Module Interface Specification: SE 4G06, TRON 4TB6

Team 26, STRONE
Jordan Bierbrier
Azriel Gingoyon
Taranjit Lotey
Udeep Shah
Abraham Taha

April 5, 2023

Revision History

Date	Version	Notes
1/18/2023	1.0	Added Content
4/5/2023	1.1	Updated for feedback and consistency

Contents

List of Figures

List of Tables

Symbols, Abbreviations and Acronyms

See SRS Documentation at *SRS.pdf Document Link*.

1 Introduction

The following document details the Module Interface Specifications for Synesthesia Wear, a wearable ~~product~~ device that assists users by using signal processing on gathered sounds to provide appropriate feedback (via vibrations) to the user according to inputted sound configuration settings. As a result, this gives the users peace of mind knowing that if their attention is needed (doorbell, ring, name call, etc.), Synesthesia Wear will be able to alert them.
Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at *Team 26 Capstone GitHub Repository*.

2 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.
The following table summarizes the primitive data types used by ~~SE 4G06, TRON 4TB6~~ STRONE.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)

Table 1: Data Notations

The specification of ~~SE 4G06, TRON 4TB6~~ STRONE uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, ~~SE 4G06, TRON 4TB6~~ STRONE uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

3 Module Decomposition

The following table is taken directly from the *Module Guide Document* for this project.

Level 1	**Level 2**
Hardware-Hiding Module (M1)	
Behaviour-Hiding Module	~~Login Module~~ (Design Decision was made to remove authentication/login aspects of the project) Bluetooth connection Module (M2) Keyword Selection Module (M3) Output Signal Module (M4) ~~Profile Module~~ ~~Battery Status Module~~ Out of Scope
Software Decision Module	Sound Classification Module (M5) Bluetooth Communication Module (M6) Microphone Module (M7)

Table 2: Module Hierarchy

4 ~~MIS of Login Module~~ Design Decision - Removed

4.1 Module

Login Type

4.2 Uses

N/A

4.3 Syntax

4.3.1 Exported Constants

N/A

4.3.2 Exported Access Programs

Name	In	Out	Exceptions
authorize	keyInput	Authorized	
login	keyInput		Not_A_Character
switchCue	keyInput		Not_Cueable

4.4 Semantics

4.4.1 State Variables

Authorized: Authorized is a boolean that is true when the correct credentials are entered and false otherwise.

4.4.2 Environment Variables

keyInput: {key.Enter, key.AlphabetCharacters, key.LeftClick}

4.4.3 Assumptions

The Synesthesia Wear application is successfully installed on the user's device and the login page has loaded onto the screen.

4.4.4 Access Routine Semantics

switchCue(key.LeftClick):

- transition: mouseLocation.navigate()

- output: None
- exception: Not_Cueable

login(key.AlphabetCharacters):

- transition: username.addCharacter() or password.addCharacter()
- output: None
- exception: Not_A_Character

authorize(key.Enter):

- transition: login.submit()
- output: Authorized
- exception: None

5 MIS of Bluetooth Connection Module (M2)

5.1 Module

Bluetooth Connection Module (M2)

5.2 Uses

N/A

5.3 Syntax

5.3.1 Exported Constants

N/A

5.3.2 Exported Access Programs

Name	In	Out	Exceptions
Find_Device	Boolean onClick	Device_ID Array	device_NotFound
Connect_Device	(Boolean, onClick Device_ID)		

5.4 Semantics

5.4.1 State Variables

- known_Device: Device_ID $\mathbb{N}$
- Paired_Device

5.4.2 Environment Variables

N/A

5.4.3 Assumptions

We assume that the mobile device that the user has the application on has a bluetooth module.

5.4.4 Access Routine Semantics

Find_Device(Boolean onClick):

- transition: (Onclick $\equiv$ True) $\implies$ Found_Devices = $\implies$ Found_Devices.Add(known_Devices) $\implies$ Found_Devices.Add(device_id1)
- output: Found_Devices(known_Device, device_id1, device_id2 ... device_idN)
- exception: Device_NotFound

Connect_Device(Boolean onClick):

- transition: (Onclick $\equiv$ True) $\implies$ Paired_Device = device_id2
- output: None
- exception: None

6 MIS of Keyword Selection Module (M3)

6.1 Module

Keyword Selection Type (M3)

6.2 Uses

- ~~Profile Module~~ (Design Decision - Removed)
- Bluetooth Communication Module (M6)

6.3 Syntax

6.3.1 Exported Constants

N/A

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
save	keyInput	Saved	Mouse_Not_On_Save_Button
keyword	keyInput		Not_A_Character
switchCue	keyInput		Not_Cueable

6.4 Semantics

6.4.1 State Variables

Saved: Saved is a boolean that is true when the "Save" button has been pressed and false otherwise.

6.4.2 Environment Variables

keyInput: {key.AlphabetCharacters, key.LeftClick}

6.4.3 Assumptions

The Synesthesia Wear application is successfully installed on the user's device, ~~the user was able to log into the app,~~ and the sound configuration settings page is loaded onto the screen.

6.4.4 Access Routine Semantics

switchCue(key.LeftClick):

- transition: mouseLocation.navigate()
- output: None
- exception: Not_Cueable

keyword(key.AlphabetCharacters)char:

- transition: keyword.addCharacter()
- output: None
- exception: Not_A_Character

save(key.LeftClick):

- transition: keyword.save()
- output: Saved
- exception: Mouse_Not_On_Save_Button

7 MIS of Output Signal Module (M4)

7.1 Module

Output Signal Module (M4)

7.2 Uses

- Sound Classification Module (M5)

7.3 Syntax

7.3.1 Exported Constants

N/A

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
onVolt	(Boolean, Class_Id)		Null_class

7.4 Semantics

7.4.1 State Variables

current_Volt: Integer of the current voltage being supplied to the motor

7.4.2 Environment Variables

N/A

7.4.3 Assumptions

We are assuming that when this module is called that the battery has sufficient charge to power the motor for the called upon number of requests

7.4.4 Access Routine Semantics

onVolt(Boolean onOFF, Class_id Current):

- transition: Count = Current | While count ! = 0 $\implies$ (current_volt $\equiv$ 0 $\wedge$ onOFF) $\implies$ current_Volt = 5 | count $-=$ count | current_Volt = 0
- output: None
- exception: Class_id $\equiv$ Null

8 ~~MIS of Profile Module~~ Design Decision - Removed

8.1 Module

Profile Module

8.2 Uses

Login Module

8.3 Syntax

8.3.1 Exported Constants

N/A

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
Set_First_Name	String		Type_Error
Set_Last_Name	String		Type_Error
Get_First_Name		String	Null_Exception
Get_Last_Name		String	Null_Exception
change_Password	String		Type_Error
Change_Username	String		Type_Error
Get_Username		String	Null_Exception

8.4 Semantics

8.4.1 State Variables

- first_name
- last_name
- password
- username

8.4.2 Environment Variables

key_input(keyboard)

8.4.3 Assumptions

We are assuming that the user has either a touchscreen keyboard or an external keyboard.

8.4.4 Access Routine Semantics

Set_first_name(String Name):

- transition: first_name = Name
- output: None
- exception: Type_Error

Set_last_name(String Name):

- transition: last_name = Name
- output: None
- exception: Type_Error

Get_first_name():

- transition: None
- output: First_name
- exception: Null_Exception

Get_last_name():

- transition: None
- output: last_name
- exception: Null_Exception

Change_password(String Name):

- transition: password = Name
- output: None
- exception: Type_Error

Change_username(String Name):

- transition: username = Name
- output: None

- exception: Type_Error

Get_username():

- transition: None
- output: username
- exception: Null_Exception

9 ~~MIS of Battery Status~~ Out of Scope

9.1 Module

Battery Status

9.2 Uses

~~Indicator~~ N/A

9.3 Syntax

9.3.1 Exported Constants

N/A

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
Low Battery	N/A	Low Status	Low Battery
Battery Level	N/A	Percentage	Battery Status

9.4 Semantics

9.4.1 State Variables

N/A

9.4.2 Environment Variables

keyOutput : key.Low Battery, key.Battery Status

9.4.3 Assumptions

Battery will always have some charge on it.

9.4.4 Access Routine Semantics

- transition: N/A
- output: Battery level
- exception: N/A

9.4.5 Local Functions

- get level(int level)
- battery condition(int cond)

10 MIS of ~~Battery Status~~ Sound Classification (M5)

10.1 Module

Sound Classification (M5)

10.2 Uses

~~Categorize detected sounds~~

- Keyword Selection (M3)
- Bluetooth Communication (M6)
- Microphone (M7)

10.3 Syntax

10.3.1 Exported Constants

N/A

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
Sound level	Microphone	decibels	High/Low Level

10.4 Semantics

10.4.1 State Variables

N/A

10.4.2 Environment Variables

keyOutput : key.microphone $\mathbb{N}$
keyOutput : key.decibels $\mathbb{N}$

10.4.3 Assumptions

Surrounding sound levels are null compared to required input level.

10.4.4 Access Routine Semantics

- transition: N/A
- output: decibels
- exception: N/A

10.4.5 Local Functions

- sound level(float volume)
- sorting volume()

11 MIS of Bluetooth Communication Module (M6)

11.1 Module

~~BTComuModule~~ Bluetooth Communication (M6)

11.2 Uses

None

11.3 Syntax

11.3.1 Exported Constants

DataTx:{ BT.send()}

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
RecvNewClass	DataRx	-	Class_Full
RmClass	DataRx	-	Class_Empty
UpdatePref	DataRx	-	-
SendBattV	BattV	DataTx	-

11.4 Semantics

11.4.1 State Variables

None

11.4.2 Environment Variables

~~BattV: {AnalogRead(BatteryVolt)}~~
DataRx: {BT.recieve(T1),BT.recieve(T2),BT.recieve(T3)}ℕ

11.4.3 Assumptions

Connection with the application is already establised. Bluetooth tries to automatically reconnect if the application is disconnected.

11.4.4 Access Routine Semantics

RecvNewClass():

- transition: if(DataRx == T1) then addclass(dataRx)

- output: None
- exception: Class_Full

RmClass():

- transition: if(DataRx == T2) then rmclass(dataRx)
- output: None
- exception: Class_Empty

UpdatePref():

- transition: if(DataRx == T3) then prefchange(DataRx.class,DataRx.pref)
- output: None
- exception: None

~~SendBattV(BattV)~~:

- transition: None
- output: DataTx == BattV
- exception:

12 MIS of Microphone Module (M7)

13 MIS of Profile Module

13.1 Module

~~MicroMod~~ Microphone (M7)
Bluetooth Communication (M6)

13.2 Uses

None

13.3 Syntax

13.3.1 Exported Constants

None

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
updateData	SoundIN,CLKINT	SoundData	-

13.4 Semantics

13.4.1 State Variables

SoundData - Array of sound inputs recorded at every sample interval

13.4.2 Environment Variables

SoundIN: { Digital.Read(Digital_Microphone) }$\mathbb{R}$
CLKINT: {clk.interupt(sample time)}

13.4.3 Assumptions

Clock interrupt is already set up to issue interrupts every 1/(sample frequency)

13.4.4 Access Routine Semantics

- transition: N/A
- output: decibels

- exception: N/A

References

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. Fundamentals of Software Engineering. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. Software Design, Automated Testing, and Maintenance: A Practical Approach. International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.

14 Appendix