Bachelor Project
Functional Biometric Authentication using Sound on Smart Devices

Johannes Waltmann

University of Duisburg-Essen
Matriculationnumber: 3029975
johannes.waltmann@stud.uni-due.de

Abstract. tbd

1 Introduction

During the last years the usage of smart-wearables (particularly smartwatches) has become more and more common with a number of 337 million units sold in 2019 and a forecast in sales of up to 527 million units by 2024 [2]. With that also comes a natural demand in data protection caused by the amount of sensors built in these devices and their ability to capture sensible personal data (e.g. health informations). Additionally smart devices can now also be used for many kinds of financial actions. Since most wearables are connected to either the distributors or the respective mobile phones virtual assistant they should contain sensors for on one hand sound conduction and on the other hand audio recording. In the current general smart-wearable design neither of these sensors is oriented towards the wearers arm. With a change in position of these sensors smartwatches become eligible for functional biometric authentication. Further motivation for this new authentication method comes from the fact that the previous classical methods (PIN, password...) are not suited for usage on smartwatches [17]. Nevertheless it is shown by Johnston [9] and Yang [19] in their respective works that there already are potential biometric authentication methods for smartwatches based on distinct hand or arm movements.

From these preconditions a prototype is defined. This prototype uses a sound based functional biometric which gets applied to its wearers arm.

2 Related Work

This section will give an overview on the terms of biometrics and its variations, authentication and also what defines an authentication system. Additionally an insight into other works on biometric or smartwatch based authentication will be provided.

2.1 Authentication

The term in general describes the process based on which a security systems tries to approve someone's claim of identity [4]. Based on the input type of this claim the term authentication can be further subdivided into explicit and implicit authentication. *Explicit* authentication is more often also known as traditional authentication [13]. This includes providing knowledge like PIN or password, using a token but also performing gestures, fingerprints etc. from the biometrical field.

Implicit authentication on the other hand describes mechanisms where a user does not provide a password, etc. directly. Instead users are authenticated based on observations of their behavioural patterns [8]. These observations are qualified for the use in e.g. biometrics since every individual has its own distinct habits which could be captured and analysed using different sensors [15]. Furthermore Shi, Jakobsson et al. state in their 2009 and 2010 works that implicit authentication is well suited for usage in combination with mobile smart-devices [15] or portable computers [8]. Based on this they propose three different application scenarios. First as a second factor in combination with passwords, second as the main authenticator and thus replacing the usage of a password and last as additional assurance or an extra trust factor when performing e.g. financial actions on a mobile device.

2.2 Biometrics

Biometrics (as in the Greek terms *bios* and *metrikos*) describes the utilization of an individuals physical traits or behaviour to clearly identify one from others. Contrary to the more known verification methods of PINs, passwords or ID-cards biometric identification does not rely on tokens or knowledge which could easily be forgotten or stolen, rather than unique personal traits like fingerprints, face geometry or the specific way someone interacts [7, chpt. 1.1][5]. Depending on the concept of usage a biometric system can be used in either verification or identification mode [7, chpt. 1.3]. These two modes can also be differed by the use of *positive* or *negative identification* techniques [16].

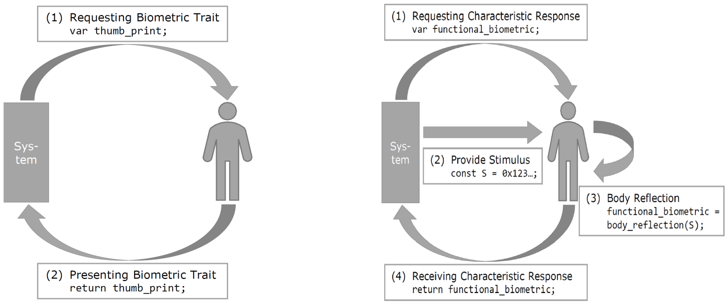
As already stated above biometric authentication uses an individuals personal traits as authentication tokens. Based on the kind of trait and the methods how they are provided the general term of biometrics can be further divided into behavioural and physiological biometrics. Also a new kind of biometrics called functional biometrics was introduced by Liebers and Schneegass in 2020 [12].

2.2.1 Behavioural Biometrics Behavioural biometrics refers to authentication systems in which the process of authentication is related to the behaviour of an individual [4]. In most cases this process is conducted with the use of primarily gestures or other actions or movements capable of being performed in everyday life [18]. Features which can be used for behavioural biometrics include e.g gait, keystrokes but also authentication patterns on smartphones could be used. One main advantage compared to physiological biometrics is that some behavioural traits must not be collected actively but can be captured whilst performing any kind of different tasks [18]. Also there is no definite need for special hardware since the sensors needed are mostly built in smart wearables which are one of the main users of behavioural biometrics [9].

2.2.2 Physiological Biometrics Apart from behavioural biometrics the classification of physiological biometrics is also existent. This form of biometric authentication uses the more ”static” traits of a users body as token such as e.g. fingerprints, hand geometry [3] or vein patterns [6].

A physiological system should also have a little higher accuracy than a behavioural one and it should be harder to use as an imposter since it is nearly impossible to identically copy a finger print, iris pattern, etc. [10], [5].

2.2.3 Functional Biometrics With functional biometrics another novel kind of biometric authentication was introduced lately by Liebers and Schneegass [12]. This new concept stands as a major influence to this work notably with its first implementation in SkullConduct [14].


(1) Requesting Biometric Trait
var thumb_print;
Sys-
tem
(2) Presenting Biometric Trait
return thumb_print;
(1) Requesting Characteristic Response
var functional_biometric;
Sys-
tem
(2) Provide Stimulus
const S = 0x123…;
(3) Body Reflection
functional_biometric =
body_reflection(S);
(4) Receiving Characteristic Response
return functional_biometric;

(a) Traditional Biometrics (b) Functional Biometrics

Figure 1. (a) shows the authentication process of a traditional biometric system where a characteristic response is created based on someone’s unique trait. (b) shows the order of steps for functional biometrics when a stimulus is added to create the characteristic based on body reflections [12].

In this category of biometric authentication the authentication system provides an additional stimulus during the enrolment phase. This stimulus is applied to the individuals body where it gets modified and afterwards captured (cf. Figure 1b). Due to different personal biological characteristics the modification of the stimulus is unique for each combination of user and stimulus. Requirements Schneegass et al. proposed are two hardware components in form of a Stimulus Generation Unit (SGU) and a Body Reflection Sensor (BRS). These two have to be designed as a dependence of the underlying biometric trait. Exemplarily when sound is used as the stimulus the SGU will be most likely some kind of microphone and the BRS a speaker. During the enrolment process described above stimulus and its transformation are saved as a secret two-tuple $(x, f(x))$ with x being the stimulus and $f(x)$ the transformation. Now when the related user wants to authenticate the stimulus is reapplied and it is expected to get $f(x)$ as response again. An additional security measure provided by functional biometrics is that when the stimulus gets leaked or lost the system is not fully compromised because the stimulus is only an exchangeable medium. The secret stems from the body reflection function which is unique to the user, unknown and hard to manipulate.

As already mentioned before the SkullConduct work by Schneegass et al. is one of the key influences to this work. Not just because it is one of the first realizations for functional biometrics but it also serves as a source of inspiration for this work. Schneegass et al. implemented a biometric authentication system using eyewear computers (e.g. Google Glass). Therefore they used the concept of bone conduction which is already frequently used by hearing aids. SkullConduct, in this case, uses the bone conduction speaker of the Google Glass to emit a sound sample against the wearers head. The sample which gets transformed due to the unique nature of each individuals head is captured by the glasses integrated microphone. Results of this study indicate that with all tested users SkullConduct had a probability of around 97% when it comes just to identify a correct user. Test of the system as an authentication tool showed an Equal Error Rate (EER) of around 6.9% in average but with significant drops the shorter the used sample gets (less than 1 second).

Other work on authentication via smart devices includes e.g. the works from A. Johnston et al. who implemented a smartwatch based authentication module that used gait recognition [9]. They adapted from previous work of theirs where gyro- and acceleration-sensors of smart phones were used to develop authentication methods [11]. The main thought behind the proposed use of gait authentication on smartwatches is that their place of wearing/usage is more consistent than the one of a smart phone and therefore more advantageous. Each participants dataset includes both data from gyro- and acceleration-sensor. Tests in regard of both design types showed that the general performance of authentication is way higher in average than the one of identification (e.g. 97.2% compared to 79.2%). Additionally the overall performances of the acceleration-sensor was higher than the gyro-sensor. Conclusions Johnston et al. drew from their results were that it is possible to authenticate someone sufficient enough using a smartwatch but they propose not to use the system for something other than a multi-modal biometric system at its current level.

3 Concept

With the introduction of functional biometrics, space was created for the development of new mechanisms and variations in biometric authentication. Contrary to other biometric systems which rely on e.g. fingerprints the body function used in functional biometrics cannot be leaked or reproduced that easily. This is due to the function being comprised of each individuals own body structure and bone density [12].

A function based biometric system authenticates its user by applying a stimulus to the users body. Based around the name-giving body-function a characteristic transformation of the stimulus happens which can be measured. This measurement can then be compared with a previously stored measurement using the same stimulus to authenticate the user.

An already existing implementation approach to functional biometrics is the SkullConduct head mounted device (HMD) introduced by Schneegass et al. [14]. There the stimulus used consisted of a sound sample being applied and recaptured by the HMD. One limitation coming with this exemplary work is, that with the authentication being reliant on the HMD it is based around a device with rather fixed location.

From this point of view it would be more useful to have a device in smaller shape e.g. the form of a bracelet, which could be worn on either arms or legs. It's functionalities could exemplarily be integrated into a smartwatch or other small wearable smart-devices. The authentication provided by the bracelet could be either used to secure data saved on the smart device or as multiple factor authentication for other connected devices like smartphones.

Possible stimuli for the bracelet could be sound patters which use either white noise, frequency changes or little melodies. The stimulus can then be applied to the wearer on a regular basis over the time the device is worn to ensure it is used by the correct user. Also, if the stimulus used gets leaked or corrupted otherwise, a functional biometric system does not need to be recalibrated completely. Since even the system itself does not really know the nature of the transformation function only a new stimulus and new sample data are needed.

The prototype realised in this work consists of a microcontroller which is used to generate and apply the stimulus as well as to capture the transformation. For this purpose it is connected to each a little microphone capsule and a speaker with small form factor.

4 Implementation

4.1 Hardware

The different hardware used included a microcontroller and it's accessories which were used to conduct the user-study and run the implemented functionalities. Additionally a casing was designed which fits the components described in section 4.1.1 and can be worn by the participants during the user-study.

4.1.1 Components The microcontroller used in this project is the LoRa 32 by Heltec Automation[1]. The controller uses an 32bit dual-core microprocessor by espressif (ESP32). Other features include onboard Wi-Fi and Bluetooth as well as an 0.97 inch OLED display. The controller can be powered using a micro-USB connection to a computer or an attachable lithium battery.

The implementation for the recordings is based around the use of an omnidirectional digital microphone[2]. This microphone uses a digital I^2S-Interface to process its input. For the output of the generated stimuli a miniature speaker with a mylar-membrane[3] is used. The speaker can conduct noise at up to 85 dB. Further used hardware includes an SD-Card Reader equipped with an 32 Gigabyte SD-Card which is used to store the generated audio-samples.

An overview of the connection between controller and the other components can be seen in Figure 2. An exact mapping of the single components can be taken from the code[4].

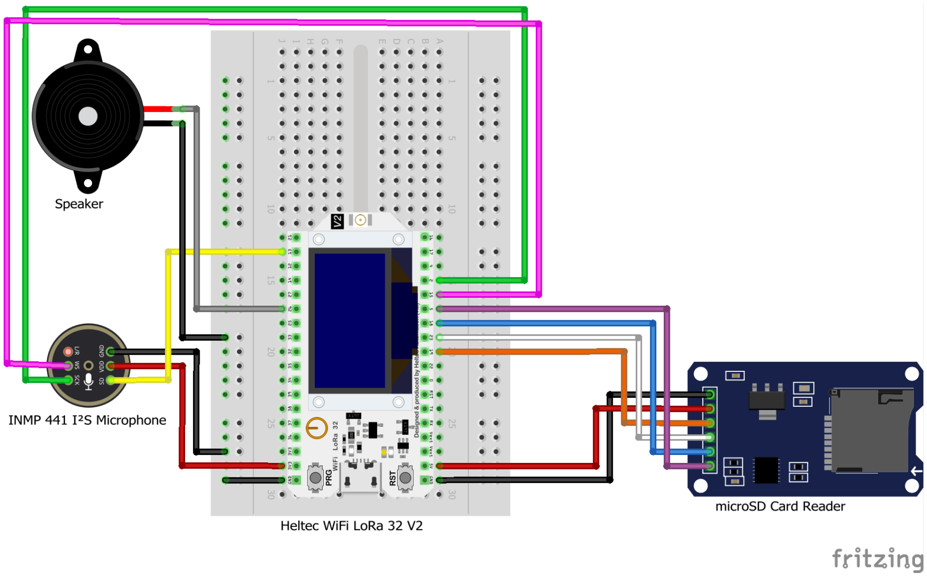


Figure 2. Pinout scheme of every component used in the project. The battery-pack which can be connected to the underside of the controller is not shown here for reasons of clarity.

[1] LoR Controller, https://heltec.org/project/wifi-kit-32/

[2] Ambility INMP441, https://www.amazon.de/Ambility-Omnidirektionales-\Mikrofonmodul-I2S-Schnittstelle-Precision/dp/B07MW95PSS

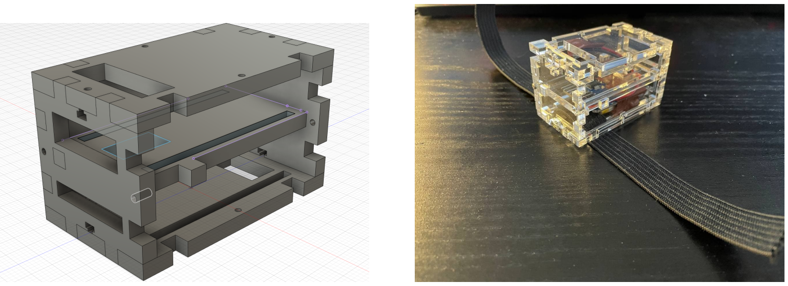
[3] LSF-15M/S, https://www.conrad.de/de/p/lsf-15m-s-8-ohm-0-8w-miniatur-\lautsprecher-geraeusch-entwicklung-85-db-0-500-w-1-st-710277.html

[4] URL zum Git einfügen

The hardware is powered by a lithium-ion battery with a capacity of 1200mAh and an output of 3.7V. It is connected to the controller on its underside using a JST-PH connector.

4.1.2 Casing To conduct the associated study it is necessary for the hardware to be wearable by the user. For this purpose a case consisting of multiple parts is designed. One part of the casing is designed to fit the microcontroller as well as the battery pack and the SD-card-reader. The other part is designed as a bracelet with two chambers one for each of microphone and speaker. Both parts can be worn by the participant using wristbands made from stretchable rubber.

For both parts the same specifications are used as they both are lasercutted from 4mm thick acrylic glass. The case for the microcontroller has a height of approximately 38mm with an extra plane in the middle at around 13mm measured from the inner border of the top-piece. The gap above this middle plane was designed to fit the microcontroller while the space below the plane is used to store the SD-card-reader and the battery.



(a) Microcontroller Case in design Form

(b) Microcontroller Case in printed Form //Need new image??

Figure 3. Image (a) shows the case as it was assembled using Fusion 360. The piece which would normally be located on the left side of the case is set to invisible so the interior can be seen. Image (b) shows the parts of the case after they have been lasercut and assembled. To easier fit the single hardware components the battery pack and the SD-card-reader are stuck to the bottom plane using tape. Image (c) shows the bracelet which is containing both microphone and speaker whith each of them placed in its own chamber.

Both the mid-plane and the bottom one have extra cut-outs to guide the wires which are connecting the single components to the controller. The cut-outs on top of the case and on its front are for the accessibility of the usb-port and the buttons located on the controller.

Since the box with the controller in it and the bracelet will not be worn directly next to each other the wires used to connect both parts are extended using two jumpers (with at least one female ending) and soldering them together. The joints are afterwards covered using scrubbed hoses. The same procedure is used for the wires of the SD-card-reader and the battery except the wires are shortened here and not extended.

// TO-DO: Bild von fertigem Prototypen, wenn alles connected einfügen

4.2 Development Tools

The majority of code in this project was written in the Arduino development environment (IDE). This open-source IDE is designed and optimized for the use with microcontrollers from the arduino family. It is also possible to add plugins for the support of other microcontroller chips like ESP32 or the Heltec LoRa-controller. Therefore it is necessary to include the functionality-packages provided by the chip developers under `"Tools > Board > Board Manager"` in the Arduino IDE. The IDE supports C and C++ as additional programming languages which can be processed by the microcontrollers.

For the design of the casing the software Fusion 360, published by Autodesk, was used. With this CAD-software it is possible to design shapes in either 2D- or 3D-format. These shapes can then be exported from Fusion 360 into the dfx-format which can be used to print the design to a 3D-printer or laser-cutter.

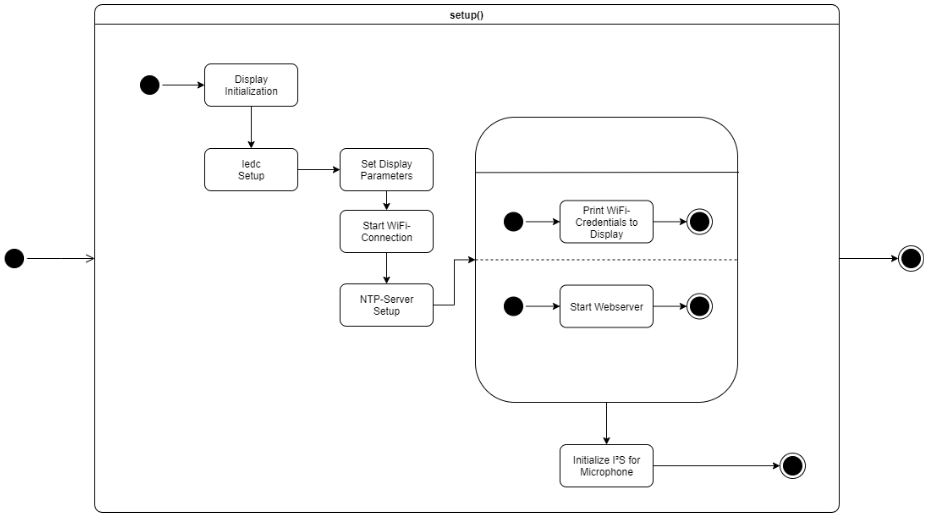


Figure 4. Individual steps the microcontroller executes during its setup ordered as Statechart.

4.3 Software

The code in this project is split up into multiple (logical) parts - the main .ino-file and additional .cpp files providing further code needed. The main file locates the default microcontroller functions as well as functionalities to generate the audio stimulus.

As seen in Figure 4 the setup functions main focus lays on initializing the different hardware components and hosting the socket for the provided web server. From this web server an interface is accessible which can be used to apply the stimulus and record its alterations. To capture the recordings the i2s interface is used.

Information needed by the user can be printed onto the controllers OLED display. For this purpose Heltec's version of the SSD1306 Library, which is included in the "`heltec.h`" package[5], is needed. It is located in the data included with the Heltec ESP32 board manager. The packages functionalities are accessible by including its header at the beginning of the sketches main-file.

To print something onto the display it is necessary to use the following statements first line:

```
Heltec.display->drawString(int x, int y, String text);
Heltec.display->display();
```

The parameters `x` and `y` represent the coordinates on the display starting on the top-left from where the string is to be printed and text being the content to be printed. Afterwards the method `display()` has to be called using the statements second line since just using the `drawString` method will only write the strings into a buffer but not print them onto the display. It is not necessary to call the method `display()` after each line but just once per block.

An additional feature which is implemented during the setup function is the connection to an NTP-Server. This server is later used to uniquely identify the recorded data by adding date- and timestamps of the recordings start to each file.

4.3.1 Stimulus Generation The approach taken for the generation of the stimulus is based around an external library [1] for the ESP microcontroller family. Using this library it is possible to play a soundfile asynchronous and thus record the audio at the same time using the exact same controller. The code excerpt shown in Listing 1.1 shows how a melody can be assembled and afterwards played using the library. The melody can be created using the statements seen in lines 6 and 8 with the parameters used in line 8 being a name for the sound to be played, the duration which the melody should take, the notes which are used (The notes can be coded either as a string representation or as integers. In the latter one the integers are the frequencies which are to be played). The last parameter states the number of individual notes which are to be played.

[5] https://github.com/HelTecAutomation/Heltec_ESP32/blob/master/src/heltec.h

```
1   if (run_melody) {
2     // Start Recording
3     Serial.println("MELODY START");
4
5     Serial.println("Loading melody...");
6     String notes[] = { "C4", "G3", "G3", "A3", "G3", "SILENCE", "B3", "C4"};
7     // Load and play a correct melody
8     Melody melody = MelodyFactory.load("Nice Melody", 2000, notes, 8);
9     player.playAsync(melody);
10
11    Serial.println("MELODY STOP");
12    Serial.println("RECORDING START");
13
14    startRecording();
15    Serial.println("RECORDING STOP");
16    run_melody = false;
17  }
```

Listing 1.1. The function which is used to scale the captured recordings volume.

In the last step the created melody can be played using the statement seen in line 9. Therefore a `MelodyPlayer` has to be declared and assigned to the DAC pin which is connected to the speaker. A melodyPlayer has access to either the method `playAsync()` or `play()` depending on the need of either synchronous or asynchronous play. Both of the play methods can be called either with a respective melody or parameterless. When no explicit parameter is given the melodyPlayer will then use the last melody which has been played whereas it will play the melody which is given as a parameter if that would be the case.

4.3.2 Recording the Sample With Inter-IC Sound (I^2S) an serial interface is used to generate the recordings needed for the study. I^2S works with a bus consisting of minimum three lines.

The lines which are mandatory include Serial Clock (SCK), Word Select (WS) and Serial Data (SD) but it is also possible to add further lines to the bus.

To work with an arduino or ESP32 based microcontroller it is necessary to include the I^2S-Drivers into the source-code. Based on this package the methods `i2sInit()`, `i2s_adc_data_scale()` and `i2s_adc()` operate.

Within `i2sInit()` the parameters needed by the I^2S framework are configured. Also the ports needed by the hardware are assigned and it is defined whether sound shall be recorded (the serial data param is set to `data_in_num`) or emitted (the serial data param is set to `data_out_num`).

The actual recording-process is controlled by the method `i2s_adc()`. There the input form the microphone is captured using the method `i2s_read()` from the I^2S package. It is used to read the data captured by e.g. a microphone into a predefined destination address. After defining the different buffer sizes needed during the recording process the input from the attached microphone is read and saved to a file. This process takes until a specified file size is reached. The file size is computed from amongst other parameters the sample rate, bits per sample but also the recordings duration.

Since the natural volume of the recording is relatively low when using only the `i2s_adc()` method each chunk of data can be scaled using the method `i2s_adc_data_scale()` before writing it to the filesystem. This scaling is done by multiplying each bit of the recording with a special factor as it can be seen in Listing 1.2 after performing a bitshift on the original values.

```
void i2s_adc_data_scale(uint8_t* d_buff , uint8_t* s_buff ,
  uint32_t len) {
  uint32_t j = 0;
  uint32_t dac_value = 0;

  for (int i = 0; i < len; i += 2) {
    dac_value = ((((uint16_t) (s_buff[i + 1] & 0xf) << 8)
             | ((s_buff[i + 0]))));
    d_buff[j++] = 0;
    d_buff[j++] = dac_value * 256 / FACTOR;
  }
}
```

Listing 1.2. The function which is used to scale the captured recordings volume.

The parameter which is represented by the placeholder ”FACTOR” in Listing 1.2 is a multiple of 1024 where the less the value is set the louder the recording will become (For this implementation the value 4096 is used). One little disadvantage of this method is that not only the recorded sound will be scaled but also the volume of other recorded noise is enhanced.

As it can be seen in Listing 1.1 a method called `startRecording()` is used to perform the recording duties. Within this method the .wav-file for the recording is created and saved to the SD-card with `i2s_adc()` being called afterwards to start the actual recording process.

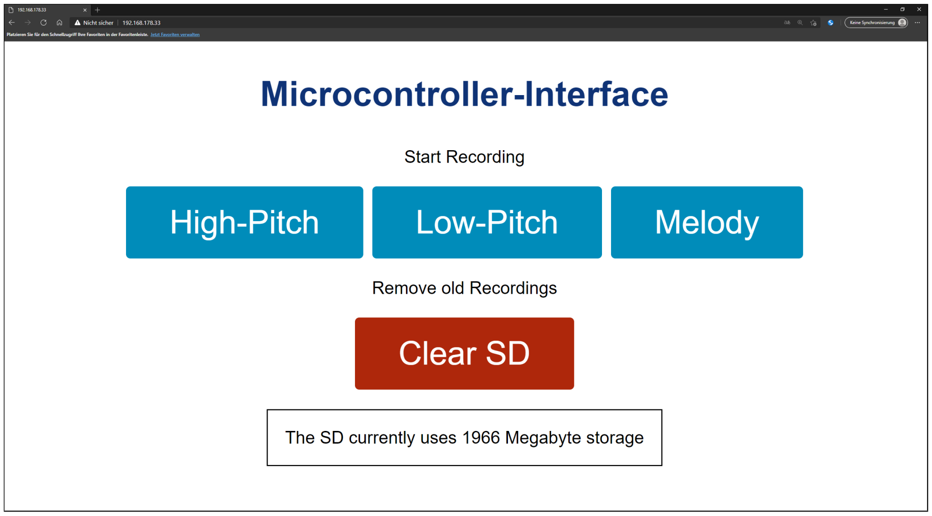
4.3.3 Webinterface To manage the functions of this project and start the recordings and playbacks a webinterface is used. It's implementation is located in the main .ino-file as a part of the loop-function.

As it can be seen in Listing 1.3 the socket listens for incoming clients and then processes the incoming requests as long as the client is connected. The first response sent to an incoming client is ”HTTP/1.1 200 OK” and the content-type of the response.

Additionally to this response code the client can process incoming GET-requests, e.g. when the button to start a recording is pressed on the webinterface. For this handling if statements like the following are used.

```
if (header.indexOf("GET /recording") >= 0) {...}
```

The last mandatory feature which is printed to the client during this first if-statement consists of several single `client.println()` statements containing the data for the web page. These prints have to go in between the last printed blank line (seen in line 25 of cf., Listing 1.3) and the if-statements to handle incoming requests as described above. The webinterfaces visual appearance was designed using a basic combination of HTML and CSS. It can be seen in Figure 5 displayed by the browser Microsoft Edge.



192.168.178.33
Nicht sicher
192.168.178.33
Platzieren Sie für den Schnellzugriff Ihre Favoriten in der Favoritenleiste. Jetzt Favoriten verwalten
Keine Synchronisierung
High-Pitch
Low-Pitch
Melody
Clear SD

Microcontroller-Interface

Start Recording

Remove old Recordings

The SD currently uses 1966 Megabyte storage

Figure 5. The Webinterface which is used to control the functions of the prototype as it is displayed by the Microsoft Edge browser. The three blue buttons can be used to start playing and recording one version of the soundstimulus each. The fourth button can be used to clear every existing .wav-File from the SD-Card.

Through the interface it is possible to on the recording of a new test sample and also to remove every recording which is saved on the attached SD-card. A third feature is located in the box on the pages bottom where the SD-cards space which is already in use is shown (represented by the character 'X'). The exact space is calculated by the method `getUsedSpace()` located in the requirements.cpp-file which then replaces the character 'X' on the interface.

```
void loop() {
  WiFiClient client = server.available();

  if (client) {
    currentTime = millis();
    previousTime = currentTime;
    String currentLine = "";
    while (client.connected() && currentTime - previousTime
    <= timeoutTime) {
      currentTime = millis();
      if (client.available()) {
        char c = client.read();
        header += c;
        if (c == '\n') {
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();

            // Handle incoming GET-Request

            // Print the content of the web page to be
            // displayed

            client.println();
            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }
      }
    }
    header = "";
    client.stop();
```

Listing 1.3. The part of the loop-function which is used to manage the connection of a socket hosted on the microcontroller to a web-client.

Other web-browsers which were used include Google Chrome and Mozilla Firefox with the latter one being the browser used most for testing the prototypes functionalities.

5 Evaluation

The following chapter will cover everything related to corresponding study. It will focus on the general aspects of it's design and conduction as well as the studies results.

5.1 Study Design

The conducted user-study followed a repeated-measures design in a controlled environment. It was held on two distinct days with every participant being tested on each day. This was done to ensure stability of the biometric recognition over time, as well as to make the authentication task more realistic. Each participant was tested with three different sound stimuli (high-pitch, low-pitch, mixed/melody) with every test being made 'X' times for 'n' seconds each.
To prevent possible carry-over effects the order of the three stimuli were randomised using a latin square matrix (cf., Table 5.1).

Table 1. Randomisation of the three different stimuli chosen for the study using a latin square matrix. E.g. the second participant would be tested first with a high-pitched sample, then the melody sample and last the low-pitched one.

P_{ID}	Order of Application		
	1	2	3
1	High-Pitch	Low-Pitch	Mixed/Melody
2	High-Pitch	Mixed/Melody	Low-Pitch
3	Mixed/Melody	High-Pitch	Low-Pitch
4	Mixed/Melody	Low-Pitch	High-Pitch
5	Low-Pitch	Mixed/Melody	High-Pitch
6	Low-Pitch	High-Pitch	Mixed/Melody

According to the proposed application scenario each participant had to record one full set of the three stimuli with the apparatus worn on each, the left and right arms wrist. To ensure that the bracelet is placed in the same areas on the second day the participants arms were photographed wearing the bracelet.

5.2 Procedure

- Prosaablauf eines Probanden - Participants gave written and informed consent und Demographics - All question answered - P could cancel the procedure at any time without detriments. - Danach Condition 1 entsprechend Latin Square gemacht - Armband gewechselt mit Unterstützung des Practitioners - Condition 2 - ... - Eventuell Fragebogen

vgl. http://florian-alt.org/unibw/wp-content/publications/liebers2021chi.pdf Kapitel 4

5.3 Results

- N Probanden, x Frauen und y Männer, Alter Median, Min, Max, Standard Deviation - Ergebnisse des Classifiers (=¿ Implementation in Kapitel 4 beschreiben)

5.4 Discussion

- Ergebnisse deuten – was bedeuten die Ergebnisse im Hinblick auf die Ideen/Hypothesen, die am Anfang der Arbeit vorgestellt wurden? - ”Our findings support functional biometrics. This can be seen due to ...” (im positiven Fall) - im negativen Fall: Woran lag es, dass was schief gelaufen ist?

6 Conclusion

References

1. Esp library supporting parallel played audio, https://github.com/fabiuz7/melody-player-arduino
2. Themenseite: Wearables (Mar 2020), https://de.statista.com/themen/3471/wearables/, retrieved on 07.05.2020
3. Alsaadi, I.M.: Physiological biometric authentication systems, advantages, disadvantages and future development: a review. international journal of scientific & technology research **4**(12), 285–289 (2015)
4. Bhattacharyya, D., Ranjan, R., Alisherov, F., Choi, M., et al.: Biometric authentication: A review. International Journal of u-and e-Service, Science and Technology **2**(3), 13–28 (2009)
5. Delac, K., Grgic, M.: A survey of biometric recognition methods. In: Proceedings. Elmar-2004. 46th International Symposium on Electronics in Marine. pp. 184–193. IEEE (2004)
6. Faltaous, S., Liebers, J., Abdelrahman, Y., Alt, F., Schneegass, S.: Vpid: Towards vein pattern identification using thermal imaging. i-com **18**(3), 259–270 (2019). https://doi.org/10.1515/icom-2019-0009
7. Jain, A.K., Flynn, P., Ross, A.A.: Handbook of biometrics. Springer Science & Business Media (2007)
8. Jakobsson, M., Shi, E., Golle, P., Chow, R.: Implicit authentication for mobile devices. In: Proceedings of the 4th USENIX conference on Hot topics in security. p. 9. HotSec’09, USENIX Association, USA (2009). https://doi.org/10.5555/1855628.1855637
9. Johnston, A.H., Weiss, G.M.: Smartwatch-based biometric gait recognition. In: 2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS). pp. 1–6. IEEE (2015). https://doi.org/10.1109/BTAS.2015.7358794
10. Koong, C.S., Yang, T.I., Tseng, C.C.: A user authentication scheme using physiological and behavioral biometrics for multitouch devices. The Scientific World Journal **2014** (2014). https://doi.org/10.1155/2014/781234
11. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Cell phone-based biometric identification. In: 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS). pp. 1–7. IEEE (2010). https://doi.org/10.1109/BTAS.2010.5634532

12. Liebers, J., Schneegass, S.: Introducing functional biometrics: Using body-reflections as a novel class of biometric authentication systems. In: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems Extended Abstracts. pp. 1–7 (2020). https://doi.org/10.1145/3334480.3383059
13. Ranjan, J., Whitehouse, K.: Automatic authentication of smartphone touch interactions using smartwatch. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct. pp. 361–364 (2016). https://doi.org/10.1145/2968219.2971370
14. Schneegass, S., Oualil, Y., Bulling, A.: Skullconduct: Biometric user identification on eyewear computers using bone conduction through the skull. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. p. 1379–1384. CHI '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2858036.2858152
15. Shi, E., Niu, Y., Jakobsson, M., Chow, R.: Implicit authentication through learning user behavior. In: International Conference on Information Security. pp. 99–113. Springer-Verlag (2010). https://doi.org/10.5555/1949317.1949329
16. Wayman, J., Jain, A., Maltoni, D., Maio, D.: An introduction to biometric authentication systems, pp. 1–20. Springer London (2005). https://doi.org/10.1007/1-84628-064-8_1
17. Xu, W., Shen, Y., Zhang, Y., Bergmann, N., Hu, W.: Gait-watch: A context-aware authentication system for smart watch based on gait recognition. In: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation. pp. 59–70 (2017). https://doi.org/10.1145/3054977.3054991
18. Yampolskiy, R.V., Govindaraju, V.: Behavioural biometrics: a survey and classification. International Journal of Biometrics **1**(1), 81–113 (2008). https://doi.org/10.1504/IJBM.2008.018665
19. Yang, J., Li, Y., Xie, M.: Motionauth: Motion-based authentication for wrist worn smart devices. In: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). pp. 550–555. IEEE (2015). https://doi.org/10.1109/PERCOMW.2015.7134097