BlueSense: Extensible Wearable Motion Sensing Platform and IoT platform
-
Quick start manual

Daniel Roggen

Wearable Technologies Lab
daniel.roggen@ieee.org

1 Revision history

– 13 January 2020: updated quickstart.

2 Bluesense description

BlueSense (figure 1) is a 30x30mm extensible wearable/IoT platform designed to be functional out-of-the-box yet extensible. It's primary purpose is to be an inertial measurement unit for wearable applications based on the Invensense MPU9250 and an AT1284p microcontroller. Yet it is extensible for IoT applications with a number of features making it appealing for this purpose.

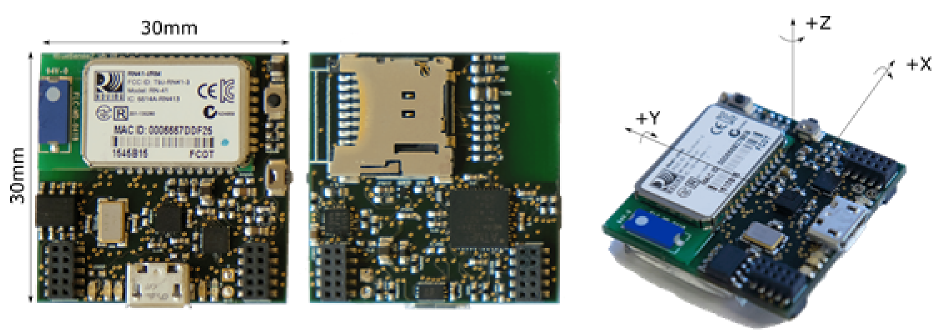


Fig. 1. BlueSense PCBs top (left) and bottom (middle), and BlueSense fitted with a 160mAh battery (right).

As a wearable platforn, BlueSense is a tiny device which can be used to capture body movement and orientation with a 9DoF motion sensor. The data can either be logged on an SD card or streamed over USB or Bluetooth 2.

BlueSense could be used to prototype a custom fitness tracker. Its extension ports allow to plug-in additional sensor modalities for research purposes. A display is being developed converting it into a smartwatch.

As an IoT device it's main appeal is a true hardware off which allows to put the device entirely to sleep (everything is powered down, including voltage regulators), yet wake up at programmed intervals thanks to a real-time clock wakeup. This allows to achieve very long battery life, as the device can wake up at desired times (once an hour, once a day, ...) to acquire and send sensor data. The extension ports allow to plug in custom sensor modalities which make this device highly versatile.

The application firmware has been designed to be useful for a wide range of applications out of the box. It has been optimised for high-speed motion data logging and streaming (500Hz), and high-speed external ADC acquisition (1+KHz), informed by the need of wearable motion tracking and activity recognition applications.

Besides USB, Bluetooth 2 and SD card interface with a FAT32 compatible filesystem, BlueSense has a real-time clock, a coulomb counter to measure it's own battery level and power consumption, true hardware off, and connectors for extensions. This makes the platform also suitable for sensor research, IoT applications, or as an compact alternative to other microcontroller boards.

3 Device description

BlueSense has 5 LEDs indicating status and a power switch (fig. 2). The LEDs, from left to right in the figure, are:

- Green logging/streaming indicator: when logging or streaming data this LED blinks, otherwise it is off.
- Standby indicator: when turning on the device, this LED blinks.
- Battery indicator: this LED is normally off. It starts to blink to indicate low battery voltage: the LED blinks 3 times for critically low battery, twice for extremely low battery, and once for low battery voltage. This LED also lights up when the power button is pressed, and blinks if the power button is continuously pressed for 10 seconds before the power is turned off.
- Blueooth connection indicator: this LED lights up when a Bluetooth connection is established.
- Charging indicator: this LED lights up when the battery is charging.

4 Setting up

The following explanations are illustrated for a Windows computer; a related process should be followed on Linux or Apple computers.

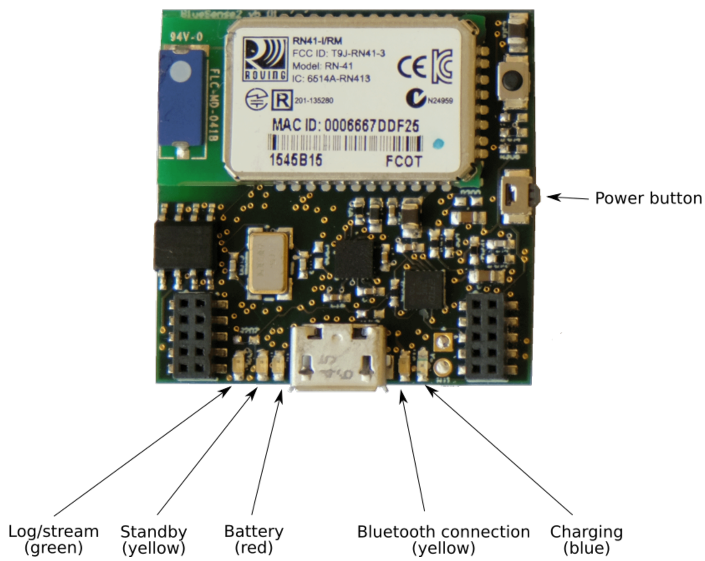


Fig. 2. BlueSense top view with LEDs and power button.

4.1 Terminal

BlueSense is controlled by sending commands over USB or Bluetooth. A terminal software 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)[1]
- TeraTerm 4.89 (or better)[2]

4.2 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.

USB When connecting the node over USB, the device manager shows the port number corresponding to the USB chip (COM11 in the figure 3). If the device appears as unrecognized, you must install the FTDI VCP drivers from `http://www.ftdichip.com/Drivers/VCP.htm`.

1 `http://www.compuphase.com/software_termite.htm`
2 `https://en.osdn.jp/projects/ttssh2/releases/`

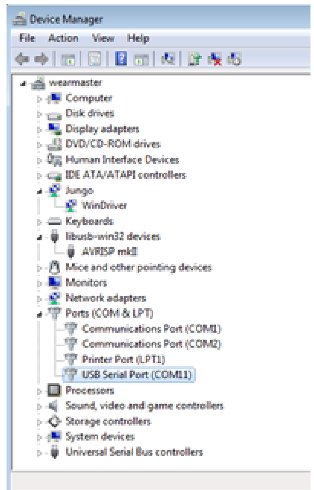


Fig. 3. Communication over USB occurs through COM11 on this computer.

Bluetooth For Bluetooth, several steps are required (figure 4):

- 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 (incoming and outgoing). The outgoing com port is the one to use to connect to BlueSense.

5 Turning the device on and off

5.1 Turning on

BlueSense can be turned on as follows:

- By pressing the power button.
- By connecting the device to a powered USB port.
- Programmatically, the device can be turned on by the real-time clock (see section ??).

5.2 Turning off

BlueSense can be turned on as follows:

- By pressing the power button for 10 seconds. The battery led will blink shortly before power is turned off.
- Programmatically, the device can be turned off by issuing the command ’O’ (see section ??).

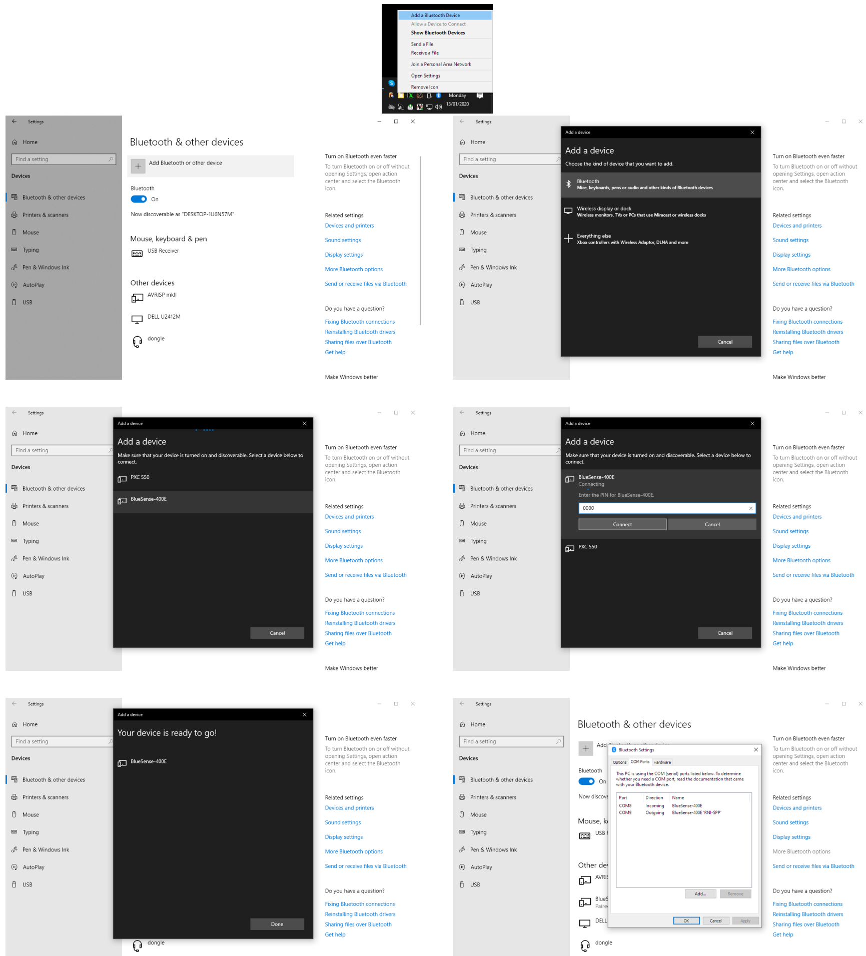


Fig. 4. Setting up Bluetooth connection to BlueSense.

6 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[3].

6.1 General principles

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

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

Commands are generally made of a single letter, optionally followed by a comma and additional parameters and terminated by a carriage return or new line character[4]. Generally, two commands are always available in any mode: a command `H` which provides help about the available commands, and a command `!` to leave the current mode and return to the main mode (unless BlueSense is already in the main mode).

Figure 5 illustrate the options available in the main mode when the user issues the help command (H).

```
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
```

Fig. 5. Output of the help command in the main mode.

6.2 Interface modes

Some commands have immediate effect, such as `T` to query or set the time. Other commands enter into a new mode, such as `M` which is used to acquire motion data.

When a command enters in a new mode a command `H` will generally be available to provide help on this mode. The command `!` can be used to leave a mode and return to the main mode.

[4] The terminal software used to interact with BlueSense must be configured to send a carriage return or new line when pressing the enter key to issue the command.

6.3 Command parameters

Several commands can take parameters. Parameters are specified after a comma following the command. The available parameters are documented by the help command.

6.4 Issuing multiple commands

Multiple commands can be entered on a single line by separating them by semicolons. For example issuing `T;D` will issue the command T (query time) followed by the command D (query date).

7 Key commands in the main mode

BlueSense is a research platform, and as such a large number of the commands included in the user interface are used for development or debugging purposes, while a smaller set of commands is designed for the end user.

In this manual we cover only the main "end user" commands. The other commands may be used but are recommended only for developers or users having a look at the firmware source code.

7.1 Time and date

BlueSense has a real-time clock (RTC) which is highly accurate (5ppm over the -40° to +85° temperature range, measured to be 0.6ppm at ambient temperature) and keeps time and date even when the device is powered off. The RTC clock is used to generate timestamps in the motion sensing and ADC sensing modes.

Figure 6 illustrates how to set the time (command `T,hhmmss`) and date (`D,ddmmyy`) or query the time and date (command `T` or `D` without parameters.

```
T,144736
Time: 14:47:36
CMDOK
D,130120
Date: 13.01.20
CMDOK
T
14:47:42
CMDOK
D
13.01.20
CMDOK
```

Fig. 6. Output of the some of the other commands in the main interface.

7.2 Battery information

BlueSense has a coulomb counter which measures battery voltage and current flowing in and out of the battery. The coulomb counter can be queried as illustrated in figure 7.

The command `q` returns the instantaneous battery voltage, instantaneous current flowing in or out of the battery (positive values indicate that the battery is charging) and instantaneous power consumption. The line underneath indicates the past 10 power consumption measurements, with each power measurement made every 10 seconds.

The command `Q` returns a longer history of the battery including timestamp, voltage, current and power. Measurements are made approximately every 3 minutes. The timestamp is the number of milliseconds since the beginning of the current month.

```
q
#V=4189 mV; I=21 mA; P=88 mW
91 91 91 93 90 90 90 90 90 88
CMDOK
Q
Battery info:
    T[ms]        mV   mA   mW
    1090951403   4189     0    0
    1091111000   4189     23   98
    1091261000   4189     22   93
    1091411000   4189     21   90
    1091561000   4189     21   88
    1091711000   4189     21   88
    1091861000   4189     20   87
    1092011000   4093     -5   -20
    1092161000   4061     -4   -17
    1092311000   4188     26   111
    1092461000   4189     22   95
    1092611000   4189     21   88
CMDOK
```

Fig. 7. The command `q` returns the instantanous battery voltage, current and power. The command `Q` returns a history of this information, measured every about 3 minutes.

7.3 Who am I

The command `!` is used to identify the node. BlueSense blinks its LEDs and prints the device ID. The device ID is made of the last 4 digits of the Bluetooth MAC address of the device. This may be useful to identify a device if the computer is connected to multiple BlueSense simultaneously.

8 Motion sensing mode and commands

8.1 Motion sensing modes

Multiple motion sensing modes are available (figure 9). Some are marked with a `x` indicating that they are experimental and the sample rate is not guaranteed.

```
?
My name is 835E
CMDOK
```

Fig. 8. Output of the some of the other commands in the main interface.

In many cases a 100Hz sample rate may be sufficient and the modes 19, 38, 44 and 45 may cover the needs of most applications.

8.2 Motion data streaming

Motion data streaming in a particular mode is initiated by `M,<mode>`.

Figure 10 shows streaming of acceleration, gyro and magnetic field in text mode after starting mode 33 (command `M,33`). Streaming can be terminated by issuing the `!` command.

8.3 Controlling the motion streaming format

The command `F` is used to specify the streaming format. Changes to the format are persistent across reboots. This command can be issued in the main mode, or in the motion mode.

The command `F` accepts the following parameters: `F,<bin>,<pktctr>,<ts>,<bat>,<label>`.

`bin` indicates whether to stream or log the data in binary format (when 1) or in text format (when 0). The other parameters indicate whether to stream a packet counter (`pktctr` when 1), timestamp (`ts` when 1), battery voltage (`bat` when 1), and label (`label` when 1).

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

Figure 11 shows motion streaming after the command `F,0,1,1,1,1`. In that example, the first column is the packet counter. The second column is the time of acquisition of the sample, in milliseconds, relative to the start of the current month. The third column is the battery voltage in mV. The fourth column is the current label (see section 8.4).

Note that the binary streaming format is not explained in this document as for most purposes it is not necessary.

8.4 Setting a label during sensing

It is sometimes necessary to indicate the start and end time of particular events being recorded. The command `N,<label>` which can be issued in the motion sensing mode sets a numeric label, which is streamed or logged alongside the motion sensor data (fourth column in figure 12).

```
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 TaitBryan/aerospace/zx'y", intrinsic (yaw, pitch, roll)
[46]      100Hz Quaternions debug (angle, x,y,z)
CMDOK
```

Fig. 9. Available motion sensing modes.

8.5 Motion sensor range and calibration

The motion sensor has different ranges for the accelerometer and gyroscopes, and the magnetic field sensor must be calibrated for optimal operation.

These settings can be changed in the “MPU test mode”. From the main mode, issue the command `m` to access the MPU settings (figure 13).

The recommended settings for human movement sensing are:

- Accelerometer range to +/- 16 G
- Gyroscope range to +/- 2000 dps
- User correction of the magnetic field.
- beta=0.35; only used in orientation sensing modes (e.g. with quaternion output). This refers to the beta time constant of Madgwick’s algorithm [1].

These settings can be set by the commands `L,3` (accelerometer range to +/-16 G), `l,3` (gyroscope range to +/-2000 dps), `G,3` (user correction of magnetic

```
M,33
CMDOK
<IDLE
SMPLMOTION>
Acc scale: 3
Gyro scale: 3
Sample rate: 50
 00374 -01084  01639 -00010  00004  00001 -00001 -00002  00001
 00379 -01093  01642 -00002 -00010  00003 -00001 -00002  00001
 00378 -01093  01636 -00004 -00016  00004 -00001 -00002  00001
 00381 -01096  01643 -00003 -00006  00003 -00001 -00002  00001
 00381 -01096  01630  00000 -00014  00004 -00001 -00002  00001
 00377 -01092  01641  00001  00004  00001 -00001 -00002  00001
 00377 -01092  01638  00001 -00015  00007 -00001 -00002  00001
 00381 -01093  01638 -00006 -00011  00002 -00001 -00002  00001
 00381 -01095  01643 -00007 -00009  00002 -00001 -00002  00001
 00378 -01099  01634  00001 -00026  00010 -00001 -00002  00001
 00379 -01088  01640  00000  00021 -00005 -00001 -00002  00001
 00378 -01094  01643  00000 -00010  00005 -00001 -00002  00001
 00379 -01093  01641 -00005 -00025  00009 -00001 -00002  00001
 00380 -01092  01640 -00008 -00007  00000 -00001 -00002  00001
 00379 -01097  01637 -00002 -00019  00007 -00001 -00002  00001
 00379 -01094  01637  00002 -00001  00002 -00001 -00002  00001
 00380 -01089  01644  00000  00006 -00001 -00001 -00002  00001
 00379 -01096  0!
1639 -00002 -00020  00008 -00001 -00002  00001
 00379 -01097  01638 -00005 -00015  00005 -00001 -00002  00001
 00380 -01091  01641 -00005 -00008  00001 -00001 -00002  00001
 00379 -01094  01642 -00002 -00013  00004 -00001 -00002  00001
 00378 -01096  01643  00000 -00003  00001 -00001 -00002  00001
 00382 -01094  01623 -00003 -00020  00006 -00001 -00002  00001
CMDOK
#t=469 ms; V=4189 mV; I=18 mA; P=78 mW; wps=2170; errbsy=0; errfull=0; errsend=0; spl=23; log=0 KB; logmax=0
       KB; logfull=4294967295 %
MPU acquisition statistics:
 MPU interrupts: 23
 Samples: 23
 Samples success: 23
 Errors: MPU I/O busy=0 buffer=0
 Buffer level: 0/64
 Spurious ISR: 0
MPU Geometry time: 0 us
Total errors: 0/23 samples (0 ppm). Err stream/log: 0, err MPU busy: 0, err buffer full: 0
<SMPLMOTION
IDLE>
```

Fig. 10. Streaming of acceleration, gyroscope and magnetometer data in text mode.

field) and `b,35`. These settings are persistent even if the device is powered off. The currently active settings can be queried with `L`, `l`, and `g` (see figure 14).

When using the user correction of teh magnetic field, a calibration routine must be performed by 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 15). All these settings and the user calibration are persistently stored.

It is recommended to perform a new magnetic calibration whenever the magnetic environment may change, such as for example: i) if an extension is plugged in on BlueSense; ii) if Bluesense is mounted on another device; iii) or if a long time has elapsed from the previous calibration (i.e. several days) or recordings are performed in a different environment.

Exit the MPU test mode with `!`.

```
M,33
CMDOK
<IDLE
SMPLMOTION>
Acc scale: 3
Gyro scale: 3
Sample rate: 50
0000000001 1092350443 04189 00000  00395 -01083  01630 -00016 -00023  00000 -00001 -00002  00001
0000000002 1092350462 04189 00000  00390 -01096  01638  00001 -00007  00003 -00001 -00002  00001
0000000003 1092350483 04189 00000  00393 -01091  01639  00004  00006 -00004 -00001 -00002  00001
0000000004 1092350502 04189 00000  00395 -01089  01645 -00004  00001 -00001 -00001 -00002  00001
0000000005 1092350523 04189 00000  00390 -01091  01639 -00008 -00028  00008 -00001 -00002  00001
0000000006 1092350542 04189 00000  00394 -01093  01626 -00005 -00033  00007 -00001 -00002  00001
0000000007 1092350563 04189 00000  00394 -01089  01643 -00003  00016 -00006 -00001 -00002  00001
0000000008 1092350582 04189 00000  00392 -01097  01635  00005 -00014  00006 -00001 -00002  00001
0000000009 1092350603 04189 00000  00388 -01088  01645  00002  00009 -00003 -00001 -00002  00001
```

Fig. 11. Streaming of acceleration, gyroscope and magnetometer data in text mode with additional metadata including packet counter, timestamp, battery voltage and label in columns 1 to 4.

8.6 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. This is obtained using an optimised implementation of Madgwick's algorithm [1]. In order to obtain accurate orientation, the device must not be moving for about 20 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 (figure 16). Quit the mode with `!`. The first column is Q_w followed by Q_x, Q_y, and Q_z.

8.7 Motion data logging

First, see section 9 on how to handle the SD card formatting.

There are two ways to log data:

- From the main mode, issue `M,<mode>,<log#>,<duration>`. This instructs to enter a specific motion sensing mode, log the data to the specified logfile number and terminate logging after the duration specified in seconds. If duration is omitted, logging continues until the command `!` is issued.
- 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 to the specified logfile number. Terminate logging with the command `!`.

9 SD card

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.

```
M,22
CMDOK
<IDLE
SMPLMOTION>
Acc scale: 3
Gyro scale: 3
Sample rate: 1
0000000001 1092937634 04189 00000  00383 -01093  01638 -00002 -00009  00002
0000000002 1092938633 04189 00000  00381 -01093  01639 -00002 -00010  00002
0000000003 1092939633 04189 00000  00383 -01093  01637 -00003 -00010  00003
0000000004 1092940633 04189 00000  00384 -01090  01642 -00002 -00010  00002
N,5
CMDOK
0000000005 1092941632 04189 00005  00385 -01096  01634 -00003 -00011  00002
0000000006 1092942632 04189 00005  00382 -01091  01637 -00003 -00011  00003
0000000007 1092943632 04189 00005  00382 -01094  01642 -00002 -00010  00003
0000000008 1092944631 04189 00005  00384 -01093  01639 -00001 -00005  00000
0000000009 1092945631 04189 00005  00381 -01090  01641 -00003 -00011  00004
N,10
CMDOK
0000000010 1092946631 04189 00010  00379 -01092  01643  00001 -00007  00004
0000000011 1092947630 04189 00010  00383 -01094  01639 -00003 -00010  00003
0000000012 1092948630 04189 00010  00380 -01095  01635 -00004 -00012  00003
0000000013 1092949629 04189 00010  00384 -01095  01638 -00004 -00011  00002
0000000014 1092950629 04189 00010  00384 -01093  01637 -00002 -00011  00003
N,0
CMDOK
0000000015 1092951629 04189 00000  00382 -01091  01638 -00003 -00011  00002
0000000016 1092952628 04189 00000  00383 -01095  01630 -00004 -00013  00003
```

Fig. 12. The label included in the motion stream (column 4) can be changed with the command `N`. This can be used to mark the start and end time of particular events.

9.1 Card compatibility

BlueSense only supports SDHC (not regular SD nor SDXC) cards with up to 32GB capacity.

9.2 Filesystem compatibility

Bluesense uses an optimised filesystem which is FAT32 compatible. This means that a computer can seamslessly read the data from the SD card.

However, 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. For this purpose, there is a fixed maximum number of log files of up to 14 (the exact number is selected during formatting). More information about how this filesystem has been implemented is provided in [2].

Consequently, if a computer writes to the SD card, the layout of the filesystem is modified and Bluesense is then unable to write to the SD card until it is formatted again. This may lead to silent errors.

It is therefore recommended to always format the SD card on Blusense prior to any data recording, if the card was previously connected to a computer.

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.

```
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
```

Fig. 13. MPU settings mode.

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 for this purpose.

9.3 Formatting

The SD card is formatted by entering the SD card mode (command `X`) and then issuing the format command: `F,<numlogfiles>` (figure 17).

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. In particular make sure the size of the log files is never 4GB or larger.

```
m
CMDOK
<IDLE
L
Acc scale: current=3 stored=3
CMDOK
l
Gyro scale: current=3 stored=3
CMDOK
g
Magnetic calibration mode: 2
CMDOK
!
CMDOK
IDLE>
```

Fig. 14. Querying the sensing range of the accelerometer and gyroscope and the magnetic field correction mode.

9.4 Reading the log files

The size of the logs can be obtained from the SD card mode (command X) with the command V (figure 18). In this example, log files 0, 5 and 10 were used.

9.5 Reading files on a computer

An external card reader is required to access the content of the files on the SD card (figure 19). 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).

9.6 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.

10 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 (¡50 us).

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 (decimal) means acquiring the channel 1 and channel 7.

```
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
```

Fig. 15. Magnetic field calibration routine.

Channels are streamed with the lowest number channel first (leftmost) and the highest channel number last (rightmost). Figure 20 illustrates data acquisition with a mask of 1 (i.e. acquisition of channel 0).

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. Therefore Bluesense would stream 8 channels of data, but only 5 of these are accessible from the external connector. Note that when streaming all the channels (mask=255) the channel 6 corresponds to an 1Hz square wave signal originating from the RTC. This may be used for debugging purposes. The value of the other channels are not meaningful.

10.1 Data format

Data format can be changed with the command `F,<bin>,<pktctr>,<ts>,<bat>,<label>`, as in the motion sensing mode.

10.2 Logging

Logging can be initiated after starting the streaming mode with the command `L`, issued from the ADC mode, in the same way as in the motion sensing mode.

```
M,44
CMDOK
<IDLE
SMPLMOTION>
Acc scale: 3
Gyro scale: 3
Sample rate: 100
 .9983  .0000 -.0001  .0000
 .9983  .0000 -.0002  .0000
 .9983  .0000 -.0001  .0001
 .9983  .0000  .0000  .0001
 .9983  .0000  .0000  .0001
 .9983  .0001  .0000  .0002
 .9983  .0002  .0000  .0002
 .9979 -.0244  .0030  .0131
 .9979 -.0244  .0029  .0131
 .9979 -.0245  .0027  .0132
```

Fig. 16. Quaternion data output.

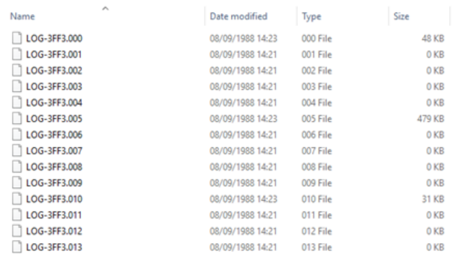
References

1. Madgwick, S., Harrison, A., Vaidyanathan, R.: Estimation of imu and marg orientation using a gradient descent algorithm. In: IEEE Int Conf on Rehabilitation Robotics. (2011)
2. Roggen, D., Pouryazdan, A., Ciliberto, M.: Bluesense - designing an extensible platform for wearable motion sensing, sensor research and iot applications. In: Proc. International Conference on Embedded Wireless Systems and Networks, ACM (2018) 177–178

```
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)
    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>
```

```
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
```

Fig. 18. SD card log files from BlueSense.



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

Fig. 19. Log files as seen on a computer when the SD card is plugged in a card reader.

```
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
```

Fig. 20. Streaming of ADC channel 0 (specified by the bitmask 1) acquired every 1000 us.