Low-power Lessons from Designing a Wearable Logger for Long-term Deployments

Eugen Berlin
AGT International, Darmstadt, Germany
eberlin@agtinternational.com

Martin Zittel, Michael Bräunlein
Technische Universität Darmstadt, Germany
{martin,michael}@ess.tu-darmstadt.de

Kristof Van Laerhoven
University of Freiburg, Germany
kristof@ese.uni-freiburg.de

***Abstract*—The advent of a range of wearable products for monitoring one's healthcare and fitness has pushed decades of research into the market over the past years. These units record motion and detect common physical activities to assist the wearer in monitoring fitness, general state of health, and sleeping trends. Most of the detection algorithms on board of these devices however are closed-source and the devices do not allow the recording of raw inertial data. This paper presents a project that, faced by these limitations of commercial wearable products, set out to create an open-source recording platform for activity recognition research that (1) is sufficiently power-efficient, and (2) remains small and comfortable enough to wear, to be able to record raw inertial data for extended periods of time. We study especially, via high-resolution power profiling, several trade-offs present in the choice for the basic hardware components of our prototype, and contribute with three key design areas that have had a significant impact on our prototype design.**

Fig. 1: The custom-made inertial data logger was designed for long-term (day and night) recording of data for activity recognition at 100 Hz, while simultaneously taking user acceptance requirements for wearing such as device into consideration. We focus on the evaluation of low-power design choices.

I. Introduction

One of the enduring challenges in creating and deploying robust activity recognition systems is the deployment of wearable sensors that are unobtrusive and lightweight. This has also been identified in earlier research on mobile sensing platforms, for example in [1]. Several publications in activity recognition research have observed that long-term operation of the wearable sensing units is critical for various types of application scenarios. Long-term monitoring of psychiatry patients, for example, can be considered as one of the more challenging scenarios, due to its constraints regarding the type of sensors, the power efficiency aspect, and usability considerations. While commercially available actigraphs provide abstracted data in form of activity counts, there were no wearable platforms that allowed the users' activities to be captured as raw acceleration data for more than a few hours.

In our case, due to lack of available off-the-shelf sensor platforms that would fit the requirements, a custom sensor unit had to be designed and built. This has led us to consider building a prototype that would explore long-term activity recognition approaches with a focus on the recording equipment. A trade-off between two features was prevalent throughout this design phase. A main requirement is that a **user accepts to wear** the wrist-worn units continuously, day and night. Design, size and weight are important, as in [2], but also functionality: Although the unit was meant to record data, many users, especially in the medical domain, would only wear the device if it provided them the function of a basic wristwatch, both for its functionality and for wearing something that would raise questions. On the other hand, the device needs to be **power-efficient** enough to be able to record inertial data at a high sampling rate for extended periods (weeks at a time) on a single battery charge, thus requiring components to be turned off or in sleep mode whenever possible. Our prototype is open-source, publicly available at www.ess.tu-darmstadt.de, whereas most other platforms are not.

We focus on the latter requirement of low-power operation, while assuming the constraint of user acceptance, which in our case led to the inclusion of a particularly small battery and the addition of an OLED display. Turning off the display whenever possible, and turning it on only when required by the user, is hereby the straight forward choice in preserving limited power resources. Figure 1 shows the current prototype.

The remainder of the paper is organized as follows. Section II presents the main components of the wearable platform. Section III describes our experimental setup to measure power consumption. Section IV is dedicated to discussing obtained evaluation results. The paper is concluded in section V.

II. Wrist-worn Unit Design

The custom-built platform is centered around a Microchip PIC18F46J50 microcontroller, which embeds in a small-scale form-factor key components for acquiring and recording inertial data. Among the most relevant features embedded in the microcontroller are the real-time clock, multiple internal oscillator circuits, digital and analogue communication interfaces, and a full-speed USB 2.0 communications module. The real-time clock is specifically important to obtain accurate time stamps during logging, which are then used in the visualization of the sensor data to the user through a human readable time axis. Furthermore, the time stamps are necessary to synchronize the sensor data with user annotations, whether kept in diaries or added interactively on recall basis.

The main sensor unit of our platform is the 3-dimensional ADXL345 microelectromechanical system (MEMS) accelerometer, which is able to obtain accelerations in a range between ± 2 up to ± 16 g with sampling frequency up to 3600 Hz. The accelerometer sensor is connected via a Serial Peripheral Interface (SPI) digital bus with the microcontroller. In our experiments, the accelerometer sensor was configured to a sensitivity of ± 4 g at 10 bit resolution and a sampling rate of 100 Hz. The accelerometer unit itself comes with important features, such as a double-tap and fall detection, low-power modes and an internal FIFO buffer that allows to transmit sensor values in bursts to the microcontroller. This latter feature is important, as it allows the main processor to switch to power-efficient sleep modes or perform other tasks between these communication burst. In addition to the time and acceleration values, ambient light intensity is regularly obtained through a photosynthetic diode.

Due to the amount of data being generated at 100 Hz, a local flash memory is needed to store the acceleration data along with the time stamps. Non-volatile flash memory suits the application scenario demands, since it is available off-the-shelf in small form factors (microSD cards) and can preserve stored sensor data even when the battery runs out of energy. Connectors and circuitry are available on the sensor board for attaching the microSD card and for storage of sensor data. The microSD card is then transparently accessible via USB as a so-called mass storage device that appears to the user as a common memory stick with FAT16 file system.

The prototypes are powered from miniature Li-Polymer rechargeable batteries with a capacity of 180 mAh. For user-friendly recharging of the battery via USB the sensor is equipped with the MAX1551 charger integrated circuit. To meet the requirements of long-term 24/7 deployment, the unit is packed in a custom shock-proof case and provided with an anti-allergic textile wrist strap. The miniature OLED display is used for visualization of the current time, date or sensor values. The display is by default turned off and can be activated by the user by double-tapping the watch.

III. Experimental Setup

This section presents the experimental setup to obtain current consumption figures for our prototypes. We first present the hardware setup to record the current drain traces. Aiming at performing reproducible tests for all our test cases, we propose to use a benchmarking platform. Lastly, to access how much a sensor would last in a real-world scenario, we deploy the sensor for multiple days to measure overall consumption.

A. Measuring Current Consumption

Critical in the power evaluation of the different parts of our prototype is the way we measured the individual components' energy consumption footprints. This section covers the details of our method to measuring and record the energy consumed by our prototype in particular modes. After presenting the basic principles of operation, we provide the details on our measurement setup based on the Arduino Due[1] platform.

[1] http://arduino.cc/de/Main/ArduinoBoardDue.

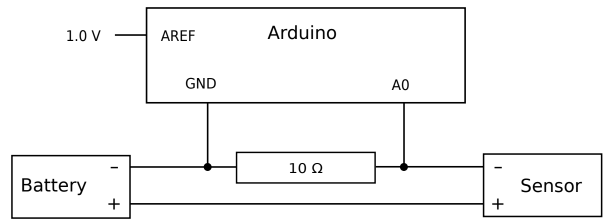


Fig. 2: The schematics to measure current consumption via voltage drop over a resistor. To acquire and log the current consumption footprint for detailed off-line analysis, we use the Arduino platform supplying a reference voltage of 1.0 V.

Figure 2 depicts the schematics of the voltage measurement circuit with which we have obtained detailed recordings of current consumption, shown later in the evaluation section. Following the Ohm's law, we can from the measured voltage drop over a sufficiently small resistor compute the current drawn by the sensor device: $I = \frac{V}{R} = \frac{1}{10}V$, with $R = 10\Omega$.

The Arduino board is connected to a computer and used as a high-resolution voltage logger over the resistor R on the analog A0 pin against the ground GND pin. A reference voltage of 1.0 V is supplied to the AREF pin, and power for both the prototype and the Arduino is provided by a regulated bench power supply. The A0 pin is sampled through an ADC at full speed, whereby the resolution is set to 12 bit. The measured values are transmitted to a computer for logging and further off-line evaluation.

To access the current consumption of different operation modes of our prototype, we need to consider the current drain over the operation's duration, which is given by the area under the curve for the current draw measurements. Most relevant information for power profiling of a sensor device can be gathered by obtaining (a) the baseline current consumption of the sensor device when in low-power idle or sleep mode, (b) current consumption of different hardware components used to perform various tasks, such as sampling the accelerometer, writing data to persistent memory, or displaying information on the display. To obtain current consumption for different operations, we need to first identify these in our voltage data set, extract corresponding voltage readings and compute the area under the curve, for which we use the composite trapezoidal rule (*trapz* in Python).

B. The Robot Arm Benchmark

In order to obtain comparable current consumption figures for all the different configurations, for each test the data logger should be moved in a similar, human-like, way and for a similar amount of time. A human is not able to perform the motions for each test as reproducible as required, which motivated us for finding a suitable platform. Luckily, we were able to use a robotic arm in one of the robotics research groups at the university for our experiments. The robot is able to perform very precise motions for a specified amount of time, or repeatedly for a given number, whereby the motion itself can be programmed by either by manually defining the trajectory in space, or by providing the angular configuration of the joints and letting the software decide on the cheapest trajectory.

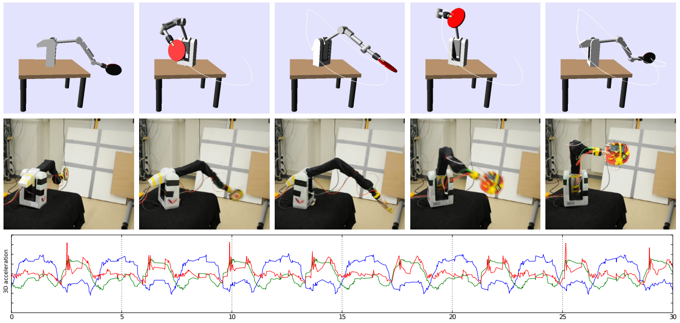


Fig. 3: A robot arm was used to perform a set of motions in a reproducible way. The simulation (top) depicts one of the motion traces, the photos (middle) show the sensor and measurement hardware attached to the platform. The acceleration data (bottom) reveals the repetitive execution of this motion.

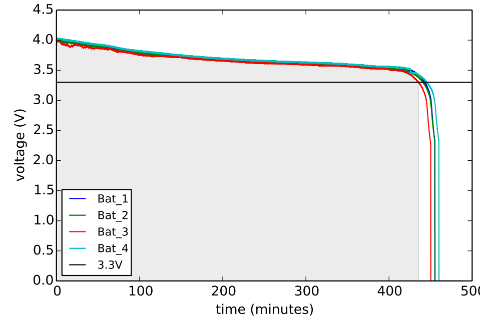


Fig. 4: Determining the capacity of four fully charged batteries via a 110 Ω resistor via the area under the curve up to the nominal voltage of 3.3 V (highlighted in gray for *Bat_3*). Similarly, we can obtain the remaining capacity of a battery that was powering a sensor in a deployment.

Figure 3 shows the trajectory simulation as well as photos from the actual deployment, with the sensor and the measurement hardware being mounted on the robot arm. This way we were able to obtain the current consumption measurements, along with acceleration data from the sensor node. Since the robot is not designed for long-term operation, only short-lasting tests with different configurations were conducted.

C. Long-term Current Consumption Measurement

While the robot arm is a great benchmark utility for comparing different combinations and parameters, a real-world experiment is impossible to model. In order to access how long a sensor would last on a single battery charge in real-world conditions, multiple sensors were deployed to be worn at the human wrist continuously for full 10 days.

In this experiment, using the aforementioned hardware to capture detailed current consumption figures is not feasible. Instead, the consumption is computed via the battery capacitance difference: Obtaining the capacity of the full battery before the deployment (Figure 4), and the remaining capacity after these 10 days, we are able to compute the average current drain over the deployment time frame. Based on the acceleration data obtained and the overall activity intensity during this period, we can estimate how long a sensor would last when deployed in a long-term application. Furthermore, attaching a sensor to a constantly moving or vibrating device will allow to evaluate the runtime characteristic of the sensor nodes on a single battery charge, and access the quality of manufacturing of the sensor or the impact of different configuration settings.

IV. Evaluation

During the design of our prototype, following design choices were found to have a high impact on the low-power operation of the system, and will be evaluated in this section:

1) Which component should control the sampling, the microcontroller or the accelerometer itself?
2) There are a multitude of microSD cards available on the market, does it matter which to use?
3) The OLED display is necessary for certain applications; how much energy does it consume?
4) How long will the prototype last on a single battery charge with a sampling rate of 100 Hz?

To investigate these questions thoroughly, a number of short tests were performed to obtain current drain figures with the setup presented before, whereby settings and components have been varied. We considered in our investigation microcontroller vs. accelerometer FIFO sampling, different data compression settings, three different low-power modes, four different flash cards, the effect of using the OLED display as well as the impact of adjusting the sampling frequency. These test cases are summarized in Table I.

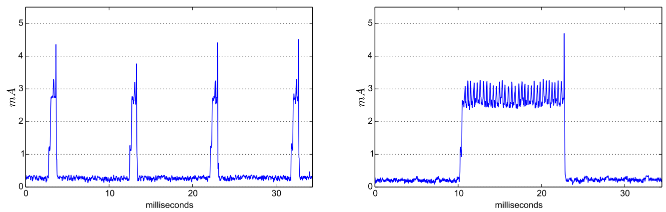
A second test aimed at measuring total current consumption for a long-term deployment, as one would expect in real human activity recognition scenarios. For that, we deployed two sensors that have been worn continuously day and night for full 10 days. By measuring the capacity of the full battery before the test and after the deployment, the current consumption for this time frame can be computed.

A. Which Component Should Control the Sampling?

In most sensor unit implementations where sensor data need to be acquired at equidistant intervals, the microcontroller is commonly the unit that times and polls for new data from the

TABLE I: Overview on the different test cases, varying different parameters, such as the sampling method and frequency, low-power modes, run-length encoding, and the microSD card.

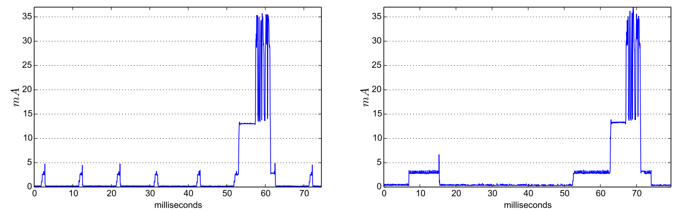
No.	Sensor	Sampling	Freq.	LPM	RLE	SD card
1	Basic	PIC 10ms	100 Hz	normal	2	TS 1 GB
2	OLED	PIC 10ms	100 Hz	normal	2	TS 1 GB
3	Basic	FIFO	100 Hz	normal	2	TS 1 GB
4	Basic	FIFO	100 Hz	low-pwr	2	TS 1 GB
5	Basic	FIFO	100 Hz	low/auto	2	TS 1 GB
6	Basic	FIFO	50 Hz	low/auto	2	TS 1 GB
7	Basic	FIFO	25 Hz	low/auto	2	TS 1 GB
8	Basic	FIFO	100 Hz	normal	2	SD 2 GB
9	Basic	FIFO	100 Hz	normal	2	TS 2 GB
10	Basic	FIFO	100 Hz	normal	2	SB 1 GB
11	Basic	FIFO	100 Hz	low/auto	0	TS 1 GB



(a) PIC polling every 10 ms causes small peaks of 1 ms width.

(b) Accelerometer's FIFO buffer allows communication in bursts.

Fig. 5: Current consumption traces showcasing the two sampling methods: (a) microcontroller polling for samples every 10 milliseconds, and (b) accelerometer sampling with its internal FIFO buffer. Note that transmitting 32 samples at a time reduces the overall communication overhead and allows the microcontroller to sleep for longer periods of time.



(a) PIC sampling every 10ms and writing data to microSD card.

(b) FIFO communication bursts and writing data to microSD.

Fig. 6: Current consumption traces with (a) polling and (b) burst communications between the microcontroller and the accelerometer (smaller peaks), along with writing to an SD card (big peaks), revealing the energy cost of this operation.

sensor. Such typical behavior for a 3D MEMS accelerometer is depicted in Figure 5a: every couple of milliseconds (10 in this case, due to sampling at 100 Hz), the sensor unit is woken up from a low-power sleep mode via a timer interrupt, to communicate with the accelerometer (via SPI) and acquire a new value tuple (consisting of the x, y, and z axis). This causes every 10 milliseconds a small peak in power consumption, taking about a millisecond.

Many recent MEMS accelerometer chips come with a large set of digital support functions, however, including an operation mode which lets the accelerometer do the acquisition of new 3D acceleration samples for storage in a local FIFO buffer. For the ADXL345 used on our prototype, this buffer holds 32 samples, which means filling a buffer takes a bit more than 300 milliseconds for our target sampling rate of 100 Hz. Figure 5b shows the typical current draw pattern in such a case. Additionally, it is possible to invoke the accelerometer's power-saving functionalities that cause more noise but has a slight effect on the current draw as well.

After obtaining initial current draw figures for sleep mode (0.24 mA), the FIFO or polling communication (2.8 mA), and writing to the microSD card (13.5 mA), a set of small tests lasting 7 minutes was conducted in order to evaluate different operation modes and hardware components (listed in Table I). The result is that FIFO sampling is more efficient: First, while the accelerometer is collecting the sensor samples, the microcontroller can be put to low-power sleep mode to preserve energy. Second, transmitting the 32 values at a time results in a reduced communication overhead and almost the half the current drain (Table II, the sampling column).

B. What a Difference an SD Card Makes

After the acquisition of the sensor values, these are typically first stored in a buffer inside random access memory (RAM). Once this buffer in the microcontroller's volatile memory is filled up, it needs to be offloaded to permanent storage. Many wearable devices that are used to record fine-grained sensor data nowadays utilize flash memory, either on-board flash chips or replaceable storage cards (e.g., microSD). Such replaceable cards have two main advantages: First, these can be easily bought in large quantities and at appropriate sizes of multiple Gigabytes. Second, broken cards can be easily replaced without affecting the sensor device itself.

In our study, writing data to the SD card is the most expensive operation with regard to current consumption (not considering the operation of the OLED display). With our main goal being able to perform activity recognition from sensor data where we specifically rely on subtle detailed information in the signal, there is also the need for high-frequent sampling (at 100 Hz). Obviously, storing the raw sensor data will result in lots of writing operations, impacting the lifetime of the sensor. Carefully designing and implementing the logging routine yields a huge power efficiency potential. Figure 6 shows examples of current consumption traces for sensor polling and burst communication, along with the writing of sensor data to the microSD card.

One of the possible approaches to reduce the amount of write operations is to compress sensor data on-line in the microcontroller's RAM, before storing it to the microSD card. For our sensor device, we use run-length encoding (K-RLE) [3], which is a very common and widely used method to compress data. In our case, the two advantages of K-RLE are: (1) it compresses identical sensor values and preserves subsequences with a varying signal, and (2) it can be used to filter out noise in the signal, thus performing very efficiently on flat sensor data, when correctly choosing the threshold. Choosing $K = 2$ over $K = 0$ in our tests resulted in a significant reduction of write to flash operations from approximately 23 to 6% of overall consumption (cf. Table II, tests 5 and 11).

Considering our data logger, the study revealed that is also mandatory to carefully chose appropriate memory cards. To show their impact on the overall current consumption, we considered four microSD flash cards from three manufacturers, namely Transcend, Sandisk and SwissBit, with capacities of 1 GB and 2 GB. Figure 7 shows the findings regarding these cards, with an unexpected result: the 2 GB Transcend card turned out to consume almost three times the current of its 1 GB version or the 2 GB card by Sandisk. The 1 GB SwissBit card has a low plateau current drain of 2.8-3 mA, matching the level of microcontroller and accelerometer communications, and a very short peak of approximately 45 mA lasting 1 millisecond. This is also reflected in the overall consumption

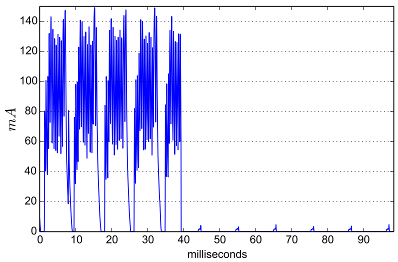


Fig. 8: Current consumption trace of a data logger with the OLED display. Using the display results in a high current drain, represented by the high peaks on the left side. After the display has been turned off by the user (double-tap feature), 10ms polling from the microcontroller becomes visible.

figures of the tests 3, 8, 9, and 10 in Table II. Hereby it is necessary to note that the SwissBit card contains single-level cell flash memory, whereas the other cards have multi-level cell flash, and therefore comes at a much higher price (10 fold), but yields advantage with regard to power consumption. The total current drain of the writing operation (namely the area under the curve) is therefore much lower than of the other microSD cards (Figure 7e), making this more expensive card much more preferable in this comparison.

The conclusion of this evaluation is that cheap consumer cards need much more thoroughly testing before being used in long-term deployments. Just relying on the peak current drain figure given in the data sheets is not sufficient. What matters is the actual current consumption trace that will reveal the details of this essential hardware component. Obviously, choosing industry single-level cell flash memory has a huge advantage of much lower energy consumption, at a high monetary cost.

C. The (Battery) Cost of an OLED Display

Our prototype is equipped with an OLED display, which is programmed to show current time and date for a few seconds whenever the user double-taps the data logger. This was particularly a requirement for several long-term trials, in which many users reported unwilling to wear a unit on the wrist that would look unfamiliar enough to raise questions. By making it look and function as a wrist-watch with the addition of the display, acceptance was much higher. However, this component comes at a higher production cost for the entire prototype, and undoubtedly at an impact on power consumption. Figure 8 shows a subsequence of the display current consumption just before it was turned off by the double-tap from the user.

The OLED display requires 3.3 V power supply for the integrated display driver, which nicely fits our data logger design. On the other hand, the OLED display also requires an additional supply of 12 V for its back-light. To achieve this, a step-up circuit is necessary, consisting of multiple additional components. The drawback of this approach is the reciprocal-proportional dependency of voltage and current: to achieve a step-up from 3.3 V to the required 25 mA at 12 V (according to the data sheet), we need to supply at least 90 mA. Our measurements show that due to the step-up circuit, the OLED display and all other hardware components, the consumption peaks reach up to 140 mA at 3.3 V. Current work therefor focuses on a more efficient implementation of the display.

TABLE II: Comparison of the electric charge consumed in total, and for the three main operations: writing to microSD, transfer of acceleration samples from the accelerometer to the microcontroller, and the operation of the OLED display.

No.	Total	Writes to SD		Sampling		Display	
	mAs	mAs	%	mAs	%	mAs	%
1	130.72	4.76	3.6	63.65	48.7		
2	353.93	1.74	0.5	4.64	1.3	347.28	98.1
3	93.40	5.78	6.2	29.22	31.3		
4	72.78	7.40	10.2	26.15	35.9		
5	84.68	6.19	7.3	28.65	33.8		
6	61.37	3.84	6.3	13.64	22.2		
7	46.80	2.12	4.5	7.28	15.6		
8	103.29	13.21	12.8	29.26	28.3		
9	84.85	7.41	8.7	28.33	33.4		
10	103.40	1.44	1.4	30.48	29.5		
11	95.71	22.28	23.3	26.44	27.6		

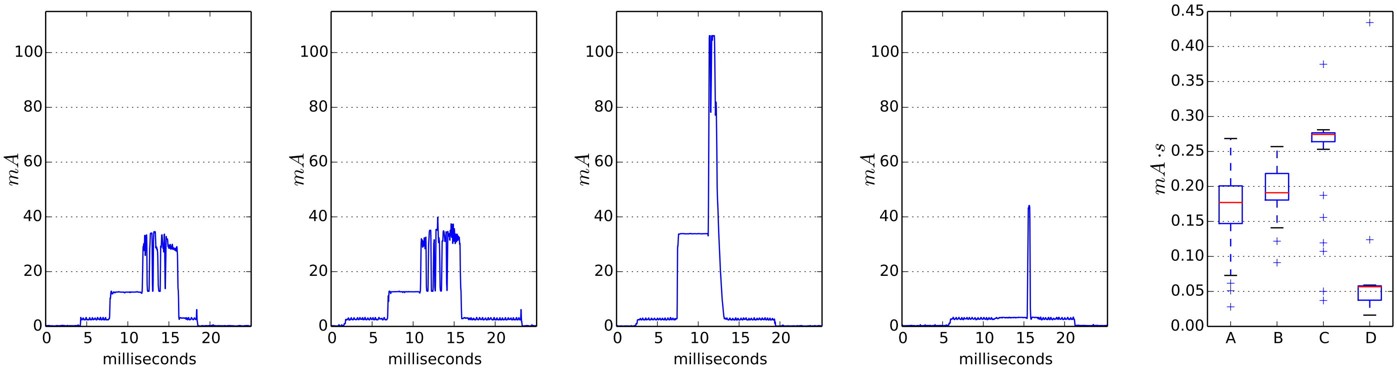
Although the display was turned on only for a short period of time during the test, the consumption of the OLED and the step-up circuitry reaches more than 98% of the total current draw (Table II). With this result, while not representative for real-world usage, it becomes very important to efficiently design the usage of the display interface. This particularly means that the display has to be turned off as often as possible when not needed by the user.

D. The Impact of Reduced Sampling Frequency

Varying the sampling frequency has a direct impact on the amount of sensor data generated by the accelerometer unit. While we aim at high sampling rate of 100 Hz to capture full detail of human motion, lowering the sampling rates can still be tolerable for various applications. For example, in sleep monitoring and actigraphy scenarios, raw sensor data is generally converted to activity counts, whereby the epoch size is generally relatively large (in the range of minutes).

With that, we face a trade-off between data granularity and life time of the sensor, which directly impacts the usability aspect. Reducing the sampling rate from original 100 to 50 or 25 Hz results in a proportional decrease of communication between the accelerometer and microcontroller (Table II, tests 5, 6, 7) and in the amount of write operations to the SD card. While the sensor node will still consume current for basic operation (microcontroller computation, accelerometer sampling, real-time clock, some loss in the circuits) on a level of approximately 0.4 mA, the overall consumption is reduced.

Thus, a reduced sampling rate will significantly improve the life-time of the sensor, which was also confirmed by multiple long-term tests. Sensors, configured for a sampling rates of 100 as well as 50 Hz were worn continuously for 14 days at the wrist. At 100 Hz, the sensors were able to obtain sensor data for up to 11 days, before the batteries were drained completely. With a sampling rate of 50 Hz, even after 14 days the batteries were not completely drained, and the sensors are able to log for additional couple of days.



(a) 1 GB Transcend (b) 2 GB Sandisk (c) 2 GB Transcend (d) 1 GB SwissBit (e) Distribution

Fig. 7: Current consumption traces of the four SD cards used in the experiment. The 1 GB Transcend and 2 GB Sandisk turn out to be very similar. The 2 GB Transcend card (c) has a much higher peak consumption, while requiring few milliseconds less to finish its task. The 1 GB SwissBit card (d) has a significantly lower power consumption, at a much higher purchasing price. Computing the area under the curve results in total amount of current spent for the writing-to-flash-memory operation. After extracting these peaks from obtained dataset, the distribution of per peak current consumption is shown in subfigure (e).

E. The 10-day Deployment Test

The evaluations presented before have focused on the impact of different settings and components on the current consumption of the prototype data logger. Aiming at deploying the data logger outside laboratory conditions at the wrist of users, in order to obtain continuous day and night human motion data, an evaluation is required that considers current consumption over a comparable time frame. For this, two sensors have been deployed for capturing human motion for a time frame of 10 days. The sensors were worn continuously day and night, and were only taken off during showering or bathing. The configuration for these sensors was exactly as in test 4 (Table I): 100 Hz FIFO sampling, low-power modes enabled, RLE with $K = 2$, Transcend 1 GB card. The OLED display was not used at all during the logging period.

The overall current drain of the sensor was obtained over the capacity change from a fully charged battery before the deployment to the capacity state after the 10 days. The delta of these values, divided by the time frame, provides us with an average current drain figure, and with that allows to estimate runtime for this particular sensor-battery pair. For example, a sensor running with the configuration above for 10 days of logging consumed an electric charge of 93 mAh. With that, one day of operation approximately drains 9.3 mAh from the battery. With a nominal battery capacity of 180 mAh, the sensor node should theoretically last more than 19 days.

Experience shows that this estimated runtime is not reached. The reason for this are the physical characteristics of the flash memory: First, the high peak consumption of the SD card (approximately 36 mA, cf. Figure 7a), and secondly, its relatively high operating voltage of 3.3 V with very little deviation tolerance. While the microcontroller, the accelerometer sensor and other components on the sensor board can tolerate lower operating voltage, writing to flash memory will fail as soon as the battery voltage drops below 3.2 V. With this constraint in mind, the runtime of a sensor node with the given setup accounts for up to 14 days.

V. Conclusions

This paper presented the experience gathered in developing a wrist-worn and low-power activity logging unit, which is able to record 3-dimensional acceleration data at a sampling rate of 100 Hertz for two weeks on one battery charge. Using the form-factor of a wrist-watch to safeguard user acceptance, we specifically focused on the choices and parts of our prototype that have the biggest impact on energy consumption of the whole unit. After presenting the details on how we obtained our measurements using an off-the-shelf low-current acquisition setup, we contribute with these findings in particular:

- Accelerometer-based FIFO sampling has shown to result in slightly better energy figures than microcontroller-based sampling, requiring shorter idle times between samples.
- The choice of SD card manufacturer and size showed strong variations on the energy footprint of the whole unit. The current draw for 2 GB Transcend consistently reaches 100 mA, whereas for others 35 mA was measured. A SwissBit 1 GB card exhibits lowest consumption for write operations.
- The OLED display, though appreciated by the users wearing our prototype, has an enormous impact on energy consumption. This adds importance to any mechanisms that turn off the display whenever it is not needed by the user.
- A 10-day test has proven to be an efficient method to estimate how long a battery under normal usage would last. For our prototype, considering the SD flash requirement for an operational voltage of 3.3 V, this was calculated to be approximately 14 days.

References

[1] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower *et al.*, “The mobile sensing platform: An embedded activity recognition system,” *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 32–41, 2008.

[2] C. Narayanaswami, T. Inoue, T. Cipolla, J. Sanford, E. Schlig, and Vishal, “IBM’s Linux Watch: The Challenge of Miniaturization,” 2002.

[3] E. P. Capo-Chichi, H. Guyennet, and J. Friedt, “K-RLE: A New Data Compression Algorithm for Wireless Sensor Network,” in *SENSORCOMM’09*, 2009, pp. 502–507.