# BlueSense2 – Quick Start Guide

## *1. Setting up*

### 1.1 Terminal

A terminal must be installed to communicate with the sensor, either over Bluetooth or USB. Examples of Terminal software on Windows include:

- Termite 3.2 (or better): http://www.compuphase.com/software_termite.htm
- TeraTerm 4.89 (or better): https://en.osdn.jp/projects/ttssh2/releases/

## *1.2 Setting up communication and checking ports*

Communication with the node occurs over two virtual serial ports through the Bluetooth chip or the USB cable. In order to communicate with the sensor the port number corresponding to these two interfaces is required.

### 1.2.1 USB

When connecting the node over USB, the device manager shows the port number corresponding to the USB chip (COM11 in the figure).
If the device appears as unrecognized, you must install the FTDI VCP drivers from:
http://www.ftdichip.com/Drivers/VCP.htm

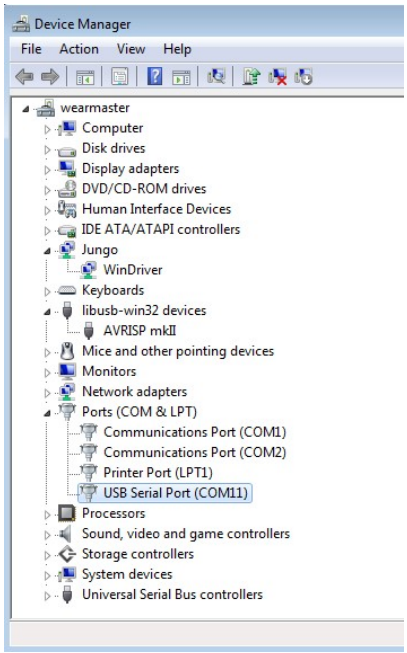



Figure 1. Communication over USB occurs through COM11 on this computer.

### 4.2.2 Bluetooth

For Bluetooth, several steps are required:

- Add the bluetooth device (using the control panel or bluetooth icon in the system tray). It will appear as "BlueSense-xxxx". Select it and provide the pin code, which is "0000"
- Windows will install a serial port profile driver and indicate which COM port corresponds to the Bluetooth link. Usually two COM ports appear and the lower numbered one is the correct one, however in case of communication issues try the other number.

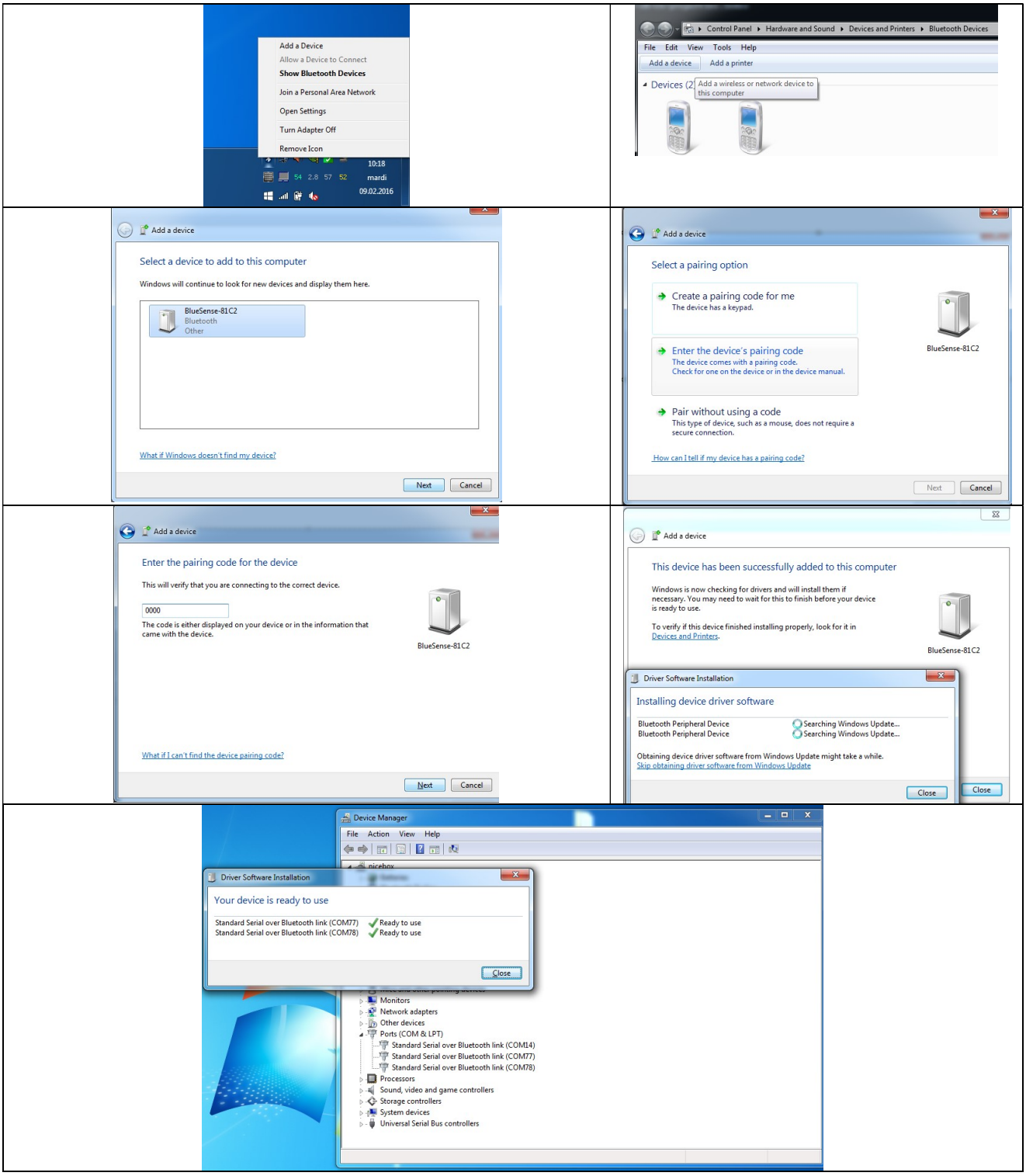



Figure 2. Setting up Bluetooth

## *2. Turning on and off*

The sensor turns on when plugged into the USB port. Alternatively press the power button a half second to turn it on.

To turn off the sensor press and hold the power button for 10 seconds.

## 3. BlueSense User Interface

After connecting BlueSense (either over USB or Bluetooth) use a Terminal application to access the user interface of BlueSense. The user can control BlueSense over Bluetooth or USB using a "command line" mode[1].

### 3.1 General principles

BlueSense can enter different modes: interactively waiting for commands, streaming data, logging data, etc. On power-up, the main mode is active

Figure 3 illustrate the options available in the main mode when the user issues the *help* command (H). Figure 4 illustrates how to query and set date and time from the RTC and battery status.

```
H
Available commands:
        H       Help
        T       T[,<hh><mm><ss>] Query or set time
        D       D[,<dd><mm><yy>] Query or set date
        Z       Z[,<hh><mm><ss><dd><mm><yy>]: synchronise time.
                No parameters: reset local time/date; otherwise sets the RTC and local
time/date to hhmmss ddmmyy
        Y       Test sync
        R       RN-41 terminal
        t       Time-related tests
        A       A,<hex>,<us>: ADC mode. hex: ADC channel bitmask in hex; us: sample
period in microseconds
        I       I<period>,<#tx>: Sets USB IO parameters. IO at (period+1)/1024Hz.
                Use #tx slots for transmission before reception.
        M       M[,<mode>[,<logfile>[,<duration>]]: without parameters lists available
modes, otherwise enters the specified mode.
                Optionally logs to logfile (use -1 not to log) and runs for the
specified duration in seconds.
        m       MPU test mode
        W       Swap primary and secondary interfaces
        O       O[,sec] Power off and no wakeup, or wakeup after sec seconds
        o       Display power used in off mode; if the node was turned off with O
        F       F,<bin>,<pktctr>,<ts>,<bat>,<label>: bin: 1 for binary, 0 for text; for
others: 1 to stream, 0 otherwise
        i       i,<ien> Prints battery and logging information when streaming/logging
when ien=1
        Q       Long-term battery info
        q       Short-term battery info
        c       Lists timer callbacks
        X       SD card test mode
        S       S,<us>: test streaming/logging mode; us: sample period in microseconds
        b       b[,run;a boot;script] Prints the current bootscript or sets a new one;
multiple commands are delimited by a ;
        ?       Identify device by blinking LEDs
        ~       Clear boot counter
CMDOK
```

Figure 3. Output of the *help* command in the main interface.

```
?
My name is 835E
CMDOK
D
12.05.18
CMDOK
T
13:45:20
CMDOK
T,135000
Time: 13:50:00
CMDOK
T
13:50:02
CMDOK
q
#V=4142 mV; I=-23 mA; P=-95 mW
-95 -94 -95 -95 -97 -95 -93 -93 -98 -95
CMDOK
```

[1] This command line mode is a text-based command/response protocol also suitable for programmatic control.

Figure 4. Output of the some of the other commands in the main interface.

Some modes are used to stream or log data. Any mode can be exited using the command !, which returns to the previous or the main mode.

# *4. Motion sensing*

## 4.2 Motion sensing modes

Multiple motion sensing modes are available (figure 5).

```
M
Motion modes (x indicates experimental modes; sample rate not respected):
[0]     Motion off
[1]       500Hz Gyro (BW=250Hz) x
[2]       500Hz Gyro (BW=184Hz)
[3]       200Hz Gyro (BW= 92Hz)
[4]       100Hz Gyro (BW= 41Hz)
[5]        50Hz Gyro (BW= 20Hz)
[6]        10Hz Gyro (BW=  5Hz)
[7]         1Hz Gyro (BW=  5Hz)
[8]      1000Hz Acc  (BW=460Hz) x
[9]       500Hz Acc  (BW=184Hz)
[10]      200Hz Acc  (BW= 92Hz)
[11]      100Hz Acc  (BW= 41Hz)
[12]       50Hz Acc  (BW= 20Hz)
[13]       10Hz Acc  (BW=  5Hz)
[14]        1Hz Acc  (BW=  5Hz)
[15]     1000Hz Acc  (BW=460Hz) Gyro (BW=250Hz) x
[16]      500Hz Acc  (BW=184Hz) Gyro (BW=250Hz) x
[17]      500Hz Acc  (BW=184Hz) Gyro (BW=184Hz)
[18]      200Hz Acc  (BW= 92Hz) Gyro (BW= 92Hz)
[19]      100Hz Acc  (BW= 41Hz) Gyro (BW= 41Hz)
[20]       50Hz Acc  (BW= 20Hz) Gyro (BW= 20Hz)
[21]       10Hz Acc  (BW=  5Hz) Gyro (BW=  5Hz)
[22]        1Hz Acc  (BW=  5Hz) Gyro (BW=  5Hz)
[23]      500Hz Acc low power
[24]      250Hz Acc low power
[25]      125Hz Acc low power
[26]     62.5Hz Acc low power
[27]    31.25Hz Acc low power
[28]        1Hz Acc low power
[29]     1000Hz Acc  (BW=460Hz) Gyro (BW=250Hz) Mag 8Hz x
[30]      500Hz Acc  (BW=184Hz) Gyro (BW=250Hz) Mag 8Hz x
[31]      500Hz Acc  (BW=184Hz) Gyro (BW=184Hz) Mag 8Hz
[32]      100Hz Acc  (BW= 41Hz) Gyro (BW= 41Hz) Mag 8Hz
[33]       50Hz Acc  (BW= 20Hz) Gyro (BW= 20Hz) Mag 8Hz
[34]     1000Hz Acc  (BW=460Hz) Gyro (BW=250Hz) Mag 100Hz x
[35]      500Hz Acc  (BW=184Hz) Gyro (BW=250Hz) Mag 100Hz x
[36]      500Hz Acc  (BW=184Hz) Gyro (BW=184Hz) Mag 100Hz
[37]      200Hz Acc  (BW= 92Hz) Gyro (BW= 92Hz) Mag 100Hz
[38]      100Hz Acc  (BW= 41Hz) Gyro (BW= 41Hz) Mag 100Hz
[39]      500Hz Acc  (BW=184Hz) Gyro (BW=184Hz) Mag 100Hz Quaternions
[40]      200Hz Acc  (BW= 92Hz) Gyro (BW= 92Hz) Mag 100Hz Quaternions
[41]      100Hz Acc  (BW= 41Hz) Gyro (BW= 41Hz) Mag 100Hz Quaternions
[42]      500Hz Quaternions Qsg=(qw,qx,qy,qz); rotates vector in earth coords G into
sensor coords S
[43]      200Hz Quaternions
[44]      100Hz Quaternions
[45]      100Hz Tait-Bryan/aerospace/zx'y", intrinsic (yaw, pitch, roll)
[46]      100Hz Quaternions debug (angle, x,y,z)
CMDOK
```

Figure 5. Available motion sensing modes

## 4.2 Motion data streaming

Figure 6 shows streaming of acceleration, gyro and magnetic field in text mode after starting mode M,33.

Streaming can be terminated by issuing the ! command.

```
M,33
CMDOK
<IDLE
SMPLMOTION>
Acc scale: 3
Gyro scale: 3
Sample rate: 50
-00061 -01870  00787 -00015  00028 -00001  00073  00106 -00030
-00071 -01870  00789 -00009 -00017 -00013  00073  00106 -00030
-00082 -01879  00767  00011 -00010 -00001  00073  00106 -00030
-00064 -01897  00755  00013  00006  00020  00073  00106 -00030
-00040 -01887  00766  00001  00010  00019  00074  00111 -00030
-00054 -01883  00763  00000 -00006 -00006  00074  00111 -00030
-00078 -01886  00772  00002  00003 -00013  00074  00111 -00030
-00073 -01878  00776 -00006  00003  00000  00074  00111 -00030
-00062 -01883  00776  00004  00004  00010  00074  00111 -00030
-00056 -01880  00764  00009  00004  00012  00074  00111 -00030
-00061 -01888  00763  00006 -00009  00001  00071  00106 -00035
```

Figure 6. Streaming of acceleration, gyroscope and magnetometer data in text mode.

## 4.3 Controlling the motion streaming format

The command F is used to specify the streaming format. Changes to the format are persistent across reboots.

The command F accepts the following parameters:

```
F,<bin>,<pktctr>,<ts>,<bat>,<label>: bin: 1 for binary, 0 for text; for others: 1 to stream, 0 otherwise
```

In order to stream the data in ASCII format, without packet counter (pktctr), without timestamp (ts), without battery level (bat) and without label (label), issue F,0,0,0,0,0.

## 4.4 Controlling the motion sensor range and calibration

The motion sensor has different ranges for the accelerometer and gyroscopes, and the magnetic field sensor can be calibrated. These settings can be changed in the “MPU test mode”. From the main mode, issue the command m to access the MPU settings (figure 7).

```
m
CMDOK
<IDLE
H
Available commands:
        H       Help
        L       L[,<scale>] read or set the accelerometer full scale; 0=2G, 1=4G, 2=8G,
3=16G; persistent
        l       l[,<scale>] read or set the gyroscope full scale; 0=250dps, 1=500dps,
2=1000dps, 3=2000dps; persistent
        G       G[,<mode>] User magnetometer correction, or set 0='no correction'
1='correction w/ factory', 2='user correction'; persistent
        g       g: get magnetometer correction mode
        B       Benchmark overheads of auto acquire
        b       b[,betax100]: gets or sets the beta correction gain for the orientation
sensing; suggested: 35 for b=0.035 (persistent)
        C       Calibrate acc/gyro
        c       Acquire calibration data
        o       o,<offX>,<offY>,<offZ> Set the gyro bias
        R       Dumps registers
        S       S[,<mode>,<autoread>] setup motion sensor mode
        X       Reset
        F       Dump FIFO content
        f       f,flags,en,rst: Set FIFO flags (TEMP GX GY GZ ACC SLV2 SLV1 SLV0),
enable (1), reset (1)
        P       Poll acc, gyr, magn, temp
        A       A[,<mode] Auto acquire (interrupt-driven) test
        M       Magnetometer registers
        m       m,<0|1|2> magnetometer: power down|enable 8Hz|enable 100Hz
        I       I,<0|1> disable|enable I2C
        E       MPU external sensor registers
        W       W,en,dly,regstart,numreg: en=1|0 enables|disables shadowing of numreg
registers from regstart at frequcency ODR/(1+dly)
        O       MPU off
        t       Magnetic selt test
```

```
       K       K,bitmap: 3-bit bitmap indicating whether to null acc|gyr|mag (not
persistent)
       !       Exit current mode
CMDOK
```

Figure 7. MPU settings

The recommended settings for human movement sensing are:

- L,3
- l,3
- G,2

These set the accelerometer to 16G range (L,3), the gyroscope to 2000dps range (l,3), and the magnetic field sensor to "user correction" (G,2).

The user magnetic field calibration must be done issuing the command G (without parameter), and rotating the sensor along all axes in space until no more data is displayed in the console. After that, press return to store the calibration data (figure 8).
All these settings are persistently stored.
Exit the MPU test mode with !.

```
G
Magnetometer calibration: far from any metal, move the sensor in all
orientations until no new numbers appear on screen and then press a
key
[418 214 -36] - [418 214 -36]
[409 214 -36] - [418 227 -35]
[409 214 -36] - [421 227 -27]
[405 214 -36] - [421 227 -27]
[346 67 -96] - [427 227 -27]
[341 67 -96] - [427 227 -27]
[336 67 -96] - [427 227 -27]
340 72 -88
[330 67 -96] - [427 227 -27]
[329 67 -96] - [427 227 -27]
[240 67 -96] - [427 227 -27]
[70 67 -96] - [427 227 14]
[49 67 -96] - [427 227 31]
[4 67 -96] - [427 227 138]
[-3 67 -96] - [427 227 518]
[-20 42 -124] - [427 227 520]
268 64 -124
[-20 42 -124] - [432 227 520]
[-20 -30 -124] - [560 227 520]
416 30 -64
[-20 -30 -124] - [567 265 520]
443 231 -31
92 312 106
[-20 -30 -124] - [567 313 532]
[-20 -30 -124] - [567 313 536]
[-20 -30 -124] - [567 313 537]
[-20 -30 -124] - [567 313 538]
[-28 -30 -124] - [567 313 538]

Mag bias: -269 -141 -207
Mag sens: 55 95 49
Mag corrmode: 2
CMDOK
```

Figure 8. Magnetic field calibration

### 4.5 Orientation sensing

Most motion sensing modes provide raw motion sensor data. However, these can be combined to obtain the attitude and heading of the device in an earth coordinate system.
In order to obtain accurate orientation, the device must not be moving for about 15 seconds from power-on, as this is used to reset the gyroscope. The magnetic field sensor must also be set to "user correction" and calibrated.
Mode M,44 streams the device orientation in quaternions. Quit the mode with !.

### 4.6 Motion data logging

First, see section 5 on how to handle the SD card formatting.

There are two ways to log data:

1. From the main mode, issue M,<*mode*>,<*log#*>,<*duration*>. This instructs to enter a specific motion sensing *mode*, log the data to logfile *log#* and terminate logging after *duration* seconds. If duration is omitted, logging continues until the command !.
2. Enter the desired motion sensing *mode* with M,<*mode*>. Once the sensor starts streaming the data start logging with L,<*log#*>. This starts logging the data on log file *log#*. Terminate logging with the command !.

## *5. SD card*

BlueSense supports only SDHC (not regular SD) cards with up to 32GB capacity[2].

BlueSense allows to log data locally to the SD card in log files. The SD card can be formatted in a FAT32 compatible filesystem from within BlueSense. The data can then be copied to a computer after logging is completed.

The application firmware has been optimised to achieve high sample rate (e.g. guaranteed 1KHz ADC, or 500Hz motion sensing). This requires to use a special layout of the files in the filesystem. In particular, log files are pre-layed out during formatting [**Error! Reference source not found.**]. There is a fixed maximum number of log files of 14. The number of log files must be selected during formatting.

The SD card is formatted, by entering the SD card mode (command X) and then issuing the format command (command F,<numlogfiles>) (figure 9).

Note that the maximum numlogfiles is 14 and minimum 1. The number of log files must be selected according to the size of the SD card and amount of data to store. Generally with an SD card of 32GB formatting with 14 log files is recommended. Each log file is up to 32/14=2.3GB in size, which in practice allows to store data continuously until the battery would run out.

```
X
CMDOK
<IDLE
SD>
F,14
Formatting with 14 log files
uFAT: Init SD...
CSD:
          CSD: 1
          TAAC: E
          NSAC: 0
          TRAN_SPEED: 32
          CCC: 5B5
          READ_BL_LEN: 9
          READ_BL_PARTIAL: 0
          WRITE_BLK_MISALIGN: 0
          READ_BLK_MISALIGN: 0
          DSR_IMP: 0
          C_SIZE: 00003A8F (7675904 KB)
```

[2] SDXC has not been tested, and SDHC with larger than 32GB capacity have not been trested.

```
          ERASE_BLK_EN: 1
          SECTOR_SIZE: 7F
          WP_GRP_SIZE         : 0
          WP_GRP_ENABLE: 0
          R2W_FACTOR: 2
          WRITE_BL_LEN: 9
          WRITE_BL_PARTIAL: 0
          FILE_FORMAT_GRP: 0
          COPY: 0
          PERM_WRITE_PROTECT: 0
          TMP_WRITE_PROTECT: 0
          FILE_FORMAT: 0
          CRC: 43
SDSTAT:
          DAT_BUS_WIDTH: 00
          SECURED_MODE: 00
          SD_CARD_TYPE: 0000
          SIZE_OF_PROTECTED_AREA: 03000000
          SPEED_CLASS: 04 (class 10)
          PERFORMANCE_MOVE: 04
          AU_SIZE: 09 (4096KB)
          ERASE_SIZE: 0008
          ERASE_TIMEOUT: 02
          ERASE_OFFSET: 02
          UHS_SPEED_GRADE: 01
          VIDEO_SPEED_CLASS: 00
          VSC_AU_SIZE: 00
          SUS_ADDR: 00
          APP_PERF_CLASS: 00
          PERFORMANCE_ENHANCE: 00
          DISCARD_SUPPORT: 00
          FULE_SUPPORT: 00
uFAT: Erasing card
uFAT: Writing MBR...
uFAT: Writing bootsect...
uFAT: Writing bkp bootsect...
uFAT: Clearing ROOT and FAT
uFAT: Writing root...
uFAT: Writing FAT for root...
uFAT: Writing FAT for log 0
uFAT: Writing FAT for log 1
uFAT: Writing FAT for log 2
uFAT: Writing FAT for log 3
uFAT: Writing FAT for log 4
uFAT: Writing FAT for log 5
uFAT: Writing FAT for log 6
uFAT: Writing FAT for log 7
uFAT: Writing FAT for log 8
uFAT: Writing FAT for log 9
uFAT: Writing FAT for log 10
uFAT: Writing FAT for log 11
uFAT: Writing FAT for log 12
uFAT: Writing FAT for log 13
uFAT: Reading MBR...
uFAT: Reading bootsect...
uFAT: key info:
          Card capacity: 15351808 sectors
          Partition start sector: 8192
          Partition capacity: 15343616 sectors
          FAT1 start sector: 8224
          FAT2 start sector: 0
          Cluster start sector: 10097
          Sectors/clusters: 64
          Root cluster: 2
          Number of data clusters: 239714
uFAT: id: E5 checksum: 35 (obtained: 35)
uFAT: numlogs: 14 startcluster: 128 sizecluster: 17024 sizebytes: 557842432
uFAT: Logs:
          LOG-3FF3.000 start sector: 18161 start cluster: 128 size: 00000000h
          LOG-3FF3.001 start sector: 1107697 start cluster: 17152 size: 00000000h
          LOG-3FF3.002 start sector: 2197233 start cluster: 34176 size: 00000000h
          LOG-3FF3.003 start sector: 3286769 start cluster: 51200 size: 00000000h
          LOG-3FF3.004 start sector: 4376305 start cluster: 68224 size: 00000000h
          LOG-3FF3.005 start sector: 5465841 start cluster: 85248 size: 00000000h
          LOG-3FF3.006 start sector: 6555377 start cluster: 102272 size: 00000000h
          LOG-3FF3.007 start sector: 7644913 start cluster: 119296 size: 00000000h
          LOG-3FF3.008 start sector: 8734449 start cluster: 136320 size: 00000000h
          LOG-3FF3.009 start sector: 9823985 start cluster: 153344 size: 00000000h
          LOG-3FF3.010 start sector: 10913521 start cluster: 170368 size: 00000000h
          LOG-3FF3.011 start sector: 12003057 start cluster: 187392 size: 00000000h
          LOG-3FF3.012 start sector: 13092593 start cluster: 204416 size: 00000000h
          LOG-3FF3.013 start sector: 14182129 start cluster: 221440 size: 00000000h
CMDOK
!
CMDOK
<SD
IDLE>
```

Figure 9. SD Card Formatting

## 5.2 Reading the log files

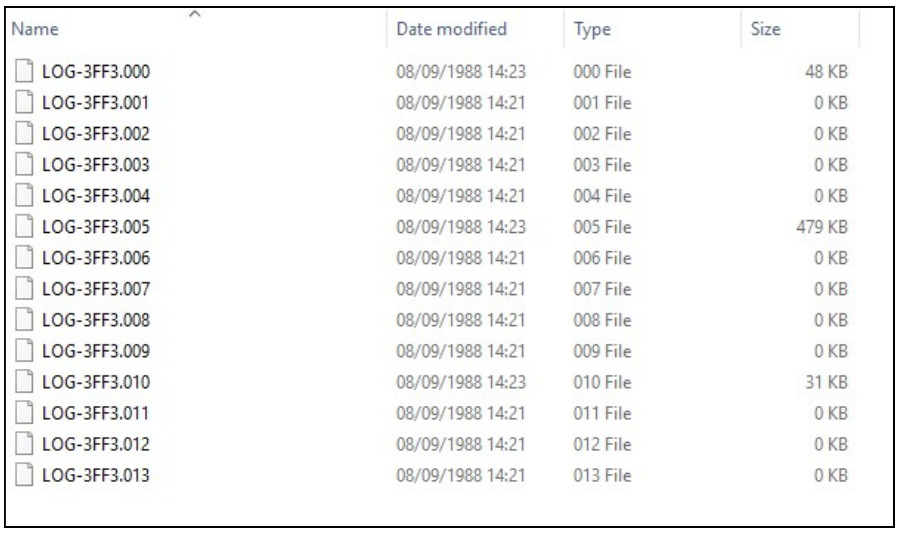
The size of the logs can be obtained from the SD card mode (command X) with the command V (figure 10).
In this example, log files 0, 5 and 10 were used.

```
X
CMDOK
<IDLE
```

```
SD>
V
uFAT: Init SD...
CSD:
          CSD: 1
          TAAC: E
          NSAC: 0
          TRAN_SPEED: 32
          CCC: 5B5
          READ_BL_LEN: 9
          READ_BL_PARTIAL: 0
          WRITE_BLK_MISALIGN: 0
          READ_BLK_MISALIGN: 0
          DSR_IMP: 0
          C_SIZE: 00003A8F (7675904 KB)
          ERASE_BLK_EN: 1
          SECTOR_SIZE: 7F
          WP_GRP_SIZE         : 0
          WP_GRP_ENABLE: 0
          R2W_FACTOR: 2
          WRITE_BL_LEN: 9
          WRITE_BL_PARTIAL: 0
          FILE_FORMAT_GRP: 0
          COPY: 0
          PERM_WRITE_PROTECT: 0
          TMP_WRITE_PROTECT: 0
          FILE_FORMAT: 0
          CRC: 43
SDSTAT:
          DAT_BUS_WIDTH: 00
          SECURED_MODE: 00
          SD_CARD_TYPE: 0000
          SIZE_OF_PROTECTED_AREA: 03000000
          SPEED_CLASS: 04 (class 10)
          PERFORMANCE_MOVE: 04
          AU_SIZE: 09 (4096KB)
          ERASE_SIZE: 0008
          ERASE_TIMEOUT: 02
          ERASE_OFFSET: 02
          UHS_SPEED_GRADE: 01
          VIDEO_SPEED_CLASS: 00
          VSC_AU_SIZE: 00
          SUS_ADDR: 00
          APP_PERF_CLASS: 00
          PERFORMANCE_ENHANCE: 00
          DISCARD_SUPPORT: 00
          FULE_SUPPORT: 00
uFAT: Reading MBR...
uFAT: Reading bootsect...
uFAT: key info:
          Card capacity: 15351808 sectors
          Partition start sector: 8192
          Partition capacity: 15343616 sectors
          FAT1 start sector: 8224
          FAT2 start sector: 0
          Cluster start sector: 10097
          Sectors/clusters: 64
          Root cluster: 2
          Number of data clusters: 239714
uFAT: id: E5 checksum: 35 (obtained: 35)
uFAT: numlogs: 14 startcluster: 128 sizecluster: 17024 sizebytes: 557842432
uFAT: Logs:
          LOG-3FF3.000 start sector: 18161 start cluster: 128 size: 0000BF06h
          LOG-3FF3.001 start sector: 1107697 start cluster: 17152 size: 00000000h
          LOG-3FF3.002 start sector: 2197233 start cluster: 34176 size: 00000000h
          LOG-3FF3.003 start sector: 3286769 start cluster: 51200 size: 00000000h
          LOG-3FF3.004 start sector: 4376305 start cluster: 68224 size: 00000000h
          LOG-3FF3.005 start sector: 5465841 start cluster: 85248 size: 0007794Ch
          LOG-3FF3.006 start sector: 6555377 start cluster: 102272 size: 00000000h
          LOG-3FF3.007 start sector: 7644913 start cluster: 119296 size: 00000000h
          LOG-3FF3.008 start sector: 8734449 start cluster: 136320 size: 00000000h
          LOG-3FF3.009 start sector: 9823985 start cluster: 153344 size: 00000000h
          LOG-3FF3.010 start sector: 10913521 start cluster: 170368 size: 00007ACDh
          LOG-3FF3.011 start sector: 12003057 start cluster: 187392 size: 00000000h
          LOG-3FF3.012 start sector: 13092593 start cluster: 204416 size: 00000000h
          LOG-3FF3.013 start sector: 14182129 start cluster: 221440 size: 00000000h
uFAT: initialised
CMDOK
```

Figure 10. SD card log files from BlueSense

An external card reader is required to access the content of the files on the SD card (figure 11). Note that the date of the files is not meaningful, but the time of the file corresponds to the file creation time (this requires the real time clock to be configured from the main mode with the T command).



| Name | Date modified | Type | Size |
|---|---|---|---|
| LOG-3FF3.000 | 08/09/1988 14:23 | 000 File | 48 KB |
| LOG-3FF3.001 | 08/09/1988 14:21 | 001 File | 0 KB |
| LOG-3FF3.002 | 08/09/1988 14:21 | 002 File | 0 KB |
| LOG-3FF3.003 | 08/09/1988 14:21 | 003 File | 0 KB |
| LOG-3FF3.004 | 08/09/1988 14:21 | 004 File | 0 KB |
| LOG-3FF3.005 | 08/09/1988 14:23 | 005 File | 479 KB |
| LOG-3FF3.006 | 08/09/1988 14:21 | 006 File | 0 KB |
| LOG-3FF3.007 | 08/09/1988 14:21 | 007 File | 0 KB |
| LOG-3FF3.008 | 08/09/1988 14:21 | 008 File | 0 KB |
| LOG-3FF3.009 | 08/09/1988 14:21 | 009 File | 0 KB |
| LOG-3FF3.010 | 08/09/1988 14:23 | 010 File | 31 KB |
| LOG-3FF3.011 | 08/09/1988 14:21 | 011 File | 0 KB |
| LOG-3FF3.012 | 08/09/1988 14:21 | 012 File | 0 KB |
| LOG-3FF3.013 | 08/09/1988 14:21 | 013 File | 0 KB |

Figure 11. SD card log files on desktop computer

### 5.3 Important precaution when reading the card on a computer

The FAT file system is a custom FAT optimised for speed and hence not fully compatible with desktop operating systems.
In particular, the SD card must only be read, never written to: in other words, immediately copy the files from the SD card to another location on your computer.
Most operating systems, however, write data to SD cards when connected to the computer (e.g. to store the recycle bin), irrespective of whether the user actively writes data to the SD card. For this reason, the SD card must be reformatted from BlueSense after every time it has been popped into a computer.
Always use the format function of BlueSense.

### 5.4 Note on the last log file

Do not use the last log file (e.g. 13 if formatted with 14 log files), as this is used to store debug info in some debug builds.

## *6. ADC streaming/logging*

BlueSense allows to acquire the data of the ADC inputs exposed on the expansion connector, and either stream over USB/Bluetooth or store the data to an SD-card.
The firmware has been optimised to allow high ADC sample rate acquisition and streaming/storage. Up to 1 KHz sampling can be achieved with low jitter (<50µs).
The ADC acquisition mode is entered with command `A,<mask>,<us>.`
With *mask* a bitmask (in decimal) indicating which ADC channel to acquire. BlueSense has ADC channels 0, 1, 2, 3 and 7 available (see extension connector in separate documentation). A mask of 130 means acquiring the channel 1 and channel 7.
All channels can be acquired (mask=255), but only the data of channels 0, 1, 2, 3 and 7 are mapped to the extension connector.
Channels are streamed with the lowest number channel first (leftmost) and the highest channel number last (rightmost).
Figure 12 illustrates data acquisition.
Logging can be initiated after starting the streaming mode with the command L, in the same way as in the motion sensing mode.

```
A,1,1000
mask: 01. period: 1000
CMDOK
<IDLE
ADC mode: period: 1000 mask: 01 binary: 0 timestamp: 0 battery: 0 label: 0
```

```
00531
00586
00669
00686
00604
00520
00412
00403
00408
00531
00616
00646
00629
00580
```

Figure 12. Streaming of ADC channel 0 (specified by the bitmask 1) acquired every 1000 µs.