To be published in Proc. IEEE Workshop on Signal Processing Systems (SiPS), 2016, Dallas, Oct. 26-28.

MEMS Inertial Motion Sensing Watch for Measuring Walking and Running Activities

F. Bardyn, M. Savary, S. Grassi, P-A. Farine
Electronics and Signal Processing Laboratory
Ecole Polytechnique Fédérale de Lausanne (EPFL)
2002 Neuchâtel, Switzerland

B. Fasel, K. Aminian
Laboratory of Movement Analysis and Measurement
Ecole Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland

Abstract — **This paper describes the development of a sport watch, as well as the software and hardware required to design an integrated solution for getting feedback during and after walking and running activities. The developed system is autonomous, so that neither a smartphone nor a computer are necessary for configuration. The user controls the watch via a touch-screen based interface and all the relevant information is displayed on an LCD. Moreover, the watch contains all the required inertial sensors to allow giving different feedbacks to the user, such as the type of activity, the number of steps taken, the covered distance, the instantaneous speed and the average speed. Although this work is multidisciplinary, in this paper the emphasis is put on the description of the device and on the electronics and the algorithms implementation. Particular focus is on the efficiency in terms of computational time, and in achieving high autonomy on a primary cell battery.**

Keywords — Inertial sensor; accelerometer; wrist watch; pedometer; step counter; distance estimation; speed estimation

I. INTRODUCTION

Wrist-worn devices such as activity trackers and smartwatches have gained interest and popularity over the last few years for monitoring physical activities.

The aim of the work described in this paper was to produce an analog watch with sport functions. This sport watch should be able to give relevant feedback to walkers and runners with high accuracy. To ensure a long battery life, only low power MEMS 3D accelerometers and a barometric pressure sensor were used. Although many different solutions exist currently in the market, none of them have at the same time a high accuracy and a long battery life. Indeed, most of the activity trackers with high autonomy use very simple algorithms to count steps or estimate distance and speed, which leads to poor accuracy. Some devices increase accuracy by adding high consumption sensors such as global navigation satellite systems (GNSS) or a continuous wireless connection to a smartphone. However, these devices have a significantly reduced battery life and require a rechargeable battery.

This paper focuses on the description of the device development and on the electronics and algorithm implementation. It is organized as follows. The device is described in Section II. The hardware (HW) and software (SW) architecture is presented in Section III as well as the acquisition of representative data sets of signals measured during physical activities, the data processing algorithms and the development methods and tools. Testing based on these data sets, field-test validation and power consumption estimation are given in Section IV. Finally, conclusion and future work are given in Section V.

II. DEVICE DESCRIPTION

The sport watch works standalone, so that the use of a computer or a smartphone is not necessary. The only setup needed from the user is entering his/her height. This is done via the user interface consisting of a touch-screen and four pushbuttons. Fig. 1 illustrates the sport watch prototype and the different parts of the user interface.

Fig. 1. The sport watch prototype and its user interface with the 7 sensitive zones of the touch-screen delimited by dashed white lines

When the user is not moving, the device behaves as a standard analog watch. But when the user does some movement, the watch enters a state of activity determination to decide if the movement is walking, running or other (parasitic) movement. If a walking or running activity is detected, the sport functions are activated and the user begins to get feedback. If the movements are parasitic, the watch stays in the state of activity determination, until a walking/running activity is detected or the user's movement stop.

A. The watch hardware

The prototype of the sport watch (see Fig. 1) has the following characteristics:

- 45 mm diameter and 13 mm height,

- 3 V primary cell Lithium battery (150 mAh),
- LCD display with backlight,
- touch-screen with 7 sensitive zones,
- 2 watch hands and 4 pushbuttons.

There are two microcontrollers, several sensors and a flash memory:

- Proprietary ultra-low power 8-bit microcontroller, handling the watch functions and the user interface parts such as the LCD display, the touch-screen, the pushbuttons and the motor for the two watch hands.
- Low power microcontroller (Kinetis KL16 from NXP [1], based on ARM Cortex M0+ processor [2]) to manage the data acquired by the sensors, the FLASH memory and the communication with the 8-bit microcontroller. This unit is also used to process the sensor data in real time for continuous activity monitoring.
- Two 3D accelerometers with different characteristics (BMA280 [3] and ADXL362 [4]). One pressure sensor (BMP280 [5]).
- Flash NOR memory (128 Mbit) [6] for logging the acquired data and the results.

The Fig. 2 shows the exploded view of the sport watch. Two PCBs are stacked onto each other. The bottom PCB contains the 8-bit microcontroller and all the elements required to handle the watch functions and the user interface. The top PCB is used to manage the sport functions. This PCB contains the KL16 microcontroller, the two accelerometers, the pressure sensor and the flash memory. The two PCBs are linked with a flexible connector allowing communication between the two microcontrollers using an SPI interface.

B. *The watch functions*

The following parameters can be displayed by the watch:

- the number of steps taken,
- the covered distance,
- the instantaneous speed (only during an activity),
- the average speed.

These values are updated at every second during the sport activity. They are automatically reset at midnight or via the user interface if required. Additionally, the number of steps and the distance measured the day before are kept in memory and can be consulted by touching 2 times the steps or distance zone of the touch-screen.

The prototype has also the ability to record data and results into the flash memory. This is a start/stop functionality, such that the user can choose when to start and stop logging data. When the data logging is requested, several values are continuously stored in the flash memory, until the data logging is stopped:

- the raw data from the 3D accelerometer,
- the raw data from the pressure sensor,
- the number of steps since the start of the logging,
- the distance since the start of the logging,
- the instantaneous speed,
- the average speed.

The data logging functionality was implemented for testing purposes only. It is useful for development of the product and has its place in a prototype, but its power consumption is too high to include it in a primary battery powered product. The logged data can be transferred to a computer via a UART serial interface that is accessible when the watch is connected to a proprietary development board (see Fig. 6 in section III.E).

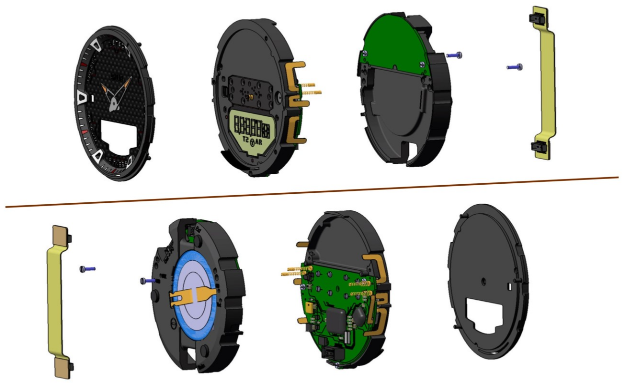

Fig. 2. Exploded view of the sport watch. There are two PCBs, one for handling the standard watch functions and one to manage the sport functions

Apart from the sport watch functions described above, the watch has also the features of a classical watch, such as time and date keeping, chronometer, and alarm.

III. Implementation

A. *Hardware architecture*

The Fig. 3 illustrates the hardware block diagram of the complete system used for the sport functions. The main component is the KL16 microcontroller, which communicates with all the others components and processes the acquired sensor data. An I^2C interface is used to communicate with the pressure sensor, one SPI interface to connect to the 8-bit microcontroller, and another SPI interface is shared to communicate with both the accelerometer and the flash memory using a logic gate not shown in the figure. The Fig. 3 also shows the UART interface used to transfer the logged data when the watch is connected to the development board.

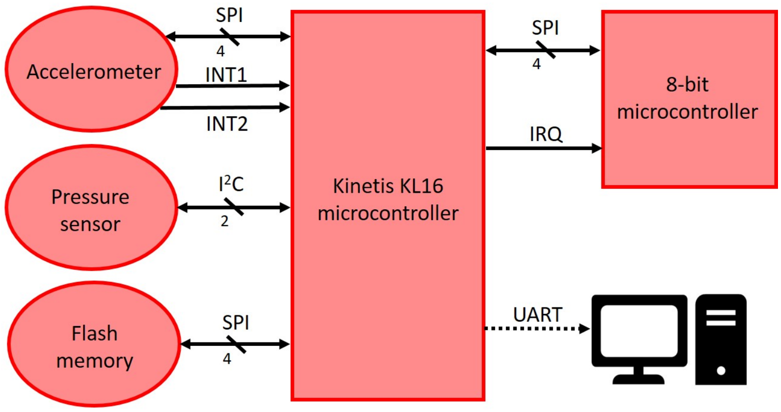

Fig. 3. Sport watch HW block diagram

The 8-bit microcontroller is the SPI master and sends commands to the KL16 microcontroller via the 4-line SPI bus. The KL16 microcontroller is the SPI slave and cannot initiate data transfers, but it can interrupt the 8-bit microcontroller via the IRQ line to request communication.

The accelerometer has two lines to interrupt the KL16 microcontroller. One is used when a movement is detected and the other during data acquisition to tell the KL16 microcontroller that the accelerometer data are ready.

B. *Software architecture*

The software architecture is shown in Fig. 4 and explained as follows. The initialization is performed once at power up: the KL16 microcontroller power modes are configured, the GPIO ports and interrupts are enabled, the communication protocols are set and the flash memory is activated. After initialization, the KL16 microcontroller is in very-low power sleep mode waiting for an interrupt from the accelerometer or the 8-bit microcontroller to occur. The accelerometer and the pressure sensor are also in sleep mode.

Three types of events can wake-up the KL16 microcontroller: the interrupt of the 8-bit microcontroller asking for communication and the two interrupts from the accelerometer if movement is detected or if the data FIFO is full. Upon an interrupt from the 8-bit microcontroller, the KL16 microcontroller quits the very-low power sleep mode, receives or sends a value to the 8-bit microcontroller, and returns to very-low power sleep mode until the next interrupt. This is used to transfer data such as the number of steps, the distance and the speed from the KL16 microcontroller to the 8-bit microcontroller in order to display the values on the watch LCD. It is also used to send the user height from the 8-bit microcontroller to the KL16 microcontroller, to be taken into account in the algorithms for distance estimation (see Section III.D). On the other hand, when an interrupt comes from the accelerometer, the complete data acquisition and processing cycle is performed:

- When there is no movement, the accelerometer is in sleep mode and wakes up only if the acceleration of one of the three axis exceeds a certain threshold, producing an interrupt to wake up the KL16 microcontroller and start the data acquisition.
- When the accelerometer detects movement, it wakes up and starts acquiring data into its FIFO, whose size is 16 x 3 words of 16 bits, corresponding to 0.8 s. The KL16 microcontroller is still in very-low power sleep mode. When the accelerometer FIFO is full, it produces a "data ready" interrupt to wake up the KL16 microcontroller. The KL16 microcontroller awakes the pressure sensor and starts one measurement of pressure. Then, it reads the data in the accelerometer FIFO. It processes this accelerometer data and the previous pressure measure, in order to count the steps, and estimate the distance and the speed using the algorithms explained in Section III.D. Finally, it completes the pressure measurement and stores it to be used in the next data acquisition and processing cycle. This allows time saving by processing the data during the pressure measurement. Although each processing cycle uses the previous pressure measure, which has 0.8 s of delay, this does not affect the accuracy of the distance and speed estimation.
- At this point, there are two options: either the data logging function is ON and the data and results are written in the flash memory, or it is OFF and the KL16 microcontroller returns to very-low power sleep mode, waiting for a new interrupt.
- This process continues until no steps are detected for a certain time. If it is the case, the accelerometer stops acquiring data and returns in sleep mode until a new motion detection, which restarts the whole process. Notice that this last step is not shown in Fig. 4, for clarity reasons.

It is important to notice that the KL16 microcontroller is most of the time in very-low power sleep mode, and only wakes up when an interrupt from the accelerometer or the 8-bit microcontroller occurs. This allows significant saving of energy and is essential to increase the watch autonomy, as explained in Section IV.C.

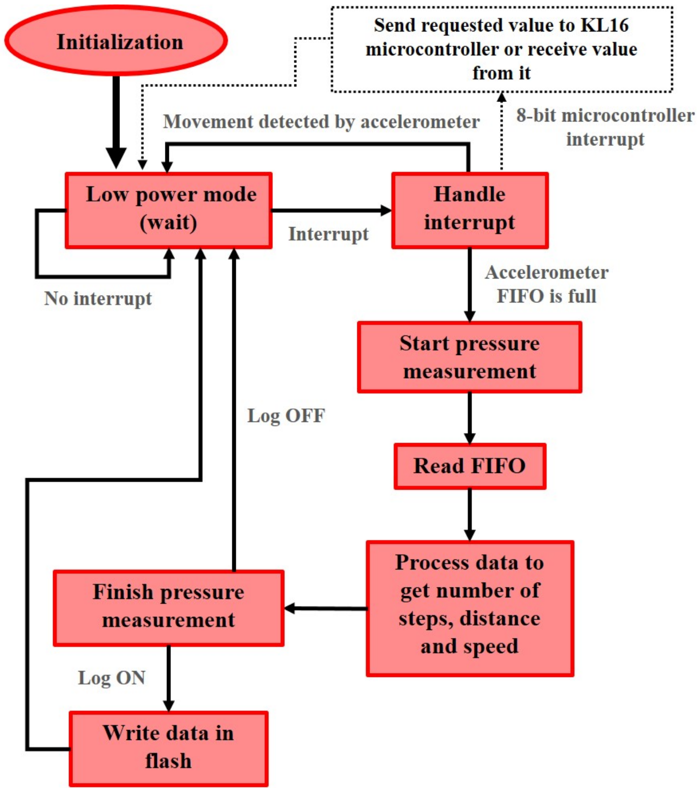

Fig. 4. Sport watch SW block diagram

C. *Reference data sets*

Reference data was recorded using inertial sensors (Physilog IV, GaitUp, Switzerland [7]) fixed to left and right wrist, and left and right foot. A GNSS receiver (CAM-M8, u-Blox, Switzerland [8]) and antenna (ANN-MS, u-Blox, Switzerland [9]) were attached to the person's head using a cap. All systems were electronically synchronized. Inertial sensors comprised 3D accelerometers and 3D gyroscopes sampling at 200 Hz and a pressure sensor sampling at 25 Hz. The GNSS signal was recorded at 10 Hz. The foot sensors were used for step detection [10] while the GNSS sensor was used for measuring walking/running speed. These detected steps and measured speed served as reference throughout the development and testing of the algorithms and the device.

- The first data set contained data from 28 participants who were asked to walk a total distance of 4.7 km on different surfaces (pavement, grass, gravel). Part of the course was walking with hands in pockets, carrying a bag, making a phone call, and walking fast or slow.
- The second data set contained data from 33 participants, who were asked to run for 3 km (once or twice according to their fitness condition). Participants ran at their usual speed and were asked by the experimenter to run faster and slower for short sections. The subjects were also asked to run 100 and 200 m in a race track at three different speeds (low, medium and fast).
- The third data set contained data from 13 participants, who were asked to walk a distance of 4.5 km, with large altitude variations. Slopes were between -41 % and 32 %. The participants were free to walk as they liked.

D. *Data processing algorithms*

The algorithm for step counting and speed and distance estimation was divided in three main parts:

- Detection of activity type (running, walking or non-locomotion movement). This is done by searching peak regularity in the norm of the 3D acceleration signal as well as comparing the amplitude of the signal and its statistical measures using different thresholds determined using data from the data sets [11].
- When the type of locomotion is known, the steps are counted, based on the norm of the accelerometer signal. Different thresholds and methods for step counting are used for the case of running and walking [12].
- Finally, the distance and the speed are estimated using two different methods (one for walking and one for running) based on non-linear estimation. The features used for the estimation are the step frequency and statistical measures (mean, variance, percentiles…) of the acceleration norm and of the barometric pressure. The methods used for the speed estimation in walking and running differ in the form of the estimator, and in the type and number of the features used. In both cases, the models were trained/tested using the data sets. Finally, in both the walking and the running speed estimation, a linear correction according to the height of the user is applied [13].

E. *Development methods and tools*

First, the algorithms were developed and tested in Matlab on a computer, using the three main data sets (see Section III.C). In parallel, the code was implemented in C and tested on a development board from NXP (FRDM-KL26Z [14]), containing the ARM Cortex M0+. The software used to program the board were the last version of Keil µVision (IDE) [15] and Processor Expert [16], which allowed quick configuration of the microprocessor and peripherals. An UART communication was used to send data of the data sets from Matlab in the computer to the KL16 microcontroller, in order to test the algorithms on the real hardware, retrieve the results and compare them with the reference implementation in Matlab. Then, a first version of the code was implemented using a Pebble smartwatch Classic [17], in order to test the algorithms in real-field conditions. This first application is illustrated in Fig. 5. This implementation does not include the differential barometric pressure as a feature, as this type of sensor is not available in the Pebble.

Fig. 5. First implementation on a Pebble smartwatch Classic

Finally, a refined version of the algorithms was implemented in the watch prototype described in Section II. Again, Keil µVision was used to program the KL16 microcontroller, via a proprietary development board used to connect the prototype to a computer and interface with it through a debug adapter (Keil ULINK-ME [18]) and the UART interface to retrieve the logged data. The board is powered by a voltage source and the watch is connected to the board via a flexible connector similar to the one linking the two PCBs in the watch. The setup is shown in Fig. 6, without the voltage source and the computer.

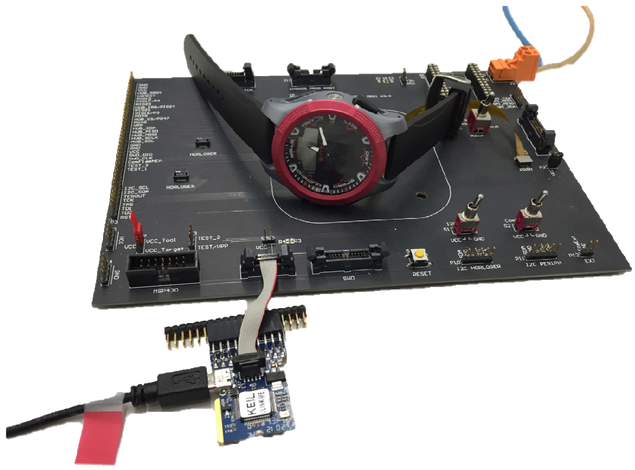

Fig. 6. Development board used to program the prototype and retrieve the logged data

IV. TESTING AND VALIDATION RESULTS

A. *Using the reference data sets*

The results of running the developed algorithms with the three data sets described in Section III.C are shown in Table I. The data from the data sets were processed on the FRDM-KL26 development board with the exact same code as the one programmed in the prototype. Then, the error in percent was calculated for each participant. After that, the mean of the absolute values was computed and those are the values reported in Table I (in bold). Moreover, the 90th percentiles of the absolute values are given between brackets. The 90th percentiles are relevant for the target product, because they show that 90% of the users should be below these numbers.

TABLE I. RESULTS USING THE REFERENCE DATA SETS

Results for the three main data sets	***Steps number error [%]***	***Distance/speed error [%]***
First data set (walk)	**0.92** (1.94)	**3.76** (7.40)
Second data set (run)	**0.32** (0.79)	**6.04** (12.97)
Third data set (walk with slopes)	**2.28** (3.77)	**4.48** (9.53)

B. *Validation in the field*

In addition to the tests made on the data sets, measurements were performed in the field with the final prototype along with the Physilogs IV to have a reference for the step count and the speed. Nine participants walked for 5 km and ten participants ran for 3 km. They could choose their own speed and walk/run freely on a given track. In order to get the best results, the higher consumption accelerometer (BMA280) was used. However, the other accelerometer (ADXL362) should give similar results, except for the running distance and speed estimation. Indeed, this sensor's range is limited to ±8 g, which leads to saturation when the user is running, whereas the BMA280 sensor has a range of ±16 g, which was sufficient for the application. In the final product, both accelerometers could be used together, using the BMA280 only to have a precise distance estimation when running, and taking advantage of the very-low power consumption of the ADXL362 the rest of the time. The results of the validation are given in Table II, with the same format as Table I. These results show that the prototype works properly and have results similar to those obtained with the data sets. The distance/speed errors are larger than the ones using the data sets. This can be explained by the small number of participants during the validation. If one or two participants have a large error, it is really significant if there are only 10 participants.

TABLE II. RESULTS OF VALIDATION IN THE FIELD

Results for the validation	***Steps number error [%]***	***Distance/speed error [%]***
Walking 5 km	**0.29** (0.69)	**6.31** (11.72)
Running 3 km	**0.36** (0.53)	**6.25** (19.47)

C. *Power consumption*

The power consumption strongly depends on the accelerometer choice. This is why two scenarios are evaluated here: one using the BMA280, and one using the ADXL362. Data from the accelerometer is acquired each 0.8 s. Therefore, the pressure is measured each 0.8 s, and one measurement lasts 7.5 ms. During this time, data are processed and the microcontroller is running, meaning it is ON during approximately 0.94 % of the time, and it is in very-low power sleep mode the rest of the time (this does not take into account the negligible time during which the KL16 microcontroller has to communicate with the 8-bit microcontroller). The barometric pressure sensor consumes 5 µA (1 µA in sleep mode) to make one measurement each 0.8 s. When the KL16 microcontroller is running, it consumes 6.2 mA and when it is sleeping, only 2.65 µA. Therefore, its averaged consumption is around 60 µA. For the BMA280, the average consumption is 66 µA (6.5 µA in sleep mode) and for the ADXL362, it is 1.8 µA (0.27 µA in sleep mode). In addition to that, the 8-bit microcontroller and the standard watch components consume 5 µA in average. The pressure sensor and the KL16 microcontroller are always sleeping unless the accelerometer detects movement. Thus, it was estimated that they are in sleep mode at least 12 hours per day. Knowing all this information, the mean consumption was computed with both accelerometers: **76 µA** with the **BMA280** and **40.7 µA** with the **ADXL362**.

With a primary cell battery of 150 mAh (CR2320), this gives a battery life of respectively 82 and 154 days. Long and high current peaks can deteriorate battery performances, but as the peak consumption (when the microcontroller is running) is around 6.2 mA and lasts only 7.5 ms , the battery life should not be reduced by more than 5%. Notice that most of the power consumption is due to the sensors. Therefore, their choice is crucial, and as the sensor market is in constant evolution it is essential to keep looking for less consuming sensors with better performances.

V. Conclusion and future work

The development and the implementation of a sport watch was presented, with emphasis on the description of the HW and SW implementation, as well as the algorithms. The main challenge was to make a device with high accuracy (Step count error of less than 2 %, walking distance error of less than 10 % and running distance error of less than 13 %, for 90 % of users) using only low power inertial sensors to have a high autonomy. Many efforts have been made to optimize the algorithms, in order reduce the consumption of the KL16 microcontroller, and all the components were carefully chosen and configured to consume the minimum power (40.7 µA in the best-case scenario). Compared to similar activity trackers with long battery life such as the Garmin Vivofit™ [19] (which was tested as reference throughout the project and has an autonomy of one year with two primary cells of 125mA each), the accuracy in terms of number of steps and distance/speed estimation is two to three times better. These good performances are mainly due to the relatively sophisticated algorithm, the use of a pressure sensor and the large and representative data sets. Moreover, the values displayed on the watch are updated each second, which is not the case for most of the popular smartwatches, like the Apple Watch or the Samsung Galaxy Gear s2, which have high accuracy but have a processing delay and need several seconds to refresh the displayed data.

Future work could include reengineering the PCB that handles the sport functions adding other sensors such as 3D magnetometer or even a 3D gyroscope and GPS, which should be used sparingly, and reworking the algorithms to include the data from these new sensors. This would greatly improve the accuracy at a cost of a higher consumption. Moreover, the current sensors could be replaced by less consuming ones that will soon be available on the market. Also note that depending on the final product, the sport functions could be activated only on demand (few hours per week) increasing the autonomy of the watch to up to 2 years. Finally, the step count error is just one side of the coin. Indeed, avoid counting steps during non-locomotion movements is very important but also extremely difficult. In our future work, this topic will be explored in more details, with new and dedicated data sets.

Acknowledgment

This project was partly supported by the Swiss Federal Department of Economic Affairs, Education and Research (EAER) through the Commission for Technology and Innovation (CTI) under Grant CTI_14787.1_PFNM-NM ("MINOS" project). We are grateful to all the people who have contributed to this work: Dr. Pradyumna Ayyalasomayajula from ESPLAB, Dr. Farzin Dadashi from LMAM and all the participants who walked or ran to help us make the largest possible data sets.

References

[1] http://www.nxp.com/products/microcontrollers-and-processors/arm-processors/kinetis-cortex-m-mcus/l-series/kinetis-kl1x-48-mhz-general-purpose-connectivity-ultra-low-power-mcus:KL1x

[2] J. Yiu, The Definitive Guide to ARM® Cortex®-M0 and Coterx.M0+ Processors, 2nd ed. Oxford: Kidlington, 2015.

[3] https://www.bosch-sensortec.com/bst/products/all_products/bma280

[4] http://www.analog.com/en/products/mems/mems-accelerometers/adxl362.html

[5] https://www.bosch-sensortec.com/bst/products/all_products/bmp280

[6] http://www.issi.com/WW/pdf/25LP128.pdf

[7] http://www.gaitup.com/

[8] https://www.u-blox.com/en/product/cam-m8-series

[9] https://www.u-blox.com/en/product/ann-ms

[10] M. Benoit, H. Rouhani, X. Crevoisier, K. Aminian. "Quantitative estimation of foot-flat and stance phase of gait using foot-worn inertial sensors," Gait and Posture, Volume 37, Issue 2, February 2013, Pages 229–234. doi:10.1016/j.gaitpost.2012.07.012.

[11] B. Fasel, K. Aminian, "Stride-to-stride gait variability in daily life measured using accelerometer attached to the wrist," 4th Internatinal Conference on Ambulatory Monitoring of Physical Activity and Movement (ICAMPAM), Limerick, Ireland, June 10-12-2015.

[12] B. Fasel, C. Duc, K. Aminian, "Stride count and frequency measured with a wrist-worn inertial sensor," 13th International Symposium on 3D Analysis of Human Movement, Lausanne, Switzerland, July 14-17-2014.

[13] B. Fasel, F. Dadashi, K. Aminian, "Instantaneous walking speed estimation for daily life activity monitoring based on wrist acceleration," 4th Internatinal Conference on Ambulatory Monitoring of Physical Activity and Movement (ICAMPAM), Limerick, Ireland, June 10-12-2015.

[14] http://www.nxp.com/FRDM-KL26Z

[15] http://www2.keil.com/mdk5/uvision/

[16] http://www.nxp.com/products/software-and-tools/software-development-tools/processor-expert-and-embedded-components

[17] https://www.pebble.com/

[18] http://www2.keil.com/mdk5/ulink/ulinkme

[19] http://garmin.ch/vivofit