

## RoadRunners-Communication-Protocol



# SUMMARY

SUMMARY .....	2
INTRODUCTION .....	3
Protocol of communication.....	4
1. Bidirectionnal packets.....	4
1.1. Generic packet.....	4
1.1.1. Format of that packet.....	4
1.1.2. Explanations.....	4
1.2. Control messages.....	5
1.2.1. Specifics packets.....	5
1.2.2. Acknowledgment packets.....	5
1.2.3. Synchronisation packet.....	5
2. Between the SCANNER and the REMOTE APPLICATION.....	6
3. Between the REMOTE APLICATION and the SCANNER .....	7
3.1. Communication messages.....	7
3.1.1. Generic Code IDs .....	7
3.1.2. Dock Code IDs.....	9
3.1.3. Bluetooth Code IDs .....	9
3.2. Scanner messages.....	10
3.2.1. Generic Code IDs.....	10
3.3. User Interface messages .....	13
3.4. Miscellaneous messages (host to scanner) .....	16
3.5. Barcode messages.....	18
3.5.1. Barcode decoder commands.....	20
ANNEXE 1 : SmartModule Developer Implementation Guide.....	21
ANNEXE 2 : BARCODE DECODER 1D - Frame Configuration for BRR-L Decoder setting via serial connexion.....	25
ANNEXE 3 : BARCODE DECODER 2D - Frame Configuration for BRR-F Decoder setting via serial connexion.....	35

## **INTRODUCTION**

*RoadRunners* is a wireless data capture product.

- Wireless communication based on the Bluetooth protocol
- Barcode reader thanks to a CMOS technology (1D & 2D)

This document is detailing the protocol of communication between the Baracoda scanner *RoadRunners* with its foreign environment through :

- Radio Frequency link (ie : Bluetooth)
- Wired link (ie : RS232 cable)

The protocol of communication does not depend on the type of link.

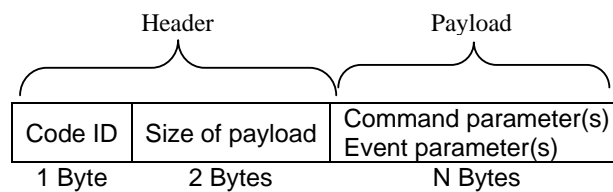
## Protocol of communication

### 1. Bidirectionnal packets

#### 1.1. Generic packet

All frames that are described in the following have to follow the following format of packet. That is available for communication from the scanner to the host and from the host to the scanner.

##### 1.1.1. Format of that packet



##### 1.1.2. Explanations

- Code ID [1 Byte] :
  - o Bits 7:5 is the logical device
  - o Bits 4:1 is the command
  - o Bit 0: when set, the message must be acknowledged ( => “sequence number” is required)

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

- Size of payload [2 Bytes]
  - o This is big-endian format
  - o This is including the “sequence number byte” if this one is required.
- Payload [N Bytes] :
  - o 1 byte for “sequence number” when applicable

## 1.2. Control messages

### 1.2.1. Specifics packets

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

These 2 sequences will be recognized and purged for backward compatibility with other Baracoda products.

### 1.2.2. 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 (see User Interface issues)

### 1.2.3. Synchronisation 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. Between the SCANNER and the REMOTE APPLICATION

<b>Code ID</b>	<b>Description</b>	<b>Payload</b>
0x32–0x33	Barcode data	Barcode string

### 3. Between the REMOTE APLICATION and the SCANNER

#### 3.1. Communication messages

##### 3.1.1. Generic Code IDs

<b>Code ID</b>	0x40-0x41
<b>Description</b>	Get Communication Descriptor
<b>Payload</b>	None
<b>Response</b>	2 bytes: {Wireless link: (Bit 0: CSR BC02)} {Wired link: (Bit 0: Serial Dock)}

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

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

<b>Code ID</b>	0x46-0x47
<b>Description</b>	Get Capture Frame Format
<b>Payload</b>	None
<b>Response</b>	1 byte {0 = Baracoda's frames, 1 = Baracoda's frames + ACK, 2 = Raw}

<b>Code ID</b>	0x48-0x49
<b>Description</b>	Set Capture Frame Format
<b>Payload</b>	1 byte {0 = Baracoda's frames, 1 = Baracoda's frames + ACK, 2 = Raw}
<b>Response</b>	1 byte: {Bit 0: Success}

<b>Code ID</b>	0x4A-0x4B
<b>Description</b>	Get Support Data
<b>Payload</b>	
<b>Response</b>	1Byte {Battery level (%)} 2Bytes {Size of CPU Version string} xBytes {CPU Version} 2Bytes {Size of BT Version} xBytes {BT Version} 6Bytes {BDAddress} 1Byte {BT Secure Mode} 1Byte {Scanner Operating Mode (Real Time/Batch)} 1Byte {Limited Mode} – No data loss mode 1Byte {Sutdown Timer (connected)} 1Byte {Sutdown Timer (disconnected)} 1Byte {Capture Data Format} 4Bytes {Active symbologies mask} – only for BRR-L product 1Byte {Timestamp status} 6Bytes {Timestamp : YY MM DD hh mm ss} 3Bytes {Trigger press count (÷10)} - since firmware v3.00 2Bytes {Charging cycles count} - since firmware v3.00

<b>Code ID</b>	0x4C-0x4D
<b>Description</b>	Get Configuration Checksum (from firmware v3.00)
<b>Payload</b>	None
<b>Response</b>	4 bytes : checksum

<b>Code ID</b>	0x4E-0x4F
<b>Description</b>	Set Configuration Checksum (from firmware v3.00)
<b>Payload</b>	4 bytes : checksum
<b>Response</b>	1 byte : {Bit 0 : Success}

<b>Code ID</b>	0x50-0x51
<b>Description</b>	Lock / unlock Scan engine beam (from firmware v3.00)
<b>Payload</b>	1 byte : {0 = Unlock, 1 = Lock}
<b>Response</b>	1 byte : {Bit 0 : Success}

<b>Code ID</b>	0x52-0x53
<b>Description</b>	Enable / Disable configuration in scanning barcodes (from firmware v3.00)
<b>Payload</b>	1 byte : {0 = Disable barcode programming fonctionnality, 1 = Enable}
<b>Response</b>	1 byte : {Bit 0 : Success}

<b>Code ID</b>	0x54-0x55
<b>Description</b>	Get / Set Dock use
<b>Payload</b>	Get : none Set : 1 byte : {0 = wireless only ; 1 use dock communication}
<b>Response</b>	Get : 1 byte : {0 = wireless only ; 1 use dock communication} Set : 1 byte : {Success?}

<b>Code ID</b>	0x56-0x57
<b>Description</b>	n.c
<b>Payload</b>	n.c
<b>Response</b>	n.c

<b>Code ID</b>	0x5C-0x5D
<b>Description</b>	Dock Commands
<b>Payload</b>	{Code ID} "Parameters"
<b>Response</b>	If the device responds: {Code ID} "Response" Else: {0}

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

### 3.1.2. Dock Code IDs

<b>Code ID</b>	0x28
<b>Description</b>	Set UART parameters
<b>Payload</b>	4 bytes: {(Baudrate / 300) [15:8]} {(Baudrate / 300) [7:0]} The baudrate must be between 300bps and 921600bps {Stop bits: 0 = 1 stop bit, 1 = 2 stop bits} {Parity: 0 = None, 1 = Even, 2 = Odd}
<b>Response</b>	1 byte: {Bit 0: Success}

<b>Code ID</b>	0x29
<b>Description</b>	Set UART parameters
<b>Payload</b>	None
<b>Response</b>	4 bytes: {(Baudrate / 300) [15:8]} {(Baudrate / 300) [7:0]} {Stop bits: 0 = 1 stop bit, 1 = 2 stop bits} {Parity: 0 = None, 1 = Even, 2 = Odd}

### 3.1.3. Bluetooth Code IDs

For the Bluetooth Code IDs, see the 'SmartModule Developer Implementation Guide' (Annex §1)

### 3.2. Scanner messages

#### 3.2.1. Generic Code IDs

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

<b>Code ID</b>	0x62-0x63
<b>Description</b>	Get Operating Mode
<b>Payload</b>	None
<b>Response</b>	1 byte: {Bit 0 = 1: real time, Bit 0 = 1: batch} {(Bit 7: limited – ie : no data loss mode)}

<b>Code ID</b>	0x64-0x65
<b>Description</b>	Set Operating Mode
<b>Payload</b>	1 byte: {Bit 0 = 0: real time, Bit 0 = 1: batch} If real time mode is set : {(Bit 7: limited – ie : no data loss mode)(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"
<b>Response</b>	1 byte: {Bit 0: Success}

<b>Code ID</b>	0x66-0x67
<b>Description</b>	Get Shutdown Timers
<b>Payload</b>	None
<b>Response</b>	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}

<b>Code ID</b>	0x68-0x69
<b>Description</b>	Set Shutdown Timers
<b>Payload</b>	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}
<b>Response</b>	1 byte: {Bit 0: Success}

<b>Code ID</b>	0x6A-0x6B
<b>Description</b>	Get RTC Time
<b>Payload</b>	None
<b>Response</b>	6 bytes: {YY}{MM}{DD}{HH}{MM}{SS}

<b>Code ID</b>	0x6C-0x6D
<b>Description</b>	Set RTC Time
<b>Payload</b>	6 bytes: {YY}{MM}{DD}{HH}{MM}{SS}
<b>Response</b>	1 byte: {Bit 0: Success}

<b>Code ID</b>	0x6E-0x6F
<b>Description</b>	Get Sending delay ( delay between beam ON and decoding process - only available for BRR-L product)
<b>Payload</b>	None
<b>Response</b>	1 byte: {sending delay (seconds)}

<b>Code ID</b>	0x70-0x71
<b>Description</b>	Set Sending delay – between beam ON and decoding process (only available for BRR-L product)
<b>Payload</b>	1 byte: {sending delay (seconds)}
<b>Response</b>	1 byte: {Bit 0: Success}

<b>Code ID</b>	0x72-0x73
<b>Description</b>	n.c
<b>Payload</b>	n.c
<b>Response</b>	n.c

<b>Code ID</b>	0x74-0x75
<b>Description</b>	Restore defaults settings
<b>Payload</b>	None
<b>Response</b>	1 byte: {Bit 0: Success} NOTE : Flash memory is also erased

<b>Code ID</b>	0x76-0x77
<b>Description</b>	Get Product Version
<b>Payload</b>	None
<b>Response</b>	16 bytes : «Baracoda Abeille»

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

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

<b>Code ID</b>	0x7A-0x7B
<b>Description</b>	n.c
<b>Payload</b>	n.c
<b>Response</b>	n.c

<b>Code ID</b>	0x7C-0x7D
<b>Description</b>	n.c
<b>Payload</b>	n.c
<b>Response</b>	n.c

<b>Code ID</b>	0x7E-0x7F
<b>Description</b>	n.c
<b>Payload</b>	n.c
<b>Response</b>	n.c

### 3.3. User Interface messages

<b>Code ID</b>	0x80-0x81
<b>Description</b>	Get MMI Descriptor (User Interface)
<b>Payload</b>	None
<b>Response</b>	2 bytes: {(Bit 6: NA) (Bit 5: Red LED 1 – left one) (Bit 4: Green LED 1 – left one) (Bit 2: NA) (Bit 1: Red LED 0 – right one) (Bit 0: Green LED 0 – right one)} {(Bit 0: Buzzer)}

<b>Code ID</b>	0x82-0x83
<b>Description</b>	Get MMI Mode (User interface)
<b>Payload</b>	None
<b>Response</b>	1 byte: {(Bit 1: Buzzer Enabled) (Bit 0: LEDs enabled)}

<b>Code ID</b>	0x84-0x85
<b>Description</b>	Set MMI Mode (User interface)
<b>Payload</b>	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)}
<b>Response</b>	1 byte: {(Bit 0: Success)}

<b>Code ID</b>	0x86-0x87
<b>Description</b>	Get MMI Signal (User interface)
<b>Payload</b>	1 byte: {Signal number, 0 - 3}
<b>Response</b>	(1 + 3n) bytes: {Number of steps, 0 - 4} For each step: {(Bit 6: NA) (Bit 5: Red LED 1) (Bit 4: Green LED 1) (Bit 2: NA) (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} <u>Nota</u> : LED 0 = right one LED 1 = left one

<b>Code ID</b>	0x88-0x89
<b>Description</b>	Set MMI Signal
<b>Payload</b>	(2 + 3n) bytes {Signal number, 0 - 3} {Number of steps, 0 - 4} For each step: {(Bit 6: NA) (Bit 5: Red LED 1) (Bit 4: Green LED 1) (Bit 2: NA) (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} <u>Nota</u> : LED 0 = right one LED 1 = left one
<b>Response</b>	1 byte: {(Bit 0: Success)}

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

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

The list of MMI signals is:

0. IHM\_SIGNAL\_CAPTURE\_READ
1. IHM\_SIGNAL\_CAPTURE\_ACK
2. IHM\_SIGNAL\_CAPTURE\_NAK
3. IHM\_SIGNAL\_CAPTURE\_LOST

<b>Code ID</b>	0x8E-0x8F
<b>Description</b>	Invert Leds
<b>Payload</b>	1 byte: {0 = default, 1 = inverted} <u>Note</u> : Default : LED 0 is on the right, LED 1 on the left Inverted : LED 1 is on the right, LED 0 on the left
<b>Response</b>	1 byte: {(Bit 0: Success)}

<b>Code ID</b>	0x90-0x91
<b>Description</b>	Get "invert Leds" status
<b>Payload</b>	
<b>Response</b>	1 byte: {0 = default, 1 = inverted} <u>Note</u> : Default : LED 0 is on the right, LED 1 on the left Inverted : LED 1 is on the right, LED 0 on the left

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

<b>Code ID</b>	0x94-0x95
<b>Description</b>	Set Beeps mode
<b>Payload</b>	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)}}}
<b>Response</b>	1 byte: {(Bit 0: Success)}

### 3.4. Miscellaneous messages (host to scanner)

<b>Code ID</b>	0xC0-0xC1
<b>Description</b>	n.c
<b>Payload</b>	n.c
<b>Response</b>	n.c

<b>Code ID</b>	0xC2-0xC3
<b>Description</b>	Get / Set PostPrefix
<b>Payload</b>	Get : None Set : 1-33 bytes {PostPrefix length} "PostPrefix string"
<b>Response</b>	Get : 1-33 bytes {PostPrefix length} "PostPrefix string" Set : 1 byte {Success ?}

<b>Code ID</b>	0xC4-0xC5
<b>Description</b>	Get / Set PostSuffix
<b>Payload</b>	Get : None Set : 1-33 bytes {PostSuffix length} "PostSuffix string"
<b>Response</b>	Get : 1-33 bytes {PostSuffix length} "PostSuffix string" Set : 1 byte {Success ?}

<b>Code ID</b>	0xC6-0xC7
<b>Description</b>	Get / Set Data Format 2
<b>Payload</b>	Get : None Set : 1 byte {(Bit 5 = 0: PostPrefix config select, =1: PostPrefix config deselect) (Bit 4 = 0: PostSuffix config select, =1: PostSuffix config deselect) (Bit 1: PostPrefix) (Bit 0: PostSuffix)}
<b>Response</b>	Get : 1 byte {(Bit 1: PostPrefix) (Bit 0 : PostSuffix)} "PostSuffix string" Set : 1 byte {(Bit 0:Success)}

<b>Code ID</b>	0xC8-0xC9
<b>Description</b>	Get / Set Barcode length
<b>Payload</b>	Get : None Set : 2 bytes {authorized barcode length}
<b>Response</b>	Get : 2 bytes {authorized barcode length} Set : 1 byte {(Bit 0:Success)}

<b>Code ID</b>	0xCA-0xCB
<b>Description</b>	Get & erase stored barcodes with 'no data loss mode'
<b>Payload</b>	None
<b>Response</b>	1 byte {(Bit 0:Success)}

### 3.5. Barcode messages

<b>Code ID</b>	0xE0-0xE1
<b>Description</b>	Get Barcode Version
<b>Payload</b>	None
<b>Response</b>	“Barcode Version String” or {0} if not applicable

<b>Code ID</b>	0xE2-0xE3
<b>Description</b>	Get Mode
<b>Payload</b>	None
<b>Response</b>	1 byte: {0 = trigger, 1 = autoscan, 2 = disabled, 3 = true autoscan, 4 = aiming trigger scan}

<b>Code ID</b>	0xE4-0xE5
<b>Description</b>	Set Mode
<b>Payload</b>	1 byte if In trigger, autoscan, disabled or true autoscan mode. Or 2 bytes if In aiming trigger scan mode {0 = trigger, 1 = autoscan, 2 = disabled, 3 = true autoscan, 4 = aiming trigger scan} {aiming trigger scan mode timeout value in second}
<b>Response</b>	1 byte: {{Bit 0: Success}}

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

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

<b>Code ID</b>	0xEA-0xEB
<b>Description</b>	Get Prefix
<b>Payload</b>	None
<b>Response</b>	1-33 bytes: {Prefix length} “Prefix String”

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

<b>Code ID</b>	0xEE-0xEF
<b>Description</b>	Get Suffix
<b>Payload</b>	None
<b>Response</b>	1-33 bytes: {Suffix length} "Suffix String"

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

<b>Code ID</b>	0xF2-0xF3
<b>Description</b>	Get and erase stored barcodes and Reset USB Plug and Scan Dongle
<b>Payload</b>	None
<b>Response</b>	1 byte: {(Bit 0:Success)}

**Note :** This command is only available using an USB Plug & Scan Dongle (Baracoda's product)

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

**Note :** Baracoda invites developers to re-synchronize the Internal clock regularly (frequency of this re-synchronization depends on the application)

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

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

<b>Code ID</b>	0xFA-0xFB
<b>Description</b>	Get and erase stored barcodes
<b>Payload</b>	None
<b>Response</b>	1 byte: {{Bit 0:Success}}

<b>Code ID</b>	0xFC-0xFD
<b>Description</b>	Erase stored barcodes
<b>Payload</b>	None
<b>Response</b>	1 byte: {{Bit 0:Success}}

<b>Code ID</b>	0xFE-0xFF
<b>Description</b>	Barcode decoder Commands
<b>Payload</b>	BRR-L : {header} {type} {size} "Parameters" BRR-F : Tag SubTag {data}{,SubTag {Data}};Tag SubTag{Data}[...] Storage
<b>Response</b>	If the device responds: BRR-L : {Code ID}{length}{header} {type} {response size} "Response"  BRR-F : "payload"ACK storage  Else: {0}

### 3.5.1. Barcode decoder commands

For the RoadRunners BRR-L :

Barcode decoder Commands, see annex 2 ; "BARCODE DECODER 1D - Frame Configuration for BRR-L Decoder setting via serial connexion"

For the RoadRunner BRR-F :

Barcode decoder Commands, see the "Frame Configuration for BRR-F Decoder setting via serial connexion" (see § annex 3)

## ANNEXE 1 : SMARTMODULE DEVELOPER IMPLEMENTATION GUIDE

The configuration frames are as follow:

Header: 1 Byte

Length: 2 Bytes (MSB, LSB)

Payload: 0 to 65535 Bytes.

### Set PinCode (0x01) (flash only)

Command	Header	Length	Payload	Answer
Set PinCode	0x01	00 00	N digits PIN. (Défaut "0000")	0x01 00 01 01 if done 0x01 00 01 00 if not

Max Pin length=16

### Get PinCode (0x07)

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

### Set Name (0x02) (flash only)

Command	Header	Length	Payload	Answer
Set Name	0x02	xx xx (new name size)	New name	0x02 00 01 01 if done 0x02 00 01 00 if not

(Names up to 248 Bytes)

### Get Name (0x08)

Command	Header	Length	Payload	Answer
Get Name	0x08	00 00		0x08 {name size} {name}

(Name size: 2 Bytes MSB, LSB Names up to 248 Bytes)

### Set Mode (0x03) (flash only)

Command	Header	Length	Payload	Answer
Set mode	0x03	00 01	0x01 if MASTER, 0x00 if SLAVE	0x03 00 01 01 if done 0x03 00 01 00 if not

Command	Header	Length	Payload	Answer
Set mode	0x03	00 02	[0x01 if MASTER, 0x00 if SLAVE], [Role switch]	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.

**Remark:** Make sure that gpio5 (gpio disconnect) is set to 0 to connect. If this gpio is set to 1, the module will not try to connect if in master mode.

You can set the size to 2. In this case, the second argument tells the module for an automatic role switch. This is an optional argument.

**Get Mode (0x04)**

Command	Header	Length	Payload	Answer
Get mode	0x04	00 00	-	0x04 00 02 {Mode (1byte)   Switch role (1byte)}

0x01 if MASTER, 0x00 if SLAVE  
0x01 if want automatic switch role, 0x00 otherwise

**Set Remote BDA (0x05)**

(Used by Master Mode of the SM)

Command	Header	Length	Payload	Answer
Set Remote BDA	0x05	00 06	BDA(ex:0x00,0x02,0xC3,0x21, 0xDE,0xFA)	0x05 00 01 01 if done 0x05 00 01 00 if not

If The SM is set to Master (using Set MODE command), the SM use this Address to connect to.

**Get Remote BDA (0x06)**

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

**Get Firmware Version (0x76)**

Command	Header	Length	Payload	Answer
Get Version	0x76	00 00	-	0x76, 0, 18, {string of 18 characters}

Answer:	Header	Length	Payload
Version	0x76	Version length (2Bytes)	Version (string)

**Restore Factory Settings ('R', 's', 't')**

(Bluetooth only)

Command	Header	Length	Payload	Answer
Restore Factory Settings			'R', 's', 't'	-

**Set Sniff (0x09)**

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

Command	Header	Length	Payload	Answer
Set sniff	0x09	00 08	[MSB of MinSniff interval, LSB of MinSniff interval, MSB of MaxSniff interval, LSB of MaxSniff interval, Sniff Attempts MSB, Sniff attempts LSB, Sniff timeout MSB, Sniff timeout LSB]	0x09 00 01 01 if done 0x09 00 01 00 if not

**Get Sniff (0x10)**

Command	Header	Length	Payload	Answer
Get sniff	0x10	00 08		0x10 00 04 [MSB of MinSniff interval, LSB of MinSniff interval, MSB of MaxSniff interval, LSB of MaxSniff interval, Sniff Attempts MSB, Sniff attempts LSB, Sniff timeout MSB, Sniff timeout LSB]

When setting only MinSniff and MaxSniff values, the default value 0x08 will be used for Sniff attempts and Sniff timeout.

Typically values:

Full speed (full power)

MinSniff = 0

MaxSniff = 0

Low Power: (sniff of 500ms Only) (low speed)

MinSniff = 0x0320

MaxSniff = 0x0320

Low Power: (sniff between 250ms to 500ms are accepted) (low speed)

MinSniff = 0x0160

MaxSniff = 0x0320

MaxSniff and MinSniff are only used for sniff negotiation. 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.**

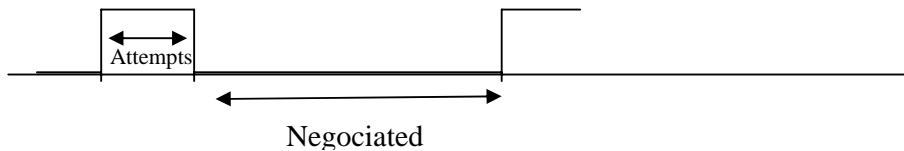
MinSniff must be inferior to MaxSniff.

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

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. Also, setting MinSniff to 0x12 means that you will have 0x12 Bluetooth slots to transmit data.

Note: This setting takes effect immediately.



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

**Set Security Mode (0x21)**

Command	Header	Length	Payload	Answer
Set secure mode	0x21	00 {size}	{00 non secured, 01 secured} {PIN CODE}	0x21 00 01 01 if done, 0x21 00 01 00 if not

Size=PINCODE size + 1

For example : 0x21 00 05 00 30 30 30 30 to disable security

**Get Secure Mode (0x20)**

Command	Header	Length	Payload	Answer
Get secure mode	0x20	00 00	-	0x20 00 01 01 if secured 0x20 00 01 00 if non secured

**Set Encryption Mode (0x41) ( flash only)**

Command	Header	Length	Payload	Answer
Set Encryption	0x41	00 01	[Encryption]	0x41 00 01 01 if done

## RoadRunners – Communication Protocol

mode			(1 byte)]	0x41 00 01 00 if not
------	--	--	-----------	----------------------

Argument is : 0x01 to enable encryption, 0x00 to disable

### **Get Encryption Mode (0x40)**

Command	Header	Length	Payload	Answer
Get Encryption mode	0x40	00 00	-	0x40 00 01 [encryption]

### **Get Local Bluetooth Address (0x43)**

Command	Header	Length	Payload	Answer
Get local BT address	0x43	00 00	-	0x43 00 06 {6 Bytes (BD_address MSB, ..., LSB)}

### **Get Maximum transmit power (0x3a)**

Command	Header	Length	Payload	Answer
Get Transmit power BT address	0x3a	00 00	-	0x3a 00 04 [default power (2 Bytes), Max power (2 Bytes)]

### **Set Maximum transmit power (0x3b)**

Command	Header	Length	Payload	Answer
Set Transmit power BT address	0x3b	00 04	[default power (2 Bytes), Max power (2 Bytes)]	0x3a 00 01 01 if done 0x3a 00 01 00 if not

Default and max power are in dBm unit

## ANNEXE 2 : BARCODE DECODER 1D - FRAME CONFIGURATION FOR BRR-L DECODER SETTING VIA SERIAL CONNEXION

### 1/ frame format

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

### 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.

### 3/ Type

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

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

### 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),

## **5/ Commands**

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

There are three types of commands:

### **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	FREE
	6	FREE
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). 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.

#### **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
MSB	6	FREE
	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
MSB	6	FREE
	7	FREE

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

	Bit	Option
LSB	0	Number System transmitted (UPC A)
	1	-
	2	-
	3	-
	4	UPC-A, transmitted as EAN 13
	5	FREE
MSB	6	-
	7	FREE

**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
MSB	6	FREE
	7	FREE

**CODABAR (Header "F")**

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

**INTERLEAVED 2 OF 5 (Header "G")**

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

**STANDARD 2 OF 5 (Header "H")**

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

**MATRIX 2 OF 5 (Header "I")**

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

**CODE 11 (Header "J")**

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

**MSI (Header "K")**

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

**EAN 8 (Header "M")**

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

### **5.2/ Usual Commands (Type "D")**

The method described above 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.

<b>COMMANDS TABLE</b>		
<b>Ascii CODE</b>	<b>COMMAND</b>	<b>RESPONDING HEADERS</b>
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	-	
S	-	
T	-	
U	-	
V	-	
W	-	

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	-	
c	-	
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

(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.

### **5.3/ 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**

CODE ASCII	DESCRIPTION	PARAMÈTRE(S)
A	DEFINE AND ENABLE MIN. LENGTH	[MIN LENGTH]
B	DEFINE AND ENABLE MAX. LENGTH	[MAX LENGTH]
C	DEFINE AND ENABLE LENGTH FRAME	[MIN LENGTH][MAX LENGTH]
D	DEFINE VOTING VALUE (*)	[VOTING]
E	GET VOTING VALUE	-
F	DEFINE GS REPLACEMENT CHARACTER	[CHAR]
G	GET GS REPLACEMENT CHARACTER	-
H	FREE	FREE
...	...	...

(\*): this command will only be 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.

## 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..

### 6.1 Get Config

#### **Example 1.1 – Getting configuration of Code 39 symbology.**

Frame to be sent to the decoder:

<b>‘E’</b>	<b>‘A’</b>	<b>0</b>	<b>-</b>
Header	Type	Size	Command

The decoder answers the following frame:

<b>‘E’</b>	<b>‘A’</b>	<b>4</b>	<b>\$01 , \$05, \$06, \$20</b>
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:

1<sup>st</sup> byte : **\$01**

Bit	Option
<b>1</b>	Symbology Enabled
<b>0</b>	Min. length Disabled
<b>0</b>	Checksum calculation Disabled
<b>0</b>	Checksum transmission Disabled
<b>0</b>	Max. length Disabled
<b>0</b>	-
<b>0</b>	-
<b>0</b>	-

2<sup>nd</sup> byte : **\$05**

Bit	Option
<b>1</b>	start-stop transmission Enabled
<b>0</b>	Full ACSII Mode Disabled
<b>1</b>	"*" as start-stop character Enabled
<b>0</b>	"\$" as start-stop character Disabled
<b>0</b>	-
<b>0</b>	-
<b>0</b>	-
<b>0</b>	-

3<sup>rd</sup> byte : Min length = 6

4<sup>th</sup> byte Max length = \$20 = 32

## 6.2 Set Config

### Example 2.1 – Setting a configuration of Codabar symbology.

Frame to be sent to the decoder:

<b>'F'</b>	<b>'B'</b>	<b>2</b>	<b>\$03 , \$01</b>
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:

1<sup>st</sup> byte : **\$03**

Bit	Option
1	0
1	1
0	2
0	3
0	4
0	5
0	6
0	7

2<sup>nd</sup> byte : **\$01**

Bit	Option
1	0
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:

<b>'A'</b>	<b>'B'</b>	<b>1</b>	<b>\$13</b>
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:

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

**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:

<b>'A'</b>	<b>'D'</b>	<b>2</b>	<b>'C', 'I'</b>
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).

### **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:

<b>‘E’</b>	<b>‘E’</b>	<b>3</b>	<b>‘C’ , 2 , 40</b>
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.

## ANNEXE 3 : BARCODE DECODER 2D - FRAME CONFIGURATION FOR BRR-F DECODER SETTING VIA SERIAL CONNEXION

### 1/ Command syntax

Tag	SubTag {Data}	[,SubTag {Data}]	[;Tag SubTag {Data}]	[...]	Storage
-----	---------------	------------------	----------------------	-------	---------

[option] : An optional part of a command.

{Data} : Alternatives in a command.

Tag : A 3 character case-insensitive field that identifies the desired menu command group. For example, all Code39 configuration settings are identified with a Tag of **C39**.

SubTag : A 3 character case-insensitive field that identifies the desired menu command within the tag group. For example, the SubTag for the code39 Start/Stop character transmission is **SSX**.

Data : The new value for a menu setting, identified by the Tag and Sub-Tag.

Suffix : A period (.) performs the command's operation on the device's non-volatile menu configuration table. Use the non-volatile table only for semi-permanent changes you want saved through a power cycle.

## 2/ symbology configuration commands

Here is the list of the configurations available for each symbology:

TA G	Symbology	SubTag	Setting (* indicates default)	Command (# indicates a numeric entry)
CBR	Codabar	DFT	Default all Codabar settings	CBRDFT.
		ENA	Off	CBRENA0.
			*On	CBRENA1.
		SSX	*S/S don't transmit	CBRSSX0.
			S/S transmit	CBRSSX1.
		CK2	* no check car.	CBRCK20.
			Check car : validate but don't transmit	CBRCK21.
			Check car : validate and transmit	CBRCK22.
		CCT	Concatenation Off	CBRCCT0.
			*Concatenation On	CBRCCT1.
Concatenation Require	CBRCCT2.			
MIN	Min length (2-60) *4	CBRMIN##.		
MAX	Max length (2-60) *60	CBRMAX##.		
C39	Code39	DFT	Default all Code39 settings	C39DFT.
		ENA	Off	C39ENA0.
			*On	C39ENA1.
		SSX	*S/S don't transmit	C39SSX0.
			S/S transmit	C39SSX1.
		CK2	* no check car.	C39CK20.
			Check car : validate but don't transmit	C39CK21.
			Check car : validate and transmit	C39CK22.
		MIN	Min length (0-48) *0	C39MIN##.
		MAX	Max length (0-48) *48	C39MAX##.
		APP	*Append Off	C39APP0.
			Append On	C39APP1.
		B32	*Code32 Pharmaceutical (PARAF) Off	C39B320.
			*Code32 Pharmaceutical (PARAF) On	C39B321.
ASC	*Code39 full ASCii Off	C39ASCO.		
	Code39 full ASCii On	C39ASC1.		
DCP	Code39 Code Page	C39DCP.		
I25	Interleaved 2 of 5	DFT	Default all interleaved 2 of 5 settings	I25DFT.
		ENA	Off	I25ENA0.
			*On	I25ENA1.
		CK2	* no check car.	I25CK20.
			Check car : validate but don't transmit	I25CK21.
			Check car : validate and transmit	I25CK22.
		MIN	Min length (2-80) *4	I25MIN##.
MAX	Max length (2-80) *80	I25MAX##.		
C93	Code 93	DFT	Default all Code93 settings	C93DFT.
		ENA	Off	C93ENA0.
			*On	C93ENA1.
		MIN	Min length (0-80) *0	C93MIN##.
		MAX	Max length (0-80) *80	C93MAX##.
DCP	Code93 Code Page	C93DCP.		
R25	Code 2 of 5	DFT	Default all Code 2 of 5 settings	R25DFT.
		ENA	*Off	R25ENA0.
			On	R25ENA1.
		MIN	Min length (1-48) *4	R25MIN##.
MAX	Max length (1-48) *48	R25MAX##.		
A25	IATA Code 2 of 5	DFT	Default all IATA Code 2 of 5 settings	A25DFT.
		ENA	*Off	A25ENA0.
			On	A25ENA1.
		MIN	Min length (1-48) *4	A25MIN#.#
MAX	Max length (1-48) *48	A25MAX##.		

X25	Matrix 2 of 5	DFT	Default all Code11 settings	X25DFT.
		ENA	*Off	X25ENA0.
			On	X25ENA1.
		MIN	Min length (1-80) *4	X25MIN##.
MAX	Max length (1-80) *80	X25MAX##.		
C11	Code 11	DFT	Default all Matrix 2 of 5 settings	C11DFT.
		ENA	*Off	C11ENA0.
			On	C11ENA1.
		CK2	1 check digit	C11CK20.
			*2 check digits	C11CK21.
		MIN	Min length (1-80) *4	C11MIN##.
MAX	Max length (1-80) *80	C11MAX##.		
128	Code 128	DFT	Default all Code128 settings	128DFT.
		ENA	Off	128ENA0.
			*On	128ENA1.
		MIN	Min length (0-80) *0	128MIN##.
		MAX	Max length (0-80) *80	128MAX##.
DCP	Code128 Code Page	128DCP.		
TEL	Telepen	DFT	Default all Telepen settings	TELDFT.
		ENA	*Off	TELENA0.
			On	TELENA1.
		OLD	*AIM Telepen Output	TELOLD0.
			Original Telepen Output	TELOLD1.
		MIN	Min length (1-60) *1	TELMIN##.
MAX	Max length (1-60) *60	TELMAX##.		
UPA	UPC-A	DFT	Default all UPC-A settings	UPADFT.
		ENA	Off	UPAENA0.
			*On	UPAENA1.
		CKX	UPC-A check digit Off	UPACKX0.
			*UPC-A check digit On	UPACKX1.
		NSX	UPC-A number system Off	UPANSX0.
			*UPC-A number system On	UPANSX1.
		AD2	*UPC-A 2 Digit Addenda Off	UPAAD20.
			UPC-A 2 Digit Addenda On	UPAAD21.
		AD5	*UPC-A 5 Digit Addenda Off	UPAAD50.
			UPC-A 5 Digit Addenda On	UPAAD51.
		ARQ	*UPC-A Addenda Not required	UPAARQ0.
			UPC-A Addenda required	UPAARQ1.
ADS	UPC-A Addenda separator Off	UPAADS0.		
	*UPC-A Addenda separator On	UPAADS1.		
UPE	UPC-E	DFT	Default all UPC-E settings	UPEDFT.
		EN0	UPC-E0 Off	UPEEN00.
			*UPC-E0 On	UPEEN01.
		EXP	*UPC-E0 expand Off	UPEEXP0.
			UPC-E0 expand On	UPEEXP1.
		ARQ	*UPC-E0 Addenda Not required	UPEARQ0.
			UPC-E0 Addenda required	UPEARQ1.
		ADS	UPC-E0 Addenda separator Off	UPEADS0.
			*UPC-E0 Addenda separator On	UPEADS1.
		CKX	UPC-A check digit Off	UPECKX0.
			*UPC-A check digit On	UPECKX1.
		NSX	UPC-A number system Off	UPENSX0.
			*UPC-A number system On	UPENSX1.
		AD2	*UPC-E0 2 Digit Addenda Off	UPEAD20.
			UPC-E0 2 Digit Addenda On	UPEAD21.
		AD5	*UPC-E0 5 Digit Addenda Off	UPEAD50.
			UPC-E0 5 Digit Addenda On	UPEAD51.
		EN1	*UPC-E1 Off	UPEEN10.
UPC-E1 On	UPEEN11.			

E13	EAN 13	DFT	Default all EAN 13 settings	E13DFT.
		ENA	Off	E13ENA0.
			*On	E13ENA1.
		CKX	EAN 13 check digit Off	E13CKX0.
			* EAN 13 check digit On	E13CKX1.
		ISB	*ISBN translate Off	E13NSX0.
			ISBN translate On	E13NSX1.
		AD2	* EAN 13 2 Digit Addenda Off	E13AD20.
			EAN 13 2 Digit Addenda On	E13AD21.
		AD5	*EAN 13 5 Digit Addenda Off	E13AD50.
EAN 13 5 Digit Addenda On	E13AD51.			
ARQ	*EAN 13 Addenda Not required	E13ARQ0.		
	EAN 13 Addenda required	E13ARQ1.		
ADS	EAN 13 Addenda separator Off	E13ADS0.		
	*EAN 13 Addenda separator On	E13ADS1.		
EA8	EAN 8	DFT	Default all EAN 8 settings	EA8DFT.
		ENA	Off	EA8ENA0.
			*On	EA8ENA1.
		CKX	EAN 8 check digit Off	EA8CKX0.
			* EAN 8 check digit On	EA8CKX1.
		ADS	EAN 8 Addenda separator Off	EA8ADS0.
			*EAN 8 Addenda separator On	EA8ADS1.
		AD2	* EAN 8 2 Digit Addenda Off	EA8AD20.
			EAN 8 2 Digit Addenda On	EA8AD21.
		AD5	*EAN 8 5 Digit Addenda Off	EA8AD50.
EAN 8 5 Digit Addenda On	EA8AD51.			
ARQ	*EAN 8 Addenda Not required	EA8ARQ0.		
	EAN 8 Addenda required	EA8ARQ1.		
MSI	MSI	DFT	Default all MSI settings	MSIDFT.
		ENA	*Off	MSIENA0.
			On	MSIENA1.
		CHK	*Check car : validate type 10 but don't transmit	MSICHK0.
			Check car : validate type 10 and transmit	MSICHK1.
		MIN	Min length (4-48) *4	MSIMIN##.
MAX	Max length (4-48) *48	MSIMAX##.		
PLS	Plessey Code	DFT	Default all Plessey settings	PLSDFT.
		ENA	*Off	PLSENA0.
			On	PLSENA1.
		MIN	Min length (4-48) *4	PLSMIN##.
MAX	Max length (4-48) *48	PLSMAX##.		
RSS	RSS 14	DFT	Default all RSS 14 settings	RSSDFT.
		ENA	Off *On	RSSENA0. RSSENA1.
RSL	RSS Limited	DFT	Default all RSS Limited settings	RSLDFT.
		ENA	Off *On	RSLENA0. RSLENA1.
RSE	RSS Expanded	DFT	Default all RSS Expanded settings	RSEDFT.
		ENA	Off *On	RSEENA0. RSEENA1.
			MIN	Min length (0-80) *4
		MAX	Max length (0-80) *74	RSEMAX##.
POS	PosiCode	DFT	Default all PosiCode settings	POSDFT.
		ENA	Off *On	POSENA0. POSENA1.
			LIM	A and B On
		A and B and limited A On		POSLIM1.
		*A and B and limited B On		POSLIM2.
		MIN	Min length (2-80) *4	POSMIN##.
MAX	Max length (2-80) *48	POSMAX##.		
CBF	Codablock F	DFT	Default all Codablock F settings	CBFDFT.
		ENA	*Off On	CBFENA0. CBFENA1.
			MIN	Min length (1-2048) *1
		MAX	Max length (1-2048) *2048	CBFMAX##.

# RoadRunners – Communication Protocol

16K	Code 16K	DFT	Default all 16K settings	16KDFT.
		ENA	*Off	16KENA0.
			On	16KENA1.
		MIN	Min length (0-160) *1	16KMIN##.
MAX	Max length (0-160) *160	16KMAX##.		
C49	Code 49	DFT	Default all Code49 settings	C49DFT.
		ENA	Off	C49ENA0.
			*On	C49ENA1.
		MIN	Min length (1-81) *1	C49MIN##.
MAX	Max length (1-81) *81	C49MAX##.		
PDF	PDF 417	DFT	Default all PDF417 settings	PDFDFT.
		ENA	Off	PDFENA0.
			*On	PDFENA1.
		MIN	Min length (1-2750) *1	PDFMIN##.
MAX	Max length (1-2750) *81	PDFMAX##.		
MPD	MicroPDF417	DFT	Default all Micro PDF417 settings	MPDDFT.
		ENA	Off	MPDENA0.
			*On	MPDENA1.
		MIN	Min length (1-366) *1	MPDMIN##.
MAX	Max length (1-366) *366	MPDMAX##.		
NET	Postnet	ENA	*Off	NETENA0.
			On	NETENA1.
PLN	Planet Code	ENA	*Off	PLNENA0.
			On	PLNENA1.
BPO	British Post	ENA	*Off	BPOENA0.
			On	BPOENA1.
CAN	Canadian Post	ENA	*Off	CANENA0.
			On	CANENA1.
KIX	Kiw (Netherlands) Post	ENA	*Off	KIXENA0.
			On	KIXENA1.
AUX	Australian Post	ENA	*Off	AUSENA0.
			On	AUSENA1.
JAP	Japanese Post	ENA	*Off	JAPENA0.
			On	JAPENA1.
CPC	China Post	DFT	Default all China post settings	CPCDFT.
		ENA	*Off	CPCENA0.
			On	CPCENA1.
		MIN	Min length (0-80) *4	CPCMIN##.
MAX	Max length (0-80) *80	CPCMAX##.		
KPC	Korea Post	DFT	Default all Korea post settings	KPCDFT.
		ENA	*Off	KPCENA0.
			On	KPCENA1.
		MIN	Min length (2-80) *4	KPCMIN##.
MAX	Max length (2-80) *48	KPCMAX##.		
QRC	QR Code	DFT	Default all QR Code settings	QRCDFT.
		ENA	*Off	QRCENA0.
			On	QRCENA1.
		MIN	Min length (1-3500) *1	QRCMIN##.
MAX	Max length (1-3500) *3500	QRCMAX##.		
IDM	Data Matrix	DFT	Default all Data Matrix settings	IDMDFT.
		ENA	Off	IDMENA0.
			*On	IDMENA1.
		MIN	Min length (1-1500) *1	IDMMIN##.
MAX	Max length (1-1500) *1500	IDMMAX##.		
MAX	MaxiCode	DFT	Default all MaxiCode settings	MAXDFT.
		ENA	*Off	MAXENA0.
			On	MAXENA1.
		MIN	Min length (1-150) *1	MAXMIN##.
MAX	Max length (1-150) *150	MAXMAX##.		
AZT	Aztec Code	DFT	Default all Data Matrix settings	AZTDFT.
		ENA	Off	AZTENA0.
			*On	AZTENA1.
		MIN	Min length (1-3750) *1	AZTMIN##.
		MAX	Max length (1-3750) *3750	AZTMAX##.
		RUN	*Disable Runes	AZTRUN0.
Enable Runes	AZTRUN1.			

### 3/ other configuration commands

TAG	SubTag	Setting	Command
ALL	ENA	Off	ALLEN0.
		On	ALLEN1.
PRE	BK2	AIM Symbology Identifier transmit (all symbologies)	PREBK2995C81.
		AIM Symbology Identifier transmit (EAN128)	PREBK2495C81.
	CA2	Do not transmit AIM ID	PRECA2.

### 4/ query commands

A special character can be used to query the device about its settings :  
 ‘?’ → What is the device’s current value for the setting(s).

#### **Tag Field Usage**

When a query is used in place of a Tag field, the query applies to the entire set of commands available for the particular storage table indicated by the Storage field of the command. In this case, the SubTag and Data fields should not be used because they are ignored by the device.

#### **SubTag Field Usage**

When a query is used in place of a SubTag field, the query applies only to the subset of commands available that match the Tag field. In this case, the Data field should not be used because it is ignored by the device.

#### **Data Field Usage**

When a query is used in place of the Data field, the query applies only to the specific command identified by the Tag and SubTag fields.

For example :

We want to know the current configuration for Code39 symbology :

The corresponding TAG is **C39**, then we sent the following command to the BRR-F :

field	Code ID	length	Payload
ASCII	-	--	<b>C39?</b>
Hex	<b>FE</b>	<b>00 05</b>	<b>43 33 39 3F 2E</b>

The BRR-F replies :

field	Code ID	length	Payload
ASCII	-	- -	<b>C39DFT[ACK],ENA1[ACK],SSX0[ACK],ASC0[ACK],APP0[ACK],CK20[ACK],B320[ACK],MIN0[ACK],MAX48[ACK],DCP2[ACK].</b>  <b>[ACK] is the ACK ascii character (0x06)</b>
Hex	<b>FE</b>	<b>00 3F</b>	<b>43 33 39 44 46 54 06 2C 45 4E 41 31 06 2C 53 53 58 30 06 2C 41 53 43 30 06 2C 41 50 50 30 06 2C 43 4B 32 30 06 2C 42 33 32 30 06 2C 4D 49 4E 30 06 2C 4D 41 58 34 38 06 2C 44 43 50 32 06 2E</b>

This means :

- Code39 is enabled (**ENA1**),
- Start/stop characters are not transmitted ()
- Full ASCII mode is disabled ()
- Code39 append is off
- Checksum is not calculated
- Code32 is off
- Code page is ISO2022

