Design of a hardware prototype for monitoring of wound infections

Andreas Könsgen, Gourav Saini, Jens Dede, Anna Förster
University of Bremen

1. Task description

In the scope of the WHISKIES project, a hardware prototype for a wearable wound monitoring device is developed. The device should be capable of predicting a non-optimal healing of the wound and generate wireless alerts to notify the injured person so that early intervention can be provided to the wound. The device should also have a long battery life to avoid frequent battery changes in order to cause as little disturbance to the wearer as possible.

The wound monitoring wearable device is to be worn by astronauts on space missions for rapid detection of abnormal wound healing, it can also be part of space suits in case of outdoor missions on space stations or habitats. The device has the following features.

- Sensors to gather health markers critical for wound healing.
- Microcontroller to identify critical health marker values and having an algorithm to differentiate between change in a critical health marker due to improper wound recovery or other reasons.
- Wireless link to sent alerts to a server when critical markers change indicating abnormal wound healing.
- Long battery life by using low-power components.
- Easy manufacturing of the device by deploying electronic components readily available in the market.

2. Wound status markers

Various body health markers change when an optimum healing of a wound is not taking place, sensors capable of measuring these health markers can be utilized to detect problems in healing of the wound and alerts can be generated so that the wound can be inspected on time.

- pH: A wound has an alkaline pH and it gradually changes to acidic as the wound heals. A stalled alkaline pH indicates that healing process has stopped. [1]
- Temperature: An increased temperature of the wound is an indicator of infection. [1]
- Oxygen saturation: A prolonged abnormal oxygen saturation level i.e. hypoxia or hyperoxia is an indication of stalled healing. [2]
- Moisture: Optimal moisture level is supportive of tissue restoration [1]

3. Selection of sensors

The criteria for selection of the sensors for the wearable device is ready availability and small size in order for the implementation on a compact PCB. Several sensors for health monitoring have been developed and are available in the market. Also, some publications have demonstrated prototypes of wearable sensors developed specifically for monitoring of wounds. The majority of these sensors either lacks market availability or they are not suitable for a compact wearable device.

- pH: The authors in [1] have successfully demonstrated wearable real time pH sensors. But the market availability is still lacking. pH sensors available in market require a probe and are therefore not suitable for wearable devices.
- Temperature: temperature sensors are widely available, such a sensors is used in the tests described in this report.
- Oxygen saturation: a sensor like MAX30100 can be employed for pulse oximetry to measure body oxygen saturation, but wound tissue oxygen saturation sensor are not readily available.
- Moisture: proof of concept has been shown [3], but no readily available sensor is available.

The only readily available sensor that can be used to detect abnormalities in healing of the wound is the body temperature sensor. The body temperature also rises with exercise. In this project, we demonstrate the concept of an architecture that can detect the rise in temperature and infer if the rise in temperature is due to body doing exercise or the body was in rest when the temperature rose.

4. Proposed low power hardware

The electronic components should be as power efficient as possible in order to minimise battery changes and keep the battery size small. Below is a list of low-power components fulfilling this requirement.

Microcontroller:	Attiny3216 with 32 kB flash and 2 kB SRAM
Temperature sensor:	TMP117
Accelerometer:	ADXL345, 3-Axis, digital interface
Power management:	TPS61322: Input voltage 0.9 V to 5.5 V, output voltage 1.8 V to 5.5 V 1.10 A current limit
Power supply:	Coin cell battery, 200 mAh

5. Testing hardware

Due to availability problems of the proposed components, the actual tests are performed with the hardware given below. For the microcontroller, we chose a model compatible to the low-power version mentioned above so that a transfer of the software is possible later on with minimum modification of the code. The circuit diagram of the test setup is shown in Fig. 3, the physical setup in Fig. 4.

Microcontroller:	Atmega1284p with 4 kB EEPROM and 128 kB flash memory
Temperature sensor:	GY-21(Si7021) – Temp range: –40 to +125 °C (see Fig. 1)
Accelerometer:	ADXL345 3-Axis, digital interface, acceleration range setting: ±4g (see Fig. 1)
Communcation:	HM-10 Bluetooth Low Energy Interface (see Fig. 2)
Circuit board:	Breadboard
Programmer:	Arduino as in-system programmer (ISP)

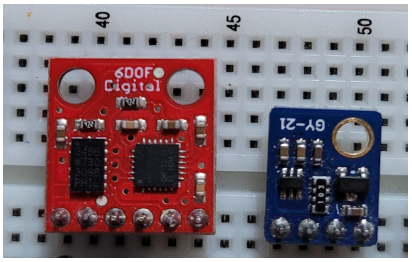


Figure 1: Accelerometer board (left) and temperature sensor board

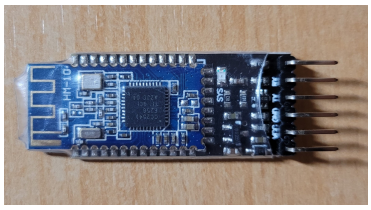


Figure 2: BLE interface

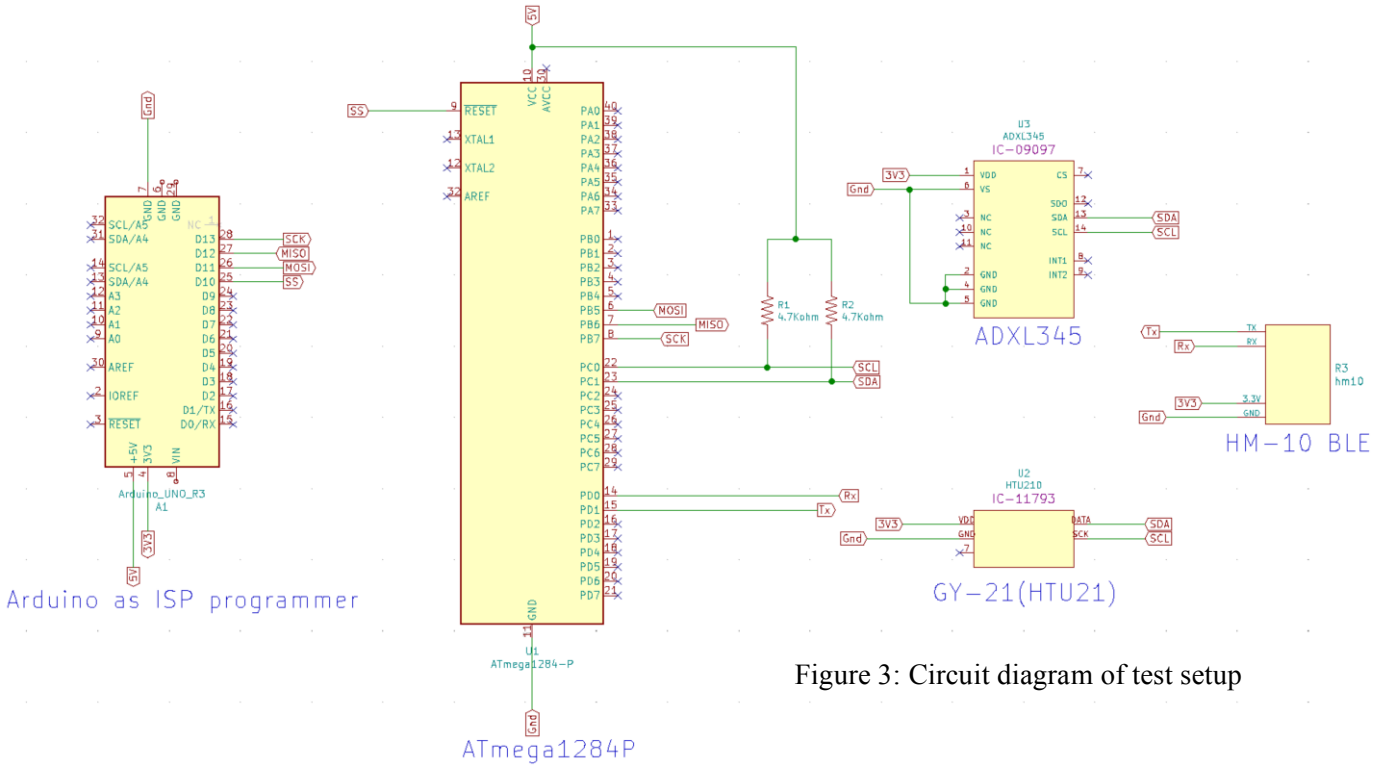


Figure 3: Circuit diagram of test setup

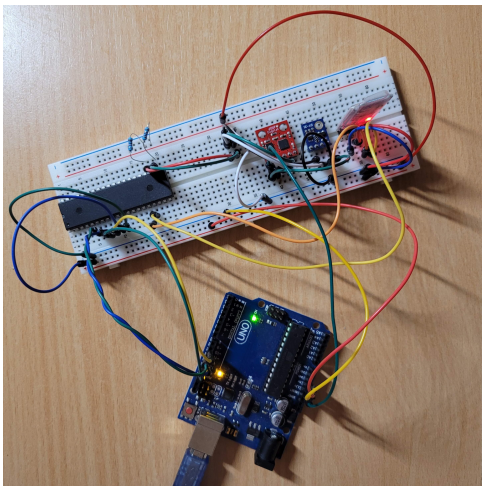


Figure 4: Physical test setup

6. Data acquisition and processing

For the demostration in this project, the algorithm is triggered when the temperature rises above a hard coded threshold (kept at 37.4°C for the project). The temperature is measured every wake up cycle and if the temperature is above the threshold then the algorithm is triggered to infer on the stored accelarometer data.

The device infers whether the body temperature is in a critical status. It has to consider that increased physical activity such as exercises are another cause for increased body temperature, besides a wound infection. Using a naive inference approach like threshold triggering therefore can result in false inferences. There is a lag between when a body is in exercise mode and the rise in body temperature which would give a false positive result. On the other hand, it may happen that a rise in temperature is due to wound infection but at the same time increased physical activity is detected by the accelerometer which would yield a false negative result.

For these reasons, a supervised machine learning approach is used to optimize the reliability of the inference. The tests were performed with Support Vector Machine (SVM), Relevant Vector Machine (RVM) and Random Forest Classifier (RFC) models. When training the machine learning models, data from the body being at rest resp. doing exercises is fed into the model. The exercise data was obtained by shaking the device by hand.

The feature vector dimension should be such that it has enough data points to generate meaningful distinction between accelerometer data from exercising and being in rest but it should be limited enough so it can be stored on the microcontroller's flash memory. Here the decision on feature vector dimension is made based on the assumption that the body temperature rises exclusively when the body is exercising for a certain period of time.

The acceleration data consists of readings of the 3 axes accelerations from the accelerometer. With the assumption that exercising for 15 minutes has an effect on the body temperature, the feature vector for the machine learner input should contain the data points from the last 15 minutes. To accomplish this, 5 samples are taken every 5 minutes. The microcontroller has a curricular buffer of size 45 which stores the data points from the last three cycles, hence it contains data points from the last 15 minutes. Therefore the feature vector has 15 samples where each sample will have values from the three axes of the accelerometer, i.e. $3 \cdot 5 \cdot 5 = 45$ data points.

Due to gravity, there is a constant offset in the acceleration measurement values for each of the three axes. The gravity offset is, respectively for each axis, removed by averaging 5 consecutive acceleration value readings and then subtracting this average from the individual samples. Any additional acceleration or deceleration is then considered as movement activity. The code to remove this offset is run every cycle before storing the values in the buffer.

In order to simplify the effort for the machine learner, in a second test the dimensionality (e.g. the number of features) of the data was reduced by preprocessing it with Principal Component Analysis (PCA). The obtained results are similar to those obtained before without PCA.

7. Results of the machine learning model tests

Since the movements of the device by shaking result in a sharp accelerometer measurement signal, all tested ML models identify body motion with unrealistically good results, e.g. zero false positive or false negative

rates in the confusion matrices. In future investigations, data from real physical exercises will have to be used with will then be more difficult for the ML model to identify and likely result in a lower ML model performance.

8. Deploying machine learning model on the microcontroller

The machine learning code is written in Python to test it on a normal computer in the first place. In order to export the tested code to the microcontroller, it is converted to C using the micromlgen[1] tool. The memory footprint of the different machine learning models was compared for different settings of the gamma value. The five models with the smallest footprint are RVC with gamma = 0.0001 or 0.001 which occupy about 2 kB of memory. RVC with gamma = 0.01 or 0.1 require about 2.6 kB whereas the next option based on SVC has a demand of approx. 11 kB.

Running the ML model is computationally expensive and requires much power which means that the frequency of invoking the ML algorithm on the microcontroller should be reduced. Therefore, the ML model is only run if the body temperature exceeds 37.4 °C.

9. Reporting inference to the PC

In the tests, the inference information were successfully transmitted via the BLE link and received on a PC. As the inference is run on historical data, a delay is expected from the point the movement type changes from "Rest" to "Exercise" or vice versa, and the point when the inference is reported to the PC. When a change of motion occurs, the inference is reported after all 45 values of the buffer have changed to the new motion type. For the demonstration this overall delay was 20 seconds. For the productive usage of the device, this delay is 10 minutes representing data points from the last three cycles with a 5 minutes period between each cycle.

10. Reduction of power consumption

Since low-power operation is crucial for the convenient usage of the device, we reduced the power consumption of the microcontroller as shown in Table 1 by disabling all unused functions. The sleeping level of the sleep mode is configurable and was set to power-down mode. The comparison between the current intake without and with different steps of power saving is shown in Table 1 as well. We expect that with the deployment of the low-power components mentioned above, the power consumption can be further reduced.

	Active mode current (mA)	Sleep mode (Power down mode) current (mA)	Volts
default settings	10	0.25	5
clock frequency reduced from 8MHz to 1MHz	3	0.25	3.8
voltage reduced to 3.6, change of voltage source	2.2	0.25	3.6
brown-out detection disabled	2.2	0.24	3.6
external pull-up resistors disabling UART0 permanantly and I2C in sleep mode	1.2	0.1	3.6
ADC disabled	0.9	0.0045	3.3
voltage reduced to 1.8 V	0.5	0.003	1.8

Table 1: Current intake of microcontroller in the test setup with different power save configurations

11. Average current intake

The average current intake is determined by measurung the current intake in the active and sleeping phase of a measurement cycle as shown in Table 1, as well as measuring the time duration of the active and sleeping phases as specified below.

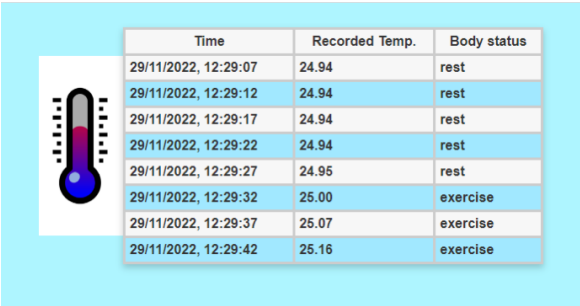
- The average current over active and sleeping times within a measurement cycle is computed.
- The time spent in sleep mode is 288 secs for all measurements.

[1]https://github.com/eloquentarduino/micromlgen

- Accelerometer reading time interval is set to 5 secs or 1 sec, clock frequency 8 MHz or 1 MHz.
- *Measurements for 5 secs accelerometer reading interval:*
  8 MHz clock: time spent in active mode approx. 6 secs, average current 0.1 mA
  1 MHz clock: time spent in active mode approx. 14 secs, average current 0.06 mA
- *Measurements for 1 sec accelerometer reading interval:*
  8 MHz clock: time spent in active mode approx. 1 sec, average current 0.037 mA
  1 MHz clock: time spent in active mode approx. 8 sec, average current 0.022 mA
- Expected battery life is approx. 80 days at 1 MHz and approx. 54 days at 8 MHz with a 200mAh coin cell battery

12. BLE receiver app

The BLE receiver app is a web application that runs on the web browser on any device. The Web Bluetooth API was used to connect to the BLE device directly from the browser. The device transmits the data as a byte array of size 69 and 73 for rest and exercise respectively. The data packet contains the temperature value with two decimal digits and the inference "rest" or "exercise". For the testing setup the inference is sent in each loop. In the final setup, the packet is only sent in case the device detects a rise in temperature when the body is at rest, in order to save battery power when the body is in a normal condition. The screen output of the app is shown in Fig. 5.



Time	Recorded Temp.	Body status
29/11/2022, 12:29:07	24.94	rest
29/11/2022, 12:29:12	24.94	rest
29/11/2022, 12:29:17	24.94	rest
29/11/2022, 12:29:22	24.94	rest
29/11/2022, 12:29:27	24.95	rest
29/11/2022, 12:29:32	25.00	exercise
29/11/2022, 12:29:37	25.07	exercise
29/11/2022, 12:29:42	25.16	exercise

Figure 5: Output of BLE receiver app

13. Conclusion and outlook

We have shown the basic feasibility of a wearable wound health monitoring device which infers the infection of a wound using a machine learner, by identifying increased body temperature occurring without physical activity. Due to the fact that shaking the device was used to simulate body motion, the ML's detection reliability was unrealistically good. In future investigations, realistic movement patterns need to be tested. Moreover, we built up the device with electronic components which were available on stock. By suitable configuration of the microcontroller, we minimized the power intake in order to achieve a long battery life. We proposed low-power components for a future enhanced design which will further enhance power efficiency.

References

[1] D. Puchberger-Enengl, C. Krutzler, M. Binder, C. Rohrer, K. R. Schröder, F. Keplinger, and M. J. Vellekoop, "Characterization of a multi-parameter sensor for continuous wound assessment," Procedia Engineering, vol. 47, pp. 985–988, 2012, 26th European Conference on Solid-State Transducers, EUROSENSOR 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705812043755

[2] H. M. Kimmel, A. Grant, and J. Ditata, "The presence of oxygen in wound healing," Wounds : a compendium of clinical research and practice, vol. 28, no. 8, p. 264—270, August 2016. [Online].
Available: http://europepmc.org/abstract/MED/27560469

[3] S. Milne, I. Seoudi, H. Hamad, A. Anoop, N. Allahverdi, Z. Zakaria, R. Menzies, and P. Connolly, "A wearable wound moisture sensor as an indicator for wound dressing change: An observational study of wound moisture and status," International wound journal, vol. 13, 11 2015.