TECHNOLOGY SOLUTIONS UK LTD

part of HID

STORM Protocol for TSL® 3N1X Modules

Version 1.7.5

www.tsl.com

Design • Development • Manufacture

Table of Contents

History 2
1. INTRODUCTION 4
1.1. Module Description 4
1.2. Module Family 4
1.3. Module command banks 4
2. MESSAGE FORMATS 6
2.1. Commands 6
2.2. Responses 6
2.3. Control 6
2.4. Parameter values 7
3. COMMANDS 8
3.1. Description 8
3.2. Command Codes 8
3.3. Command ID 9
4. COMMAND RESPONSES 28
4.1. Inventory Response 28
4.2. Datastore Response 29
5. RESPONSE HEADERS 30
5.1. Header codes 30
5.2. Header code types 31
6. APPENDIX 32
6.1. Length and checksum 32
6.2. Access response codes (EA) 33
6.3. Backscatter response codes (EB) 33
6.4. Command response codes (EC) 33
6.5. GPIO line usage 34
6.6. Inventory Stop Reasons 35
About TSL® 36
Contact 36
About HID Global 36

History

Version	Date	Modifications
1.0.0	11/08/2021	Original and (some would say the) best
1.1.0	02/12/2021	RF Modes are now the 3-digit versions in SDK v1.01.00
1.2.0	08/02/2022	Re-ordered lock flags in line with G2V2 spec
1.2.1	04/03/2022	Fixed some typos
1.3.0	04/04/2022	Added $ir -ps
1.3.1	12/04/2022	Maximum value of select offset is 2^{14}-1
1.3.2	27/05/2022	Added trigger mode 3 (while pressed) Added $sy regulatory commands Added new trigger mode Fixed some typos
1.4.0	29/06/2022	Uses SDK v1.02.00 Added RF Mode 202 Refactored Stop codes Added $au Authenticate (experimental)
1.4.1	30/06/2022	Correction of DRM modes in tables
1.4.2	14/07/2022	Marked RFMode 202 as experimental
1.4.3	25/08/2022	Temperature checks now also in $sy
1.5.0	21/10/2022	Added all supported reg regions Added 3419
1.5.1	27/10/2022	Added engineering options for power trim
1.5.2	02/11/2022	Added engineering option for SJC tests $en -ca -pc -sj in ASCII mode only
1.6.0	29/11/2022	Added multi-bank support Added new $i2 command to support external I2C Added support for DevKit MonzaX-8K test tag Fixed some typos Removed engineering options from this document
1.6.1	15/12/2022	Added $sy regulatory for PRBS & ETSI_BURST
1.6.2	06/01/2023	Removed MonzaX-8K direct support (may be done by using generic I2C commands)
1.6.3	30/01/2023	Changed Go multi-bank to $ba-gm
1.6.4	17/03/2023	Documented control commands ^D and ^R Added $sy-po power offset for integrators
1.6.5	18/04/2023	Added $ba-mc multi-bank copy
1.7.0	28/06/2023	Added $ds DataStore Added block-encoding parameter -be to $wr $lo $ac $bw Added $sy-ce command echo (default off)
1.7.1	30/06/2023	Fixed some typos
1,7.2	05/07/2023	Added $ds -ma -ud -mo

Version	Date	Modifications
1.7.3	12/07/2023	$ac note about needing to send twice Corrected $sy-rd value range $ds -ud is now -rl Fixed some typos
1.7.4	14/07/2023	Added $ir -ds to tell inventory to use Datastore Added RP: header in $MI example Missed $ds from list of commands Added example Datastore response
1.7.5	21/07/2023	Added $ds -cs Added global -id <n> parameter

1. INTRODUCTION

This document describes the Technology Solutions *STORM Protocol* as used in the 3N1X RAIN RFID modules. The commands, responses and their parameters are detailed along with the communication formats.

It is important to note that this document is not a transponder tutorial and users should have at least a basic understanding of UHF RFID transponders.

The *STORM Protocol* allows a simple method of configuring the module, whilst retaining high-performance and without the complexity of running low-level RFID transponder operations.

Multiple RFID operations can be pre-configured in ten independent command banks which can then be executed at will.

1.1. Module Description

The 3N1X family of powerful, self-contained modules enable high performance UHF RFID reading within mobile or fixed UHF RFID reader applications.

It is compact, lightweight and provides industry leading RFID performance for this form factor.

Just add a suitable antenna and then connect to your system of choice using USB or high-speed serial port.

The USB is a Full-Speed CDC virtual comport. The VID is 0x2D05 and the PID is set to the module type in hex (eg 0x3117) and the USB serial number is set to the module serial number.

The serial port should be configured as *921600,8,N,1* The baudrate may be changed, but for performance reasons, we recommend keeping this speed as high as possible.

Additional connections are available for hardware start and stop triggers as well external I2C communications (see $i2 command).

1.2. Module Family

The 3N1X family provides a range of antenna and performance options

Module Name	Antennas	RFID processor
3113	1	E310
3115	1	E510
3117	1	E710
3119	1	E910
3417	4	E710
3419	4	E910

1.3. Module command banks

The module has ten independent commands banks. Each bank has its own set of ***Inventory*** round parameters and ten command-slots which may be used for Gen2 ***Select*** or other access commands.

The command-slots and the ***Inventory*** round parameters may be configured in any order. However, please note that whilst the ***Inventory*** round parameters may be set by multiple commands, the Gen2 access commands are atomic and will replace previous settings.

The ***Bank*** command allows a complete reset of the specified bank as well as being used to execute the command configured in that bank.

Additionally, the command banks may be executed by pulling GPIO3 to ground. The bank to be used in this case should be selected via the **System** command.

If no stop conditions are configured in a bank, then the RFID processing will continue until GPIO4 is pulled to ground and released.

Multi-bank support is available, meaning that a sequence of banks may be executed in a one-shot or continuous mode via a single command, with an overall set of stop conditions.

2. MESSAGE FORMATS

Communication with the 3N1X module may be sent via either the USB or serial connection, in either ASCII (human-readable) or BINARY (computer-efficient) formats and may be freely mixed.

Responses will be in the same format and on the same connection as the message sent. ASCII format command codes and parameters may be upper or lower case.

2.1. Commands

FORMAT	COMMAND STRUCTURE	
ASCII	$XX-T1V1-T2V2...-TnVn<LF>	
	$	Start of an ASCII formatted command
	XX	Command code
	Tn	Parameter codes (2-bytes)
	Vn	Parameter values (0-m bytes)
	<LF>	Line Feed (0x0A)
BINARY	<SOH><length>X<TLV1>...<TLVn><checksum>	
	<SOH>	Start of a BINARY formatted command (0x01)
	<length>	16-bit length *(see Appendix)*
	X	Command token
	TLVn	Parameters in TLV format (see below)
	<checksum>	16-bit checksum *(see Appendix)*

2.2. Responses

FORMAT	COMMAND STRUCTURE	
ASCII	SC: XX<LF> T1: V1<LF> Tn: Vn<LF> EC: <status><LF>	
	SC: XX	ASCII Response to Command XX
	Tn:	Parameter codes
	Vn	Parameter values
	EC: <status>	Successful or error completion
	<LF>	Line Feed (0x0A)
BINARY	<SOH><length><TLV1>...<TLVn><checksum>	
	<SOH>	BINARY response (0x01)
	<length>	16-bit length *(see Appendix)*
	TLVn	Parameters in TLV format (see below)
	<checksum>	16-bit checksum *(see Appendix)*

2.3. Control

There are also two control commands which may be sent to the module.

- ABORT current command (ETX / 0x04 / Control-D)
- REPEAT last command (DC2 / 0x12 / Control-R)

2.4. Parameter values

2.4.1. ASCII format

Command parameters are almost all specified as numeric values.

Numbers may be prefixed by a number-base letter. For example, the decimal number 123 can be written as:

Hexadecimal	x7B
Decimal	123 (or d123)
Octal	o173
Binary	b01111011

Boolean values are sent as 0 (FALSE) and 1 (TRUE).

Tag-data parameters are sent as an ASCII hexadecimal array (eg *1234ABCD*) of 8, 16 or 32-bit words as appropriate.

2.4.2. BINARY format

Command parameters are almost all specified as numeric values, sent in big-endian format.

Boolean values are sent as 0 (FALSE) and 1 (TRUE).

Data parameters are sent as an array of 8, 16 or 32-bit words as appropriate, in big-endian format.

Parameter tags without a value should have a length byte of 0.

3. COMMANDS

3.1. Description

The 3N1X module has 10 independently configurable inventory command banks, each of which can have up to 10 Select or Gen2 commands.

Typically, the Inventory command and (optionally) Gen2 Select and other access commands are used to configure or update a command bank.

The inventory is then executed by using the Bank command or by using the hardware trigger line.

3.2. Command Codes

Command	Description	Code	Token
Module information	Get module information	mi	a
System	Set System Information	sy	b
Engineering	*Reserved, Factory use only*	*en*	*c*
Bank	Execute / Reset bank	ba	d
Select	Add a G2 Select	se	e
Read	Add a G2 Read	re	f
Write	Add a G2 Write	wr	g
Kill1	Add a G2 Kill1	k1	h
Kill2	Add a G2 Kill2	k2	i
Lock	Add a G2 Lock	lo	j
Access	Add a G2 Access	ac	k
BlockWrite	Add a G2 BlockWrite	bw	l
BlockPermalock	Add a G2 BlockPermalock	bp	m
Authenticate	Add a G2 Authenticate	au	n
Untraceable	*Add a G2 Untraceable*	*un*	*o*
Inventory	Configure inventory round in bank	ir	p
Interface I2C	Transmit & receive on the external I2C interface	i2	q
Datastore	Setting up list of tags for processing	ds	r

3.2.1. Module information ($mi)

Returns the module version and configuration information. Multiple fields are returned.

Responses: SC, MM, MS, MT, MB, MF, MU, RT, RB, RF, RU, RR, RP, AG, EC

See Response Headers for ASCII and BINARY field headers.

ASCII example

```
SC: $mi
MM: Technology Solutions (UK) Ltd
MS: 3117-21360000
MT: 3117 RAIN RFID module
MB: 1.1.0.0
MF: 0.9.1.0
MU: 0038002E-34305117-33353839
RT: 0710
RB: 1.0.0.6
RF: 1.1.0.47
RU: 00000710-00000000-002C0006
RR: 0 (FCC)
RP: 3000
AG: 0
EC: 0
```

In BINARY

MB: & MF: are returned as an array of 4 x 8-bit numbers
MU: & RU: are returned as an array of 3 x 32-bit numbers

3.3. Command ID

All STORM commands may utilise an additional numeric parameter.

Parameter	Value (default in bold)	Code	Token
Identification	0 – 2^{32}-1	-id	0x1F

This parameter will (when non-zero) be echoed back in the ***SC:*** header (irrespective of the *$sy -ce* setting).

In ASCII mode the value will be sent as a hexadecimal number (eg. xA1)

This can be used by the host system (or SDK) to sequence and keep track of commands sent to the module. It has no effect on module command processing beyond this.

3.3.1. System ($sy)

For the use of system integrators. A password is required to access most of these options. The system command is used to configure the non-volatile settings of the module as well as issuing regulatory test commands.

Parameter	Value (default in bold)	Code	Token
Password	XXXXXXXX	-pa	0x20
Set Password *(default not set)*	XXXXXXXX	-sp	0x21
Trigger bank ǂ	0 – 9, 255 **(0)**	-tb	0x22
Trigger mode ǂ	See table below	-tm	0x23
Temperature Sensor ǂ	-	-ts	0x24
Temperature Logging ǂ	**0**-900 **(0)**	-tl	0x35
Antenna gain *	**0** – 10 dB	-ag	0x26
Serial baudrate	See table below	-br	0x27
Regulatory region	See table below	-rr	0x28
Regulatory antenna	1 – 4 **(1)**	-ra	0x29
Regulatory mode	See below	-rm	0x2A
Regulatory power *(100th's dBm)*	100 – 3000 **(3000)**	-rp	0x2B
Regulatory frequency *(kHz)*	865000-868000, 902000-928000 **(0)**	-rf	0x2C
Regulatory remain on	**0** – 1 **(0)**	-ro	0x2D
Regulatory duration (seconds)	0 – 3600 (**0**)	-rd	0x2E
Regulatory test	See below	-rt	0x2F
Go to module bootloader	-	-gb	0x30
Power offset *(100th's dBm)*	0 – 200 **(0)**	-po	0x31
Enable SC: command echo	0 – 1 **(0)**	-ce	0x32

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Set the Trigger bank to 255 to use the multi-bank option

(ǂ) Password not required
(*) Currently not supported

Regulatory Region	Value	Default RF Mode
FCC	**0**	244
Hong Kong	3	244
Taiwan	4	244
ETSI_LOWER	7	241
Korea	8	244
Malaysia	9	244
CHINA	10	244
South Africa	12	244
Brazil	13	244
Thailand	14	244
Singapore	15	244
Australia	16	244
India	17	241
Uruguay	18	244
Vietnam	19	244
Philippines	21	244
Indonesia	23	244
New Zealand	24	244
Japan2	25	244
Peru	27	244
ETSI_UPPER	29	344

Trigger mode	Value
Disabled	0
Single-shot	**1**
Repeat	2
While pressed	3

Serial baudrate
921600
460800
230400
115200
57600
38400
19200
9600

RF Mode	Optimisation	FLM	PIE	Miller	Link Freq	Tari uS	FCC	EUB	EUL	CHI	9	7	5	3
103	Read rate	DSB	1.5	1	640	6.25	Y				Y	Y		
302	Read rate	PR_ASK	2	1	640	7.5	Y	Y			Y	Y		
120	Read rate	DSB	1.5	2	640	6.25	Y				Y	Y	Y	
323	Read rate	PR_ASK	2	2	640	7.5	Y	Y			Y	Y	Y	
202*	Read rate	PR_ASK	2	1	426	15	Y	Y	Y	Y	Y	Y		
344	ETSI UB	PR_ASK	2	4	640	7.5	Y	Y			Y	Y	Y	
345	Read rate	PR_ASK	1.5	4	640	7.5	Y	Y			Y	Y	Y	
223	ETSI LB	PR_ASK	2	2	320	15	Y	Y	Y	Y	Y	Y	Y	Y
222	ETSI LB	PR_ASK	2	2	320	20	Y	Y	Y	Y	Y	Y	Y	Y
241	ETSI LB DRM	PR_ASK	2	4	320	20	Y	Y	Y	Y	Y	Y	Y	Y
244	FCC DRM	PR_ASK	2	4	250	20	Y	Y	Y	Y	Y	Y	Y	Y
285	Sensitivity	PR_ASK	2	8	160	20	Y	Y	Y	Y	Y	Y	Y	Y

(*) Mode 202 currently has measured performance gaps, especially in the ETSI Lower frequency band and at long distances.

Regulatory Test	Value
Show Test parameters	0
Unmodulated carrier	1
Inventory	2
Inventory with tag reports	3
PRBS test	4
ETSI burst	5

3.3.2. Bank ($ba)

Execute or reset the command configured in the requested bank.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
Reset	**-**	-re	0x22
Execute	**-**	-go	0x23
Trigger *	-	-tr	0x24
Multi-bank sequence	Hex byte array	-mb	0x25
Repeat Multi-bank	0 – 1 **(0)**	-rm	0x26
Send Multi-bank summary	**0** – 1 **(0)**	-sm	0x27
MB stop after n rounds	0 – 2^{32}-1 (**0**)	-mr	0x28
MB stop after n tags	0 – 2^{32}-1 (**0**)	-mt	0x29
MB stop after duration (ms)	0 – 2^{32}-1 (**0**)	-md	0x2A
Multi-bank Execute	-	-gm	0x2B
Multi-bank Copy	-	-mc	0x2C

Responses: SC, EC

See Inventory Response for more details.

See Response Headers for ASCII and BINARY field headers.

(*) This is a returned parameter only, shown in the SC line (when enabled), indicating when processing has been started by the hardware trigger line.

Define the sequence of banks to use as an array of two-digit bank numbers. eg *-mb 00010203* to specify banks 0, 1, 2, 3. Banks may be repeated and there may be up to 32 entries.

Note: Multi-bank stop conditions are *only* checked *after* each individual bank has been run.

Multi-bank copy will copy the current bank configuration to all banks listed by -mb

3.3.3. Select ($se)

Configure a G2 Select in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Mask	Hex byte array	-ma	0x22
Mask length (bits)	0 – 255	-ml	0x23
Mask offset (bits)	0 – 2^{14}-1	-mo	0x24
Select target	See below	-st	0x25
Select action	See below	-sa	0x26
Select memory bank	See below	-sb	0x27
Truncate	0 – 1	-tr	0x28

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Select Target	Value
Session 0	**0**
Session 1	1
Session 2	2
Session 3	3
Selected Flag	4

Select Action	Value	Matching Action		Non-matching Action	
		Selected Flag	Session Flag	Selected Flag	Session Flag
Action 000	**0**	**Assert**	**Set A**	**De-assert**	**Set B**
Action 001	1	Assert	Set A	-	-
Action 010	2	-	-	De-assert	Set B
Action 011	3	Toggle	Toggle	-	-
Action 100	4	De-assert	Set B	Assert	Set A
Action 101	5	De-assert	Set B	-	-
Action 110	6	-	-	Assert	Set A
Action 111	7	-	-	Toggle	Toggle

Select Memory Bank	Value
EPC	**1**
TID	2
USER	3

3.3.4. Read ($re)

Configure a G2 Read in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Data memory bank	See below	-db	0x23
Data length (words)	0 – 32	-dl	0x24
Data offset (words)	0 – 2^{16}-1	-do	0x25

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Data Memory Bank	Value
RESERVED	0
EPC	1
TID	2
USER	3

3.3.5. Write ($wr)

Configure a G2 Write in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Data	Hex word array	-da	0x22
Data memory bank	See below	-db	0x23
Data length (words)	1 – 32	-dl	0x24
Data offset (words)	0 – 2^{16}-1	-do	0x25
Bulk encode parameter	**0** - 9	-be	0x26

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Note: If *Data length* is not provided, it will be calculated from *Data* provided

If *Bulk encode* is non-zero, the *Data and Length* will be obtained from the *DataStore ($ds)* at run-time

Data Memory Bank	Value
RESERVED	0
EPC	1
TID	2
USER	3

3.3.6. Kill1 ($k1)

Configure a G2 Kill1 in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Kill password	**0** – 2^{16}-1	-kp	0x22

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

3.3.7. Kill2 ($k2)

Configure a G2 Kill2 in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Kill password	**0** – 2^{16}-1	-kp	0x22

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

3.3.8. Lock ($lo)

Configure a G2 Lock in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Lock Flags	See below	-lf	0x22
Bulk encode parameter	**0** - 9	-be	0x26

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Note: If *Bulk encode* is non-zero, the *Lock Flags* will be obtained from the *DataStore ($ds)* at run-time

Lock Payload	Flag Value
Kill password read write mask	0x00080000
Kill password permalock mask	0x00040000
Access password read write mask	0x00020000
Access password permalock mask	0x00010000
EPC memory write mask	0x00008000
EPC memory permalock mask	0x00004000
TID memory write mask	0x00002000
TID memory permalock mask	0x00001000
File0 memory write mask	0x00000800
File0 memory permalock mask	0x00000400
Kill password read write lock	0x00000200
Kill password permalock	0x00000100
Access password read write lock	0x00000080
Access password permalock	0x00000040
EPC memory write lock	0x00000020
EPC memory permalock	0x00000010
TID memory write lock	0x00000008
TID memory permalock	0x00000004
File0 memory write lock	0x00000002
File0 memory permalock	0x00000001

NB Flag values should be OR'd together

3.3.9. Access ($ac)

Configure a G2 Access in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Access password	**0** – 2^{16}-1	-ap	0x22
Bulk encode parameter	**0** - 9	-be	0x26

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Note: **This command must be sent twice. Upper 16-bits of 32-bit password, then lower 16-bits**

If *Bulk encode* is non-zero, the *Access password* will be obtained from the *DataStore ($ds)* at run-time

3.3.10. BlockWrite ($bw)

Configure a G2 Blockwrite in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Data	Hex word array	-da	0x22
Data memory bank	See below	-db	0x23
Data length (words)	1 – 32	-dl	0x24
Data offset (words)	0 – 2^{32}-1	-do	0x25
Bulk encode parameter	**0** - 9	-be	0x26

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Note: If *Data length* is not provided, it will be calculated from *Data* provided

If *Bulk encode* is non-zero, the *Data and Length* will be obtained from the *DataStore ($ds)* at run-time

Data Memory Bank	Value
RESERVED	0
EPC	1
TID	2
USER	3

3.3.11. BlockPermalock ($bp)

Configure a G2 Read in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Data	Hex word array	-da	0x22
Data memory bank	See below	-db	0x23
Data length (words)	0 – 32	-dl	0x24
Data offset (words)	0 – 2^{32}-1	-do	0x25
Read-Lock	See below	-rl	0x26

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Note: If *Data length* is not provided, it will be calculated from *Data* provided

Data Memory Bank	Value
RESERVED	0
EPC	1
TID	2
USER	3

Read-Lock	Value
Read	0
Lock	1

3.3.12. Authenticate ($au) - *experimental*

Configure a G2 Authenticate in the requested bank / slot.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
G2 slot number	**0** – 9	-gn	0x21
Challenge message	Hex byte array	-cm	0x22
Challenge length (bits)	**0** – 255	-cl	0x23
Send response	**0** – 1	-sr	0x24
Include response length	**0** – 1	-il	0x25
Response length (bits)	**0** – 255	-rl	0x26
CSI	**0** – 255	-cs	0x27

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

3.3.13. Untraceable

Not yet available.

3.3.14. Inventory round ($ir)

Configure the inventory round parameters for the requested bank.

Parameter	Value (default in bold)	Code	Token
Bank number	**0** – 9	-bn	0x20
Initial Q	0 – 15 (**4**)	-iq	0x22
Max Q	0 – 15 (**4**)	-hq	0x23
Min Q	0 – 15 (**4**)	-lq	0x24
Number of min Q cycles	0 – 15 (**1**)	-mc	0x25
Fixed Q	0 – 1 **(1)**	-fq	0x26
Q increase use query	0 – 1 **(0)**	-qi	0x27
Q decrease use query	0 – 1 **(0)**	-qd	0x28
Session	See below **(1)**	-se	0x29
Select	See below **(0)**	-sl	0x2A
Target	0 – 1 **(0)**	-ta	0x2B
Tagfocus	0 – 1 **(0)**	-tf	0x2C
Fast ID	0 – 1 **(0)**	-fi	0x2D
Max queries since valid EPC	0 – 2^8-1 **(0)**	-sv	0x2E
Dual target	0 – 1 **(1)**	-dt	0x2F
Antenna number	1 – 4 **(1)**	-an	0x30
RF mode	See below	-rf	0x31
Power *(100th's dBm)*	10 – 3000 **(3000)**	-db	0x32
Filter to strongest	0 – 1 **(0)**	-fs	0x33
Tag reports	See below **(0x3401)**	-tr	0x34
Stop after n tags	0 – 2^{32}-1 (**0**)	-st	0x35
Stop after n rounds	0 – 2^{32}-1 (**0**)	-sr	0x36
Stop after duration (ms)	0 – 2^{32}-1 (**100**)	-sd	0x37
Pause after selects (ms)	0 – 10 **(0)**	-ps	0x38
Use Data Store during processing	See below **(0)**	-ds	0x39

Responses: SC, EC

See Response Headers for ASCII and BINARY field headers.

Session	Value
Session 0	0
Session 1	**1**
Session 2	2
Session 3	3

Select	Value
Select all	**0**
Select all 2	1
Not asserted	2
Asserted	3

RF Mode	Optimisation	FLM	PIE	Miller	Link Freq	Tari uS	FCC	EUB	EUL	CHI	9	7	5	3
103	Read rate	DSB	1.5	1	640	6.25	Y				Y	Y		
302	Read rate	PR_ASK	2	1	640	7.5	Y	Y			Y	Y		
120	Read rate	DSB	1.5	2	640	6.25	Y				Y	Y	Y	
323	Read rate	PR_ASK	2	2	640	7.5	Y	Y			Y	Y	Y	
202*	Read rate	PR_ASK	2	1	426	15	Y	Y	Y	Y	Y	Y		
344	ETSI UB	PR_ASK	2	4	640	7.5	Y	Y			Y	Y	Y	
345	Read rate	PR_ASK	1.5	4	640	7.5	Y	Y			Y	Y	Y	
223	ETSI LB	PR_ASK	2	2	320	15	Y	Y	Y	Y	Y	Y	Y	Y
222	ETSI LB	PR_ASK	2	2	320	20	Y	Y	Y	Y	Y	Y	Y	Y
241	ETSI LB DRM	PR_ASK	2	4	320	20	Y	Y	Y	Y	Y	Y	Y	Y
244	FCC DRM	PR_ASK	2	4	250	20	Y	Y	Y	Y	Y	Y	Y	Y
285	Sensitivity	PR_ASK	2	8	160	20	Y	Y	Y	Y	Y	Y	Y	Y

(*) Mode 202 currently has measured performance gaps, especially in the ETSI Lower frequency band and at long distances.

Tag Report	Value
EPC *(EP)*	0x0001
TID *(TD)*	0x0002
PC *(PC)*	0x0004
XPC *(XP)*	0x0008
CRC *(CR)*	0x0010
TIMESTAMP *(TS)*	0x0020
RSSI *(RI)*	0x0040
PHASE *(PH)*	0x0080
CHANNEL *(CF)*	0x0100
INDEX *(IX)*	0x0200
BANK HEADER *(BH)*	0x0400
SUMMARY *(SU)*	0x1000
CONDENSED FORMAT	0x2000

NB Above values should be OR’d together

Use Datastore	Value
Use if required by G2 commands	**0**
Use as filter, process once	1
Use as filter, process multi	2

3.3.15. External I2C interface ($i2)

Send a command to an external I2C device connected to GPIO1 (SCL) and GPIO2 (SDA). The clock-rate is fixed at 100kHz

Parameter	Value (default in bold)	Code	Token
Slave 7-bit address	0x00 – 0x7F	-sa	0x20
Transmit data bytes	Hex byte array	-tx	0x23
Receive data length	1 – 64	-rx	0x24
Memory address (8-bit)	0x00 – 0xFF	-ba	0x21
Memory address (16-bit)	0x0000 – 0xFFFF	-wa	0x22
Write memory data bytes	Hex byte array	-wm	0x25
Read memory data length	1 – 64	-rm	0x26

Responses: SC, I2, ME, EC

See Response Headers for ASCII and BINARY field headers.

3.3.16. Data Store ($ds)

Used to pre-load data to be used during inventories to bulk-encode tags. The *$wr, $ac, $lo & $bw* commands may use their *-be* parameters to access data from this store. It could also be used to limit tag processing to a list of known tags.

Parameter	Value (default in bold)	Code	Token
Show Available size	-	-sz	0x20
Release DS memory	-	-rm	0x21
Reset DS	-	-re	0x22
Read DS	See below (**0**)	-rd	0x23
Max Attempts	1 – 63 **(7)**	-ma	0x24
Report Level	See below **(0)**	-rl	0x25
Match EPC	Hex word array	-ep	0x26
Match TID	Hex word array	-td	0x27
Match offset (words)	0-31 **(0)**	-mo	0x28
Parameter 1	Hex word array	-p1	0x29
Parameter 2	Hex word array	-p2	0x2A
Parameter 3	Hex word array	-p3	0x2B
Parameter 4	Hex word array	-p4	0x2C
Parameter 5	Hex word array	-p5	0x2D
Parameter 6	Hex word array	-p6	0x2E
Parameter 7	Hex word array	-p7	0x2F
Parameter 8	Hex word array	-p8	0x30
Parameter 9	Hex word array	-p9	0x31
Clear status bytes	See below **(0)**	-cs	0x32

Responses: SC, SZ, IC, DS, ME, EC

See Response Headers for ASCII and BINARY field headers.

Note: Provide ***EITHER*** -ep ***OR*** -td and as many parameters (-p1 to -p9) as required

The Status byte is used as below:

	B7	B6	B5	B4	B3	B2	B1	B0
Status byte	**Complete**	**Failed**	**Attempts**					

Read Datastore	Value
Show all records	**0**
Show unprocessed	1
Show Completed	2
Show Failed	3

Report Level	Value
Only report tags in Datastore	**0**
Report all tags seen	1

Clear status bytes	Value
Reset failed records	**0**
Reset all records	1

4. COMMAND RESPONSES

4.1. Inventory Response

The data returned during inventory processing is dependent upon how the command bank has been configured, what Gen2 commands have been requested and what Tag Data Reports are enabled.

Headers: SC, BH, TR, EP, IX, TS, CF, CR, PC, RI, G2, EA, EB, WW, DA, SD, SU, SM, XP, EC

See Response Headers for ASCII and BINARY field headers.

The bank header contains:

B	Bank number
A	Antenna number
R	RF mode
P	Power (cdBm)

The summary headers (SU & SM) contain:

T	Number of tags processed
R	Number of inventory rounds
D	Duration of inventory in ms
S	Stop reason (see Appendix 6.6)
P	Approximate tags/second

The Datastore summary header (SD) contains:

T	Number of entries in DataStore
U	Unfinished
C	Complete
F	Failed

Bank summary headers (SM) are sent automatically for multibank. Datastore summary header (SD) is sent automatically when Datastore is used.

This is an example of a response to a *$ba -go* command:

ASCII example (with a Gen2 Read command of 4 words from the start of the Reserved Memory)

```
SC:
BH: B=9, A=1, R=244, P=3000
TR:
EP: 123456780000000000000052
IX: 0
TS: 42302
CF: 927250
CR: FFB7
PC: 3000
RI: -4752
G2: 1 (RD)
EA: 1
EB: 16
DA: 0000000000000000
TR:
EP: 123456780000000000000055
IX: 1
TS: 65769
CF: 927250
CR: 72F5
PC: 3400
RI: -3286
G2: 1 (RD)
```

```
EA: 1
EB: 16
DA: 0000000000000000
.
.
SU: T=7, R=5, D=284, S=1, P=26
EC: 0
```

If **TAG REPORT:*CONDENSED FORMAT*** has been selected, then the headers for each tag are returned on a single line, separated by " | " (0x20, 0x7C, 0x20)

ASCII condensed example

```
SC:
BH: B=9, A=1, R=244, P=3000
TR: | EP: 123456780000000000000052 | IX: 0 | TS: 42302 | CF: 927250 | CR:
FFB7 | PC: 3000 | RI: -4752 | G2: 1 (RD) | EA: 1 | EB: 16 | DA:
0000000000000000
TR: | EP: 123456780000000000000055 | IX: 1 | TS: 65769 | CF: 927250 | CR:
72F5 | PC: 3400 | RI: -3286 | G2: 1 (RD) | EA: 1 | EB: 16 | DA:
0000000000000000
.
.
SU: T=7, R=5, D=284, S=1, P=26
EC: 0
```

4.2. Datastore Response

Headers: SC, DS, EP, IC, SB, SZ, TD, EC

See Response Headers for ASCII and BINARY field headers.

This is an example of a response to a *$ds -rd 0* command

```
SC:
DS: | TD: E2801191200000000000001 | SB: 80 (C)
DS: | TD: E2801191200000000000002 | SB: 81 (C)
DS: | TD: E2801191200000000000003 | SB: 47 (F)
DS: | TD: E2801191200000000000004 | SB: 00 (U)
DS: | TD: E2801191200000000000005 | SB: 03 (U)
EC: 0
```

This is an example of a response to a *$ds -sz* command

```
SC: $ds
SZ: 16339
IC: 5
EC: 0
```

5. RESPONSE HEADERS

5.1. Header codes

Code	Token	Description	Binary Type
AG	0x40	Antenna gain dB	Number
CF	0x41	Channel frequency	Number
CR	0x42	Tag CRC	Number
DA	0x43	Data returned	Array16
EA	0x44	Access response (see Appendix)	Number
EB	0x45	Backscatter response (see Appendix)	Number
EC	0x46	End of command response (see Appendix)	Number
EP	0x47	EPC	Array16
G2	0x48	Gen2 command processed	Number
IX	0x49	Index within inventory	Number
MB	0x4A	Module bootloader version	Array8
ME	0x4B	Information message	String
MF	0x4C	Module firmware version	Array8
MM	0x4D	Manufacturer	String
MS	0x4E	Module serial number	String
MT	0x4F	Module type	String
MU	0x50	Module unique ID	Array32
PC	0x51	Tag PC word	Number
PH	0x52	Phase information	Array16
RB	0x53	Radio bootloader version	String
RF	0x54	Radio firmware version	String
RI	0x55	Tag RSSI value in cdBm	Number
RR	0x56	Regulatory region	Number
RT	0x57	Radio type	Number
RU	0x58	Radio unique ID	Array32
SC	0x59	Start of command	String
SU	0x5A	Inventory summary	Array32
TD	0x5B	TID	Array16
TR	0x5C	Start of Tag Report	N/A
TS	0x5D	Timestamp in µSeconds	Number
WW	0x5E	Words written	Number
XP	0x5F	Extended PC word(s) if present	Array16
BH	0x60	Bank Header	Array16
SM	0x61	Multi-bank summary	Array32
I2	0x62	I2C data response	Array 8
RP	0x63	Region Max power in cdBm	Number
SZ	0x64	Data Store free space in bytes	Number
IC	0x65	Data Store item count	Number
DS	0x66	Start of Data Store record	N/A
SB	0x67	Data Store record status	Number
SD	0x68	Data Store summary	Array16

5.2. Header code types

Type	Description
Number	Numeric value in any valid base
String	Sequence of utf-8
Array8	Sequence of uint8 bytes specified in ASCII hex groups of 2
Array16	Sequence of uint8 bytes specified in ASCII hex groups of 4
Array32	Sequence of uint8 bytes specified in ASCII hex groups of 8

6. APPENDIX

6.1. Length and checksum

When the command format used is binary, then the packet length and checksum are calculated as shown below. The same length and checksum calculations are used for incoming and outgoing messages.

COMMAND BYTES								
<SOH>	Length MSB	Length LSB	T	TLV1	…	TLVn	Check MSB	Check LSB

The 16-bit length includes all the bytes <SOH> to TLVn inclusive (highlighted in orange). This length is sent in big-endian format. It can be clearly seen that with a command token, the minimum length value is 4.

The 16-bit checksum is calculated from the all the bytes <SOH> to TLVn inclusive (highlighted in orange). The algorithm used is Fletcher-16 and the result is sent in big-endian format.

```
// Calculate Fletcher-16 checksum
//
uint16_t CalculateCheckSum(uint8_t *str, uint16_t len)
{
      uint16_t   C0, C1;

      C0 = 0;
      C1 = 0;
      for (int l=0; l<len; l++)
      {
            C0 = (C0 + (uint16_t)*str++) % 255;
            C1 = (C1 + C0) % 255;
      }

      return (C1 << 8) | C0;
}
```

6.2. Access response codes (EA)

Error code	Description
1	OK
2	Bad CRC
3	No reply
4	Invalid reply type
5	Status cover code failed

6.3. Backscatter response codes (EB)

Error code	Description
0	Other
1	Not supported
2	Insufficient privileges
3	Memory overrun
4	Memory locked
5	Crypto suite
6	Command not encapsulated
7	Response buffer overflow
8	Security timeout
11	Insufficient power
15	Non-specific
16	OK

6.4. Command response codes (EC)

Error code	Description
0	OK
1	Timeout (message not completed within 15s)
2	ASCII formatted command too short
3	BINARY formatted command length error (<5 or >510)
4	Checksum error
5	Unknown command code
6	Parameter error
7	Missing parameter
8	Not permitted
9	No valid regulatory region set
10	Unable to start RF processing
11	Ex10 firmware update required
12	I2C error on external interface
13	Data Store: Records must have the same format
14	Data Store: Full
15	Data Store: Record not unique

6.5. GPIO line usage

GPIO	Description
1	External I2C SCL
2	External I2C SDA
3	Inventory Start *
4	Inventory Stop ǂ

(*) If GPIO3 is held to ground and then released, whilst the module is powered-up, the serial port baud rate will be set to 921600, irrespective of any stored setting.

(ǂ) If GPIO4 is held to ground, whilst the module is powered-up, the module will enter bootloader mode, to allow firmware updates.

6.6. Inventory Stop Reasons

Reason code	Description
1	User requested
2	Round limit reached
3	Tag limit reached
4	Duration limit reached
5	RF operation error
6	RF SDK timeout error
7	RF device command error
101	Module too hot
102	Invalid packet received from RF processor
103	Unknown packet received from RF processor
104	No response received from RF processor
105	Multi-bank stop condition

About TSL®

Technology Solutions UK Ltd (TSL®), part of HID Global, is a leading manufacturer of high performance mobile RFID readers used to identify and track products, assets, data or personnel.

For over two decades, TSL® has delivered innovative data capture solutions to Fortune 500 companies around the world using a global network of distributors and system integrators. Specialist in-house teams design all aspects of the finished products and software ecosystems, including electronics, firmware, application development tools, RF design and injection mould tooling.

TSL® is an ISO 9001:2015 certified company.

ISO 9001: 2015

Contact

Address:	Technology Solutions (UK) Limited, Suite A, Loughborough Technology Centre, Epinal Way, Loughborough, Leicestershire, LE11 3GE, United Kingdom.
Telephone:	+44 1509 238248
Fax:	+44 1509 214144
Email:	enquiries@tsl.com
Website:	www.tsl.com

About HID Global

HID Global powers the trusted identities of the world's people, places and things. We make it possible for people to transact safely, work productively and travel freely. Our trusted identity solutions give **people** convenient access to physical and digital **places** and connect **things** that can be identified, verified and tracked digitally. Millions of people around the world use HID products and services to navigate their everyday lives, and billions of things are connected through HID technology. We work with governments, educational institutions, hospitals, financial institutions, industrial businesses and some of the most innovative companies on the planet. Headquartered in Austin, Texas, HID Global has over 4,000 employees worldwide and operates international offices that support more than 100 countries. HID Global is an ASSA ABLOY Group brand.

For more information, visit www.hidglobal.com.

Technology Solutions (UK) Ltd reserves the right to change its products, specifications and services at any time without notice.