e-obs

GPS-Acceleration-Tags

Application Note:
How to use the acceleration sensor, interpret, analyse its data and how to get values in m/s^2.

Issued 2011-02-08

1 Introduction: What measures the acceleration sensor?

From http://en.wikipedia.org:
“In physics, acceleration is defined as the rate of change of velocity, or as the second derivative of position (with respect to time). It is then a vector quantity with dimension length/time². In SI units, acceleration is measured in meters/second² ($m\cdot s^{-2}$). The term "acceleration" generally refers to the change in instantaneous velocity.
In common speech, the term acceleration is only used for an increase in speed; a decrease in speed is called deceleration. In physics, any increase or decrease in speed is referred to as acceleration and similarly, motion in a circle at constant speed is also an acceleration, since the direction component of the velocity is changing. See also Newton's Laws of Motion.”

Further it is well known (since Einstein) that, without additional measures, it is not possible, to generally differentiate between earth's gravitation and actual acceleration. This has an important practical consequence for the acceleration sensor which is used somewhere in the gravitational field of our planet: There are always two phenomena causing changes of the ACC sensor output:

- Changing orientation of the device
- Accelerated translational movement of the device

Orientation:
Some will ask: what is meant with orientation? Orientation is a common term in physics to describe how a rigid object is orientated in space. This means that it is NOT important WHERE an object is located, but HOW. For example if the device is upside down or not, or for an animal if it is lying on the right side or on the left side (this means: right side is down or left side is down) or if the head is up and tail is down or head is down and tail is up.

Further orientation can mean if the animal is facing north, south, west or east etc., **BUT** it is important to understand that the acceleration sensor will not make a difference between north, south, west and east, of course. This means that orientation has an influence on the acceleration sensor only in terms of “up” and “down” in relation to earth.

Accelerated translational movement:
As stated above, the acceleration sensor also outputs a signal caused by accelerated translational movement of the device. The E-OBS tags can record the acceleration signal for up to 3 axes called X, Y, Z, each perpendicular to each other. It is important to understand that these axes are aligned fixed with the device. As the orientation isn't known (orientation isn't known, because acceleration sensor output is a combination of orientation and translational acceleration, even though orientation has a great influence), the direction of acceleration relative to earth isn't known. It is only known the direction of acceleration relative to the device.

Acceleration output format or data:
The E-OBS devices just record raw digital readings[1] of the ADC (analogue digital converter). These raw values can be or have to be converted to physical units, e.g. m/s^2, using formulas explained further below, if necessary. As every sensor and axis is a little different from the other, there is no single formula with fixed constants valid for all E-OBS devices and axes. Instead you have to test each device and each axis one time before putting it on the animal or after getting it back from the animal to determine the two constants of the formula for each axis and sensor. The formulas are simple of the linear type $y = m\cdot(x+b)$, further explained below. The two constants m and b are:

- m=slope
- b=x-intercept: this is the ADC output value for zero acceleration (another term for this is 'offset')

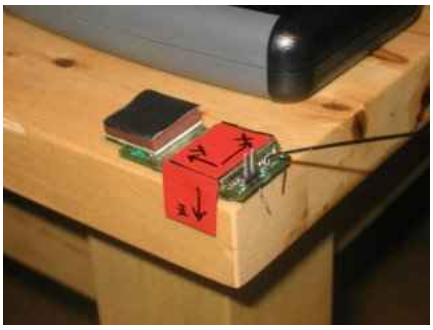
[1] The values are independent of the operating or battery voltage. Thus operating voltage doesn't influence accuracy and needn't be considered.

2 Example: Ideal ADC-readings of the acceleration sensor depending on orientation in earth's gravity field (high sensitivity setting)

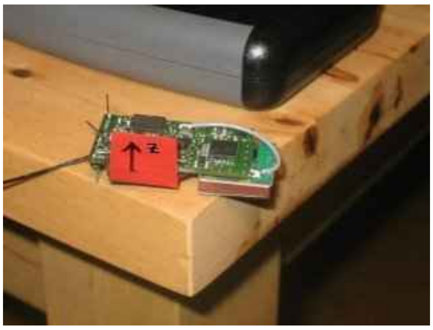
In the following photos the device is shown in different orientations relative to earth. The equations below the photos explain the ideal ADC-readings (x,y,z) (=raw values) and the acceleration (this equals -1 times the gravitation in this case) observed in the frame of reference of the device. “Ideal” means that the raw values shown for (x,y,z) are based on nominal values for slope and x-intercept. Further the (x,y,z) values shown here are valid for **high sensitivity setting.**

Remark: x,y,z don't form a right-handed cartesian coordinate system here, instead it is a left handed coordinate system.

In the following left photo you can see that x-axis and y-axis are parallel to earth's surface, so the observed accelerations a_x and a_y are zero. The z-axis points down, towards centre of earth, in the direction of gravitational force. Thus the sensor has the impression of being accelerated up, towards the sky, opposite the direction of z-axis. This means that a_z must be negative, thus $a_z =-g$. For the right photo it is right the other way round, thus $a_z = +g$.

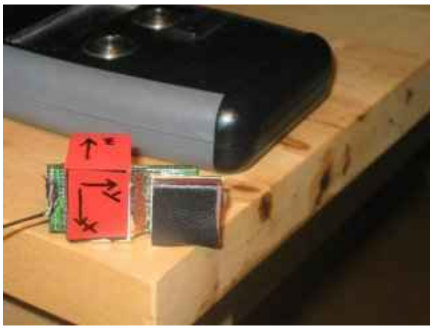


A: (x,y,z)=(2048, 2048, 1048)
$(a_x,a_y,a_z)=(0,0,-1)g$

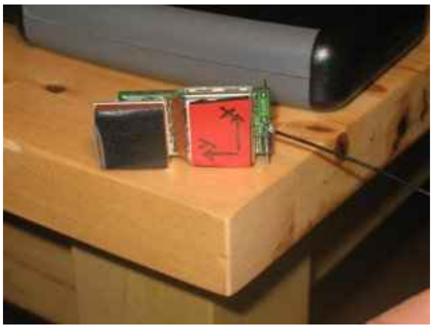


B: (x,y,z)=(2048, 2048, 3048)
$(a_x,a_y,a_z)=(0,0,+1)g$

The following photos aren't explained, this is left as an exercise for the reader.



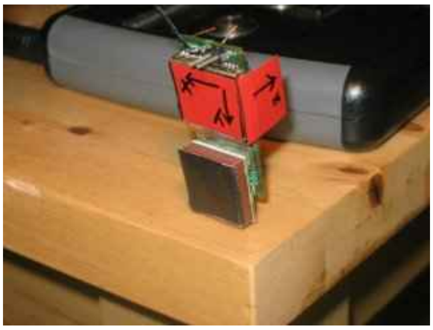
C: (x,y,z)=(1048, 2048, 2048)



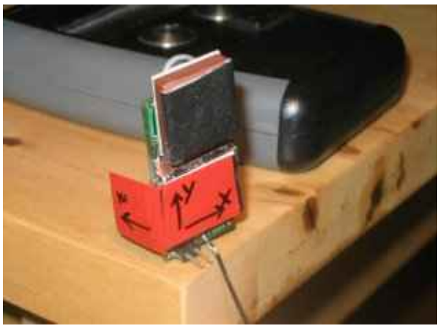
D: (x,y,z)=(3048, 2048, 2048)

$(a_x,a_y,a_z)=(-1,0,0)g$

$(a_x,a_y,a_z)=(+1,0,0)g$



E: $(x,y,z)=(2048, 1048, 2048)$
$(a_x,a_y,a_z)=(0,-1,0)g$



F: $(x,y,z)=(2048, 3048, 2048)$
$(a_x,a_y,a_z)=(0,+1,0)g$

The value “g” stands here for the magnitude of earth's acceleration $g=9.81m/s^2$. The whole system is not calibrated or adjusted, **so the real values might differ**. For the values given here it should be assumed that the angles formed by the PCB and an exactly horizontal plane are exact 0,90,180 degrees, etc.

3 Fundamentals of acceleration recording

3.1 ACC bursts

Lets start by comparing acceleration recording with the recording of GPS fixes. Usually, you switch on GPS for example every 5 minutes and then you get one GPS fix or you get several GPS fixes (GPS burst mode) and write them to memory. This sounds similar to Acceleration recording, but there is one important difference: Usually it doesn't make sense to just record one or two or three acceleration samples every 5 minute, because it is much more interesting to watch the dynamic and fast movements within short time. For this reason you probably want to get for example a minimum of 10 acceleration samples per second (BTW: 10 acceleration samples per second is the minium sampling rate that you can choose). But there is a problem: If you would record acceleration nonstop all day long at sample rate of 10 Hz (=10 samples per second), you would get about 1 Million samples per day, filling the onboard memory within eight days[2]. For this reason you can record acceleration in bursts. This means that you don't record nonstop, instead you record for example every 5 minutes (=setting ACC interval) for a burst of 5 seconds (=corresponds to setting ACC burst byte count). This extends memory lifetime for this example to 60 times, in this case roughly 480 days.

3.2 ACC samples over time and the sampling frequency

As already explained, acceleration is recorded at a certain sampling frequency[3] f in bursts. In this section we want to have a closer look at the timing of the acceleration samples within one burst. The sampling frequency f for one axis is important for this timing. Don't mix this up with the sampling frequency for all axes together. For example if you have three ACC axes enabled, then the sampling frequency for all axes together is three times the sampling frequency for one axis. The

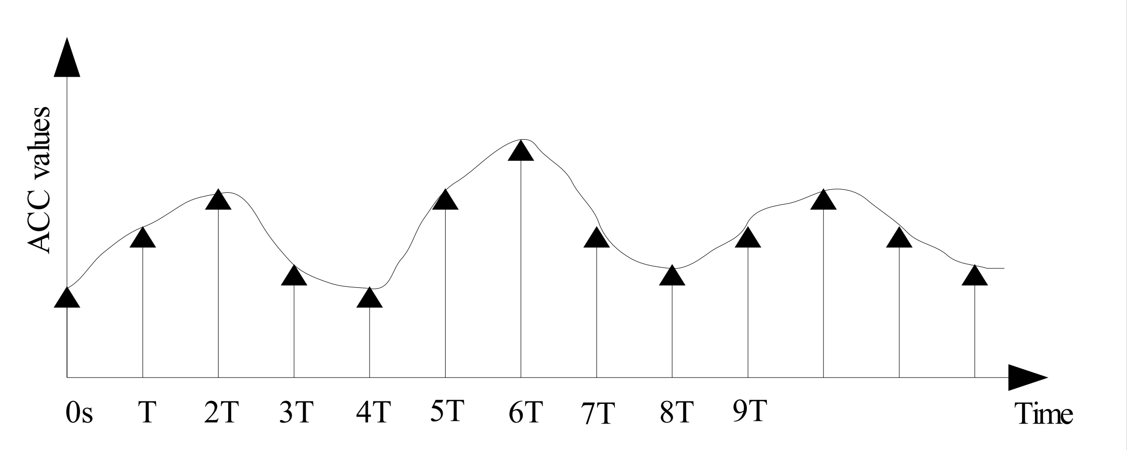
[2] Assuming 8 Megabyte. Additionally don't forget that you also want to download the data wirelessly and that the maximum download speed under best conditions is approximately one Megabyte per minute. This means that for the full memory you would need about 10 minutes under good conditions.

[3] The terms *sampling frequency* and *sampling rate* are the same. Please read more at http://en.wikipedia.org/wiki/Frequency

sampling frequency is measured in the unit *Hertz*, which is the same as 1/second. For example a sampling frequency f of 10 Hz means that you get 10 samples per second. The *period* T is the duration of the time between the acceleration samples:

$$T=\frac{1}{f}\Leftrightarrow f=\frac{1}{T}$$

If the sampling frequency is for example 10 Hz (which means 10 samples per second), then T=0.1s. Please see following Drawing 1 to understand the timing of the accleration samples. Time “0s” in this drawing is the beginning of the acceleration burst. The continuous curve is an example for the analogue acceleration signal of one axis over time. The 13 arrows depict 13 points in time (in practise the burst has more samples) at which the analogue acceleration signal is measured (resulting in *samples*), converted to digital format and stored to onboard memory.



Drawing 1: Timing of the sequence of acceleration values of one axis

In this context it should be mentioned that the first acceleration sample is not exact at the time of the acceleration timestamp (see section about the output format of data decoder). Especially for older e-obs-tags, there is a small delay (up to one second) between the timestamp and the first acceleration sample.

4 Acceleration data Output format of the data decoder

4.1 Common aspects of all output formats

Timing of the Acceleration samples

The e-obs-Tags don't record Acceleration Data nonstop[4]. Instead the acceleration data is recorded in *bursts*. One acceleration data burst is a relatively short time (e.g. 10 seconds) while the e-obs tag is recording acceleration data. With the acceleration interval in the e-obs-tag setup menu, you can set the amount of time between these bursts. If the acc interval for example is 30 minutes, then this means that every 30 minutes one burst of e.g. 10 seconds is recorded.

Three fundamental types of output format

Regarding the data formats, there are three fundamental possibilities: The first possibility is to have a data format that reflects the fact that acceleration data is recorded in bursts and not continuously. In this case one line represents one burst with for example 300 acceleration samples. The second

[4] Imagine if you would record acceleration data nonstop with a sample rate of for example 10Hz: Then you would get about 1.5 Mbyte of data per day! Within only 5 days, memory would be full (assuming 8 Mbyte of onboard nonvolatile memory).

possibility is to have a data format that neglects the existence of these bursts. In this case one line has one, two or three samples (for axes X,Y,Z) so that you just get continuous columns. The third possibilitiy is to process or transform (like Fourier transformation or variance or standard deviation calculation, etc.) the data before it is output by the data decoder.

All beginners should start using data decoder option "3. All recorded data in one file per logger". This will write the original raw values of the acceleration sensor to the file, one line per acceleration burst.

4.2 Sequence of acceleration samples

You know already that there are up to three acceleration axes: You can enable 1, 2 or 3 axes to be recorded. After downloading and decoding data using the data decoder, you get for example lines like this (option “All data in one file”):

```
ACC,99,11.6.2007,Mo,21:01:00,1530,1488,1170,1522,1498,1166,1528,1493,....
```

The acceleration values are printed here in bold letters. But what sample belongs to which axis X, Y or Z? Of course this question is very important for you. For this reason, the sequence of axes is well defined: The sequence of axes is determined by the alphabetical order of the enabled axes. The following table Table 1 shows the result of this rule:

Number of enabled axes	First enabled axis red	Second enabled axis green	Third enabled axis blue
3	X	Y	Z
2	X	Y	
2	X	Z	
2	Y	Z	
1	X		
1	Y		
1	Z		

Table 1: Sequence of acceleration axes

If you use the e-obs visualization tool to plot the acceleration curves, the axes are differentiated by the colors as shown in Table 1 above.

4.3 Example output formats

Example for data decoder option “All data in one file”: 3 axes XYZ

```
                               X    Y    Z    X    Y    Z    X    Y
ACC,99,11.6.2007,Mo,21:01:00,1530,1488,1170,1522,1498,1166,1528,1493,....
```

Example for data decoder option “All data in one file”: 2 axes XZ

```
                               X    Z    X    Z    X    Z    X    Z
ACC,99,11.6.2007,Mo,21:01:00,1530,1488,1170,1522,1498,1166,1528,1493,....
```

Example for data decoder option
"9.Acceleration data reduced to one average and one stddev value (after hp filter)"

One line is one burst:

Day number, Date, day of week, time of day, average value, standard deviation

```
1185,1.6.2010,Tu,18:51:04,1872,27.906608
1185,1.6.2010,Tu,19:01:39,1950,81.045572
1185,1.6.2010,Tu,19:20:49,1724,109.841046
1185,1.6.2010,Tu,19:40:27,1467,91.403437
```

Example for data decoder option "Acceleration data in columns":
This means that you don't get one line for each burst. Instead in every line there is only one sample (or e.g. 3 samples XYZ if you enabled all 3 axes). This is useful, if you want to plot acceleration in Open Office Spreadsheet or Mircosoft Excel without considering that in reality there are bursts. The time of day is the same for all samples of one burst. Although the format doesn't reflect the existence of bursts, it is still possible to recognize the bursts by looking on the time stamps.

```
day number, date, day of week, time of day, sample
1185,1.6.2010,Tu,18:51:04,1867
1185,1.6.2010,Tu,18:51:04,1867
1185,1.6.2010,Tu,18:51:04,1868
1185,1.6.2010,Tu,18:51:04,1868
1185,1.6.2010,Tu,18:51:04,1869
```

Example for data decoder option "DFT":
This option doesn't output directly the acceleration data. Instead a DFT is calculated for each burst separately. Then the result of each DFT is written, one line represents the DFT of one burst.

```
ACCF, day number, date, day of week, time of day, axis, number of DFT Values,standard deviation, DFT-Values........
ACCF,1185,1.6.2010,Tu,18:51:04,Z,166,27.907,1872,2,1,3,3,2,1,2,2,2,2,0,0,0,1,1,0,2,2,3,3,3,2,3,2,2,1,0,0,1,1....
```

4.4 Movebank Format

The Movebank Format (Option 'm') was implemented for http://www.movebank.org . But of course you can also use the Movebank format without using Movebank if you like this format more than the other available output formats of the data decoder. Here we just describe the output format of the data decoder to be used as input for Movebank. We don't describe the output format of Movebank. Please ask Movebank if you want to know more about the output format of Movebank.

In the first line of the generated tagXXX_acc.txt file is the description of the contents of the following lines:

```
key-bin-checksum,tag-serial-number,start-timestamp,acceleration-sampling-frequency-per-axis,acceleration-axes,accelerations-raw
```

This first line is used by the Movebank-Software to automatically determine the meaning of the different comma separated values in the following lines.

`key-bin-checksum:`
This is a checksum of the original binary data, so that Movebank can quickly compare different lines by comparing their checksums.

`tag-serial-number:`
This is the tag id. This number usually always is identical with the number in the file name.

`start-timestamp:`
This is the date and time of day when the acceleration burst belonging to this line begins. The exact time of the first sample is a little later (up to 1 second) for older e-obs tags, especially when the ACC-pinger is enabled (4 pings before the start of each ACC burst).

`acceleration-sampling-frequency-per-axis:`
This is the sampling frequency for one axis. Don't mix this up with the sampling frequency for all axes together. For example if you have three ACC axes enabled, then the sampling frequency for all axes together is three times the sampling frequency for one axis. The sampling frequency is

measured in Hertz, which is the same as 1/second. For example a sampling frequency of 10 Hz means that you get 10 samples per second.

`acceleration-axes:`
This tells you about the enabled ACC axes, like X and/or Y and/or Z.

`accelerations-raw:`
Acceleration along the X and or Y and or Z axes of the tag, depending on which axes were activated on the tag as described in the field "eobs:acceleration-axes". Measurements alternate one measurement for each active axis in alphabetical order. The values are digital readings between 0 and 4095 of the analogue digital converter on the tag, and can be converted to m/s² with proper calibration. These samples are made at the rate described by the field "`acceleration-sampling-frequency-per-axis`" starting with the first sample at the time described in the field "`start-timestamp`".

5 Formulas for high and low sensitivity:

$$a_X = (n_X - n_{X,zerog}) \cdot c_X \cdot g$$
$$a_Y = (n_Y - n_{Y,zerog}) \cdot c_Y \cdot g$$
$$a_Z = (n_Z - n_{Z,zerog}) \cdot c_Z \cdot g$$

Variable	Name	Value	Remark
c_i	slope, high sensitivity	0.001	Must be calibrated/tested/determined/verified for very precise requirements
c_i	slope, low sensitivity	0.0027	Must be calibrated/tested/determined/verified for very precise requirements
$n_{i,zerog}$	X-intercept (or “offset” or “raw value for zero acceleration/gravitation”)	Nominal 2048	Must be calibrated/tested/determined/verified in many cases if orientation is of interest; Needn't be tested if just relative changes are of interest, e.g. for activity measurements
a_i	Acceleration of one axis		Result using physical units m/s^2
i	Axis	x, y or z	Stands for the name of one of the three axes X,Y,Z
g	Magnitude of observed acceleration caused by earth gravitation	$9.81\frac{m}{s^2}$	
n_i	One digital sample of raw data for one of the three axes	0...4095	

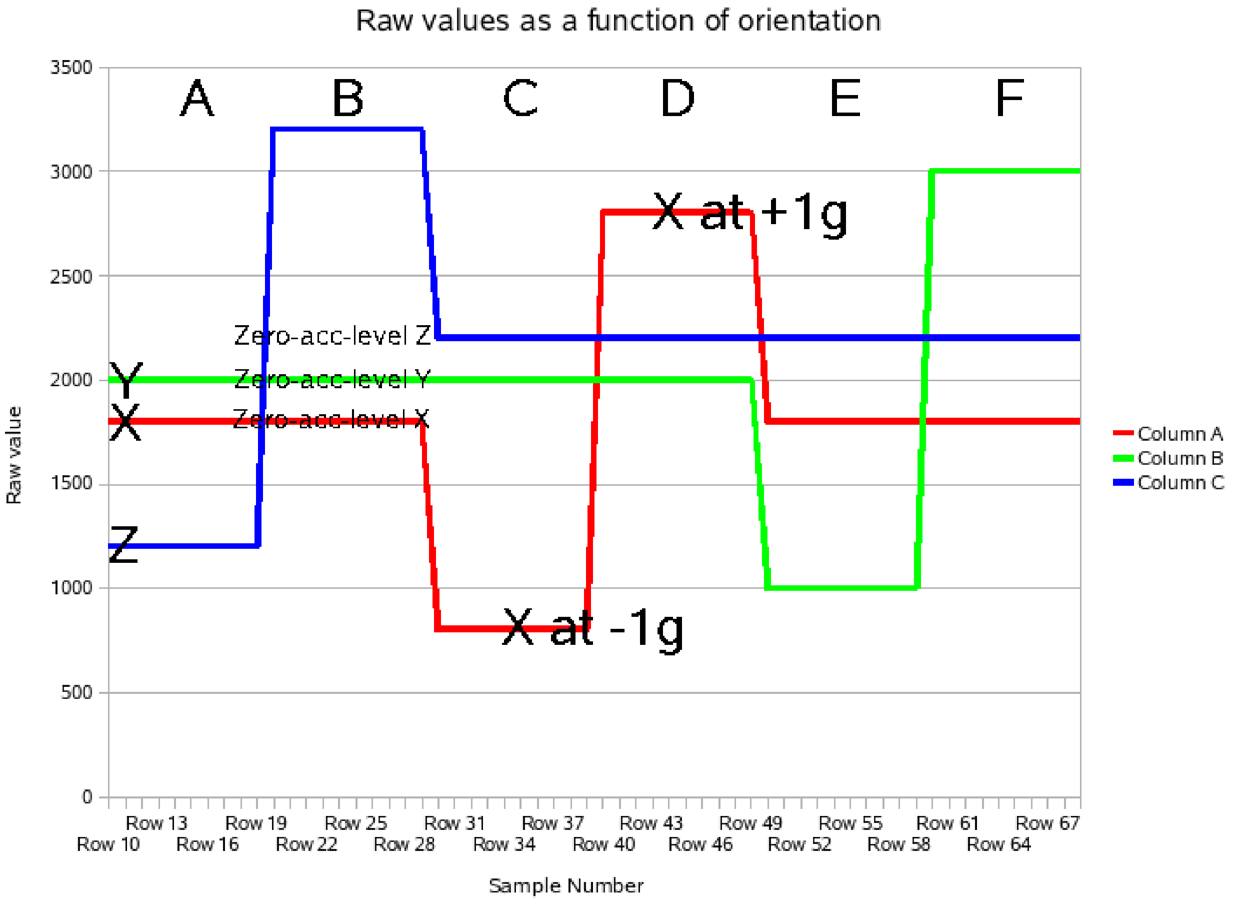
Remark: x,y,z don't form a right-handed cartesian coordinate system, instead it is a left handed coordinate system.

6 How to determine or test the constants in the formula

I explain here how to calibrate/test/determine/verify the constants $n_{i,zerog}$ and c_i in the formulas above. These constants are different for each axis *i*. One possibility to determine these constants is following:

- Configure the device for recording (e.g. record ACC for about 5 seconds every 10 seconds) acceleration
- Start recording
- Put the device on a table sequentially in six different, exactly defined orientations A, B, C, D, E, F in exactly the same sequence as shown in the photos in the example above.
- Download the data
- Analyse the data using your favourite software

The result will look like the following:



Then for example for the X-axis you can get the constants in the formula as described following: The X-axis in this graph is the red line. We know that for orientations A, B, E, F there cannot be any acceleration in the direction of the X-axis, because the X-axis is always parallel to earth's surface, so this is one possibility to get the raw value (=ADC-reading) for zero gravitation/acceleration: Just read the value from the graph: In our case this value seems to be about $n_{X,zerog}=1800$. But we also want to determine the slope factor in the formula, and this can be combined with determination of the zero level: In the graph read the value of X for orientations C and D (this corresponds to full negative earth gravitation/acceleration and full positive earth gravitation/acceleration). In our case we get about $n_{X,-1g}=800$ and $n_{X,+1g}=2800$. It is clear that we can also get the zero level from these two values as the arithmetic mean of both:

$$n_{X,zerog}=\frac{n_{X,-1g}+n_{X,+1g}}{2}=\frac{800+2800}{2}=1800$$

The slope factor c_X can be calculated as follows:

$$c_X = \frac{2}{n_{X,+1g} - n_{X,-1g}} = \frac{2}{2800-800} = 0.001$$

Then the formula for the X-axis is finished:

$$a_X = (n_X - n_{X,zerog}) \cdot c_X \cdot g = (n_X - 1800) \cdot 0.001 \cdot g$$

These calculations have to be repeated for the other axes.

7 Possible misinterpretation of acceleration data

- It is very easy to get very high observed acceleration values, e.g. 10g or 20g. This is physically correct, but normally these accelerations occur only for very short time. This happens for example when knocking or tipping with a rigid object on the E-OBS device. This means that before using acceleration data to measure exact values of acceleration, it maybe required to filter the data with a low pass filter.
- Don't forget that there is no anti aliasing filter in front of the ADC. This means that the sample rate always must be chosen high enough to avoid aliasing. Otherwise you might observe signals in your analysis that in reality are different.
- For the sake of completeness, it should be noted here, that the acceleration sensor has a certain frequency response (3dB-frequencies between 150Hz and 300Hz). But this is not relevant for most of the biologists.

8 How to set “ACC BYTE COUNT”

In the setup/configuration menu of the E-OBS tag, the number of bytes to be used for one burst of acceleration data must be entered. Due to the limited resources on the tag, we didn't implement at this point an easy user interface. Thus some users might misunderstand the meaning of the bytecount. So we explain this here in more detail. In reality, always a multiple of 64 bytes is used on the onboard memory. Further some bytes are needed for housekeeping. This means that you should look in the following table if you want to know exactly how much memory will be used and how many samples you will get:

GPS-ACC tag version	Your setting ACC BYTE COUNT in setup-menu:	Number of available acceleration data bytes	Number of acceleration samples for all ACC-axes together	Onboard memory used in number of bytes
All versions	0	0	0	0
All versions	1-63	54	36	64
All versions	64-126	117	78	128
All versions	127-189	180	120	192
All versions	190-252	243	162	256
All versions	253-315	306	204	320
All versions	316-378	369	246	384
All versions	379-441	432	288	448
All versions	442-504	495	330	512
All versions	505-567	558	372	576
All versions	568-630	621	414	640
All versions	631-693	684	456	704
Up to software version rv4_s1	694-65535	747	498	768
≥ rv5	694-756	747	498	768
≥ rv5	757-819	810	540	832
≥ rv5	820-882	873	582	896
≥ rv5	883-945	936	624	960
≥ rv5	946-1008	999	666	1024
≥ rv5	1009-1071	1062	708	1088
≥ rv5	1072-1134	1125	750	1152
≥ rv5	1135-65535	1188	792	1216

Example: Setting the byte count to 800 will result in 747 Bytes of data per burst for GPS-Tag software versions up to rv4_s1. Including the overhead, 768 Bytes of memory will be used. Since each individual sample point needs 1.5Bytes, a total of 498 data samples will be recorded. Let’s further assume that 2 axes are active, then 249 data samples will be recorded per axis.

9 Some more general hints and hints on continuous sampling

It is important to understand that originally we didn't have continuous acceleration sampling in mind when we designed the devices. So this means that the original intention was that the devices record acceleration e.g. every 5 minutes for a few seconds or e.g. every 2 minutes etc.

The period (e.g. 5 minutes or 2 minutes) we called "**ACC INTERVAL**".

The short time of ACC recording we called "burst" as this is a common technical term describing fast repeated events (in this case these events are ACC measurements with e.g. 100 Hz sampling rate) within a short span of time.

You cannot directly set the length of a burst (e.g. a few seconds) in the configuration menu. Instead you can only adjust the amount of memory (**ACC BYTECOUNT**) needed for one burst. This makes it easy for the user to know how much memory will be consumed. On the other hand the user has to calculate the length of the burst in seconds if he is interested.

The maximum of **ACC BYTECOUNT** is 756 for older devices (see table above) and 1197 for newer devices. The devices also will operate correctly if you set **ACC BYTECOUNT** to higher values, but this effectively will not increase the burst length higher than as explained here.

The sampling rate only sets the frequency of sampling within one burst.

If you want (nearly) continuous sampling, you can achieve this indirectly: Set the **ACC INTERVAL** to a smaller value than the length of one burst.

Example for **continuous** sampling for newer devices:

1. Set the total samplerate to 100 Hz (33 Hz per axis)

2. Set total ACC BYTECOUNT to 1197

3. Set ACC INTERVAL to 1 second

4. If it is of interest for you: Read the number of samples in the table above: This will be 792. And calculate length of burst in seconds: 792/(100Hz)=7.92 seconds

Example for **continuous** sampling for older devices:

1. Set the total samplerate to 100 Hz (33 Hz per axis)

2. Set total ACC BYTECOUNT to 756

3. Set ACC INTERVAL to 1 second

4. If it is of interest for you: Read the number of samples in the table above: This will be 498. And calculate length of burst in seconds: 498/(100Hz)=4.98 seconds

10 Using the Fourier Transformation

10.1 Formulas for the transformation of one acceleration axis

The formula for the time discrete Fourier Transformation in general is:

$$c_n=\sum_{k=0}^{N-1} e^{\frac{-2\pi\cdot i n\cdot k}{N}} a_k \quad \text{with} \quad n\in\mathbb{N}_0 \quad , \quad 0\leq n\leq N-1 \quad \text{and the imaginary unit i}$$

In our case the a_k are the N acceleration samples of one axis with $0\leq k\leq N-1$. This formula gives as result N DFT values which are complex numbers c_n with $0\leq n\leq N-1$. Half of the DFT values c_n are redundant, further we would like to get results more independent from the original number of acceleration samples, further many people don't like complex numbers (please tell me if you really need complex numbers, it would be no problem to implement this in the data decoder). This is the reason why we implemented in the data decoder a division by N and why we calculate the absolute of the complex numbers c_n , so you get from the data decoder following real numbers r_n as DFT values:

$$r_n=\frac{1}{N}\left|\sum_{k=0}^{N-1} e^{\frac{-2\pi\cdot i n\cdot k}{N}} a_k\right| \quad \text{with} \quad n\in\mathbb{N}_0 \quad \text{and} \quad 0\leq n\leq\frac{N}{2}$$

This calculation is done by the data decoder separately for every axis and for every block of acceleration data. For example lets assume that the ACC INTERVAL is 900 seconds = 15 minutes and that three axes are enabled. Then you get for every 15 minutes three transformations, one for each axis.

10.2 The number of DFT values

As described above, the number of original ACC samples (for one axis) is N .

The data decoder will give you the real numbers r_n for $0\leq n\leq\frac{N}{2}$ as DFT values (remark: n is an integer number). For example if $N=6$ you will get four real numbers r_0, r_1, r_2, r_3 as result for the transformation because $\frac{N}{2}=3$. And if $N=7$ you will also get only four real numbers r_0, r_1, r_2, r_3 , because 7/2=3.5 is not an integer number so that the biggest valid integer number for n is three.

Short: If you want to know the number of DFT values *for one axis* then calculate $\frac{N}{2}+1$ whereby N is the number of original ACC samples *for one axis*. If this is not an integer number, then round down.

10.3 The frequencies

It's the essence of the Fourier Transformation, that each DFT value corresponds to a certain frequency. The relation between a DFT value and its corresponding frequency is following:

$$r_n \text{ belongs to frequency } f_n=\frac{n}{N}\cdot f_S$$

whereby (important!) f_S is the sampling frequency *for one axis* and N is the number of original ACC samples *for one axis.*

11 Output of data decoder for option “Standard deviation after HP-Filter”

Using the datadecoder, you have the possibility to generate output files which tell you the standard deviation of each axis for each block after a Highpass filter. This means that the data decoder at first sends the data separately for each axis through a low pass filter (Bessel filter 10th order, forward-backward algorithm), then substracts the output from the original unfiltered data (=removing the low frequency components, so that only the high frequency components are left), and finally the decoder calculates the standard deviation, whereas the standard deviation is defined as follows:

$$stddev_{acc}=\sqrt{\frac{1}{N}\sum_{k=1}^{N}(a_k-\bar{a})^2}\ \text{ whereas the arithmetic mean }\ \bar{a}\ \text{ is defined as }\ \bar{a}=\frac{1}{N}\sum_{k=1}^{N}a_k\ .$$

(The differences and commons of population standard deviation, sample standard deviation and squareroot of second central moment, mean error etc. are neglected here because it's not important here). The a_k are the N raw acceleration values from one axis recorded on the data logger with $1\leq k\leq N$ (“raw” means that these values are the raw readings coming from the Analogue-Digital-Converter, without being transformed to a physical meaningful dimension measured in m/s^2. Please read the rest of the application note so that you understand what this means). This calculation is done for each axis separetely.