Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

Wearable Motion Capture for Dancers

Bachelor's Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Matthias Sokolowski

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: 06.07.18
Submission date: 27.09.18

Eidesstattliche Versicherung

Sokolowski, Matthias	356445
Name, Vorname	Matrikelnummer

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel

Wearable Motion Capture for Dancers

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum

Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

Ort, Datum

Unterschrift

Contents

List of Figures

List of Tables

Abstract

In cooperation with artists of multiple artistic genres, we describe the development process of a wearable motion capturing system for a dance-music application. Eventually, the system acquires data from multiple motion sensors attached to a dancers body and transmits it to a central receiver. The data is processed and imported into professional sound manipulation software, where it influences music in real-time. The iterative development process of both hardware and software elements is described.

A minimally sized sensor node is constructed based on a Bluetooth enabled system-on-chip, which acquires motion data from an Inertial Measurement Unit and transmits it through the radio interface to a central host. Ultimately, the data arrives on a computer, where it is processed to produce output values suitable for sound manipulation. We will not focus on the music manipulation effects, but illustrate technical aspects in that context and describe the interface necessary to complete the system.

As part of our evaluation, we consider the application scenario of a stage performance enhanced by the motion system. For this, the artistical concept of connecting motion and sound effects is illustrated, and we will verify the conformance of the desired and achieved motion-sound junction.

The developed system is easily customizable allowing for a wide range of usage scenarios. In the course of the thesis, we will describe the implemented communication interfaces comprehensively, allowing for easy expansion or modification of individual components. Such possible enhancements will be pointed out at the end.

Überblick

Die vorliegende Arbeit dokumentiert den Entwicklungsprozess eines Bewegungserfassungssystems zur Anwendung in einer Musik-Tanz-Vorstellung. Das System wurde in Zusammenarbeit mit Musikern und Tänzern entwickelt und erfasst Tanzbewegungen mit Hilfe von mehreren Sensoren, die am Kostüm der Tänzer angebracht sind. Die Sensordaten werden an einen zentralen Empfänger übermittelt, am Computer in Echtzeit aufbereitet und umgerechnet. Letztendlich fließen die Daten in ein Musik-Effekt-Programm, welches die Musik während der Aufführung elektrisch verändert und künstlerisch erweitert.

Im Hardware-Entwicklungsprozess müssen besonders mechanische Anforderungen erfüllt werden. Das miniaturisierte Sensor-Board wird von einem Bluetooth System-on-chip gesteuert, welches die Daten vom Bewegungssensor ausliest und drahtlos an die Basisstation übermittelt.

Aufbereitet werden die Daten von einem Computer zur Generierung anwendungsspezifischen Datenformate die in Ton-Filter und Audio-Effekte einfließen. Die konkrete künstlerische Anwendung und die mit Bewegung verbundenen Effekte werden einführend beschrieben. Die Evaluation basiert auf der Qualität dieser beabsichtigten bzw. erreichten Verknüpfungen von Bewegung und Ton.

Besonderer Fokus beim Entwicklungsprozess des Systems lag auf Erweiter- und Skalierbarkeit. In diesem Zusammenhang werden die implementierten Kommunikationsschnittstellen hier dokumentiert, um eine einfache Anpassungen für anderen Anwendungs- bzw. Problembreichen zu gewährleisten.

Acknowledgements

First of all, I would like to thank my supervisor Anke Brocker for the great collaboration during the long development process of the project and mentoring of this thesis.

Thanks also to all other participants of our work, Paul Pankert, Audrey Apers and Gilles Doneux, for initiating the project and inspiring the development.

I thank the Media Computing group for providing me with this project and in particular Professor Jan Borchers and Professor Ulrik Schroeder for the examination of the thesis.

A special word of thanks goes to my friends at the FabLab for supporting me during countless hours we spent working there.

Chapter 1

Introduction

The presented thesis describes the development process of a wearable motion capturing system for an application in a dance-music-performance. Aim of the project is to construct a system that allows dancers to influence music and audio aspects and thereby expand the possibilities of an audio-visual performance.

In the following chapters, the hardware and software implementation will be illustrated. The functional and non-functional requirements are illustrated and solutions and design decisions discussed. A sensor system will be constructed, which transfers motion information of the dancer to a computer. This motion data is fed into a music manipulation program in order to generate audio effects, which are applied to live music and output by speakers parallel to the live music. In total this creates a bidirectional performance where musicians create a rhythmic base for the dancer and the dancer manipulates the audio arrangement.

The idea of connecting live music with dance motion originated from the composer Paul Pankert. Mr. Pankert is a violinist and composer of classical music and experimented in his recent work with computerbased sound manipulation for live music. Furthermore, the dancer Audrey Apers will perform with system on stage and was also involved in the development process. The technician Gilles Doneux from the Centre Henri Pousseur supported the digital audio composition.

Chapter 2

Related work

Although our project is focused and inspired by one specific use case, the problems and solutions discussed also appear in other fields of study including health, sports and gaming.

In a study by Edgar et al. [2012] acceleration and pressure data acquired at the dominant arm's wrist and one shoe was used to identify common household and athletic activities (see Figure 2.1). Since such activities are performed over a relatively long timeframe of several minutes, in this application a low data sample rate of 10 Hz was sufficient. The data was then evaluated by a neural network, which was eventually able to identify the performed activity with high accuracy. Inspired by this, Edgar et al. points out the possibilities of biofeedback systems in context with body weight management and stroke rehabilitation.

Similar healthcare purposes are illustrated by Heiduk et al. [2018]. In their study, an inertial sensor network is described, capable of accurately monitoring sports activities. The described system can identify physiological issues during training, provide health data collection and injury prevention resulting in a reduction of overall health care costs.

Other sports related studies focus on training and skill improvement. In a study by Okamoto [2012] an exercise support system is constructed based on wearable acceleration

(a) (b) (c)

Figure 2.1: Acceleration sensor nodes attached to the dominant wrist (a) and shoe (b). The sole pressure sensor (c) connects to the shoe sensor node.

sensors. Notably, the wearable system is compared to a visual motion capturing system and the batting technique of professional and amateur baseball players was evaluated. Okamoto shows that acceleration sensors provide similarly qualitative data compared to the more established visual technique. Ultimately, Okamoto concludes that the wearable solution is more feasible in a consumer product scenario due to a number advantages including convenience, complexity and cost.

Großhauser et al. [2012] developed a sensor system similar to ours in order to provide a closed-loop feedback channel for dance training. The data acquisition is focused on the dancers legs and consists of a goniometer measuring the knee's angle, an accelerometer and gyroscope combination attached to the foot and a pressure sensor measuring the weight balance on the sole. Utilizing sound synthesis algorithms, an audio feedback is generated as well as output to the dancer. In their application, the additional perception channel gives the dancer new, continuous live-feedback about the performance, improving training results and learning quality. The study describes several technical requirements and design choices, many of which are equally relevant for our system and will be discussed in the following chapters.

Fujimoto et al. [2009] use accelerometers on dancers shoes to play music according to dance steps. For this, the dancer records the motion sequence of dance steps into the sys-

tem and connects them to sound snippets in a scripting language. During the performance, the system compares live motion data with the stored samples based on Dynamic Time Warping. This algorithm measures similarity between both motion sequences and eventually identifies the step event. Caused by the step event, a sound is played according to the script previously programmed by the dancer.
The system aims to provide a new expressive element to the dance performance and was applied and evaluated with break-dance steps/performances. A relevant detail, which we will come back to later, are Fujimoto et al. studies on the negative impact of delay between the physical motion event and the played sound.

In conclusion, all these studies produced results which are partially overlapping with our field of work. However, all of them separate in key aspects like project scope, sensor technology, sensor placement or performance. A complete overview about the requirements of our system will be provided in section 3.3, where the final system design will partially be coherent with aspects mentioned above, but also addresses the disparities with custom solutions.

Chapter 3

Design and Development

This chapter describes the iterative design process of the project. We will start by illustrating our initial prototype and later develop more sophisticated elements of the system. Each problem and its solution will be described in detail and ultimately will lead to the final product.

3.1 First Prototype

Since the artists with whom we have been working together were unfamiliar with the technical subject, we attempted to convey technical possibilities and limits early on in the project. Thus, we constructed an immature prototype based on off-the-shelf prototyping hardware, namely Bluetooth modules, Arduino boards and a sensor breakout board.

In more detail, the sensor node was constructed using an MPU-6050, an AVR microprocessor on an Arduino Feather breakout-board and a HC-05 Bluetooth module. The MPU-6050 is a 6-DOF IMU consisting of an accelerometer in combination with a gyroscope and thus provides translatory and rotational motion data. The AVR microprocessor con-

DOF, Degrees of freedom
IMU, Inertial measurement unit

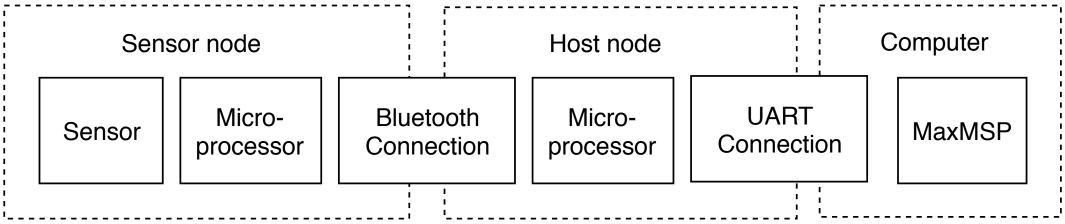


Figure 3.1: System profile of the first prototype

tinuously reads these sensor values and transmits the data through the HC-05 Bluetooth module to a receiver node. The receiver node is based on the same microprocessor and Bluetooth arrangement and forwards the data to the computer through a serial connection. The complete arrangement is illustrated in Figure 3.1.

MaxMSP is discussed further in subsection 3.5.5

On the computer we setup MaxMSP, which is signal processing environment often used by music artists. In MaxMSP the data received from the sensor is read from the serial connection and exemplified by visual and audible data representation.

After discussion on the prototypes fundamental abilities the general project scope was defined more closely with specific concepts about data processing and hardware design.

3.2 System Design

Since the first prototype proved to be generally successful, there was no need to change the fundamental system design. At this point we also decided against any visual motion capturing solutions. Such systems - as used in movie and video game industry - are based on a number of cameras tracking one or multiple individuals. However, these systems are highly complex, require trained operators and are not designed for stage use.

Therefore, the design as illustrated in Figure 3.1 remains mostly but with the the following two extensions.

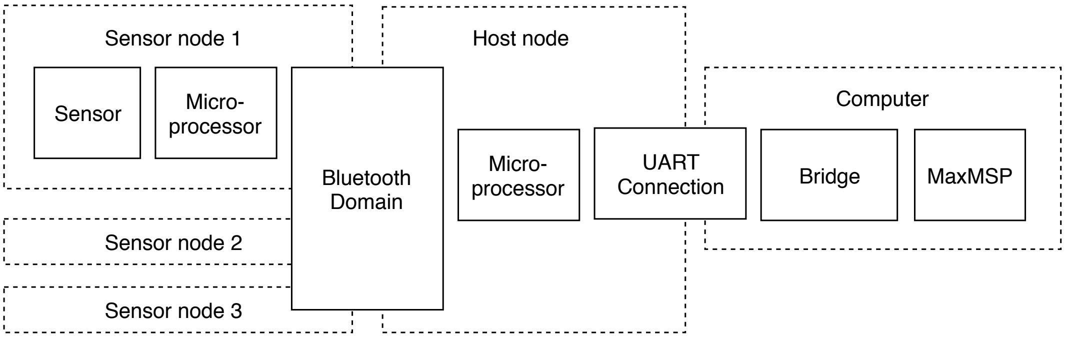


Figure 3.2: Complete system profile

The first significant adaption is an additional software element between the serial connection and the music manipulation software MaxMSP (refer to Figure 3.1). This software element will consume the raw motion data received from the sensor system before passing the processed information onwards to MaxMSP. Primarily, this stage will process the rotation information and generate an intuitive output format understandable for the artists. However, also other tasks like zeroing of the sensors, filtering and monitoring can be addressed in this stage. This bridge element will be discussed in detail in subsection 3.5.4.

The second adaption to the system design is the usage of multiple motion sensors. Since we aim to monitor motion of separate body parts and also relative motion - e.g. the interior angle at an elbow - multiple sensors at different locations are required. These sensors can be interconnected either by a wired connection on the dancer's body together with a single radio connection to the host or by individual radio connections from each sensor to one common host.
We selected the later option in order to avoid cables and connectors on the dancer's body. This decision will be described in more detail during evaluation in section 4.1.

A complete system overview is illustrated in Figure 3.2.

3.3 System Requirements

Before focusing on the individual elements and engineering solutions, the general requirements and design considerations are presented. Some of these were already observed in other studies, referred to and mentioned in chapter 2. Especially, this includes mechanical requirements for the sensor nodes which have to be small, light-weight, unobtrusive and robust. In this way the dancer is not restricted or handicapped by the system.
Further considerations are safety and usability. In a sports application, sharp corners on the device might puncture clothing, are uncomfortable or hurtful. Additionally, the system should be simple to attach with minimal effort and reliable for stage use.

Next to these non-functional aspects, further functional requirements have to be met. Ideally, the data acquisition and transfer should be fast with minimal latency and high sample rate. The transmission chain ends at the computer, where the data is eventually delivered to a music manipulation software or another application program. In our case, the output data is consumed by the music processing software MaxMSP.

As the project was constructed for a specific application, some minor requests by the artists and customizations were addressed, e.g. the final data output and format. Nevertheless, we aimed to keep the individual elements encapsulated and thereby the project adaptable for other applications, use-cases and scenarios. The constructed hardware allows for extensibility with yet unused features already provided. All communication interfaces are well defined and described in the following. The software solutions were developed in a modular approach and are easily customizable.

3.4 Hardware Design

This section covers the development of the two main hardware components in the system being the sensor board and the host receiver. For this, physical aspects regarding the radio transmission and sensor technology are considered and key component characteristics outlined.

3.4.1 Radio communication

One important element of the system is the radio transmission line between the dancer and the receiving host. For this a radio frequency band has to be selected from the available ISM bands. The industrial, scientific and medical (ISM) radio bands are free to use with no need for licensing or authorization. Commonly used ISM frequency bands are 433 MHz, 868 MHz and 2.4 GHz. The later is the broadly used frequency band for 2.4 GHz WLAN and Bluetooth, while the others are normally used for home appliances and alarm systems.

For selecting a frequency band, we have to consider multiple arguments. At first, higher radio frequencies utilize wider bandwidth, which corresponds proportionally to a theoretically higher data rate[1]. This makes the 2.4 GHz band much more preferable compared to the sub-GHz alternatives.

Another aspect to consider is the congestion on the individual bands. As WLAN and Bluetooth are widely used technologies, a moderate amount of congestion has to be expected. Transmissions by other parties may interfere with our own, which eventually results in transmission errors. Necessary retransmission or missing data obviously reduces the effective throughput.

At last, available protocols and communication mechanics should be evaluated. Both sub giga-hertz bands have no in-

---

[1] This fundamental relation can be seen from the basic transmission line capacity formula by Shannon [2001].

herent protocols meaning the transmission scheme is freely defined by the individual appliances. This is both an up- and downside as it allows for a well-matched protocol solution but also causes the necessity of implementing or adapting such a one.
WLAN implies the Internet Protocol and hence numerous high-level protocol standards. Bluetooth provides high-level transport features already within its own specification. Furthermore, Bluetooth is designed for ad-hoc networks and setup. WLAN requires essential administrative effort both in hardware and user configuration, e.g. router, DHCP server, network setup and selection.
Last, it should be considered that the wide usage of these technologies increases the scope of the system in general as it allows adding other, already established devices to the system based on the widely distributed, integrated hardware, e.g. in smartphones and laptops.

Ultimately, due to a multitude of advantages, Bluetooth was chosen as radio carrier.

3.4.2 Sensor node

The primary task of the sensor node is to gather motion data and dispatch it to a Bluetooth connection. As Bluetooth is a complex standard covering multiple communication layers, the Bluetooth radio module is often paired with a programmable processing unit to accomplish both radio and user tasks. Besides some miscellaneous peripheral parts and a battery power solution, the motion sensor and Bluetooth module already complete the essential components on the sensor board.

As a motion sensor, the BNO055 was selected. Similar to the sensor used in the prototype - the MPU-6050 - the BNO055 merges multiple sensors into one package. As an advantage compared to the MPU, the BNO features an additional magnetometer (besides accelerometer and gyroscope) and conveniently offers on-chip sensor fusion.
As pointed out previously, the accelerometer monitors translatory motion and the gyroscope rotational motion.

The magnetometer measures the magnetic field of the earth, which reduces sensor drift of the gyroscope and allows for absolute orientation reference.
Besides providing the raw data of all three sensor units, the BNO is capable of fusing these together. In fusion mode the BNO calculates its absolute or relative orientation by combining information from all three sensors and thus reducing uncertainty and sensor drift. The orientation is then provided as a quaternion, a four dimensional vector format fully describing rotations in 3-dimensional space (more detail in subsection 3.5.1).

Bluetooth capabilities are provided by the BL652, a Bluetooth Low Energy module based on the nRF52832 SoC with radio circuitry and antenna on board. The module has a minimal PCB footprint ($10\,\mathrm{mm} \times 14\,\mathrm{mm}$) with no peripheral hardware requirements. It offers all common hardware communication interface and is programmed based on the extensive Nordic SDK.

SoC, System-on-Chip

SDK, System Development Kit

Lithium-ion polymer batteries (LiPo) are used as the power solution of the sensor board. A single cell provides sufficient supply voltage of $\sim 3.7\,\mathrm{V}$ and high-energy density. Additionally, single cell LiPos are available in a wide variety of capacities and form factors.
Commodity cells often come with a minimal protection circuitry and cable with connector already attached. The established connector is the JST-PH-2[2], for which the corresponding socket is placed on the sensor board. The connector is safe against polarity reversal and locks due to insertion friction.

No additional power switch or input buttons are provided. A sensor node is active whenever power is supplied and acts autonomously with no user input. This reduces board size and possibilities for mechanical failure. In addition, accidental or unintended inputs during a performance can be avoided.

---

[2] Japanese Solderless Terminals, PH series, 2 terminals

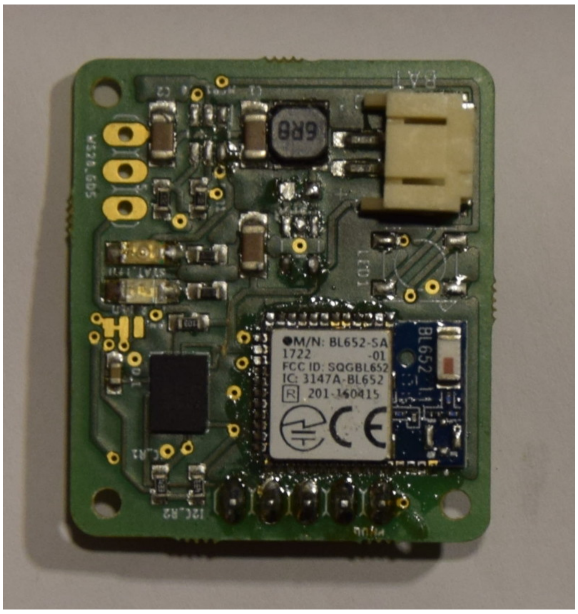


Figure 3.3: Sensor board populated with Bluetooth module (blue, lower right corner) and BNO (black, lower left corner).

As an additional feature, an RGB LED together with a 5V-step-up circuitry was added to the design. The WS2812B LED is used, which offers 24-bit RGB color range with no peripheral hardware due to an embedded LED controller. However, blue diodes require a relatively high forward voltage, and for reliable, true-color operation the WS2812B specifies a supply voltage of 5V. As this exceeds the battery voltage, a step-up circuitry is added to bridge the gap.

The populated sensor board is shown in Figure 3.3. No components were placed on the bottom side of the PCB, which allows the board to be mounded based on adhesive Velcro applied to the bottom side. This way, the sensor board can be attached to a costume or alternative fixings. One possible option based on elastic straps is shown in Figure 3.4. The schematic and board layout is provided in Appendix A.

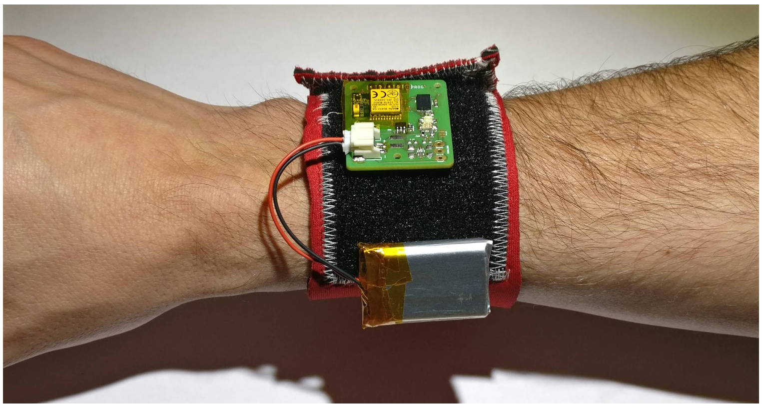


Figure 3.4: Sensor board together with battery strapped to the wrist.

3.4.3 Host node

In general, any other Bluetooth hardware can communicate with the sensor nodes. For example any smartphone or laptop with integrated Bluetooth module might connect to the sensors and acquire their data. In this project however, we aim to explore and test the possibilities and limits of the hardware when used in the dance application context. The use of custom, distinct hardware for the host allows for experimentation with the physical hardware as well as the close-hardware software options and implementations. A custom host device also avoids hardware implementation on the computer, thus avoiding driver errors, cross-platform requirements, and reducing the implementation complexity. Furthermore, it physically separates the receiver from the host, which will be beneficial for stage use later.

The hardware of the host node is minimalistic as it only has to forward received data to a computer. Therefore, the host device only consists of a single Bluetooth module and a USB-to-serial-interface. The later is provided by an Arduino board and the BLE module connects to this from a piggyback board. The Arduino board has a well established USB-serial-converter with widely supported drivers on all operating systems. It can reliably provide power from both

(a) (b)

Figure 3.5: Host node hardware implementation. (a) Populated host piggyback board with included charging connector (lower left corner). (b) Completed host device packaged and connected to antenna and computer.

the USB connection or possibly an external power pack. This power source is also utilized for a LiPo battery charging circuit together with the appropriate JST connector. This allows charging of the used batteries on the host node while the system is not actively used.

The BL652 module is available in two configurations. One provides an integrated on-board antenna and is used on the sensor board, the other features a connector for an external antenna. The second option is used for the receiving node and will be provided with a external antenna, which will improve the radio reception due to the increased size.

The internal hardware of the host node is shown in Figure 3.5(a) and the final device as used on stage is displayed in Figure 3.5(b).

3.5 Software

In this section, the software elements of the system are described. This also contains a number of communication interfaces in between the individual programs and firmwares. At first, we describe the data format of the IMU at the sensor node and continue with all components up to the sound manipulation software.

3.5.1 BNO sensor data

Microprocessor and IMU are connected by an *Inter-Integrated Circuit* bus (in the following referred to as I2C). The BNO specifies the bus communication as the I2C common register map scenario. Sensor data can be read from and configuration written into this register table.

Depending on the configured operation mode, various output values are available. The most advanced output mode is the nine-degrees-of-freedom fusion mode utilizing all three elemental sensors and outputting rotation and translatory acceleration data.

The sensor values correspond to a right-handed Cartesian coordinate system where the Z-axis is pointing up. Relating to that coordinate system, the acceleration data is provided by an X-, Y- and Z-component, which can also be interpreted as an acceleration-vector.
The rotation of the sensor can either be absolutely referenced to earths magnetic north pole or relatively to the initial orientation on reset. The rotation is read form the sensor as a quaternion, a four dimensional vector format describing rotation in three-dimensional space. A quaternion, also referred to as versor, is a unit vector $Q = (W \quad X \quad Y \quad Z)$. Without going into the arithmetic details, when Q is applied to a vector $\vec{v}$, $\vec{v}$ is rotated W radians around the rotation axis $(X \quad Y \quad Z)$. We will later use these vector operations when processing the data.

In total, the microprocessor reads the following numbers

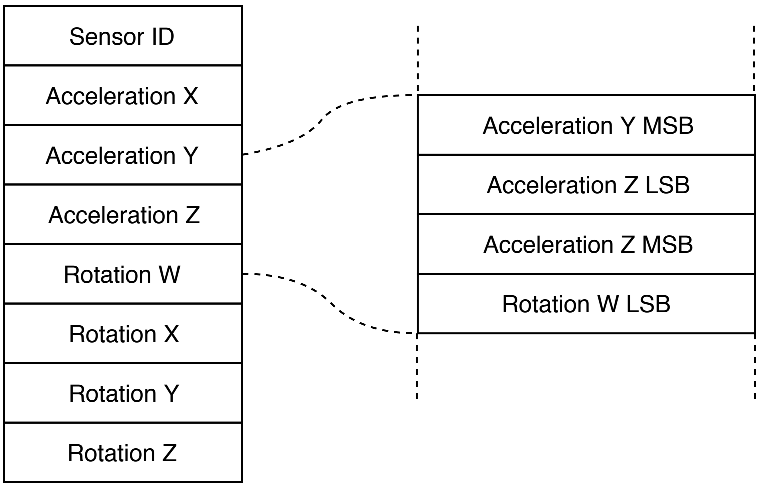


Figure 3.6: Sensor data communication format. Sequence order (left) and detailed byte representation (right).

from the sensor: Acceleration in X, Y and Z direction, Quaternion values W, X, Y and Z.
Each of the seven numbers is provided by a 16-bit wide signed integer in Little-Endian byte order. For acceleration data, the precision is $0.01\,\mathrm{m/s^2}$ per least-significant-bit.
Rotation quaternions are unit vectors and not associated with a physical unit.

In total, the relevant sensor data consist of 14 bytes, which are transmitted from sensor to host. Additionally, the sensor node adds two bytes to the beginning of the data sequence, which are deterministically calculated from a chip unique identifier. Later, this will allow for persistent distinction and identification of sensor nodes.
The total data sequence is illustrated in Figure 3.6.

3.5.2 Bluetooth transmission

Bluetooth is an extensively specified standard covering multiple communication layers and aspects. The application layer specified for Bluetooth communication is the GATT (Generic Attributes) system. Utilizing this foundation, a communication system was implemented for our purpose which is explained and documented in the following. Many technical terms of this description refer to the Bluetooth specification of the Generic Access Profile (GAP) and the Generic Attribute Profile (GATT).

The sensor node constantly advertises itself as a peripheral when no connection is established. A dedicated, custom service for motion data is advertised, which consists of a single notification characteristic. Eventually, a central connects to the sensor node and subscribes to this characteristic. The sensor node constantly samples data from the IMU and issues data by updating the characteristic and sending a notification. In our case, the transmitted data has a fixed length of 16 bytes, but the payload format and length can vary depending on the scenario.

3.5.3 Host

The host node implements the Bluetooth central role and connects to any available sensor nodes. It subscribes to the single data characteristic and forwards any incoming sensor data through a serial connection to the computer.

However, since UART barely covers the Data-Link-Layer[3] additional packetizing on this transmission element is required and implemented by standard character stuffing.

Standard character stuffing is performed as follows: The transmitted payload is framed by a start (STX[4]) and end flag (ETX[4]). In order to distinguish the control characters STX and EXT from payload, they are marked with an additional preceding escape byte (DLE, data link escape). Whenever the DLE byte occurs in the payload, it is stuffed with an additional DLE byte.
As control characters, their standard byte values from the ASCII table were selected; giving `DLE = 0x10, STX = 0x02, ETX = 0x03`.
No further control characters or protocol is specified for the

---

[3] Data-Link-Layer (DLL) referring to OSI protocol layer model

[4] STX refers to "Start of text" or "Start of transmission", ETX respectively to end

host implementation and any data sent to the host is discarded.
The underlying UART connection is clocked at 115200 Baud with 8 data-bits, using no parity and one stop-bit.

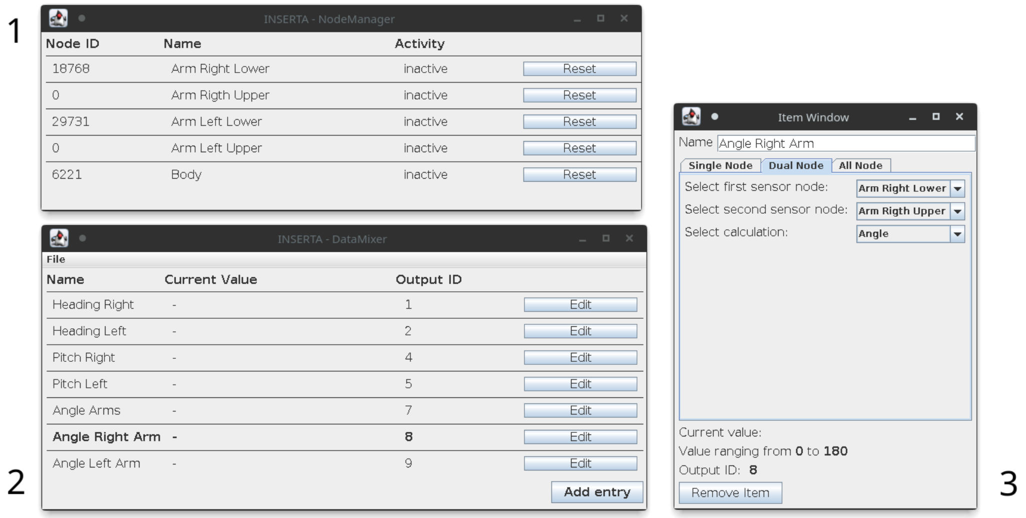
3.5.4 Bridge

In order to process the received sensor data and apply potential filters or mathematical manipulations, an additional bridge element between the receiving host node and MaxMSP is required. This software tool allows configuration of the system by managing the sensor nodes and their associated calculations. The two primary tasks of the bridge are monitoring the sensors and composing output mappings.

When working with rotation data, it became necessary to reset the orientation to a neutral position respectively zeroing out the sensor node. This is achieved by buffering the orientation when zeroed and applying the inverse rotation to following data.

In the following, the calculations generating a single output value will be referred to as a *mapping*. A mapping can be based on one or multiple sensors and produces an integer output value, which is later transferred to MaxMSP. Some basic mapping calculations are described in section 3.6 and section 4.3 illustrates their concrete application.

In order to generate output values effectively, a data mapping framework has been implemented, which allows developers to abstractly define mapping calculations and users to configure them.
A mapping can utilize both just incoming and stored data. The user configures these mapping based on predefined options and specifies which sensor nodes are queried. Eventually, the mapping outputs an integer value which is transmitted to MaxMSP. An overview of this use case is exemplary pictured in Figure 3.7.



Node ID	Name	Activity	
18768	Arm Right Lower	inactive	Reset
0	Arm Rigth Upper	inactive	Reset
29731	Arm Left Lower	inactive	Reset
0	Arm Left Upper	inactive	Reset
6221	Body	inactive	Reset

Name	Current Value	Output ID	
Heading Right	-	1	Edit
Heading Left	-	2	Edit
Pitch Right	-	4	Edit
Pitch Left	-	5	Edit
Angle Arms	-	7	Edit
Angle Right Arm	**-**	**8**	Edit
Angle Left Arm	-	9	Edit

Name: Angle Right Arm
Select first sensor node: Arm Right Lower
Select second sensor node: Arm Rigth Upper
Select calculation: Angle
Current value:
Value ranging from **0** to **180**
Output ID: **8**

Figure 3.7: Bridge application in use. (1) Sensor monitoring, (2) Mapping overview, (3) Mapping configuration.

3.5.5 MaxMSP

MaxMSP is a graphical development tool intended for music and multimedia processing. In our final theater-stage application, it will be used to provide sound effects and manipulations, which are influenced by the acquired sensor data respectively the produced output values.

MaxMSP, more precisely *Max/MSP*, are basically two separate software elements. Firstly, the *Max* development environment primarily provides a graphical programming system designed for real-time processing. Secondly, the *MSP* component - Max Signal Processing - is responsible for sound signal manipulation, which is also the originally intended use case of the system.
Over the past, also other components/expansions for Max have emerged adding further features like graphic processing and expanding the tool to a broader community. However, although Max is generally not use case specific, most of the elements are biased to a single intended application.

Thus, we cannot rely on existing Max building blocks when injecting the output values from the bridge software to

MXJ, Java in Max

MaxMSP. Instead we need to construct a custom Max Object [5] utilizing MXJ, an interface allowing Java integration in Max. Based on a Java class conforming to the MXJ specification and utilizing the corresponding library, we constructed the `mxj data` object, which can receive data from the bridge through a local UDP connection and output it for further use.

The desired output value can be selected by definition of an ID attribute, which can select from the list of output values defined in bridge application. The output value is provided as integer and provided on the Object's first and single outlet, the Max Object's output interface.

3.6 Mappings

In this section a number of output mappings based on the orientation of one or multiple sensor nodes is described. These mappings were particularly interesting for our stage application, but in theory also use of the acceleration data or combinations of both data domains is feasible.

A relevant aspect for our study is the dancer's arm posture, more specifically the pointing direction and twist of the arm. For this, a single sensor node attached to the dancer's wrist is sufficient, which transmits its relative rotation as a quaternion. Based on this incoming data, a more intuitive rotation format based on Euler angles is calculated. Euler angles describe a rotation based on three angles, each describing the rotation around a Cartesian axis. However, the straightforward mathematical conversion from a quaternion to Euler angles is not suitable for our purpose as it proved to be inconsistent and instable with rapid changes of the output values. Therefore, the following calculations are performed to extract the three rotation angles yaw, pitch and roll.

As a first computation step the rotation R in form of the received quaternion is applied to two orthonormal reference

---

[5] Program building blocks are called Objects in Max

vectors, in this case the normal vectors of X- and Y-axis. The resulting vectors $\vec{p}$, pointing indicator, and $\vec{r}$,roll indicator, fully represent the orientation of the sensor.

$$\vec{p} = R \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \qquad \vec{r} = R \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{3.1}$$

Yaw represents the horizontal, the cardinal direction of the sensor. In spherical coordinates this corresponds to the polar component. The yaw angle is calculated by projecting $\vec{p}$ to the X-Y-plane and measuring the angle to the X-axis in the plane. This angle cannot be measured directly from the scalar product since this only measures the *enclosed* angle. However, when measuring the angle from the X-axis to the project vector in a mathematical positive sense, values larger than $180°$ can occur. This case can be identified by checking the sign of the Y-component of $\vec{p}$, where a negative value indicates yaw angles larger than $180°$. Based on this information the value from the scalar product is corrected.

Pitch describes the sensor pointing up and down, which corresponds to the azimuth component of spherical coordinates. The pitch angle is calculated from the pointing indicators angle to the X-Y-plane. Similar to the yaw calculation, $\vec{p}$ is projected to the plane and the angle between projection and $\vec{p}$ is calculated as before based on the scalar product and here the sign of the Z-component.

Last, the Eulerian roll angle describes the rotation around the pointing direction. Again, comparing this to a vector described by spherical coordinates, the roll corresponds to the rotation around the vector itself.
The following mapping, however, does not calculate roll in the a mathematical sense, but has a intuitive, physically based definition. The roll mapping measures whether the dancer's palm is facing up or down, which can be derived from the angle between roll indicator $\vec{r}$ and the Z-axis. Although, this is mathematically not accurate and the original rotation is not accurately described by combinations of these three mappings, those definitions proofed to be intuitive and expressive for the artists.

The last, rotation based mapping is the measurement of the enclosed angle between two body parts, e.g. the angle at an elbow. This mapping is simply calculated by the scalar product of two sensors pointing indicators.

Chapter 4

Evaluation

All hardware and software components as well as the associated design choices were regularly presented to and discussed with the artists. In this process, the participants were interviewed about the overall content of the project and possible improvements or requirements guiding the next development steps.

In the following, we will reflect on the design decisions based on technical analysis, user opinions and field use.

4.1 Hardware Evaluation

Regarding hardware construction, no negative feedback has been issued by the artists. Also, upon further questioning the dancer did not express concerns regarding physical obstruction.

Nevertheless, we evaluate the design decisions of hardware construction. At first, the system utilizes a number of separate sensor nodes each working autonomously with individual processing unit, battery and radio connection. Opposite to this concept is a system consisting of a centralized processing unit connecting to a number of remote sensors by cables attached to the dancers body.

Advantages of this concept are the overall reduced amount

of hardware and reduced payload on the radio transmission. In exchange any cables attached to the dancers body are fundamentally obstructive, which requires the system to be specifically tailored to an individual. Overall, this limits the versatility and scalability, fixes the sensor arrangement and is more susceptible to mechanical stress.
Both concepts are practically applicable and the decision is guided by external factors induced from the application domain.

Another hardware aspect to evaluate is the JST connector used to attach the battery to the sensor board. As this connector is not designed for consumer application, issues are likely to occur due to incorrect use, especially when considering a technically inexperienced user group. However, after this component has been installed in the early prototypes, no person interviewed expressed concerns or problems regarding this component.
Furthermore, the design remained due to lack of viable, more user-friendly alternatives. Notably, no consumer established connector is designed to withstand high physical stress, which is critical in our usage scenario.

The BL652 Bluetooth module is available in two hardware configurations; one utilizing a small PCB mounted antenna and one providing a connector for a larger external antenna. Due to the mechanical limitations an external antenna is obviously not feasible for the sensor nodes. On the host node side, however, we experimented with both the PCB mounted and external option, which resulted the following observations.
The radio performance of both options is sufficient for application within a $8\,\mathrm{m}$ radius. Working with the system indoors in large rooms did not produce any issues related to the radio connectivity during normal use. However, intercepting the radio line by a wall or fully enclosing the sensor node in the hand disturbs or fully breaks the radio connection. Tests with the external, significantly larger antenna produces better, although inconsistent results. Due to lack of dedicated test equipment we cannot support these observations with quantitative data, but would recommend usage of external antennas when possible.

4.2 Software Evaluation

Regarding the embedded system firmware developed for sensor and host node, there are no substantial aspects to evaluate. Although development of the firmware is not trivial due to the complexity introduced from the Bluetooth stack on embedded systems and the design of communication interfaces, the addressed tasks and functions are fully accomplished.

The bridge software, which offers a non-scientific community the opportunity of experimenting with the system, was appreciated by the involved artists. According to Mr. Pankert, this freely configurable software makes the project accessible for artists who are not yet familiar with or introduced to our work.

However, when Mr. Pankert was working with the system, we observed some mistakes and that he was not able to perfectly produce desired sound effects or application scenarios. The overall system complexity and depth - sensor node, receiver, bridge, MaxMSP - appears to be overwhelming for untrained individuals. From professional music technicians, however, the response was very positive about the system's conceptual design. The electronic music artist Gilles Doneux gave positive feedback regarding the ready-to-use output values from the bridge software. He was also very receptive to setting up and using the system and proposed the enhancement of outputting the values not separately from individual Max Objects, but having one Max Object outputting all values in a list format. This minor modification was implemented and later added to the system.

4.3 Specific Customization

For the final theater-stage application we specified together with the composer Mr. Pankert a total of eleven specific output mappings based on five sensor nodes. One of these five sensors is located at the chest above the sternum, one at each wrist and one on each upper arm. In the following, we will discuss the output mappings and their intended usage in the music-dance-performance.

The most prominent sensor nodes are the ones attached to the wrist as they are visually apparent to the audience and highly expressive due to a wide range of motion. Each of these sensors is utilized in two fundamental mappings providing heading and pitch of each sensor (referring to section 3.6). Based on the heading, the dancer is capable of directing the sound distribution to different areas of the stage. In simplified terms this means: Pointing to the left of the stage shifts the stereo sound distribution from right to left (commonly known as *Panning* or *Panorama*). The actual stage implementation will utilize a more complex 2D sound distribution allowing sound to be directed in any direction of the horizontal plane and utilizing the full available output range of 360 degrees. This effect also utilizes the pitch output value to alter the intensity. Pointing up or down reduces the intensity; a horizontally directed arm corresponds to a strong influence. Together this gives an intuitive and relatable sound effect, where the hearable audio source can be moved by the dancer.

Based on an audio setup of four speakers this concept was tested and positively experienced. Mr. Pankert confirmed the desired effect and a good match was achieved. In the next section, we attempt to quantify these observations in order to provide more robust results.

4.4 System Performance

To allow for a more accurate assessment of the system we analyze sensor drift, radio transmission performance and list some further specifications.

4.4.1 Sensor drift

The measurement of the three axis gyroscope within the IMU is not completely stable, which causes the given output value to drift away from the actual physical orientation. Since these errors can accumulate over time, the BNO utilizes a magnetometer to counteract the drift and reference rotation additionally to earth's magnetic field.

Nevertheless, the magnetometer is very susceptible to other magnetic fields close-by and - although reduced - error has to be expected. In order to quantify drift error, the sensor board was placed stationary over a period of 30 minutes and the acquired data was recorded. The test was repeated 10 times and the sensor was fully reset in between.

In conclusion, the recorded data showed that the orientation of the sensor, when considering the pointing indicator as described in section 3.6, drifts from the original orientation by $17.2°$ on average. Further distinguishing between heading and pitch values as calculated for our application, heading drifted by $16.4°$ and pitch $19.3°$. Further statistical information on this data is listed in Table 4.1.

Reference	**Mean**/°	**Median**/°	**Min.**/°	**Max.**/°
Pointing Indicator	17.2	15.6	7.7	29.9
Heading	16.4	11.9	5.5	47.4
Pitch	19.3	18.5	4.6	39.4

Table 4.1: Statistical data on sensor drift from 10 test runs over 30 minutes each. All values rounded within $0.1°$.

4.4.2 Transmission quality

In order to maintain a pleasant sound performance, the output values consumed by the audio effect have to be continuous and stable.

One influence to good continuity between values is the time between each sensor reading respectively the sensor data rate. We analyze the same data-set as in subsection 4.4.1 by comparing timestamps of arriving packets. In this setup the sensor node transmits with a fixed rate of 20 Hz. Figure 4.1 shows a histogram of the intervals from a single data record. However, the theoretically expected peak at 50 ms is slightly shifted to the right and appears within 60 to 80 ms.
All counts with an interval greater than 100 ms between consecutive packets indicate a packet loss or high unexpected delay. Also, a significant number of anomalous packet delivery of two packets arriving almost simultaneously is present. This is effect is likely due to congestion in the transmission line and an immediate delivery of several stalled or re-transmitted packets. Latter observation is supported by a low packet loss below 4 %.

4.4.3 Further specifications

During the development LiPo batteries were used with the envelope dimensions 35 mm × 30 mm × 5 mm and a capacity of 400 mAh. This form-factor is particularly handy when the battery is attached on the bottom side of the sensor. In this case battery and sensor board line up and provide a minimal footprint. For this configuration the total envelope dimensions add up to 35 mm × 30 mm × 12 mm for a single sensor node.

The current consumption of a sensor node when active and transmitting is 22 mA. Considering the battery configuration from above, the sensor board can to run for over 18 h on a single charge. Use of batteries with different form factor can reduce the total volume even further at cost of battery run-time.

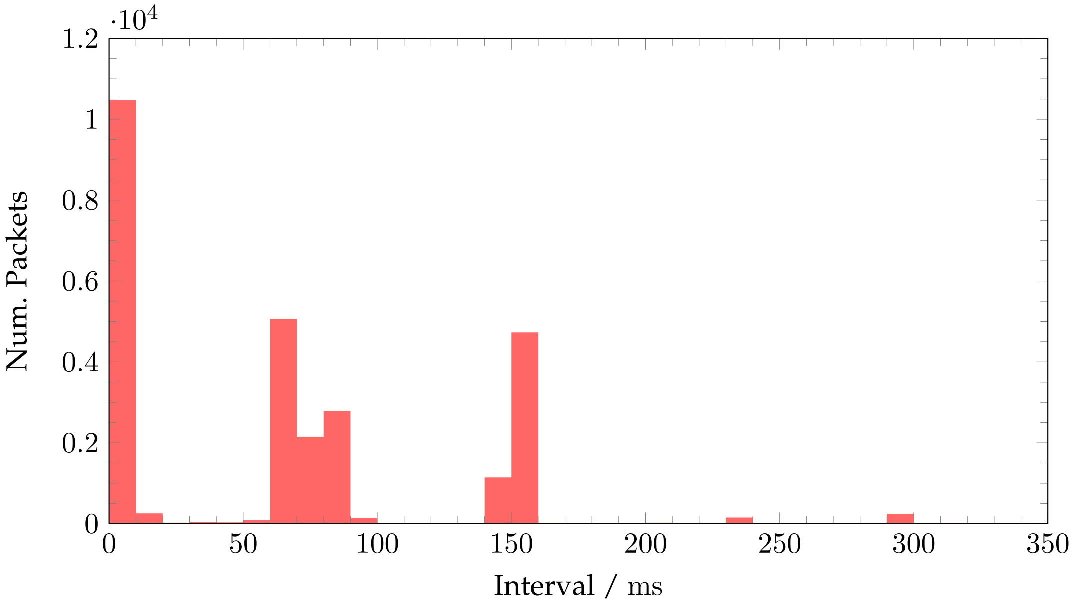


Figure 4.1: Histogram of packet intervals measured by the bridge application (bin width: 20 ms).

Chapter 5

Conclusion

5.1 Summary

In this Bachelor thesis, we presented the development process of a wearable motion tracking systems for a dance application. First, we introduced the broader topic and mentioned other involved participants.

In chapter 2 we pointed out related work close to our field of study; concentrating especially on those working with IMU based systems. Some of these papers targeted a different field of application, but this also indicates the extensive scope covered by the adaptable foundation we developed. The studies most closely related to our application already primed upcoming issues and tasks.

We began the iterative development process with an early prototype, which allowed first observations of the system and defined fundamental design aspects. Furthermore, it initiated a dialog with the artists about the technical aspects and inspired the final usage scenario.

After evaluating the flaws of the early prototype, each individual component was replaced by more advanced options. Our primary focus for selecting the hardware was a minimal PCB footprint and the use of components already working together with no peripheral requirements or side effects. Besides this, we also considered the technical aspects when selecting the radio frequency and sensor technology. A Bluetooth based transmission was implemented, which largely benefits from the radio protocol infrastructure designed for autonomous, ad-hoc systems.

The data carried by the wireless transmission originates from the BNO055, an Inertial Measurement Unit, which internally utilizes several different sensor technologies and outputs the accurate rotational orientation.
Data from multiple sensor nodes is collected at the host receiver and processed within a bridge application before being used to influence musical sound effects. The bridge software provides configuration and management of the system including the setup, various data conversions and calculations. Eventually, the produced output values are used within MaxMSP for sound manipulations, which were developed in cooperation with artists and other technicians.

Based on the completed system, the design decisions were evaluated regarding hardware, software and performance individually. Quantitative analysis indicate room for improvement by calibration and optimization of the data acquisition. However, qualitative evaluation based on user interviews affirms a correct functioning and successful application in the artistical field.

5.2 Future work

By separately defining each element of the system and comprehensively describing the connecting interface, we hope to have made the system open and accessible for extensions and improvements in further work.

Regarding the hardware, the designated WS2812 LED has not yet been implemented and will possibly offer an additional artistic opportunity. First, a corresponding driver which conforms to the custom timing protocol of the LED has to be implemented on the main processing unit. In cooperation with the involved artists, the usage of the visual output device has to be explored and all software elements must be expanded accordingly.

Another opportunity for expansion of the system is the utilization of already available Bluetooth hardware. A laptop with integrated Bluetooth module could directly connect to the sensor nodes and bypass the external host node. This would reduce the system's cross section, make it more accessible to untrained users and reduce hardware cost.

As already pointed out in chapter 2, our hardware might be applicable for other body tracking scenarios. By connecting the sensor nodes with modern smartphones, they might be used in the context of sports applications, where they provide high accuracy data acquisition during specific motion sequences. Other areas include motion based game applications using computer or smartphone and long-term activity tracking in the context of health care applications.

Appendix A

Circuit Designs

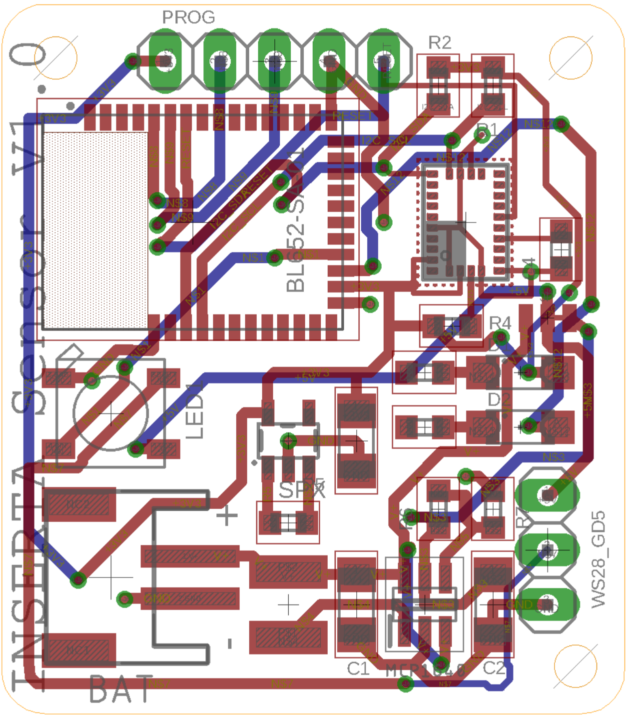


Figure A.1: Electrical layout of the sensor board.

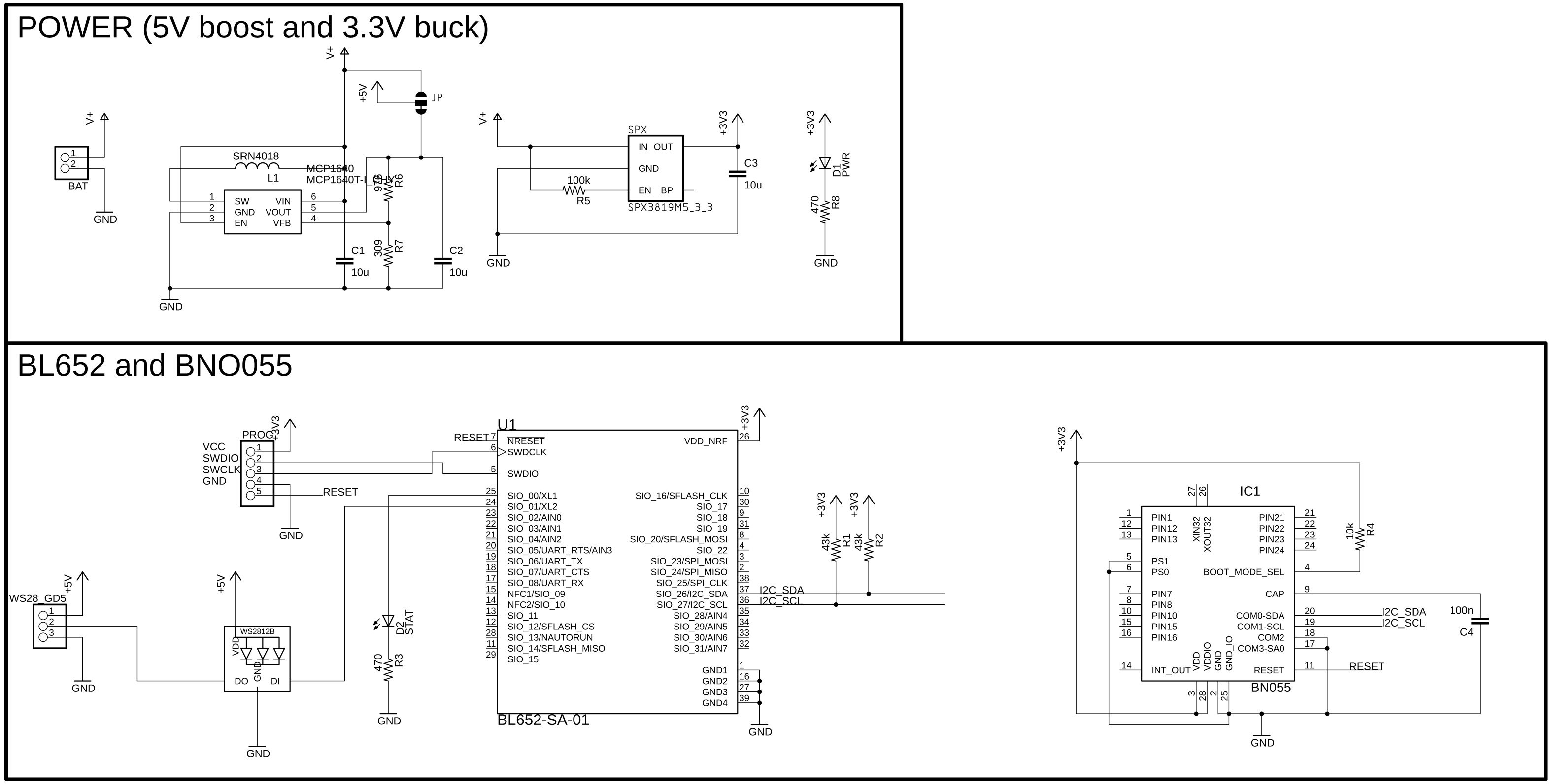


Figure A.2: Electrical schematic of the sensor board.

Bibliography

S. Ryan Edgar, G. Fulk, and E. Sazonov. Recognition of household and athletic activities using smartshoe. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2012:6382–5, 08 2012.

M. Fujimoto, N. Fujita, Y. Takegawa, T. Terada, and M. Tsukamoto. A motion recognition method for a wearable dancing musical instrument. *2009 International Symposium on Wearable Computers*, pages 11–18, Sept 2009. ISSN 1550-4816. doi: 10.1109/ISWC.2009.22.

T. Großhauser, B. Bläsing, C. Spieth, and T. Hermann. Wearable sensor-based real-time sonification of motion and foot pressure in dance teaching and training. *Journal of the Audio Engineering Society*, 60:580–589, 07 2012.

T. Heiduk, P. S. Excell, A. Osanlou, and A. M. Thike. A miniaturised wearable motion-monitoring device to promote prevention of physical injuries. *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 1947–1951, Jan 2018. doi: 10.1109/EIConRus.2018.8317491.

S. Okamoto. Evaluation of skills in batting technique in baseball using acceleration sensors. *Journal of the Japanese Society for Experimental Mechanics*, 12(Special Issue):s255–s260, 2012. doi: 10.11395/jjsem.12.s255.

C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5 (1):3–55, January 2001. ISSN 1559-1662. doi: 10.1145/584091.584093. URL `http://doi.acm.org/10.1145/584091.584093`.

Typeset November 15, 2018