B42 Protocol Specification

v1.1
2020-11-11

B42 is a generic low level protocol for byte based communication channels (such as serial). Its main purpose is easy transmission of small command packets between hosts (computers) and microcontroller boards.
B42 supports up to 3 data bytes (up to 18 bits payload) along with the (mandatory) command byte. The command byte contains 4 bits command code and 2 bits for the number of following data bytes.
Framing and basic error detection is accomplished by a 2 bit sequence number for every transmitted byte. Zero bytes (0x00) are considered protocol errors. Invalid and incomplete frames are ignored. Each valid command byte starts a new frame.

The higher level application protocols define their custom command codes and data bits on top of B42. Command code zero (0x0) is reserved and should not be used by application protocols.

developed by:
Archimedes Exhibitions GmbH, Berlin, Germany, Earth

B42 COMMAND BYTE

7	6	5	4	3	2	1	0
0	0	n1	n0	c3	c2	c1	c0

B42 DATA BYTE(S)

7	6	5	4	3	2	1	0
s1	s0	d5	d4	d3	d2	d1	d0

n:	[0..3]	number of data bytes
s:	[1..3]	sequence number
c:	[0..15]	command code
d:	[0..63]	data

B42 FRAMES

n=0

7	6	5	4	3	2	1	0
0	0	0	0	c3	c2	c1	c0

c: [0..15]

n=1

7	6	5	4	3	2	1	0
0	0	0	1	c3	c2	c1	c0

s=1

7	6	5	4	3	2	1	0
0	1	d5	d4	d3	d2	d1	d0

c: [0..15]
d: [0..63]

n=2

7	6	5	4	3	2	1	0
0	0	1	0	c3	c2	c1	c0

s=1

7	6	5	4	3	2	1	0
0	1	d11	d10	d9	d8	d7	d6

s=2

7	6	5	4	3	2	1	0
1	0	d5	d4	d3	d2	d1	d0

c: [0..15]
d: [0..4095]

n=3

7	6	5	4	3	2	1	0
0	0	1	1	c3	c2	c1	c0

s=1

7	6	5	4	3	2	1	0
0	1	d17	d16	d15	d14	d13	d12

s=2

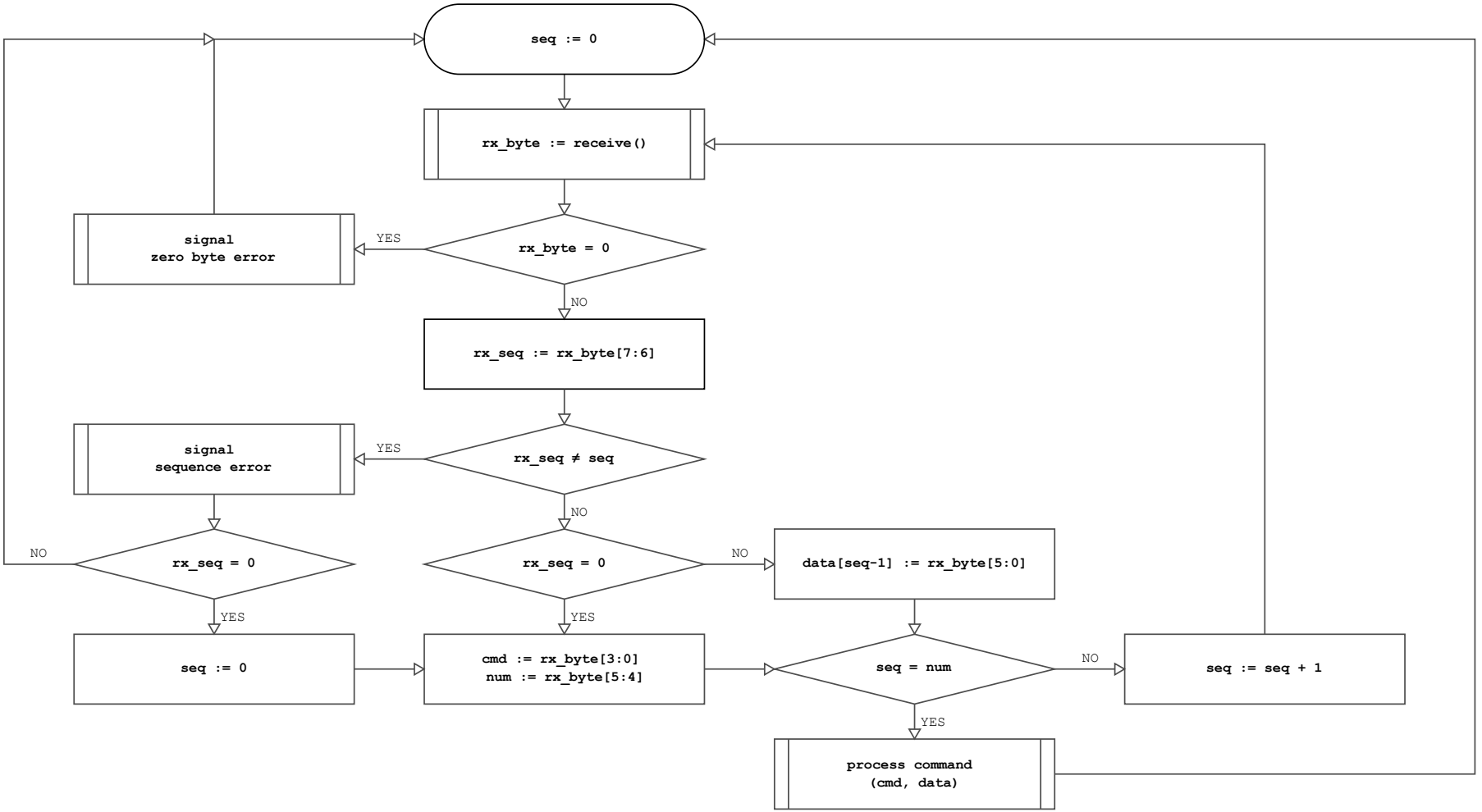
7	6	5	4	3	2	1	0
1	0	d11	d10	d9	d8	d7	d6

s=3

7	6	5	4	3	2	1	0
1	1	d5	d4	d3	d2	d1	d0

c: [0..15]
d: [0..262143]

```
Start:
          seq := 0                              /* expected sequence number (0: command byte) */
          num := 0                              /* expected number of data bytes */
          cmd := 0                              /* command code */
          data := [0, 0, 0]                     /* command data */

Loop:
          rx_byte := receive_byte()

          /* check for invalid zero byte */
          if rx_byte = 0:
               signal_zero_byte_error()
               goto Start                       /* ignore byte / current frame */

          /* check for invalid byte sequence */
          rx_seq := rx_byte[7:6]
          if rx_seq != seq:
               signal_sequence_error()
               if rx_seq = 0:
                    goto Command                /* command byte -> start of a new frame */
               goto Start                       /* ignore byte / current frame */

          /* process received byte */
          if seq = 0:
               goto Command                     /* command byte -> start of a new frame */
          data[seq - 1] := rx_byte[5:0]         /* data byte -> store data bits */
          if seq = num:
               process_command(cmd, data)       /* frame complete */
               goto Start
          seq := seq + 1                        /* expect next data byte */
          goto Loop

Command:
          cmd := rx_byte[3:0]
          num := rx_byte[5:4]
          if num = 0:
               process_command(cmd)             /* frame complete (no data bytes) */
               goto Start
          seq := 1                              /* expect first data byte */
          goto Loop
```


seq := 0
rx_byte := receive()
rx_byte = 0
YES
signal
zero byte error
NO
rx_seq := rx_byte[7:6]
rx_seq ≠ seq
YES
signal
sequence error
NO
rx_seq = 0
NO
YES
seq := 0
rx_seq = 0
NO
YES
cmd := rx_byte[3:0]
num := rx_byte[5:4]
data[seq-1] := rx_byte[5:0]
seq = num
NO
YES
seq := seq + 1
process command
(cmd, data)

The following reference/example implementations are available:

Board side	b42lib	B42 handler firmware library for Arduino/PlatformIO. https://github.com/amdx/b42lib
Host side	pyb42	B42 and convenience handlers library for Python3. https://github.com/amdx/pyb42

VERSION/DATE	AUTHOR	CHANGES
1.0 2015-02-02	Thomas Schott, Archimedes Exhibitions GmbH	Initial version.
1.1 2020-11-11	Thomas Schott, Archimedes Exhibitions GmbH	Add purpose note on Front page. Add sequence number reset to Receiver flowchart. Add Implementations page. Add Change log page.