Technical Reference Manual

Intermec Scanner Control Protocol

P/N 3-122049-10
revision level 3-122049-10-011

Revisions

Revision	Paragraph affected	Changes
Edition 1.0 18 Sept 2001	Initial release	Created and Issued
Edition 1.1 15 Jan 2002	1-Code 11	1-modified check digit codes (0x00 not possible as parameter) -added length codes
	2-MSI code	2-modified check digit codes (0x00 not possible as parameter)
	3-Beep led indicator	3-added codes for PDF (crackle and flicker)
	4-Trigger setting	4-added spotter beam mode code
	5-Setup Configuration	5-added configuration by label and permanent transparent mode codes
	6-Data editing	6-added data editing explanation and codes
	7-MicroPDF	7-added code mark FID
	8-PDF417	8-corrected switched codes - not transmitted=0x00 and transmitted=0x01
	9-E1022 quick reference default table	9-table taken out and put in E1022 IG-document now general ISCP
	10-Index	10-Index added
	11-Frame Management	11-added new bit information on frame management
	12-Chapter 9, Commands	12-added explanations for each type of command group
	13-Administrator reset factory defaults	13-added explanation and administrator reset factory defaults bar code
	14-Setup Permission Write	14-modified explanation

Contents

1. Introduction

The Intermec Scanner Control Protocol (ISCP) is a secure structured protocol using a frame format to send and receive messages via an RS-232 port between an Intermec scanner and a host system. ISCP allows you to easily configure, retrieve information from or control your scanner directly from the host system.

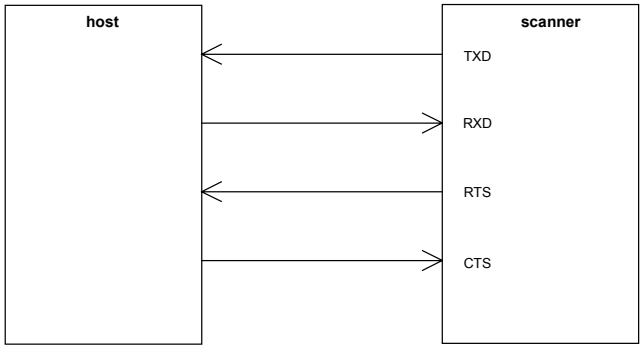
ISCP uses High Level Frames to send commands, information and data. Each High Level Frame is acknowledged to ensure correct reception. Low Level Frames are used to control the flow of High Level Frames. All frames use Data Link Escape (DLE) to avoid frame desynchronization.

The ISCP Technical Reference Manual is designed to provide you with the information necessary to create a host application for use with an Intermec scanner. The first part of this manual gives information on the physical interface, frame format and the types of frames that can be sent. The second part contains all information concerning the different command groups used and gives the hexadecimal values needed to send messages to and from the host.

2. ISCP – Physical interface

2.1. Communication

The scanner communicates with the host using the hardware interface lines as described in the table below.



Signal	Description
TXD	scanner transmission line (output)
RXD	scanner reception line (input)
RTS	when active (*), the scanner requests to transmit
CTS	when active (*), the host authorizes the scanner to transmit

(*) See the following section on RTS/CTS hardware protocol.

2.1.1. Scanner communication parameters

Baud rate

The scanner can be configured to operate from 1200 to 57600 bauds.

Stop bits

1 or 2 stop bits (configurable)

Fixed RS parameters

When using ISCP, the following parameters are fixed:

- 8 bits
- no parity

2.1.2. Host hardware

When receiving data, RTS/CTS hardware protocol may be mandatory depending on the host's hardware.

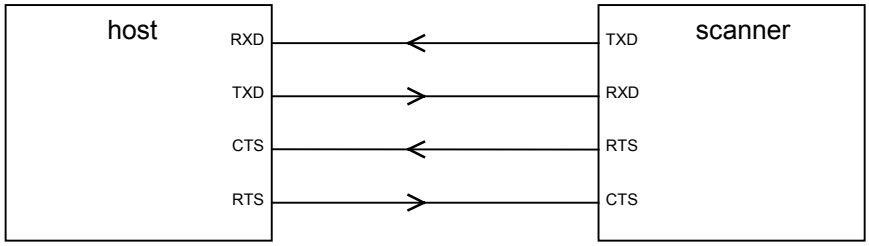
Communication Port	
Full duplex	RTS/CTS hardware protocol NOT necessary
Half duplex	RTS/CTS hardware highly recommended

If the host's hardware is capable of transmitting and receiving messages simultaneously (full duplex), the RTS/CTS hardware protocol is not necessary. However, if it is not capable of managing both at the same time (half duplex), we strongly recommend using the RTS/CTS hardware protocol to avoid losing information.

The scanner is always ready to receive a character from the host on its RXD line so the host does not need to use a hardware protocol when sending data to the scanner. If the host uses RTS/CTS hardware protocol when receiving data, this option must be enabled in the scanner (see the setup groups in section 9.1.21, Serial Interface).

2.2. RTS / CTS hardware protocol

RTS/CTS hardware protocol uses four lines as indicated below:



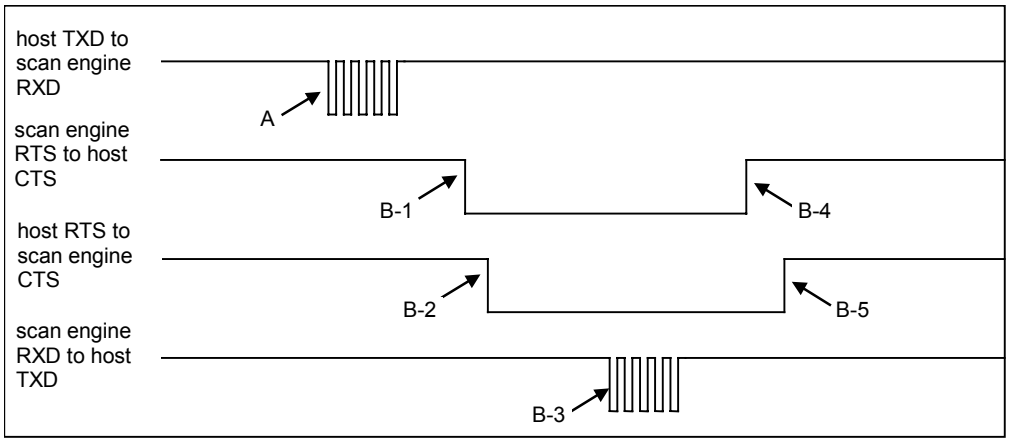
host signals	scanner signals	Description
RXD (input)	TXD (output)	scanner transmits data to the host
TXD (output)	RXD (input)	host transmits data to the scanner
CTS (input)	RTS (output)	scanner asks the host for permission to transmit a character
RTS (output)	CTS (input)	host authorizes the scanner to transmit a character

The scanner:

- requests permission from the host each time it wants to transmit data
- transmits data after receiving permission from the host
- removes the request after transmission of the data
- waits for the host to remove permission before making another request

The host:

- grants permission when the scanner requests to transmit data
- awaits reception of the data
- removes permission after receiving the data



Transmission from host to scanner

- A: The host transmits data to the scanner without using the hardware protocol

Transmission from scanner to host

- B-1: The scanner requests permission to transmit data
- B-2: The host grants permission to the scanner
- B-3: The scanner transmits the data
- B-4: The scanner removes the request
- B-5: The host removes the permission after receiving the data

Note: If RTS/CTS hardware protocol is used, it must be enabled in the scanner (see section 9.1.21, Setup Groups / Serial Interface). "Data" refers to a character or whole message.

3. High level frames

High level frames are used to transmit commands and data (configuration commands, decoded bar codes, control commands, etc.). All high level frames must be acknowledged (low level ACK frame response) to be valid. Refer to Chapter, 4 *Low Level frames* for more information.

3.1. High level frame format

<STX> <SN> <TYPE> <PARM / DATA> <FM> <CHK> <ETX>

Item	Length	Description
<STX>	1 byte	frame delimiter 0x02
<SN>	1 byte	Sequence Number
<TYPE>	1 byte	frame type
<PARM / DATA>	[0..n] bytes	parameters / data
<FM>	1 byte	Frame Management: bit 7 = always 0 bit 6 = Multi-Frame bit (last frame) bit 5 = Multi-Frame bit (first frame) bit 4 = Error bit bit 3 = Restart bit bits 2 to 0 = Frame Number
<CHK>	2 bytes	checksum
<ETX>	1 byte	frame delimiter 0x03

3.1.1. Frame delimiter <STX>

Start character = 0x02

3.1.2. Sequence Number <SN>

The Sequence Number is generated by the host for all frames sent from the host. The host chooses how this number is generated. When the scanner replies to a frame sent by the host, it uses the same Sequence Number.

For all scanner-initiated high level frames (Barcode Data or Event Notification) the Sequence Number is 0.

3.1.3. Frame type <TYPE>

The frame type indicates what kind of frame is being sent. There are many possible frame types.

High level frames from the host:

- Setup Read (SR 0x40)
- Setup Write (SW 0x41)
- Control Command (CCMD 0x42)
- Status Read (STR 0x43)
- Setup Permission Read (SPR 0x44)
- Setup Permission Write (SPW 0x45)

High level frames from the scanner:

- Setup Reply (SRP 0x50)
- Result (RSLT 0x51)
- Status Reply (STRP 0x53)
- Setup Permission Reply (SPRP 0x54)
- Barcode Data (BCD 0x60)
- Event notification (EVT 0x61)
- Setup Barcode Data (SBCD 0x62)

See section 3.3 and 3.4 *High level frame types* for more details.

3.1.4. Parameters / data <PARM / DATA>

The <PARM/DATA> field is used to send commands, information or barcode data.

Most commands consist of a command group, a function identifier and a parameter value.

Result frames only have a function identifier and a parameter value when applicable.

Barcode Data frames contain barcode data in packet format.

See section 3.2*, <PARM / DATA> field format,* for more information.

3.1.5. Frame management <FM>

Frame Management is used by both the host and the scanner. Each manage their own frames independently. The 8 bits of the Frame Management byte are used for different functions.

Frame Management byte:

b7	b6	b5	b4	b3	b2	b1	b0
always 0	multi-frame (last)	multi-frame (first)	error	restart	frame number	frame number	frame number

Bit 7

Bit 7 is always set at zero.

Bits 6 and 5 – Multi-Frame

A long message sent by the scanner may consist of many frames. The Multi-Frame bits (6 and 5) are used to indicate to the receiver whether the message is a single-frame message or a multi-frame message. By using the values 1 and 0, the host is able to know which is the first frame, middle frame(s) and last frame of a message.

Frame of multi-frame message	bit 6	bit 5
first frame of message	0	1
last frame of message	1	0
frames in between the first and last	0	0
single-frame message	1	1

The scanner can not receive multi-frame messages but it can send multi-frame messages. All frames received by the scanner that are NOT single-frame messages (bit 6 = 1 and bit 5 = 1) are discarded. A "not implemented" NAK frame is returned.

The Maximum Reception Frame Size (MRFS) is the maximum frame size (including DLE) that can be processed by the scanner. The frames sent by the host to the scanner must be less than or equal to this value. You can find this value by sending a Status Read. See section 3.3.4, Status Read, for details.

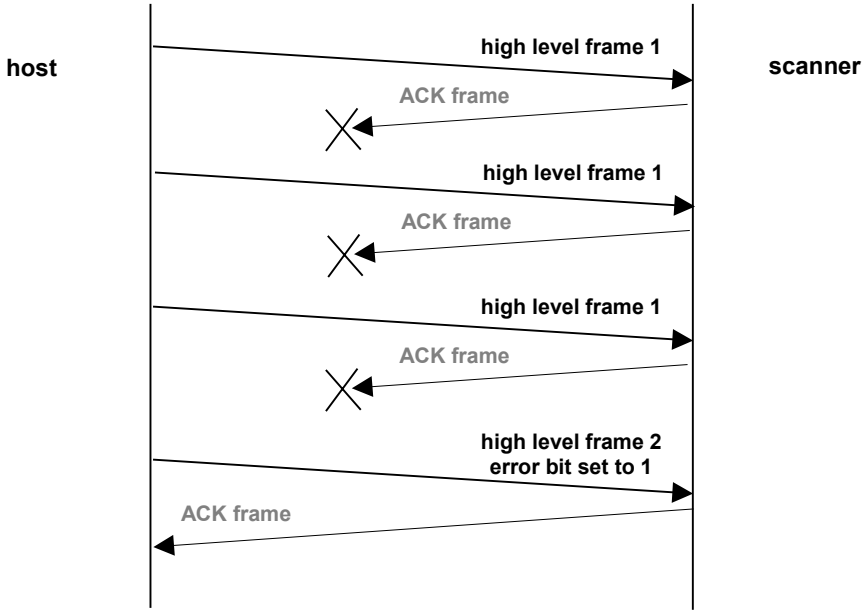
The Maximum Transmission Frame Size (MTFS) is the size of the longest frame (including DLE) that can be sent by the scanner's hardware. The MTFS cannot be changed. However, the size of the frame sent, transmission frame size (TFS), can be configured using the TFS setup parameter. By default, the TFS equals the MTFS but can be set to a smaller value to be compatible with the host. If a message is longer than the TFS value, the scanner will split the message into several frames.

Bit 4 – Error

The error bit is used to indicate that no low level ACK frame was received for the last high level frame sent.

In the following example, the high level frame is sent three times (maximum number of times a frame can be sent) by the host, then abandoned. The next high level frame has the error bit set to 1, indicating that the low level ACK frame was never received for the precedent high level frame.

The error bit is reset to 0 only after the sender has received a low level ACK frame.



Bit 3 – Restart

The Restart bit indicates that the emitter (host or scanner) has been turned on or reset.

The first frame sent after power-on or reset must have the Restart bit set to 1.

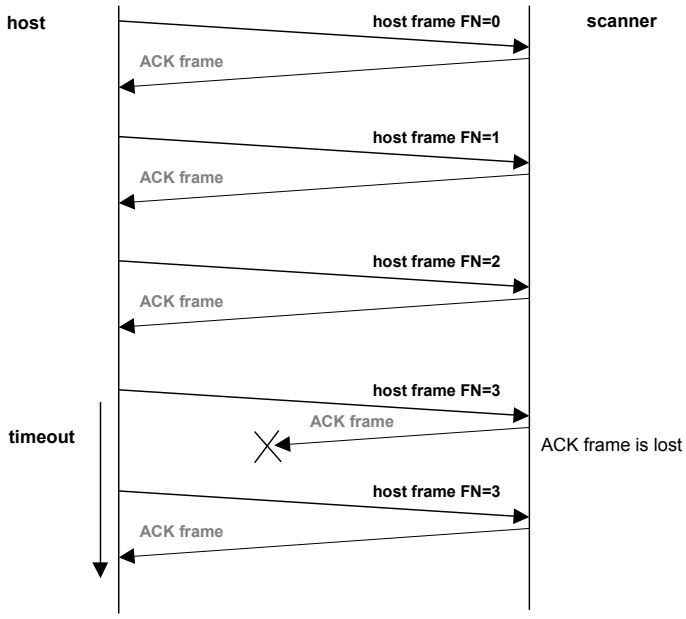
Once the receiver replies to the first frame with an ACK, all of the following frames have the Restart bit set to 0.

If the receiver does not reply to the first frame (Restart bit = 1), the emitter must maintain the Restart bit at 1 in the next frame.

Bits 2 to 0 – Frame Number

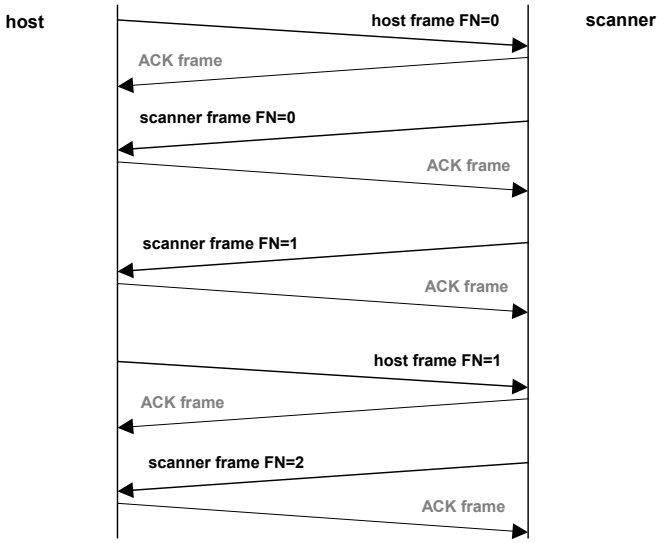
The Frame Number bit is used to indicate if the frame is a new frame or if it has already been sent. This ensures that each frame is correctly received and avoids frame desynchronization. The Frame Number changes when the frame has been acknowledged (ACK) or after being sent 3 times unsuccessfully. The frame number value increments by 1 from 0 to 7 (8 different values possible) and is reset to 0 when the scan engine receives an abort frame or auto-synchronization frame.

If the emitter (host or scanner) sends the same frame again, the Frame Number bit must remain the same.



The above example shows the Frame Number for the frames sent by the host. The ACK following the fourth frame (FN=3) is lost. Since the host has not received an ACK before the timeout, it sends the same frame again using the same Frame Number.

There is no link between the Frame Number of a frame sent by the host and the Frame Number of a frame sent by the scanner.



3.1.6. Checksum

The checksum is the weighted sum of each byte in the frame except for STX, CHK and ETX (weight decreases by 1 for each successive value).

Calculate the modulo 65536 on this sum.

The resulting checksum is a word. The most significant byte of this word must be transmitted first.

Checksum example

Frame	STX	SN	TYPE	PARAM / DATA							FM	CHK	STX
Value	0x02	0x22	0x10	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x60	0x0B74	0x03
Weight		10	9	8	7	6	5	4	3	2	1		
Value * Weight		0x154	0x90	0x208	0x1CE	0x192	0x154	0x114	0xD2	0x8E	0x60		

Sum = 0x154 + 0x90 + 0x208 + 0x1CE + 0x192 + 0x154 + 0x114 + 0XD2 + 0x8E + 0x60 = 0x0B74

Checksum = 0x0B74 modulo 65536 = **0x0B74**

Sample code for checksum calculation:

```
unsigned int   BufferSize;
unsigned char  Buffer[ ];
unsigned int   Check( void )
{
   unsigned char  *Ptr;
   unsigned int   Sum, Temp, Idx;

   Sum = 0;
   Temp = 0;
   if ( BufferSize > 1 )
   {
      Ptr = Buffer;
      for (Idx = 0; Idx < BufferSize; Idx++)
         {
            Temp + = *Ptr++;
            Sum + = Temp;
         }
   }
   return (Sum);
}
```

Note: Data Link Escape (DLE) values are not taken in to account when calculating the checksum.

3.1.7. Frame delimiter <ETX>

Stop character value = 0x03

3.2. <PARM / DATA> field format

All high level frames except Barcode Data and Result frames have a command group, a function identifier and a parameter value (when applicable) in the <PARM/DATA> field. This section describes these three elements and gives examples of how to use them when sending commands.

3.2.1. Structure

As different high level frame types use different sets of commands, commands are organised into groups.

Command groups

- Setup Groups <SG> (Codabar, Serial interface, etc.) are configuration commands.
- Control Groups <CG> (Operating, Decoding, etc.) are commands used to control the scanner.
- Status Groups <STG> (Hardware, etc.) are commands used to communicate the status of the scanner.
- Event Groups <EG> (Decoding, Hardware, etc.) are commands used to notify the host that certain events have taken place.

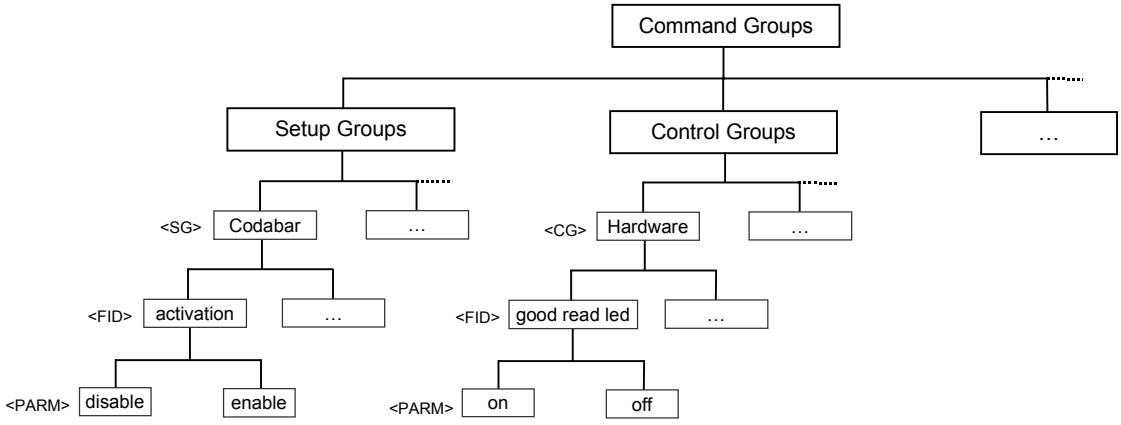
Function identifier

Each Setup Group, Control Group, Status Group and Event Group has different functions (example: Activation, Start of read session, Buzzer). The value (1 byte) that identifies each function is called the function identifier (<FID>).

Parameter

Most functions have a choice of parameters such as Enable or Disable. The size of the parameter varies depending on the type of frame. The parameter can be 1 byte, 2 bytes or an ASCII string. Some functions have no parameters.

Command group structure example:



Note: See Chapter 9, Commands, for all group, function and parameter hexadecimal values.

<PARM/DATA> field example

Frame format for reference:

<STX> <SN>	<TYPE>	<PARM / DATA>	[…]	<FM> <CHK> <ETX>

Setup Write frame example:

	<SW>	<SG> <FID> <PARM>	<SG> <FID> <PARM>	
<STX> <SN>	<0x41>	**<0x40> <0x40> <0x01>**	**<0x42> <0x58> <0x01>**	<FM> <CHK> <ETX>

A Setup Write frame is used to configure the scanner. In this example frame, there are two <PARM/DATA> blocks: The first enables the Codabar symbology and the second enables the start/stop transmission for Code 39.

<PARM/DATA> block 1:

- Setup Group <0x40> = Codabar
- Setup function ID <0x40> = Activation
- Parameter <0x01> = Enable

<PARM/DATA> block 2:

- Setup Group <0x42> = Code 39
- Setup function ID <0x58> = Start/stop transmission
- Parameter <0x01> = Enable

As shown in the example above, you can send many <PARM/DATA> blocks in the same frame (<PARM/DATA> <PARM/DATA> […] […]). This is useful when sending a Setup Write frame (setup parameters) to the scanner.

Note: See sections 3.3 and 3.4 for details on what information is used for each type of High Level frame.

3.2.2. Function identifier encoding

Since the parameter that follows a Function Identifier can be different sizes, the size of the parameter is encoded in bits 7 and 6 of the Function Identifier (FID). This helps to ensure correct reception of each frame whether it has a parameter or not. The table below shows the encoding of bits 7 and 6.

bit 7	bit 6	Parameter size
0	0	no parameter
0	1	1 byte
1	0	2 bytes
1	1	string: the size of the string (2 bytes) follows the identifier

The Function Identifier values are predefined so you will not need to add this value. However, you will need to be aware of how the size of each parameter is managed so the host can also correctly send and receive information.

Note: Function Identifier Encoding does not apply to Setup Permission Write <SPW> and Setup Permission Reply <SPRP> frames.

Example 1 – no parameter:

The predefined <FID> value is 0x02 (Reset Administrator reset factory defaults). When converted to binary, bit 7 = 0 and bit 6 = 0, indicating that no parameter follows.

<CG>	<FID>	<PARM>
0x40	0x02	
Configuration	administrator reset factory defaults	None

Example 2 – parameter = 1 byte:

The predefined <FID> value is 0x40 (Activation). When converted to binary, bit 7 = 0 and bit 6 = 1, indicating that the parameter that follows is 1 byte (0x01 = Enable).

<SG>	<FID>	<PARM>
0x40	0x40	0x01
Codabar	Activation	Enable

Example 3 – parameter is 2 bytes:

The predefined <FID> value is 0x81 (Good read beep duration). When converted to binary, bit 7 = 1 and bit 6 = 0, indicating that the parameter that follows is 2 bytes (0x012C = 300 miliseconds).

<SG>	<FID>	<PARM>
<0x72>	<0x81>	<0x012C>
Beep/Led indicator	Good beep duration	300 miliseconds

Example 4 – parameter is an ASCII string:

The predefined <FID> value is 0xC1 (Message Format). When converted to binary, bit 7 = 1 and bit 6 = 1, indicating that the parameter that follows is an ASCII string. Since the size of the string can vary, the first two bytes of the string give the size (0x00 0x02 = 2 bytes). The ASCII string then follows (<CR> <LF>).

In this case, when you have an ASCII string, you must add the two bytes that indicate the size of the ASCII string.

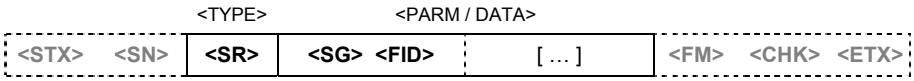
<SG>	<FID>	<PARM>			
<0x60>	<0xC1>	<0x00>	<0x02>	<0x0D>	<0x0A>
Message Format	Postamble	size		<CR>	<LF>

3.3. High level frame types - host to scanner

This section explains the different types (<TYPE>) of high level frames sent from the host to the scanner and details what information is needed in the <PARM / DATA> field for each type.

Frames generated by host				
Type	**Name**	**Hex value**	**Description**	**Scanner reply**
Setup Read	SR	40	host requests value of one or more setup parameters	Setup Reply
Setup Write	SW	41	host requests modification of one or more setup parameter values	Result
Control Command	CCMD	42	host sends one or more control commands to be executed by scanner	Result
Status Read	STR	43	host requests information concerning scanner status	Status Reply
Setup Permission Read	SPR	44	host requests permission status for one or more setup parameters	Setup Permission Reply
Setup Permission Write	SPW	45	host requests permission status modification for one or more setup parameters	Result

3.3.1. Setup Read <SR> 0x40



<SR> A Setup Read frame is used to get information on the configuration of the scanner.

<SG> The Setup Group (SG) is the group of setup parameters (example: Codabar 0x40).

<FID> The Function Identifier (FID) identifies the setup parameter function (example: Activation 0x40).

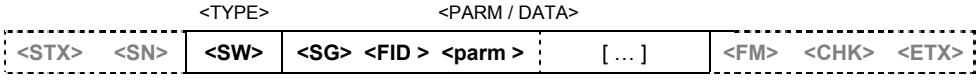
The host sends a Setup Read to know how the scanner is configured. The host can ask for one setup parameter or several at a time by listing the SG/FID pairs in the <PARM/DATA> field. To request all the setup parameters of one group, list the SG followed by 0x00 in the <PARM/DATA> field.

Scanner responses

The scanner first responds with a low level ACK or NAK frame indicating if the Setup Read is correctly received or not. If correctly received, the scanner then sends back the values of the requested setup parameters using a Setup Reply frame <SRP>.

If the host requests the value of an unsupported setup parameter, no error is generated. The unsupported SG/FID pair is ignored. If none of the requested values are supported, the scanner sends an empty Setup Reply <SRP>.

3.3.2. Setup Write <SW> 0x41



<SW> A Setup Write frame is used to send configuration commands to the scanner.

<SG> The Setup Group (SG) is the group of setup parameters (example: Codabar 0x40).

<FID> The Function Identifier (FID) identifies the setup parameter function (example: Activation 0x40).

<parm> The value of the parameter (example: disable 0x00 or enable 0x01)

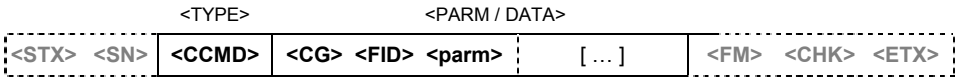
The host sends a Setup Write to configure the scanner or modify the current configuration. The host can change one setup parameter or several at a time by listing the SG/FID/parameter triplets in the <PARM/DATA> field.

Scanner responses

The scanner first responds with a low level ACK or NAK frame indicating if the Setup Write was correctly received or not. When the Setup Write is correctly received, the scanner then informs the host that the changes have been applied by replying with a Result frame.

If a Setup Write frame contains valid and invalid setup modification requests, the valid modifications are executed and the scanner replies with a Result frame containing only the invalid requests. Thanks to this behavior, the host can use the same setup file for different products.

3.3.3. Control Command <CCMD> 0x42



<CCMD> A Control Command is used to control the scanner

<CG> The Control Group (CG) is the group of control commands (example: Decoding 0x20).

<FID> The Function Identifier (FID) identifies the command function (example: Decode 0x40).

<parm> The value of the parameter (example: Decode Off 0x00 or Decode On 0x01)

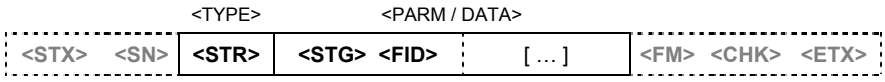
The host can request the execution of one control command or several at a time by listing the CG/FID/parm triplets in the <PARM/DATA> field. For certain Control Commands there is no parameter (<parm>), only the <CG> and <FID>.

Scanner responses

The scanner first responds with a low level ACK or NAK frame indicating if the Control Command was correctly received or not. When the command has been received and executed, the scanner then informs the host that the control commands have been executed by replying with a Result frame.

If a control command frame contains valid and invalid control commands, NO PART of the control command is executed and a Result frame containing the invalid control commands is sent back.

3.3.4. Status Read <STR> 0x43



<STR> A Status Read is used to find out information on the status of certain parameters in the scanner.

<STG> The Status Group (STG) is the group of status parameters (example: Hardware 0x30).

<FID> The Function Identifier (FID) identifies the status parameter function (example: Firmware version 0xC0).

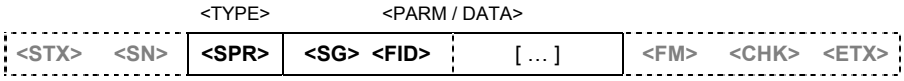
The host sends a Status Read to know the value of certain status parameters in the scanner. The host can request the value of one status parameter or several at time by listing the STG/FID pairs in the <PARM/DATA> field. To request all the status parameters of a group, list the STG followed by 0x00 in the <PARM/DATA> field.

Scanner responses

The scanner first responds with a low level ACK or NAK frame indicating if the Status read was correctly received or not. If correctly received, the scanner then sends the values of the requested setup parameter back using a Status Reply frame <STRP>.

If the host requests the value of an unsupported status parameter, no error is generated. The STG/FID pair of the unsupported parameter is ignored. If none of the requested values are supported, the scanner sends an empty Status Reply <STRP>.

3.3.5. Setup Permission Read <SPR> 0x44



<SPR> A Setup Permission Read frame is used to find out what setup parameters can or cannot be changed by reading configuration bar codes.

<SG> The Setup Group (SG) is the group of setup parameters (example: Codabar 0x40).

<FID> The Function Identifier (FID) identifies the setup parameter function (example: Activation 0x40).

Setup Permissions allow the host to prevent users from modifying the scanner's setup parameters when reading configuration bar codes (see section 3.3.6, *Setup Permission Write* <*SPW*> in this chapter).

The host sends a Setup Permission Read asking what the permission status is for certain setup parameters in the scanner. The host can ask what the permission status is for one setup parameter or several at a time by listing the SG/FID pairs in the <PARM/DATA> field. To request all the setup parameters of one group, list the SG followed by 0x00 in the <PARM/DATA> field.

Scanner responses

The scanner first responds with a low level ACK or NAK frame indicating if the Setup permission read was correctly received or not. If correctly received, the scanner sends back the permission status of the requested setup parameters using a Setup Permission Reply frame <SPRP>.

If the host requests the permission status of an unsupported setup parameter, no error is generated. The unsupported SG/FID pair is ignored. If none of the setup parameters are supported, the scanner sends an empty Setup Permission Reply <SPRP>.

3.3.6. Setup Permission Write <SPW> 0x45

<SPW>	A Setup Permission Write frame is used to lock or unlock access to modify setup parameters when reading configuration bar codes.
<SG>	The Setup Group (SG) is the group of setup parameters (example: Codabar 0x40).
<FID>	The Function Identifier (FID) identifies the setup function (example: Activation 0x40). Function identifier encoding does not apply to this frame (bits 7 and 6).
<permission>	The permission value (0x00 = unlocked: configuration can be modified by reading a bar code and 0x01 = locked: modification impossible by reading a bar code).

Setup Permissions allow the host to prevent users from modifying the scanner's setup parameters when reading configuration bar codes. When the host sends a Setup Permission Write, it locks (0x01) or unlocks (0x00) one or more parameters in the scanner. When a parameter is locked, the user cannot modify the parameter setting by reading a configuration bar code. This helps to avoid unnecessary or accidental changes in the scanner's configuration. However, the host can always modify any of the scanner parameters by sending an ISCP command.

The host can change the permission status of one setup parameter or several at a time by listing the SG/FID/permission triplets in the <PARM/DATA> field.

Scanner responses

The scanner first responds with a low level ACK or NAK frame indicating if the Setup permission write was correctly received or not. If correctly received, the scanner informs the host that the changes have been done by replying with a Result frame.

If a Setup Permission Write frame contains valid and invalid setup parameter codes (SG/FID), the permission status is modified for the valid codes and the invalid codes are sent back by the scanner in a Result frame.

*Note: **Resetting Factory Defaults and Permissions** – The Reset Factory Defaults bar code (available in EasySet) is systematically refused when read by the scanner if one or more of the scanner's parameters are locked. However, if the scanner reads the **Administrator** Reset Factory Defaults bar code (only available in this manual), all parameters are reset – including locked parameters. See Reset Factory Defaults in chapter 10.2, Permissions and configuration bar codes.*

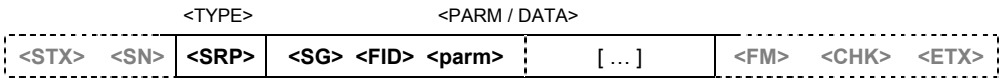
3.4. High level frame types - scanner to host

This section explains the different types (<TYPE>) of high level frames sent from the scanner to the host and details what information is needed in the <PARM / DATA> field for each type.

Frames generated by the scanner				
Type	**ID**	**Hex value**	**Description**	**Host reply**
Setup Reply	SRP	50	scanner sends setup parameter values requested by host	none
Result	RSLT	51	scanner informs host of result of setup modifications or control commands (error or operation done)	none
Status Reply	STRP	53	scanner sends status information requested by host	none
Setup Permission Reply	SPRP	54	scanner sends setup permission requested by host	none
Barcode Data*	BCD	60	scanner sends barcode data to host in packet format	none
Event Notification*	EVT	61	scanner informs host that an event has occurred	none
Setup Barcode Data*	SBCD	62	scanner sends setup barcode data to host in packet format.	none

*(*)Note: For these frame types to be sent by the scanner, certain parameters must be activated accordingly.*

3.4.1. Setup Reply <SRP> 0x50



<SRP> A Setup Reply is used to send to the host information about the configuration of the scanner. This frame is sent in response to a Setup Read <SR>.

<SG> The Setup Group (SG) is the group of setup parameters that the host requested in the Setup Read (example: Codabar 0x40).

<FID> The Function Identifier (FID) identifies which function the host requested in the Setup Read (example: Activation 0x40).

<parm> The scanner sends the value of the parameter that the host requested according to the scanner's configuration (example: disabled 0x00 or enabled 0x01, etc.)

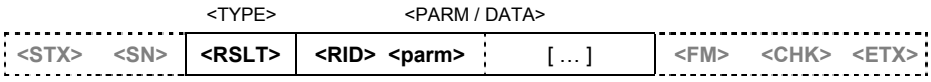
After receiving a Setup Read from the host, the scanner replies by sending the requested setup parameter values to the host in a Setup Reply frame <SRP>.

If the host requests the value of an unsupported setup parameter, no error is generated. The SG/FID pair of the unsupported parameter is ignored. If none of the requested values are supported, the scanner sends an empty Setup Reply <SRP>.

Host response

The host responds with a low level ACK or NAK frame indicating if the Setup Reply was correctly received or not.

3.4.2. Result <RSLT> 0x51



<RSLT> Informs the host of the result after processing a frame.

<RID> The Result Identifier (RID) identifies which result frame the scanner is sending to the host (example: DNE 0x00).

<parm> The scanner may or may not send parameters depending on the kind of result frame sent. See chart below.

All Result frames, except Unknown Frame Type, are sent in response to three different kinds of frames: Setup Write, Control Command and Setup Permission Write. The Unknown Frame Type result frame can be sent in response to any type of frame.

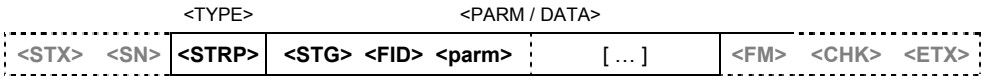
The scanner sends back one of the following result frames to let the host know if and how the frame has been processed.

<RID>	Value (hex)	Name	<parm>	Description
Done	0x00	DNE	none	all setup changes requested by a Setup Write have been done all requests in a Control Command have been executed all permissions changes in a Setup Permission Write have been done
Unknown Frame Type	0x01	UFT	none	the type of high level frame received is unknown to the scanner
Group Unknown	0x81	GU	<SG> <FID>	one or more groups requested by the host are invalid
			<CG> <FID>	
			<STG> <FID>	
Identifier Unknown	0x82	IDN	<SG> <FID>	one or more identifiers requested by the host are invalid
			<CG> <FID>	
			<STG> <FID>	
Invalid Parameter	0x83	IPM	<SG> <FID>	one or more parameters requested by the host are invalid
			<CG> <FID>	

Host response

The host responds with a low level ACK or NAK frame indicating if the Result frame was correctly received or not.

3.4.3. Status Reply <STRP> 0x53



<STRP> A Status Reply frame is used to send status information to the host. This frame is sent in response to a Status Read <STR>.

<STG> The Status Group (STG) is the group of status parameters (example: Hardware 0x30).

<FID> The Function Identifier (FID) identifies the status function (example: Firmware version 0xC0).

<parm> The scanner sends the value of the parameter that the host requested according to the status of the scanner (example: Firmware version xxx.)

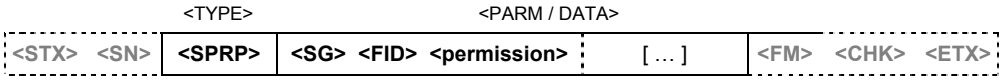
After receiving a Status Read <STR> from the host, the scanner replies by sending the requested status parameter values to the host.

If the host requests the value of an unsupported status parameter, no error is generated. The STG/FID pair of the unsupported parameter is ignored. If none of the requested values are supported, the scanner sends an empty Status Reply <STRP>.

Host response

The host responds with a low level ACK or NAK frame indicating if the Status Reply frame was correctly received or not.

3.4.4. Setup Permission Reply <SPRP> 0x54



<SPRP> A Setup Permission Reply informs the host what the permission status is for certain setup parameters. This frame is sent in response to a Setup Permission Read <SPR>.

<SG> The Setup Group (SG) is the group of setup parameters (example: Codabar 0x40).

<FID> The Function Identifier (FID) identifies the setup function (example: Activation 0x40). Function identifier encoding does not apply to this frame (bits 7 and 6).

<permission> The permission value (0x00 = configuration can be modified by reading a bar code and 0x01 = modification impossible by reading a bar code).

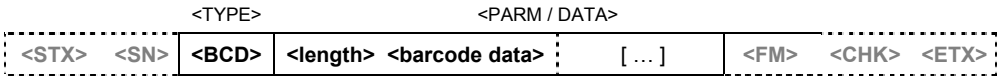
After receiving a Setup Permission Read <SPR> from the host, the scanner replies by sending the permission status of the requested setup parameters.

If the host requests the permission status of an unsupported setup parameter, no error is generated. The unsupported SG/FID pair is ignored. If none of the SG/FID codes are supported, the scanner sends an empty Setup Permission Reply.

Host response

The host responds with a low level ACK or NAK frame indicating if the Setup Permission Reply frame was correctly received or not.

3.4.5. Barcode Data <BCD> 0x60



<BCD>	This frame is used when barcode data is sent in packet format.
<length>	The length (1 word) of the barcode data is specified (number of bytes).
<barcode data>	Decoded barcode data.

A Barcode Data frame is a scanner-initiated frame. When the scanner reads a bar code, data is sent from the scanner to the host. This data can be sent in packet format or raw format. The format must be configured in the ISCP parameters (see section 9.1.23., ISCP parameters). Packet format sends data using the Barcode Data frame. When data is sent in raw format, no frame is used.

The Barcode Data frame differs from other high level frames. It does not have a group identification or function identification in the <PARM / DATA> field.

The following is an example of the Barcode Data frame the host would receive when reading an EAN 8 bar code with a postamble:

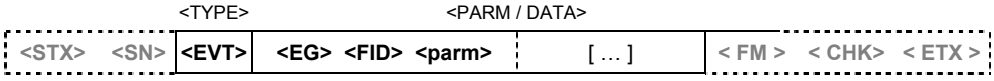
<STX>	<SN>	**<BCD>**	**<length>**	**<PARM / DATA>**	<FM>	<CHK>	<ETX>
02	00	**60**	**00 0A**	**30 31 32 33 34 35 36 35 0D 0A**	67	12 3A	03

Note: As Barcode Data frames are scanner-initiated frames, the <SN> is always 0x00.

Host response

The host responds with a low level ACK or NAK frame indicating if the Barcode Data frame was correctly received or not.

3.4.6. Event Notification <EVT> 0x61



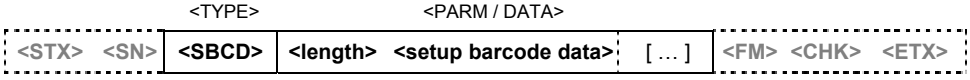
<EVT>	Event notifications let the host know when certain events have taken place.
<EG>	The Event Group (EG) is the group of events (example: Configuration 0x40).
<FID>	The Function Identifier (FID) identifies the event function (example: Setup modification 0x80).
<parm>	The scanner specifies which parameter has been affected by the event (example: <SG> <FID>).

An Event notification is a scanner-initiated frame. It is used to inform the host when certain events have taken place such as an unsuccessful decoding, etc. Event notifications can only be sent when the scanner is configured to send data in packet format.

Event notifications can be configured. If you want the scanner to send event notifications to the host, you must first enable each event desired when setting up your scanner. See section *9.1.23, ISCP parameters).*

Note: As Even Notification frames are scanner-initiated frames, the <SN> is always 0x00.

3.4.7. Setup Barcode Data <SBCD> 0x62



<SBCD>	This frame is used to send configuration barcode data via the scanner to the host in packet format.
<length>	The length (1 word) of the configuration barcode data is specified (number of bytes).
<setup barcode data>	Decoded configuration barcode data.

A Setup Barcode Data frame is a scanner-initiated frame. When the scanner reads a configuration bar code and the Permanent Transparent Mode function is enabled (see section 9.1.20 in Setup Groups), setup data is sent from the scanner to the host. This data can be sent in packet format or raw format. The format must be configured in the ISCP parameters (see section 9.1.23., ISCP parameters). Packet format sends data using the Setup Barcode Data frame. When data is sent in raw format, no frame is used.

Like the Barcode Data frame, the Setup Barcode Data differs from other high level frames. It does not have a group identification or function identification in the <PARM / DATA> field.

Note: As Barcode Data frames are scanner-initiated frames, the <SN> is always 0x00.

Host response

The host responds with a low level ACK or NAK frame indicating if the Setup Barcode Data frame was correctly received or not.

3.5. Data Link Escape (DLE) for high level frames

To avoid frame desynchronization, DLE encoding is applied if the following values are encountered inside the formatted frame (not including the frame delimiters STX/ETX):

- 0x02 (STX)
- 0x03 (ETX)
- 0x10 (DLE)

DLE encoding:

- the DLE character (0x10) is inserted before the ambiguous character
- 0x40 is added to the value of the ambiguous character

This method simplifies locating the frame delimiters because the ambiguous character cannot be mistaken for a delimiter.

Note: DLE values are NEVER taken in to account when calculating the checksum, ONLY when sending the values.

Example 1: 0x02 in the <PARM/DATA> field and in the checksum

Before DLE encoding:

<STX>	<SN>	<CCMD>	<CG>	<FID>	<FM>	<CHK>	<ETX>
02	32	42	40	**02**	68	**02** D4	03

As you can see in this Control Command (administrator reset factory defaults) the value 0x02 appears as the <FID> and in the checksum. DLE must be applied so that the host understands that these characters are not the <STX>.

Note: The checksum is calculated BEFORE applying DLE to the Control Command frame.

After DLE encoding:

<STX>	<SN>	<CCMD>	<CG>	<FID>	<FM>	<CHK>	<ETX>
02	32	42	40	**10 42**	68	**10 42** D4	03

These are the actual values that will be sent.

4. Low Level frames

These frames are used to control the flow of high level frames.

4.1. Low level frame format

<STX> <SN> <TYPE> <PARM> <ETX>

Item	Length	Description
<STX>	1 byte	frame delimiter 0x02
<SN>	1 byte	Sequence Number
<TYPE>	1 byte	frame type (see section *Low Level Frame Types* in this chapter)
<PARAMETER>	1 byte	for all TYPES of frames, PARM = 0x00 except for NAK frames
<ETX>	1 byte	frame delimiter 0x03

4.1.1. Frame delimiter <STX>

Start character = 0x02.

4.1.2. Sequence Number <SN>

The Sequence Number is generated by the host for all frames sent from the host. When the scanner replies to a frame sent by the host, it uses the same Sequence Number.

4.1.3. Type of frame <TYPE>

The type of frame indicates what kind of frame is being sent. There are four types frames: ACK, NAK, BUSY and RESEND. Each type of frame is explained in the following section.

4.1.4. Parameter <PARM>

The value in the <PARM> field is used to give the different parameters possible for the Type of frame sent.

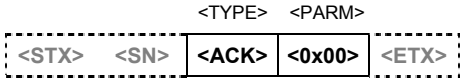
4.1.5. Frame delimiter <ETX>

Stop character value = 0x03.

4.2. Low level frame types

Type	Value	Parameter	Description	Sent by
ACK	0x06	0x00	frame format is correct frame will be processed	host or scanner
NAK	0x15	multi-frame error (MF) 0x20	frame format is incorrect *the Multi-Frame bits are not in accordance with those of the previous frame received* frame will be discarded	host
		frame number error (FN) 0x30	frame format is incorrect *frame number bit is not in accordance with the previous frame received* frame will be discarded	host or scanner
		not implemented 0x40	host sends a multi-frame message to scanner *frame format is correct, but the scanner is not able to process a multi-frame message* frame will be discarded	scanner
		bad checksum 0x50	checksum is not correct	scanner
BUSY	0x1B	0x00	frame format is correct, but scanner is not able to process frame received frame will be discarded	scanner
RESEND	0x05	0x00	host asks scanner to send last frame again	host

4.2.1. ACK frame 0x06



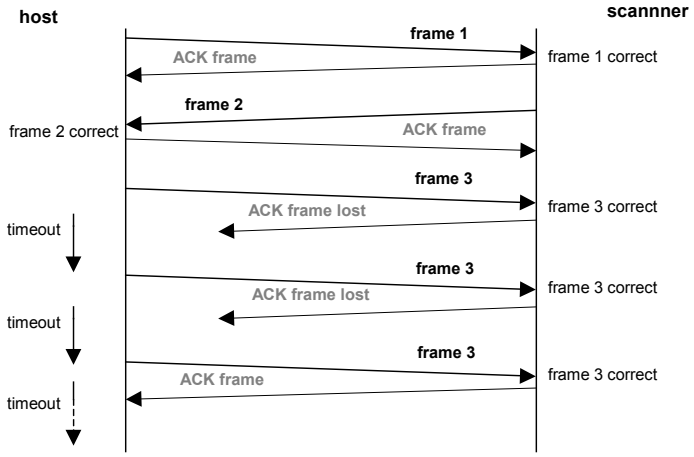
Each high level frame received must be checked:

- STX / ETX delimiters
- Multi-Frame bits
- Frame Number bit
- checksum

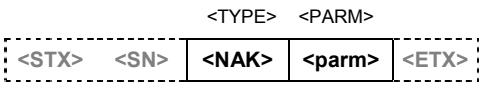
When the format of the high level frame received is correct, a low level ACK frame is returned (host or scanner).

If no low level ACK frame is received before the end of the timeout, the frame is sent again.

Each frame can be resent two times.



4.2.2. NAK frame 0x15

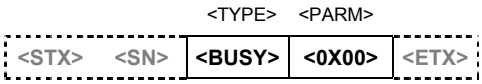


When the format of the high level frame received is incorrect, a low level NAK frame is returned. There are four kinds of NAK error frames and each has its own code:

- bad Multi-Frame bits
- bad Frame Number bit
- not implemented
- bad checksum

A low level NAK frame is NOT sent when the STX or ETX is missing.

4.2.3. BUSY frame 0x1B

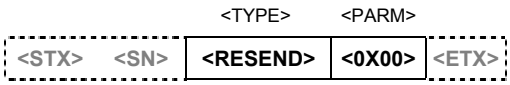


Messages from the scanner take priority over messages from the host.

If the host sends a message (high level frame) when the scanner transmits a bar code or an event notification (high level frame), the message from the host is discarded. After sending the barcode data or event notification, the scanner sends a Busy frame (instead of an ACK frame) to notify the host that its message cannot be processed.

The scanner then waits for the acknowledgment of its high level frame (ACK/NAK frame) or a Resend frame if the scanner's message has been lost by the host.

4.2.4. RESEND frame 0x05



Using this low level frame, the host asks the scanner to send the last high level frame again. The scanner resends only high level frames that have not been acknowledged by the host.

4.3. Data Link Escape (DLE) for low level frames

To avoid frame desynchronization, DLE encoding is applied if the following values are encountered inside the formatted frame (not including the frame delimiters STX/ETX) :

- 0x02 (STX)
- 0x03 (ETX)
- 0x10 (DLE)

DLE encoding:

- the DLE character (0x10) is inserted before the ambiguous character.
- 0x40 is added to the value of the ambiguous character.

This method simplifies locating the frame delimiters because the ambiguous character cannot be mistaken for a delimiter.

Example: 0x02 in the <Sequence Number> field

Before DLE encoding:

<STX>	<SN>	<ACK>	<PARM>	<ETX>
02	**02**	06	00	03

As you can see in this Low Level ACK frame the value 0x02 appears as the <SN>. DLE must be applied so that the receiver understands this character is not the <STX>.

After DLE encoding:

<STX>	<SN>	<ACK>	<PARM>	<ETX>
02	**10 42**	06	00	03

5. Special frames

5.1. Abort / Abort Done

An Abort frame <ABT> is sent by the host to ask the scanner to stop all operations in progress. When the operation has been aborted, the scanner responds to the host by sending an Abort Done frame <ABTD>. After the Abort Done frame is sent, the scanner automatically enters Silent Mode (explained below).

5.1.1. Abort / Abort Done frame formats

Abort frame:

<STX> <SN> <ABT> <0x00> <ETX>

Type	Value	Parameter	Description	Sent by
<ABT>	0x18	0x00	Abort - host asks the scanner to abort an operation in progress	host

All operations can be aborted except when writing in non-volatile memory (Setup Write Setup Permission Write and certain Control Commands). If an Abort frame is received during the actual writing of parameters in non-volatile memory, the abort frame is lost and the setup parameters are modified.

Abort frame done:

<STX> <SN> <ABTD> <0x00> <ETX>

Type	Value	Parameter	Description	Sent by
<ABTD>	0x04	0x00	Abort Done - scanner notifies the host that an operation has been aborted	scanner

After the host receives the Abort Done frame, the host must wait 20ms before sending the next frame.

5.1.2. Silent mode

After sending the Abort Done frame, the scanner automatically enters silent mode. When the scanner is in silent mode, it can only receive and respond to commands received directly from the host. The host has total control over the scanner. This mode can be useful to avoid frame contention when configuring the scanner.

Silent mode is a temporary mode. The scanner exits silent mode when it receives an Exit Silent Mode control command, when the scanner is turned off or when the host sends a Decode On command.

5.2. Auto-synchronization

The Auto-synchronization frame allows the host to establish communication with the scanner no matter how it is configured. This frame automatically puts the scanner in temporary ISCP mode. When in temporary ISCP mode, you can activate the permanent ISCP mode or simply modify the scanner's configuration, exit temporary ISCP mode and return to the scanner's normal configuration.

The Auto-synchronization frame can be used to find the scanner's current parameters for communication and/or activate ISCP mode if not already activated.

5.2.1. Auto-synchronization / auto-synchronization done frame formats

Auto-synchronization frame:

<STX> <SN> <AUTOSYNC> <PARM> <ETX>

Type	Value	Parameter	Description	Sent by
<AUTOSYNC>	0x16	0x00	no RTS/CTS hardware protocol	host
		0x01	RTS/CTS hardware protocol on each character	
		0x02	RTS/CTS hardware protocol on whole message	

The Auto-synchonization frame tells the scanner to:

- abort all current operations
- set the data bits to 8 and parity to none (ISCP fixed parameters)
- set the RTS/CTS hardware protocol parameters (see table above)
- activate temporary ISCP mode

Since the host may not be aware of the scanner's parameters for communication (baud rate, parity and data bits) the Auto-sync frame must be sent using all possible parameters (including 7 data bits). When the frame has been sent with the correct parameters, the scanner responds with an Auto-synchronization done frame.

Note: Try sending the Auto-sync frame first using the scanner default parameters – generally 19200; 8; none. If the scanner does not respond, then try all other possible configurations. ATTENTION: Since this frame can be sent with 7 data bits, the sequence number <SN> must be less than 128.

Auto-synchronization done frame:

<STX> <SN> <AUTOSYNCD> <0x00> <EXT>

Type	Value	Parameter	Description	Sent by
<AUTOSYNCD>	0x0C	0x00	auto-synchronization done	scanner

When the scanner receives the Auto-sync frame from the host with the correct baud rate, parity and data bits, the scanner does these three things in the following order:

1. Aborts all current operations except when writing in non-volatile memory (Setup Write, Setup Permission Write and certain Control Commands) and enters Silent mode (as explained in the previous section).
2. Responds with an Auto-synchronization done frame using the same parameters for communication plus the new RTS/CTS mode (if applicable). The RTS/CTS mode requested by the host is activated immediately so that no characters are lost when sending the Auto-sync done frame. If RTS/CTS is activated, the timeout = 1 second.
3. Enters the temporary ISCP mode and the parameters for communication are changed: 8 data bits and no parity (fixed ISCP parameters).

The scanner can now be configured or controlled by the host using ISCP commands.

Any configuration modifications made by sending a Setup Write at this time will immediately be taken in to account and become permanent. For example, if ISCP mode is activated by sending a Setup Write, the scanner is no longer in temporary ISCP mode.

5.2.2. Temporary ISCP mode

Exiting temporary ISCP mode

You can exit temporary ISCP mode in three ways:

- turning off and turning back on the scanner
- sending a hardware/reset control command (ATTENTION: not reset factory defaults or administrator reset factory defaults)
- sending an ISCP protocol none Setup Write command

After exiting temporary ISCP mode, the scanners returns to its original configuration before receiving the Auto-sync frame from the host EXCEPT if any parameters have been modified while in temporary ISCP mode. All parameters changes made while in temporary ISCP mode are kept as permanent.

Silent mode and temporary ISCP mode

It is possible for the scanner to be in Silent mode when temporary ISCP mode is active. You can exit silent mode **without** exiting temporary ISCP mode. This can be useful if you want to be able to read bar codes, etc. while in temporary ISCP mode. The scanner exits silent mode when:

- it receives a Decode On control command
- it receives an Exit Silent Mode control command

5.3. Data Link Escape (DLE) for special frames

To avoid frame desynchronization, DLE encoding is applied if the following values are encountered inside the formatted frame (not including the frame delimiters STX/ETX) :

- 0x02 (STX)
- 0x03 (ETX)
- 0x10 (DLE)

DLE encoding:

- the DLE character (0x10) is inserted before the ambiguous character.
- 0x40 is added to the value of the ambiguous character.

This method simplifies locating the frame delimiters because the ambiguous character can not be mistaken for a delimiter.

<STX> <SN> <TYPE> <PARM> <0x03>

Example: 0x02 in the <Sequence Number> field

Before DLE encoding:

<STX>	<SN>	<ABT>	<PARM>	<ETX>
02	**02**	18	00	03

As you can see in this Special Abort frame, the value 0x02 appears as the <SN>. DLE must be applied so that the receiver understands this character is not the <STX>.

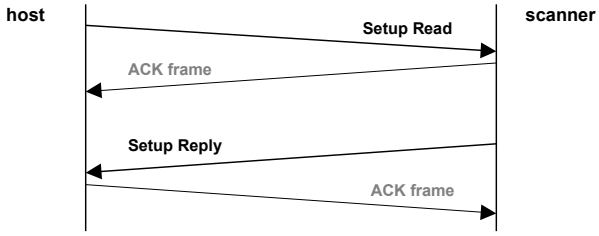
After DLE encoding:

<STX>	<SN>	<ABT>	<PARM>	<ETX>
02	**10 42**	18	00	03

6. Frame contention

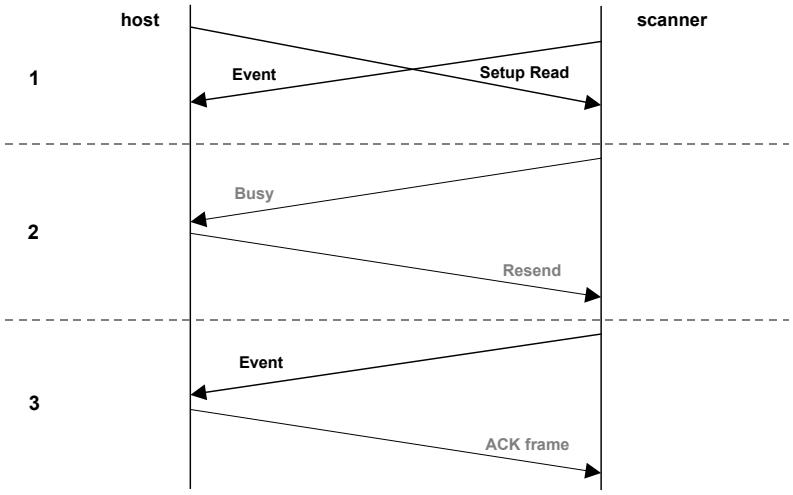
6.1. Transmission without RTS/CTS handshaking

No contention



Contention – scanner has priority

Scanner-initiated frames (Events and Barcode Data) have priority over all frame types sent by the host except the Abort frame. The following example shows a situation where the scanner has priority over the host.



1. Host and scanner send high level frames at the same time.

   The host ignores the Event as it is waiting for an ACK/NAK for the Setup Read.

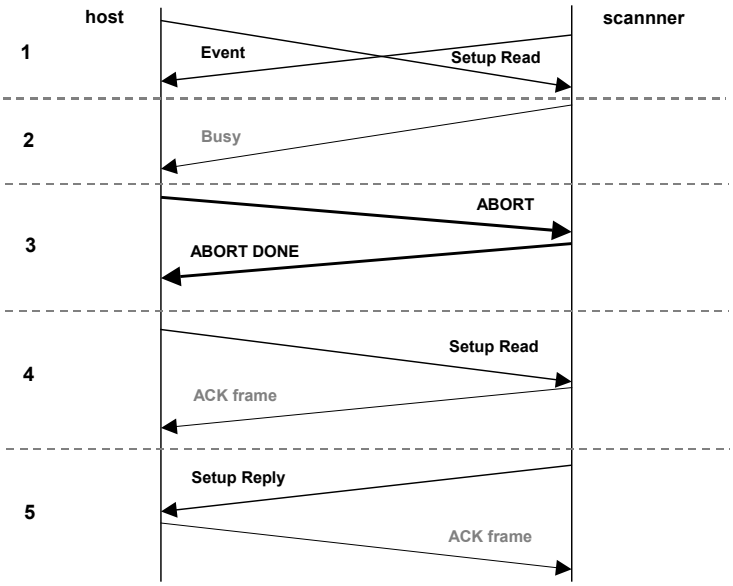
   The scanner cannot process the Setup Read as it is waiting for an ACK/NAK for the Event (the Setup Read is discarded).

2. The scanner sends Busy to tell the host that it cannot not process the Setup Read and has not received acknowledgment for the Event.

   The host sends a Resend to ask the scanner to send the same frame again.

3. The scanner resends the Event and the host responds with ACK.

Contention – host aborts the scanner's transmission



1. Host and scanner send high level frames at the same time.

   The host ignores the Event as it is waiting for an ACK/NAK for the Setup Read.

   The scanner cannot process the Setup Read as it is waiting for an ACK/NAK for the Event (the Setup Read is discarded).

2. The scanner sends Busy to tell the host that it cannot not process the Setup Read and has not received acknowledgment for the Event.

3. Host sends Abort to take priority away from the scanner to avoid any further contention.

   The scanner stops waiting for acknowledgement of the Event frame and sends an Abort Done to the host.

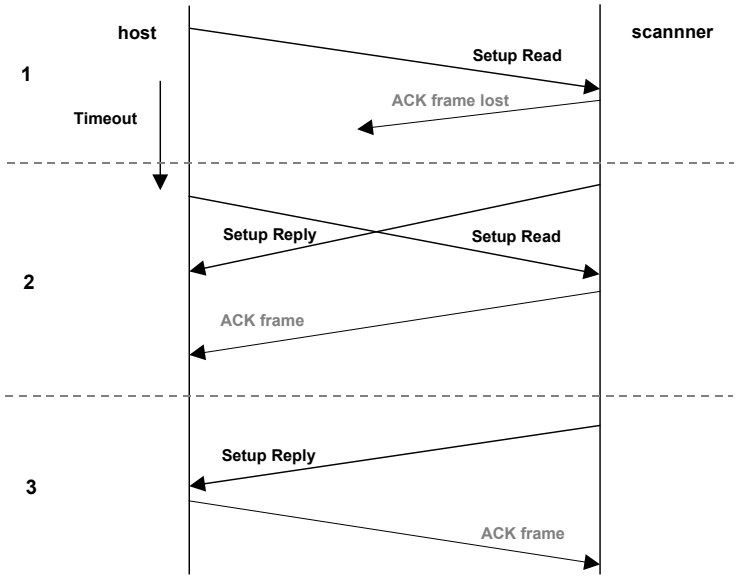
   The scanner is now in Silent Mode.

4. Host sends again the Setup Read and scanner replies with an ACK.

5. Scanner sends Setup Reply and host responds with an ACK.

Note: To exit Silent Mode, you must send a Silent Mode disable control command or turn the scanner off.

Host loses acknowledge



1. Host sends a Setup Read. Scanner replies with a low level ACK frame but the ACK is lost.

2. Host waits for the ACK until the end of the timeout.

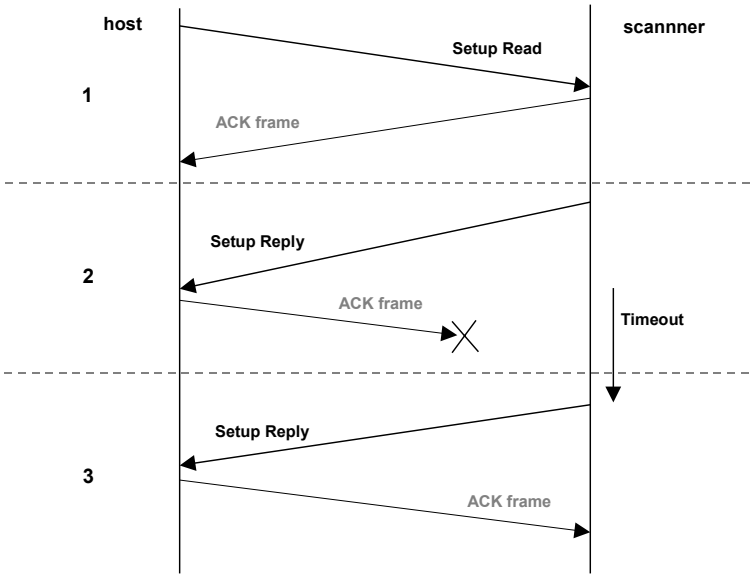
   Since the host hasn't received a response by the end of the timeout, it sends the Setup Read again.

   At the same time, the scanner sends the Setup Reply, not knowing that the host has not received the ACK frame. The host ignores this frame.

   When the scanner receives the second Setup Read, it detects that this frame has already been received and it sends the ACK frame again.

3. Scanner sends Setup Reply again. Host responds with a low level ACK frame.

Scanner loses Acknowledge



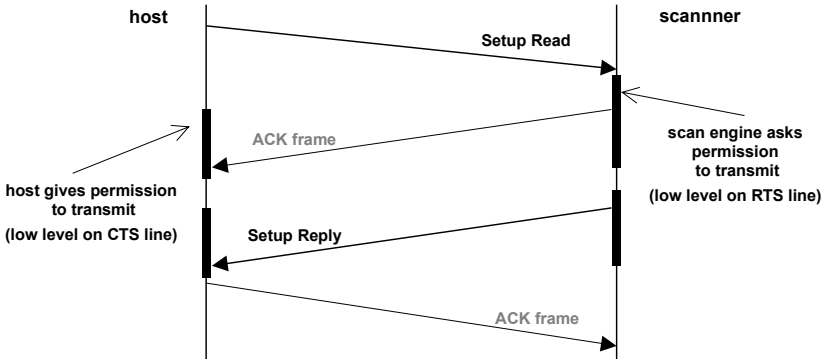
1. Host sends Setup Read. Scanner replies with a low level ACK frame.
2. Scanner sends Setup Reply. Host replies with an ACK frame but the frame is lost.
3. Scanner waits for the ACK until the end of the timeout.

   Since the scanner hasn't received a response by the end of the timeout, it sends the Setup Reply again.

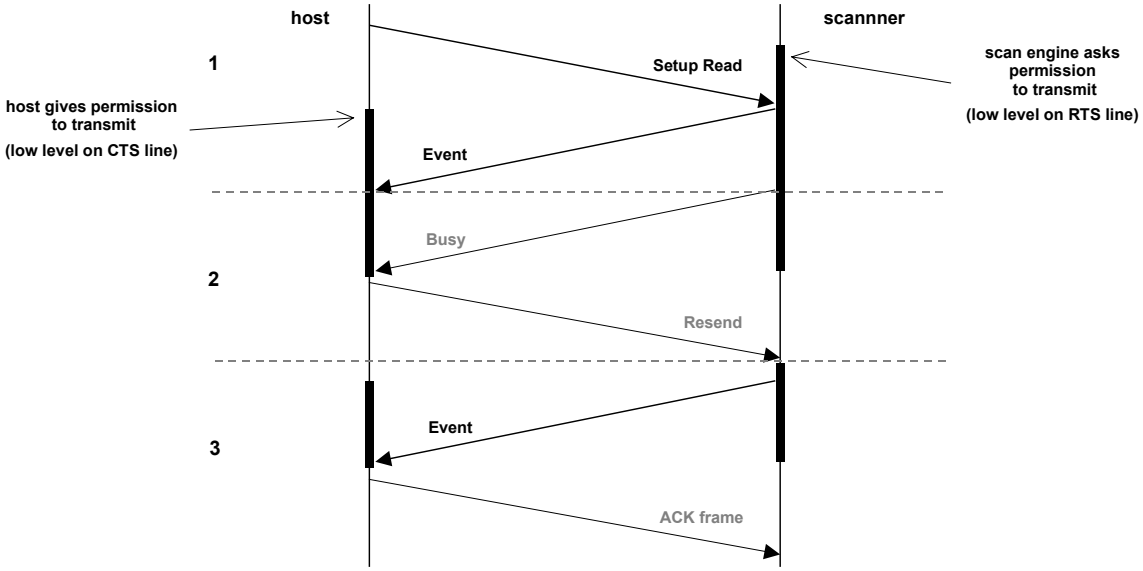
   When the host receives the same Setup Reply, it detects that this frame has already been received and it sends the ACK frame again.

6.2. Transmission with RTS/CTS handshaking

No contention



Contention – scanner has priority



1. Scanner asks permission to transmit while host is sending a Setup Read.

   (the host's frame has not yet been completely received). When the Setup Read transmission is completed, the host gives the permission to the scanner to transmit.

   The scanner sends the event.

   After sending the Event; the scanner cannot process the Setup Read as it is waiting for an ACK/NAK for the Event (the Setup Read is discarded).

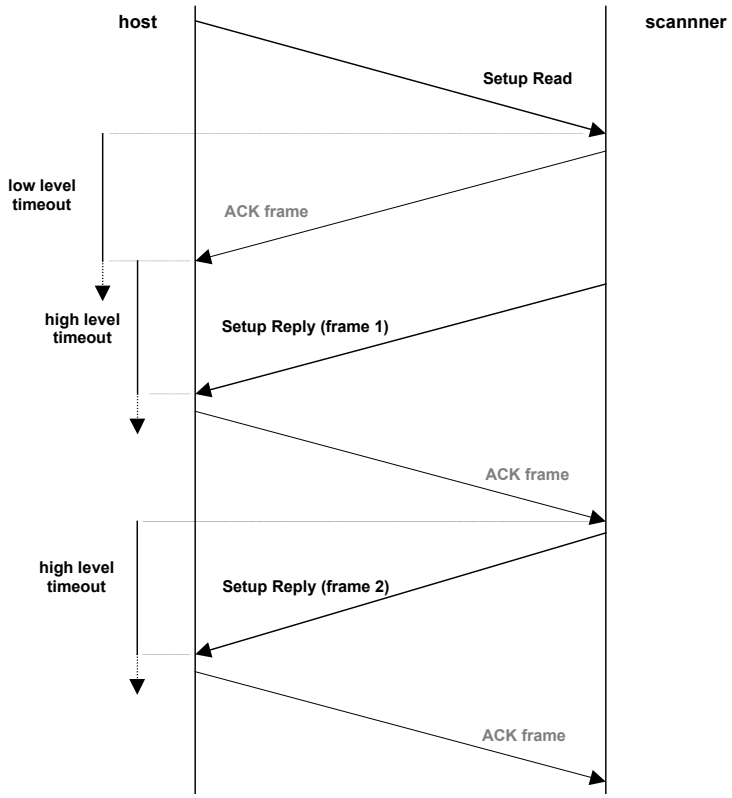
2. The scanner sends Busy to tell the host that it cannot not process the Setup Read and has not received acknowledgment for the Event.

   The host sends a Resend to ask the scanner to send the same frame again.

3. The scanner resends the Event and the host responds with ACK.

7. Timeouts

7.1. Host side



Low level timeout

The low level timeout must always be set to the highest of these two values:

$$\text{low level timeout(s)} = 0.5$$

or

$$\text{low level timeout(s)} = \frac{(\text{GRB\_duration} * \text{GRB\_number}) + (0.06 * (\text{GRB\_number} - 1))}{2}$$

GRB Duration = Good Read Beep Duration in second (setup parameter)
GRB Number = Good Read Beep Number (setup parameter)

High level timeouts

In standard situations:

$$\text{high level timeout(s)} = \frac{\text{TFS} * 10}{\text{baud rate}} + 0.05$$

When a scanner receives a Setup Write frame, the high level timeout must be adjusted to allow the the setup parameters to be written in the non volatile memory. Calculate as follows:

$$\text{high level timeout(s)} = \frac{\text{TFS} * 10}{\text{baud rate}} + 1$$

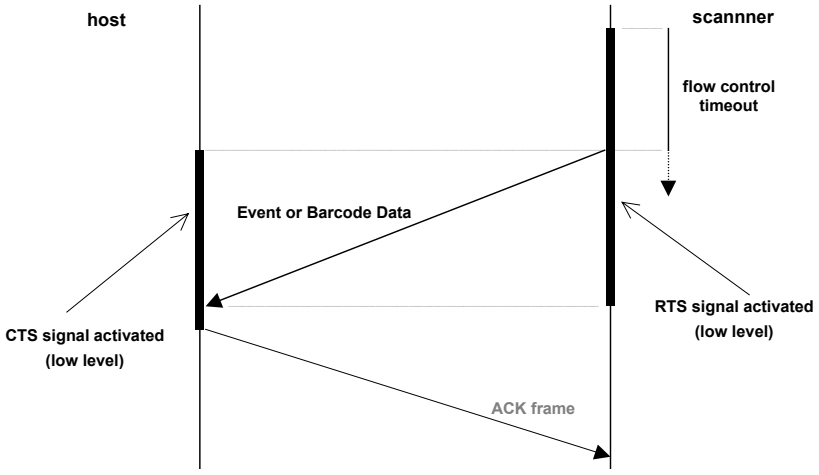
For a buzzer or beep sequence control command, the high level timeout depends on the execution time of the control command. Calculate as follows:

$$\text{high level timeout(s)} = \frac{\text{TFS} * 10}{\text{baud rate}} + \text{command execution duration}$$

Timeout for special frames

$$\text{special frame timeout(s)} = 0.5$$

7.2. Scanner side

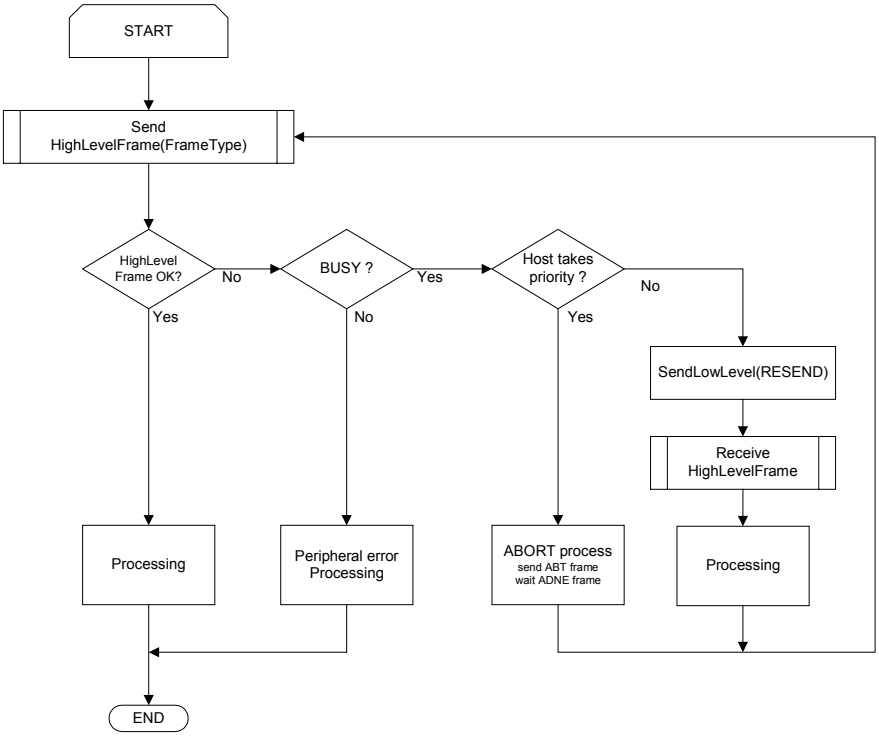


When the RTS/CTS protocol is activated, the scanner uses the Flow Control Timeout (section 9.1.21, Serial Interface) when it transmits a frame.

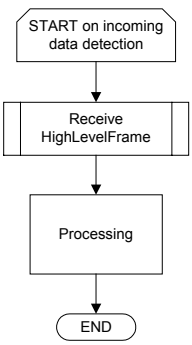
The user must adapt the Flow Control Timeout value to the host frame reception capabilities.

8. Host transmission and reception flow charts

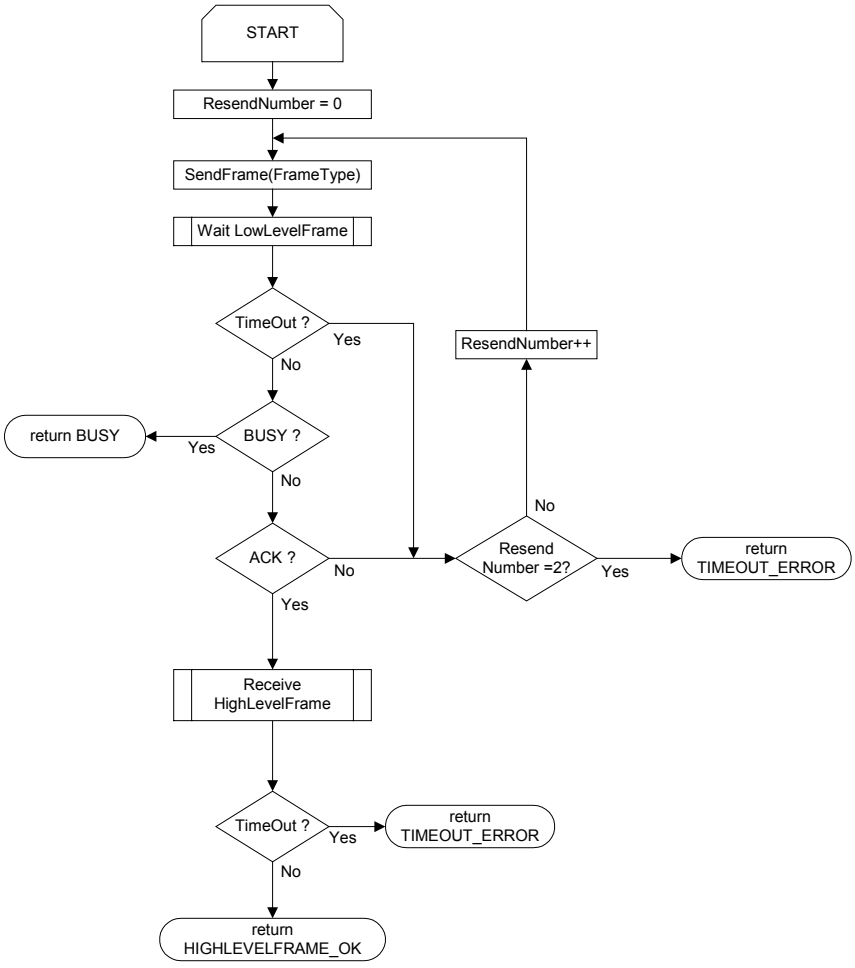
8.1. Host transmission



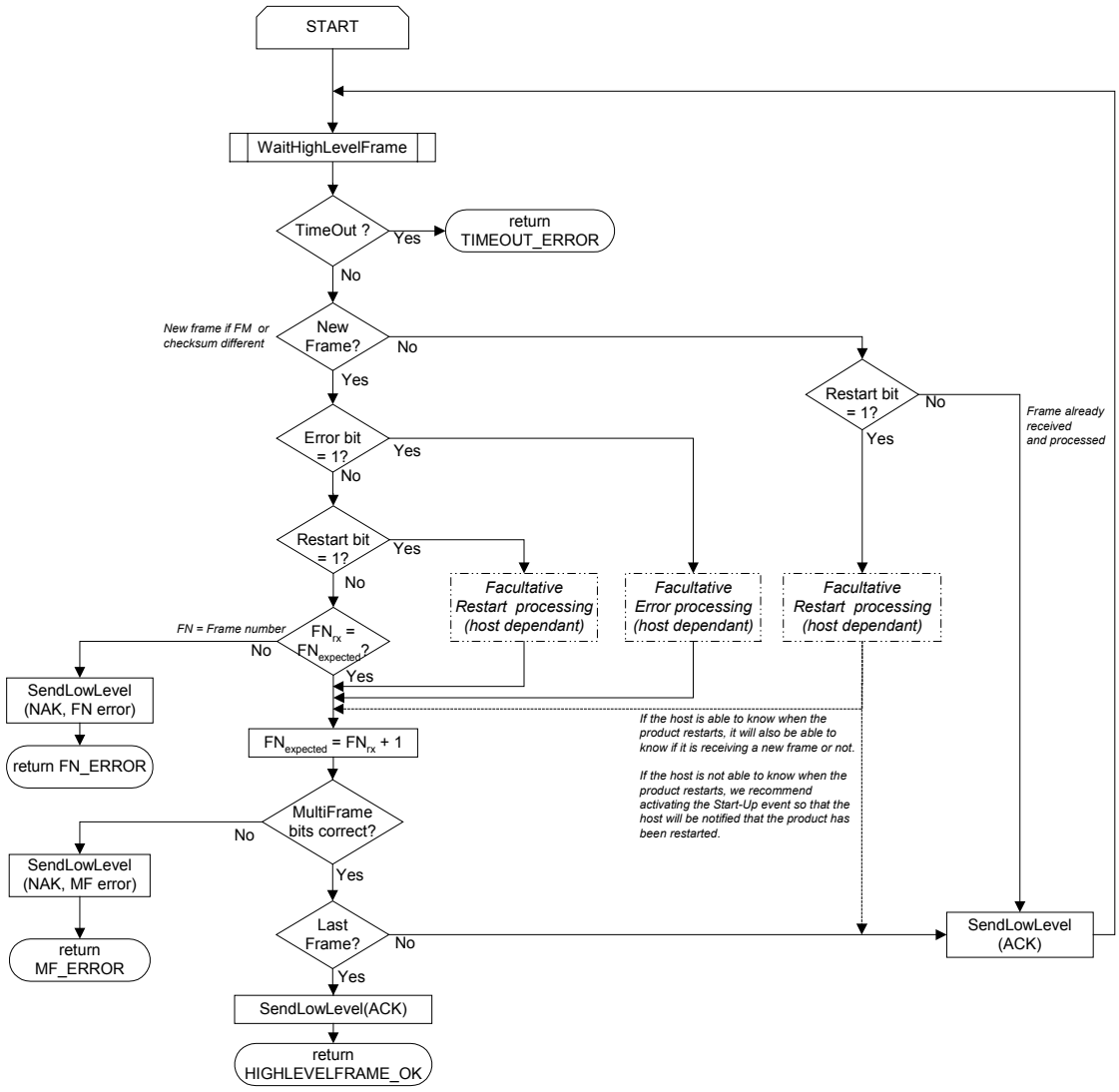
8.2. Host reception



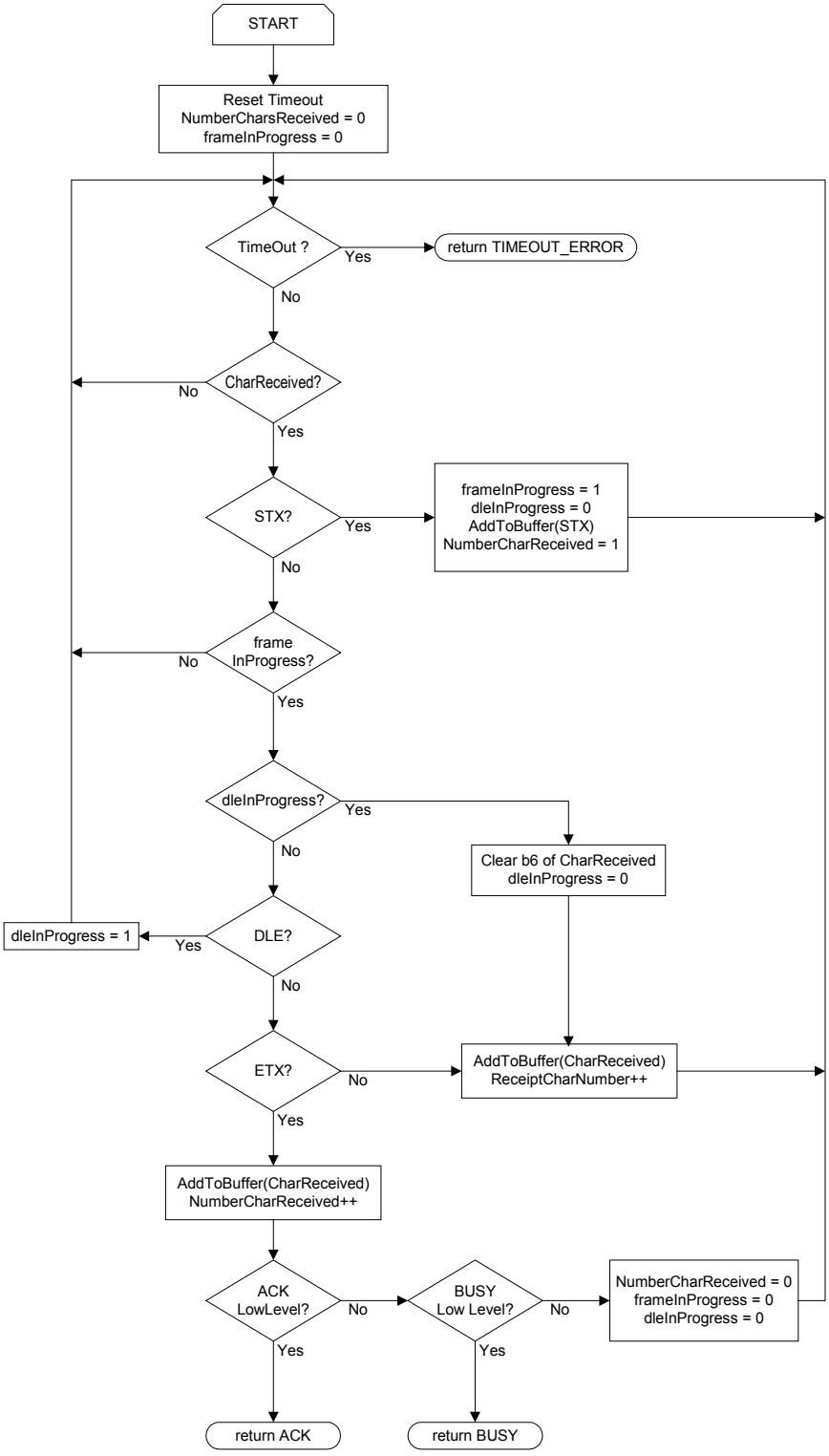
8.3. Send HighLevelFrame



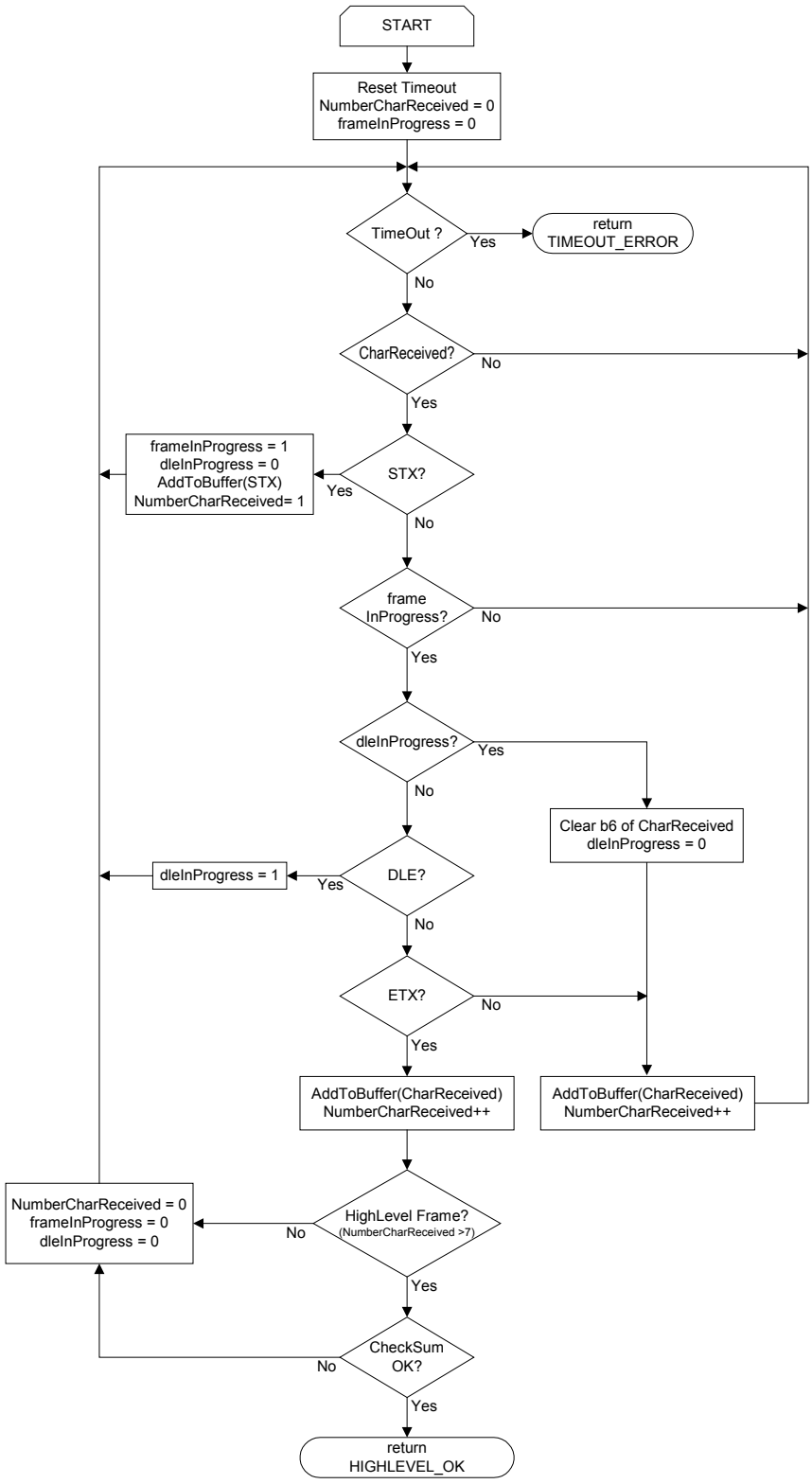
8.4. Receive HighLevelFrames



8.5. Waiting for low level frames



8.6. Waiting for high level frames



9. Commands

As different high level frame types use different sets of commands, commands are organised into groups :

Setup Groups <SG>

Control Groups <CG>

Status Groups <STG>

Event Groups <EG>

9.1. Setup Groups <SG>

The Setup Group commands are the commands used when setting up the scan engine or getting setup information from the scan engine. Setup Groups are used with the following frame types:

sent from host

Setup Read	<SR>	0x40
Setup Write	<SW>	0x41
Setup Permission Read	<SPR>	0x44
Setup Permission Write	<SPW>	0x45

sent from the scan engine

Setup Reply	<SRP>	0x50
Result	<RSLT>	0x51
Setup Permission Reply	<SPRP>	0x54

Example – Setup Write

Codabar - activation - enable

Using the hexadecimal values from the table, this frame is the setup command that enables Codabar activation:

<STX>	<SN>	**<SW>**	**<SG>**	**<FID>**	**parm**	<FM>	<CHK>	<ETX>
02	19	**41**	**40**	**40**	**01**	67	04 04	03

Note: All numerical string values are hexadecimal. For default settings, please refer to the Integration Guide for your product.

9.1.1. Codabar <SG> = 0x40

setup function	FID	parameter	value	string
activation	40	disable	00	40 40 00
		enable	01	40 40 01

start stop transmission	58	not transmitted	00	40 58 00
		a, b, c, d	01	40 58 01
		A, B, C, D	02	40 58 02
		a, b, c, d / t, n, *, e	03	40 58 03
		DC1, DC2, DC3, DC4	04	40 58 04
CLSI library system	59	disable	00	40 59 00
		enable	01	40 59 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	40 48 nn
		example: D	44	40 48 44
check digit verification	4C	disable	00	40 4C 00
		enable	01	40 4C 01
check digit transmission	54	disable	00	40 54 00
		enable	01	40 54 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	40 50 nn
		example: 6	06	40 50 06
length L2 *second length parameter*	51	numerical value (1 byte)	[00..FF]	40 51 nn
		example: 0	00	40 51 00
length L3 *third length parameter*	52	numerical value (1 byte)	[00..FF]	40 52 nn
		example: 0	00	40 52 00
length mode *selects length requirements* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	40 53 00
		L1, L2, L3 as fixed length	01	40 53 01

9.1.2. Codablock <SG> = 0x4D

setup function	FID	parameter	value	string
codablock F	40	disable	00	4D 40 00
		enable	01	4D 40 01
codablock A	41	disable	00	4D 41 00
		enable	01	4D 41 01
codablock F code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	4D 48 nn
		example: D	44	4D 48 44
codablock A code mark *custom identifier for symbology*	49	ASCII value (1 byte)	[00..FF]	4D 49 nn
		example: D	44	4D 49 44

9.1.3. Code 11 <SG> = 0x4A

setup function	FID	parameter	value	string
activation	40	disable	00	4A 40 00
		enable	01	4A 40 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	4A 48 nn
		example: " * "	2A	4A 48 2A

check digit verification	4C	1 digit	01	4A 4C 01
		2 digits	02	4A 4C 02
check digit transmission	54	disable	00	4A 54 00
		enable	01	4A 54 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	4A 50 nn
		example: 0	00	4A 50 00
length L2 *fixed parameter*	51	0	00	4A 51 00
length L3 *fixed parameter*	52	0	00	4A 52 00
length mode *fixed parameter* *If L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	4A 53 00

9.1.4. Code 39 <SG> = 0x42

setup function	FID	parameter	value	string
activation	40	disable	00	42 40 00
		enable	01	42 40 01
full ASCII conversion	5A	disable	00	42 5A 00
		enable	01	42 5A 01
reading range	47	normal	00	42 47 00
		extended	01	42 47 01
start/stop transmission	58	disabled	00	42 58 00
		enabled	01	42 58 01
accepted start character	59	" $ "	01	42 59 01
		" * "	02	42 59 02
		" $ " and " * "	03	42 59 03
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	42 48 nn
		example: " * "	2A	42 48 2A
check digit verification	4C	disable	00	42 4C 00
		modulo 43	01	42 4C 01
		French CIP	02	42 4C 02
		Italian CPI	03	42 4C 03
check digit transmission	54	disable	00	42 54 00
		enable	01	42 54 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	42 50 nn
		example: 0	00	42 50 00
length L2 *fixed parameter*	51	0	00	42 51 00
length L3 *fixed parameter*	52	0	00	42 52 00

length mode *fixed parameter* *If L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	42 53 00

9.1.5. Code 93 <SG> = 0x41

setup function	FID	parameter	value	string
activation	40	disable	00	41 40 00
		enable	01	41 40 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	41 48 nn
		example: D	44	41 48 44
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	41 50 nn
		example: 0	00	41 50 00
length L2 *fixed parameter*	51	0	00	41 51 00
length L3 *fixed parameter*	52	0	00	41 52 00
length mode *fixed parameter* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	41 53 00

9.1.6. Code 128 <SG> = 0x43

setup function	FID	parameter	value	string
activation	40	disable	00	43 40 00
		enable	01	43 40 01
EAN 128 identifier	58	disable	00	43 58 00
		enable	01	43 58 01
FNC1 conversion	59	ASCII value (1 byte)	[00..FF]	43 59 nn
		example: GS	1D	43 59 1D
ISBT	41	disabled	00	43 41 00
		enabled	01	43 41 01
ISBT concatenation transmission	5A	disabled	00	43 5A 00
		transmit only concatenated codes	01	43 5A 01
		transmit concatenated or single codes	02	43 5A 02
concatenate any pair of ISBT codes	5B	disabled	00	43 5B 00
		enabled	01	43 5B 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	43 48 nn
		example: D	44	43 48 44
check digit verification	4C	disable	00	43 4C 00
		French CIP	01	43 4C 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	43 50 nn
		example: 0	00	43 50 00

length L2 *fixed parameter*	51	0	00	43 51 00
length L3 *fixed parameter*	52	0	00	43 52 00
length mode *fixed parameter* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	43 53 00

9.1.7. Interleaved 2 of 5 <SG> = 0x44

setup function	FID	parameter	value	string
activation	40	disable	00	44 40 00
		enable	01	44 40 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	44 48 nn
		example: I	49	44 48 49
check digit verification	4C	disable	00	44 4C 00
		modulo 10	01	44 4C 01
		French CIP HR	02	44 4C 02
check digit transmission	54	disable	00	44 54 00
		enable	01	44 54 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	44 50 nn
		example: 6	06	44 50 06
length L2 *second length parameter*	51	numerical value (1 byte)	[00..FF]	44 51 nn
		example: 0	00	44 51 00
length L3 *third length parameter*	52	numerical value (1 byte)	[00..FF]	44 52 nn
		example: 0	00	44 52 00
length mode *selects length requirements* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	44 53 00
		L1, L2, L3 as fixed length	01	44 53 01

9.1.8. Matrix 2 of 5 <SG> = 0x45

setup function	FID	parameter	value	string
activation	40	disable	00	45 40 00
		enable	01	45 40 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	45 48 nn
		example: D	44	45 48 44
check digit verification	4C	disable	00	45 4C 00
		modulo 10	01	45 4C 01
check digit transmission	54	disable	00	45 54 00
		enable	01	45 54 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	45 50 nn
		example: 6	06	45 50 06

length L2 *fixed parameter*	51	0	00	45 51 00
length L3 *fixed parameter*	52	0	00	45 52 00
length mode *fixed parameter* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	45 53 00

9.1.9. MSI Code <SG> = 0x46

setup function	FID	parameter	value	string
activation	40	disable	00	46 40 00
		enable	01	46 40 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	46 48 nn
		example: D	44	46 48 44
check digit verification	4C	modulo 10	01	46 4C 01
		double modulo 10	02	46 4C 02
check digit transmission	54	disable	00	46 54 00
		enable	01	46 54 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	46 50 nn
		example: 6	06	46 50 06
length L2 *fixed parameter*	51	0	00	46 51 00
length L3 *fixed parameter*	52	0	00	46 52 00
length mode *fixed parameter* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	46 53 00

9.1.10. PDF 417 <SG> = 0x4C

setup function	FID	parameter	value	string
activation	40	disable	00	4C 40 00
		enable	01	4C 40 01
macro PDF	41	unbuffered	00	4C 41 00
		buffered	01	4C 41 01
control header	58	not transmitted	00	4C 58 00
		transmitted	01	4C 58 01
file name	59	not transmitted	00	4C 59 00
		transmitted	01	4C 59 01
segment	5A	not transmitted	00	4C 5A 00
		transmitted	01	4C 5A 01

time stamp	5B	not transmitted	00	4C 5B 00
		transmitted	01	4C 5B 01
sender	5C	not transmitted	00	4C 5C 00
		transmitted	01	4C 5C 01
addressee	5D	not transmitted	00	4C 5D 00
		transmitted	01	4C 5D 01
file size	5E	not transmitted	00	4C 5E 00
		transmitted	01	4C 5E 01
checksum	5F	not transmitted	00	4C 5F 00
		transmitted	01	4C 5F 01
micro PDF 417 activation	42	disable	00	4C 42 00
		enable	01	4C 42 01
Micro PDF code 128 emulation	45	Disable	00	4C 45 00
		Enable	01	4C 45 01
Codemark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	4C 48 nn
		example: " * "	2A	4C 48 2A

9.1.11. Plessey Code <SG> = 0x47

setup function	FID	parameter	value	string
activation	40	disable	00	47 40 00
		enable	01	47 40 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	47 48 nn
		example: D	44	47 48 44
check digit transmission	54	disable	00	47 54 00
		enable	01	47 54 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	47 50 nn
		example: 0	00	47 50 00
length L2 *fixed parameter*	51	0	00	47 51 00
length L3 *fixed parameter*	52	0	00	47 52 00
length mode *fixed parameter* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	47 53 00

9.1.12. Standard 2 of 5 <SG> = 0x48

setup function	FID	parameter	value	string
activation	40	disable	00	48 40 00
		enable	01	48 40 01
standard 2 of 5 format	58	identicon *6 start/stop bars*	00	48 58 00
		computer identics *4 start/stop bars*	01	48 58 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	48 48 nn
		example: D	44	48 48 44
check digit verification	4C	disable	00	48 4C 00
		modulo 10	01	48 4C 01
check digit transmission	54	disable	00	48 54 00
		enable	01	48 54 01
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	48 50 nn
		example: 6	06	48 50 06
length L2 *second length parameter*	51	numerical value (1 byte)	[00..FF]	48 51 nn
		example: 0	00	48 51 00
length L3 *third length parameter*	52	numerical value (1 byte)	[00..FF]	48 52 nn
		example: 0	00	48 52 00
length mode *selects length requirements* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	48 53 00
		L1, L2, L3 as fixed length	01	48 53 01

9.1.13. Telepen <SG> = 0x49

setup function	FID	parameter	value	string
activation	40	disable	00	49 40 00
		enable	01	49 40 01
format	58	ASCII	00	49 58 00
		numeric	01	49 58 01
code mark *custom identifier for symbology*	48	ASCII value (1 byte)	[00..FF]	49 48 nn
		example: " * "	2A	49 48 2A
length L1 *first length parameter*	50	numerical value (1 byte)	[00..FF]	49 50 nn
		example: 0	00	49 50 00
length L2 *fixed parameter*	51	0	00	49 51 00
length L3 *fixed parameter*	52	0	00	49 52 00
length mode *fixed parameter* *if L1, L2, L3 and length mode = 0x00, no length is specified and all bar codes are read*	53	L1 as minimum length	00	49 53 00

9.1.14. UPC / EAN <SG> = 0x4B

setup function	FID	parameter	value	string
UPC-A activation	40	disable	00	4B 40 00
		enable	01	4B 40 01
UPC-E activation	41	disable	00	4B 41 00
		enable	01	4B 41 01
EAN-8 activation	42	disable	00	4B 42 00
		enable	01	4B 42 01
EAN-13 activation	43	disable	00	4B 43 00
		enable	01	4B 43 01
ISBN conversion EAN-13 activation	44	disable	00	4B 44 00
		enable	01	4B 44 01
add-on digits	5D	not required but transmitted if read	00	4B 5D 00
		required and transmitted	01	4B 5D 01
add-on 2	45	disable	00	4B 45 00
		enable	01	4B 45 01
add-on 5	46	disable	00	4B 46 00
		enable	01	4B 46 01
check digit UPC-A transmitted	54	disable	00	4B 54 00
		enable	01	4B 54 01
check digit UPC-E transmitted	55	disable	00	4B 55 00
		enable	01	4B 55 01
check digit EAN-8 transmitted	56	disable	00	4B 56 00
		enable	01	4B 56 01
check digit EAN-13 transmitted	57	disable	00	4B 57 00
		enable	01	4B 57 01
UPC-A number system transmitted	58	disable	00	4B 58 00
		enable	01	4B 58 01
UPC-E number system transmitted	59	disable	00	4B 59 00
		enable	01	4B 59 01
UPC-A transmitted as EAN-13	5A	disable	00	4B 5A 00
		enable	01	4B 5A 01
UPC-E transmitted as UPC-A	5B	disable	00	4B 5B 00
		enable	01	4B 5B 01
EAN-8 transmitted as EAN-13	5C	disable	00	4B 5C 00
		enable	01	4B 5C 01
UPC-A code mark	48	ASCII value (1 byte)	[00..FF]	4B 48 nn
custom identifier for symbology		example: A	41	4B 48 41
UPC-E code mark	49	ASCII value (1 byte)	[00..FF]	4B 49 nn
custom identifier for symbology		example: E	45	4B 49 45
EAN-8 code mark	4A	ASCII value (1 byte)	[00..FF]	4B 4A nn
custom identifier for symbology		example: 0xFF	FF	4B 4A FF

EAN-13 code mark *custom identifier for symbology*	4B	ASCII value (1 byte)	[00..FF]	4B 4B nn
		example: F	46	4B 4B 46

9.1.15. Message format <SG> = 0x60

setup function	FID	parameter	value	string
preamble	C0	ASCII string of 1 to 20 characters *size of string (2 bytes) must predede ASCII string.*	[00..FF]	60 C0 nn nn nn .. ..
		Example: no preamble	00 00	60 C0 00 00
postamble	C1	ASCII string (max. 20 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	60 C1 nn nn nn .. ..
		example: <CR> <LF>	0D 0A	60 C1 00 02 0D 0A
symbology identifier *activates identifier transmission for all symbologies inserted between prefix and decoded data* *Intermec identifiers can be customized by modifying the Code Mark of each symbology*	40	disable	00	60 40 00
		Intermec / custom identifier	01	60 40 01
		AIM identifier	02	60 40 02
inter-character delay (output to host) *avoids dropping characters if transmitting decoded data too fast for the host* *(not for use with ISCP)*	80	value in milliseconds (1 word from 0 to 2550 ms)	[0000 … 09F6]	60 80 nn nn
		example: 0 ms	0000	60 80 00 00
inter-message delay (output to host) *allows host enough time to process each message received* *(not for use with ISCP)*	81	value in milliseconds (1 word from 0 to 2550 ms)	[0000 … 09F6]	60 81 nn nn
		example: 0 ms	0000	60 81 00 00
output message on unsuccessful read *sends a message (Output Message Selection) to host if unsuccessful read* *(not applicable when sending data in packet format, see section 9.1.23, Data Format)*	41	disable	00	60 41 00
		enable	01	60 41 01
output message selection *selects message sent following an unsuccessful read* *(not applicable when sending data in packet format, see section 9.1.23, Data Format)*	C2	ASCII string (max. 10 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	60 C2 nn nn nn .. ..
		example: NO READ	00	60 C2 00 07 4E 4F 20 52 45 41 44

9.1.16. Data editing <SG> = 0x65

The scanner is able to modify data before sending it to the host. This option is called Data Editing.

Scenarios

There are seven possible Data Editing scenarios. Each scenario is made up of a mask, a barcode symbology identifier, a barcode length and a list of actions.

Mask

You can define one mask for each scenario. The mask is used as a filter to determine which incoming data you want to modify.

The value "null" means that the editing actions will be applied to all data (no mask used).

wildcards	definition	example
*	zero or more characters	*123* = any code with a consecutive 1 2 3
?	one character	???A* = any code with A in the fourth position followed by zero or more characters
\ + wildcard	a backslash plus a special character takes away the wildcard function	*1*3* = any code with a consecutive 1 * 3 anywhere in the data
\C	non-numerical value	\C* = any code that starts with a non-numerical value
\N	numerical value	*\N* = any code that contains a numerical value

The wildcards in the table above can be used to define the mask. Each mask is sent as an ASCII string of up to 26 characters.

Example of mask for scenario 1:

<STX>	<SN>	<SW>	<SG>	<FID>	<size>	mask = *123*	<FM>	<CHK>	<ETX>
02	24	**41**	**65**	**C0**	**00 05**	**2A 31 32 33 2A**	60	1C 4F	03

Barcode symbology identifier

value	symbology	value	symbology	value	symbology
0	all symbologies	13	Code 39	26	Code 11
1	EAN-13	14	reserved	27	Telepen
2	EAN-8	15	Interleaved 2 of 5	28	Code 49
3	UPC-A	16	Standard 2 of 5	29	Code 39 Italian CPI
4	UPC-E	17	Matrix 2 of 5	30	Codablock A
5	EAN-13 with add-on 2	18	reserved	31	Codablock F
6	EAN-8 with add-on 2	19	Codabar	32	reserved
7	UPCA with add-on 2	20	Ames Code	33	PDF 417
8	UPCE with add-on 2	21	MSI	34	EAN 128
9	EAN-13 with add-on 5	22	Plessey	35	ISBT 128
10	EAN-8 with add-on 5	23	Code 128	36	Micro PDF
11	UPCA with add-on 5	24	Code 16K		
12	UPCE with add-on 5	25	Code 93		

Using the values above, you can define one type of symbology for each scenario. The editing actions will be applied only to the type of symbology that you define (example: Code 39).

Example of a Code 39 barcode symbology identifier for scenario 1:

<STX>	<SN>	**<SW>**	**<SG>**	**<FID>**	**Code 39**	<FM>	<CHK>	<ETX>
02	24	**41**	**65**	**50**	**13**	60	05 27	03

Barcode length

You can also define the barcode length for each scenario. This allows you to apply the editing actions only to bar codes of a certain length. The length is a numerical value from 0 to 65535 (1 word).

The value " 0 " means that the editing actions will be applied to symbologies of all lengths.

Example of a barcode length of 25 characters for scenario 1:

<STX>	<SN>	<SW>	<SG>	<FID>	length	<FM>	<CHK>	<ETX>
02	24	41	65	80	00 25	60	0C 4C	03

Note: The mask, barcode symbology filter and barcode length can be used together or separately for each scenario. For example, if you want scenario 1 to be applied to Code 39 bar codes, you only need to define the barcode symbology filter for scenario 1. If you want scenario 1 to be applied to all Code 39 bar codes that start with "123", you must define both the mask and barcode symbology filter for scenario 1.

Action list

The action list for the scenario defines the modifications to be made to all data matching the scenario filter criteria before sending it to the host. You can choose one or several actions. The actions will be executed in the order that you send them in the command.

action	value	first	second	third
move block *allows you to move a block of characters*	0x01	source address (2 bytes)	length (2 bytes)	destination address (2 bytes)
delete block *allows you to delete a block of characters*	0x02	source address (2 bytes)	length (2 bytes)	–
insert *allows you to insert characters*	0x03	destination address (2 bytes)	length (1 byte)	ASCII string (variable)
replace *allows you to replace all occurrences of one character by another character*	0x04	original character (1 byte)	final character (1 byte)	–
delete character *allows you to delete all occurrences of one character*	0x05	character (1 byte)	–	–
copy block *allows you to copy a block of characters to another destination in the data string*	0x06	source address (2 bytes)	length (2 bytes)	destination address (2 bytes)
do not transmit *all data matching the scenario filter criteria will NOT be transmitted*	0x07	–	–	–

Source, Destination and Length bytes

When specifying the source, destination or length, a negative number indicates that the position is counted from the *end* of the data instead of counting from the beginning. To send a negative number, bit 7 of the first byte must be set to 1. The following is an example of the 2 bytes that make up the source, destination or length.

Example – 1:

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0	0	0

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	1

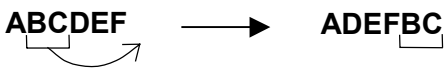
Bit 7 of the first byte is set to 1 to indicate that the value is negative.

Example of an action list for scenario 1:

move a block of 2 characters (length), counting from position 2 (source), to the end of the data (destination –1)

<STX>	<SN>	**<SW>**	**<SG>**	**<FID>**	**<size>**	**move**	**source**	**length**	**dest.**	<FM>	<CHK>	<ETX>
02	24	**41**	**65**	**D0**	**00 07**	**01**	**00 02**	**00 02**	**80 01**	60	15 2E	03

This is an example of the above action list applied to a barcode containing the data ABCDEF:



setup function	FID	parameter	value	string
activation – scenario 1	40	disable	00	65 40 00
		enable	01	65 40 01
activation – scenario 2	41	disable	00	65 41 00
		enable	01	65 41 01
activation – scenario 3	42	disable	00	65 42 00
		enable	01	65 42 01
activation – scenario 4	43	disable	00	65 43 00
		enable	01	65 43 01
activation – scenario 5	44	disable	00	65 44 00
		enable	01	65 44 01
activation – scenario 6	45	disable	00	65 45 00
		enable	01	65 45 01
activation – scenario 7	46	disable	00	65 46 00
		enable	01	65 46 01
mask – scenario 1	C0	ASCII string (max. 26 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 C0 nn nn nn..
mask – scenario 2	C1	ASCII string (max. 26 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 C1 nn nn nn..
mask – scenario 3	C2	ASCII string (max. 26 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 C2 nn nn nn..
mask – scenario 4	C3	ASCII string (max. 26 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 C3 nn nn nn..
mask – scenario 5	C4	ASCII string (max. 26 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 C4 nn nn nn..
mask – scenario 6	C5	ASCII string (max. 26 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 C5 nn nn nn..

mask – scenario 7	C6	ASCII string (max. 26 characters) *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 C6 nn nn nn..
barcode symbology identifier – scenario 1	50	barcode symbology code (1 byte) *see barcode symbology identifier table*	[00..MAX ID]	65 50 nn
barcode symbology identifier – scenario 2	51	barcode symbology code (1 byte) *see barcode symbology identifier table*	[00..MAX ID]	65 51 nn
barcode symbology identifier – scenario 3	52	barcode symbology code (1 byte) *see barcode symbology identifier table*	[00..MAX ID]	65 52 nn
barcode symbology identifier – scenario 4	53	barcode symbology code (1 byte) *see barcode symbology identifier table*	[00..MAX ID]	65 53 nn
barcode symbology identifier – scenario 5	54	barcode symbology code (1 byte) *see barcode symbology identifier table*	[00..MAX ID]	65 54 nn
barcode symbology identifier – scenario 6	55	barcode symbology code (1 byte) *see barcode symbology identifier table*	[00..MAX ID]	65 55 nn
barcode symbology identifier – scenario 7	56	barcode symbology code (1 byte) *see barcode symbology identifier table*	[00..MAX ID]	65 56 nn
barcode length – scenario 1	80	numerical value (1 word)	[0..FFFF]	65 80 nn nn
barcode length – scenario 2	81	numerical value (1 word)	[0..FFFF]	65 81 nn nn
barcode length – scenario 3	82	numerical value (1 word)	[0..FFFF]	65 82 nn nn
barcode length – scenario 4	83	numerical value (1 word)	[0..FFFF]	65 83 nn nn
barcode length – scenario 5	84	numerical value (1 word)	[0..FFFF]	65 84 nn nn
barcode length – scenario 6	85	numerical value (1 word)	[0..FFFF]	65 85 nn nn
barcode length – scenario 7	86	numerical value (1 word)	[0..FFFF]	65 86 nn nn
action list – scenario 1	D0	ASCII string of 1 to 100 characters *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 D0 nn nn nn..
action list – scenario 2	D1	ASCII string of 1 to 100 characters *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 D1 nn nn nn..
action list – scenario 3	D2	ASCII string of 1 to 100 characters *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 D2 nn nn nn..
action list – scenario 4	D3	ASCII string of 1 to 100 characters *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 D3 nn nn nn..
action list – scenario 5	D4	ASCII string of 1 to 100 characters *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 D4 nn nn nn..

action list – scenario 6	D5	ASCII string of 1 to 100 characters *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 D5 nn nn nn..
action list – scenario 7	D6	ASCII string of 1 to 100 characters *size of string (2 bytes) must precede ASCII string*	[00..FF]	65 D6 nn nn nn..

9.1.17. Decoding security <SG> = 0x71

setup function	FID	parameter	value	string
consecutive same read data validation *selects the number of consecutive same read before transmission* *the value 0 sets the scanner to auto-adapt consecutive same read according to the bar code read*	40	value from 0 to 10 (1 byte)	[00… 0A]	71 40 nn
		example: 1	01	71 40 01
timeout between identical consecutive reads	80	value in milliseconds from 0 to 2550 (1 word)	[0000 … 09F6]	71 80 nn nn
		example: 300 ms	012C	71 80 01 2C
timeout between different consecutive reads	81	value in milliseconds from 0 to 2550 (1 word)	[0000 … 09F6]	71 81 nn nn
		example: 0 ms	0000	71 81 00 00

9.1.18. Beep / Led indicator <SG> = 0x72

setup function	FID	parameter	value	string
tone frequency *selects buzzer tone frequency* *affects all standard beeps (error, good read, setup and parameter beeps)*	80	value in Hertz from 1000 to 4095 (1 word)	[03E8 … 0FFF]	72 80 nn nn
		example: 2093 Hz	082D	72 80 08 2D
power up beep / power up led	40	disable	00	72 40 00
		enable	01	72 40 01
error beep	43	disable	00	72 43 00
		enable	01	72 43 01
setup beep / parameter beep	44	disable	00	72 44 00
		enable	01	72 44 01
good read beeps number *selects the number of beeps*	41	none	00	72 41 00
		one	01	72 41 01
		two	02	72 41 02
good read beep duration	81	value in milliseconds from 0 to 2550 (1 word)	[0000 … 09F6]	72 81 nn nn
		example: 80 ms	0050	72 81 00 50
good read led duration *when 0 ms, the good read led is always off* *when good read led is on, scanner can still read barcodes and receive commands*	82	value in milliseconds from 0 to 5110 (1 word)	[0000 … 13F6]	72 82 nn nn
		example: 80 ms	0050	72 82 00 50

good read event timing	42	before transmission	00	72 42 00
		after transmission	01	72 42 01
beep volume	45	low	00	72 45 00
		high	01	72 45 01
		medium	02	72 45 02
stacked code crackle	46	disable	00	72 46 00
		enable	01	72 46 01
stacked code flicker	47	disable	00	72 47 00
		enable	01	72 47 01

9.1.19. Trigger settings <SG> = 0x70

Trigger modes

There are different trigger modes possible according to how you want to use the scanner.

Continuous

When the scanner is turned on, a continuous reading session begins automatically (lighting and decode processing on). The reading session is stopped when the scanner receives a Decode Off command or a software input synchronization signal. A new continuous reading session begins when a Decode On command or software input sychronization signal is received.

Level

A reading session begins (lighting and decode processing on) when the Trigger line is activated, a Decode On command is received or a software input syhchronization signal is received. The reading session stops when the trigger line is deactivated, a Decode Off command is received, a software input sychronization signal is received or a bar code is decoded (only if Turn off After Good Read is enabled).

Pulse

A reading session begins when the Trigger line is activated, a Decode On command is received or a software input synchronization signal is received. The reading session stops when there is a period of inactivity (lasting the time specified by the timeout T1) or when a bar code is decoded (only if Turn Off After Good Read is enabled).

Flashing

When the scanner is turned on, a reading session begins (lighting and decode processing on). After a period of inactivity (lasting the time specified by the timeout T1), the light starts flashing. In flashing mode, the scanner is checking if there is a bar code to be read or not. When it detects a bar code, the lighting automatically turns on and the bar code is read and decoded. The lighting stays on until a new period of inactivity (timeout T1), then the lighting starts flashing again.

The reading session can start or stop when a Decode On or Off command is received or a software input synchronization signal is received.

Autostand

Autostand trigger mode allows you to switch between Flashing mode and Level mode. When the scanner is turned on, it is in Flashing mode (see above). You can automatically switch to Level mode (see above) by activating the Trigger line, sending a Decode On command or sending a software input synchronization signal. The scanner automatically switches back to Flashing mode after a periode of inactivity (lasting the time specified by the timeout T1).

setup function	FID	parameter	value	string
trigger mode	40	continuous	00	70 40 00
		level	01	70 40 01
		pulse	02	70 40 02
		flashing	03	70 40 03
		autostand	04	70 40 04

timeout T1 *select the timeout T1 duration*	80	value in seconds from 0 to 4095 (1 word)	[0000 … 0FFF]	70 80 nn nn
		example: 2 s	0002	70 80 00 02
hardware trigger *enable / disable the trigger line input*	41	disable	00	70 41 00
		enable	01	70 41 01
software input synchronization *allows the host to control reading sessions by sending Start and Stop characters through the serial interface when Remote Control Mode is disabled.* *(not for use with ISCP)*	42	disable	00	70 42 00
		enable	01	70 42 01
start character *selects the start character of the software input synchronization* *(not for use with ISCP)*	43	ASCII value (1 byte)	[00..FF]	70 43 nn
		example: STX	02	70 43 02
stop character *selects the stop character of the software input synchronization* *if the same value is selected for the start and stop characters, the stop character will be read as a start character.* *(not for use with ISCP)*	44	ASCII value (1 byte)	[00..FF]	70 44 nn
		example: ETX	03	70 44 03
aiming beam *a straight laser beam that allows you to locate the bar code you want to read*	45	disable	00	70 45 00
		1 pull, aim and read *pull and hold trigger – aiming beam (programmable duration) then reading beam*	01	70 45 01
		1 pull aim, 1 pull read *first pull aiming beam, second pull reading beam*	02	70 45 02
		1 pull read, 1 pull aim *first pull reading beam, second pull aiming beam*	03	70 45 03
aiming beam duration *programmable duration of the aiming beam (1 pull, aim and read)*	81	value in milliseconds from 0 to 2550 (1 word)	[0000 … 09F6]	70 81 nn nn
		example: 500 ms	0500	70 81 01 F4

Turn off after good read

This setup parameter is different depending on the trigger mode used.

Continuous:

Not used.

Level:

When enabled, the scanner stops the reading session after a successful decoding. When disabled, the reading session continues and stops only when the trigger line is deactivated, a Decode Off command is received or a software input synchronization signal is received.

Pulse:

When enabled, the scanner ends the reading session after a successful decoding. When disabled, the reading session continues and stops after a period of inactivity (lasting the time specified by the timeout T1).

Flashing:

Not used.

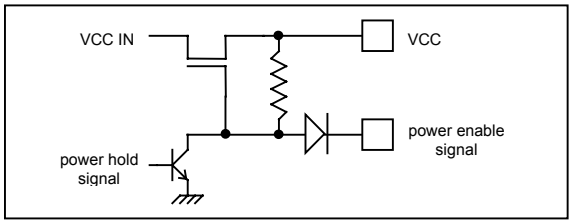
Autostand:

See Level mode and Flashing mode

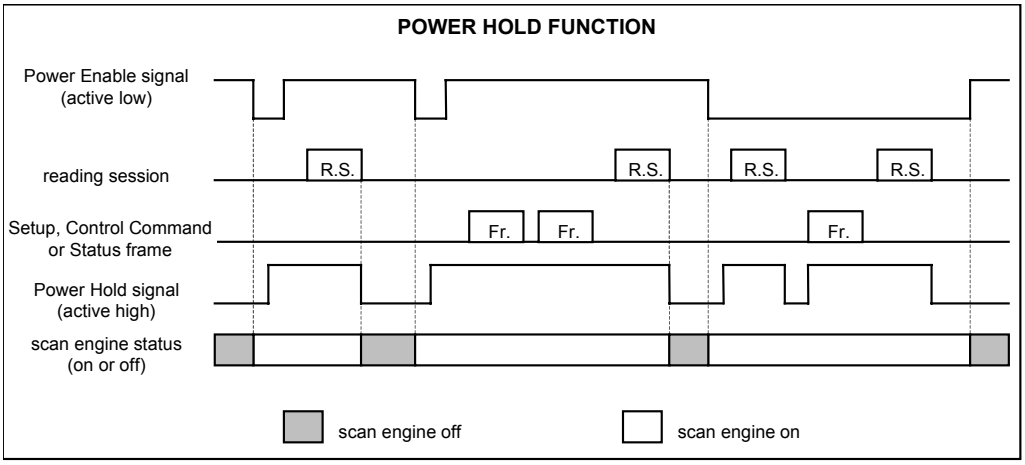
setup function	FID	parameter	value	string
turn off after good read	46	disable	00	70 46 00
		enable	01	70 46 01

Power hold function

This setup parameter selects the behavior of the Power Hold internal signal.



The internal power mosfet that switches the power on and off is driven by two signals: Power Enable (external hardware driven signal) and Power Hold (internal software driven signal). Using the Power Hold signal, the scanner is able to stay on even if the host changes the voltage level on the Power Enable line from low to high. See figure below.



When the power hold function is disabled, the scanner activates the power hold signal only when writing setup parameters in non volatile memory.

When the power hold function is enabled, the scanner activates the power hold signal:

- at start up
- at the beginning of a reading session
- after reception of a correct ISCP frame
- while writing setup parameters in non volatile memory

The scanner deactivates the signal:

- at the end of a reading session
- after receiving a Sleep command

setup function	FID	parameter	value	string
power hold	47	disable	00	70 47 00
		enable	01	70 47 01

9.1.20. Setup configuration <SG> = 0x74

Setup function	FID	parameter	value	string
configuration using bar codes *allows you to configure the scanner by reading configuration bar codes* *if disabled, you can only re-enable by sending a control command to the scanner directly from the host*	40	enable	00	74 40 00
		inhibit after 1 minute	01	74 40 01
		disable	02	74 40 02
permanent transparent mode *allows you to use your scanner to set up other products by reading configuration bar codes* *setup codes are transmitted to the other product and do not affect the configuration of the scanner*	41	disable	00	74 41 00
		enable	01	74 41 01

9.1.21. Serial interface <SG> = 0x63

Setup function	FID	parameter	value	string
baud rate *parameters are modified AFTER the Result Done frame is akcnowledged.*	40	1200	04	63 40 04
		2400	05	63 40 05
		4800	06	63 40 06
		9600	07	63 40 07
		19200	08	63 40 08
		38400	09	63 40 09
		57600	0A	63 40 0A
RTS / CTS hardware protocol *parameters are modified AFTER the Result Done frame is acknowledged.*	41	disable	00	63 41 00
		enable on each character	01	63 41 01
		enable on whole message	02	63 41 02
flow control timeout *timeout applied to RTS/CTS, ACK/NAK and XON/XOFF flow controls (0 = unlimited timeout)* *parameters are modified AFTER the Result Done frame is acknowledged.*	80	value in milliseconds from 0 to 2550 (1 word)	[0000 … 09F6]	63 80 nn nn
		example: 1000 ms	1000	63 80 03 E8
data bits *fixed ISCP parameter* *modification of this parameter is taken in to account ONLY when not using ISCP*	42	7	00	63 42 00
		8	01	63 42 01
Parity *fixed ISCP parameter* *modification of this parameter is taken in to account ONLY when not using ISCP*	43	none	00	63 43 00
		even	01	63 43 01
		odd	02	63 43 02
stop bits	44	1	00	63 44 00
		2	01	63 44 01
ENQ *(not for use with ISCP)*	45	disable	00	63 45 00
		enable	01	63 45 01
ENQ character *(not for use with ISCP)*	46	ASCII value (1 byte)	[00..FF]	63 46 nn
		example: ENQ	05	63 46 05

ACK *(not for use with ISCP)*	47	disable	00	63 47 00
		enable	01	63 47 01
ACK character *(not for use with ISCP)*	48	ASCII value (1 byte)	[00..FF]	63 48 nn
		example: ACK	06	63 48 06
NAK *(not for use with ISCP)*	49	disable	00	63 49 00
		enable	01	63 49 01
NAK character *(not for use with ISCP)*	4A	ASCII value (1 byte)	[00..FF]	63 4A nn
		example: NAK	15	63 4A 15
XON / XOFF software protocol *(not for use with ISCP)*	4B	disable	00	63 4B 00
		enable	01	63 4B 01
LRC (longitudinal redundancy check)	4C	disable	00	63 4C 00
		enable	01	63 4C 01

9.1.22. Protocol <SG> = 0x61

setup function	FID	parameter	value	string
activation	40	none	00	61 40 00
		ISCP	01	61 40 01

9.1.23. ISCP parameters <SG> = 0x73

setup function	FID	parameter	value	string
data format	40	raw format data *barcode data is sent without a frame and no acknowledgment is necessary*	00	73 40 00
		packet format data *data is sent to the host in an ISCP frame*	01	73 40 01
TFS (transmission frame size) *length of the longest frame that can be received by the host* *MTFS can be found by sending a Status Read Frame <STR> (see section 9.3.1)*	80	value from 32 to MTFS of scanner (1 word)	[0020… MTFS]	73 80 nn nn
		example: 32 *value = scanner's maximum transmission frame size*	00	73 80 00 20
Setup Modification by Configuration Bar Code event *(only applicable when sending data in packet format)*	41	disable	00	73 41 00
		enable	01	73 41 01
Configuration Bar Code Rejected event *(only applicable when sending data in packet format)*	42	disable	00	73 42 00
		enable	01	73 42 01
Unsuccessful Decoding event *(only applicable when sending data in packet format)*	46	disable	00	73 46 00
		enable	01	73 46 01
Start of Read Session event *(only applicable when sending data in packet format)*	47	disable	00	73 47 00
		enable	01	73 47 01
End of Read Session event *(only applicable when sending data in packet format)*	48	disable	00	73 48 00
		enable	01	73 48 01

Start-Up event *When activated, this event is managed differently than the others. See section 9.4.2, Start-Up event.* *(only applicable when sending data in packet format)*	49	disable	00	73 49 00
		enable	01	73 49 01
Trigger Pulled event *(only applicable when sending data in packet format)*	4A	disable	00	73 4A 00
		enable	01	73 4A 01
Trigger Released event *(only applicable when sending data in packet format)*	4B	disable	00	73 4B 00
		enable	01	73 4B 01

9.2. Control Groups <CG>

The Control Group commands are used when the host wants to control the scan engine. Control Groups are used with the following frame types:

sent from the host

Control Command	<CCMD>	0x42

sent from the scanner

Result	<RSLT>	0x51

Example – Control Command

Hardware – good read led on

Using the hexadecimal values from the table, the following frame is a Control Command that tells the scanner to turn on the good read led:

<STX>	<SN>	**<CCMD>**	**<CG>**	**<FID>**	**<parm>**	<FM>	<CHK>	<ETX>
02	1C	**42**	**30**	**40**	**01**	62	**03** D6	03

HOWEVER, you will notice that there is a 0x03 in the checksum. Since this is the <ETX> value, DLE must be applied.

Control Command after applying DLE:

<STX>	<SN>	**<CCMD>**	**<CG>**	**<FID>**	**<parm>**	<FM>	<CHK>	<ETX>
02	1C	**42**	**30**	**40**	**01**	62	**10 43** D6	03

Note: Don't forget, DLE values are NEVER taken in to account when calculating the checksum, ONLY when sending values to the host system.

9.2.1. Decoding <CG> = 0x20

A Decode On/Off command starts or stops a reading session (lighting on, decoding process on and data transmission after a successful decode). However, the scanner behaves differently according to the Trigger Mode used. See the *Trigger Mode* section in Chapter 8.

Several actions can start or stop a reading session (trigger line activation/deactivation or decode on/off). In all possible cases, the scanner always executes the last order received.

command function	FID	parameter	value	string
decode *starts and stops a reading session (lighting on, decoding process on and data transmission)*	40	off	00	20 40 00
		on	01	20 40 01

9.2.2. Hardware <CG> = 0x30

command function	FID	parameter	value	string
good read led *switches the good read led on/off* *scanner always executes the last order received*	40	off	00	30 40 00
		on	01	30 40 01
buzzer *during beep, no other control commands are accepted* *after beep completion scanner sends the Result Done frame* *host can generate a user defined beep sequence*	C0	parameter 1 *value in Hz from 1000 to 4095 (1 word)* parameter 2 *on-time value in milliseconds from 0 to 4095 (1 word)* parameter 3 *off-time value in milliseconds from 0 to 4095 (1 word)* *size of the character string precedes the parameters*	[03E8.. 0FFF] [0000.. 0FFF] [0000.. 0FFF]	30 C0 nn nn nn nn nn nn … …
		example: p1 = 1000 Hz p2 = 100 ms p3 = 0 ms *all three parameters are sent in the same frame*	03 E8 00 64 00 00	30 C0 00 06 03 E8 00 64 00 00
beep sequence *generates a predefined beep sequence*	41	beep sequence 1 *frequency = setup tone* *number of beeps = 6* *beep duration = 40ms* *duration between beeps = 30ms*	00	30 41 00
		beep sequence 2 *frequency = setup tone* *number of beeps = setup good read beep number* *beep duration = setup good read beep duration* *duration between beeps = 60ms*	01	30 41 01
		beep sequence 3 *frequency = setup tone* *number of beeps = 2* *beep duration = 50ms then 100ms* *duration between beeps = 50ms*	02	30 41 02
		beep sequence 4 *frequency = setup tone* *number of beeps = 1* *beep durations = 80ms*	03	30 41 03
sleep *deactivates the power hold signal (see Power Hold Function in section 9.1.19, Trigger Settings)*	03	no parameter		30 03
flash memory upgrade *puts the scanner in firmware download mode (refer to the Firmware Download Technical Reference Manual)*	01	no parameter		30 01

reset *same as turning off and turning back on the scanner*	02	no parameter		30 02

9.2.3. Configuration <CG> = 0x40

command function	FID	parameter	value	string
administrator reset factory defaults *resets all parameters – including locked parameters*	02	no parameter		40 02
disable all symbologies	03	no parameter		40 03

9.2.4. Operating <CG> = 0x50

command function	FID	parameter	value	string
silent mode *temporary silent mode* *scanner cannot spontaneously send frames (barcode data or event frame)* *host may configure scanner without risk of contention* *silent mode is disabled when the exit silent mode command is sent, a decode on command or the scanner is turned off*	40	exit silent mode	00	50 40 00

Note: The scanner automatically enters Silent Mode after receiving an Abort frame from the host. This is the only way the scanner can enter Silent Mode. See chapter 5, Special frames.

9.3. Status Groups <STG>

The Status Group commands are used when the host wants to know the value of certain status parameters in the scanner. Status Groups are used with the following frame types:

sent from the host

Status Read	<STR>	0x43

sent from the scanner

Status Reply	<STRP>	0x53

Example – Status Read

Hardware – firmware version

Using the hexadecimal values from the table, the following frame is a Status Read that asks the scanner to communicate the current firmware version:

<STX>	<SN>	**<STR>**	**<STG>**	**<FID>**	<FM>	<CHK>	<ETX>
02	24	**43**	**30**	**C0**	62	04 32	03

9.3.1. Hardware <STG> = 0x30

status function	FID	parameter	value	string
MRFS (maximum reception frame size) *the maximum length of a frame that can be processed by the scanner* *frames sent by the host must be equal to or shorter than this size* *return value = 1 word*	80	no parameter		30 80
MTFS (maximum transmission frame size) *the maximum length of a frame that can be sent by the scanner* *the Transmission frame size parameter is set to this size by default* *return value = 1 word*	81	no parameter		30 81
firmware version *scanner reports the firmware version present in the flash memory* *return value = ASCII string (first 2 bytes indicate string size)*	C0	no parameter		30 C0
hardware identifier *reports the hardware identifier present in the scanner* *return value = 1 word*	82	no parameter		30 82

9.4. Event Groups <EG>

Event Group commands are used to inform the host when certain events have taken place in the scanner. Event Groups are used with the following frame type:

sent from the scanner

Event Notification	<EVT>	0x61

Note: An Event Notification is a scanner-initiated frame. The scanner must be configured to send an Event Notification. See Setup Groups, ISCP Parameters.

Example – Event Notification

Configuration – setup modification – codabar – activation

Using the hexadecimal values from the table, the following frame is an Event Notification sent by the scanner to notify the host that the symbology Codabar has been activated by reading a configuration bar code:

<STX>	<SN>	**<EVT>**	**<EG>**	**<FID>**	**<parm>**	<FM>	<CHK>	<ETX>
02	00	**61**	**40**	**80**	**40 40**	62	07 28	03

Notice, as with all frame sent by the scanner, the SN is 0.

9.4.1. Decoding <EG> = 0x20

event function	FID	parameter	value	string
unsuccessful decoding	20	no parameter		20 20
start of read session	21	no parameter		20 21
end of read session	22	no parameter		20 22

9.4.2. Hardware <EG> = 0x30

event function	FID	parameter	value	string
start-up *after the scanner sends this frame, it MUST then receive a low level ACK frame. If the scanner never receives the ACK frame, it will continually reset itself until the ACK frame is received.*	20	no parameter		30 20
trigger pulled	21	no parameter		30 21
trigger released	22	no parameter		30 22

9.4.3. Configuration <EG> = 0x40

event function	FID	parameter	value	string
setup modification by reading a configuration bar code	80	<SG> <FID>		40 80 <SG> <FID>
configuration bar code rejected	20	no parameter		40 20

10. Using configuration bar codes

10.1. Configuration bar codes

Most ISCP commands listed in this guide have a corresponding configuration bar code which can be found in the EasySet System configuration software. However, certain configuration bar codes modify more than one ISCP parameter. The values used for the ISCP commands are not the same values used in the configuration bar codes.

Refer to your product in EasySet System configuration software to print out configuration bar codes. EasySet is on the Intermec products cd-rom or you can download the latest version from the data capture website at http://datacapture.intermec.com.

10.2. Permissions and configuration bar codes

Permissions (see 3.3.6, Setup Permission Write <SPW> 0x45) allow the user to lock one or more parameters so that when reading configuration bar codes, the locked parameters are not modified. This helps to avoid unwanted changes in the scanner configuration.

We recommend locking all necessary parameters to assure that your application works correctly.

10.3. Reset factory defaults

There are two configuration bar codes for resetting the factory defaults in the scanner: Reset Factory Defaults and Administrator Reset Factory Defaults.

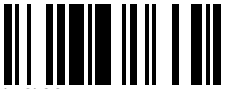
Reset Factory Defaults

Used to reset the scanner when **NO** parameters are locked. If one or more parameters are locked, the scanner refuses to read this configuration bar code. This barcode is available in EasySet.

Administrator Reset Factory Defaults

Used to reset **ALL** parameters – even parameters that are locked. This bar code is only available in this manual.

Administrator Reset Factory Defaults



\59\60

11. Index

T

U

W

X