Number of blocks to read minus one.
t	Timeslot switch ('s': 1; 'm': 16).
l	Mask length in bits.
m	Mask data, padded with zeros to byte boundary.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CA0" <s> <f> [ [ <ii...> ] <pp> <dd...> ] [ <s> <f> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
i	Part of tag ID.
p	Protection information byte.
d	Tag data.

## 6CA4: Create IR Inventory

Builds an inventory of all tags currently present in the field. Instead of the standard anticollision, the ICode SLI custom command "Inventory Read" (IR) is used. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

The ICode SLI specific custom anticollision options are controlled via the keys ISOS\_SLIS\_INV\_PG (first block to read), ISOS\_SLIS\_INV\_ID (add tag ID at beginning of data) and ISOS\_SLIS\_INV\_R (double data rate) in the ISO settings. For compatibility with the corresponding SLI-L/S command, this command is fixed to a size of 4 blocks (16 bytes).

<i>Host to RF System Request</i>	
<b>STX</b> "6CA4" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CA4" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

## 6CA5: Get IR Inventory

This function will return all data from the current IR inventory.

<i>Host to RF System Request</i>	
<b>STX</b> "6CA5"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CA5" <ssss> [ [ <iiiiiiiiiiiiiiii> ] <dddddddddddddddddddddddddddddd> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of four memory blocks (start block is selected via ISOS_SLIS_INV_PG).

### 6CA6: Get ID Range from IR Inventory

To read a range of IDs or one single ID from a previously created IR inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b> "6CA6" <iiii> [ <1111> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CA6" [ <cccc> ] [ <iiiiiiiiiiiiiiii> ] <dddddddddddddddddddddddddd> [ ... ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of four memory blocks (start block is selected via ISOS_SLIS_INV_PG).

### 6CA7: Create/Get IR Inventory

This command is a combination of the commands "6CA4" and "6CA5". An IR inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6CA7" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative.

	'C': Cumulative without HF reset.
--	-----------------------------------

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CA7" <ee> <ssss> [ [ <iiiiiiiiiiiiiiii> ] <dddddddddddddddddddddddddddd> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of four memory blocks (start block is selected via ISOS_SLIS_INV_PG).

### 6CA8: Change EAS Flag

This command can be used to set/unset the EAS flag of an ICode SLI label.

<i>Host to RF System Request</i>	
<b>STX</b> "6CA8" <e> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
e	EAS flag ('0': clear; '1': set).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CA8" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CA9: Lock EAS Flag

This command can be used to lock the EAS flag of an ICode SLI label.

<i>Host to RF System Request</i>	
<b>STX</b> "6CA9" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	

a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CA9" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CAA: EAS

Attempts to read an EAS code from all tags present in the field.

<i>Host to RF System Request</i>	
<b>STX</b> "6CAA"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CAA" <rr...>	
<b>ETX</b> {c}	
<i>Response Data</i>	
r	EAS response (a 256 bit hex constant).

### 6CAB: EAS Alarm

Searches for an EAS code and on success activates the EAS alarm. In looped mode, this command will be internally repeated until an EAS code has been found. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "6CAB" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode ('n': normal; 'l': looped).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CAB" <mm> <a>	
<b>ETX</b> {c}	
<i>Response Data</i>	
m	Number of matching bits, 0 on perfect match.
a	Alarm state ('y': alarm; 'n': no alarm).

## 17.4 KSW TempSens Custom Functions

Please consult the tag data sheet for more information.

### 6CB0: Start Log

This command resets the tag and puts it into log mode, using the given settings. In "cyclic" mode, the memory serves as a ring buffer, i. e. when it is full, logging will restart at the beginning, overwriting old values. It is also possible to skip ("preserve") 0-3 upper banks when logging, leaving them for user data. Warning: This function resets the whole chip, any data previously contained will be lost. If you plan to program the log start time to block #48h, do this after starting the log.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CB0" <tttt> <c> <p> <m> [ <llll> <uuuu> ] <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
t	Timer period for measurement; the unit is ticks (approximately 0.6s).
c	Cyclic mode ('c': cyclic; 's': single).
p	Preserve upper banks (0 – 3).
m	Mode ('r': values outside range; 'l': only values crossing limits).
l	Lower limit for 'l' mode, given in Kelvin*10.
u	Upper limit for 'l' mode, given in Kelvin*10.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CB0" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CB2: Stop Log

This command turns off logging, i. e. when the tag leaves the RF field, it will remain inactive. No measurement will be done; the battery will not be used up.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CB2" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CB2" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CB4: Get Log Entry

This command reads a single entry from the temperature log. You need to specify an offset - if in cyclic mode, the oldest value will be offset 0, whereas in single mode, the offset is equal to the block number.

<i>Host to RF System Request</i>	
<b>STX</b> "6CB4" <oo> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
o	Offset from first recorded value (00 – 3F).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CB4" <s> <f> [ <bb> <tttt> <mmmm> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
b	Block number of this entry.
t	Time of measurement (ticks).
m	Measured temperature, given in Kelvin*10.

### 6CB6: Get Status

This command comes in handy if you want to know the current measurement status of a tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6CB6" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CB6" <s> <f> [ <tttt> <cccc> <pp> <nn> <ll> ]	



<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
t	Timer period setting (ticks).
c	Current timer value (ticks).
p	Log pointer, i. e. the next block used for logging.
n	Total number of measurements so far.
l	Log flags: Bit #6: Overflow; the complete log has been filled at least once. Bit #7: Battery warning; during measurement the battery was low.

### 6CB8: Lock

This command write protects the complete tag using a password.

<i>Host to RF System Request</i>	
<b>STX</b> "6CB8" <pppppppppppp> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	Password needed later on to unlock the tag.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CB8" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CB9: Unlock

This command removes a previously set write protection.

<i>Host to RF System Request</i>	
<b>STX</b> "6CB9" <pppppppppppp> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	Password previously used for the "lock" command.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CB9" <s>	

<b>ETX</b>	{ c }
<i>Response Data</i>	
s	Status value.

## 17.5 EM 4x3x Custom Functions

Please consult the tag data sheet for more information.

### 4CC8: Fast Read Multiple Blocks

This function reads data blocks from an EM4237 tag using its double data rate “Fast Read Multiple Blocks” command.

<i>Host to RF System Request</i>	
<b>STX</b>	"4CC8" <bb> <cc> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
b	Start block number.
c	Number of blocks minus one (0 = 1 block).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4CC8" <s> <f> [ [ <pp> ] <dd> [ <dd> [ ... ] ] ] [ <s> <f> [ [ <pp> ] <dd> [ <dd> [ ... ] ] ] [ ... ] ]
<b>ETX</b>	{ c }
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	Protection/security information.
d	Tag data.

### 4CCA: Read Signature

This function reads the signature data from an EM4237 tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"4CCA" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	

<b>STX</b>	"4CCA" <s> <f> [ <dd...> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Signature data.

### 5CC8: Change Key

This function can be used to write a new password or secret key to an EM4237 tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"5CC8" <i> <k> <a> [ <i> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Key identifier: 00h: Secure Key [31:0] or password. 01h: Secure Key [63:32]. 02h: Secure Key [95:64]. 03h: Secure Key [127:96].
k	Key data (8 hex digits).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5CC8" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CC0: Login

This command enables the Secure Mode of an EM4034 or EM4237 tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CC0" <p> <a> [ <i> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
p	Tag password.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CC0" <s>

<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CC1: Login using Key/Password Buffer

This command enables the Secure Mode of an EM4034 or EM4237 tag. In contrast to the previous function, the password is not expected as a parameter but rather taken from the reader's Key/Password Buffer, which can be set using command 6010 (password must be 8 hex digits long) and made permanent using command E002.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CC1" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CC1" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CC2: Inventory Block Read

This command utilizes the Inventory Block Read of an EM4237 tag. Instead of using this low level function, it is generally a good idea to use the high level inventory functions 6CCC-6CCF.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CC2" <i> <pp> <cc> <t> <ll> [ <mm...> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Add part of ID to data ('i': add ID; 'd': data only).
p	Start block to read.
c	Number of blocks to read minus one.
t	Timeslot switch ('s': 1; 'm': 16).
l	Mask length in bits.
m	Mask data, padded with zeroes to byte boundary.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CC2" <s> <f> [ [ <ii...> ] <ss> <dd...> ] [ <s> <f> [ ... ] ]
<b>ETX</b>	{c}

<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
i	Part of tag ID.
s	DSFID byte.
d	Tag data.

### 6CC3: Destroy

This command sets target EM4237 tag's "destroyed" bit to render it permanently disabled.

<i>Host to RF System Request</i>	
<b>STX</b> "6CC3" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CC3" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CC4: Get Multiple Block Protection Status

This command reads the protection status of one or multiple EM4237 blocks.

<i>Host to RF System Request</i>	
<b>STX</b> "6CC4" <bb> <cc> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Start block number.
c	Number of blocks minus one (0 = 1 block).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CC4" <s> <pp> [ <pp> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
p	Block protection status: 00h: Readable + Writeable.

	01h: Read protected. 10h: Write protected. 11h: Read + Write protected.
--	---

### 6CC5: Protect Memory Page

This command changes the protection status of a single EM4237 memory page.

<i>Host to RF System Request</i>	
<b>STX</b> "6CC5" <pp> <ss> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	Memory page to alter (1 page = 4 blocks).
s	New protection status: 00h: Readable + Writeable. 01h: Read protected. 10h: Write protected. 11h: Read + Write protected.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CC5" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CC6: Set Privacy Mode

This command enables or disables the Privacy Mode of an EM4237 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6CC6" <e> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
e	Enable or disable: 0: Disable privacy. 1: Enable privacy.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CC6" <s>	
<b>ETX</b> {c}	

<i>Response Data</i>	
s	Status value.

### 6CC7: Set Random ID Mode

This command enables or disables the Random ID Mode of an EM4237 tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6CC7" <e> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
e	Enable or disable: 0: Disable random ID. 1: Enable random ID.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CC7" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CC8: Change Authentication Level

This command changes the authentication level of an EM4237 tag. It is used to switch from password mode to crypto mode.

<i>Host to RF System Request</i>	
<b>STX</b> "6CC8" <l> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
l	New authentication level: 02h: Level 1 (crypto mode).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CC8" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

## 6CC9: Enhanced Authentication

This command can be used for authentication towards an EM4237 tag. However, it is highly recommended to use one of the convenient high level commands 6CCA/6CCB instead.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CC9" <cc> <oo> <kk> <a> [ <iiiiiiiiiiiiiiii> ] <dd...>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
c	Crypto ID: 03h: Default.
o	Security Options: 80h: Authentication step 1. 90h: Authentication step 2 (no stream encryption). 92h: Authentication step 2 (stream encryption).
k	Key number: 00h: Secure key.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
d	Data to be transmitted.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CC9" <s> <f> <rr...>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
r	Response data.

## 6CCA: Authentication Sequence

This command can be used for authentication towards an EM4237 tag. This is a high level function which manages the whole challenge/response process in one single command. For higher security applications, it is recommended to use function 6CCB instead, to avoid the security key being transferred to the reader and potentially being eavesdropped.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CCA" <mm> <kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
m	Authentication mode: 00h: No stream encryption. 01h: Stream encryption.
k	Secret key (128 bits, 32 hex digits).



a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CCA" <ss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value: 00h: Authentication successful. 01h: Step 2 mismatch. 02h: Step 2 no response. 03h: Secure communication not supported. 04h: Step 1 no response.

### 6CCB: Authentication Sequence using Key/Password Buffer

This command can be used for authentication towards an EM4237 tag. This is a high level function which manages the whole challenge/response process in one single command. In contrast to the previous function, the password is not expected as a parameter but rather taken from the reader's Key/Password Buffer, which can be set using command 6010 (key must be 32 hex digits long) and made permanent using command E002.

<i>Host to RF System Request</i>	
<b>STX</b> "6CCB" <mm> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Authentication mode: 00h: No stream encryption. 01h: Stream encryption.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CCB" <ss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value: 00h: Authentication successful. 01h: Step 2 mismatch. 02h: Step 2 no response. 03h: Secure communication not supported. 04h: Step 1 no response.

## 6CCC: Create IBR Inventory

Builds an inventory of all tags currently present in the field. Instead of the standard anticollision, the EM4237 custom command "Inventory Block Read" (IBR) is used. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

The EM4237 specific custom anticollision options are controlled via the keys ISOS\_SLIS\_INV\_PG (first block to read), ISOS\_SLIS\_INV\_ID (add tag ID at beginning of data) and ISOS\_SLIS\_INV\_R (double data rate) in the ISO settings (these alias names have historical reasons, obviously). For compatibility with the corresponding SLI(-L/S) command, this command is fixed to a size of 4 blocks (16 bytes).

<i>Host to RF System Request</i>	
<b>STX</b> "6CCC" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CCC" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

## 6CCD: Get IBR Inventory

This function will return all data from the current IBR inventory.

<i>Host to RF System Request</i>	
<b>STX</b> "6CCD"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CCD" <ssss> [ [ <iiiiiiiiiiiiiiii> ] <dddddddddddddddddddddddddddddd> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of four memory blocks (start block is selected via ISOS_SLIS_INV_PG).

### 6CCE: Get ID Range from IBR Inventory

To read a range of IDs or one single ID from a previously created IBR inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b> "6CCE" <iiii> [ <1111> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CCE" [ <cccc> ] [ <iiiiiiiiiiiiiiii> ] <dddddddddddddddddddddddddd> [ ... ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of four memory blocks (start block is selected via ISOS_SLIS_INV_PG).

### 6CCF: Create/Get IBR Inventory

This command is a combination of the commands "6CCC" and "6CCD". An IBR inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6CCF" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset.

	'c': Cumulative. 'C': Cumulative without HF reset.
--	---

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CCF" <ee> <ssss> [ [ <iiiiiiiiiiiiiiii> ] <dddddddddddddddddddddddddddddd> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of four memory blocks (start block is selected via ISOS_SLIS_INV_PG).

## 17.6 KSW VarioSens Custom Functions

Please consult the tag data sheet for more information.

### 6CD0: Init

This command deletes the blocks 10h to FFh, sets the customer data area size and stops the RC oscillator. The KSW-VarioSens memory is divided into a temperature and customer data area. The whole user and measurement area is reset. The parameter blocks (00h..0Fh) are not reset.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CD0" <cc> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
c	Number of custom blocks to preserve (00h-EFh).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CD0" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CD1: Set Log Mode

This command writes block 0Ah, containing all log mode information, to the tag. Please see KSW VarioSens datasheet for explanation of data format.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD1" <mmmmmmmm> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Log mode data to be written to block Ah.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD1" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CD2: Set Log Timer

This command writes block 09h, containing all log timer information, to the tag. Please see KSW VarioSens datasheet for explanation of data format.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD2" <tttttttt> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	Log timer data to be written to block 09h.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD2" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CD3: Set Custom

This command sets the signal state of the custom output and the temperature time product.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD3" <s> <i> <tttt> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	

<i>Request Parameters</i>	
s	Output state.
i	TTP sign flag (0: positive; 1: negative).
t	TTP value (Temperature Time Product).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD3" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CD4: Start Log

This command starts the log procedure using the settings of the Init, Set Log Mode, Set Log Timer, and Set Log Custom commands.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD4" <ttttttttt> <a> [ <iiiiiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	Start time in UTC format (coded as 32 bit value like in C language).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD4" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CD5: Get Log State

This command reads all status blocks relevant for the log process from the tag. Please see KSW VarioSens datasheet for explanation of data format.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD5" <a> [ <iiiiiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD5" <s> <f> [ <aaaaaaaa> <bbbbbbbbb> <cccccccc> <dddddddd> <eeeeeeee> <ffffffff> <gggggggg> <hhhhhhhh> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
a	Block 05h.
b	Block 06h.
c	Block 07h.
d	Block 08h.
e	Block 09h.
f	Block 0Ah.
g	Block 0Bh.
h	Block 0Ch.

### 6CD6: Set Passive

This command stops the log procedure of the KSW-VarioSens. The RC oscillator will stop, reducing the power consumption to a minimum. It is advised to use this command after each logging procedure to save battery power.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD6" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD6" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CD7: Get Timer State

This function can be used to obtain the time elapsed since the measurement has been started, and the battery voltage level. Please see KSW VarioSens datasheet for explanation of data format.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD7" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	

a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD7" <s> <f> [ <aaaaaaaa> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
a	Timer state.

### 6CD8: Get State

This command reads all system status blocks from the tag. Please see KSW VarioSens datasheet for explanation of data format.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD8" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD8" <s> <f> [ <aaaaaaaa> <bbbbbbbb> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
a	Block 03h.
b	Block 04h.

### 6CD9: Get One Block

This function can be used to read any of the system blocks 00h-0Fh.

<i>Host to RF System Request</i>	
<b>STX</b> "6CD9" <b> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number (0h-Fh)
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.



<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CD9" <s> <f> [ <ddddddddd> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	32 bits of block data.

### 6CDA: Timer Sync

This command synchronizes the oscillator of the KSW VarioSens with the reader frequency of 13,56MHz. For a correct synchronization the KSW VarioSens has to be started using the Start Log command before.

<i>Host to RF System Request</i>	
<b>STX</b> "6CDA" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CDA" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CDB: Get Voltage

This command gets the battery voltage level of the KSW VarioSens. For a correct battery test the KSW VarioSens has to be started using the Start Log command before. Afterwards, switch off the reader field for 100ms. Then switch on the reader field and send the Get Voltage command. This test needs approx. 100ms and the KSW VarioSens must be out of the reader field. The result of this test can be read out with the Get Timer State command.

<i>Host to RF System Request</i>	
<b>STX</b> "6CDB" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	

<b>STX</b>	"6CDB" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CDC: Set Calibration

This command writes the three calibration data blocks of the KSW VarioSens.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CDC" <b> <cccccccc> <a> [ <iiiiiiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	Block number (1, 2, 3).
c	Calibration block data.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CDC" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CDD: Verify Password

This command is used to login to a VarioSens Tag. You must specify a password level (1, 2, 3) and the appropriate valid password.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CDD" <l> <pppppppp> <a> [ <iiiiiiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
l	Password level.
p	32-bit password.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CDD" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

## 6CDE: Set Password

To change the password for a given password level of a VarioSens tag, use this command.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CDE" <l> <pppppppp> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
l	Password level.
p	32-bit password.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CDE" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

## 6CDF: Verify Buffered Password (Schreiner e-Temp)

This command is used to login to a Schreiner e-Temp Label. The user key is assumed to be stored in the global password buffer (see command "Set Password Buffer") and will be internally converted into the password data required to unlock the e-Temp Label.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CDF" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CDF" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

## 17.7 VarioSens High Level Custom Functions

Please consult the tag data sheet for more information.

## 6CBA: Easy Init

This function combines the commands „Init“, „Set Log Mode“ and „Set Log Timer“. It also reads the calibration data from target tag and converts the actual temperature values.

Example temperature values: 0B72h for 20°C, 0BA4h for 25°C, 0BD6h for 30°C.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CBA" <cc> <m> <bb> <uuuu> <llll> <o> <e> <sssss> <vvvv> <a> [ <iiiiiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
c	Number of custom blocks to preserve (00h-EFh).
m	Log mode (0: internal and external; 1: all values; 2: all values outside limits; 3: extrema outside limits)
b	Battery check interval (00h: no check; xxh: after xx measurements).
u	Upper temperature limit in Kelvin*10 (Centigrade plus 273).
l	Lower temperature limit in Kelvin*10 (Centigrade plus 273).
o	Storage mode (0: don't overwrite; 1: overwrite).
e	Sensor flag (0: internal; 1: external).
s	Standby time; a multiple of 256 seconds.
v	Log interval; a multiple of 2 seconds, 0 is not allowed.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CBA" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value of last command.

## 6CBB: Get Config

This function reads all configuration data which has previously been set using the „Easy Init“ command. The response data looks similar to the Easy Init parameter string.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CBB" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>
<b>ACK</b>
<b>STX</b> "6CBB" <s> <f>

[ <cc> <m> <bb> <uuuu> <llll> <o> <e> <sssss> <iiii> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
c	Number of custom blocks to preserve (00h-EFh).
m	Log mode (0: internal and external; 1: all values; 2: all values outside limits; 3: extrema outside limits)
b	Battery check interval (00h: no check; xxh: after xx measurements).
u	Upper temperature limit in Kelvin/10.
l	Lower temperature limit in Kelvin/10.
o	Storage mode (0: don't overwrite; 1: overwrite).
e	Sensor flag (0: internal; 1: external).
s	Standby time; a multiple of 256 seconds.
i	Log interval; a multiple of 2 seconds.

## 6CBC: Get Progress

This function reads all log progress data from the tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6CBC" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CBC" <s> <f> [ <eeeeeeee> <pp> <rr> <l> <mmmm> <ttttttt> <a> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
e	Start time in UTC format (coded as 32 bit value like in C language).
p	Log pointer, i. e. number of next measurement block. First measurement block is 00h.
r	Number of memory replacements.
l	Log stopped flag (0: Tag did not autonomously stop logging; 1: Tag stopped logging due to low battery or memory full).
m	Number of measurements.
t	Timer count value as a multiple of 64 seconds.
a	Start flag (0: Timer was stopped; 1: Timer is running).

## 6CBE: Read Single Measurement

This command reads one block of logged data from the tag. It automatically determines the active log mode and formats its response depending on it. Warning: it is necessary to call the “Get Config” command once before reading a tag's measurement data. This is required because the “Get Config” function reads and temporarily stores the calibration values required for temperature calculation. Note that if the tag is located in the reader field during measurement, values of 0 are stored instead of actual temperature values.

There is one response format for each log mode. The first response field is always the log mode (0-3), which can be used to dynamically process the remaining response data.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CBE" <mm> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
m	Measurement number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response: Log mode 0</i>	
<b>ACK</b>	
<b>STX</b>	"6CBE" <s> <f> [ <m> <cccc> <iii> <eeee> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
m	Log mode (0).
c	Log counter.
i	Measurement value of internal temperature sensor.
e	Measurement value of external sensor.

<i>RF System to Host Response: Log mode 1</i>	
<b>ACK</b>	
<b>STX</b>	"6CBE" <s> <f> [ <m> <aaaa> <bbbb> <cccc> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
m	Log mode (1).
a	Measurement value 1.
b	Measurement value 2.
c	Measurement value 3.

<i>RF System to Host Response: Log mode 2</i>	
<b>ACK</b>	
<b>STX</b>	"6CBE" <s> <f> [ <m> <cccc> <vvvv> ]

<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
m	Log mode (2).
c	Log counter.
v	Measurement value outside the limits.

<i>RF System to Host Response: Log mode 3</i>	
<b>ACK</b>	
<b>STX</b> "6CBE" <s> <f> [ <m> <cccc> <vvvv> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
m	Log mode (3).
c	Log counter.
v	Measurement value.

### 6CBF: Get Battery Voltage

This command tries to obtain the tag battery voltage. It is possible to either perform a new measurement or to read the most recent value from tag memory (see also: "Easy Init" parameter "Battery check interval"). Important: in measurement mode, it is necessary to start the tag's log mode before sending this command. If there is no measurement data available, 0 is returned. The measurement itself will take approximately 0.5s.

<i>Host to RF System Request</i>	
<b>STX</b> "6CBF" <m> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode ('m': measure now; 'r': return most recent value from memory).
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CBF" <s> <f> [ <vvvv> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
v	Voltage value (millivolts).

## 17.8 ICode SLI-S Custom Functions

The ICode SLI-S supports all ICode SLI functions, plus the SLI-S functions described here.

Please consult the tag data sheet for more information.

### 6CAC: Password Protect EAS

This command enables the password protection for EAS if the EAS password has to be transmitted before with the Set Password command.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CAC" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CAC" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CAD: Write EAS ID

With the command Write EAS ID a new EAS Identifier is stored in the corresponding configuration memory. If EAS is password protected (for Set and Reset EAS) the EAS password has to be transmitted before with the Set Password command.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CAD" <eeee> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
e	16 Bit EAS identifier.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CAD" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.



## 6CAE: Read EPC

This command reads the 96 bit EPC code from an ICode SLI-S.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CAE" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CAE" <s> <f> [ <eeeeeeeeeeeeeeeeeeee> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
e	96 Bit (i. e. 12 Bytes) EPC code.

## 6CE0: Inventory Page Read

This function reads the tag data blocks in a special mode which operates much the same as ISO 15693 anticollision (see also the command description for "Single Anticollision Round"). Instead of the tag ID, specified pages of tag data are transmitted to the host. Optionally, the tag ID can be transmitted together with the data, but in that mode the masked parts of the ID are *not* transmitted. Another option is the data rate used by the tag for transmission, which can be either ISO 15693 compliant or twice as fast (which is not ISO compliant). In Double data rate mode, the system will automatically use ASK; the tag does not support FSK in that mode.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CE0" <r> <i> <pp> <cc> <t> <ll> [ <mm...> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
r	Downlink data rate ('n': normal ISO 15693; 'h': double).
i	Add part of ID to data ('i': add ID; 'd': data only).
p	Start page to read.
c	Number of pages to read minus one.
t	Timeslot switch ('s': 1; 'm': 16).
l	Mask length in bits.
m	Mask data, padded with zeros to byte boundary.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CE0" <s> <f> [ [ <ii...> ] <pp> <dd...> ] [ <s> <f> [ ... ] ]
<b>ETX</b>	{c}

<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
i	Part of tag ID.
p	Protection information byte.
d	Tag data.

### 6CE2: Get Random Number

The Get Random Number command is required to receive a random number from the label IC. The passwords that will be transmitted with the Set Password command have to be calculated with the password and the random number.

<i>Host to RF System Request</i>	
<b>STX</b> "6CE2" <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CE2" <s> <f> [pppp]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	16 bit random number.

### 6CE3: Set Password

With the Set Password command the different passwords can be transmitted to the label to get access to the different protected functionalities on the following commands. The Set Password command has to be executed just once for the related passwords if the label remains powered.

This command works only in addressed or selected mode; non-addressed mode is not supported.

The encrypted password is password[31:0] XOR { random[15:0], random[15:0] }.

<i>Host to RF System Request</i>	
<b>STX</b> "6CE3" <nn> <pppppppp> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
n	Password identifier: 01h: Read. 02h: Write.

	04h: Privacy. 08h: Destroy. 10h: EAS.
p	32 bit Password, encrypted using random number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CE3" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CE4: Write Password

With the Write Password command a new password will be written into the related memory, if the related old password has already been transmitted with a Set Password command before and the addressed password is not locked.

This command works only in addressed or selected mode; non-addressed mode is not supported.

<i>Host to RF System Request</i>	
<b>STX</b> "6CE4" <nn> <pppppppp> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
n	Password identifier: 01h: Read. 02h: Write. 04h: Privacy. 08h: Destroy. 10h: EAS.
p	32 bit Password.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CE4" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CE5: Lock Password

With the Lock Password command the addressed password will be locked, if the related password

has already been transmitted with a Set Password command before. A locked password cannot be changed any longer.

<i>Host to RF System Request</i>	
<b>STX</b> "6CE5" <nn> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
n	Password identifier: 01h: Read. 02h: Write. 04h: Privacy. 08h: Destroy. 10h: EAS.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CE5" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

## 6CE6: Protect Page

With the Page Protection command the page protection can be changed. One page equals four blocks, which makes a total of 16 bytes.

<i>Host to RF System Request</i>	
<b>STX</b> "6CE6" <nn> <pp> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
n	Page number.
p	Protection status: 01h: Public. 02h: Read and write protected by the read password. 10h: Write protected by the Write password. 11h: Read protected by the Read password and write protected by the Write password.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CE6" <s> <f>	
<b>ETX</b> {c}	
<i>Response Data</i>	

s	Status value.
f	Data flag ('n': no data; 'y': data follows).

### 6CE7: Lock Page Protection

With the Lock Page Protection command the status of the page protection of the related page will be locked if the related passwords (Read an/or Write password) have been transmitted before with the Set Password command. If the page is public no password is required.

<i>Host to RF System Request</i>	
<b>STX</b> "6CE7" <pp> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	Page number.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CE7" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CE8: Get Protection Status

To this command the label will respond with the block protection status of the requested blocks.

<i>Host to RF System Request</i>	
<b>STX</b> "6CE8" <pp> <cc> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	Start page.
c	Number of pages minus one.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CE8" <s> <f> [ <pp> [ <pp> [ ... ] ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	Protection status: Bit #0: Block is locked.

	Bit #1: Block is read password protected. Bit #2: Block is write password protected.
--	---

### 6CE9: Destroy

With the Destroy command the ICode SLI-S label IC can be destroyed if the Destroy Password has been transmitted before. This command is irreversible and the ICode SLI-S will never respond to any command again (ISO and EPC commands).

<i>Host to RF System Request</i>	
<b>STX</b>	"6CE9" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CE9" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 6CEA: Enable Privacy

With the Enable Privacy command the ICode SLI-S label IC can be set into the Privacy mode. The ICode SLI-S will not respond to any command except Get Random Number and Set Password. To get out of the Privacy Status the valid Privacy password has to be transmitted to the IC with the Set Password command.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CEA" <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CEA" <s> <f>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

## 6CEB: 64 Bit Password Protection

With the 64 Bit Password Protection command the label IC can be instructed that the Read and Write passwords are required to get access to password protected blocks (pages). This mode can be enabled or disabled if the Read and Write passwords have already been transmitted with a Set Password command before.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CEB" <pp> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
p	64 bit Password parameter.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CEB" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

## 17.9 ICode SLI-S High Level Custom Functions

Please consult the tag data sheet for more information.

### 6CAF: Login

This function combines the low level commands "Get Random Number" and "Set Password". The random number is read from the tag and then used to encode the password provided by the user. Please note that if this function is used, the password is not coded on its way between host and reader and thus may be intercepted by others.

This command works only in addressed or selected mode; non-addressed mode is not supported.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CAF" <nn> <pppppppp> <a> [ <iiiiiiiiiiiiiiii> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
n	Password identifier: 01h: Read. 02h: Write. 03h: Privacy. 04h: Destroy. 05h: EAS.
p	32 bit Password, encrypted using random number.

a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CAF" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6CEC: Create IPR Inventory

Builds an inventory of all tags currently present in the field. Instead of the standard anticollision, the ICode SLI-S custom command "Inventory Page Read" (IPR) is used. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

The ICode SLI-S specific custom anticollision options are controlled via the keys ISOS\_SLIS\_INV\_PG (page number to read), ISOS\_SLIS\_INV\_ID (add tag ID at beginning of data) and ISOS\_SLIS\_INV\_R (double data rate) in the ISO settings. Please note that only one page of data (16 bytes) can be read during one anticollision sequence if this command is used.

<i>Host to RF System Request</i>	
<b>STX</b> "6CEC" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CEC" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.



### 6CED: Get IPR Inventory

This function will return all data from the current IPR inventory.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CED"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CED" <ssss> [ [ <iiiiiiiiiiiiiiii> ] <div style="margin-left: 150px;">&lt;dddddddddddddddddddddddddddddd&gt;</div> <div style="margin-left: 150px;">[ ... ] ]</div>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of memory page (which is selected via ISOS_SLIS_INV_PG).

### 6CEE: Get ID Range from IPR Inventory

To read a range of IDs or one single ID from a previously created IPR inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b>	"6CEE" <iiii> [ <1111> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6CEE" [ <cccc> ] [ <iiiiiiiiiiiiiiii> ] <div style="margin-left: 150px;">&lt;dddddddddddddddddddddddddddddd&gt; [ ... ]</div>
<b>ETX</b>	{c}
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of memory page (which is selected via ISOS_SLIS_INV_PG).

### 6CEF: Create/Get IPR Inventory

This command is a combination of the commands "6CEC" and "6CED". An IPR inventory is built

and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6CEF" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CEF" <ee> <ssss> [ [ <iiiiiiiiiiiiiiii> ] <dddddddddddddddddddddddddddddddd> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit) in reverse byte order (only present if ISOS_SLIS_INV_ID is 0001).
d	Data of memory page (which is selected via ISOS_SLIS_INV_PG).

## 17.10 ScemTag Sensor Transponder Custom Functions

Please consult the tag data sheet for more information.

### 6CF0: Single Measurement

Perform a single measurement using specified sensor. Please refer to tag's datasheet about available sensor types and sensor numbers. Temperature values are returned in Kelvin\*10 (Kelvin is equal to Centigrade plus 273); humidity values are returned in percent\*10.

<i>Host to RF System Request</i>	
<b>STX</b> "6CF0" <t> <s> <a> [ <iiiiiiiiiiiiiiii> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	Sensor type: 'a': Analogue sensor attached directly to transponder ASIC.

	'd': Digital ( $\mu$ C-powered) sensor attached via SPI.
s	Sensor number; in theory sensors 0, 1 and 2 can be connected.
a	Addressed mode ('n': non-addressed; 'a': addressed; 's': selected).
i	Tag ID (64 bit) in reverse byte order.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6CF0" <s> <f> [ <dddd> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Log value.

## 18 ICode EPC/UID

### 18.1 Features

Although the EPC and UID tags use a quite similar air protocol, there is a complete set of STX/ETX commands for each of them.

#### EPC:

- Read/Write transponder (OTP) 13,56 MHz
- 96 Bit EPC
- 16 Bit CRC
- 24 Bit Destroy Code

#### UID:

- Read/Write transponder 13,56 MHz
- 96 Bit User Data
- 16 Bit User CRC
- 40 Bit UID (manufacturer programmed, read only)
- 16 Bit UID CRC
- 24 Bit Destroy Code

### 18.2 EPC Low Level Functions

These commands are basically those offered by the transponder air protocol.

#### **6A80: Single Round**

Performs a single EPC anticollision round.

<i>Host to RF System Request</i>	
<b>STX</b> "6A80" <tt> <hh>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	Timeslot number key: 00: 1 timeslot. 01: 4 timeslots. 03: 8 timeslots. 07: 16 timeslots. 0F: 32 timeslots. 1F: 64 timeslots. 3F: 128 timeslots. 7F: 256 timeslots. FF: 512 timeslots.
h	Hash value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A80" <f> <s> <d> [ <eeeeeeeeeeeeeeeeeeee> <cccc> ] [ <s> <d> [ <eeeeeeeeeeeeeeeeeeee> <cccc> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
f	Slot f ('e': empty – no halted tags present; 'f': halted tags found).
s	Status value.
d	Data flag ('n': no data; 'y': data follows).
e	EPC code.
c	CRC16.

### 5A80: Write Block

Writes a single block (one byte) to an EPC transponder.

<i>Host to RF System Request</i>	
<b>STX</b> "5A80" <bb> <dd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
d	Block data.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5A80"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

### 6A84: Destroy

Puts an EPC tag into "destroyed" mode.

<i>Host to RF System Request</i>	
<b>STX</b> "6A84" <eeeeeeeeeeeeeeeeeeee> <dddddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
e	EPC code.
d	Destroy code.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6A84"	

<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 18.3 UID Low Level Functions

These commands are basically those offered by the transponder air protocol.

#### 6A90 Single Round

Performs a single UID anticollision round.

<i>Host to RF System Request</i>	
<b>STX</b>	"6A90" <tt>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
t	Timeslot number key: 00: 1 timeslot. 01: 4 timeslots. 03: 8 timeslots. 07: 16 timeslots. 0F: 32 timeslots. 1F: 64 timeslots. 3F: 128 timeslots. 7F: 256 timeslots. FF: 512 timeslots.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6A90" <f> <s> <d> [ <uuuuuuuuuuuuuuuuuuuuuuuu> <cccc> <nnnnnnnnnn> <iiii> ] [ <s> <d> [ <uuuuuuuuuuuuuuuuuuuuuuuu> <cccc> <nnnnnnnnnn> <iiii> ] [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
f	Slot f ('e': empty – no halted tags present; 'f': halted tags found).
s	Status value.
d	Data flag ('n': no data; 'y': data follows).
u	User data.
c	Data CRC16.
n	Tag ID.
i	ID CRC16

### 5A90: Write Block

Writes a single block (one byte) to a UID transponder.

<i>Host to RF System Request</i>	
<b>STX</b>	"5A90" <bb> <dd>
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
b	Block number.
d	Block data.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5A90"
<b>ETX</b>	{ c }
<i>Response Data</i>	
-	None.

### 6A94: Destroy

Puts a UID tag into "destroyed" mode.

<i>Host to RF System Request</i>	
<b>STX</b>	"6A94" <uuuuuuuuuuuuuuuuuuuuuuuu> <cccc> <nnnnnnnnnn> <dddddd>
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
u	User data.
c	Data CRC16.
n	Tag ID.
d	Destroy code.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6A94"
<b>ETX</b>	{ c }
<i>Response Data</i>	
-	None.

## 18.4 EPC High Level Functions

These commands combine and/or simplify some of the low level commands.

### 6AA0: Create Inventory

Builds an inventory of all tags currently present in the field by their serial number. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously

created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "6AA0" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6AA0" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

### 6AA1: Get Inventory

This function will return all serial numbers from the current inventory.

<i>Host to RF System Request</i>	
<b>STX</b> "6AA1"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6AA1" <ssss> [ <eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee<cccc> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
e	EPC code.
c	CRC16.



## 6AA2: Get ID Range from Inventory

To read a range of IDs or one single ID from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b> "6AA2" <iiii> [ <l111> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6AA2" [ <nnnn> ] [ <eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
n	Number of tag IDs that follow (only present if a number of IDs was specified).
e	EPC code.
c	CRC16.

## 6AA3: Realtime Inventory

Builds an inventory of all tags currently present in the field by their serial number. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

In contrast to the “create” inventory functions, this command sends tag IDs “on the fly”, i. e. as soon as they are detected, in multiple responses. The last response is always empty (0 IDs) and contains accumulated status flags. When the last response has been transmitted, the device is ready for the next command.

<i>Host to RF System Request</i>	
<b>STX</b> "6AA3" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6AA3" <ww> <ssss> [ <eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee <cccc> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	

w	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
e	EPC code.
c	CRC16.

#### 6AA4: Create/Get Inventory

This command is a combination of the commands "6AA0" and "6AA1". An inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6AA4" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6AA4" <ww> <ssss> [ <eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee> <cccc> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
w	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
e	EPC code.
c	CRC16.

#### 5AA0: Write User Data

This command writes the complete OTP memory of an EPC tag, calculating the CRC16 as well.

<i>Host to RF System Request</i>	
<b>STX</b>	"5AA0" <eeeeeeeeeeeeeeeeeeee>
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
e	User data (EPC code) to be written.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5AA0" <s>
<b>ETX</b>	{ c }
<i>Response Data</i>	
s	Status value.

### 5AA1: Write Destroy Code

This command writes a destroy code to an EPC transponder.

<i>Host to RF System Request</i>	
<b>STX</b>	"5AA1" <cccccc>
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
c	Destroy code.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5AA1"
<b>ETX</b>	{ c }
<i>Response Data</i>	
-	None.

### 4AA8: Looped Read

Continuously repeats a single-timeslot inventory attempt until a tag has been detected. This command does not support the anticollision feature. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b>	"4AA8"
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4AA8" <s> <f> <eeeeeeeeeeeeeeeeeeee> <cccc>
<b>ETX</b>	{ c }

<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
e	EPC code.
c	CRC16.

## 18.5 UID High Level Functions

These commands combine and/or simplify some of the low level commands.

### 6AB0: Create Inventory

Builds an inventory of all tags currently present in the field by their serial number. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "6AB0" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6AB0" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

### 6AB1: Get Inventory

This function will return all serial numbers from the current inventory.

<i>Host to RF System Request</i>
----------------------------------

<b>STX</b>	"6AB1"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6AB1" <ssss> [ <uuuuuuuuuuuuuuuuuuuuuu> <cccc> <nnnnnnnnnn> <iiii> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags in reader memory.
u	User data.
c	Data CRC16.
n	Tag ID.
i	ID CRC16.

## 6AB2: Get ID Range from Inventory

To read a range of IDs or one single ID from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b>	"6AB2" <iiii> [ <llll> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6AB2" [ <ssss> ] [ <uuuuuuuuuuuuuuuuuuuuuu> <cccc> <nnnnnnnnnn> <iiii> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Number of tag IDs that follow (only present if a number of IDs was specified).
u	User data.
c	Data CRC16.
n	Tag ID.
i	ID CRC16.

## 6AB3: Realtime Inventory

Builds an inventory of all tags currently present in the field by their serial number. This function is

breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

In contrast to the “create” inventory functions, this command sends tag IDs “on the fly”, i. e. as soon as they are detected, in multiple responses. The last response is always empty (0 IDs) and contains accumulated status flags. When the last response has been transmitted, the device is ready for the next command.

<i>Host to RF System Request</i>	
<b>STX</b> "6AB3" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6AB3" <ee> <ssss> [ <uuuuuuuuuuuuuuuuuuuuuu> <cccc> <nnnnnnnnnn> <iiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
u	User data.
c	Data CRC16.
n	Tag ID.
i	ID CRC16.

#### 6AB4: Create/Get Inventory

This command is a combination of the commands "6AB0" and "6AB1". An inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6AB4" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single.

	'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.
--	--

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6AB4" <ee> <ssss> [ <uuuuuuuuuuuuuuuuuuuuuuuu> <cccc> <nnnnnnnnnn> <iiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
u	User data.
c	Data CRC16.
n	Tag ID.
i	ID CRC16.

### 5AB0: Write User Data

This command writes the complete user memory of a UID tag, calculating the CRC16 as well.

<i>Host to RF System Request</i>	
<b>STX</b> "5AB0" <dddddddddddddddddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
d	User data to be written.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5AB0" <s> <b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 5AB1: Write Destroy Code

This command writes a destroy code to a UID transponder.

<i>Host to RF System Request</i>	
<b>STX</b> "5AB1" <cccccc>	

<b>ETX</b>	{ c }
<i>Request Parameters</i>	
c	Destroy code.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5AB1"	
<b>ETX</b> { c }	
<i>Response Data</i>	
-	None.

#### 4AB8: Looped Read

Continuously repeats a single-timeslot inventory attempt until a tag has been detected. This command does not support the anticollision feature. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "4AB8"	
<b>ETX</b> { c }	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4AB8" <s> <f> <uuuuuuuuuuuuuuuuuuuuuu> <cccc> <nnnnnnnnnn> <iiii>	
<b>ETX</b> { c }	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
u	User data.
c	Data CRC16.
n	Tag ID.
i	ID CRC16.

## 18.6 Settings Functions

These commands are used to view or edit EPC/UID specific settings.

#### 1A32: Request Protocol Specific Setting

This function is used to retrieve the value of a certain EPC/UID setting. Numeric values are unsigned 16 bit hex values; switch values are "0000" for "false" or "0001" for "true".

<i>Host to RF System Request</i>	
----------------------------------	--



<b>STX</b>	"1A32" <kk>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1A32" <vvvv>
<b>ETX</b>	{c}
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Numeric/Switch, Default, Description):

Key	C	Alias	N/S	Def.	Description
01		EUS_RETRIES	n	0000	Retries on read and inventory functions.
02		EUS_CRC_NOCHECK	n	0000	CRC check level (0=all; 1=EPC/UID; 2=none).
03		EUS_CRC_MAXREPEAT	n	000A	Halt tag after having read the same ID n times.
04		EUS_INV_RESET_S	s	0000	Inventory: perform HF reset at start.
05		EUS_INV_RESET_E	s	0001	Inventory: perform HF reset at end.
06		EUS_SYNC_TS	n	0007	Fixed timeslot value to be used in SYNC mode.
07		EUS_SYNC_DUMMYCRC	n	0000	Dummy CRC value to be used in SYNC mode.
08		EUS_HASH_START	n	0008	EPC start hash value.
09		EUS_VERIFY	s	0000	Verify mode, i. e. automatic "read after write".

### 1A33: Edit Protocol Specific Setting

Used to change the value of a certain ISO-Tag setting. See "Request Setting" ("1A32") for more details.

<i>Host to RF System Request</i>	
<b>STX</b>	"1A33" <kk> <vvvv>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	Key.
v	Value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1A33"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 1A34: Get ID Mask

This command reads a currently set EPC/UID Mask from the reader.

<i>Host to RF System Request</i>	
<b>STX</b> "1A34"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1A34" <bb> <mm> [ <mm> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
b	Current mask size in bits.
m	Mask bytes.

### 1A35: Set ID Mask

To set a new EPC/UID Mask, use this command.

<i>Host to RF System Request</i>	
<b>STX</b> "1A35" <bb> <mm> [ <mm> [ ... ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	New mask size in bits.
m	Mask bytes.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1A35"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

## 19 SC-433

### 19.1 Features

- Active read/write transponder 125kHz upstream / 433MHz downstream
- 32 bit (4 byte) UID address
- User memory 62 blocks of 4 bytes size (System memory 2 blocks)
- Anticollision

### 19.2 Low Level Functions

These commands are basically those offered by the transponder air protocol.

#### 6F10: Single Anticollision Round

This command performs a complete Inventory cycle. The response always contains all 4 timeslots. It is strongly recommended that you use the high level command "Create Inventory" (6F1C) instead, unless you know exactly what you are doing.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F10" <ll> [ <mm...> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
l	Mask length in bits.
m	Mask (rounded up to byte boundary, max. 64 bits / 8 bytes).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F10" <s> <f> [ <iiiiiiii> ]
	<s> <f> [ <iiiiiiii> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
i	Tag ID (32 bit).

#### 6F12: Select

To put a single tag into "selected" state, use this command.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F12" <iiiiiiii>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Tag ID (32 bit).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F12" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

#### 4F10: Read Single Block

This command allows for reading a single block from an SC-433 tag. Only a selected tag will respond.

<i>Host to RF System Request</i>	
<b>STX</b> "4F10" <bb>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4F10" <s> <f> [ <dddddddd> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Block data.

#### 4F12: Read Multiple Blocks

This command allows for reading several blocks at once from an SC-433 tag. Only a selected tag will respond.

<i>Host to RF System Request</i>	
<b>STX</b> "4F12" <bb> <cc>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number.
c	Number of blocks minus one.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4F12" <s> <f> [ <dddddddd> [ <s> <f> <dddddddd> [ ... ] ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

f	Data flag ('n': no data; 'y': data follows).
d	Block data.

#### 4F16: Get Version

This command returns version information on the selected tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"4F16"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4F16" <s> <f> [ <aa> <vv> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
a	Application ID.
v	Version.

#### 4F17: Get Size

This command returns memory information on the selected tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"4F17"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4F17" <s> <f> [ <bb> <nn> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
b	Bytes per block minus one.
n	Number of available blocks minus one.

#### 4F1A: Get Battery Status

This command returns the tag's battery status.

<i>Host to RF System Request</i>	
<b>STX</b>	"4F1A"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4F1A" <s> <f> [ <bb> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
b	Battery status (00: empty; 01: full).

### 5F10: Write Single Block

This function can be used to write a single block of data. Only a selected tag will be written to.

<i>Host to RF System Request</i>	
<b>STX</b>	"5F10" <bb> <ddddddd>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	Block number.
d	Block data.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5F10" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

### 5F18: Set ID

This function can be used to set the selected tag's UID.

<i>Host to RF System Request</i>	
<b>STX</b>	"5F18" <pppp> <iiiiiii>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
p	Password (16 bit).
i	New UID (32 bit).

<i>RF System to Host Response</i>	
-----------------------------------	--

<b>ACK</b>	
<b>STX</b> "5F18" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 5F19: Set Password

This function can be used to set the selected tag's password.

<i>Host to RF System Request</i>	
<b>STX</b> "5F19" <oooo> <nnnn>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
o	Old password.
n	New password.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5F19" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 5F1A: Set Defaults

Resets all tags currently in the field to their default ID and password. USE WITH CAUTION!

<i>Host to RF System Request</i>	
<b>STX</b> "5F1A"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5F1A" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6F18: Quiet

Use this function to put the selected tag into QUIET mode. The tag will not respond to any anticollision command afterwards, until it receives an RESET or a SELECT command.

<i>Host to RF System Request</i>	
<b>STX</b> "6F18"	
<b>ETX</b> { C }	
<i>Request Parameters</i>	
–	None.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F18" <s>	
<b>ETX</b> { C }	
<i>Response Data</i>	
s	Status value.

### 6F19: Sleep

Use this function to put the selected tag into SLEEP mode. The tag will enter its low power mode.

<i>Host to RF System Request</i>	
<b>STX</b> "6F19"	
<b>ETX</b> { C }	
<i>Request Parameters</i>	
–	None.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F19" <s>	
<b>ETX</b> { C }	
<i>Response Data</i>	
s	Status value.

### 6F1A: Reset

To reset all tags in the field, use this command. Any QUIET condition will be removed.

<i>Host to RF System Request</i>	
<b>STX</b> "6F1A"	
<b>ETX</b> { C }	
<i>Request Parameters</i>	
–	None.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F1A"	
<b>ETX</b> { C }	
<i>Response Data</i>	
–	None.



## 6F1B: Erase Tag Data

This function can be used to initialize all data blocks. Only a selected tag will be erased.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F1B"
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F1B" <s>
<b>ETX</b>	{ c }
<i>Response Data</i>	
s	Status value.

## 19.3 High Level Functions

These commands combine and/or simplify some of the low level commands.

### 6F1C: Create Inventory

Builds an inventory of all tags currently present in the field by their serial number. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F1C" <m>
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
m	Inventory mode: 's'/'S': Single. 'c'/'C': Cumulative.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F1C" <ee> <ssss>
<b>ETX</b>	{ c }
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found.

	10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

### 6F1D: Get Inventory

This function will return all serial numbers from the current inventory.

<i>Host to RF System Request</i>	
<b>STX</b> "6F1D"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F1D" <ssss> [ <iiiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (32 bit).

### 6F1E: Get ID Range from Inventory

To read a range of IDs or one single ID from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b> "6F1E" <iiii> [ <1111> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F1E" [ <cccc> ] [ <iiiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
i	Tag ID (32 bit).

### 6F1F: Create/Get Inventory

This command is a combination of the commands "6F1C" and "6F1D". An inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6F1F" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's'/'S': Single. 'c'/'C': Cumulative.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F1F" <ee> <ssss> [ <iiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (32 bit).

#### 6F14: Create Inventory (Compatibility Mode)

Builds an inventory of all tags currently present in the field by their serial number. This command is used for the so-called "compatibility mode" of the tag. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
<b>STX</b> "6F14" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's'/'S': Single. 'c'/'C': Cumulative.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F14" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow.

	02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

### 6F15: Get Inventory (Compatibility Mode)

This function will return all serial numbers from the current inventory. It is used for the so-called “compatibility mode” of the tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F15"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F15" <ssss> [ <iiiiiiii> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (32 bit).

### 6F16: Get ID Range from Inventory (Compatibility Mode)

To read a range of IDs or one single ID from a previously created inventory, this function can be used. It is used for the so-called “compatibility mode” of the tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F16" <iiii> [ <1111> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F16" [ <cccc> ] [ <iiiiiiii> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
i	Tag ID (32 bit).

### 6F17: Create/Get Inventory (Compatibility Mode)

This command is a combination of the commands "6F14" and "6F15". An inventory is built and instantly transmitted to the host. This command is used for the so-called "compatibility mode" of the tag.

<i>Host to RF System Request</i>	
<b>STX</b> "6F17" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's'/'S': Single. 'c'/'C': Cumulative.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F17" <ee> <ssss> [ <iiiiiiii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (32 bit).

## 19.4 Settings Functions

These commands are used to view or edit SC-433 specific settings.

### 1F10: Request Protocol Specific Setting

This function is used to retrieve the value of a certain SC-433 setting. Numeric values are unsigned 16 bit hex values; switch values are "0000" for "false" or "0001" for "true".

<i>Host to RF System Request</i>	
<b>STX</b> "1F10" <kk>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1F10" <vvvv>	

<b>ETX</b> {c}	
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Numeric/Switch, Default, Description):

Key	C	Alias	N/S	Def.	Description
01		SCS_RETRIES	n	0000	Retries on read and inventory functions.
02		SCS_COLL_DEPTH	n	0000	Anticollision depth limit (0=none).
03		SCS_DATA_RATE	s	0000	Data rate reader->tag (0=low; 1=high).
04		SCS_SAFE_INV	s	0001	Assume collision on detected edge.
05		SCS_COMP_ADVANCED	n	0001	Select advanced Wakeup patterns (0-2).
06		SCS_COMP_WLOOP	n	0000	Repetitions of Wakeup pattern.

### 1F11: Edit Protocol Specific Setting

Used to change the value of a certain SC-433 setting. See "Request Setting" ("1F10") for more details.

<i>Host to RF System Request</i>	
<b>STX</b> "1F11" <kk> <vvvv>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key.
v	Value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1F11"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

## 20 ISO 11784 / ISO 11785 (FDX-B) Transponder

### 20.1 Features

- Read only transponder
- Continuous data output (Transponder Talks First)
- Data frequency = RF/64

### 20.2 Transponder Recognition

#### 3D00: ISO 11784 / 11785 (FDX-B) transponder Recognition

This function determines in Single mode if there is a ISO 11784 / 11785 (FDX-B) transponder transponder in reading distance to the antenna or not.

<i>Host to RF system</i>	
<b>STX</b>	"3D00" <m>
<b>ETX</b>	{c}
m	Mode: at the moment only „S“ = single mode.

  

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"3D00" <xx>
<b>ETX</b>	{c}
x	„00“ = no ISO 11784 / 11785 (FDX-B) transponder transponder in reading distance, „01“ = transponder was recognized.

### 20.3 Transponder Read Functions

#### 4D00: Read ISO 11784 / 11785 (FDX-B) transponder

<i>Host to RF system</i>	
<b>STX</b>	"4D00" <m>
<b>ETX</b>	{c}
m	Read mode: at the moment only „S“ = single mode.

  

<i>RF system to Host</i>	
<b>ACK</b>	
<b>STX</b>	"4D00" <iiiiiiiiiii> <cccc> <dd> <rrrr> <aa> <eeeeee>
<b>ETX</b>	{c}
i	National identification code (38 Bit)
c	Country code according to ISO 3166 (10 Bit)

d	Data block flag (1 Bit) “01” Data block following “00” no Data block following
r	Reserved field (14 Bit)
a	Animal flag (1 Bit) “00” non-animal application “01” animal application
e	Extended block



## 21 EPC Class 1 Gen.2

### 21.1 Features

Since there are various different flavours of EPC Gen.2 tags (e. g. different memory size/arrangement, special features etc.), it is strongly recommended to consult the transponder datasheet for further information.

### 21.2 Low Level Functions

These commands are basically those offered by the transponder air protocol.

Currently, there are no low level functions available for public.

### 21.3 High Level Functions

There are several high level functions defined for easy and comfortable use of the reader device without operating at the low level of the EPCglobal Class 1 Generation 2 protocol.

#### 4F38: Read Multiple Blocks

Inventory a tag and read one or more 16 bit blocks from its memory.

<i>Host to RF System Request</i>	
<b>STX</b> "4F38" <a> <m> <ssss> <cc> [ <ii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a':addressed; 's':selected).
m	Memory bank (1: EPC; 2: TID; 3: User).
s	Start block.
c	Number of blocks to be read minus one.
i	EPC of transponder.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4F38" <s> <f> [ <dddd> [ <s> <f> [ <dddd> [ ... ] ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('y': data follows; 'n': no data).
d	Tag data.

#### 5F38: Write Single Block

Inventory a tag and write a single 16 bit block to its memory.

<i>Host to RF System Request</i>	
<b>STX</b> "5F38" <a> <m> <ssss> <dddd> [ <ii> [ ... ] ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a':addressed; 's':selected).
m	Memory bank (1: EPC; 2: TID; 3: User).
s	Block number.
d	Data to be written.
i	EPC of transponder
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5F38" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6F3C: Create Inventory

Builds an inventory of all tags currently present in the field by their serial number. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

The parameters q, s, l and t are currently not supported by all devices and their use is generally discouraged.

<i>Host to RF System Request</i>	
<b>STX</b> "6F3C" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'c': Cumulative.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F3C" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: OK. 01: Inventory overflow.
s	Current inventory size, i. e. the number of tags found.

### 6F3D: Get Inventory

This function will return all PC and EPC bundles from the current inventory. According to EPC Class 1 Gen.2 specification, bits #10-14 of the PC (Protocol Control) word contain the EPC length in words. This may be important when parsing the response in case of different length EPCs.

The most common EPC length is 96 bits, i. e. 12 bytes.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F3D"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F3D" <ssss> [ <pppp> <eeeeeeeeeeeeeeee...> [ ... ] ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
p	PC (Protocol Control) word.
e	EPC.

### 6F3E: Get Single ID from Inventory

To read one single PC and EPC bundle from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F3E" <iiii>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F3E" <pppp> <eeeeeeeeeeeeeeee...>
<b>ETX</b>	{c}
<i>Response Data</i>	
p	PC (Protocol Control) word.
e	EPC.

### 6F3F: Create/Get Inventory

This command is a combination of the commands "6F3C" and "6F3D". An inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6F3F" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'c': Cumulative.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F3F" <ee> <ssss> [ <pppp> <eeeeeeeeeeeeeeeeeeee...> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow.
s	Current inventory size, i. e. the number of tags found.
p	PC (Protocol Control) word.
e	EPC.

## 21.4 Settings Functions

These commands are used to view or edit EPC Class 1 Gen.2 specific settings.

### 1F30: Request Protocol Specific Setting

Read a protocol specific setting given by a key as defined in table above.

<i>Host to RF System Request</i>	
<b>STX</b> "1F30" <kk>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1F30" <vvvv>	
<b>ETX</b> {c}	
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Numeric/Switch, Default, Description):

*This table is considered preliminary and some parameters don't apply to all devices supporting EPC Class 1 Gen.2!*

Key	C	Alias	N/S	Def.	Description
01		G2S_TAG_PROFILE	n	0004	Communication Parameters 0: 320 kHz, Miller 4, Tari 25 µs, ASK 1: 320 kHz, Miller 4, Tari 25 µs, PR-ASK (DRM) 2: 320 kHz, Miller 8, Tari 25 µs, ASK 3: 320 kHz, Miller 8, Tari 25 µs, PR-ASK (DRM) 4: 160 kHz, Miller 4, Tari 25 µs, ASK 5: 160 kHz, Miller 8, Tari 25 µs, ASK 6: 40 kHz, FM0, Tari 25 µs, ASK 7: 40 kHz, Miller 2, Tari 25 µs, ASK 8: 40 kHz, Miller 4, Tari 25 µs, ASK
02		G2S_HOPCHANNELS	n	00FF	0xFF for LBT and automatic channel selection 0x01 to 0x0F for a fix channel
...					
07		G2S_ASK_MOD_DEPTH	n	005A	ASK modulation depth between 80% and 100%
...					
18		G2S_DEF_Q_START	n	0003	Start value for Q used if not given as parameter
19		G2S_Q_MAX	n	0005	Upper bound for Q value
1A		G2S_Q_MIN	n	0002	Lower bound for Q value
1B		G2S_DEF_SESSION	n	0000	Session used if not given as parameter
1C		G2S_DEF_SELECT	n	0000	Select used if not given as parameter
1D		G2S_DEF_TARGET	n	0000	Target used if not given as parameter
1E		G2S_RETRIES	n	0000	Retries on read and inventory functions.

### 1F31: Edit Protocol Specific Setting

Write a protocol and device specific setting like modulation index or range for the Q-value.

<i>Host to RF System Request</i>	
<b>STX</b>	"1F31" <kk> <vvvv>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	Key (see table).
v	Value.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1F31"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

## 22 EPC Class 1 Gen.2 Custom

### 22.1 ScemTag Sensor Transponder Custom Functions

#### 6F60: Single Measurement

Perform a single measurement using specified sensor. Please refer to tag's datasheet about available sensor types and sensor numbers. Temperature values are returned in Kelvin\*10 (Kelvin is equal to Centigrade plus 273); humidity values are returned in percent\*10.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F60" <t> <s> <a> [ <eeeeeeeeeeeeeeeeeeee> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
t	Sensor type: 'a': Analogue sensor attached directly to transponder ASIC. 'd': Digital ( $\mu$ C-powered) sensor attached via SPI.
s	Sensor number; in theory sensors 0, 1 and 2 can be connected.
a	Addressed mode ('n': non-addressed; 'a': addressed).
e	Tag EPC (96 bit).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F60" <s> <f> [ <dddd> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Measurement value.

### 22.2 IDS SL900A Based Sensortransponder Custom Functions

#### 6F70: Single Measurement

Perform a single measurement using specified sensor. Please refer to tag's datasheet about available sensor types and sensor numbers. Temperature values are returned in Kelvin\*10 (Kelvin is equal to Centigrade plus 273); Measurements from external inputs are returned in millivolts.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F70" <t> <s> <a> [ <eeee...> ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
t	Sensor type: 'a': Analogue sensor attached directly to transponder ASIC.

	'd': Digital ( $\mu$ C-powered) sensor attached via SPI.
s	Sensor number.
a	Addressed mode ('n': non-addressed; 'a': addressed).
e	Tag EPC.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F70" <s> <f> [ <dddd> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Measurement value.

### 6F73: Configure Log

This command will configure the logging mode of the Transponder.

<i>Host to RF System Request</i>	
<b>STX</b> "6F73" <m> <o> <ssss> <iiii> <a> [ <eeee...> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Logging Mode ('0': Normal; '1': Out of Limits; '2': Boundary Crossing).
o	Storage Rule ('0': Stop; '1': Overwrite).
s	Binary coded sensor selection: 0001: internal Temperature sensor. 0002: external input 1. 0004: external input 2. 0008: Battery voltage.
i	Interval in seconds.
a	Addressed mode ('n': non-addressed; 'a': addressed).
e	Tag EPC.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F73" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6F74: Initialize Log

The Initialize Function will reset all internal counters. This will destroy the current log.

<i>Host to RF System Request</i>	
<b>STX</b> "6F74" <dddd> <a> [ <eeee...> ]	

<b>ETX</b> {c}	
<i>Request Parameters</i>	
d	Delay between Start command and first measurement in seconds.
a	Addressed mode ('n': non-addressed; 'a': addressed).
e	Tag EPC.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F74" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6F75: Start Log

This function will start the log. The log should be initialized with function 6F74 unless the new measurements should appended to the previous log. The timestamp has to be omitted if the device has a real time clock.

<i>Host to RF System Request</i>	
<b>STX</b> "6F75" [ <tttttttt> ] <a> [ <eeee...> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
t	UTC Timestamp.
a	Addressed mode ('n': non-addressed; 'a': addressed).
e	Tag EPC.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F75" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6F76: Stop Log

Stops the log.

<i>Host to RF System Request</i>	
<b>STX</b> "6F76" <a> [ <eeee...> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
a	Addressed mode ('n': non-addressed; 'a': addressed).
e	Tag EPC.

<i>RF System to Host Response</i>	
-----------------------------------	--



<b>ACK</b>	
<b>STX</b> "6F76" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

## 6F77: Read Log Data

Read the complete log or parts of it.

<i>Host to RF System Request</i>	
<b>STX</b> "6F77" <ssss> <cccc> <a> [ <eeee...> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
s	First measurement to read.
c	Maximum number of measurements to read.
a	Addressed mode ('n': non-addressed; 'a': addressed).
e	Tag EPC.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F77" <s> <f> <tttttttt> [ <a> <c> <iiii> <vvvv> ] ...	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
t	Start time (UTC-Timestamp).
a	Sensor type: 'a': Analogue sensor attached directly to transponder ASIC. 'd': Digital ( $\mu$ C-powered) sensor attached via SPI.
c	Sensor number.
i	Index.
v	Value in dK or mV.

## 6F7F: CoolLog Custom Command

Low Level command to enable all functions of the SL900a not available through high level functions.

<i>Host to RF System Request</i>	
<b>STX</b> "6F7F" <cc> [ <pp...> ] <a> [ <eeee...> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
c	CoolLog Command Code (A0-AE).
p	Optional Parameters.
a	Addressed mode ('n': non-addressed; 'a': addressed).

e	Tag EPC.
---	----------

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F7F" <s> <f> [ <rr...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
r	Response Data.

## 23 ISO 18000-3 Mode 3

### 23.1 Features

- Read/write transponder 13,56 MHz
- Typically 96 bits EPC used for anticollision

Since there are various different flavours of ISO 18000-3 Mode 3 tags (e. g. different memory size/arrangement, special features etc.), it is strongly recommended to consult the transponder datasheet for further information.

Due to memory limitation, Scemtec 13.56MHz devices can process only up to 256 blocks of data at once. The maximum supported Q value is 8, which corresponds to 256 (2<sup>8</sup>) slots.

### 23.2 Low Level Functions

These commands are basically those offered by the transponder air protocol part of ISO 18000-3 Mode 3. Please note that the commands Begin Round and Next Slot have been replaced by the Single Round command, the reason for this being strict timing limitations making the single low level functions useless in combination with a host driven protocol. For the same reason, Resize Round is not available as a low level function. Also, the optional Block Write command has been left out, for it is contained in the high level Write function.

#### 6F80: Low Level Select

This command represents the Select command as described in ISO 18000-3 Mode 3. For most applications, it is recommended to rather use the high level Select command (8F80) which is more integrated and also more convenient to use. On the other hand, the low level version provides additional functionality for more sophisticated applications.

As tags generally don't respond to a Select command, the response is always empty. Since only the selection is done, this function does not acquire a tag handle for subsequent operations. For those, the SL flag is either provided via parameter (e. g. "Single Round") or via settings using key M3S\_SEL (e. g. "Create Inventory").

<i>Host to RF System Request</i>	
<b>STX</b>	"6F80" <t> <a> <m> <r> <pppp> <ll> <mmmm...>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
t	Target: 0: Inventoried (S0). 2: Inventoried (S2). 4: SL.
a	Action: 0: Assert matching; deassert non-matching.

	1: Assert matching. 2: Deassert non-matching. 3: Toggle matching. 4: Deassert matching; assert non-matching. 5: Deassert matching. 6: Assert non-matching. 7: Toggle non-matching.
m	Memory bank: 0: Reserved. 1: UII (EPC) memory. 2: TID memory. 3: User memory.
r	Truncate (0 = disable truncation; 1 = enable truncation).
p	Bit pointer within specified memory bank.
l	Length of mask in bits.
m	Bit mask to compare to specified memory area.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F80"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

### 6F81: Single Round

This may be called a “lower mid-level function”, for it combines Begin Round plus multiple calls to Next Slot and ACK, finally resulting in one complete anticollision round. These functions had to be encapsulated because the maximum time in between them is 1.2ms, which would certainly be violated if host transfers were involved for each of them. Hence, this command can be considered the “smallest sensible unit” of anticollision.

On the other hand, Single Round won't be of much use for practical applications, because it does not return any tag handles which could be used for further operations, nor does it provide a complete inventory. It is recommended to use the high level Inventory functions instead. This command has been implemented rather for diagnosis purposes.

<i>Host to RF System Request</i>	
<b>STX</b> "6F81" <l> <s> <q>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
1	Selection: 0, 1: All 2: ~SL. 3: SL.

s	Session (0 = S0; 2 = S2).
q	Q value ( $2^q$ yields the number of slots to be used).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F81" <s> <f> [ <pppp> <eeeeeeeeeeeeeeee...> ] [ <s> <f> [ <pppp> <eeeeeeeeeeeeeeee...> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	PC (Protocol Control) word.
e	EPC data.

#### 4F80: Low Level Read

This is the low level Read command. The only reason to prefer this one over the appropriate high level function would be the fact that this command supports "0" as the number of blocks, making the tag transmit all data until the end of specified memory bank.

If there is an error when reading all remaining blocks with the zero length as described above, one set of status value and data flag is returned with the error code. On success, one set of status value, data flag and block data is returned for each block the tag provides.

<i>Host to RF System Request</i>	
<b>STX</b> "4F80" <m> <ssss> <cc> [ <hhhh> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Memory bank: 0: Reserved. 1: UII (EPC) memory. 2: TID memory. 3: User memory.
s	Start block number.
c	Number of blocks to read; 0 can be specified to read all remaining blocks until the end of specified memory bank.
h	Handle, i. e. usually RN16 (if omitted, previously stored handle is used).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4F80" <s> <f> [ <dddd> ] [ <s> <f> [ <dddd> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Block data (16 bit).

### 5F80: Low Level Write

It is generally recommended to use the high level Write command instead of this one, which is a plain depiction of the ISO 18000-3 Mode 3 Write command.

<i>Host to RF System Request</i>	
<b>STX</b> "5F80" <m> <ssss> <dddd> [ <hhhh> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Memory bank: 0: Reserved. 1: UII (EPC) memory. 2: TID memory. 3: User memory.
s	Block number to write.
d	Data word to be written (16 bit).
h	Handle, i. e. usually RN16 (if omitted, previously stored handle is used).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5F80" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 6F82: Low Level Custom Command

To submit a custom command, this function can be used. It has been designed as generic as possible so that it can even be used to emulate some of the mandatory and optional commands. The custom command is submitted as a bit field of specific length. As responses to write commands (all commands performing write operations on the tag's EEPROM cells, that is) are received different compared to normal read responses, it must be specified if this is a read command or a write command.

<i>Host to RF System Request</i>	
<b>STX</b> "6F82" <m> <tttt> <rrrr> <cc...>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode ('r' = Read command; 'w' = Write command).
t	Number of bits to transmit.
r	Number of bits expected as response.
c	Custom command to be transmitted in form of a hex data stream, padded to byte boundary.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F82" <s> <f> [ <r...> ]	
<b>ETX</b> {c}	

<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
r	Response data.

### **Examples:**

a) Read two blocks:

**STX** "6f86" "r" "002c" "c2" "c" <ss> <cc> <hhhh> **ETX** {c}

s: Start block  
 c: Block count  
 h: Handle

b) Write one block:

**STX** "6f86" "w" "0034" "c3" "c" <bb> <dddd> <hhhh> **ETX** {c}

b: Block number  
 d: Data word  
 h: Handle

## **23.3 High Level Functions**

These commands combine and/or simplify some of the low level commands. Some low level commands even cannot be used alone due to timing limitations which would inevitably be violated by interleaved host communication.

### **6F8C: Create Inventory**

Builds an inventory of all tags currently present in the field by their EPC. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

To change the default start Q value, session and selection parameters, use the ISO 18000-3 Mode 3 settings M3S\_QVAL\_START, M3S\_SESSION and M3S\_SEL.

<i>Host to RF System Request</i>	
<b>STX</b> "6F8C" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset.

	'c': Cumulative. 'C': Cumulative without HF reset.
--	---

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F8C" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

### 6F8D: Get Inventory

This function will return all EPCs from the current inventory.

<i>Host to RF System Request</i>	
<b>STX</b> "6F8D"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F8D" <ssss> [ <pppp> <eeeeeeeeeeeeeeeeeeee...> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
p	PC (Protocol Control) word.
e	EPC data.

### 6F8E: Get ID Range from Inventory

To read a range of EPCs or one single EPC from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b> "6F8E" <iiii> [ <l111> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of EPC to be read, first is 0000.
l	Number of EPCs to be read minus one (i. e. 0000 means 1 EPC).



<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F8E" [ <cccc> ] [ <pppp> <eeeeeeeeeeeeeeeeee...> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Number of tag EPCs that follow (only present if a number of EPCs was specified).
p	PC (Protocol Control) word.
e	EPC data.

### 6F8F: Create/Get Inventory

This command is a combination of the commands "6F8C" and "6F8D". An inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6F8F" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F8F" <ee> <ssss> [ <pppp> <eeeeeeeeeeeeeeeeee...> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
p	PC (Protocol Control) word.
e	EPC data.

### 8F8C: Realtime Inventory

Builds an inventory of all tags currently present in the field by their EPC. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

In contrast to the “create” inventory functions, this command sends tag EPCs “on the fly”, i. e. as soon as they are detected, in multiple responses. The last response is always empty (0 EPCs) and contains accumulated status flags. When the last response has been transmitted, the device is ready for the next command.

<i>Host to RF System Request</i>	
<b>STX</b> "8F8C" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F8C" <ee> <ssss> [ <pppp> <eeeeeeeeeeeeeeeeeeee...> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
p	PC (Protocol Control) word.
e	EPC data.

### 8F80: Select

This is the high level Select command, which does not only call the low level Select command, but also perform a single anticollision round (with one slot) to obtain the EPC, and finally a ReqRN to obtain a handle, thus laying the groundwork for any subsequent transaction command (like Read, Write, ...). Afterwards, any high level command can be called with address mode 's' to perform operations on the selected tag exclusively. This may be preferable over using address mode 'a' in case multiple operations are to be performed on one tag, but also in case a memory range different than the EPC is to be used to address it.

To change the default session parameter, use the ISO 18000-3 Mode 3 setting M3S\_SESSION.

<i>Host to RF System Request</i>	
<b>STX</b> "8F80" <o> <m> <pppp> <ll> <mmmm...>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
o	Mode: 'h': Return handle only (faster transfer, if PC+EPC is not needed).

	'u': Return handle and UII (EPC).
m	Memory bank: 0: Reserved. 1: UII (EPC) memory. 2: TID memory. 3: User memory.
p	Bit pointer within specified memory bank.
l	Length of mask in bits.
m	Bit mask to compare to specified memory area.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F80" <s> <f> [ <hhhh> [ <pppp> <eeeeeeeeeeeeeeee...> ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
h	Handle (RN16).
p	PC (Protocol Control) word.
e	EPC data.

### 8F81: Iterate Inventory

This might be called an “upper mid-level function”, for it operates slightly below the Inventory commands described above, hence allowing for interleaved operations performed on each tag without the necessity to repeat selection. Basically, it can either be “started” or “continued”. When starting, a new inventory is initiated and either the first found EPC or an empty response returned. When continuing, the next found EPC or an empty response is returned. An empty response means that there are no more tags to iterate – the anticollision cycle is complete. Internally, this command performs one or multiple inventory rounds, depending on the actual tag population. However, the Q value is specified once upon start and must not be varied afterwards. Like the high level Select command, this one also calls ReqRN, so the tag is ready for interaction.

To change the default session and selection parameters, use the ISO 18000-3 Mode 3 settings M3S\_SESSION and M3S\_SEL.

<i>Host to RF System Request</i>	
<b>STX</b> "8F81" <m> <q>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode: 's': Start inventory iteration, beginning with HF reset. 'S': Start inventory iteration without HF reset. 'c': Continue inventory iteration.
q	Q value ( $2^Q$ yields the number of slots to be used).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F81"	
<b>ETX</b> {c}	
<i>Response Data</i>	
–	None.

#### 4F88: Read

This is the high level Read command. In most cases it should be preferred over the low level Read command, since it is more convenient to use. The only reason to use the low level command would be the possibility to use "0" as the number of blocks, making the tag transmit all data until the end of specified memory bank. The high level Read command, on the other hand, continues the tradition of having blocks specified by actual count minus one – which, of course, cannot be used to specify zero length.

<i>Host to RF System Request</i>	
<b>STX</b> "4F88" <o> <m> <ssss> <cc> <a> [ <eeee...>   <hhhh> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
o	Mode ('s': Use single commands for each block; 'm': Use one multiple blocks command).
m	Memory bank: 0: Reserved. 1: UII (EPC) memory. 2: TID memory. 3: User memory.
s	Start block number.
c	Number of blocks to read minus one.
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).
e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4F88" <s> <f> [ <dddd> ] [ <s> <f> [ <dddd> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Block data (16 bit).

## 5F88: Write

This is the high level Write command, which should also be preferred over its low level pendant. Along with the convenient address mode parameter, it integrates the optional Block Write command, which can write multiple blocks at once (but only if the tag supports this, of course!). In single command mode, each specified block is written using the normal Write command – the result is almost the same, only it is a bit slower.

<i>Host to RF System Request</i>	
<b>STX</b> "5F88" <o> <m> <ssss> <cc> <dddd...> <a> [ <eeee...>   <hhhh> ] <b>ETX</b> {c}	
<i>Request Parameters</i>	
o	Mode ('s': Use single commands for each block; 'm': Use one multiple blocks command).
m	Memory bank: 0: Reserved. 1: UII (EPC) memory. 2: TID memory. 3: User memory.
s	Start block number.
c	Number of blocks to be written minus one.
d	Data to be written; a multiple of 4-digit 16-bit hex values matching specified number of blocks.
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).
e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b> <b>STX</b> "5F88" <s> [ <s> [ ... ] ] <b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

## 8F82: Kill

This is the Kill command as specified in ISO 18000-3 Mode 3, which can be used for killing and/or recommissioning a tag, depending on its specification.

<i>Host to RF System Request</i>	
<b>STX</b> "8F82" <r> <pppppppp> <a> [ <eeee...>   <hhhh> ] <b>ETX</b> {c}	

<i>Request Parameters</i>	
r	Recommissioning bits (see ISO 18000-3 Mode 3 specification).
p	Kill password.
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).
e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F82" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 8F83: Lock

This is the Lock command as specified in ISO 18000-3 Mode 3, which can be used for locking certain areas of tag memory. Please see ISO 18000-3 Mode 3 specification for the mask and action parameters' actual meanings.

<i>Host to RF System Request</i>	
<b>STX</b> "8F83" <mmm> <ccc> <a> [ <eeee...>   <hhhh> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Lock mask bits (see ISO 18000-3 Mode 3 specification).
c	Lock action bits (see ISO 18000-3 Mode 3 specification).
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).
e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F83" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 8F84: Access

This is the optional Access command as specified in ISO 18000-3 Mode 3, which can be used to put a tag from “open” into “secured” state. Requirement is a valid access password.

<i>Host to RF System Request</i>	
<b>STX</b>	"8F84" <pppppppp> <a> [ <eeee...>   <hhhh> ]
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
p	Access password.
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).
e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"8F84" <s>
<b>ETX</b>	{ c }
<i>Response Data</i>	
s	Status value.

### 8F85: Erase

This is the optional Block Erase command as specified in ISO 18000-3 Mode 3, which can be used to delete the specified blocks from target memory bank.

<i>Host to RF System Request</i>	
<b>STX</b>	"8F85" <m> <ssss> <cc> <a> [ <eeee...>   <hhhh> ]
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
m	Memory bank: 0: Reserved. 1: UII (EPC) memory. 2: TID memory. 3: User memory.
s	Start block number.
c	Number of blocks to be erased minus one.
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).

e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F85" <s> [ <s> [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 8F86: Permalock

This is the optional Block Permalock command as specified in ISO 18000-3 Mode 3. Please see ISO 18000-3 Mode 3 specification for the parameters' actual meanings.

<i>Host to RF System Request</i>	
<b>STX</b> "8F86" <r> <m> <ssss> <rr> [ <dddd...> ] <a> [ <eeee...>   <hhhh> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
r	Mode ('r': Read lock status; 'l': Change lock status).
m	Memory bank: 0: Reserved. 1: UII (EPC) memory. 2: TID memory. 3: User memory.
s	Mask start address, specified in clusters of 16 blocks.
r	Mask range, specified in clusters of 16 blocks.
d	Mask data (a multiple of 16 bits, each specified in 4 hex digits). This parameter is omitted if mode is specified as 'r'.
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).
e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F86" <s> [ <f> [ <1111> ] ] [ <s> [ <f> [ <1111> ] ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value for one 16 block cluster.



f	Data flag for one 16 block cluster ('n': no data; 'y': data follows). This field is omitted if mode is specified as 'w'.
l	Lock status of one 16 block cluster, binary coded.

### 8F88: Custom Command

To submit a custom command, this function can be used. It has been designed as generic as possible so that it can even be used to emulate some of the mandatory and optional commands. The custom command is submitted as a bit field of specific length. As responses to write commands (all commands performing write operations on the tag's EEPROM cells, that is) are received different compared to normal read responses, it must be specified if this is a read command or a write command.

<i>Host to RF System Request</i>	
<b>STX</b> "8F88" <m> <tttt> <rrrr> <cc...> <a> [ <eeee...>   <hhhh> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode ('r' = Read command; 'w' = Write command).
t	Number of bits to transmit.
r	Number of bits expected as response.
c	Custom command to be transmitted in form of a hex data stream, padded to byte boundary.
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).
e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F88" <s> <f> [ <r...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
r	Response data.

## 23.4 Settings Functions

These commands are used to view or edit ISO 18000-3 Mode 3 specific settings.

### 1F80: Request Setting

This function is used to retrieve the value of a certain ISO 18000-3 Mode 3 setting. Numeric values

are unsigned 16 bit hex values; switch values are "0000" for "false" or "0001" for "true".

<i>Host to RF System Request</i>	
<b>STX</b>	"1F80" <kk>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	Key (see table).

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"1F80" <vvvv>
<b>ETX</b>	{c}
<i>Response Data</i>	
v	Value.

Key table (Number, Customer specific, Alias, Numeric/Switch, Default, Description):

Key	C	Alias	N/S	Def.	Description
01		M3S_RETRIES	n	0000	Retries on read and inventory functions.
02		M3S_METHOD	n	0000	Method: 0=ASK; 1=PJM (supported: ASK).
03		M3S_DIVIDE_RATIO	n	0000	Divide Ratio: 0=FL=423kHz; 1=FL=847kHz (supported: 423kHz).
04		M3S_ASK_MODTYPE	n	0003	ASK Modulation type: 0=FM0; 1=Miller8; 2=Manchester2; 3=Manchester4 (supported: Manchester4).
05		<reserved>	-	0000	
06		M3S_TR_EXT	s	0000	Pilot tone: 0=off; 1=on.
07		M3S_TARI	n	0008	Tari (8us-25us) in us (-> 0x8-0x19).
08		M3S_FAST_MODE	s	0001	Length of Data-1 (1=1.5 or 0=2.0).
09		M3S_QVAL_START	n	0002	Start value of Q to use for anticollision.
0A		M3S_SESSION	n	0000	Session parameter to use for anticollision.
0B		M3S_SEL	n	0000	Selection parameter to use for anticollision.
0C		<reserved>	-	0000	
0D		M3S_INV_RESET_S	s	0001	Perform HF reset at start of inventory command.
0E		M3S_INV_RESET_E	s	0000	Perform HF reset at end of inventory command.
0F		M3S_WRITE_COVER	s	0000	User cover code when writing.
10		M3S_VERIFY	s	0000	Verify mode, i. e. automatic "read after write".
11		M3S_ACK_MODE	n	0000	What to do in case of no valid response to ACK: 0: Send NAK if response received is invalid. 1: Repeat ACK if response received is invalid. 2: Send NAK if no response or response is invalid. 3: Repeat ACK if no response or response is invalid.

### 1F81: Edit Setting

Used to change the value of a certain ISO 18000-3 Mode 3 setting. See "Request Setting" ("1F80") for more details.

<i>Host to RF System Request</i>	
<b>STX</b> "1F81" <kk> <vvvv>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
k	Key.
v	Value.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "1F81"	
<b>ETX</b> {c}	
<i>Response Data</i>	
-	None.

## 24 ISO 18000-3 Mode 3 Custom

### 24.1 NXP ICode ILT Custom Functions

In addition to the commands listed here, the Write command can be used in multiple block mode for two blocks at once when writing to ICode ILT transponders. However, more than two blocks are not supported and result in an error response.

#### 6F84: ICode ILT Config

This command is used to read and/or change the configuration word of an ICode ILT transponder.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F84" <www> <a> [ <eee...>   <hhh> ]
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
w	Toggle config word (or 0000 to read current config word).
a	Address mode: 'n': Non-addressed (neither e nor h specified). 'a': Addressed (e must be specified). 's': Selected (neither e nor h specified). 'h': Handle (h must be specified).
e	EPC of tag to be read (Address mode 'a').
h	Handle, i. e. usually RN16 (Address mode 'h').

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F84" <s> <f> [ <www> ]
<b>ETX</b>	{ c }
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
w	Current config word.

#### 6F85: ICode ILT EAS

The ICode ILT EAS feature is supported by two separate commands. This is the low level command as specified in the ILT data sheet, returning PC and EPC upon EAS detection.

<i>Host to RF System Request</i>	
<b>STX</b>	"6F85"
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
–	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F85" <s> <f> [ <pppp> <eeeeeeeeeeeeeeeeee...> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
p	PC (Protocol Control) word.
e	EPC data.

### 6F86: ICode ILT EAS Alarm

The ICode ILT EAS feature is supported by two separate commands. This is the high level command which can be used similarly to the ICode SL1 and ICode SLI/SL2 EAS commands.

<i>Host to RF System Request</i>	
<b>STX</b> "6F86" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	EAS Mode: 'n': Normal. 'l': Looped.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F86" <mm> <a>	
<b>ETX</b> {c}	
<i>Response Data</i>	
m	Number of matching bits (RFU, currently always zero).
a	Alarm state ('y': alarm; 'n': no alarm).

## 25 Hitag AES

### 25.1 Features

- 125 kHz Read/Write Transponders
- Supports 128 bit AES encryption

### 25.2 Transponder Read Functions

#### 4840: Read Hitag AES Transponder

Read a Page (4 Bytes) from a Hitag AES Transponder.

<i>Host to RF System Request</i>	
<b>STX</b> "4840" <s> <pp>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Read mode : at the moment only „S“ or „s“= single mode.
n	Number of the page to be read, e. g. „01“ for page 1, „05“ for page 5 etc.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4840" <ddddddddd>	
<b>ETX</b> {c}	
<i>Response Data</i>	
d	Page content.

### 25.3 Transponder Write Functions

#### 5840: Write Hitag AES Transponder

With this function a Hitag AES transponder can be written page by page, (i. e. 4 bytes). To be sure that the page was successfully written, command “4840“ should be used for a read-after-write.

<i>Host to RF System Request</i>	
<b>STX</b> "5840" <pp> <ddddddddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	Page number, e. g. „02“ for 2nd or „07“ for 7th page.
d	Data to be written. 2 characters per byte have to be transmitted.
<i>RF System to Host Response</i>	

<b>ACK</b>	
<b>STX</b>	"5840"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

## 25.4 Transponder Specific Functions

### 6840: Get Hitag AES ID

With this function the ID (Page 0) of a Hitag AES can be determined.

<i>Host to RF System Request</i>	
<b>STX</b>	"6840"
<b>ETX</b>	{c}
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6840" <iiiiiiii>
<b>ETX</b>	{c}
<i>Response Data</i>	
i	ID of transponder (Page 0).

### 6841: Set Hitag AES Key

With this function the 128 bit cryptographic key of the Hitag AES can be transmitted to the RF system. The AES key at start-up defaults to 11112222333344445555666677778888.

<i>Host to RF System Request</i>	
<b>STX</b>	"6841" <kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	AES Key.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6841"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

## 26 Hitag Pro

### 26.1 Features

- 125 kHz Read/Write Transponders
- Supports 128 bit AES encryption

### 26.2 Transponder Read Functions

#### 4880: Read Page

Read one page from the transponder memory.

<i>Host to RF System Request</i>	
<b>STX</b> "4880" <m> <s> <pp>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode ('p': Plain, 'a': Authent, 'c': Config).
s	Segment (0..7).
p	Page.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "4880" <dddddddd>	
<b>ETX</b> {c}	
<i>Response Data</i>	
d	Tag data.

#### 4881: Read Byte

Read one byte from the transponder memory.

<i>Host to RF System Request</i>	
<b>STX</b> "4881" <m> <s> <pp> <b>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Mode ('p': Plain, 'a': Authent, 'c': Config).
s	Segment (0..7).
p	Page.
b	Byte.
<i>RF System to Host Response</i>	
<b>ACK</b>	



<b>STX</b>	"4881" <dd>
<b>ETX</b>	{ c }
<i>Response Data</i>	
d	Tag data.

## 4882: Read Multiple Pages

Read one or more pages from the transponder memory.

<i>Host to RF System Request</i>	
<b>STX</b>	"4882" <m> <s> <pp> <c>
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
m	Mode ('p': Plain, 'a': Authent, 'c': Config).
s	Segment (0..7).
p	First Page.
c	Number of pages minus 1.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4882" <ddddddddd> ...
<b>ETX</b>	{ c }
<i>Response Data</i>	
d	Tag data.

## 26.3 Transponder Write Functions

### 5880: Write Page

Write one page to the transponder memory.

<i>Host to RF System Request</i>	
<b>STX</b>	"5880" <m> <s> <pp> <ddddddddd>
<b>ETX</b>	{ c }
<i>Request Parameters</i>	
m	Mode ('p': Plain, 'a': Authent, 'c': Config).
s	Segment (0..7).
p	Page.
d	Data to be written.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5880"

<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 5881: Write Byte

Write one byte to the transponder memory.

<i>Host to RF System Request</i>	
<b>STX</b>	"5881" <m> <s> <pp> <b> <dd>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
m	Mode ('p': Plain, 'a': Authent, 'c': Config).
s	Segment (0..7).
p	First Page.
b	Byte (0..3).
d	Data to be written.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5881"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

### 5882: Write Multiple Pages

Write one or more pages to the transponder memory.

<i>Host to RF System Request</i>	
<b>STX</b>	"5882" <m> <s> <pp> <c> <dddddd> ...
<b>ETX</b>	{c}
<i>Request Parameters</i>	
m	Mode ('p': Plain, 'a': Authent, 'c': Config).
s	Segment (0..7).
p	First Page.
c	Number of pages minus 1.
d	Data to be written.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"5882"
<b>ETX</b>	{c}
<i>Response Data</i>	

-	None.
---	-------

## 26.4 Transponder Specific Functions

## 6880: Get Hitag Pro ID

Get the transponder ID.

<i>Host to RF System Request</i>		
<b>STX</b>	"6880"	
<b>ETX</b>	{c}	
<i>Request Parameters</i>		
-	None.	

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6880" <iiiiiiii>	
<b>ETX</b> {c}	
<i>Response Data</i>	
i	Transponder ID.

## 6881: Set Immobilizer Secret Key

This function is used to configure the secret key used for transponder operations with encrypted communication.

<i>Host to RF System Request</i>	
<b>STX</b>	"6881" <kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
k	Immobilizer Secret Key.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6881"
<b>ETX</b>	{c}
<i>Response Data</i>	
-	None.

## 27 ISO 14443-A

### 27.1 Low Level Functions

#### 6F42: Select

This command selects a single tag for subsequent operations (such as read, write, ...).

<i>Host to RF System Request</i>	
<b>STX</b>	"6F42" <iiiiiii> [ <iiiiii> [ <iiiiii> ] ]
<b>ETX</b>	{c}
<i>Request Parameters</i>	
i	Tag ID (4, 7 or 10 bytes depending on tag type).
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"6F42" <s>
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.

#### 4F40: Read Page

This command reads a single page of data (4 blocks of 8 bytes each) from selected tag.

<i>Host to RF System Request</i>	
<b>STX</b>	"4F40" <bb>
<b>ETX</b>	{c}
<i>Request Parameters</i>	
b	Block number to start reading from.
<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b>	"4F40" <s> <f> [ <dddddddd> ] <s> <f> [ <dddddddd> ] <s> <f> [ <dddddddd> ] <s> <f> [ <dddddddd> ]
<b>ETX</b>	{c}
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Page data (4 blocks).

#### 5F40: Write Block

This command writes a single block of data (8 bytes) to selected tag.

<i>Host to RF System Request</i>	
<b>STX</b> "5F40" <bb> <dddddddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number to write data to.
d	Data to be written to tag.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5F40" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

### 5F41: Write4 Block

This command writes a single block of data (8 bytes) to selected tag. It is faster than the Write Block function (5F40) but less compatible. Some tags (e. g. Mifare classic) won't support this command.

<i>Host to RF System Request</i>	
<b>STX</b> "5F41" <bb> <dddddddd>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number to write data to.
d	Data to be written to tag.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "5F41" <s>	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.

## 27.2 High Level Functions

### 6F4C: Create Inventory

Builds an inventory of all tags currently present in the field by their serial number. In "single" mode, the inventory is initialized at the beginning, whereas in "cumulative" mode, a previously created inventory will be kept. This function is breakable, that means sending an ESC character in busy state will force it to return an incomplete result.

<i>Host to RF System Request</i>	
----------------------------------	--

<b>STX</b> "6F4C" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F4C" <ee> <ssss>	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.

### 6F4D: Get Inventory

This function will return all serial numbers from the current inventory. Since ISO14443A tag IDs can have a length of 4, 7 or 10 bytes, IDs are always returned as 10 bytes, padded with trailing zeroes if necessary.

<i>Host to RF System Request</i>	
<b>STX</b> "6F4D"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
-	None.

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F4D" <ssss> [ <iiiiiiiiiiiiiiiiiiii> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (80 bit), padded with trailing zeroes if necessary.

### 6F4E: Get ID Range from Inventory

To read a range of IDs or one single ID from a previously created inventory, this function can be used.

<i>Host to RF System Request</i>	
<b>STX</b> "6F4E" <iiii> [ <1111> ]	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
i	Index of ID to be read, first is 0000.
l	Number of IDs to be read minus one (i. e. 0000 means 1 ID).

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F4E" [ <cccc> ] [ [ <iiiiiiiiiiiiiiiiiiii> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
c	Number of tag IDs that follow (only present if a number of IDs was specified).
i	Tag ID (64 bit) in reverse byte order.

## 6F4F: Create/Get Inventory

This command is a combination of the commands "6F4C" and "6F4D". An inventory is built and instantly transmitted to the host.

<i>Host to RF System Request</i>	
<b>STX</b> "6F4F" <m>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
m	Inventory mode: 's': Single. 'S': Single without HF reset. 'c': Cumulative. 'C': Cumulative without HF reset.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "6F4F" <ee> <ssss> [ [ <iiiiiiiiiiiiiiiiiiii> ] [ ... ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
e	Binary coded error/warning information: 00: Ok. 01: Inventory overflow. 02: Collision queue overflow. 08: Incomplete warning - *possibly* not all tags have been found. 10: Halt failure warning - can be ignored.
s	Current inventory size, i. e. the number of tags found.
i	Tag ID (64 bit) in reverse byte order.

## 28 ISO 14443-A Custom

### 28.1 NXP NTAG Custom Functions

#### 8F40: Get Version

This command reads the version information from selected tag.

<i>Host to RF System Request</i>	
<b>STX</b> "8F40"	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
–	None.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F40" <s> <f> [ <vvvvvvvvvvvvvvvvvv> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
v	8 bytes of version information (see NTAG data sheet).

#### 8F41: Fast Read

This command reads specified number of blocks (8 bytes each) from selected tag.

<i>Host to RF System Request</i>	
<b>STX</b> "8F41" <bb> <cc>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
b	Block number to start reading.
c	Number of blocks to be read minus one.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F41" <s> <f> [ <ddddddddd> [ <s> <f> <ddddddddd> [...] ] ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
d	Block data.



### 8F42: Read Counter

This command reads specified counter from selected tag. Please be aware that the counters must be enabled in the tag's configuration.

<i>Host to RF System Request</i>	
<b>STX</b> "8F42" <cc>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
c	Counter address.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F42" <s> <f> [ <vvvvvv> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
v	Counter value.

### 8F43: Password Authentication

This command transfers a password (8 bytes) to selected tag.

<i>Host to RF System Request</i>	
<b>STX</b> "8F43" <pppppppp>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	
p	Password data.

  

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F43" <s> <f> [ <aaaa> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
a	The PACK response from the tag (see NTAG data sheet).

### 8F44: Read Signature

This command reads the NTAG signature from selected tag.

<i>Host to RF System Request</i>	
<b>STX</b> "8F44" <aa>	
<b>ETX</b> {c}	
<i>Request Parameters</i>	

a	Address (reserved for future use, leave at 00).
---	---

<i>RF System to Host Response</i>	
<b>ACK</b>	
<b>STX</b> "8F44" <s> <f> [ <nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn> ]	
<b>ETX</b> {c}	
<i>Response Data</i>	
s	Status value.
f	Data flag ('n': no data; 'y': data follows).
n	NTAG signature (see NTAG data sheet).

## 29 Document Version History

<i>Version</i>	<i>Date</i>	<i>Changed by:</i>	<i>Description</i>
4.00	29.07.2002	Lenz	<ul style="list-style-type: none"> <li>Document structure completely reworked.</li> <li>Many errors corrected.</li> <li>Obsolete version history entries removed.</li> </ul>
4.01	30.09.2002	Lenz	<ul style="list-style-type: none"> <li>ISO service settings and default values changed.</li> </ul>
4.02	10.12.2002	Lenz	<ul style="list-style-type: none"> <li>Some new/changed SHL-2xxx settings.</li> <li>Command "Get NAK Log" removed from general protocol.</li> <li>A few minor fixes.</li> <li>RS485 protocol extension changed and marked preliminary.</li> </ul>
4.03	19.12.2002	Lenz	<ul style="list-style-type: none"> <li>ISO 15693 Proximity: "Submit Custom Command" was split into one function for read commands and one for write commands.</li> <li>New command "Set ISO 15693 Operation Mode".</li> <li>New command "Change Baud Rate Setting".</li> <li>Support for the KSW TempSens has been added to the ISO 15693 Proximity protocol.</li> </ul>
4.04	16.01.2003	Lenz	<ul style="list-style-type: none"> <li>KSW TempSens protocol update.</li> </ul>
4.05	19.02.2003	Lenz	<ul style="list-style-type: none"> <li>KSW TempSens long range protocol added.</li> <li>New commands "2006" and "2007".</li> <li>Barcode Read function moved from chapter "2xxx" to "4xxx".</li> </ul>
4.06	09.07.2003	Lenz	<ul style="list-style-type: none"> <li>New command "Write Q5 with lock bit".</li> <li>New TempSens command "Set calibration values".</li> <li>ISO long range options updated for TempSens.</li> <li>TempSens "Start Log" commands were corrected.</li> <li>New version strings for commands 1000 and 1001.</li> <li>TempSens "Get Status" command now features enhanced mode.</li> </ul>
4.07	01.09.2003	Lenz	<ul style="list-style-type: none"> <li>Document migrated to OpenOffice 1.1</li> </ul>
4.08	21.10.2003	Lenz	<ul style="list-style-type: none"> <li>Obsolete passage about Writing ISO 15693 (prox.) has been updated.</li> <li>Security notes on Q5 passwords have been added.</li> <li>Slight cosmetic fixes have been made.</li> </ul>
4.09	03.02.2004	Lenz	<ul style="list-style-type: none"> <li>Icode SLI EAS functionality has been added.</li> <li>Parameters for SHL-2xxx reader synchronization have been added.</li> </ul>
4.10	19.04.2004	Lenz	<ul style="list-style-type: none"> <li>New "Get Single ID from Inventory" functions.</li> <li>New Reader ID functions.</li> <li>New SHL-2xxx setting keys.</li> </ul>
4.11	15.06.2004	Lenz	<ul style="list-style-type: none"> <li>New My-D Custom Commands.</li> <li>Some SHL global settings added.</li> </ul>
4.12	30.06.2004	Lenz	<ul style="list-style-type: none"> <li>SHL Inventory functions: HF reset is now optional.</li> <li>I*Code EPC/UID has been added.</li> </ul>
4.13pre	09.09.2004	Witt	<ul style="list-style-type: none"> <li>configuration for e5550 completed.</li> <li>examples for 6600 and 6601(Temic e5550) has been</li> </ul>

			added.
4.13	02.12.2004	Lenz	<ul style="list-style-type: none"> <li>Commands "Sync Listen" and "Ping Sync Slave" have been added.</li> <li>ICode EPC/UID timeslot translation tables corrected.</li> <li>ISO 15693 Modulation index setting changed.</li> <li>Tag type specific factory/user settings commands have been removed.</li> <li>"Calibrate Detection Thresholds" functions have been removed.</li> </ul>
4.14	13.01.2005	Lenz	<ul style="list-style-type: none"> <li>Sync diagnosis commands have been added/updated.</li> <li>SHL-2001 settings and default values updated.</li> </ul>
4.15	15.03.2005	Lenz	<ul style="list-style-type: none"> <li>13.56MHz signal analysis functions have been changed.</li> <li>Several minor corrections.</li> </ul>
4.16	22.06.2005	Witt	<ul style="list-style-type: none"> <li>Hitag S added to Hitag chapter.</li> </ul>
4.17	14.11.2005	Lenz/Witt	<ul style="list-style-type: none"> <li>13.56MHz long range protocol reworked for new digital board and firmware.</li> <li>ISO 11784 / ISO 11785 (FDX-B) added.</li> </ul>
4.18	15.05.2006	Lenz	<ul style="list-style-type: none"> <li>13.56MHz long range protocol reworked (again) for new digital board and firmware.</li> <li>13.56MHz proximity protocol removed. 13.56MHz long range protocol becomes 13.56MHz protocol.</li> <li>SC-433 added.</li> </ul>
4.19	10.08.2006	Lenz	<ul style="list-style-type: none"> <li>Background functions removed.</li> <li>13.56MHz settings updated.</li> <li>Various minor corrections.</li> </ul>
4.20	31.08.2006	Kalbitzer	<ul style="list-style-type: none"> <li>Background functions EM4050 removed</li> </ul>
4.21	24.10.2006	Lenz	<ul style="list-style-type: none"> <li>KSW VarioSens protocol added.</li> <li>Description for command 1020 completed.</li> <li>ICode SLI-S protocol added.</li> <li>New command 5C82 added.</li> <li>New command 2012 added.</li> <li>New command 1026 added.</li> </ul>
4.22	20.11.2006	Lenz	<ul style="list-style-type: none"> <li>Command 100A: added enhanced bit field.</li> <li>New commands 4F12 and 6F1B added.</li> </ul>
4.23	16.04.2007	Lenz	<ul style="list-style-type: none"> <li>13.56MHz settings updated.</li> </ul>
4.24	10.10.2007	Kalbitzer	<ul style="list-style-type: none"> <li>ISO 11784/5 Tag added to Function 100A</li> <li>Minor Changes on 1001 description</li> </ul>
4.25	17.12.2007	Pauls	<ul style="list-style-type: none"> <li>New Command 102A added.</li> <li>New Command 102B added.</li> <li>New Command 102C added.</li> <li>New Command 102D added.</li> <li>New Command 102E added.</li> <li>New Command 102F added.</li> </ul>
4.26	14.02.2008	Lenz	<ul style="list-style-type: none"> <li>13.56MHz settings updated.</li> <li>New function 1023 added.</li> <li>Command Buffer functions adapted to multi line functionality.</li> <li>ISO 15693 inventory commands enhanced to</li> </ul>

			support DSFID output. • New function 4C24 added. • ISO Setting ISOS_REVERT_BYTES is now available again. • All “Get Single ID from Inventory” functions were enhanced and renamed to “Get ID Range from Inventory”. • RS485 part moved to a separate document since RS485 is no longer officially supported.
4.27	11.04.2008	H. Lenz	• Realtime Inventory functions added. • 13.56MHz global settings enhanced.
4.28	16.04.2008	H. Lenz	• New functions 5C24 and 6C28 added.
4.29	16.07.2008	B. Witt	• New function F003 added
4.30	14.08.2008	H. Lenz	• SC433 compatibility mode functions added. • GSM specific functions added. • Interface board settings updated.
4.31	22.10.2008	H. Lenz	• GSM receive/transmit commands corrected.
4.32	03.12.2008	B. Witt	• Commands for PCF7937 (Hitag2 extended) added • Commands for Cryptology coprocessor added
4.33	18.12.2008	H. Lenz	• Port manager mode descriptions corrected and enhanced. • ScemTag sensor transponder functions added. • More interface board settings added. • New ISO functions 5C26, 5C27, 6C2A, 6C2B added.
4.34	30.03.2009	H. Lenz	• ISO15693 settings enhanced. • Command 6CF8 syntax description corrected. • New functions 1028 and 1029 added. • New functions 6010 and 6CDF added.
4.35	02.09.2009	B. Witt	• Added multiple XMA-CFG for Hitag2ext
4.36	26.03.2010	C. Pauls	• New functions 5102, 5103 and 6105 added.
4.37	29.04.2010	H. Lenz	• Transponder tables for command 100A completed.
4.38	05.07.2010	H. Lenz	• Response format of realtime inventory functions changed. • Function 200A now reports RTC power on reset. • New functions 1C1E and 1C38 added. • Command 6CAD parameters corrected.
4.39	05.08.2010	H. Lenz	• New SYN codes #13 and #14 added.
4.40	07.12.2010	H. Lenz	• New functions 6CA0, 6CA4, 6CA5, 6CA6 and 6CA7 added.
4.41	09.03.2010	J. Kalbitzer	• New functions 6220, 6201
4.42	28.04.2011	H. Lenz	• New function 100E added; ScemTag functionality reduced to command 6CF0.
4.43	11.05.2011	H. Lenz	• EPC Class 1 Gen.2 Protocol added.

4.44	24.05.2011	H. Lenz	<ul style="list-style-type: none"> <li>Redundant function 2012 removed.</li> <li>New function 6F60 added.</li> </ul>
4.45	22.06.2011	J.Kalbitzer	<ul style="list-style-type: none"> <li>Changed function 6220.</li> </ul>
4.46	27.07.2011	H. Lenz	<ul style="list-style-type: none"> <li>New functions 1068 and 1069 added.</li> <li>New functions 1070, 1071, 1072 and 1073 added.</li> <li>New functions E008 and E009 added.</li> </ul>
4.47	29.09.2011	C. Pauls	<ul style="list-style-type: none"> <li>New functions 6F70, 6F73, 6F74, 6F75, 6F76, 6F77 and 6F7F added.</li> </ul>
4.48	07.11.2011	H. Lenz	<ul style="list-style-type: none"> <li>Settings updated.</li> </ul>
4.49	07.02.2012	C. Pauls	<ul style="list-style-type: none"> <li>New setting GLS_REGION added.</li> </ul>
4.50	15.02.2012	H. Lenz	<ul style="list-style-type: none"> <li>ISO 18000-3 Mode 3 added.</li> <li>ICode SLI(-S) "Get ID Range from I(P)R Inventory" command descriptions corrected to match actual functionality.</li> </ul>
4.51	13.03.2012	C. Pauls	<ul style="list-style-type: none"> <li>Hitag AES added.</li> <li>Hitag Pro added.</li> </ul>
4.52	14.05.2012	C. Pauls	<ul style="list-style-type: none"> <li>EPC class 1 gen 2 communication profiles added.</li> </ul>
4.53	17.09.2012	H. Lenz	<ul style="list-style-type: none"> <li>New functions 200C and 200D added.</li> </ul>
4.54	15.10.2012	H. Lenz	<ul style="list-style-type: none"> <li>New function 6C1C added.</li> </ul>
4.55	16.10.2012	H. Lenz	<ul style="list-style-type: none"> <li>ISO 15693 settings updated.</li> </ul>
4.56	29.10.2012	H. Lenz	<ul style="list-style-type: none"> <li>FIFO buffer support functions (1078, 1079, 107A, 107B, 107C, 107D) added.</li> <li>SmartRead resume function (1074) added.</li> </ul>
4.57	04.11.12	J.Kalbitzer	<ul style="list-style-type: none"> <li>New function 6604: Q5 Transponder Set to 4001 Compatible Mode</li> </ul>
4.58	08.11.12	J.Kalbitzer	<ul style="list-style-type: none"> <li>Functions for EM 4205/4305 added</li> </ul>
4.59	19.02.2013	H. Lenz	<ul style="list-style-type: none"> <li>Functions 2020 and 2021 adapted to new MUX format.</li> </ul>
4.60	11.03.2013	H. Lenz	<ul style="list-style-type: none"> <li>New functions 1076 and 1077 added.</li> </ul>
4.61	03.04.2013	C. Pauls	<ul style="list-style-type: none"> <li>New function 68FF added.</li> <li>Fixed Hyperlinks in Table of Contents</li> </ul>
4.62	21.11.2013	C. Pauls	<ul style="list-style-type: none"> <li>New function 1075 added.</li> </ul>
4.63	14.02.2014	C. Pauls	<ul style="list-style-type: none"> <li>SC433 settings updated.</li> </ul>
4.64	10.04.2014	H. Lenz	<ul style="list-style-type: none"> <li>Functions for EM4237 added.</li> </ul>
4.65	15.04.2014	H. Lenz	<ul style="list-style-type: none"> <li>ISO 18000-3 Mode 3 settings updated.</li> </ul>
4.66	17.04.2014	H. Lenz	<ul style="list-style-type: none"> <li>ISO 18000-3 Mode 3 custom command splitted into low level and high level function.</li> </ul>
4.67	30.04.2014	H. Lenz	<ul style="list-style-type: none"> <li>New functions 2040 and 2048 added.</li> </ul>
4.68	09.10.2014	H. Lenz	<ul style="list-style-type: none"> <li>Settings for 13MHz corrected.</li> <li>Command 1069: return values documented.</li> <li>Global settings and interface settings updated.</li> <li>Note about multiple responses added to basics.</li> </ul>
4.69	05.01.2015	H. Lenz	<ul style="list-style-type: none"> <li>New function 106A added.</li> <li>Status code for function 6CCB corrected.</li> </ul>

			<ul style="list-style-type: none"> <li>Function 100A return flags updated.</li> </ul>
4.70	26.02.15		<ul style="list-style-type: none"> <li>New functions 6412,6413 for EM4305 added.</li> </ul>
4.71	24.03.2015	H. Lenz	<ul style="list-style-type: none"> <li>ISO14443A and NXP NTAG functions added.</li> </ul>
4.72	27.03.2015	H. Lenz	<ul style="list-style-type: none"> <li>Corrections to ISO14443A and NXP NTAG functions.</li> </ul>
4.73	06.05.2015	J.Kalbitzer	<ul style="list-style-type: none"> <li>Corrections to EM4205/4305 functions.</li> </ul>