Fachhochschule
Nordwestschweiz

Fachbericht

Team 1

Projekt 4

Fachhochschule Nordwestschweiz FHNW

9. Juni 2018

Studiengang:	Elektro- und Informationstechnik EIT
Auftraggeber/in:	Prof. Hans Gysin Jana Kalbermatter
Fachexperten:	Matthias Meier Prof. Dr. Pascal Schleuniger Pascal Buchschacher Dr. Roswitha Dubach Dr. Anita Gertiser Bonnie Domenghino
Projektteam:	Adrian Annaheim Benjamin Ebeling Jonas Rosenmund Michael Schwarz Samuel Wey Andres Minder

Inhaltsverzeichnis

1 Einleitung

Museen bieten die Möglichkeit verschiedenste Ausstellung den Interessenten nahe zu bringen. Um dem Besucher zu ermöglichen, sich mit den persönlich relevanten Objekten auseinander zu setzen, werden unter anderem Audio-Guides verwendet. Die Audioguides werden meist mit Hilfe von Standard-Kopfhörern umgesetzt. Diese Kopfhörer müssen aufgrund von Hygienegründen gereinigt werden. Diese Reinigungskosten sind verhältnismässig zu gross. Zudem bieten die Museen nicht die Möglichkeit, eine Ausstellung einzugrenzen. Eine Unterteilung der Ausstellung würde dem Besucher die Option geben, eine Eintrittskarte zu erwerben, welche sich nur auf einen Teil der Ausstellung bezieht. Dies könnte Museumsaustellung attraktiver gestalten.

Ziel ist es, einen Audioguide zu entwickeln welcher es ermöglicht eine Ausstellung einzugrenzen, in dem es nur die gewählten Audiodateien abspielt und allenfalls als Zutrittsberechtigung fungiert. Die Audioübertragung soll über einen Knochenschallgeber funktionieren, dies würde die Reinigungskosten senken. Mit Hilfe des Audioguides soll es möglich sein, Sounddateien abzuspielen, sobald man sich in der Nähe eines Kunstobjektes befindet.

Mit Hilfe von Bluetooth Beacons können den Kunstobjekten Audiodateien zugeordnet werden. Durch einen Vibrator kann angekündigt werden, dass eine Audiodatei vorhanden ist, welche durch Bestätigung des Besuchers über den Knochenschallgeber abgespielt werden kann.

Mit Hilfe des Prototyps ist es möglich, Beacons in einem Abstand von xxx Metern korrekt zu erfassen und die zugeordnete Audiodatei abzuspielen. Auf dem Dojo können YYY Audiodateien abgespeichert werden. Am Eintritt können je nach Wunsch die verschiedenen Audiofiles ausgewählt werden und für den Besuch freigegeben werden. Dem Besucher wird die Möglichkeit gegeben, für gewünschte Objekte mit Hilfe eines Like-Buttons am Ende des Besuches Zusatzinformationen in Form einer Broschüre zu erhalten.

Der nachfolgende Bericht ist in sechs Teile aufgeteilt. Durch das Gesamtkonzept wird die Funktion sowie das Prinzip von Dojo erklärt. In den folgenden zwei Kapiteln wird die benötigte Energieversorgung sowie die Software für das Dojo dargelegt. In den letzten drei Kapiteln werden die Schnittstellen sowie die Firmware erläutert.

2 Gesamtsystem

Abbildung 2.1 zeigt das grobe Gesamtsystem des Dojos. Die einzelnen Kapitel dieses Berichts orientieren sich an den Blöcken und deren Funktion im Gesamtsystem. In Kapitel **3 Software** wird die Computersoftware erklärt, welche verwendet wird, um den Dojo zu konfigurieren. Das Kapitel **4 Energieversorgung** befasst sich mit dem Akku, der Lade -schaltung und Überwachung, sowie der Spannungsversorgung. Im Kapitel **5 USB** wird erläutert, wie über die USB-Schnittstelle mit dem Mikrocontroller kommuniziert wird und wie Audiofiles auf die SD-Karte übertragen werden. Das Kapitel **6 Audioausgabe** behandelt die Verarbeitung und Ausgabe der Audiofiles. Das nächste Kapitel **7 Bluetooth** befasst sich mit dem Bluetooth. Das heisst, mit den Beacons zur Erkennung der Kunstobjekte und dem Bluetoothmodul im Dojo für den Empfang. Wie die Firmware als Gesamtsystem funktioniert, erfährt man in Kapitel **8 Firmware** detailliert. Die Validierung der einzelnen Blöcke wird im jeweiligen Kapitel abgehandelt.

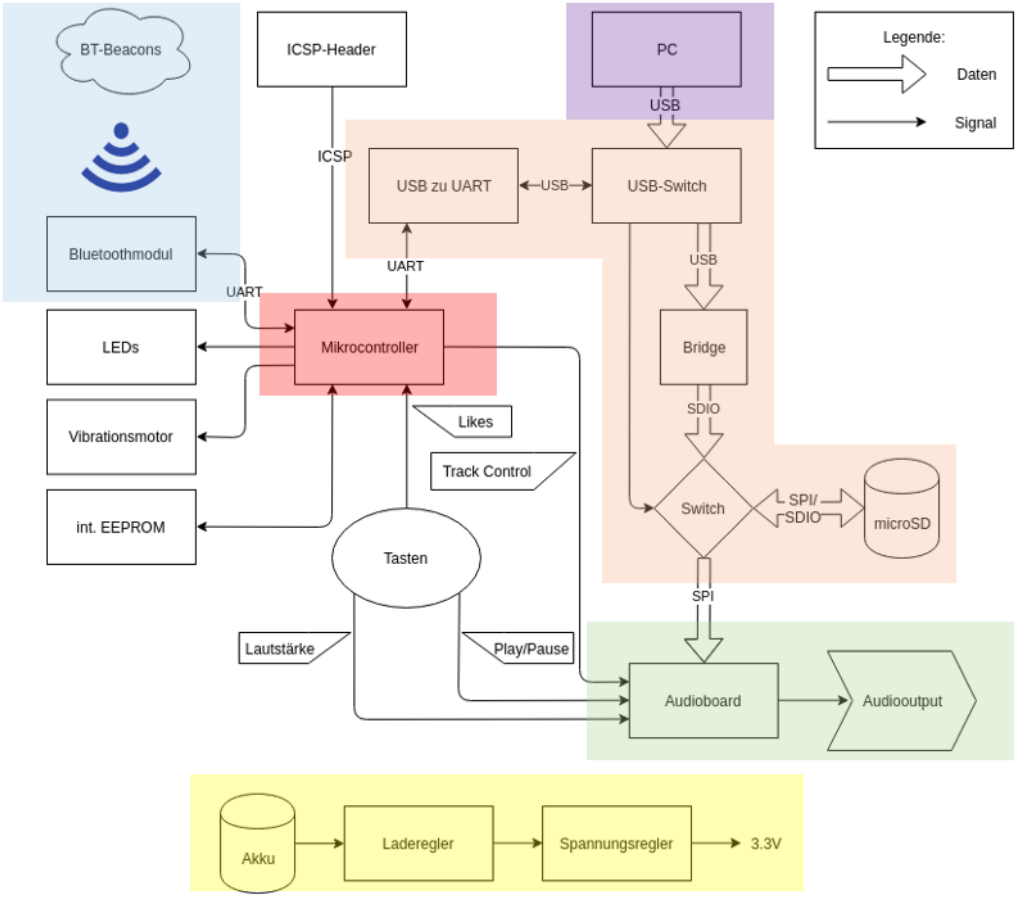

Abbildung 2.1: Blockschaltbild Gesamtsystem

Software	Energieversorgung	USB	Audioausgabe	Bluetooth	Firmware

Tabelle 2.1: Legende Gesamtsystem

3 Software

Im Kapitel Software wird erläutert, wie die Audiodateien auf das Dojo gelangen und die Korrespondenztabelle angepasst werden kann. Es wird erklärt, wie die Likes ausgewertet und verarbeitet werden können.

3.1 Konzept

In diesem unter Kapitel wir das erarbeitete Konzept für die Datenübertragung von einem Computer auf das Dojo dargelegt.

3.2 Datenverwaltung

Alle Ausstellungsobjekte sind in einer Textdatei aufgelistet, in der vermerkt wird, welche Audio- und Textdatei zu welchem Beacon gehört. Die Form dieser Inventardatei ist in Abbildung 3.1 zu sehen.
Zudem wird in einer weiteren Textdatei ein Preset definiert, mit der die SD-Karte beschrieben werden kann. Die Form dieser Presetdatei ist in Abbildung 3.2 zu sehen.

```
exhibition romans{
    233(
        de  roemischerhelm.mp3  roemischerhelm.tex
        en  romanhelmet.mp3     romanhelmet.tex
    )
    10(
        de  becher_rom.mp3      becher_rom.tex
        en  cup_rome.mp3        cup_rome.tex
    )
}

exhibition middle_ages{
    15(
        de  mittelalterlicher_schaedel.mp3  mittelalterlicher_schaedel.tex
        en  skull_medieval.mp3              skull_medieval.tex
    )
    11(
        de  morgenstern.mp3         morgenstern.tex
        en  primitive_weapon.mp3    primitive_weapon.tex
    )
}
```

Abbildung 3.1: Inventardatei für eine Testausstellung

Der CLI-Befehl `makepackage` nimmt die Inventardatei und Presetdatei und generiert ein Ordner in dem alle Audiodateien die in der Inventardatei aufgelisteten sind sich befinden. Die Dateien

```
languages: de, en
exhibitions: romans, middle_ages
```

Abbildung 3.2: Presetdatei für eine Testausstellung

Anzahl gespeicherter Beacons	Sprachcode
ID Beacon1	Like Beacon1, Zutritt Beacon1
ID Beacon2	Like Beacon2, Zutritt Beacon2
⋮	⋮

Abbildung 3.3: EEPROM Adressierungsmuster

des generierten Audiopakets haben als namen nur noch eine Zahl, die bestimmt an welcher Stelle sie auf der SD-Karte abgespeichert wird.
Der CLI-Befehl `loadsd` nimmt das vorher generierte Audiopaket und schreibt es auf die SD-Karte des Audioguides. Dazu schickt es dem Mikrocontroller den Befehl, dass er den Multiplexer umschaltet. Das Betriebsystem des PC sollte dann die SD-Karte mounten und dem Python Skript verfügbar machen, worauf dieses die Daten kopiert. Es wird dabei angenommen, dass nur ein Audioguide am PC angeschlossen ist und dass alle SD-Karten "dojo-sd"heissen.
Der CLI-Befehl `maketicket` nimmt die Inventardatei und die Presetdatei als Argumente und fragt den Nutzer, welche Ausstellungen und Sprache dem Besucher zur Verfügung stehen sollen. Mit diesen Inputs wird ein Ticket generiert, das über die Serielle Schnittstelle auf das EEPROM des Mikrocontrollers geladen wird.
Auf dem EEPROM hat jeder Beacon zwei Bytes, die die Beacon-ID, Like und Zutritt beinhalten. Die Likes sind zu Beginn des Besuchs alle auf Null und können vom Besucher gesetzt werden. Das Zutrittsbits der bezahlten Ausstellungen werden zu Beginn des Besuchs gesetzt und können vom Besucher nicht verändert werden.

3.3 Auslesen, Schreiben

Das EEPROM des Mikrocontrollers wird über eine serielle Schnittstelle beschrieben, wobei alle Befehle das gleiche Muster haben:

```
[1 char] op-code | [3 char] arg1 | [3 char] arg2 ;
```

Um ein Byte mit Wert 42 in die Adresse 2 zu schreiben, muss der String `p002042` auf die serielle Schnittstelle geschrieben werden. Um das Byte an der Adresse 2 auszulesen, muss der String `r002 ;` auf die serielle Schnittstelle geschrieben werden. Wichtig ist, dass alle Befehle mit einem Semikolon terminiert werden.
Das Python Modul beinhaltet neben Schreib- und Lesefunktionen auch noch Funktionen, die ein ganzes Ticket auf das EEPROM schreiben können, bzw. wieder einlesen können. Hier ist es wichtig, dass das vordefinierte Muster des EEPROMs in Abbildung 3.3 eingehalten wird.

3.4 Validierung

Für die Validierung der Datenverwaltung wurden Python Unittests eingesetzt, die mit einer Testdatenbank alle Funktionen testen. Neben den einzelnen Funktionen wurde auch ein kompletter

Durchlauf mit der Testdatenbank gemacht.
Die EEPROM-Operationen werden mit einem Arduino Uno-Board getestet, wobei das EEPROM vom PC aus beschrieben und wieder eingelesen wird. Die Unittests stellen sicher, dass das EEPROM die gewünschten Daten beinhaltet und dass diese mit der spezifizierten Geschwindigkeit übertragen werden.
Es werden zudem noch Benchmarks durchgeführt, welche sicherstellen, dass nicht unnötig gewartet wird. Ein Test, der misst, wie lange das beschreiben von verschieden lange Tickets dauert zeigt, dass das beschreiben eines vollen Tickets mit 250 Beacons 6.17 dauert.

4 Energieversorgung

4.1 Technische Grundlagen

In diesem unter Kapitel werden die technischen Grundlagen für das Verständnis der Energieversorgung dargelegt.

4.2 Konzept

Hier wird das verwendete Konzept der Energieversorgung respektive der Ladeschaltung erklärt und begründet.

4.3 Hardware

Hier steht welche Bauteile in welcher Anordnung verwendet wurden.

4.4 Validierung

Hier wird erklärt, wie die Validierung der Ladeschaltung gelöst wurde. Die Resultate der Validierung (der Energieversorgung) und die eventuellen Abweichungen zu den Wünschen werden in diesem Kapitel beschrieben.

5 USB

Im folgenden Kapitel wird erklärt, wie über die USB-Schnittstelle Daten auf den Dojo geladen werden und ausgelesen werden. In einem Unterkapitel wird sogleich die Validierung beschrieben.

5.1 Technische Grundlagen

Nachfolgend werden einige Begriffe zur Datenübertragung kurz erklärt, welche im Kapitel von Bedeutung sind.

UART Universal Asynchronous Receiver Transmitter (UART) realisiert eine digitale serielle Schnittstelle. Über die UART-Schnittstelle können Daten über einen Datenstrom gesendet und empfangen werden. Die Daten sind in einem fixen Rahmen, bestehend aus Start-Bit, fünf bis neun Datenbits, optionalem Parity-Bit zur Fehlererkennung und einem Stopp-Bit.

SDIO Secure Digital Input Output (SDIO) bezeichnet ein Interface für die Datenübertragung zwischen SD-Karten. Die Daten werden wahlweise im SPI, 1-Bit oder im 4-Bit Modus übertragen.

5.2 Konzept

Zur Kommunikation mit dem Mikrocontroller im Gerät besteht eine USB-UART-Schnittstelle. Um neue Audio-Dateien auf die SD-Karte zu schreiben, wurde ausserdem eine USB-SDIO-Schnittstelle realisiert. Folgende Abbildung 5.1 zeigt das Konzept mit den beiden Schnittstellen.

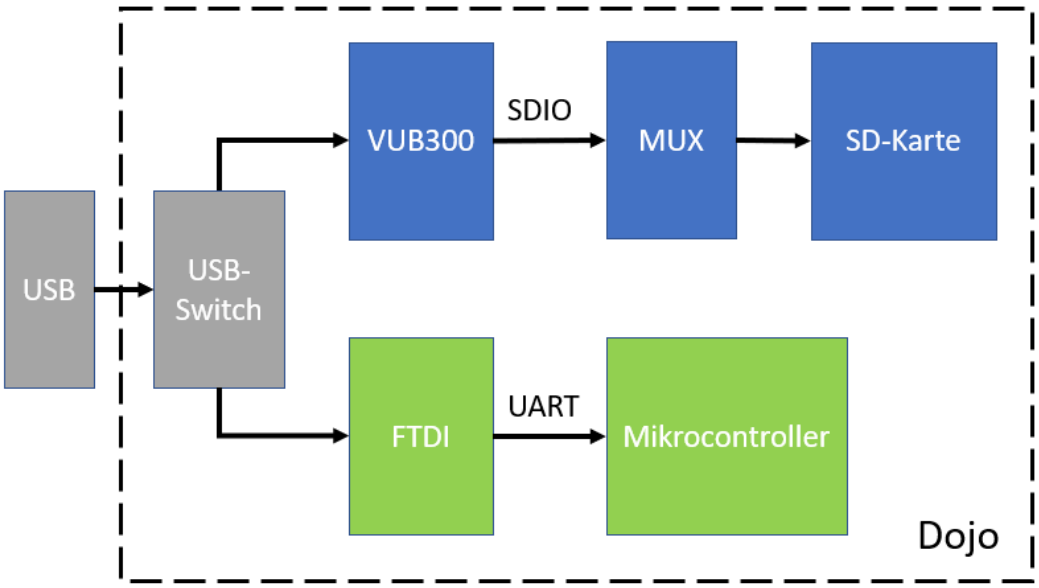

Abbildung 5.1: Konzept USB

Eine micro USB-Typ B Schnittstelle ist die gemeinsame Schnittstelle nach aussen. Dahinter schaltet ein USB-Switch, gesteuert durch den Mikrocontroller, einen der beiden Pfade durch. Es kann jeweils entweder der SDIO-Pfad oder UART-Pfad durchgeschaltet werden, nie beide Pfade gleichzeitig.
Als USB-Switch wird der Baustein TS3USB30E von Texas Instruments verwendet. Dieser Chip wurde dafür entwickelt, um high-speed USB 2.0 Signale in portablen Geräten und Konsumelektronik zu schalten. Gesteuert wird der USB-Switch über zwei Pins. Über den Enable-Pin kann der Baustein in einen hochohmigen Zustand gebracht werden, um den Bus nicht zu belasten und den Stromverbrauch zu senken. Mit dem Select-Pin wird der durchzuschaltende Pfad ausgewählt. Folgende Tabelle 5.1 zeigt die Wahrheitstabelle für die entsprechenden Pfade:

Select (Pin24)	$\overline{Enable}$ (Pin23)	Funktion
X	HIGH	hochohmig
LOW	LOW	Pfad = UART
HIGH	LOW	Pfad = SDIO

Tabelle 5.1: Wahrheitstabelle USB-Pfade

5.3 USB zu UART

Der Baustein FT231X von FTDI stellt das Interface von USB zu UART bereit. Das ganze USB-Protokoll wird dabei auf dem Chip abgehandelt und es sind kaum zusätzliche Bauteile nötig (siehe Schema). Über zwei Pins (Rx und Tx) werden die Daten seriell übertragen.
Bei aktivem UART-Pfad wird der Dojo am Computer als serielle Schnittstelle erkannt. Über diese Schnittstelle kommuniziert nun die Software auf dem Computer mit dem Mikrocontroller im Gerät, um beispielsweise die ‹Likes›auszulesen.

5.4 USB zu SDIO

Der VUB300 Chip vom Hersteller elan stellt das USB zu SDIO Interface bereit. Damit kann die SD-Karte im Dojo über den USB-Anschluss beschrieben und ausgelesen werden. Wird mit dem USB-Switch der SDIO-Pfad durchgeschaltet, erscheint die SD-Karte am Computer als Datenträger und kann gelesen sowie beschrieben werden. Zu beachten ist, dass zwischen dem VUB300 Chip und der SD-Karte noch ein weiterer Multiplexer (MUX) eingebaut ist. Mit diesem wird die SD-Karte zwischen dem VUB300 und dem Audio-Chip umgeschaltet. Genaueres dazu findet man in Abschnitt **6 Audioausgabe**. Damit die SD-Karte mit dem SDIO-Pfad verbunden ist, muss der MUX TS3A27518 (siehe Schema im Anhang) folgendermassen angesteuert werden:

$\overline{Enable}$ (Pin25)	IN1 (Pin26)	IN2 (Pin27)	Funktion
HIGH	X	X	hochohmig
LOW	LOW	LOW	Pfad = SDIO(VUB300)
LOW	HIGH	HIGH	Pfad = Sound-Modul

Tabelle 5.2: Wahrheitstabelle SD-Pfade

5.5 Validierung UART-Pfad

Um die korrekte Funktion der seriellen Schnittstelle zwischen dem USB-Anschluss und dem Mikrocontroller zu überprüfen, wurde eine kleine Testsoftware auf den Mikrocontroller geladen. Sobald der Mikrocontroller am RX-Pin Zeichen empfängt, sendet er diese über den TX-Pin wieder zurück. Mit dem Serial Monitor der Arduino IDE wurden die Zeichenketten erfolgreich gesendet und empfangen.

5.6 Validierung SDIO-Pfad

Mit einem weiteren Testprogramm wurde die Funktionsfähigkeit des SDIO-Pfades überprüft. Dafür wurde der USB-Switch und der MUX mit dem Mikrocontroller entsprechend angesteuert. Die SD-Karte wurde daraufhin von einem Computer mit Linux Betriebssystem als Datenträger erkannt und konnte erfolgreich ausgelesen und beschrieben werden. Leider stellte sich nach weiteren Tests heraus, dass auf einem Windows-Rechner keine SD-Karte erkannt wird. Die Gründe dafür sind nicht bekannt und das Problem konnte auch nicht behoben werden. Somit muss die Computer-Software auf einem Linux Betriebssystem gestartet werden.

Anmerkung
Durch die begrenzte Verfügbarkeit des VUB300 Chips, sowie der geringen Informationsdichte über den Baustein und dem Kompatibilitätsproblem mit Windows, ist dieser nicht für weitere Projekte oder Serienproduktionen zu empfehlen.

6 Audioausgabe

6.1 Technische Grundlagen

6.2 Konzept

6.3 Hardware

6.4 Firmware

6.5 Validierung

7 Bluetooth

7.1 Technische Grundlagen

7.2 IBeacons auslesen und identifizieren

Mittels der Funktion scan() im State SCAN wird mit dem hm-11 nach den verfügbaren IBeacons in der Umgebung gesucht. Über eine serielle Schnittstelle (UART) wird vom MCU der Befehl AT+DISI? dem hm-11 geschickt, wobei als Antwort dann alle umliegenden IBeacons in einem String zurückkommen (siehe Abbildung 7.1). Dafür wird vom MCU jeder char, rsp. jedes

Send: AT+DISI?

Recv: OK+DISCS (Scan start)

Recv: OK+DIS[P0:P1:P2:P3:P4] (if have one device)

Recv: OK+DIS[P0:P1:P2:P3] (if have two devices)

.....

Recv: OK+DISCE (Scan end)

Abbildung 7.1: P0: Factory ID (8 Byte); P1: IBeacon UUID (32 Byte); P2: Major-, Minorvalue, Measured Power (10 Byte); P3: Media-Access-Control-Adresse MAC (12 Byte); P4: RSSI (4 Byte) [1]

Byte vom Datenbuffer ausgelesen und verwertet. Es wird zuerst nach einer UUID eines IBeacons gefiltert und dann die ersten drei Zahlen in einen Integer gecastet. Anschließend soll das Majorvalue[1], welches einen bestimmten, selbst wählbaren Wert hat, abgeglichen werden, damit nur die IBeacons berücksichtig werden, die relevant sind. Ist der empfangene RSSI-Wert grösser als -90dbm, wird der IBeacon eingespeichert. Dies wird mit jedem empfangenen IBeacon wiederholt und immer die RSSI-Werte miteinander verglichen, wobei dann der IBeacon mit dem grösseren RSSI-Wert ins System gespeichert und als Returnvalue von der scan() Funktion zurückgegeben wird.

7.3 Validierung

Für die Validierung wurden zwei IBeacons mit Androidhandys simuliert (Beacon Simulator App). Dafür wurden diese mit einer Distanz von ungefähr $6m$ zueinander auf einer Höhe von ca. $2m$ platziert. Um die detektierten IBeacons direkt auszuwerten, wurde die UUID auf einen Emulator (Putty) über eine serielle Schnittstelle (USB Typ micro B) herausgeschrieben. Somit konnte verifiziert werden, dass der vom Dojo detektierte IBeacon auch der sich am Dojo nächsten befindliche IBeacon war.

Eine definitive Distanzbestimmung zwischen dem IBeacon und dem Dojo ist schwierig, da die Sendeleistung der simulierten IBeacons etwas schwanken. Auch die Auslegung des Raumes, sowie

[1] Major- und Minorvalues dienen hauptsächlich zur zusätzlichen Identifikation eines IBeacons

die Einrichtung kann das Signal abschwächen, was direkten Einfluss auf die Erkennungsdistanz hat. Es kann aber festgehalten werden, dass ein IBeacon in einem Umkreis von $3m$ garantiert erkennt wird, solange sich keine Gegenstände unmittelbar zwischen den beiden Objekten befinden.

8 Firmware

8.1 Konzept

8.2 Statemachine

Auf dem Mikrocontroller läuft eine Mealy-Statemachine mit fünf Zuständen:

SCAN: Es wird vom Dojo nach IBeacons in naher Umgebung gesucht. Falls einer gefunden wurde, dann wird direkt das dazugehörige Audiofile bereitgestellt. Wird der Playbutton gedrückt, ändert sich der State zu PLAY. Ansonsten wird weiter gescanned.

PLAY: Hier werden die bereitgestellten Audiofiles abgespielt. Wird der Playbutton nochmals gedrückt, wird das Abspielen abgebrochen und der State wechselt wieder zu SCAN.

Wird das Dojo an den Computer angeschlossen, ist der Zustand des Dojos abhängig von der gewünschten Tätigkeit. Dafür wird vom Computer aus ein Befehl über das USB-Kabel gesendet, woraufhin sich der State ändert:

GET_Likes: Die vom Besucher getätigten Likes werden vom EEPROM auf den Computer transferiert.

LOAD_SD: Die microSD-Karte wird mit den gewünschten Audiofiles beschrieben.

LOAD_CONFIG: Ein Ticket wird auf das interne EEPROM geladen.

8.3 Datenverwaltung

In diesem Kapitel wird die Datenverwaltung auf dem internen EEPROM erklärt.

8.4 Validierung

Hier wird die Validierung der kompletten Firmware sowie dessen Ergebnisse dargelegt.

9 Verzeichnisse

Abbildungsverzeichnis

Literaturverzeichnis

[1] JN Huamao Technology Company. Bluetooth 4.0 BLE module. Datenblatt, July 2017.

10 Anhang

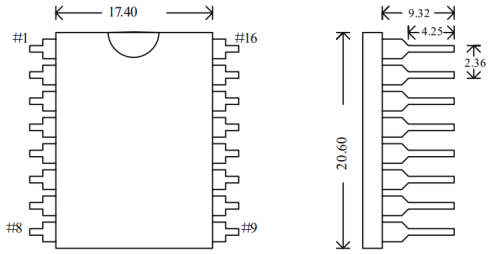

Abbildung 10.1: Abmessungen WTV020