V-MAP

A wireless multi-device voice commander

Appendix D: Architectural Design
P05D03.V3

BMET 4402

Kyuhee Lee, Tim Kong, Weston Park

April 1, 2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The scope of this document is to provide the information which comprises the architectural design of the *V-Map* device, which includes the software, electrical, and mechanical designs, as well as any associating architectural information.

1.1 Intended use

The *V-Map* is intended to be used by a quadriplegic client who has a clear speech as a wireless voice activated remote controller. This voice commander allows the client to generate various environmental control functions.

2. CONCEPT

The *V-Map* is the front end of a system of modules that allows a client to trigger environmental functions. The *V-Map* provides the voice recognition function and transmits control signals to various receiver modules that carry out some other function. The voice recognition module picks up and learns the voice commands which are then processed into radio frequency signals and sent to the transmitter. The transmitter communicates with the receiver on the custom device's end which carries out the desired function. Thus, the *V-Map* allows the clients to activate the custom device by voice commands.

One basic example of such a device is a receiver that controls a solid-state relay to produce a simple switch closure signal. The client plans to use it to trigger an automatic door-opening device.

2.1 Command interpreter

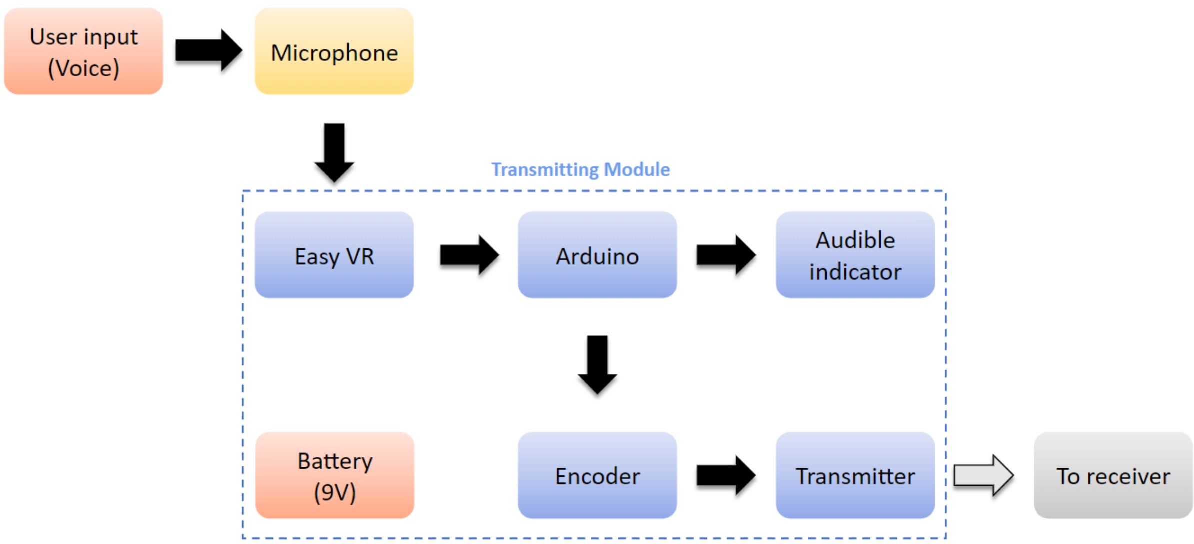

Figure 1: Block diagram of command interpreter

The block diagram above illustrates how our device is implemented with connections from component to component, working together to construct an efficiently operating unit. The command interpreter module contains three major parts: a voice recognition part, a processing part and a signal transmission part.

2.1.1 Voice recognition module

The voice recognition module consists of a microphone and EasyVR. The microphone picks up the voice commands and sends the commands to the EasyVR which can be used with any host with a UART interface powered at 3.3V – 5V, such as PIC and Arduino boards.

2.1.2 Processing module

The processing module contains Arduino which decodes the messages from the EasyVR and determines the address and data to be delivered to the encoder, which then further transmits the signals to the corresponding target receiver.

2.1.3 Signal transmission module

The signal transmission module consists of an encoder and a transmitter. The address determined in the Arduino will be converted into serial data in order to transmit over RF link using the transmitter. The transmitter transmits this serial data to the receiving part using radio signals.

2.2 Receiver switch

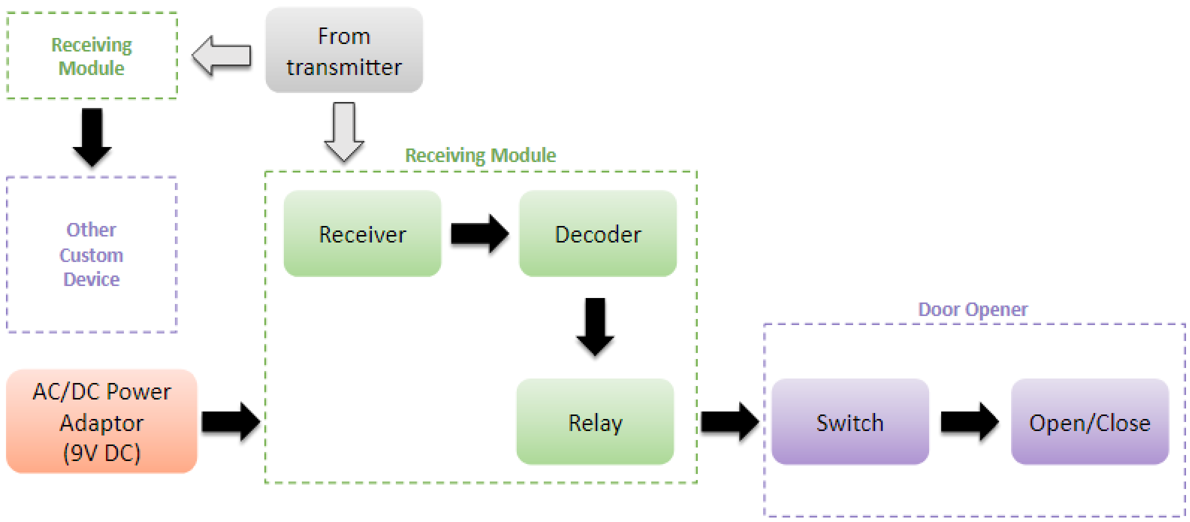

Figure 2: Block diagram of receiver switch

The receiver switch is a generic adaptive switch that responds to messages from the voice command interpreter. By matching the transmitter's address to that of the receiver's, the connection is successfully done and they are ready to communicate with each other. Then the transmitter can communicate with different receivers regardless of the functions of each receiver. In this project, the device to be activated requires its own receiver switch which contains a receiver and a decoder. The receiver picks up the signal transmitted from the command interpreter. If the address of the transmitted signal matches that of the decoder, the decoder will decode the signal. The switch closure relay is specifically used in this project, which activates the door opener to turn on or off.

3. SOFTWARE DESIGN

3.1 Training a command

EasyVR Commander software can be used to train commands and update sound tables. To start, go to "File" and click "Connect". *Figure 3* shows the main application window of the software.

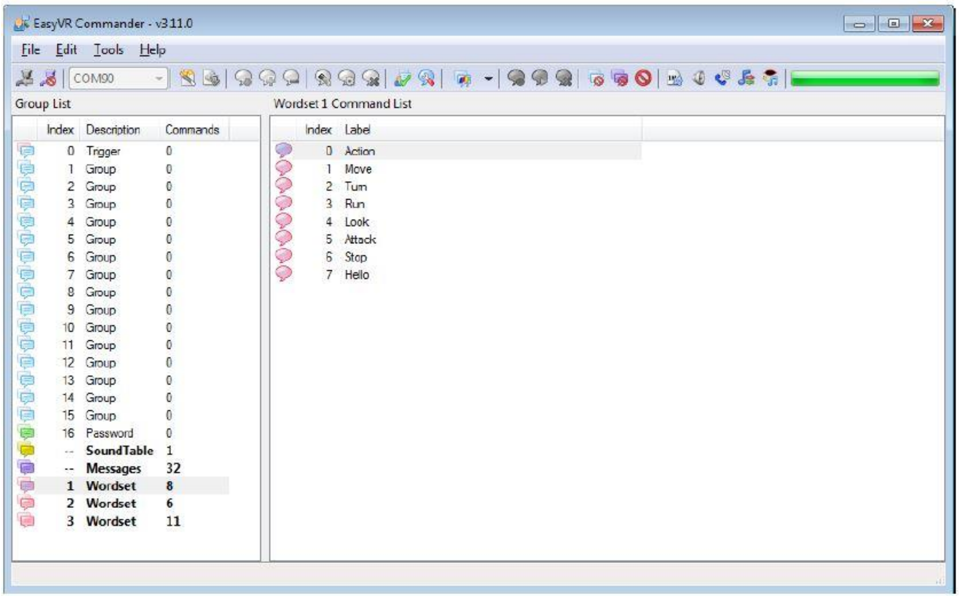

Figure 3: Main application window

Then, a new command can be added by first selecting the group in which the command needs to be created and then using the toolbar icons or the "Edit" menu. A command should be given a label and then it should be trained twice with the user's voice: the user will be guided throughout this process (*Figure 4*) when the "Train Command" action is invoked.

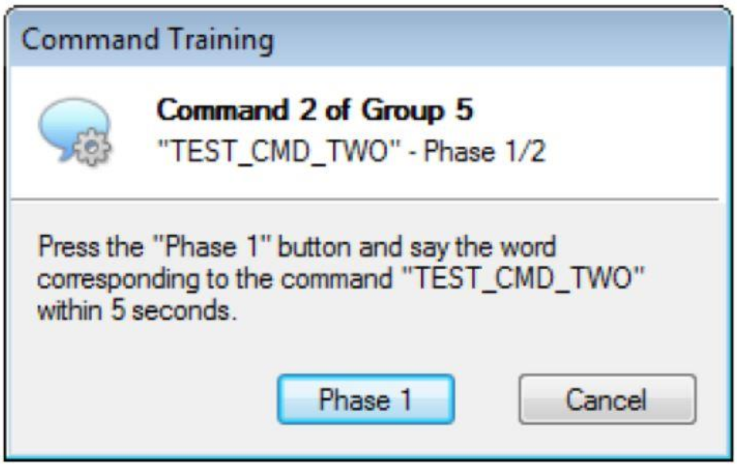

Figure 4: Guided training dialog

After clicking on "Phase 1" or "Phase 2" buttons, start speaking when a little window say "Speak Now" appears. The little window is shown in *Figure 5*.

Figure 5: Speak now window

If any error happens, command training will be cancelled. Errors may happen when the user's voice is not heard correctly, there is too much background noise or when the second word heard is too different from the first one.

After all, all the commands are trained and stored in the EasyVR module.

3.2 Trained commands and corresponding bits

In the EasyVR module, the trained commands are categorized in groups, and each command is indexed within the group. *Table 1* shows the list of all recognized commands and their corresponding address and data bits.

Table 1: List of trained commands and their corresponding bits

Group	Index	Address bit	Data bit
Group 1	0: Blind open	101	001
	1: Blind close	101	010
	2: Open door	011	001
	3: Blind stop	101	000
Group 2	0: Yes	000	000
	1: No	000	000

3.3 Sound table

A mixture of sounds and feedback messages are created to allow the user to capture the timing to speak a command and which group of commands to speak during operation.

Table 2: List of sounds and feedback messages stored in EasyVR

Sound	**Description**
0	Beep sound: A. play once = listening to a command B. play twice = trained command is recognized
1	Feedback: “Blind’s opening”
2	Feedback: “Blind’s closing”
3	Feedback: “Door’s opening”
4	Feedback: “Entering sleeping mode”
5	Feedback: “Error”
6	Feedback: “Time out”
7	Feedback: “Power on”
8	Feedback: “Do you want to perform another task?”
9	Feedback: “Low battery, please get a new replacement”
10	Feedback: “Battery percentage good”
11	Feedback: “Say a command in group 1”
12	Feedback: “Please stop the blind after the beep”

3.4 Software flow chart

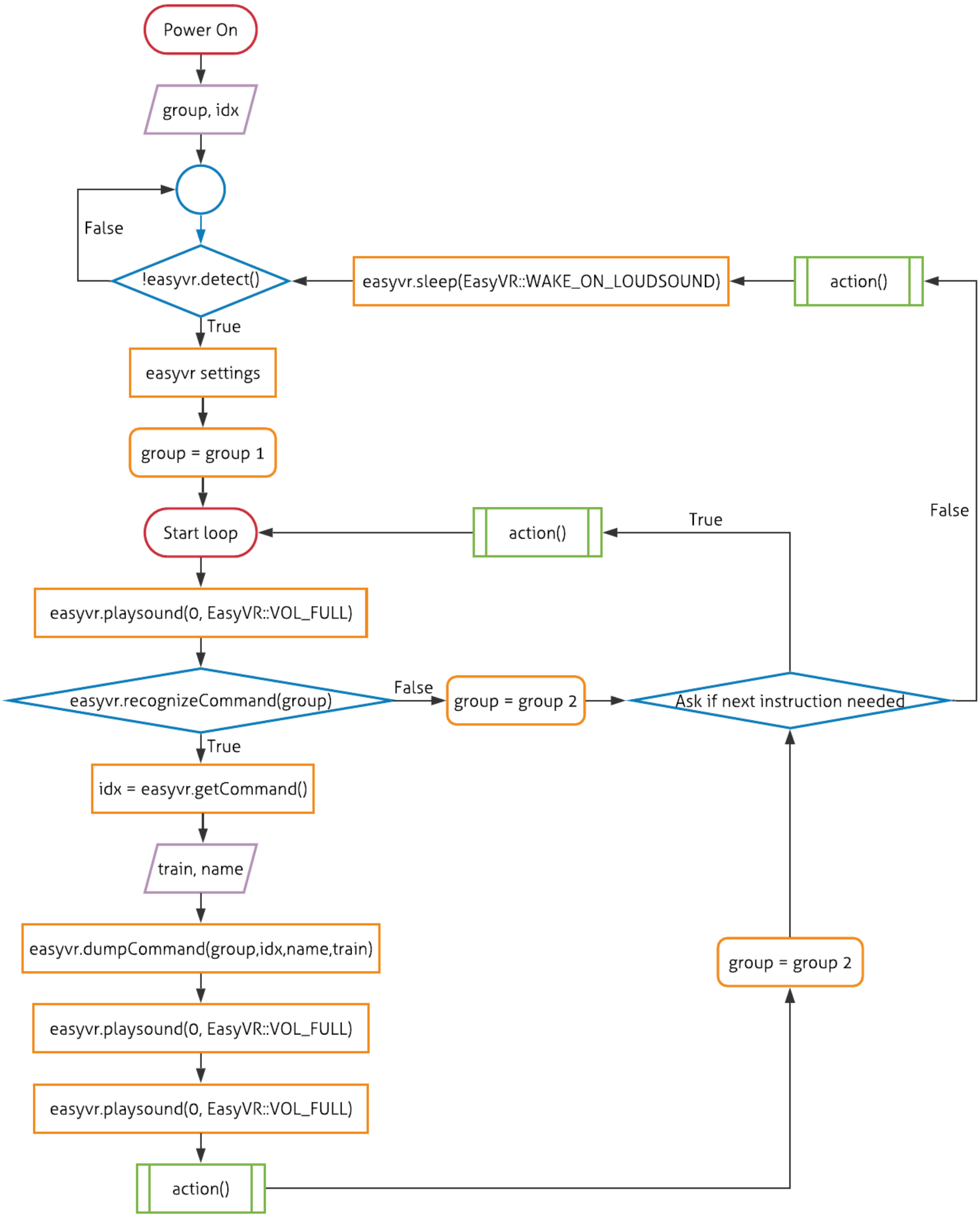

Figure 6: Software flow chart of the main loop

Figure 6 shows the flow of the recognition procedure of the EasyVR. The EasyVR has handy library functions which help the user in designing the software code. Some of the library functions used in this project are:

- *easyvr.detect*
- *easyvr.playsound*
- *easyvr.recognizeCommand*
- *easyvr.hasFinished*

- *easyvr.getCommand*
- *easyvr.dumpCommand*
- *easyvr.sleep*

In the EasyVR settings, the time for the EasyVR to listen to a command is set using a function called *easy.setTimeout*. The *easyvr.detect* senses incoming instructions for waking EasyVR from sleep mode. For example, the *easyvr.sleep (EasyVR::WAKE_ON_LOUDSOUND)* puts EasyVR into sleep, and only if the *easyvr.detect* senses a loud sound, typically more than 90 dB, the EasyVR will wake up from sleep mode. Once powered, the EasyVR is automatically awake until the *easyvr.sleep* puts it into sleep.

In the EasyVR module, the trained commands are categorized in groups, and each command is indexed within the group. The EasyVR recognizes a command one group at a time, and this is done by the *easyvr.recognizeCommand*. When the EasyVR is ready to listen to a command, a beep sound instructs the user to say a trained command in Group 1 to begin with. One example of a Group 1 command will be “blind open”. A list of the trained commands is recorded in *Table 1*. The beep sound is displayed by the function called *easyvr.playsound*, and the list of all the sounds stored in the EasyVR is recorded in *Table 2*. Once a command is being recognized, user can retrieve the group and index of a recognized command from the serial monitor of Arduino IDE. The beep sound will be played twice to indicate that a command is being recognized.

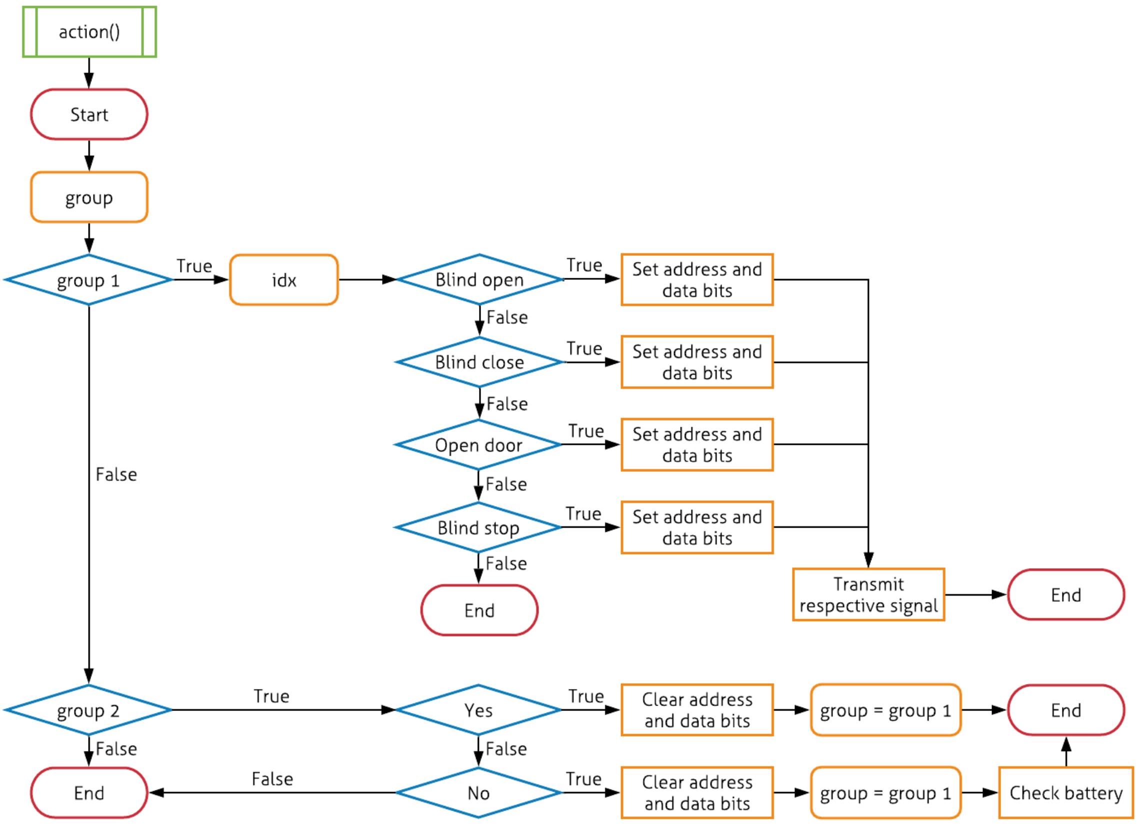

Figure 7: Software flowchart of action function

For transmission of RF signals, the action function will then be called to set the address and data bits which correspond to the recognized command accordingly. The list of all bits can also be found in *Table 1* and the flow of the function is shown in *Figure 7*. When the action in Group 1 is taken, group will be set to Group 2, where the EasyVR will then ask the user if next instruction is needed. In this case, “yes / no” commands will be listened, all bits will be cleared when either of them is being recognized.

The “yes / no” are categorized in Group 2. At the meantime, if any error occurs during recognition of Group 1, for example recognition failure or recognition time out, the group will also be set to Group 2, the bits will be also be cleared, and the EasyVR will ask the user if further instructions are needed. If the answer is “yes”, the program will loop back and start a new recognition in Group 1. If “no” is obtained, the EasyVR will check the battery percentage and tell the user if a new battery replacement is needed before going into sleep.

3.4.1 Input pins

The EasyVR module is stacked on top of the Arduino. Pin 13 and pin 12 were used for serial communication between the EasyVR and Arduino.

- Pin 13: Serial transmitter pin for EasyVR module, and Serial receiver pin for Arduino
- Pin 12 : Serial receiver pin for EasyVR module, and Serial transmitter pin for Arduino

3.4.2 Output pins

Pin 2 to pin 7 are the address and data bits which will be to be sent to the corresponding receivers.

- Pin 2: Address Bit 1
- Pin 3: Address Bit 2
- Pin4: Address Bit 3
- Pin5: Data Bit 1
- Pin6: Data Bit 2
- Pin 7: Data Bit 3

4. ELECTRICAL DESIGN

4.1 Functional section

The key commercial modules used in the *V-Map* are listed below:

Table 3: List of key commercial modules used in the V-Map

	Command Interpreter	**Receiver Switch**
Voice recognition shield	EasyVR 3	N/A
Microprocessor board	Arduino Duemilanove	N/A
Transmitter and receiver pair	RF link transmitter 434 MHz	RF link receiver 434 MHz
Encoder and decoder pair	HT12E encoder	HT12D decoder
Relay	N/A	JWD series reed relay
Battery	3 AA lithium batteries	9V AC/DC power adaptor

The user interface components used in the *V-Map* are:

- Microphone
- Speaker
- LEDs

The custom interfaces used in the *V-Map* are:

- Command interpreter board
- Receiver switch board

The *V-Map* will require a prototyping board for each module, and the command interpreter and receiver switch will be implemented as compact as possible.

4.2 Power section

Each module has different power requirements according to the components that are used. Therefore, the proper way of power production has to be taken into account.

4.2.1 Command interpreter

Due to the fact that the voltage source provides enough voltage to operate the 4 following major components: EasyVR, Arduino, encoder, and transmitter, the command interpreter needs high power source. However, because this module has to be movable, using battery is inevitable; as a result, we chose to use 3 3V AA lithium batteries in series to make 9V source.

It is better to use this kind of battery than 9V battery, based on the reason of it having better amp hours (1200mAh) compared to the 9 V battery (500mAh).

Each component's power requirements of the command interpreter are shown in *Table 4* provided below:

Table 4: Power Requirements for the command interpreter components and IC

	Parameter	Min	Typical	Max	Unit
EasyVR 3	**V_{SEL} (Input Voltage)**	3.15	5	5.5	V
Arduino Duemilanove (ATmega328)	**V_{DD} (operating voltage)**	—	5	—	V
	V_I (Input voltage)	6	7-12	20	V
Encoder (HT12E)	**V_{DD} (operating voltage)**	2.4	5	12	V
	I_{STB} (standby current)	—	0.1	1	µA
	I_{DD} (operating current)	—	40	80	µA
	V_{IH} (input High voltage)	$0.8V_{DD}$	—	V_{DD}	V
	V_{IL} (input Low voltage)	0	—	$0.2V_{DD}$	V
	I_{sink} (V_{out} = High)	-1	-1.6	—	mA
	I_{source} (V_{out} = Low)	1	1.6	—	mA
Transmitter (TWS-BS)	**V_{DD} (operating voltage)**	3	5	12	V
	I_{DD} (operating current)	—	—	8	mA

4.2.2 Receiver switch

The main components of this module include a decoder, receiver and relay. Since the receiver switch does not need to be portable like the command interpreter, a 9V AC/DC power adaptor with current of 500mA will be used to power this module.

Each component's power requirements of the receiver switch are shown in *Table 5*Table 5 provided below:

Table 5: Power Requirements for the receiver switch components and IC

	Parameter	Min	Typical	Max	Unit
Decoder (HT12D)	**V_{DD} (operating voltage)**	2.4	5	12	V
	I_{STB} (standby current)	—	0.1 (for V_{DD} = 5V) 2 (for V_{DD}=12V)	1 (for V_{DD} = 5V) 4 (for V_{DD}=12V)	μA
	I_{DD} (operating current)	—	200	400	μA
	V_{IH} (input High voltage)	3.5	—	5	V
	V_{IL} (input Low voltage)	0	—	1	V
	I_{sink} (V_{out} = High)	-1	-1.6	—	mA
	I_{source} (V_{out} = Low)	1	1.6	—	mA
Receiver (RWS-371)	**V_{DD} (operating voltage)**	4.9	5	5.1	V
	V_{OH} (output High voltage)	V_{DD} – 0.5	V_{DD}	V_{DD}	V
	V_{OL} (output Low voltage)	—	—	0.3	V
Relay (JWD-107-5)	**V_{DD} (operating voltage)**	—	3.8	—	V
	Coil voltage	—	5/6	—	V

5. MECHANICAL DESIGN

This section is to support the design of the enclosures for each module.
The following is a list of physical components, both commercial and custom, that are used for the hardware.

Table 6: List of physical components available commercially and custom made

Component			Commercially Available	Custom Built
Internal	**Command Interpreter**	Encoder	HT-12E	—
		Transmitter	RF Link Transmitter – 434 MHz (TWS-BS)	—
		Microprocessor	Arduino Duemilanove	—
		Prototyping Board	—	Cut and drilled to fit
		Voice Recognition Module	EasyVR 3	—
	Receiver Switch	Decoder	HT-12D	—
		Receiver	RF Link Receiver – 434 MHz (RWS-371)	—
		Relay	JWD-107-5	
		Prototyping Board	—	Cut and drilled to fit
		Stereo Jack Female Connector	3.5 mm Audio Jack STX-3120-5B	—
External	**Command Interpreter**	Chassis	—	3D printing
		Speaker	8 ohm speaker	—
		Microphone	3.5mm Microphone External Assembly For Car Vehicle	—

		Battery Holder	Enclosed AA*3 with switch	—
	Receiver Switch	Chassis	Commercial enclosure	Cut and drilled for inserts
		Power adaptor	9V AC/DC Power Adaptor (500mA)	—
screws		2- 56 x 1/2”	bulk	—
cable bundles		2x (Red, Orange) speaker wire	Stranded Wire	Cut, use heat shrink tubing, and soldered to fit
		2x (Red, Black) circuitry power	Stranded Wire	Cut, use heat shrink tubing, and soldered to fit
		2x(Green, Yellow) microphone connecter	Solid Wire	Cut, heat shrunk, and soldered to fit
		2x (Red, Green) LED indicator	Stranded Wire	Cut and connector attached to fit

5.1 Enclosures

The chassis for the *V-Map* are made of plastic printed by 3D modelling program for the command interpreter and purchased from a commercial seller for receiver switch. The size was determined primarily by the dimensions of the internal circuit boards and by the overall aim to make the device compact and small enough that can be fit on any other applied parts such as transmitting parts on wheelchair and receiving parts closed to the custom door opener.

5.1.1 Command interpreter

The mechanical design for interaction between the patient and the *V-Map* is achieved by placing the speaker and microphone close to the patient in order to interact clearly. Transmitting circuit board, EasyVR and Arduino establish the 3-layer board. Putting headers on the right position on the transmitting circuit board makes it to be stacked on top of the EasyVR that has been stacked on top of the Arduino. *Figure 8* shows the layout of the 3 layer-board. As the headers are positioned on one side, 3D printed pillar supports the other side of the transmitting circuit board.

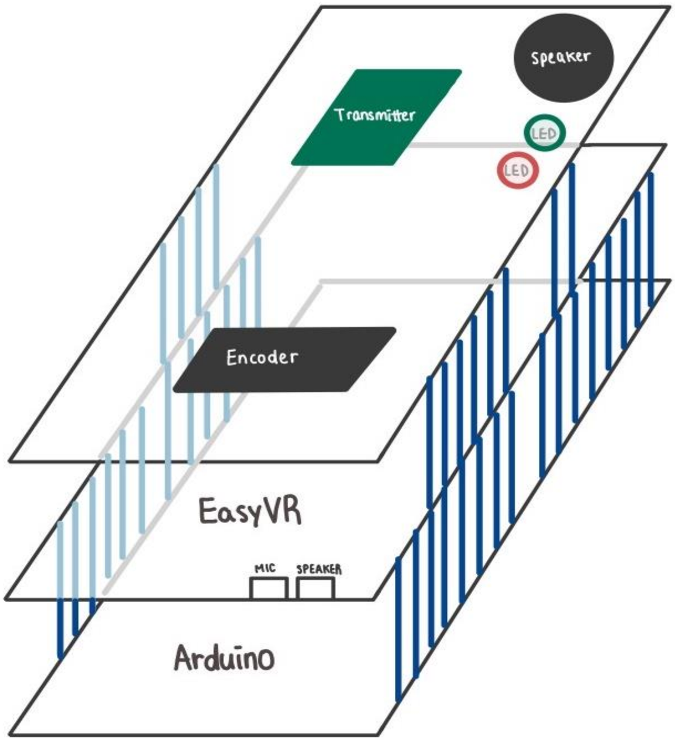

Figure 8: Layout of the circuit assembly

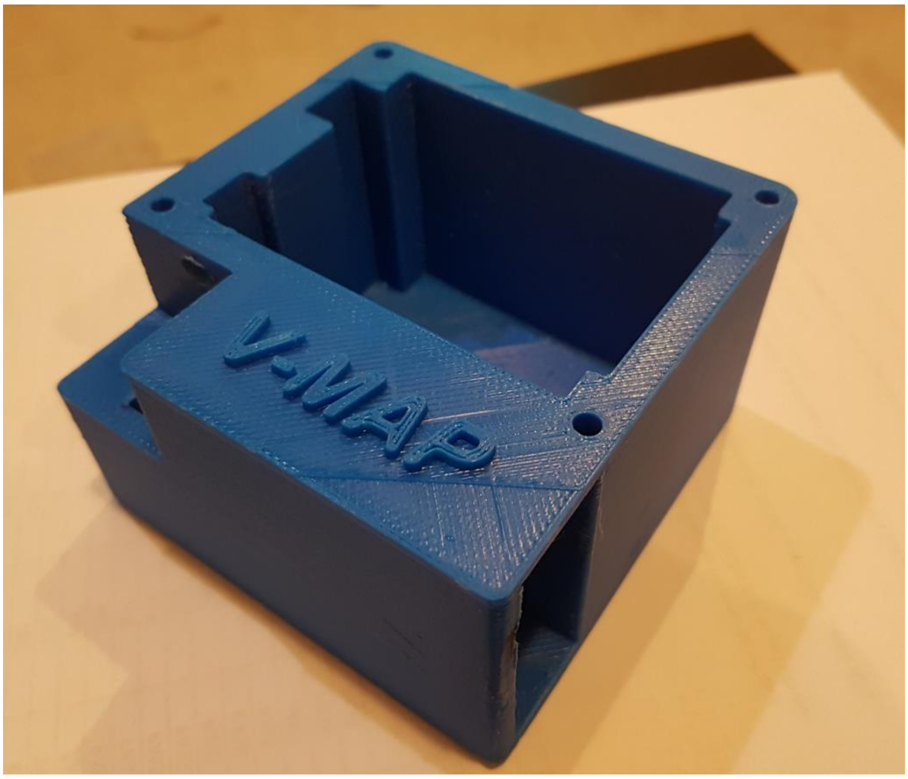

Figure 9: External design of the command interpreter

The main chassis consists of two major parts, one being the main holder which the 3 layer-board is put into, and the second being the battery closet that holds the 3 AA battery holder which can be slide in. The battery closet is placed right next to the main holder, so there is a better connection to the power pins of the circuit board from the battery. By making the opposite site of the entrance of the closet as the exposed part, where the on/off button of the holder is, users can turn on and off the battery without effort, and easily pull out the battery holder when the battery needs to be replaced by pushing the exposed part. The external design of the command interpreter is shown in *Figure 9*.

The cap of the enclosure covers the component and prevents them from exposure; however, there are holes for microphone, speaker, and LEDs to be able to achieve user interface.

5.1.2 Receiver switch

The components of the receiver switch are simply implemented together on a prototyping board, and the 3.5mm audio jack for sending the switch closure signal to the door opener is placed on the edge of the board. The power jack is installed through the hole on the side of the chassis to receive the power from the 9V adaptor. The chassis for this module is selected to use the commercial enclosure made of plastic. We will be modifying the chassis with the use of power tools and hand to create holes by drilling for the power jack and the audio jack. The prototyping board will be secured with the board standoffs. The figures below show the different layout views of the receiving housing.

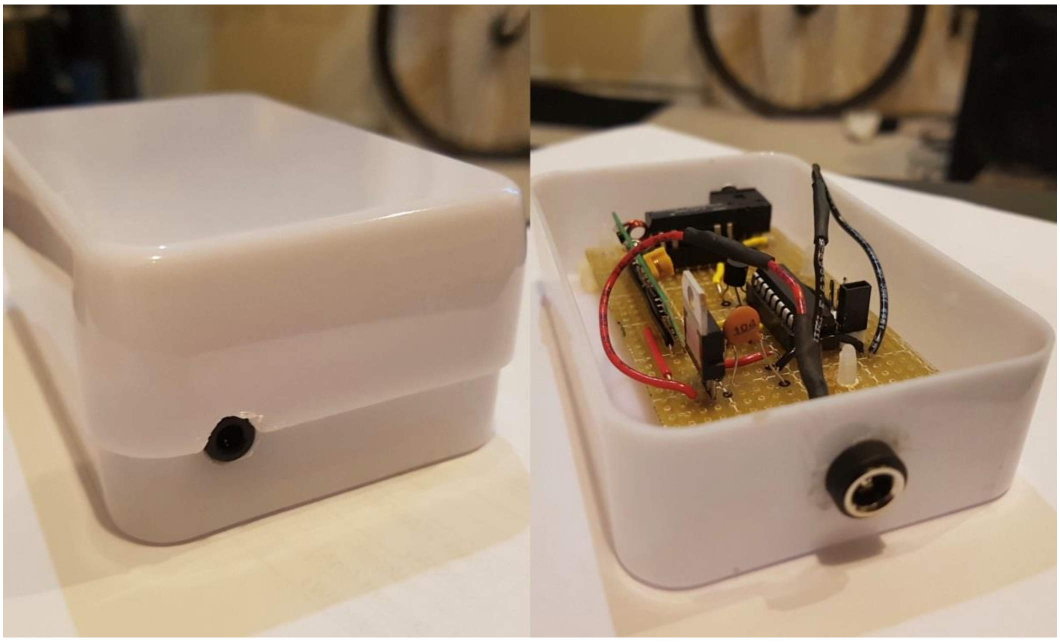

Figure 10: The side views of the receiver switch: audio jack hole (left), power jack hole (right)

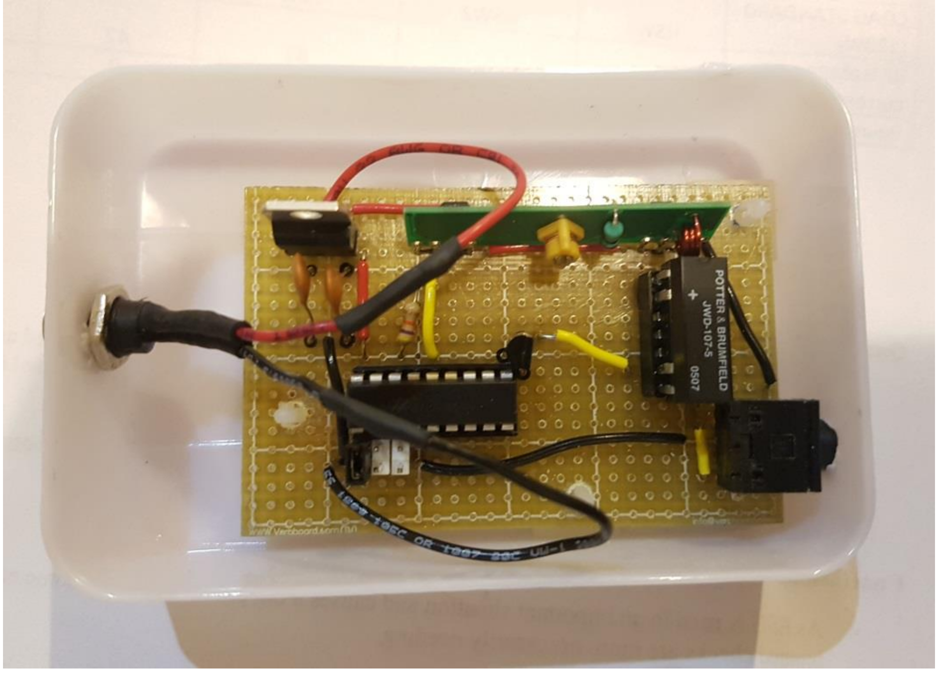

Figure 11: The internal layout of the receiver switch

5.2 Labelling

The *V-Map* will include external labelling.
External labels include:

- User interface labels
    - Power On/Off
    - Microphone Port
    - Audible Indicator
    - Visible Indicator
    - Battery