D3.1 – Hardware specifications and APIs of smart garments

Work Package 3 – Lighthouse Projects at Laboratory Nodes

Authors in Alphabetical Order:

Ramyah Gowrishankar (AALTO)
Tim Moesgen (AALTO)
Yu-Han Tseng (AALTO)
Esa Vikberg (AALTO)
Yu Xiao (AALTO)

Grant Agreement number	**101070533**
Action Acronym	**EMIL**
Action Title	**European Media and Immersion Lab**
Call	**HORIZON-CL4-2021-HUMAN-01**

EMIL project is funded by the European Union and UK Research and innovation.

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Version date of the Annex I against which the assessment will be made	*Start date of the project*	*Due date of the deliverable*	*Actual date of submission*	*Lead BEN / AP for the deliverable*	*Dissemination level of the deliverable*
21.7.2023	1.9.2022	30.11.2023	1.12.2023	AALTO	Public

Document reviewer(s)	
Name	*Beneficiary*
Christof Lutteroth	UB

Abstract

This document, "Hardware specifications and APIs of smart garments" (D3.1) describes the hardware design of smart garments that provide tactile and thermal feedback, including the spatial layout and default configurations of motors, heating and cooling elements and sensors, as well as techniques for integrating the hardware components into the textiles. The report also describes open APIs for (re)configuring and (de)activating components, and for obtaining sensor data.

Contents

Tables

Figures

List of Abbreviations

API	Application Programming Interface
DC	Direct Current
EDA	Electronic Design Automation
ERM	Eccentric Rotating Mass
FSTP	Financial Support for Third Party
GSR	Galvanic Skin Response
NTC	Negative Temperature Coefficient
PCB	Printed Circuit Board
PTC	Positive Temperature Coefficient
PWM	Pulse Width Modulation
UPDI	Unified Program and Debug Interface
VR	Virtual Reality

1 Introduction

This document, “Hardware specifications and APIs of smart garments” (D3.1) provides a detailed description of the hardware design and Application Programming Interface (API) of smart garments, which are wearable devices that can create tactile and thermal feedback and sense the wearer’s bio signals like body temperatures and electrical conductance of skin. The report describes open APIs for controlling and accessing the components of the smart garments, such as actuators and sensors. The report also shows the spatial layout and default configurations of these components, as well as the techniques for integrating the hardware components into textiles. The document can serve as a resource for potential Financial Support for Third Party (FSTP) projects that develop smart garments and integrate them in Virtual Reality (VR). The document is structured as follows:

System Overview: This chapter gives an overview of the system and the main components that make up the smart garments.

Hardware Specification: This chapter provides a detailed description of the hardware components and the circuit design of the smart garments, including the spatial layout and default configurations of the actuators and sensors.

API Specification: This chapter explains the open APIs that can be used to control the smart garments from a software application, including the commands and data formats for configuring and activating the hardware components, and for obtaining sensor data from them.

Textile Integration: This chapter describes the methods and techniques for integrating the electronic components into textiles, using different types of structures and connectors.

Application Examples: This chapter presents some examples of how the hardware specification and API can be used to implement a smart garment and integrate it with a VR game, demonstrating some possible use cases and scenarios.

2 System Overview

The wearable haptic device is composed of a master board, several slave boards that connect with sensors and/or actuators, a power board for regulating voltage, and a power supply. The master board is an Arduino board which acts as the central controller of the system. Through the master board, the wearable haptic device is connected to a software running on a computer over a Serial or Wi-Fi connection. The software collects sensor readings and controls the haptic feedback (i.e., configurations of actuators) using the open API described in Chapter 4.

The master board communicates with the slave boards using the I2C protocol. The slave boards are small boards with microcontrollers, such as ATtiny, that act as the drivers of actuators and sensors. In our Lighthouse project, which creates a virtual sauna experience, the actuators and sensors in use include Peltier modules, conductive yarn-based heating elements, vibration motors, and thermal sensors. These actuators and sensors are connected to the slave boards using various connectors. The power supply provides the necessary voltage and current to the master and slave boards, as well as to the actuators and sensors. The modular design of the system allows for configuring the desired number of sensors and actuators in the garment.

3 Hardware Specification

This chapter is split into two sections. Section 3.1 describes the actuators and sensors used in the smart garment, and section 3.2 gives an overview of the hardware used to control them.

3.1 Actuators and Sensors

The wearable device can control various sensors and actuators. In this project, these include Peltier elements and textile heating elements for thermal feedback, eccentric rotating mass (ERM) vibration motors for haptic feedback, thermistors for temperature sensing, and a galvanic skin response (GSR) sensor for measuring the user's GSR. New types of sensors and actuators can be added by following the same principles.

3.1.1 Peltier Modules

A Peltier module, also known as a thermoelectric module, is a device that utilizes the Peltier effect to create a temperature differential between its two sides when an electric current is applied. A Peltier module typically consists of two semiconductor plates made of different materials, often bismuth telluride, sandwiched between two ceramic plates. When an electric current is passed through the module, it causes one side of the module to become cooler while the other side becomes hotter. This enables Peltier modules to be used for various applications, including cooling and heating in electronics, temperature control, and thermal management systems. Two Peltier modules are shown in Figure 1.

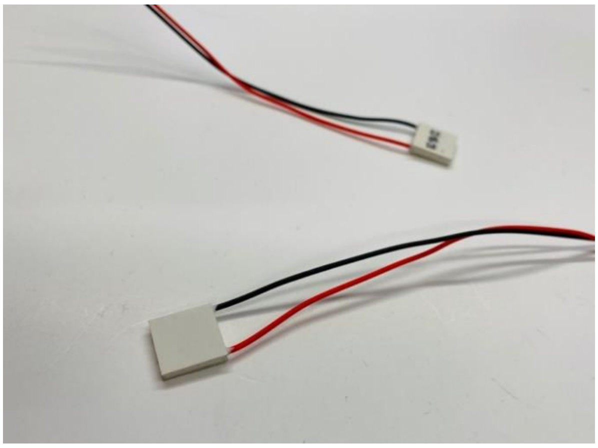

Figure 1 Peltier modules

3.1.2 Textile Heating Elements

Conductive yarns can be used as a heating element integrated on or within the textile substrate. Heat is generated and transferred, which affects the design and area of the heating patches. Yarn-based heating elements can be used instead of or in addition to Peltier elements in the heating garment.

The yarn-based heating elements used in the project utilise Karl Grimm silverized copper 7X1 yarn. This yarn is attached over a base fabric using an industrial embroidery machine where the conductive yarn is used as the bobbin thread (Figure 2). The base material used is a thick polyester-cotton blend woven fabric. Figure 3 illustrates the surface temperature of this sample when connected to a power supply with adjustable voltages of 7.5V, 9V and 12 V.

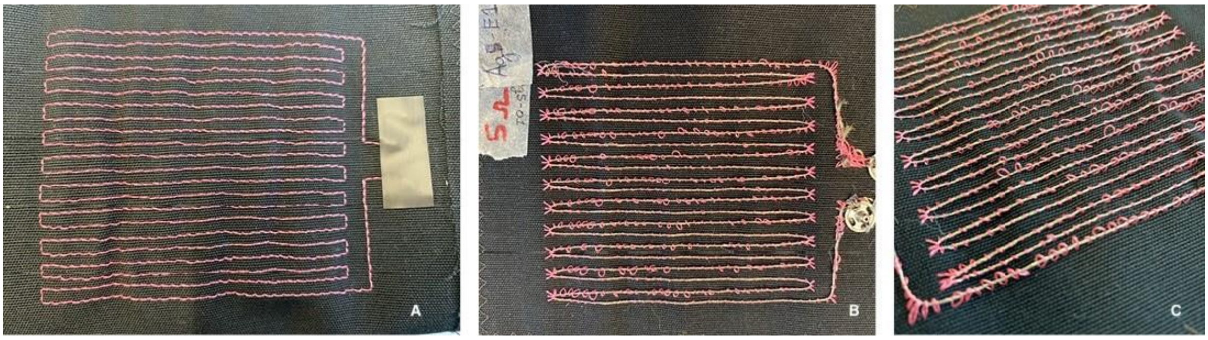

Figure 2 Sample of embroidered heating patch made with silver-coated copper yarn. A) Top of fabric. B) Back of fabric (heating element). C) Detail of the embroidered copper yarn.

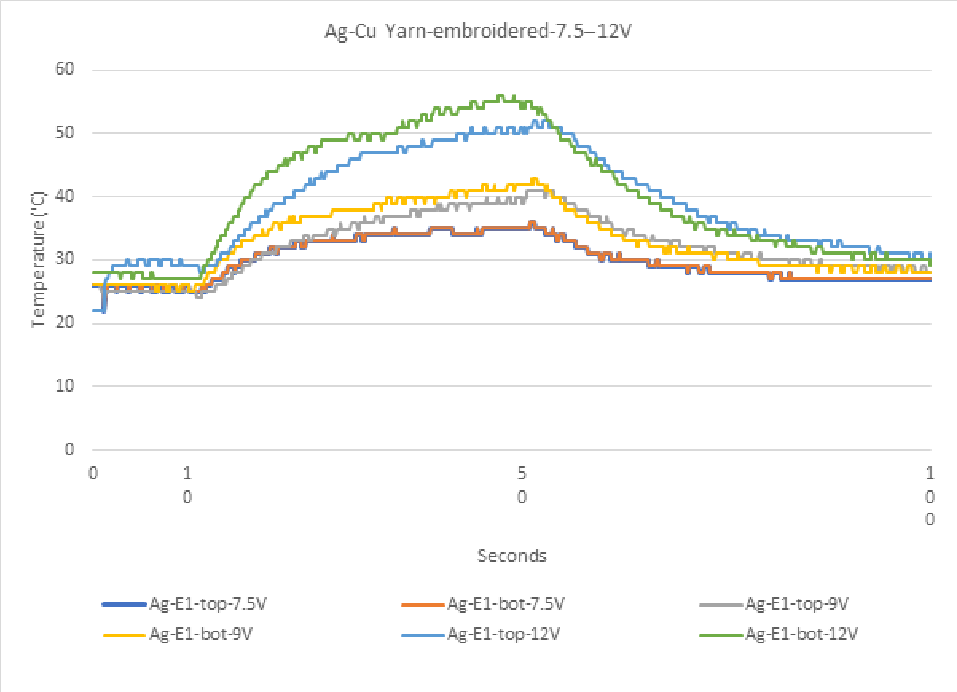

Figure 3 Embroidered sample with Silver-coated Copper yarn. Surface temperature of top and bottom of fabric for 7.5V, 9V and 12V.

A modular design that can be tested on different parts of the arm and upper body was chosen. Once a better understanding of the required size, pattern and location is achieved through user testing, these heating elements can be further integrated as part of the garment design and construction.

Figure 4 shows an example of a prototype of strap-on heat pads constructed for initial testing with users in a lab setting. We covered the heating element with another piece of fabric for insulation. Our previous test had shown how the heat transfers through the fabric. The density of the heating elements varies from the sides to the centre to create a slightly different heat intensity in the centre. The straps have Velcro at the edges that help to easily wear and remove for user testing. The heating circuit can be attached to the pads using the snap buttons attached to the ends of the heating element.

Since the conductive yarns are not coated, it is essential to insulate them properly. The connectors or conductive yarns used to connect the heating element to the rest of the circuit should also be tested for high amperage and low resistance, so they do not get heated up. Additionally, choosing a heat-resistant fabric to integrate the heating elements is important.

Figure 4 Prototype of wearable yarn-based heating pad for initial user testing. A1) Top of the heat pad. A2) Bottom of the heat pad with an additional fabric cover. B) Metal snap buttons to connect the heating element to the control circuit. C) Prototype worn on the upper arm.

3.1.3 Eccentric Rotating Mass Vibration Motor

An ERM vibration motor is a commonly used type of vibration motor in electronic devices to provide haptic feedback, such as the vibrating alert in a mobile phone. It consists of a direct current (DC) motor connected to an eccentric mass. As the DC motor rotates, the unbalanced mass induces a distinctive vibration, perceptible to the user. The vibration intensity can be adjusted by controlling the DC motor's rotation speed through changes in the input voltage. This dynamic control allows for a customizable haptic experience in various applications.

3.1.4 Thermistor

Thermistors are resistive sensors which change their resistance based on temperature. Thermistors are classified into negative temperature coefficient (NTC) and positive temperature coefficient (PTC) thermistors, where the resistance decreases for the former or increases for the latter, when temperature is increased. The thermistors used for the smart wearable are thin film NTC thermistors, TT6-10KC8-9-25, which have a no-load resistance of 10000 ohms at a temperature of 25 degrees. These were selected for easy placement in the proximity of the actuators, as well as an operating temperature which is close to the human body temperature.

3.1.5 Galvanic Skin Response Sensor

GSR sensor measures the change in conductivity of the user's skin caused by sweating. Sweating can happen as a response to high temperatures, but it can also be affected by an emotional reaction, such as stress or excitement. As the resistance of the skin is high, and the voltage used for measuring it is relatively low, so as not to cause side effects, the signal is weak and susceptible to noise. GSR sensors account for this by using noise filtering and amplifying circuits. The GSR sensor used is a MIKROE ECG/GSR Click-board. It incorporates the measuring circuits, as well as I2C communication, so it can be considered a slave board in the system architecture.

3.2 Control systems

This section gives an overview of the system from the hardware perspective and the details of the printed circuit board (PCB) design. There are three main objectives for the hardware design:

1) Modularity: Different PCBs with various functionalities are modularized, allowing users to create different combinations depending on their requirements. For example, based on the VR content and the garment design, users can use different amounts of vibration motors and Peltier elements.
2) Customizability: Each slave PCB is equipped with a programming pin. Users can not only use them with pre-designed API but also develop their own controlling programs and reprogram the board.
3) Minimal size: To apply the PCBs to the smart garment, the size of the PCBs must be as small as possible. The wiring is also simplified to make the installation easier.

Due to the modular design, the system can be extended by connecting more units. A unit consists of one Master Board, one or more Main Power Modules, and one or more Sensor/Actuator Modules, which are also called slave boards in the I2C protocol. The Main Power Module is connected to the main power supply to transfer both original and regulated power to individual slave boards based on their functional requirements. It is also connected to the master board to get commands through I2C, and to power the master board. An overview of the system is shown in Figure 5.

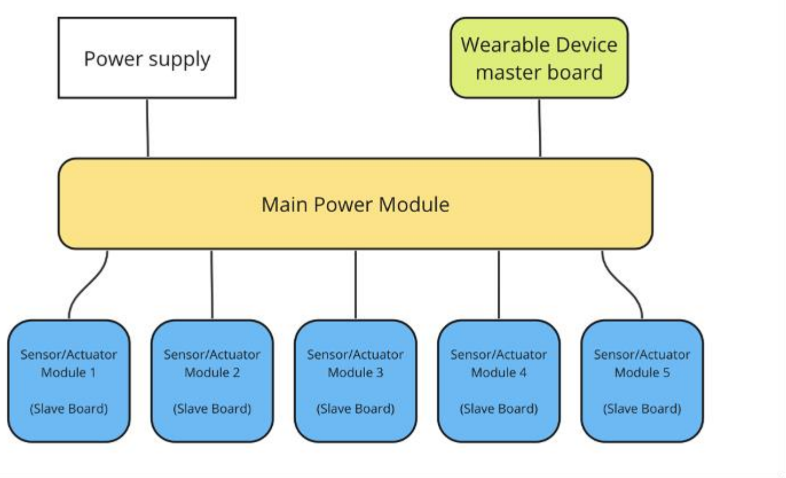

Figure 5 The hardware connection diagram of a unit

Each slave board can be connected to the Main Power board via a JST-PH connector, which combines the connection of original power, 5v regulated power, GND, SDA, and SCL from the Main Power board. There are four types of slave boards developed for the wearable: Peltier slave, Motor slave, Wire Heater slave, and Thermistor slave. Additionally, the GSR sensor board can be considered a slave board. The following sections are detailed descriptions of each PCB module design, apart from the GSR sensor. Each section will include:

- Theory of operation: This part introduces the features of the PCB and explains how it works.
- Circuit design: This part provides the schematic of the PCB and the detail of the circuit.
- Electronic component: This part is the list of the electronic components for fabricating the PCB. The electronic components are chosen with the consideration of functionality, accessibility, and price.
- PCB fabrication: This part provides the assembly drawings and the fabrication drawings of the PCB to assist in producing this PCB. All the PCBs design are made by KiCAD, which is an open-source electronic design automation (EDA) software suite.

All of the designs are also made available on the project website: https://emil-xr.eu.

3.3 Main Power Module

The Main Power Module, shown in Figure 6, is a double-sided PCB. Its primary features include:

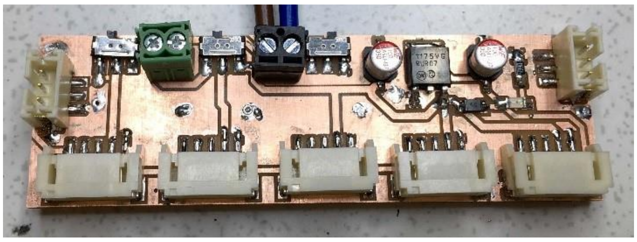

Figure 6 Main Power Module

- Input voltage range of 6V-20V.
- Regulating the input voltage to 5V.
- A main switch to turn on/off the entire system manually.
- One extra input power with switches, allowing two slave boards to select different power sources.
- I2C connection to the master board.

3.3.1 Theory of Operation

The Main Power Module transfers and regulates the power to the slave boards. It is also connected to the master board and relays the commands to the slave boards via I2C. With a low dropout positive voltage regulator NCV1117, the board can output a fixed 5.0 V voltage that is in excess of 1.0 A, which is the power for operating microcontrollers, vibration motors, and thermistors of the slave boards. There are some advantages of connecting a main power board between the power supply and the slave boards, instead of connecting the slave boards to the power supply directly:

- The high current will not go through the slave boards, which decreases the possibility of breaking the slave boards.
- Can shut down all the slave boards with a switch in an urgent situation.
- The slave board does not need to regulate the power to operate the microcontroller and the other components, which can decrease the size of the board.

3.3.2 Circuit Design

The circuit design for the main power board in shown in Figure 7. There are six parts comprising a main power module:

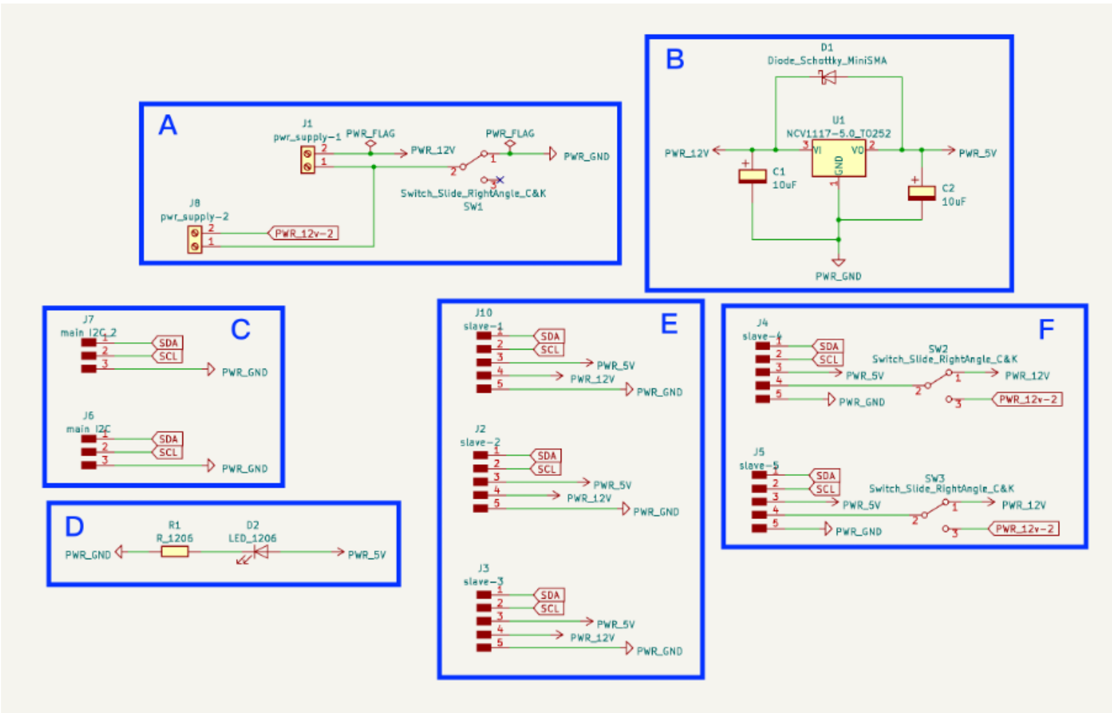

Figure 7 Schematic of the main power module. The main power input is represented by PWR_12V in the schematic, but it can be any input voltage in the range of the board's limit.

A. Input power: There are two input power terminals (J1, J8) on the board. J1 is the main power source of the entire system, and J8 is an extra power source when users need more power for the slave boards. There is a slide switch (SW1) for turning on/off the entire system.
B. Power regulating: The board has a voltage regulator NCV1117 (U1) to convert the input voltage to 5.0 V, which is for operating the microcontrollers, vibration motors, and thermistors of the slave boards. There are two 10 µF capacitors (C1, C2) to stabilize the circuit and one Schottky diode (D1) to protect the circuit.

C. I2C connection: There are two JST-PH connectors (J6, J7) for I2C connection. One can be connected to the master board, and another can be connected to another Main Power module if users want to extend the system.

D. LED: There is one LED (D2) to indicate if the board is turned on. It can also be a simple sign for checking if the board operates normally.

E. Slave connection: There are three JST-PH connectors (J2, J3, J10) for connecting to the slave module. The connector has five connections, which are for I2C (SDA, SCL), PWR_5V, and main power (PWR_12V, PWR_GND). All the slave modules are equipped with the same type of connector, which allows users to connect different types of slave boards to the Main Power board.

F. Slave connection with power selection: There are two JST-PH connectors (J4, J5) same as part E but equipped with switches respectively. The switches allow users to select the power source for the slave board.

3.3.3 Electronic Components

Item	Quantity	Manufacture Part	Reference	Part description
1	2	ED555/2DS	J1, J8	TERM BLK 2POS SIDE ENT 3.5MM PCB
2	3	AYZ0102AGRLC	SW1, SW2, SW3	Slide Switch SPDT Surface Mount, Right Angle
3	1	NCV1117DT50RKG	U1	IC REG LINEAR 5V 1A DPAK
4	1	CDBM1100-G	D1	DIODE SCHOT 100V 1A MINISMA
5	2	EEEFN1K100XL	C1, C2	CAP ALUM 10UF 20% 80V SMD
6	1	RC1206JR-07100RL	R1	RES 100 OHM 5% 1/4W 1206
7	1	SML-LX1206IC-TR	D2	LED RED CLEAR 1206 SMD
8	2	B3B-PH-SM4-TB	J6, J7	CONN HEADER SMD 3POS 2MM
9	5	S5B-PH-SM4-TB	J2, J3, J4, J5, J10	CONN HEADER SMD R/A 5POS 2MM

Table 1 Bill of materials of the Main Power module

3.3.4 PCB Fabrication

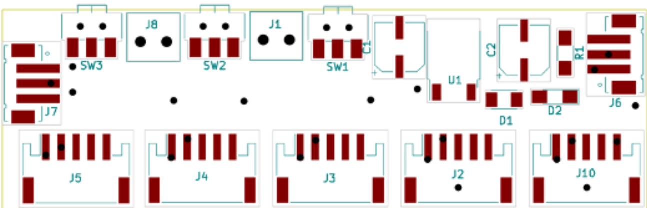

Figure 8 Assembly drawing of the Main Power module.

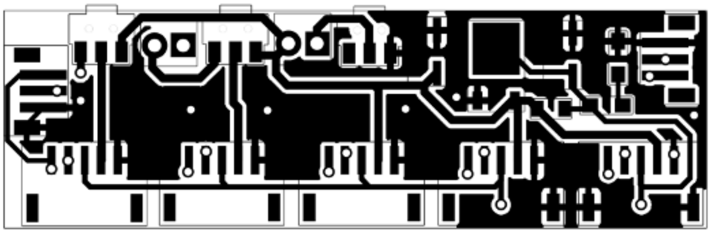

Figure 9 Fabrication drawing (front side) of the Main Power module.

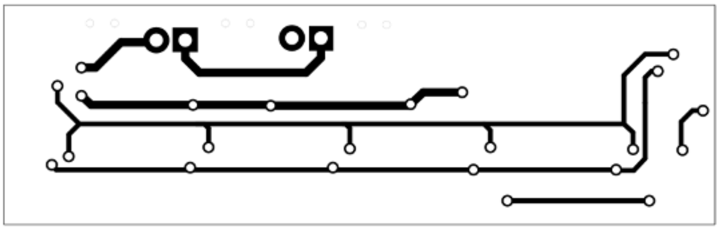

Figure 10 Fabrication drawing (back side) of the Main Power module.

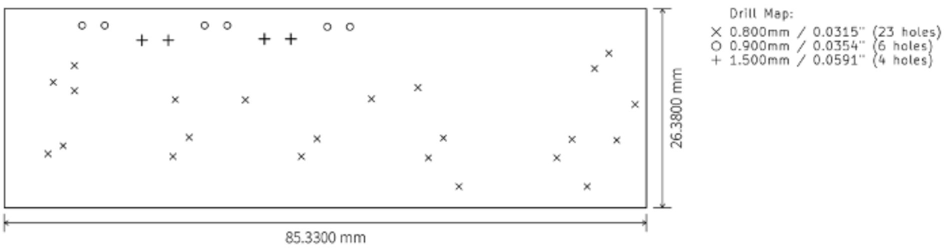

Figure 11 Fabrication drawing (drill drawing view) of the Main Power module.

Here is the note for fabricating the Main Power module:

- This board is a double-sided PCB, but all the components are assembled from the front side.
- The 0.8mm drill holes are for the smaller vias which can connect the front side and the back side of the PCB. The holes will be installed rivets (inner diameter: 0.6mm, outside diameter: 0.8mm).
- The 1.5mm drill holes are for the through-hole components. The holes will be installed rivets (inner diameter: 0.1mm, outside diameter: 1.5mm).
- The 0.9mm drill holes are for installing the slide switches (SW1, SW2, SW3).
- This board might process more power than other module boards, so the traces are thicker to stabilize the circuit. The traces that go through the power are 1.0mm in width, and the other traces are 0.7mm in width.
- The front side is also the ground plan of the PCB.

3.4 Peltier Slave Module

Peltier Slave Module, shown in Figure 12, is a double-sided PCB. The primary features including:

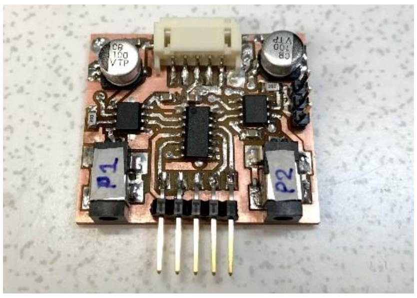

Figure 12 Peltier Slave Module.

- Input voltage range of 4.5 V – 20 V.
- Control up to two Peltiers to heat up and cool down with pulse width modulation (PWM) signal.
- Connect with up to four 10K thermistors.
- Unified Program and Debug Interface (UPDI) pins for programming the board.
- I2C connection with the master board.

3.4.1 Theory of Operation

The Peltier Slave Module allows the users to control the Peltiers easily with the I2C API. The board is equipped with two TB67H451FNG, which are PWM chopper-type DC brushed motor drivers. These drivers dynamically adjust the output voltage based on the PWM signals generated from the microcontroller Attiny1614. PWM is a way of simulating an analog signal using a digital signal, and can change the output voltage to the Peltier, allowing users to control the intensity of the Peltier. With a built-in H-bridge circuit, the users can change the current direction supplied to the Peltier and decide to heat up or cool down the Peltier.

The board can read the temperature, by using a thermistor and a fixed 10K resistor in a voltage divider. The analog data read by ATtiny1614 will be converted to Celsius degrees and sent back to the master board via I2C.

3.4.2 Circuit Design

The circuit design for a Peltier slave module is shown in Figure 13. There are four parts comprising a Peltier Slave module:

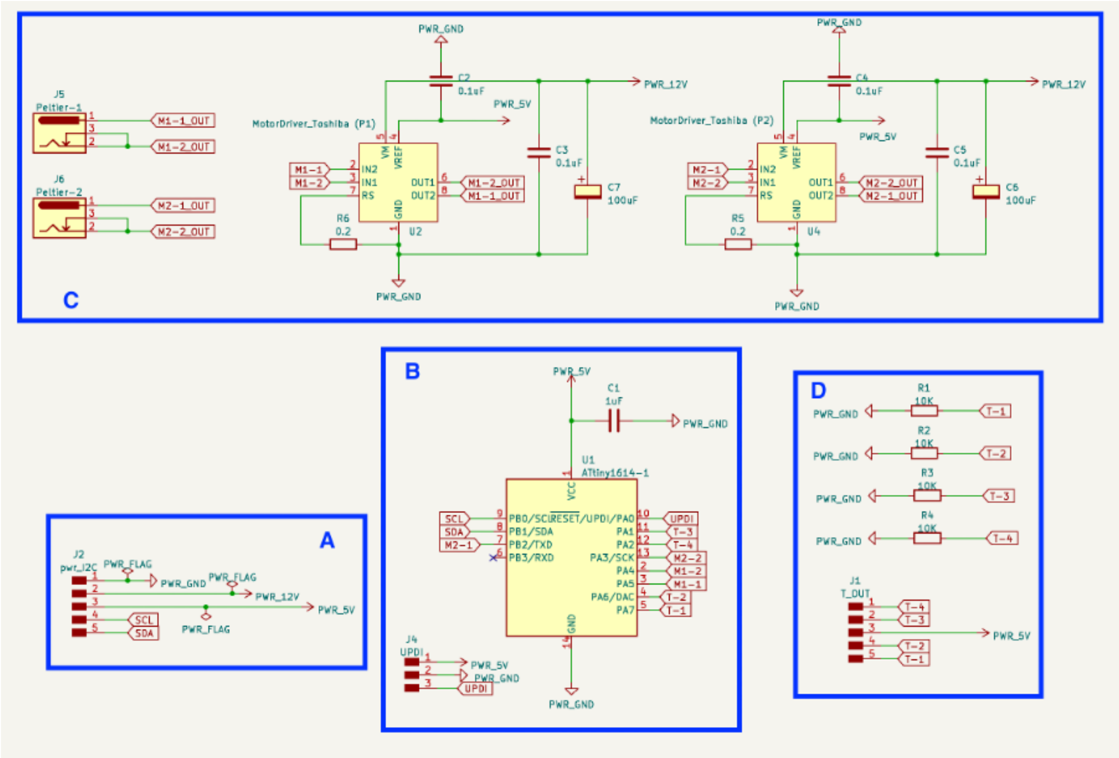

Figure 13 Schematic of the Peltier Slave module. The main power input is represented by PWR_12V in the schematic, but it can be any input voltage in the range of the board's limit.

A. Input power and I2C connection: There is one 5-pin JST-PH connector (J2) that combines the required power (PWR_GND, PWR_12V, PWR_5V) and I2C connection (SCL, SDA), which can be transferred from the Main Power board. The connector is the same type as each slave connection of the Main Power board, allowing the users to have more flexibility in arranging the Peltier Slave board with the Main Power board.
B. Microcontroller: The board is mainly controlled by the microcontroller ATtiny1614 (U1), which has a High-Performance Low-Power AVR® CPU. It is powered by the 5V from part A (PWR_5V). Three pins (J4) are out from the Attiny1614 to let the users program the board.
C. Peltier drivers: Two TB67H451FNG (U2, U4) for driving the Peltiers with PWM signal. The drivers adjust the input power (PWR_12V) with PWM signal from the microcontroller (M1-1, M1-2, M2-1, M2-2), outputting to the Peltiers (M1_1_OUT, M1_2_OUT, M2_1_OUT, M2_2_OUT). Two connection jacks (J5, J6) for connecting the Peltiers more conveniently and fixing the connecting direction.
D. Sensor connection: The board applies voltage divider with fixed 10K resistor. A 5-pin JST-PH connector (J1) for connecting four thermistors.

3.4.3 Electronic Components

Item	Quantity	Manufacture Part	Reference	Part description
1	1	ATTINY1614-SSNR	U1	IC MCU 8BIT 16KB FLASH 14SOIC
2	2	TB67H451FNGEL	U2, U4	50V/3A BRUSHED MOTOR DRIVER
3	2	S5B-PH-SM4-TB	J1, J2	CONN HEADER SMD R/A 5POS 2MM
4	4	RK73H2ATTD1002F	R1, R2, R3, R4	RES 10K OHM 1% 1/4W 0805
5	2	D1WEL0805CR200F-T5	R5, R6	RES 0.2 OHM 1% 3/4W 0805
6	1	C0805C105J3REC7800	C1	CAP CER 0805 1UF 25V X7R 5%
7	4	KGM21NR71H104JT	C2, C3, C4, C5	CAP CER 0.1UF 50V X7R 0805
8	2	35TPV100M6.3X6.1	C6, C7	CAP ALUM 100UF 20% 35V SMD
9	1	61300311121	J4	CONN HEADER VERT 3POS 2.54MM
10	2	PJ1-023-SMT-TR	J5, J6	CONN PWR JACK 0.7X2.35MM SOLDER

Table 2 Bill of materials of the Peltier Slave module.

3.4.4 PCB Fabrication

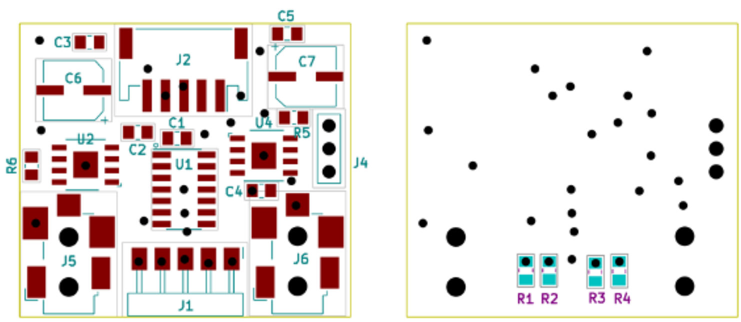

Figure 14 Assembly drawing (left: front side, right: back side) of the Peltier Slave module.

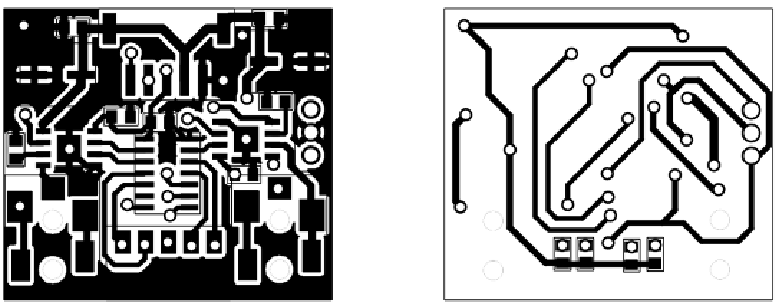

Figure 15 Fabrication drawing (left: front side, right: back side) of the Peltier Slave module.

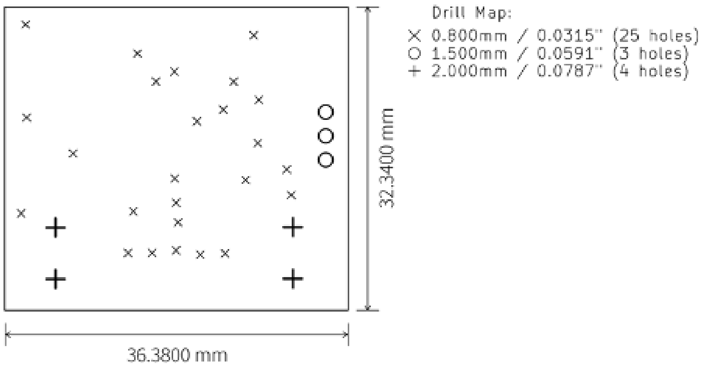

Figure 16 Fabrication drawing (drill drawing view) of the Peltier Slave module.

Here is the note for fabricating the Peltier Slave module:

- This board is a double-sided PCB. Most of the components are installed on the front side, except four resistors (R1, R2, R3, R4) are on the backside.
- The 0.8mm drill holes are for the smaller vias which can connect the front side and the back side of the PCB. The holes will be installed rivets (inner diameter: 0.6mm, outside diameter: 0.8mm).
- The 1.5mm drill holes are for the through-hole components. The holes will be installed rivets (inner diameter: 0.1mm, outside diameter: 1.5mm).
- The 2.0mm drill holes are for installing the power jacks (J5, J6), which are for connecting the Peltier modules.
- The traces are in different widths based on the circuit's purpose. The traces that go through the main power are 1.0mm in width, the traces that go through logic power (5V) are 0.6mm, and the others are 0.55mm.
- The front side is also the ground plan of the PCB.

3.5 Heater Slave Module

Heater Slave Module, shown in Figure 17, is a double-sided PCB. The primary features include:

- Input power can be 20 V maximum.
- Control up to two textile heaters, adjusting with PWM
- Each wire heater with one 10K thermistor for measuring the temperature.
- UPDI pins for programming the board.
- I2C connection with the master board.

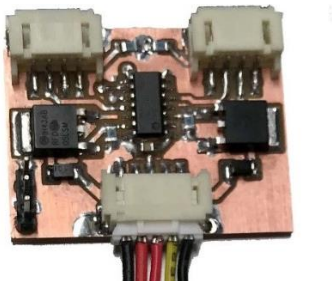

Figure 17 Heater Slave Module.

3.5.1 Theory of Operation

The Heater Slave Module allows the users to control the wire heaters easily with the I2C API. The board has two RFD16N05LSMs, N-Channel logic level power MOSFETs, which can drive high voltage and current with 5v low-voltage logic-level signals. It adjusts the output voltage with the PWM signal from the microcontroller Attiny1614, so the users can control the heating intensity of the wire heaters.

The board can read the temperature, by using a thermistor and a fixed 10K resistor in a voltage divider. The analog data read by ATtiny1614 is converted to Celsius degrees and sent back to the master board via I2C.

3.5.2 Circuit Design

The circuit design of a Heater Slave Module is shown in Figure 18. There are five parts comprising a Heater Slave module:

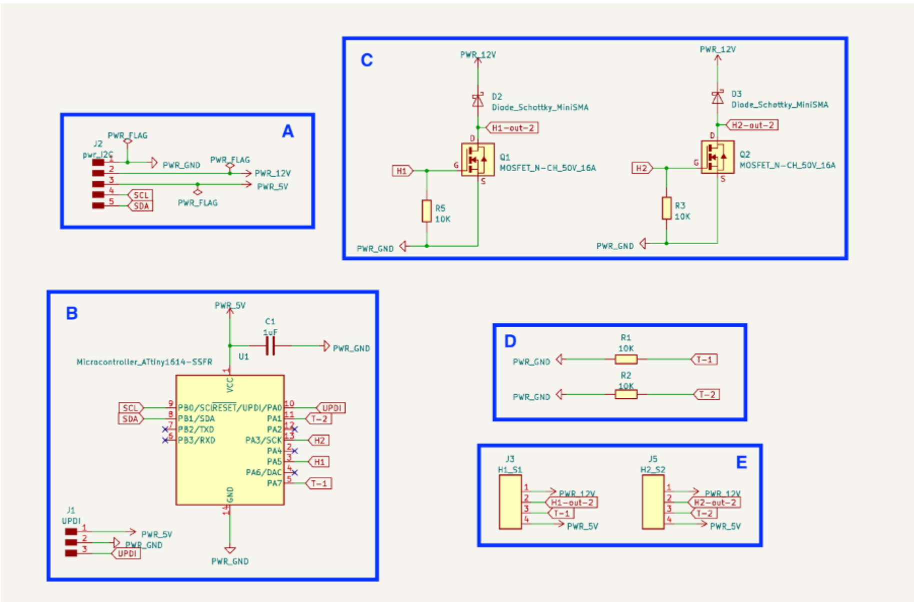

Figure 18 Schematic of the Heater Slave module. The main power input is represented by PWR_12V in the schematic, but it can be any input voltage in the range of the board's limit.

A. Input power and I2C connection: There is one 5-pin JST-PH connector (J2) that combines the required power (PWR_GND, PWR_12V, PWR_5V) and I2C connection (SCL, SDA), which can be transferred from the Main Power board. The connector is the same type as each slave connection of the Main Power board, allowing the users to have more flexibility in arranging the Heater Slave board with the Main Power board.
B. Microcontroller: The board is controlled by the microcontroller ATtiny1614 (U1), which has a High-Performance Low-Power AVR® CPU. It is powered by the 5V from part A (PWR_5V). Three pins (J1) are out from the Attiny1614 to let the users program the board.
C. Wire heater drivers: Two RFD16No5LSM MOSFETs (Q1, Q2) for driving the wire heaters with PWM signal. The MOSFETs adjust the input power (PWR_12V) with PWM signal from the microcontroller (H1, H2), outputting to the wire heaters (H1-out-2, H2-out-2). There is a 10K resistor (R3, R5) in the circuit as a gate pulldown resistor, preventing the MOSFET from turning on unintentionally.
D. Sensor output: The board applies voltage divider with fixed 10K resistor (R1, R2) to receive the output signal from the thermistor.
E. Heater and sensor connection: Two 4-pin JST-PH connectors (J3, J5) integrate one wire heater with one thermistor, making the wiring more organized.

3.5.3 Electronic Components

Item	Quantity	Manufacture Part	Reference	Part description
1	1	ATTINY1614-SSNR	U1	IC MCU 8BIT 16KB FLASH 14SOIC
2	2	RFD16N05LSM9A	Q1, Q2	MOSFET N-CH 50V 16A TO252AA
3	1	61300311121	J1	CONN HEADER VERT 3POS 2.54MM
4	1	S5B-PH-SM4-TB	J2	CONN HEADER SMD R/A 5POS 2MM
5	4	RK73H2ATTD1002F	R1, R2, R3, R5	RES 10K OHM 1% 1/4W 0805
6	1	C0805C105J3REC7800	C1	CAP CER 0805 1UF 25V X7R 5%
7	2	S4B-PH-SM4-TB	J3, J5	CONN HEADER SMD R/A 4POS 2MM
8	2	CDBM1100-G	D2, D3	DIODE SCHOT 100V 1A MINISMA

Table 3 Bill of materials of the Heater Slave module.

3.5.4 PCB Fabrication

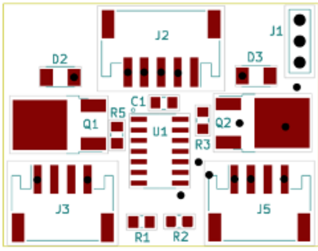

Figure 19 Assembly drawing of the Heater Slave module.

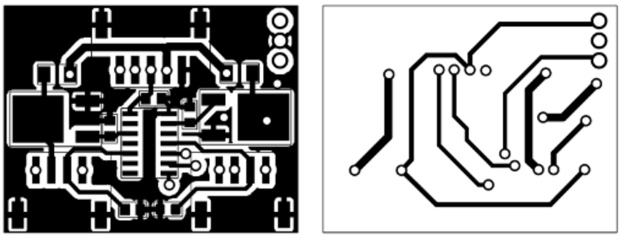

Figure 20 Fabrication drawing (left: front side, right: back side) of the Heater Slave module.

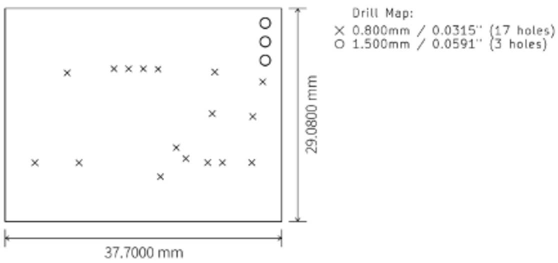

Figure 21 Fabrication drawing (drill drawing view) of the Heater Slave module.

Here is the note for fabricating the Heater Slave module:

- This board is double-sided. All the components are installed on the front side.
- The 0.8mm drill holes are for the smaller vias which can connect the front side and the back side of the PCB. The holes will be installed rivets (inner diameter: 0.6mm, outside diameter: 0.8mm).
- The 1.5mm drill holes are for the through-hole components. The holes will be installed rivets (inner diameter: 0.1mm, outside diameter: 1.5mm).
- The traces are in different widths based on the circuit's purpose. The traces that go through the main power are 1.0mm in width, the traces that go through logic power (5V) are 0.6mm, and the others are 0.5mm.
- The front side is also the ground plan of the PCB.

3.6 Vibration Motor Slave Module

Vibration Motor Slave module is a double-sided PCB. The primary features include:

- Operating with 5V power.
- Control up to two ERM vibration motors or any 5V DC motor with PWM signal.
- UPDI pins for programming the board.
- I2C connection with the master board.

3.6.1 Theory of Operation

The Motor Slave module allows the users to control vibration motor or any DC motor easily with I2C API. The board is equipped with two NDS355ANs, which are N-channel transistors. It adjusts the output voltage with the PWM signal from the microcontroller Attiny412, so the users can control the intensity of the motor.

3.6.2 Circuit Design

The circuit design of a Vibration Motor Slave Module is shown in Figure 22. There are three parts comprising a Motor Slave module:

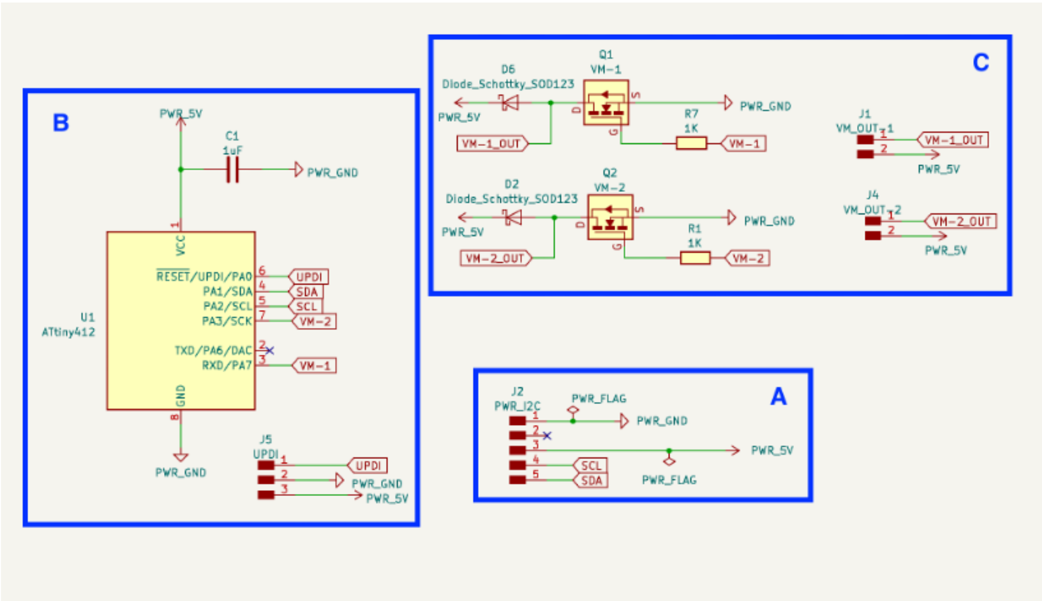

Figure 22 Schematic of the Vibration Motor Slave module.

A. Input power and I2C connection: There is one 5-pin JST-PH connector (J2) that combines the required power (PWR_GND, PWR_5V) and I2C connection (SCL and SDA), which can be transferred from the Main Power board. The connector is the same type as each slave connection of the Main Power board, allowing the users to have more flexibility in arranging the Vibration Motor Slave board with the Main Power board.

B. Microcontroller: The board is mainly controlled by the microcontroller ATtiny412 (U1), which has a High-Performance Low-Power AVR® CPU. It is powered by the 5V from part A (PWR_5V). Three pins (J5) are out from the Attiny1614 to let the users program the board.

C. Motor driving: Two NDS355AN (Q1, Q2) for driving the motors with PWM signal. The transistors adjust the input power (PWR_5V) with the PWM signal from the microcontroller (VM-1, VM-2), outputting to the motors (VM-1-OUT, VM-2-OUT). Two 2-pin headers (J1, J4) for connecting the motors. Two Schottky diodes (D2, D6) for protecting the driving circuits.

3.6.3 Electronic Components

Item	Quantity	Manufacture Part	Reference	Part description
1	1	ATTINY412-SSFR	U1	IC MCU 8BIT 4KB FLASH 8SOIC
2	1	C0805C105J3REC7800	C1	CAP CER 0805 1UF 25V X7R 5%
3	2	NDS355AN	Q1, Q2	MOSFET N-CH 30V 1.7A SUPERSOT3
4	2	RNCP0805FTD1K00	R1, R7	RES 1K OHM 1% 1/4W 0805
5	2	61300211121	J1, J4	CONN HEADER VERT 2POS 2.54MM
6	1	S5B-PH-SM4-TB	J2	CONN HEADER SMD R/A 5POS 2MM
7	1	61300311121	J5	CONN HEADER VERT 3POS 2.54MM
8	2	BAT46ZFILM	D2, D6	DIODE SCHOTTKY 100V 150MA SOD123

Table 4 Bill of materials of the Motor Slave module.

3.6.4 PCB Fabrication

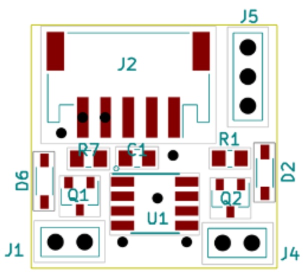

Figure 23 Assembly drawing of the Motor Slave module.

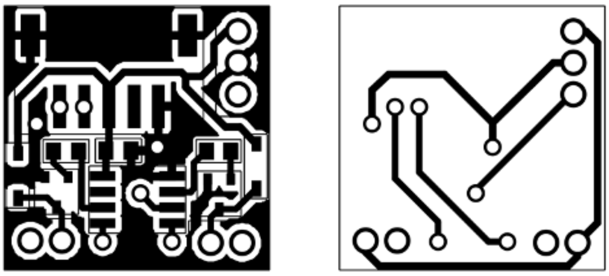

Figure 24 Fabrication drawing (left: front side, right: back side) of the Motor Slave module.

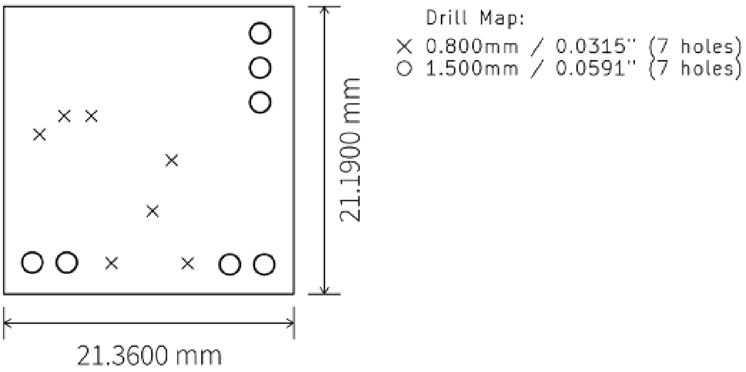

Figure 25 Fabrication drawing (drill drawing view) of the Motor Slave module.

Here is the note for fabricating the Motor Slave module:

- This board is a double-sided PCB. All the components are assembled on the front side.
- The 0.8mm drill holes are for the smaller vias which can connect the front side and the back side of the PCB. The holes will be installed rivets (inner diameter: 0.6mm, outside diameter: 0.8mm).
- The 1.5mm drill holes are for the through-hole components. The holes will be installed rivets (inner diameter: 0.1mm, outside diameter: 1.5mm).
- The traces that go through the main power are 0.6 mm in width, and the others are 0.5mm.
- The front side is also the ground plan of the PCB.

3.7 Thermistor Slave Module

Thermistor Slave Module is a double side PCB. The primary features include:

- Operating with 5V power.
- Connect with up to three 10K thermistors or three 10K resistive sensors.
- UPDI pins for programming the board.

- I2C connection with the master board.

3.7.1 Theory of Operation

Thermistor Slave module is designed for reading extra temperature data or other sensor applications. With three voltage dividers, the board can connect with up to three 10K thermistors or other types of resistive sensors. The analog data read by ATtiny412 is converted to Celsius degree and sent back to the master board via I2C.

3.7.2 Circuit Design

The circuit design of a Thermistor Slave Module is shown in Figure 26. There are three parts consisting of a Thermistor Slave module:

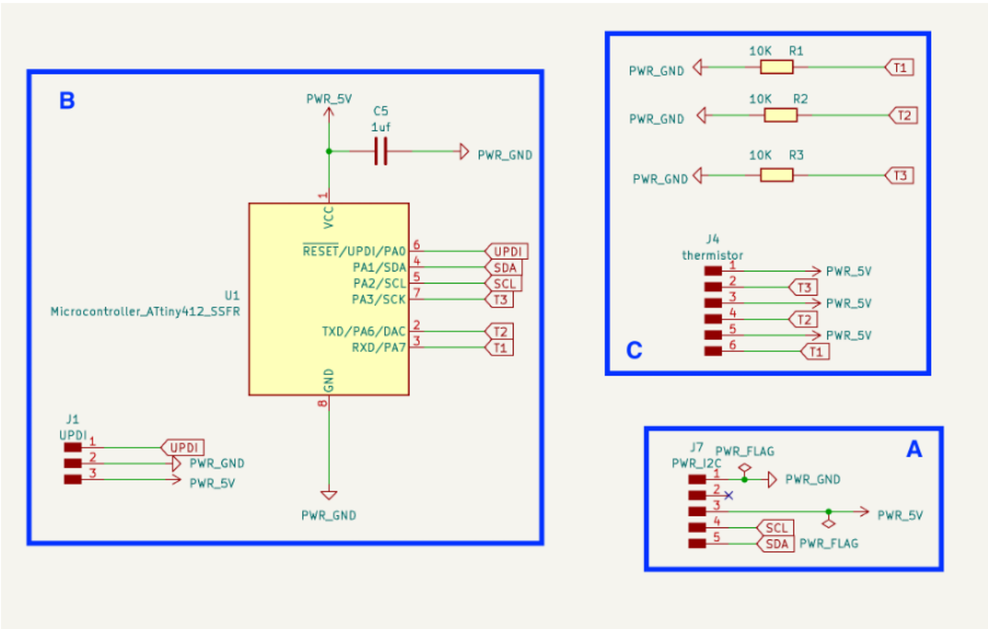

Figure 26 Schematic of the Thermistor Slave module.

A. Input power and I2C connection: There is one 5-pin JST-PH connector (J7) that combines the required power (PWR_GND, PWR_5V) and I2C connection (SCL, SDA), which can be transferred from the Main Power board. The connector is the same type as each slave connection of the Main Power board, allowing the users to have more flexibility in arranging the Thermistor Slave board with the Main Power board.
B. Microcontroller: The board is mainly controlled by the microcontroller ATtiny412 (U1), which has a High-Performance Low-Power AVR® CPU. It is powered by the 5V from part A (PWR_5V). Three pins (J1) are out from the Attiny412 to let the users program the board.

C. Sensor reading and connection: The board applies three voltage dividers with three fixed 10K resistors (R1, R2, R3). A 6-pin header (J4) for connecting up to three sensors (T1, T2, T3) respectively.

3.7.3 Electronic Components

Item	Quantity	Manufacture Part	Reference	Part description
1	1	ATTINY412-SSFR	U1	IC MCU 8BIT 4KB FLASH 8SOIC
2	1	61300311121	J1	CONN HEADER VERT 3POS 2.54MM
3	3	RK73H2ATTD1002F	R1, R2, R3	RES 10K OHM 1% 1/4W 0805
4	1	C0805C105J3REC7800	C5	CAP CER 0805 1UF 25V X7R 5%
5	1	61300611121	J4	CONN HEADER VERT 6POS 2.54MM
6	1	S5B-PH-SM4-TB	J7	CONN HEADER SMD R/A 5POS 2MM

Table 5 Bill of materials of the Thermistor Slave module.

3.7.4 PCB Fabrication

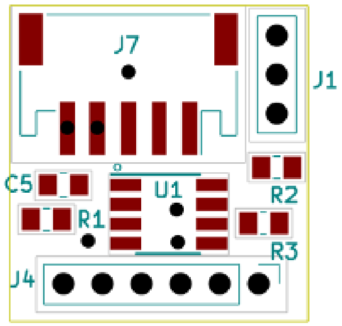

Figure 27 Assembly drawing of the Thermistor Slave module.

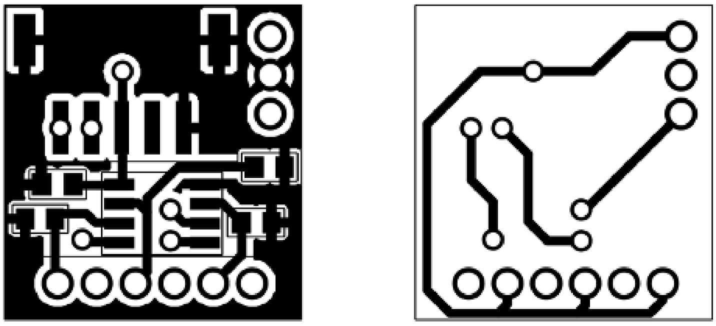

Figure 28 Fabrication drawing (left: front side, right: back side) of the Thermistor Slave module.

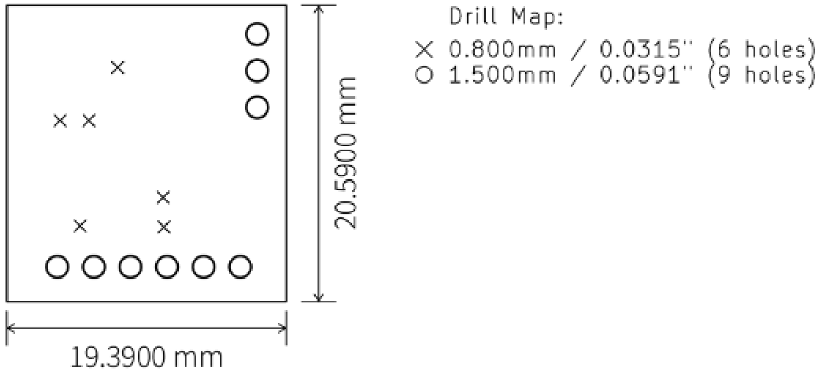

Figure 29 Fabrication drawing (drill drawing view) of the Thermistor Slave module.

Here is the note for fabricating the Thermistor Slave module:

- This board is a double-sided PCB. All the components are assembled on the front side.
- The 0.8mm drill holes are for the smaller vias which can connect the front side and the back side of the PCB. The holes will be installed rivets (inner diameter: 0.6mm, outside diameter: 0.8mm).
- The 1.5mm drill holes are for the through-hole components. The holes will be installed rivets (inner diameter: 0.1mm, outside diameter: 1.5mm).
- The traces are 0.6 mm in width.
- The front side is also the ground plan of the PCB.

4 API Specification

This chapter gives an overview of the API used to control the haptic garment and retrieve sensor data from it. The API files can be downloaded from the project website: https://emil-xr.eu.

The functional requirement set for the API is to allow individually controlling each actuator, creating any number of different heating and vibrotactile patterns, as well as reading the sensors, and controlling the actuators based on the sensor readings. To achieve this, the API provides different levels of abstraction and functionality for controlling the haptic garment: low-level, mid-level and high-level.

The low-level tier of the API is based on the I2C protocol, which allows communication between a master board and multiple slave boards. The slave boards are responsible for driving the actuators and sensors of the wearable device. The low-level tier of the API defines the commands and data formats that can be used to send and receive information between the master and slave boards.

The mid-level tier of the API is based on serial or wireless communication between a software application and the master board. It defines the commands and data formats that can be used to control the haptic garment from the software application. The mid-level tier of the API allows both individual and global control of the actuators and sensors, such as setting the vibration intensity or desired temperature of a specific actuator or shutting down all actuators at once.

The high-level tier of the API is based on a library that can be integrated into the software application. it defines the functions and data structures that can be used to perform advanced control of the haptic garment from the software application. The high-level tier of the API allows complex and dynamic control of the actuators and sensors, such as playing out a heating pattern over certain heating elements or creating custom haptic effects.

An overview of the communication hierarchy is shown in Figure 30. The following sections detail the functionalities available on each tier of the API.

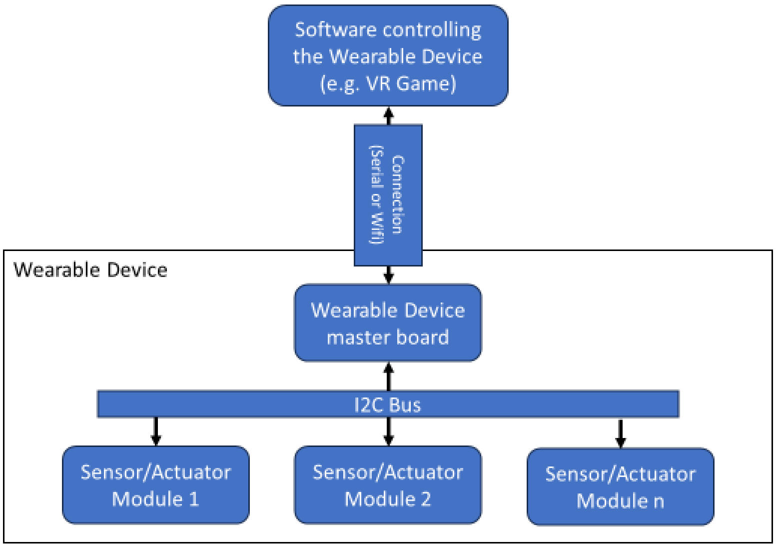

Figure 30 Communication hierarchy of the wearable device.

4.1 I2C API

The I2C API aims to provide minimum required API with access to all actuator and sensor functionalities, while keeping memory and calculation constraints minimal on the slave boards. This is so that the microcontrollers can function with their limited memory and computing capabilities.

As the I2C protocol follows master-slave architecture, the commands are in the form of requests. The master sends a request, after which the slave executes the command and optionally sends a response.

The master must first decide which address the message is sent to. Slave board addresses are defined when programming the boards. The message is a series of bytes containing all the parameters of the command. The requests start with the command type, which is a byte with a value between 0 and 255, followed by additional parameters, e.g. [1,255,1,255,1,1000] for setting intensity to 255 on two Peltier elements for a duration of 1000 milliseconds. The message length is pre-defined, and the message is parsed as such, so the parameters must be in the correct order. Likewise, any response is a series of bytes, which is parsed back into different data types by the master board.

4.1.1 Installation

The API must be installed on the slave boards on a per-board level. The program code is chosen depending on the type of board. New boards and functionalities can be supported by editing the existing code files. Before uploading the code, the user should define the desired address for the board, by changing it in the code. Afterwards, the code can be uploaded to the board using a UPDI programmer.

4.1.2 API Commands

The API commands for the different slave boards are listed in Tables Table 6 through Table 9. The GSR sensor has its own API, which is documented on the manufacturer website: https://www.mikroe.com/ecg-gsr-click.

Table 6 Peltier slave I2C commands

Peltier Actuator				
Command	**Parameters**	**Data Type**	**Response**	**Data Type**
Get Type	command	command – byte, 0	type	type – char, ‘p’
Set Intensity	command intensity1 direction1 intensity2 direction2 duration	command – byte, 1 intensity – byte, 0-255 direction – byte, 0-1 duration – long, 0-2147483647	-	-
Read Sensor	command	command – byte, 2	temperature1 temperature2 temperature3 temperature4	temperature – byte, 0-255
Set Minimum and Maximum values	command minimumSide1 maximumSide1 minimumSide2	command – byte, 3 minimum – byte, 0-50	-	-

	maximumSide2	maximum – byte, 0-50		
Set target temperature	command target1 target2 duration	command – byte, 4 target – byte, 0-255 duration – long, 0-2147483647	-	-

Note that regarding the Peltier orientation, the API is configured such that when the direction is set to 0, side 1 should be heating up. Keeping this in mind, thermistors 1 and 3 should be placed on side 1 of the two Peltiers respectively, and thermistors 2 and 4 should be placed on side 2. When calling set temperature target, the desired temperature is set for side 1, within the limits of the minimum and maximum values for each side, and the capabilities of the Peltier element.

Table 7 Heater slave I2C commands

Heater Actuator				
Command	Parameters	Data Type	Response	Data Type
Get Type	command	command – byte, 0	type	type – char, ‘h’
Set Intensity	command intensity1 intensity2 duration	command – byte, 1 intensity – byte, 0-255 duration – long, 0-2147483647	-	-
Read Sensor	command	command – byte, 2	temperature1 temperature2	temperature – byte, 0-255
Set Maximum value	command maximum	command – byte, 3 maximum – byte, 0-255	-	-
Set target temperature	command target1 target2 duration	command – byte, 4 target – byte, 0-255 duration – long, 0-2147483647	-	-

Table 8 Vibration motor slave I2C commands

Vibration Motor Actuator				
Command	Parameters	Data Type	Response	Data Type
Get Type	command	command – byte, 0	type	type – char, 'v'
Set Intensity	command intensity1 intensity2 duration	command – byte, 1 intensity – byte, 0-255 duration – byte, 0-2147483647	-	-

Table 9 Temperature sensor slave I2C commands

Temperature Sensor				
Command	Parameters	Data Type	Response	Data Type
Get Type	command	command – byte, 0	type	type – char, 't'
Read Sensor	command	command – byte, 1	temperature1 temperature2 temperature3	temperature – byte, 0-255

4.2 Master Board API

The master board API aims to provide access to any one of the individual slave boards and actuators, while implementing some whole wearable control functions, such as shutting off all actuators, or listing all connected slave boards.

For the master board, we implement 2 versions of the API: Serial and Wi-Fi. The communication over both is done using the same API specification, with the same message format, and the Wi-Fi socket is used as if it was a Serial connection, rather than using HTTP for example. This ensures that the requirements for the master-board remain similar, and we can use Arduino MRK Wi-Fi 1010 for both.

Like the I2C API, the messages have a pre-defined format and length for easy parsing. The message format, without spaces, is: APIType CommandNumber Parameters, e.g., p 0 001 1 255 1 255 1000 for setting intensity to 255 on two Peltier elements for a duration of 1000 milliseconds on board 001. The responses are returned as JSON-objects, as that format is easy to parse on the receiving side.

4.2.1 Installation

To have access to the API, you need to upload the appropriate master board code to the master board. To pick the version of the API, you need to upload either i2c_master_serial.ino or i2c_master_wifi.ino to the master board. The GSR Sensor API needs to be enabled or disabled in the code by changing the value of the variable gsrEnabled to true or false respectively. When the GSR Sensor API is enabled, no slave board should occupy the addresses 48, 96 or 107, as these are taken up by the GSR Sensor.

4.2.2 API Commands

Table 10 Master board API general commands

General API				
Command	Request Parameters	Data Type	Response Parameters	Data Type
Get Devices	-	command – 0	{ devices: [(id, type)] }	id – 1-127 type – char
Soft Shutdown	-	command – 1	{ code: code}	code – 0 (ok) / -1 (error)
Hard Shutdown	-	command – 2	{ code: code}	code – 0 (ok) / -1 (error)
Set Temperature Multiplier	intensity	command – 3 intensity – 0.0-1.0	{ code: code}	code – 0 (ok) / -1 (error)
Set Vibration Multiplier	intensity	command – 4 intensity – 0.0-1.0	{ code: code}	code – 0 (ok) / -1 (error)

Table 11 Master board API Peltier commands

Peltier API				
Command	Request Parameters	Data Type	Response Parameters	Data Type
Set Intensity	address intensity1 direction1 intensity2 direction2 duration	command – 0 address – 001-127 intensity – 000-255 direction – 0/1 duration – 0-2147483647	{ code: code, msg: message }	code – 0 (ok) / -1 (error) message – string

Read Sensor	address	command – 1 address – 001-127	{ ts: timestamp, values: [value]} Or { code: code, msg: message }	timestamp – int (ms) value – int (0-50) code – -1 (error) message – string
Set Minimum and Maximum values	address minimumSide1 maximumSide1 minimumSide2 maximumSide2	command – 2 address – 001-127 minimum – 000-255 maximum – 000-255	{ code: code, msg: message }	code – 0 (ok) / -1 (error) message - string
Set target temperature	address target1 target2 duration	command – 3 address – 001-127 target – 000-255 duration – 0-2147483647	{ code: code, msg: message }	code – 0 (ok) / -1 (error) message - string

Table 12 Master board API heater commands

Wire Heater API				
Command	Request Parameters	Data Type	Response Parameters	Data Type
Set Intensity	id intensity1 intensity2 duration	command – 0 id – 001-127 intensity – 000-255 duration – 000-2147483647	{ code: code, msg: message }	code – 0 (ok) / -1 (error) message – string

Read Sensor	Id	command – 1 id – 001-127	{ ts: timestamp, values: [value]} Or { code: code, msg: message }	timestamp – int (ms) value – int (0-50) code – -1 (error) message – string
Set Maximum temperature	id maximum	command – 2 id – 001-127 maximum – 0-255	{ code: code, msg: message }	code – 0 (ok) / -1 (error) message – string
Set target temperature	id target1 target2	command – 3 id – 001-127 target – 0-255	{ code: code, msg: message }	code – 0 (ok) / -1 (error) message – string

Table 13 Master board API vibration motor commands

Vibrator API				
Command	Request Parameters	Data Type	Response Parameters	Data Type
Set Intensity	address intensity1 intensity2 duration	command - 0 address - 001-127 intensity - 000-255 duration - 0-2147483647	{ code: code, msg: message }	code – 0 (ok) / -1 (error) message – string

Table 14 Master board API Temperature sensor commands

Temperature Sensor API				
Command	Request Parameters	Data Type	Response Parameters	Data Type
Read Sensor	Id	command - 0 id - 001-127	{ ts: timestamp, values: [value] }	timestamp – int (ms) value – int (0-50)

			Or { code: code, msg: message }	code – -1 (error) message – string

Table 15 Master board API GSR sensor commands

GSR Sensor API				
Command	Request Parameters	Data Type	Response Parameters	Data Type
Initialize	-	command – 0	{ code: code, msg: message }	code – 0 (ok) / -1 (error) message – string
Read Sensor	-	command – 1	{ ts: timestamp, value: [value]} Or { code: code, msg: message }	timestamp – int (ms) value – int (0-50) code – -1 (error) message - string

4.3 API Library

On the application side, we implemented a library of functions, that can be called to access the different API functionalities, without having to write the control messages manually. This way the communication protocol is abstracted away, and the experience designer can access the devices directly by calling the functions. This makes it convenient for the user to create different patterns, while retaining individual-level access to all the sensors and actuators.

4.3.1 Installation

The library is implemented for Unreal Engine and is accessible in WearableAPI.uasset. It has been tested with both Unreal Engine 4 and 5.

As prerequisites, the project needs to have the SerialCOM plugin for serial communication and TCP Socket plugin for wireless communication. The WearableAPI and TCP Socket are actors, which need to be placed in the current level. Then, the address of the board can be configured by editing the WearableAPI actor.

It is possible to have multiple master boards connected to the same software instance simultaneously. In this case, you need to have multiple instances of TCP Socket and WearableAPI actors in the level.

A library with similar functionalities can be implemented for other languages. The developer will need to select the communication medium, either network socket or serial connection, and follow the message format of the Master Board API.

4.3.2 API Commands

Table 16 API Library general command functions

General API Functions			
Command	Description	Parameters	Data Type
Get Devices	Returns list of devices, along with IDs and types	-	-
Soft Shutdown	Shutdown all actuators	-	-
Hard Shutdown	Shutdown all actuators by cutting access to power supply	-	-
Set Temperature Multiplier	Multiply pwm intensity values for temperature actuators with this value	intensity	intensity - 0.0-1.0
Set Vibration Multiplier	Multiply pwm intensity values for vibration actuators with this value	intensity	intensity - 0.0-1.0

Table 17 API Library Peltier command functions

Peltier API Functions			
Command	Description	Parameters	Data Type
Set Intensity	Set pwm value and direction for peltiers	id intensity1 direction1	id - 1-127 intensity - 0-255

		intensity2 direction2 duration	direction - 0/1 duration - int
Read Sensor	Read all sensors	id	id - 1-127
Set Minimum and Maximum values	Turn pwm intensity to 0, if maximum reached while heating, or minimum reached while cooling	id minimumSide1 maximumSide1 minimumSide2 maximumSide2	id - 1-127 minimum - 0-50 maximum - 0-50
Set target temperature	Turn pwm intensity to 255, until reaching temperature, then fluctuate 1 degree up and down	id target1 target2	id - 1-127 target - 0-50

Table 18 API Library heater command functions

Wire Heater API Functions			
Command	Description	Parameters	Data Type
Set Intensity	Set pwm value	id intensity1 intensity2 duration	id - 1-127 intensity - 0-255 duration - int
Read Sensor	Read all sensors	id	id - 1-127
Set Maximum value	Turn pwm intensity to 0, if maximum reached while heating, or minimum reached while cooling	id maximum	id - 1-127 minimum - 0-50 maximum - 0-50
Set target temperature	Turn pwm intensity to 255, until reaching temperature, then fluctuate 1 degree up and down	id target1 target2	id - 1-127 target - 0-50

Table 19 API Library temperature sensor command functions

Temperature Sensor API Functions			
Command	Description	Parameters	Data Type
Read Sensor	Read all sensors	id	id - 1-127

5 Textile Integration

This chapter describes the methods and techniques for integrating the electronic components into the textiles, using different types of structures and connectors, as well as building textile-based circuitry. The first section discusses creating connections between hard components and textiles, both conductive and non-conductive, and the second section discusses electric circuit integration into textiles.

5.1 Connecting hard components to textile surfaces

5.1.1 Press buttons

Press buttons are frequently used in textiles as a convenient way to connect and disconnect parts. In textile electronics, they not only connect the components to the textile. Press buttons made of conductive metal can be used to create electronic connections. This allows for attaching and detaching components, PCBs, layers, and conductive traces. Figure 31 presents a commercially available example of a development kit with press button connected PCB and electronic components.

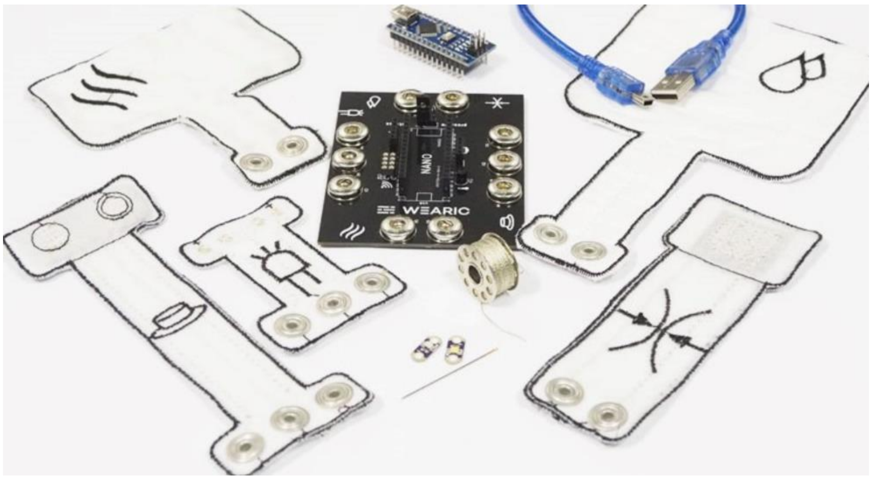

Figure 31 Wearic Smart Textile Kit[1]

[1] https://www.wearic.com/product/smart-textiles-kit-wit-nano-controller/

5.1.2 Button-hole connectors on PCBs

Small PCBs can be designed with perforations that allow for easy stitching onto fabric surfaces, providing a secure and visually unobtrusive method for embedding electronic components into clothing or other textile-based items.

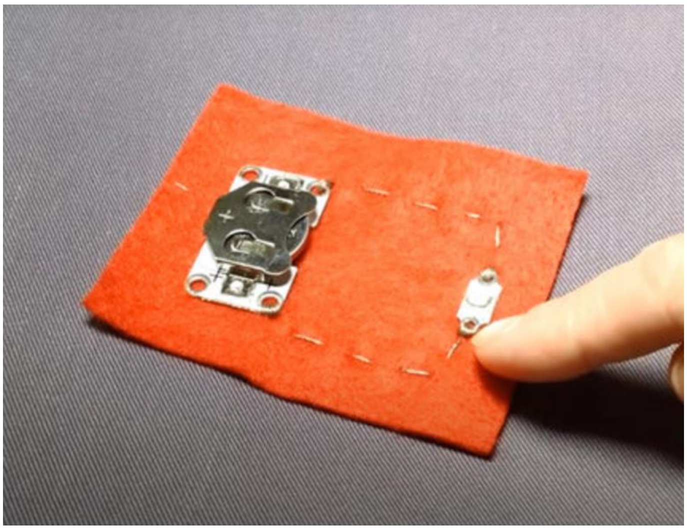

Figure 32 Battery holder PCB with perforations stitched onto a fabric

5.1.3 Magnetic connectors

Magnetic connectors allow for easy and quick attachment and detachment, facilitating the removal of electronic components for maintenance or washing. This method ensures a streamlined and aesthetically pleasing design, as there are no visible clasps or cumbersome attachment mechanisms. The flexibility of magnetic connectors also accommodates the dynamic movements of the textile, making them particularly well-suited for applications in wearable electronics, where comfort and adaptability are paramount. However, magnets can induce current into conductive traces, when moving, potentially causing signal noise.

5.1.4 Velcro for component placement

Velcro, with its hook-and-loop fastening system, presents an efficient and user-friendly solution for component placement in various applications. Its strong grip ensures stability, and the ability to cut Velcro strips to custom sizes enhances flexibility in design. The modular nature of Velcro enables easy reconfiguration, repositioning, and adjustment of components. The Velcro material can even be conductive itself, as shown in Figure 33, allowing for current to flow when connected.

Figure 33 Conductive Velcro by Light Stiches[2]

5.2 Electric circuit integration

5.2.1 Conductive traces

Conductive traces are needed for carrying power and signals between the electronical components. Ideally, the traces should offer low resistance for clean signalling and loss. The conductive traces can be made using yarns or wires integrated into the textile, or by using a conductive textile as the material.

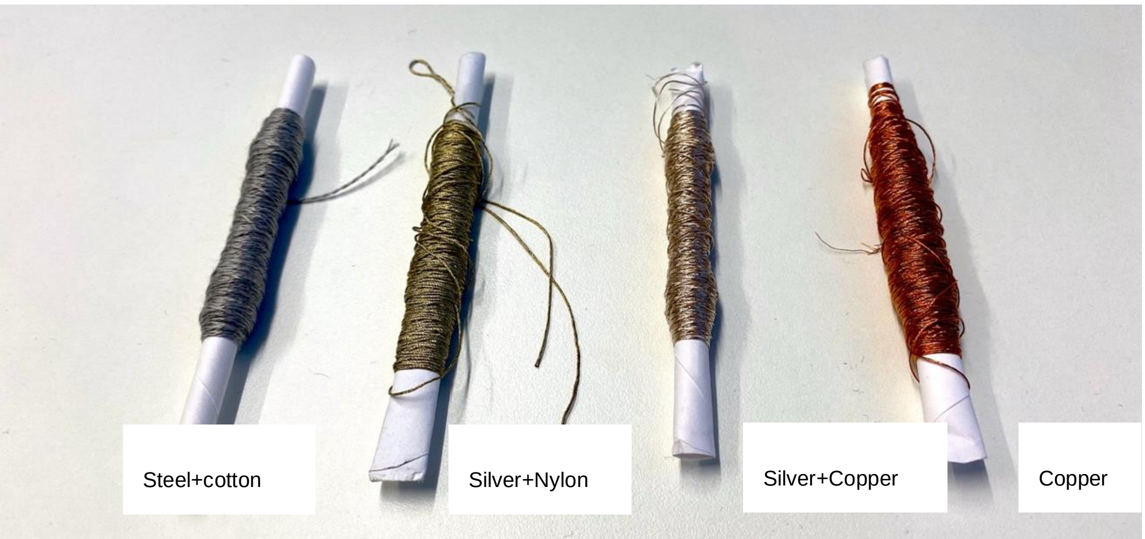

Figure 34 Examples of different kinds of commercially available conductive yarns

[2] https://lightstitches.co.uk/

Figure 35 Conductive traces of various strengths made by sewing copper wires

Figure 36 Conductive traces made of electrically conductive fabric strips

5.2.2 Insulation strategies

To prevent short circuits and ensure reliable performance, insulation is needed between the conductive traces. A layer of the base-textile can act as an insulation, if it is made from a non-conductive material. Other alternatives such as tape or glue also allow insulating, while maintaining the textile's flexible nature. The selection of the particular insulation material depends on other requirements, such as breathability, washability and durability.

5.2.3 Layering

Insulating the electric circuits can be done by layering non-conductive textile layers in between the conductive parts. This also allows for crossing electric traces, similar to multi-layer PCBs. Components can be placed on different layers, for example conductive layers for carrying signals and power, and functional layers for actuator and sensor placement.

6 Application Examples

The thermal and tactile garment has a versatile range of applications across various contexts. They play a pivotal role in enhancing immersion and embodiment within the realm of VR, as they are designed to stimulate the player's sense of touch and temperature, thereby making virtual experiences more lifelike and engaging. Using the API, one can activate actuators located in various parts of the body and customize thermo-tactile patterns by adjusting parameters like location, duration, and intensity. Furthermore, it is feasible to generate dynamic patterns by activating thermal or tactile actuators in quick succession, resulting in a sensation of motion.

The garment is not limited to gaming alone but also finds utility in guiding user awareness to specific body regions through the activation of location-specific actuators. This capability has far-reaching implications, extending to applications such as mindfulness meditation, where the garments can help individuals maintain focus and tranquillity. Addressing health and well-being, thermal garments can be used to ease individuals' pain in joints or discomfort through controlled warm or cold feedback. Lastly, thermal and tactile feedback integrated into textiles might serve for navigation purposes, by providing directional cues. Smart garments capable of providing multimodal haptic feedback are at the forefront of merging technology with everyday life, promising exciting new possibilities for enhanced experiences and improved well-being.

In the Lighthouse project, the modular design has been used to build a thermal sleeve, shown in Figure 37, to be used in the context of a VR sauna game. The hardware used consists of a main power board, and multiple Peltier slave boards. When the player pours water on the stove in the game, the API is called to create a moving pattern of heat along the hand of the user, while the user simultaneously sees steam in the game and hears water being turned to steam. The thermal pattern adds an additional layer of immersion to the experience. In the next iterations, we intend to extend the sleeve to cover a larger area of the user's body, and to use a more advanced control logic for creating patterns.

Figure 37 Thermo-haptic sleeve with attached Peltier elements and vibration motors that allows for the programming of thermal and haptic patterns such as a moving sensation by calling the actuators successively.

7 Conclusions

This document describes a modular wearable haptic system and an API for controlling and accessing its components, as well as its integration into textiles. The presented designs can be used as is or as a starting point for developing own designs, with different layout of components and additional API commands. The application examples demonstrate how the system works in various use contexts including gaming or health and well-being.