



## 1.1 Table of Contents

1.1	TABLE OF CONTENTS .....	2
<b>2.</b>	<b>GENERAL .....</b>	<b>3</b>
2.1	GENERAL SYNTAX AND SETTINGS .....	3
2.2	CONFIGURATION SEQUENCE NUMBERS.....	4
2.3	MESSAGE EXCHANGE BETWEEN PRC AND PC .....	4
2.4	MESSAGE TIMING.....	4
<b>3.</b>	<b>PRC MESSAGES.....</b>	<b>5</b>
3.1	PRC TO PC - MONITORING/MEASUREMENT ('LIVE') DATA (M-MESSAGE) .....	5
3.2	PRC TO PC - CONFIGURATION SETTING (NON-TEXTS) (S-MESSAGE).....	7
3.3	PRC TO PC - CONFIGURATION SETTING (TEXTS) (T-MESSAGE).....	8
3.4	PRC TO PC – REPLY/CONFIRM MESSAGE (R-MESSAGE).....	9
<b>4.</b>	<b>PC MESSAGES.....</b>	<b>10</b>
4.1	PC TO PRC – REQUEST CONFIGURATION SETTINGS (Q).....	10
4.2	PC TO PRC – SET CONFIGURATION SETTINGS (NON-TEXT) (S) .....	10
4.3	PC TO PRC – SET CONFIGURATION SETTINGS (TEXT) (T) .....	11

## 2. General

This document describes the protocol for serial communication between the Portable Repeater Controller (PRC) and a PC (PC). Where a PC is mentioned in this document, it can be any equipment on the other end of the RS-232 link, for example a laptop, Raspberry Pi, Arduino or other microcontroller, just as long as it implements the protocol described in this document.

See <http://shop.emco-electronics.nl/> for the latest information about the Portable Repeater Controller project, including the instruction manual with a detailed, but more general description of the PRC.

### 2.1 General syntax and settings

#### Serial port settings

9600 8N1 (9600 baud, 8 bits, no parity, 1 stop bit)

#### Message format

ASCII	:	<T>	D	D	..	..	C	C	\r	\n
Decimal	58		ASCII of byte value in hex				ASCII of byte value in hex		13	10

- A message always starts with a ":" and ends with a \r\n.
- The second character/byte defines the message type.
- The two bytes before the \r\n contain the checksum CC of the message.
- All other bytes between the message type and the checksum CC are data (DD); the data structure is dependent on the message type.
- Both data (DD) and checksum (CC) are the ASCII values of the hexadecimal representation of the byte values (one byte is represented by two characters: 00 .. FF).

#### Data representation

The hexadecimal representation of binary is coded in ASCII alphanumeric characters. For example, the 8-bit binary value 0011 1110 is 3E in hexadecimal. To code this in ASCII, one 8-bit byte containing the ASCII code for the character "3" (0011-0011 or 033H) and one 8-bit byte containing the ASCII code for the character "E" (0100-0101 or 045H) are required. For each byte value, the high-order hexadecimal digit is always the first digit of the pair of hexadecimal digits.

#### Checksum

Each record ends with a checksum field that contains the ASCII hexadecimal representation of the two's complement of the 8-bit bytes that result from the summation of each byte, from and including the type field (so without the ":" ) to and including the last byte of data. Only the last byte of the result is used. Calculating the two complements is equal to invert all bits of the checksum byte plus one.

## 2.2 Configuration sequence numbers

- To indicate a change in the configuration settings, the PRC holds six sequences numbers, one for every call text and one for all other configuration settings together.
- When a configuration setting changes, the corresponding sequence number will be incremented. When the maximum value of the sequence number is reached it will restart at 0.
- For call texts the sequence numbers are 3-bit (0 to 7).
- The sequence number for the other settings is 8-bit (0-255).
- By observing the sequence numbers, the PC only has to request the configuration data when a sequence number has changed.
- The sequence numbers are sent from the PRC to the PC, see Chapter 3.

The five call text sequence numbers are combined together in one 16-bit word:

bit	Configuration sequence number
0-2	Sequence number call 1 (item 1)
3-5	Sequence number call 2 (item 2)
6-8	Sequence number call 3 (item 3)
9-11	Sequence number call 4 (item 4)
12-14	Sequence number call 5 (item 5)
15	Not used

## 2.3 Message exchange between PRC and PC

1. Four different types of messages can be sent from the PRC to the PC:
  1. M-messages with monitoring ('live') data.
  2. S-messages with configuration settings data (all other than call texts).
  3. T-messages with a call text.
  4. R-messages with a confirmation/reply to a new setting received from the PC.
2. Three types of messages can be sent from the PC to the PRC:
  1. Q-messages for requesting settings data from the PRC.
  2. S-messages for sending a (new) setting (all other than call texts).
  3. T-messages for sending a call text.
3. When no other messages have to be sent, every half a second an M-message with monitoring/measurement data is sent from the PRC to the PC. This message will include all sequence numbers to notify the PC of any configuration setting changes.
4. When a Q-message was received by the PRC, the corresponding S or T-message will be sent back to the PC with the requested setting(s).
5. When an S or T-message was received by the PRC, a reply message with the received settings is sent back to the PC. The reply message will be sent back immediately after the received data is processed in the PRC.

## 2.4 Message timing

1. The PRC only has a single receive buffer for incoming messages from the PC, therefore **only send one message at a time to the PRC.**
2. Only send a message to the PRC directly after a message was sent out by the PRC!

### 3. PRC messages

#### 3.1 PRC to PC - Monitoring/measurement ('live') data (M-message)

##### Message syntax

```
:M [14 * DD] CC \r\n
```

Valid example:

```
:M1432004100010101020077112C003A + \r\n
```

byte	Value	Details
0	:	ASCII ":", decimal byte value 58
1	M	ASCII "M", decimal byte value 77
2+3	Firmware version	VVV is version VV.V (item 90)
4+5	Config settings seq number	Configuration settings sequence number (1)
6+7	Config texts seq number high	Configuration texts sequence number H (2)
8+9	Config texts seq number low	Configuration texts sequence number L (2)
10+11	System status	See details below
12+13	RX status	See details below
14+15	TX status	See details below
16+17	Time HH	0-23 (item 56)
18+19	Time MM	0-59 (item 57)
20+21	ADC high byte – voltage	ADC battery voltage high value (3)
22+23	ADC low byte – voltage	ADC battery voltage low value (3)
24+25	ADC - CTCSS	ADC CTCSS level (3)
26+27	ADC – DTMF main	ADC DTMF level main (4)
28+29	ADC – DTMF sub	ADC DTMF level sub (4)
30+31	Checksum	See 2.1
32	\r	Decimal byte value 13
33	\n	Decimal byte value 10

- (1): An 8-bit (0-255) rolling sequence number (see 2.2).
- (2): Five 3-bit (0-7) rolling sequence numbers (see 2.2).
- (3): 0-30 Volt: 0-300.
- (4): 0-100%: 0-100

Bit	System status	Details
0	Disabled internal	0 = enabled, 1= disabled (item 51)
1	Disabled external	0 = enabled, 1= disabled
2	Disable timer	0 = stopped, 1 = running (item 58)
3	Enable timer	0 = stopped, 1 = running (item 59)
4	Battery low	0 = battery OK, 1 = low battery
5	Read-only buttons	Read-only because local buttons active
6	Read-only DTMF	Read-only because DTMF command active
7	Read-only ser off	Read-only because serial off (item 83)

Bit	RX status	Details
0	SQL	1 = squelch open
1	Time-out	1 = time-out
2	1750 main	1 = 1750 detected
3	CTCSS	1 = CTCSS detected
4	Overload CTCSS	1 = ADC CTCSS overload
5	Overload main	1 = ADC DTMF MAIN overload
6	Overload sub	1 = ADC DTMF SUB overload
7	Reserved	-

Bit	TX status	Details
0	TX On	0 = TX off, 1 = TX on (also when PTT blocked)
1	CW active	1 = CW call
2	CW active	1 = CW beacon
3	CW active	1 = CW roger
4	Blocked internal	1 = blocked (item 50)
5	Blocked external	1 = blocked
6	Reserved	-
7	Reserved	-

## 3.2 PRC to PC - Configuration setting (non-texts) (S-message)

Configuration settings from 10 to 99 are all byte values. See the Instruction Manual Chapter 7.2 for the IDs and valid values for each configuration item.

### Message syntax

```
:S [46 * DD] CC \r\n
```

Valid example:

```
:S320101040506020103060107010107000900090000010A0A13000A0000000A050A0D1103  
000006000401016300009B + \r\n
```

Byte	Value	Details
0	:	ASCII ":", decimal byte value 58
1	S	ASCII "S", decimal byte value 83
2+3	Config settings seq number	Configuration settings sequence number (1)
4+5	Item 10	
6+7	Item 11	
8+9	Item 12	
10+11	Item 13	
12+13	Item 14	
14+15	Item 15	
16+17	Item 20	
18+19	Item 21	
20+21	Item 22	
22+23	Item 23	
24+25	Item 24	
26+27	Item 30	
28+29	Item 31	
30+31	Item 32	
32+33	Item 40	
34+35	Item 41	
36+37	Item 42	
38+39	Item 43	
40+41	Item 44	
42+43	Item 45	
44+45	Item 46	
46+47	Item 47	
48+49	Item 48	
50+51	Item 52	
52+53	Item 53	
54+55	Item 54	
56+57	Item 55	
58+59	Item 58	
60+61	Item 59	
62+63	Item 60	
64+65	Item 61	
66+67	Item 62	

68+69	Item 63	
70+71	Item 70	
72+73	Item 71	
74+75	Item 72	
76+77	Item 73	
78+79	Item 74	
80+81	Item 75	
82+83	Item 80	
84+85	Item 81	
86+87	Item 82	
88+89	Item 91	
90+91	Item 92	
92+93	Item 93	
94+95	Checksum	See 2.1
96	\r	Decimal byte value 13
97	\n	Decimal byte value 10

(1): An 8-bit (0-255) rolling sequence number (see 2.2).

### 3.3 PRC to PC - Configuration setting (texts) (T-message)

Configuration settings 1 to 5 (the calls) are 15 characters call texts.

#### Message syntax

```
:T DD [15 * TT] CC \r\n
```

Valid example:

```
:T0101504930505243202020202020202010 + \r\n
```

Byte	Value	Details
0	:	ASCII ":", decimal byte value 58
1	T	ASCII "T", decimal byte value 84
2+3	Config texts seq number	Configuration text sequence number (1)
4+5	Text setting ID	Item ID 1, 2, 3, 4 or 5.
6-35	Text setting bytes	15 ASCII character value for call text (2)
36+37	Checksum	See 2.1
38	\r	Decimal byte value 13
39	\n	Decimal byte value 10

(1): One 3-bit (0-7) rolling sequence number (see 2.2).

(2): Valid characters are 'A' ... 'Z' (decimal 65 ... 90), '0' ... '9' (decimal 48 ... 57), '/' (decimal 47) and space (decimal 32).



### 3.4 PRC to PC – Reply/confirm message (R-message)

R-messages from the PRC confirm the reception of a configuration setting message from the PC to the PRC. Even if the value of a configuration setting was invalid, a reply will still be sent by the PRC to the PC. By detecting the changed configuration settings number, the PC can request the changed value.

#### Message syntax

```
:R [4 * DD] CC \r\n
```

Valid example:

```
:R011012C214 + \r\n
```

Byte	Value	Details
0	:	ASCII ":", decimal byte value 58
1	R	ASCII "R", decimal byte value 82
2+3	Confirmed setting ID	
4+5	Config settings seq number	Configuration settings sequence number (1)
6+7	Config texts seq number high	Configuration texts sequence number H (2)
8+9	Config texts seq number low	Configuration texts sequence number L (2)
10+11	Checksum	See 2.1
12	\r	Decimal byte value 13
13	\n	Decimal byte value 10

- (1): An 8-bit (0-255) rolling sequence number (see 2.2).
- (2): Five 3-bit (0-7) rolling sequence numbers (see 2.2).

## 4. PC messages

### 4.1 PC to PRC – Request Configuration settings (Q)

For text configuration settings (id 1, 2, 3, 4 or 5) one setting at a time can be requested. The PRC will reply with a T-message (see 3.3).

For non-text settings (all ids between 10 and 99) all settings can be requested at once by requesting item 255. The PRC will reply with an S-message (see 3.2).

#### Message syntax

```
:Q DD CC \r\n
```

Valid examples:

```
:Q034C + \r\n  
:QFF23 + \r\n
```

Byte	Value	Details
0	:	ASCII ":", decimal byte value 58
1	Q	ASCII "Q", decimal byte value 81
2+3	Setting ID	Id (1, 2, 3, 4 or 5) or 255 for all non-texts
4+5	Checksum	See 2.1
6	\r	Decimal byte value 13
7	\n	Decimal byte value 10

### 4.2 PC to PRC – Set Configuration settings (non-text) (S)

Configuration settings from 10 to 99 are all byte values. See the Instruction Manual Chapter 7.2 for the IDs and valid values for each configuration item.

#### Message syntax

```
:S DD DD CC \r\n
```

Valid example:

```
:S5B43CF + \r\n
```

Byte	Value	Details
0	:	ASCII ":", decimal byte value 58
1	S	ASCII "S", decimal byte value 83
2+3	Setting ID	Item ID from 10 to 99
4+5	Setting value	
6+7	Checksum	See 2.1
8	\r	Decimal byte value 13
9	\n	Decimal byte value 10

### 4.3 PC to PRC – Set Configuration settings (text) (T)

Configuration settings 1 to 5 (the calls) are 15 characters call texts.

#### Message syntax

```
:T DD [15 * TT] CC \r\n
```

Valid example:

```
:T01504930505243202020202020202071 + \r\n
```

Byte	Value	Details
0	:	ASCII ":", decimal byte value 58
1	T	ASCII "T", decimal byte value 84
2+3	Setting ID	Item ID 1, 2, 3, 4 or 5.
4-33	Text setting bytes	15 ASCII character value for call text (1)
34+35	Checksum	See 2.1
36	\r	Decimal byte value 13
37	\n	Decimal byte value 10

(1): Valid characters are 'A' ... 'Z' (decimal 65 ... 90), '0' ... '9' (decimal 48 ... 57), '/' (decimal 47) and space (decimal 32).