baracoda™
Traceability

RoadRunners Evolution
Communication Protocol

©Baracoda ™ – January 2009

SUMMARY

Revision History

Changes to the original manual are listed below.

Document	Date	Description
1.0	02 July 08	Initial release
1.2	03 July 08	Presentation updated
1.3	11 July 08	"Data string format" section updated (2.2.1)
1.4	29 Sept. 08	Added new commands : 0x5A-0x5B : Get/Set master mode Changed "barcode prefix/suffix" into "capture prefix/suffix" Changed §2.3.5 "barcode messages" into "capture messages"
1.5	02 Jan. 09	Add "Decoder Communication Protocol" section Add "Appendix 1 – Bluetooth Protocol" section

1. Introduction

1.1. Generalities

RoadRunners Evolution is a wireless data capture product.

This document is detailing the protocol of communication between the Baracoda scanner RoadRunners Evolution and its foreign environment through Radio Frequency link (ie: Bluetooth).

Wireless communication is based on the Bluetooth protocol, thanks to the embedded Baracoda Equinox Bluetooth Stack.

Barcode reading capabilities are enabled thanks to a CMOS technology (1D & 2D) or laser (1D).

The messages described in this document can be:

Host to scanner messages: the packet is sent only by the host to the scanner

Scanner to host messages: the packet is sent only by the scanner to the host

Bidirectional messages: the packet format is the same whether it is sent by the host or the scanner

http://www.baracoda.com

1.2. Generic packet

All the frames described in this document are formatted as shown:

Code ID	Length	Payload
1 Byte	2 Bytes	N Bytes

- 1 byte for code ID

Bits 7:5 is the logical device

Bits 4:1 is the command

Bit 0: when set, the message must be acknowledged

- 2 bytes for the size of the payload (big-endian), including the sequence number byte which is considered as part of the payload

- Payload (including 1 byte for sequence number when applicable).

The response will have the same code ID as the command.

2. Communication protocol

2.1. Bidirectional packets

2.1.1. Control messages

2.1.1.1. Specific packets

Code ID	Description	Frame
0x01	Legacy	0x01 0x01 0x01 Or 0x01 0x02 0x01

These two (2) sequences will be recognized and purged for backward compatibility with older Baracoda products.

2.1.1.2. Acknowledgment packets

Code ID	Description	Frame
0x06	ACK	0x06 0x01 0xYY
0x15	NACK	0x15 0x01 0xYY

These messages acknowledge the reception of a valid message with the expected sequence number 0xYY, before processing it.
For captured data from the scanner, ACK and NAK have the same meaning but will trigger a different event on the scanner.

2.1.1.3. Synchronization packet

Code ID	Description	Frame
0x16	SYN	0x16 0x01 0xYY

This message acknowledges the reception of a message to acknowledge with an unexpected sequence number. 0xYY is the expected sequence number.
The device will resynchronize its remote sequence number when receiving this message.

2.2. Scanner to host messages

2.2.1. Encapsulation scheme

Code ID	Description	Payload
0x32–0x33	**Barcode data**	DATA string

2.2.2. Data string format

Timestamp	Data Prefix	Capture Prefix	Symbology Prefix	AIM Identifier	Barcode / RFID data	Symbology suffix	Capture suffix	Data suffix
12 bytes	0-32 bytes	0-32 bytes	0-4 bytes	0, 2 or 3 bytes	-	0-4 bytes	0-32 bytes	0-32 bytes

2.3. Host to scanner messages

2.3.1. Communication messages

Code ID	0x40-0x41
Description	**Get Communication Descriptor**
Payload	None
Response	2 bytes: {Wireless link: (Bit 0: Bluetooth)} {Wired link: (Bit 0: Serial Dock)}

Code ID	0x42-0x43
Description	**Get Retransmission Parameters**
Payload	None
Response	2 bytes: {Max number of retransmission, 1 to 0xFE, 0xFF = infinity} {Delay between transmission, 1 to 0xFF, in tenth of seconds}

Code ID	0x44-0x45
Description	**Set Retransmission Parameters**
Payload	2 bytes: {Max number of retransmissions, 1 to 0xFE, 0xFF = infinity} {Delay between transmissions, 1 to 0xFF, in tenth of seconds}
Response	1 byte: {(Bit 0: Success)}

Code ID	0x46-0x47
Description	**Get Capture Frame Format**
Payload	None
Response	1 byte {0 = Baracoda, 1 = Baracoda + ACK, 2 = Raw}

Code ID	0x48-0x49
Description	**Set Capture Frame Format**
Payload	1 byte {0 = Baracoda, 1 = Baracoda + ACK, 2 = Raw}
Response	1 byte: {Bit0: Success}

Code ID	0x50-0x51
Description	**Lock/Unlock Scan engine**
Payload	1 byte : {0 = Unlock, 1 = Lock}
Response	1 byte : {Bit 0 : Success}

Code ID	0x52-0x53
Description	**Disable Barcode Programming Functionality**
Payload	1 byte : {1 = Disable barcode programming functionality, 0 = Enable}
Response	1 byte : {Bit 0 : Success}

Code ID	0x54-0x55
Description	**Get/Set dock use**

Payload	Get : None
	Set : 1 byte :
	     {0 = Disable communication via RS232 dock
	      1 = Enable}

Response	Get : 1 byte :
	     {0 = Disable communication via RS232 dock
	      1 = Enable}
	Set : 1 byte :
	     {Bit 0 : Success}

Code ID	0x56-0x57
Description	**Get/Set In charge behavior**
Payload	Get : None Set : 1 byte : {shutdown timeout (0 = leave current timeouts (default), 255 = infinity)}

Response	Get : 1 byte :
	     {shutdown timeout (0 = leave current timeouts (default), 255 = infinity)}
	Set : 1 byte :
	     {Bit 0 : Success}

Remarks	When scanner in charge, the shutdown timers can be modified

Code ID	0x5A-0x5B
Description	**Get/Set master mode**
Payload	Get : None Set : 1-2 byte(s) : 　　　　{1 : enable, 0 : disable} 　　　　[OPTIONAL]{1 : disconnect the scanner if connected}
Response	Get : 1 byte : 　　　　{1 : enabled 　　　　0 : disabled} Set : 1 byte : 　　　　{Bit 0 : Success}

Code ID	0x5E-0x5F
Description	**Bluetooth Commands**
Payload	{Code ID} "Parameters"
Response	If the device responds: {Code ID} "Response" Else: {0}

Bluetooth specific commands from the Platform2 Bluetooth communication protocol are to be framed within the payload of this message.

2.3.2. Scanner messages

Code ID	0x60-0x61
Description	**Get Scanner Status**
Payload	None
Response	2 bytes: {(Bit 7: Upgrading) (Bit 1: Docked) (Bit 0: Charging)} {Battery level, 0 to 100}

Code ID	0x62-0x63
Description	**Get Operating Mode**
Payload	None
Response	1 byte: {Bit 0 = 0:real time, Bit 0 = 1: batch} {(Bit 7: limited)}
Remarks	"limited" means barcode buffer = 0 when in real time, no data loss mode and disconnected

Code ID	0x64-0x65
Description	**Set Operating Mode**
Payload	1 byte: {Bit 0 = 0: real time, Bit 0 = 1: batch} If real time mode is set : {(Bit 7: limited)(Bit 6: ACK beep) (Bit 5: no ACK beep)} NOTE : the ACK beep enable / disable is only effective when Capture Frame Format is "Baracoda + ACK"
Response	1 byte: {Bit 0: Success}
Remarks	Batch mode is not available for D-Fly scanner

Code ID	0x66-0x67
Description	**Get Shutdown Timers**
Payload	None
Response	2 bytes: {Number of minutes before shutdown when connected, 1 to 0xFE, 0xFF = infinity} {Number of minutes before shutdown when disconnected, 1 to 0xFE, 0xFF = infinity}

Code ID	0x68-0x69
Description	**Set Shutdown Timers**
Payload	2 bytes: {Number of minutes before shutdown when connected, 1 to 0xFE, 0xFF = infinity}

	{Number of minutes before shutdown when disconnected, 1 to 0xFE, 0xFF = infinity}
Response	1 byte: {Bit 0: Success}

Code ID	0x6A-0x6B
Description	**Get RTC time**
Payload	None
Response	6 bytes: {YY}{MM}{DD}{HH}{MM}{SS}

Code ID	0x6C-0x6D
Description	**Set RTC time**
Payload	6 bytes: {YY}{MM}{DD}{HH}{MM}{SS}
Response	1 byte: {Bit 0: Success}

Code ID	0x72-0x73
Description	**Get/Set Delivery Mode Parameters**
Payload	Get : None Set : 1, 3 or 5 bytes : {Mode}[{min len recepisse}{max len recepisse}][{min len statut}{max len statut}]
Response	Get : 5 bytes {Mode}{min len recepisse}{max len recepisse}{min len statut}{max len statut} Set : 1 byte: {Success?}
Remarks	Mode : {(bit 7 : 1 = need a validation barcode for sending, 0 no need) (bits[3:0] : 0 = disabled, 1 = BBox send via email, 2 = BBox send via Socket)} Default values : recepisse : min = 8 max = 16 ; statut : min = 3 max = 28

Code ID	0x74-0x75
Description	**Restore defaults settings**
Payload	None
Response	1 byte: {Bit 0: Success}
Remarks	External Flash memory is also erased

Code ID	0x76-0x77
Description	**Get Product Version**
Payload	None
Response	x bytes : «Baracoda RRNA…»

Code ID	0x78-0x79
Description	**Get Switching On Delay**
Payload	None
Response	1 byte : {1 = 0 second, 2 = 1 second, 3 = 2 seconds}

Code ID	0x7A-0x7B
Description	**Set Switching On Delay**
Payload	1 byte : {1 = 0 second, 2 = 1 second, 3 = 2 seconds}
Response	1 byte : {Bit 0 :Success}

2.3.3. User Interface messages

LED 1 : left LED

LED 0 : right LED

Code ID	0x80-0x81
Description	**Get MMI Descriptor**
Payload	None
Response	2 bytes: {(Bit 6: Blue LED 1) (Bit 5: Red LED 1) (Bit 4: Green LED 1) (Bit 2: Blue LED 0) (Bit 1: Red LED 0) (Bit 0: Green LED 0)} {(Bit 0: Buzzer)}

Code ID	0x82-0x83
Description	**Get MMI Mode**
Payload	None
Response	1 byte: {(Bit 1: Buzzer Enabled) (Bit 0: LEDs enabled)}

Code ID	0x84-0x85
Description	**Set MMI Mode**
Payload	1 byte: {(Bit 1: Buzzer Enabled) (Bit 0: LEDs enabled) (Bit 7 = 0:Buzzer config select, =1: Buzzer config deselect)(Bit 6 = 0:leds config select, =1: leds config deselect)}
Response	1 byte: {(Bit 0: Success)}

Code ID	0x86-0x87
Description	**Get MMI Signal (User interface)**
Payload	1 byte: {Signal number, 0 - 3}
Response	(1 + 3n) bytes: {Number of steps, 0 - 4} For each step: {(Bit 6: Blue LED 1) (Bit 5: Red LED 1) (Bit 4: Green LED 1) (Bit 2: Blue LED 0) (Bit 1: Red LED 0) (Bit 0: Green LED 0)} {Buzzer frequency, 0 – 0xFF * 50Hz = 0 – 12750Hz} {Delay until next step, in tenth of seconds}

Code ID	0x88-0x89
Description	**Set MMI Signal**
Payload	(2 + 3n) bytes

| | {Signal number, 0 - 3}
{Number of steps, 0 - 4}
For each step:
{(Bit 6: Blue LED 1) (Bit 5: Red LED 1) (Bit 4: Green LED 1) (Bit 2: Blue LED 0) (Bit 1: Red LED 0) (Bit 0: Green LED 0)}
{Buzzer frequency, 0 – 0xFF * 50Hz = 0 – 12750Hz}
{Delay until next step, in tenth of seconds} |
| Response | 1 byte: {(Bit 0: Success)} |

Code ID	0x8A-0x8B
Description	**Play Signal**
Payload	2 bytes: {Signal number, 0 - 3} {Number of loops, 0 – 0xFE, 0xFF = infinity}
Response	1 byte: {(Bit 0: Success)}

Code ID	0x8C-0x8D
Description	**Stop Signal**
Payload	1 byte: {Signal number, 0 – 3}
Response	1 byte: {(Bit 0: Success)}

The list of MMI signals is:

IHM_SIGNAL_CAPTURE_READ	:	0
IHM_SIGNAL_CAPTURE_ACK	:	1
IHM_SIGNAL_CAPTURE_NAK	:	2
IHM_SIGNAL_CAPTURE_LOST	:	3
IHM_SIGNAL_USER_DEFINED	:	13

Code ID	0x92-0x93
Description	**Get Beeps mode**
Payload	
Response	1 byte: {(Bit 0: Beep level 0=low, 1=high) (Bit 1: Read beep) (Bit 2: ACK beep)}

Code ID	0x94-0x95
Description	**Set Beeps mode**
Payload	1 byte: {(Bit 0: Beep level; 0=low, 1=high) (Bit 1: Read beep) (Bit 2: ACK beep)

	(Bit 7 = 0: ACK beep config select, =1: ACK beep config deselect)
	(Bit 6 = 0: Read beep config select, =1: Read beep config deselect)
	(Bit 5 = 0: Beep level config select, =1: Beep level config deselect)}}
Response	1 byte:
	{(Bit 0: Success)}

2.3.4. Miscellaneous messages

Code ID	0xC2-0xC3
Description	**Get/Set DataPrefix**
Payload	Get : None
	Set : 1-33 bytes:
	{ DataPrefix length}
	"DataPrefix String"
Response	Get : 1-33 bytes:
	{ DataPrefix length}
	"DataPrefix String"
	Set : 1 byte:
	{Success?}

Code ID	0xC4-0xC5
Description	**Get/Set DataSuffix**
Payload	Get : None
	Set : 1-33 bytes:
	{ DataSuffix length}
	"DataSuffix String"
Response	Get : 1-33 bytes:
	{ DataSuffix length}
	"DataSuffix String"
	Set : 1 byte:
	{Success?}

Code ID	0xC6-0xC7
Description	**Get/Set Data Format 2**
Payload	Get : None
	Set : 1 byte:
	{(Bit 5 = 0: DataPrefix config select, =1: DataPrefix config deselect)
	(Bit 4 = 0: DataSuffix config select, =1: DataSuffix config deselect)
	(Bit 1: DataPrefix)
	(Bit 0: DataSuffix)
Response	Get : 1 byte:
	{(Bit 1: DataPrefix)
	(Bit 0: DataSuffix)}
	Set : 1 byte:
	{(Bit 0:Success)}

Code ID	0xC8-0xC9
Description	**Get / Set Barcode length**
Payload	Get : None

Response	Set : 2 bytes
	{authorized barcode length. 0 = disabled}
	Get : 2 bytes
	{authorized barcode length. 0 = disabled}
	Set : 1 byte
	{(Bit 0:Success)}

Code ID	0xCA-0xCB
Description	**Get and erase stored barcodes with no data loss mode**
Payload	None
Response	1 byte
	{(Bit 0:Success)}

Code ID	0xCC-0xCD
Description	**Reset modes**
Payload	None = restore defaults, keep link keys, reboot scanner
	1byte :
	0 = restore defaults, keep link keys, reboot scanner
	1 = switch off scanner (no restoring defaults)
	2 = reboot scanner (no restoring defaults)
Response	1 byte
	{(Bit 0:Success)}

Code ID	0xCE-0xCF
Description	**Batch upload commands**
Payload	{Code ID} "Parameters" (cf. below)
Response	{Code ID} "Response"

The UPLOAD Code IDs are:

Code ID	0
Description	Launch upload process (typically used only for the upload barcode)
Payload	1 byte:
	{0 mandatory}
Response	1 byte:
	{Bit 0: Success}

Code ID	1
Description	Ready to start upload (Scanner to host message)
Payload	3 bytes :
	{0 mandatory}
	{ number of elements to be uploaded MSB }
	{ number of elements to be uploaded LSB }
Response	None

Code ID	2
Description	Start uploading barcodes
Payload	1 byte: {0 mandatory}
Response	1 byte: {Bit 0: Success}

Code ID	3
Description	RESERVED
Payload	N/A
Response	N/A

Code ID	4
Description	Set upload status and end process
Payload	2 bytes : {0 mandatory} {1 : upload successful, data can be erased from the scanner 0 : upload failed, do not erase data}
Response	1 byte: {Bit 0: Success}

Code ID	0xD2-0xD3
Description	**Get/Set Anti duplicate scans**
Payload	Get : None Set : 1 byte {0 = disabled 1 = no consecutive duplicate scans + error signal 2 = no consecutive duplicate scans + no decoding}

Response	Get : 1 byte
	      {0 = disabled
	       1 = no consecutive duplicate scans + error signal
	2 = no consecutive duplicate scans + no decoding }
	Set : 1 byte
	      {(Bit 0:Success)}

Comments	The comparison will be made over the 32 first characters of the barcodes only.

Code ID	0xD4-0xD5
Description	**Restore last batch**
Payload	None or 1 byte (optional): {1 = upload data after retrieving}
Response	1 byte: {(Bit 0:Success)}
Comments	This is only available if no new scan has been made.

Code ID	0xD8-0xD9
Description	**Enable remote trigger**
Payload	None : use default 5s timeout
	1 byte : {timeout (s)}
Response	1 byte
	{(Bit 0:Success)}

2.3.5. Capture messages

Code ID	0xE0-0xE1
Description	**Get Capture Version**
Payload	None
Response	"Capture Version String" or {0} if not applicable
Remarks	Capture Version Strings can be : "EV10" →the scan engine is an Intermec EV10 "DECODED_EV15" → the scan engine is an Intermec EV15 in decoded mode "IT4180" → the scan engine is a 2D HHP one (IT4180/5180 Series).

Code ID	0xE2-0xE3
Description	**Get Mode**
Payload	None
Response	1 byte: {0 = trigger, 1 = smart autoscan, 2 = disabled, 3= autoscan, 4=aiming trigger scan}

Code ID	0xE4-0xE5
Description	**Set Mode**
Payload	1 byte OR 2 bytes if aiming trigger scan mode {0 = trigger, 1 = smart autoscan, 2 = disabled, 3= autoscan, 4=aiming trigger scan} {aiming trigger scan mode timeout value in second}
Response	1 byte: {(Bit 0: Success)}

Code ID	0xE6-0xE7
Description	**Get Data Format**
Payload	None
Response	1 byte: {(Bit 2:Timestamp) (Bit 1: Capture Prefix) (Bit 0: Capture Suffix)}

Code ID	0xE8-0xE9
Description	**Set Data Format**
Payload	1 byte: {(Bit 7 = 0: Timestamp config select, =1: Timestamp config deselect) (Bit 6 = 0: Capture Prefix config select, =1: Capture Prefix config deselect) (Bit 5 = 0: Capture Suffix config select, =1: Capture Suffix config deselect) (Bit 2:Timestamp) (Bit 1: Capture Prefix) (Bit 0: Capture Suffix) }
Response	1 byte: {(Bit 0:Success)}

Code ID	0xEA-0xEB
Description	**Get Capture Prefix**
Payload	None
Response	1-33 bytes:
	{ Capture Prefix length}
	"Capture Prefix String"

Code ID	0xEC-0xED
Description	**Set Capture Prefix**
Payload	1-33 bytes:
	{ Capture Prefix length}
	"Capture Prefix String"
Response	1 byte:
	{(Bit 0:Success)}

Code ID	0xEE-0xEF
Description	**Get Capture Suffix**
Payload	None
Response	1-33 bytes:
	{ Capture Suffix length}
	"Capture Suffix String"

Code ID	0xF0-0xF1
Description	**Set Capture Suffix**
Payload	1-33 bytes:
	{ Capture Suffix length}
	"Capture Suffix String"
Response	1 byte:
	{(Bit 0:Success)}

Code ID	0xF4-0xF5
Description	**Set Timestamp**
Payload	6 bytes:
	{YY}{MM}{DD}{HH}{MM}{SS}
Response	1 byte:
	{Bit 0: Success}

Code ID	0xF6-0xF7
Description	**Set AIM Symbology Identifier transmission**
Payload	1 byte:
	{0 : disabled, 1 : enabled}
Response	1 byte:
	{Bit 0: Success}

Code ID	0xF8-0xF9
Description	**Get stored barcodes count**
Payload	None
Response	2 bytes: {Stored barcodes count [15:8]} {Stored barcodes count [7:0]}

Code ID	0xFC-0xFD
Description	**Erase stored barcodes**
Payload	None
Response	1 byte: {(Bit 0:Success)}

Code ID	0xFE-0xFF
Description	**Barcode decoder Commands**
Payload	{header} {type} {size} "Parameters"
Response	If the device responds: {Code ID}{length}{header}{type} {response size} "Response" Else: {0}

Barcode decoder specific commands from the Platform2 Decoder communication protocol are to be framed within the payload of this message.

2.4. Decoder Communication Protocol

2.4.1. frame format

Header	Type	Size (Bytes)	Command
1Byte	1Byte	1Byte	

2.4.2. Header

The Header field defines the type of symbology to select; it is 1 byte long (ACSII code):

Header	Selected Symbology
A	SELECT ALL
B	Code 93
C	Code 128 / EAN 128
D	EAN 13 / UPC A
E	Code 39
F	Codabar
G	Interleaved 2 of 5
H	Standard 2 of 5 (industrial 2 of 5)
I	Matrix 2 of 5 (symbology disabled)
J	Code 11
K	MSI
L	UPC E
M	EAN 8
N	RSS14 (not available on RoadRunners product)
O	RSSLTD (not available on RoadRunners product)

Note: The "A" header (SELECT ALL) allows the selection of all the symbologies available. Thus, only general commands will be allowed.

2.4.3. Type

The Type field defines the type of command to be sent to the reader, it is 1 byte long.

Type	Description
A	**Get config**: asks the reader to give the configuration options for the selected symbology. (1) (*)
B	**Set config** : sets an options configuration for the selected symbology. (1)
C	**Set Default:** sets the default options configuration for the selected symbology(ies).
D	**Usual Command.**
E	**Special Command** (with parameters). (2)

(*): This Type of command is not available with "A" header.

(1): This command concerns the whole set of options available for one symbology. Its description will be given in the section "Command field".

(2): This type is used for commands requiring non Boolean information. Their length will be at least 2 bytes, the first one defining the type of command, the other(s) being the parameter(s) to use. More details will be given in the section "Command field".

All the commands will answer "0" if the frame is wrong.

Commands with type B, C, D or E will answer "1" as an acknowledgment of good reception of the command.

The "Get config" command (type A) will answer 2 or 4 bytes : the two firsts follow the format described below (see "set config" command field). The third and fourth bytes correspond to minimum and maximum lengths if the selected symbology supports this option.

2.4.4. Size

This field specifies the length (bytes) of the following field (commands). It will be set to "0" if the type was "A" (Get Config) or "C" (Set Defaults),

2.4.5. Command

This field contains the commands, its length must be the one specified in the Size field.

There are five types of commands:

2.4.5.1. Set Config (Type "B")

This command is made up of 1 or 2 bytes. The first one contains information for configuration of general options (common to all the symbologies). The second one, optional, relates to specific options to each symbology.

For a Set Config (type "B") with a SELECT ALL (header "A"), the command sent can only be 1 byte long.

Note: A high level bit ("1") corresponds to an Enable, a "0" bit means Disable.

BYTE 1 (general options):

The format of this byte is the same for all symbologies.

	Bit	Option
LSB	0	Enable/Disable Symbology
	1	Enable/Disable Min. length (1)
	2	Enable/Disable Checksum calculation (2)
	3	Enable/Disable Checksum transmission
	4	Enable/Disable Max. length (3)
	5	Enable/Disable symbology prefix (4)
	6	Enable/Disable symbology suffix (4)
MSB	7	FREE

(1): If Min. length is enabled without having been set with the special command, the default minimal length will be 6 characters for all symbologies.

(2): This option will not have any effect on symbologies that require a checksum (EAN/UPC, code93, Code128, RSS). Concerning the symbologies that allow two check digits (MSI, code11), the first check digit is obligatory. Thus, this option will affect the calculation/non calculation of the second check digit.

(3): If Max. length is enabled without having been set with the special command, the default minimal length will be 32 characters for all symbologies.

(4): if the prefix/suffix is enabled without having been defined at least once (cf. special command), there will be no effect.

BYTE 2 (specific options):

Each symbology will have a different configuration of this byte, depending on the specific options available on each.

CODE 93 (Header "B")

	Bit	Option
LSB	0	FREE
	1	FREE
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

CODE 128 / EAN 128 (Header "C")

	Bit	Option
LSB	0	GS transmit (EAN128)
	1	AIM Symb ID transmit (EAN128)
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

EAN 13 / UPC-A (Header "D")

	Bit	Option
LSB	0	Number System transmitted (UPC A)
	1	Enable/disable ISBN and ISSN
	2	ISSN hyphen transmission
	3	ISSN price code transmission
	4	UPC-A, transmitted as EAN 13
	5	Add-on Digits required/not required
	6	Enable/disable Add-on 2
MSB	7	Enable/disable Add-on 5

CODE 39 (Header "E")

	Bit	Option
LSB	0	Enable/Disable start-stop transmission
	1	Enable/Disable Full ACSII Mode
	2	Enable/Disable "*" as start-stop character
	3	Enable/Disable "$" as start-stop character
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

CODABAR (Header "F")

	Bit	Option
LSB	0	Enable/Disable start-stop transmission
	1	FREE
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

INTERLEAVED 2 OF 5 (Header "G")

	Bit	Option
LSB	0	FREE
	1	FREE
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

STANDARD 2 OF 5 (Header "H")

	Bit	Option
LSB	0	FREE
	1	FREE
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

MATRIX 2 OF 5 (Header "I")

	Bit	Option
LSB	0	FREE
	1	FREE
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

CODE 11 (Header "J")

	Bit	Option
LSB	0	FREE
	1	FREE
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

MSI (Header "K")

	Bit	Option
LSB	0	FREE
	1	FREE
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

UPC-E (Header "L")

	Bit	Option
LSB	0	Number System transmitted
	1	-
	2	-
	3	-
	4	FREE
	5	UPC-E transmitted as UPC-A
	6	-
MSB	7	FREE

EAN 8 (Header "M")

	Bit	Option
LSB	0	FREE
	1	-
	2	-
	3	-
	4	EAN 8 transmitted as EAN 13
	5	FREE
	6	-
MSB	7	FREE

RSS 14 (Header "N")

	Bit	Option
LSB	0	LINKAGE FLAG PRINT
	1	APPLICATION ID PRINT
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE

MSB	7	FREE

RSS Limited (Header "O")

	Bit	Option
LSB	0	LINKAGE FLAG PRINT
	1	APPLICATION ID PRINT
	2	FREE
	3	FREE
	4	FREE
	5	FREE
	6	FREE
MSB	7	FREE

2.4.5.2.   Get config(Type "A")

This command allows to retrieve the whole configuration of a given symbology. The response to it is made of 2 or 4 bytes:
The two first bytes are the ones described in the above section (set config).
The two following are given only if the length of the barcode is variable with the selected symbology. These bytes are the min and max length of the barcode.

For some settings (e.g. prefix/suffix…), the "special commands" type should be used (see below for details).

2.4.5.3.   Usual Commands (Type "D")

The method described above (set config) allows a fast and effective setting of a whole set of configurations for a given symbology. It exclusively allows a simultaneous configuration of all the options available for a given symbology.

The usual commands are designed to palliate this effect. It is possible, with this type of commands, to modify a limited number of options in a configuration.

A command is one byte long and orders an enabling or a disabling of an option. Several commands can be sent in the same frame by simply specifying the number in the Size field.

The commands table is unique. All the symbologies will thus understand this same table. However, since certain options are not available for some symbologies, the corresponding commands will be quite simply ignored for these symbologies. This will allow the configuration of several symbologies at the time by sending only one frame.

COMMANDS TABLE		
Ascii CODE	**COMMAND**	**RESPONDING HEADERS**
A	Enable Symbology	All
B	Disable Symbology	All
C	Disable Min. length	A, B, C, E, F, G, H, J, K
D	Enable Min. length (1)	A, B, C, E, F, G, H, J, K
E	Enable Checksum calculation (2)	E, F, G, H, J, K
F	Disable Checksum calculation (2)	E, F, G, H, J, K
G	Enable Checksum transmission	C, D, E, F, G, H, J, K, L, M,N,O
H	Disable Checksum transmission	C, D, E, F, G, H, J, K, L, M,N,O
I	Enable start/stop characters transmission	E, F
J	Disable start/stop characters transmission	E, F
K	Start/stop accepted characters "*" only	E
L	Start/stop accepted characters "$" only	E
M	start/stop accepted characters "*" and "$"	E
N	Enable Code 39 full ASCII mode	E
O	Disable Code 39 full ASCII mode	E
P	Enable number system transmission	D, L
Q	Disable number system transmission	D, L
R	Disable prefix	All
S	Enable prefix	All
T	Disable suffix	All
U	Enable suffix	All
V	Enable ISBN and ISSN	D
W	Disable ISBN and ISSN	D
X	UPC-A/EAN 8 transmitted as EAN 13	D, M
Y	UPC-A/EAN 8 transmitted as UPC-A/EAN 8	D, M
Z	UPC-E transmitted as UPC-A	L
a	UPC-E transmitted as UPC-E	L
b	Enable hyphen transmission for ISSN	D
c	Disable hyphen transmission for ISSN	D
d	Disable Max. length	A, B, C, E, F, G, H, J, K
e	Enable Max. length (3)	A, B, C, E, F, G, H, J, K
f	Enable all EAN/UPC symbologies	D, L, M
g	Disable all EAN/UPC symbologies	D, L, M
i	Enable linkage flag transmission	N,O
j	Dissable linkage flag transmission	N,O
k	Enable application identifier transmission	N,O
l	Dissable application identifier transmission	N,O
m	Enable Group separator transmission (EAN128)	C
n	Dissable Group separator transmission (EAN128)	C

o	Enable AIM symbology identifier transmission	C
p	Dissable AIM symbology identifier transmission	C
q	Enable Price Code transmission for ISSN	D
r	Disable Price Code transmission for ISSN	D
s	Add-on Digits not required but transmitted if read	D
t	Add-on Digits required and transmitted	D
u	Enable 2-digit Add-on	D
v	Disable 2-digit Add-on	D
w	Enable 5-digit Add-on	D
x	disable 5-digit Add-on	D

(1): The default minimal length is 6 characters for all symbologies.

(2): This option will not have any effect on symbologies requiring a checksum calculation (EAN/UPC, Code128).
Concerning the symbologies that allow two check digits (code93, code11), the first check digit is obligatory.
Thus, this option will affect the calculation/non calculation of the second check digit.

(3): The default maximal length is 32 characters for all symbologies.

2.4.5.4.   Special Commands (Type "E")

Some commands require more than a Boolean digit and require one or more arguments.

Special commands will be used in this case (defined code "E" in type field). It is made of one byte corresponding to the type of command. Then, depending on the type of command, a certain number of parameters will be expected. Each of these will be 1 byte long.

This type of command allows the sending of multiple commands in one frame. The Size field must then specify the total length, in bytes, of the command field.

SPECIAL COMMANDS

ASCII CODE	DESCRIPTION	PARAMTER(S)	RESPONSE
A	DEFINE AND ENABLE MIN. LENGTH	[MIN LENGTH]	-
B	DEFINE AND ENABLE MAX. LENGTH	[MAX LENGTH]	-
C	DEFINE AND ENABLE LENGTH FRAME	[MIN][MAX]	-
D	DEFINE VOTING VALUE (*)	[VOTING]	-
E	GET VOTING VALUE	-	[VOTING]
F	DEFINE GS REPLACEMENT CHARACTER	[CHAR]	-
G	GET GS REPLACEMENT CHARACTER	-	[CHAR]
H	DEFINE AND ENABLE PREFIX	[length] « prefix »	-
I	DEFINE AND ENABLE SUFFIX	[length] «suffix»	-
J	GET PREFIX	-	[length] « prefix »
K	GET SUFFIX	-	[length] «suffix»
L	FREE	-	-
…	…	…	…

(*): this command will only de effective with 'A' header. Otherwise, it will be ignored. Values accepted for voting are: 2, 3, 4. this command is also unavailable with Pencil2 scanner.

2.4.6. Examples

Here are some examples to illustrate each type of frame to be sent to the decoder and the possible answer from the decoder.

2.4.6.1. Get Config

Example 1.1 – Getting configuration of Code 39 symbology.

Frame to be sent to the decoder:

'E'	'A'	0	-
Header	Type	Size	Command

The decoder answers the following frame:

'E'	'A'	4	$01 , $05, $06, $20
Header	Type	Size	Command

By reading this answer frame, we can check that the decoder has well understood the selected symbology (header "E"). The command field contains the configuration itself:

1st byte : **$01**

	Bit	Option
1	0	Symbology Enabled
0	1	Min. length Disabled
0	2	Checksum calculation Disabled
0	3	Checksum transmission Disabled
0	4	Max. length Disabled
0	5	-
0	6	-
0	7	-

2nd byte : **$05**

	Bit	Option
1	0	start-stop transmission Enabled
0	1	Full ACSII Mode Disabled
1	2	"*" as start-stop character Enabled
0	3	"$" as start-stop character Disabled
0	4	-
0	5	-
0	6	-
0	7	-

3rd byte : Min length = 6
4th byte Max length = $20 = 32

2.4.6.2.   Set Config

Example 2.1 – Setting a configuration of Codabar symbology.

Frame to be sent to the decoder:

'F'	'B'	2	$03 , $01
Header	Type	Size	Command

Reading this frame, the decoder understands:

The user wants to set a configuration (Type "B") for Codabar (Header "F"). the configuration will contain general options and others specific to cadabar (Size = 2). Then, the Command field contains the configuration itself:

1st byte : **$03**

	Bit	Option
1	0	Symbology Enabled
1	1	Min. length Enabled (set to 6 as default)
0	2	Checksum calculation Disabled
0	3	Checksum transmission Disabled
0	4	Max. length Disabled
0	5	-
0	6	-
0	7	-

2nd byte : **$01**

	Bit	Option
1	0	start-stop transmission Enabled
0	1	-
0	2	-
0	3	-
0	4	-
0	5	-
0	6	-
0	7	-

Example 2.2 – Setting a configuration of all symbologies.

Frame to be sent to the decoder:

'A'	'B'	1	$13
Header	Type	Size	Command

Reading this frame, the decoder understands:
The user wants to set a configuration (Type "B") for all the symbologies (Header "A"). The configuration can only contain general options and the Size field must be equal to 1. Then, the Command field contains the configuration itself:

$013	Bit	Option
1	0	Symbology Enabled
1	1	Min. length Enabled (set to 6 as default)
0	2	Checksum calculation Disabled
0	3	Checksum transmission Disabled
1	4	Max. length Enabled (set to 32 as default)
0	5	-
0	6	-
0	7	-

2.4.6.3. Usual command

Example 3.1 – Modifying a configuration of all symbologies.

We now want disable Min length and Enable start/stop characters (for the symbologies having those), regardless of the other options' settings.
Frame to be sent to the decoder:

'A'	'D'	2	'C' , 'I'
Header	Type	Size	Command

Reading this frame, the decoder understands:
The user wants to send a command (Type "D") to all the symbologies (Header "A"). Any command can be sent but not all may be effective on all symbologies.
The command "C" will first be sent to all symbologies, and applied to all of them since they all have this option available.
Then the command "I" will also be sent to all symbologies but only some of them will apply it (Code39, Codabar).

2.4.6.4. Special command

Example 4.1 – Sending a special command to Code39 symbology.

We now want set a larger frame of Min-Max length for Code39 symbology.

Frame to be sent to the decoder:

'E'	'E'	3	'C' , 2 , 40
Header	Type	Size	Command

Reading this frame, the decoder understands:

The user wants to send a special command (Type "E") to Code39 symbology (Header "E").

The size must be at least 2 and the first byte of the command field must contain a code (ACSII) that will tell (indirectly) the number of parameters following.

The command "C" will first read, it means "setting a Min. length and a Max. length". Then the usual commands 'D' and 'e' will be sent to enable Min length and Max. length for Code 39 symbology.

Then the Min length will be set to 2 and the Max. length will be set to 40.

APPENDIX 1: Bluetooth Protocol

The configuration frames are as follows:

Header: 1 Byte	Length: 2 Bytes (MSB, LSB)	Payload: 0 to 65535 Bytes.

Commands

Command	Set Pin Code
Header	0x01 (flash only)
Length	xx xx (new pin size)
Payload	N digits PIN. (Defaut "0000")
Response	0x01 00 01 01 if done 0x01 00 01 00 if not
Remark	Max Pin length=16

Command	Get Pin Code
Header	0x07
Length	00 00
Payload	N digits PIN. (Défaut "0000")
Response	0x07 {PinCode size} {Pincode}
Remark	

Command	Set Name
Header	0x02 (flash only)
Length	xx xx
Payload	(new name size)
Response	New name 0x02 00 01 01 if done 0x02 00 01 00 if not
Remark	(Names up to 248 Bytes)

Command	Get Name
Header	0x08
Length	00 00
Payload	
Response	0x08 {name size} {name}
Remark	Name size: 2 Bytes MSB, LSB Names up to 248 Bytes

Command	Set Mode
Header	0x03 (flash only)
Length	00 01
Payload	0x01 if MASTER, 0x00 if SLAVE
Response	0x03 00 01 01 if done 0x03 00 01 00 if not

Command	Set Mode
Header	0x03 (flash only)
Length	00 02
Payload	0x01 if MASTER, 0x00 if SLAVE, [Role switch]
Response	0x03 00 01 01 if done
	0x03 00 01 00 if not

When in Master, the Module connects to the address specified by Set REMOTE BDA or to the last paired device.

The real MASTER in a Bluetooth piconet is the device which manages the clock used for the frequency hopping. We used to speak about MASTER too for devices which create the connection (that's true if you do not switch the clock role)

A device with a slave BT clock role is unable to synchronize more than one master clock. If more than one SmartModule needs to connect to the same other device (PC, Access Point…) you will need to switch the clock role to allow the slave to be connected to more than one master. Note that most of the BT access point already generates the BT clock role switch when a master device creates a connection.

Command	Get Mode
Header	0x04
Length	00 00
Payload	
Response	0x04 00 02 {Mode (1byte) | Switch role (1byte)}
Remark	0x01 if MASTER, 0x00 if SLAVE
	0x01 if want automatic switch role, 0x00 otherwise

Command	Set Remote BDA (Used by Master Mode of the SM)
Header	0x05
Length	00 06
Payload	BDA(ex:0x00,0x02,0xC3,0x21, 0xDE,0xFA)
Response	0x05 00 01 01 if done
	0x05 00 01 00 if not
Remark	If The SM is set to Master (using Set MODE command), the SM use this Address to connect to.

Command	Get Remote BDA
Header	0x06
Length	00 00
Payload	
Response	0x06 00 06 {6 bytes of BDA}
Remark	

Command	Get Bluetooth Version
Header	0x76
Length	00 00
Payload	
Response	0x76, x, x, {version string }

Command	Restore Factory Settings
Header	
Length	
Payload	('R', 's', 't')
Response	

Command	Get inquiry scan timeout
Header	0x27
Length	00 00
Payload	
Response	0x27 00 04 [Inquiry Interval (MSB)	Inquiry Interval (LSB)	Inquiry Window (MSB)	Inquiry Window (LSB)]
Remark	Inquiry Interval and Inquiry Window are in number of Bluetooth slots) (1 slot = 0.625 ms)

Command	Set inquiry scan timeout
Header	0x26
Length	00 04
Payload	Inquiry Interval (MSB)	Inquiry Interval (LSB)	Inquiry Window (MSB)	Inquiry Window (LSB)] (default 0xC80, 0x18)
Response	0x26 00 01 01 if done 0x26 00 01 00 if not
Remarks	Inquiry Scan TimeOuts are used by the Module to answer to Inquiries. So, if you set both values to 0, the Module will not be discoverable.

Command	Set page scan timeout
Header	0x24
Length	00 04
Payload	[Page Interval (MSB) | Page Interval (LSB) | Page Window (MSB) | Page Window (LSB)] (default 0x320, 0xb0)
Response	0x24 00 01 01 if done 0x24 00 01 00 if not
Remark	Page Scan TimeOuts are used by the Module to answer to Connect Inquiries. So, if you set both values to 0, the Module will not be Connectable.

Command	Get page scan timeout
Header	0x25
Length	00 00
Payload	
Response	0x25 00 04 [Page Interval (MSB) | Page Interval (LSB) | Page Window (MSB) | Page Window (LSB)]
Remark	Page Scan Interval and Page Scan Window are in number of Bluetooth slots) (1 slot = 0.625 ms)

Typical values are:

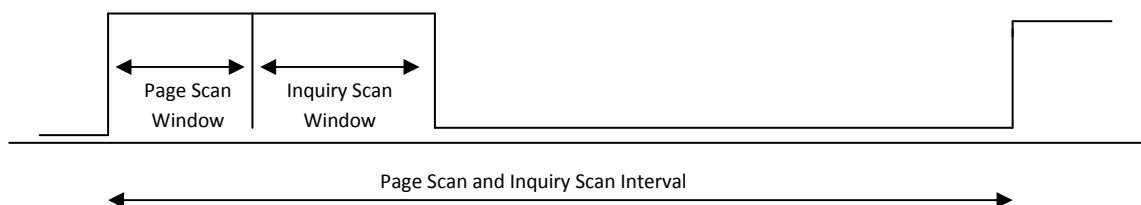Full power:

Inquiry Interval = 0x400
Inquiry Window = 0x200
Page Scan Interval = 0x400
PageScan Window = 0x200

Low power:

Inquiry Interval = 0x320
Inquiry Window = 0x80
Page Scan Interval = 0x320
PageScan Window = 0x80

Here is how these values change the consumption of the Module:

Page Scan and Inquiry Scan Interval

Command	Set sniff
Header	0x09
Length	00 04
Payload	[MSB of MinSniff interval, LSB of MinSniff interval, MSB of MaxSniff interval, LSB of MaxSnif interval]
Response	0x09 00 01 01 if done 0x09 00 01 00 if not
Remark	

Command	Set sniff (advanced)
Header	0x09
Length	00 08
Payload	[MSB of MinSniff interval, LSB of MinSniff interval, MSB of MaxSniff interval, LSB of MaxSnif interval, Sniff Attempts MSB, Sniff attempts LSB, Sniff timeout MSB, Sniff timeout LSB]
Response	0x09 00 01 01 if done 0x09 00 01 00 if not
Remark	

Command	Get Sniff
Header	0x10
Length	00 00
Payload	
Response	0x10 00 08 [MSB of MinSniff interval, LSB of MinSniff interval, MSB of MaxSniff interval, LSB of MaxSnif interval, Sniff Attempts MSB, Sniff attempts LSB, Sniff timeout MSB, Sniff timeout LSB]
Remark	When setting only MinSniff and MaxSniff values, the default value 0x08 will be used for Sniff attempts and Sniff timeout.

Typical values are:

Full speed (full power)

MinSniff = 0

MaxSniff = 0

Very Low Power (low speed): (sniff of 500ms Only are accepted. If the remote device does not support sniffs of 500ms, no sniff will be used)

MinSniff = 0x0320

MaxSniff = 0x0320

Very Low Power (low speed): (sniff between 250ms to 500ms are accepted. No sniff will be used if the remote device does not support any sniff values in this specified range)

MinSniff = 0x0160

MaxSniff = 0x0320

Low Power (medium speed):

MinSniff = 0x0050

MaxSniff = 0x00F0

Attempt = 0x0008

Timeout = 0x0030

MaxSniff and MinSniff are only used for sniff negotiation between the Smart Module and the other BT device. If both sides allow sniff value MaxSniff, then MaxSniff will be used. If the other side does not accept Sniff values MinSniff to MaxSniff, no sniff will be used.

Values are in number of Bluetooth slots (1 slot = 625µs)

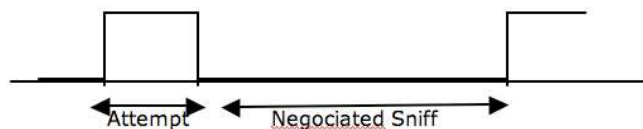Set MinSniff and MaxSniff to 0 to disable Sniff.

MinSniff must be inferior to MaxSniff.

Possible values for MinSniff and MaxSniff are 0x12 to 0xFFFF.

Sniff attempts of 0 is not allowed.

Warning: Setting MaxSniff to 0xFF means a sniff period of 40s! You will have very very low data rate.

Note: This setting takes effect immediately.

For further details on Sniff values, see the Bluetooth spec 1.1, chapter 10.8.2

Command	Get link timeout
Header	0x18
Length	00 00
Payload	
Response	0x18 00 02 [MSB of link Tmo, LSB of link Tmo]
Remark	

Command	Set link timeout
Header	0x19
Length	00 02
Payload	[MSB of link Tmo, LSB of link Tmo]
Response	0x19 00 01 01 if done
Remark	The link Time Out is a multiple of 625µsec (625µs = 1 Bluetooth slot) (default 0x7D00 (=20s))

This Timeout is use by the Link Manager to monitor the Bluetooth Link. If there is no answer from the other device after this timeout, the Link Manager assumes that we are disconnected. By default, this value is set to 20 seconds. You can go down to 1s, but then you can have disconnection even if it's only a temporary perturbation.

This value will take effect at the next connection.

Command	Get Security Mode
Header	0x20
Length	00 00
Payload	
Response	0x20 00 01 01 if secured
	0x20 00 01 00 if non secured
Remark	

Command	Set Security Mode
Header	0x21
Length	00 {size}
Payload	{00 non secured, 01 secured} {PIN CODE
	(default 01)
Response	0x21 00 01 01 if done,
	0x21 00 01 00 if not
Remark	Size=PINCODE size + 1
	For example : 0x21 00 05 00 30 30 30 30 to disable security

Command	Get Bluetooth class device
Header	0x30
Length	00 00
Payload	
Response	0x30 00 04 [Class of device]
Remark	See the Bluetooth specification for more details

Command	Set Bluetooth class device
Header	0x31
Length	00 04
Payload	[Class of Device (4 bytes, MSB->LSB)] (default 0x500)
Response	0x31 00 01 01 if done
	0x31 00 01 00 if not

Typical Bluetooth class of device:

Peripheral	0x000500 (default)
Undefined	0x001F00
Phone	0x502204
Computer	0x120104
PDA	0x100114
Access Point	0x120320

Command	Set Remote rfcomm channel
Header	0x36
Length	00 01
Payload	[channel (1byte)]
Response	0x36 00 01 01 if done 0x36 00 01 00 if not
Remark	

Command	Get Remote rfcomm channel
Header	0x37
Length	00 00
Payload	
Response	0x37 00 01 [channel]
Remark	

If "channel" is not zero, the Module will directly try to connect (if in master mode) to the specified rfcomm channel.
Setting the channel to zero will force the Module to connect (if in master mode) to the first specified Remote Service UUID (by default SPP).
The services in the Module are all set to channel 1.

Command	Set Target Service UUID
Header	0x38
Length	00 02
Payload	[UUID (2 Bytes)] (default 0x1101)
Response	0x38 00 01 01 if done 0x38 00 01 00 if not
Remark	Try to connect to this remote service.

Command	Get Target Service UUID
Header	0x39
Length	00 00

Payload	
Response	0x39 00 02 [UUID]
Remark	Try to connect to this remote service.

Here are some service UUID:

SPP	0x1101
DUN	0x1103
FAX	0x1102

You can get more UUIDs by reading the Bluetooth spec.

Command	Get Encryption Mode
Header	0x40
Length	00 00
Payload	
Response	0x40 00 01 [encryption]
Remark	

Command	Set Encryption Mode
Header	0x41 (flash only)
Length	00 01
Payload	[Encryption (1 byte)]
Response	0x41 00 01 01 if done
	0x41 00 01 00 if not
Remark	Argument is: 0x01 to enable encryption, 0x00 to disable.

Command	Get local Bluetooth Address
Header	0x43
Length	00 00
Payload	
Response	0x43 00 06 {6 Bytes (BD_address MSB, ..., LSB)}
Remark	