



B42 Protocol Specification

v1.0
2015-02-02

B42 is a generic low level protocol for byte based communication channels (such as serial). It 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.

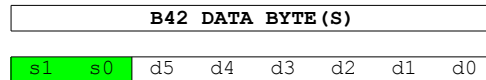
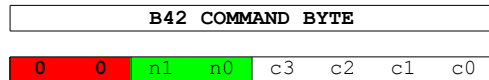
Framing and basic error detection is accomplished by a 2 bit sequence number for every transmitted byte. Zero bytes (0x00) are considered protocol errors. Invalid and incomplete frames are ignored. Each valid command byte starts a new frame.

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.

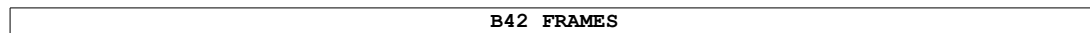
developed by:
Archimedes Exhibitions GmbH, Berlin, Germany, Earth



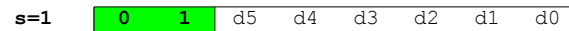
Bytes and Frames



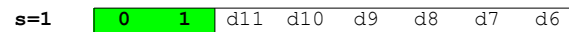
n: [0..3] number of data bytes
 s: [1..3] sequence number
 c: [0..15] command code
 d: [0..63] data



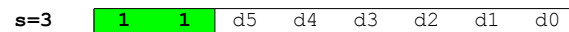
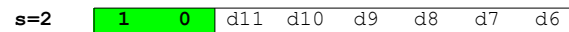
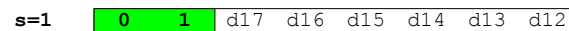
c: [0..15]



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



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



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

Receiver pseudo code

```
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
```

Receiver flowchart

