MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Vietnamese Sign Language Recognition

Group 05	
Group members	Nguyễn Hữu Kỳ Long – Team leader – SE60984 Nguyễn Đình Tân – Team member – SE61115 Nguyễn Xuân Ý – Team member – SE60869 Lê Phương Bình – Team member – SE61049
Supervisor	Mr. Đỗ Đức Minh Quân Mr. Nguyễn Đức Lợi
Ext. Supervisor	N/A
Capstone Project code	VSLR

-Ho Chi Minh City, 12/05/2015-

This page is intentionally left blank

Table of Contents

List of Tables

List of Figure

Definitions, Acronyms, and Abbreviations

Name	Definition
VSLR	Vietnamese Sign Language Recognition
LCD	Liquid crystal display
SVM	Support vector machine
GUI	Graphic User Interface
LED	Light Emitting Diode
AC	Alternating Current
DC	Direct Current
GPIO	General-purpose input/output
HDMI	High-Definition Multimedia Interface
USB	Universal Serial Bus

A. Report No. 1 Introduction

1. Project Information

- Project name: **Vietnamese Sign Language Recognition**
- Project Code: **VSLR**
- Product Type: **Embedded system**
- Start Date: **May 12th, 2015**
- End Date: **August 16th, 2015**

2. Introduction

Nowadays, the communication is the way people can understand each other, is the way people can express their ideas, their thoughts to others. As we know, speaking is the most common way to communicate in life. However, to dumb person, they still need to communicate with others so they have a different way to expose themselves, it is called hand sign language or dumb language.

In this project, we want to develop a device that can help dumb person communicate with not only another mute but also everyone. The device can capture hand signs and then recognize them into text or sound with the same meaning.

3. Current Situation

When you want to talk to a dumb person or when a mute wants to present his / her ideas, presentations in a meeting but you are not able to get their signs. Furthermore, when two dumb persons talk to each other but they are from different countries, they have distinct hand sign language, which way can they understand each other? Obviously, there are some ways, they can write out what they want or they can use some signs that are familiar to the daily life, and they can even hire a translator to interpret.

4. Problem Definition

The following disadvantages of current situation*:*

- Handwritten: Time consuming to write out all content is very high.
- Using familiar signs: Without time consuming, the accuracy of the content is not high.
- Hand sign language translator cannot respond the instant needs of communication. Moreover, the price for hiring a translator is very costly.

5. Proposed Solution

To meet the needs of users we offer a solution based on translating hand signs into content and then show them.

Our system is a small device with a camera to capture hand signs and then translate them.

In more detail, our system has the following functions:

5.1 Feature functions

- The system detects your hands, and then recognizes it and outputs recognition result with the same meaning.
- Showing the translated content for users on text and sound.
- Learning sign language hand for people who want to know about the language in order to better communicate with dumb people.

5.2 Advantages and disadvantages

The advantages and disadvantages of the proposed solution*:*

- Advantages:
  - Quick and easy communicate for dumb person.
  - Train for person who don't know about mute language.
  - Standardized for hand sign language.
  - People get used to the dumb language easily.
- Disadvantages:
  - In some cases, this solution does not work exactly with the hands have weird characterize.
  - This solution needs stable environment (light, background) and some accessories.
  - This solution cannot solve the problem about hand motion language.

6. Functional Requirements

Function requirements of the system are listed as below:

6.1 Tracking hand

Allow users can move the hand inside the area camera working but the system still works correctly.

6.2 Hand recognition

The system analyzes the images, which is captured by camera, then detects and recognizes the hand sign on these images into content.

6.3 Showing the content

The translated content is shown not only on text but also on sound.

6.4 Learning hand sign

Users select and learn words existed in the system with images express the hand gesture.

6.5 Controlling System

- Allow users can turn on / off the system by the power button.
- Users can select functions by hand signs.
- Users can perform operations of function by hand signs.

6.6 Controlling power

- System uses battery power gives users more flexibility in using.
- Combining with controlling the battery capacity that helps users to use the most effective

7. Role and Responsibility

No	Full Name	Role	Position	Contact
1	Đỗ Đức Minh Quân	Scrum Master/Product Owner	Instructor	minhquandd@fpt.edu.vn
2	Nguyễn Đức Lợi	Scrum Master/Product Owner	Instructor	loind@fpt.edu.vn
3	Nguyễn Hữu Kỳ Long	Developer	Leader	longnhkse60984@fpt.edu.vn
4	Nguyễn Đình Tân	Developer	Member	tanndse61115@fpt.edu.vn
5	Nguyễn Xuân Ý	Developer	Member	ynxse60896@fpt.edu.vn
6	Lê Phương Bình	Developer	Member	binhlpse61049@fpt.edu.vn

Table 1: Roles and Responsibilities

B. Report No.2 Software Project Management Plan

1. Problem Definition

1.1 Name of this Capstone Project

Vietnamese Sign Language Recognition

1.2 Problem Abstract

As we know, in the daily life, there is a lot of ways people can understand others such as speech, expression of act, gesture or feelings, etc. However, it is better to express oneself in speech. At the same time, it is an actual matter to mute people to get other people and in the opposite way. The current solution for them is sign language but that means it requires everyone to know sign language of mute people or need someone play as a translator. Nevertheless, these solutions just solve the problem at that time and these are not a long-term strategy. It expects a long time and high cost for preparation from them to solve the problem. In additional, there still are some temporary solutions such as handwriting or using familiar signs, but these ways will not produce the desired effect and requires lots of time or effort.

To solve those problems mentioned above, we propose a solution, which can help dump person to express him or herself in speech or text. That is a device playing a translator and act as intermediary role.

1.3 Project Overview

1.3.1 Current Situation and Disadvantages

Below are some current behaviors of user:

- Handwriting:
  - People will use something can write on as vehicle for communication.

    - They can write out exactly what they want to say to the recipient.
    - The recipient can receive and read the content immediately.
- Familiar signs:
    - Speakers will describe the word which they want say through action; describe the shape, body language.
    - Listeners observe the speaker's actions. They predict information that the speaker shown.
- Interpreters:
    - Act as intermediary to translate the content of communication.
    - Speakers express words by their language, the interpreter receive information from the speaker and then convey that information by the language of the listener.
    - Degree of accuracy of translated content is quite high for both two sides.

Below are the disadvantages of current situation:

- Hand-writing :
    - Users must use an intermediary for communication such as paper, pens. However, these things are not always available.
    - Users spend more time to write out all their wishes and read them.
    - User can meet difficulties about different languages.
    - The error can be caused by user handwriting.
- Using familiar signs :
    - Maybe be misleading because the symbols are not standardized.
    - It is trending towards personally identifiable user.
    - It is difficult to show all wishes of communicator.
    - Time consuming for understanding the content is long.
- Translator :
    - Hiring a translator must be costly.
    - Translator who work only in the fixed time, thus not always can meet user's demands.
    - Translator must be an experienced person.
    - Number of translator is limited.

Analyzing image is the most common way to solve many problems in the real life. One of those problems is recognition. Today, with growth of supported analyzing image library and algorithms provided to process image is widespread, tracking and recognition can be performed more easily. Our project is taking into consideration about it to recognize hand signs to help people can communicate with another people.

- Advantages:
  - The system can be implemented on many different platforms.
  - Operating costs are less expensive.
  - Recognition is implemented quickly by many image-processing algorithms.
- Disadvantages:
  - Analyzing image is still remains restriction on process environment, point of view.
  - Recognition has still not covered every case yet. Within weird characterizes, the result maybe not high accurate.
  - Currently, analyzing image and recognition just detect and recognize hand signs without motion.
  - To get high degree of accuracy, it requires some accessories from users.

1.3.2 The Proposed System

Exploiting the development of embedded technology and the growing of image processing, we put forward a system which can recognize hand sign language to help dumb people can communicate. This system includes a camera, which captures hand signs from user, a raspberry board plays role as central processing unit that analyzes these captures, processes some algorithms to recognize them and performs some different functions in the system, and a LCD that shows interfaces of the system and recognition result. Besides that, the system still provides some electronic devices to user can control battery, or devices.

1.3.2.1 Controlling System

- Users can turn on/off the system by a switch button.
- Users use hand gestures to select the functions and move between functions.

1.3.2.2 Portable System

- Users can monitor the battery capacity.
- Users can charge battery.

1.3.2.3 Hand Sign Language Recognize

- Users express hand gestures, which describe the desired content, and then they can receive the hand sign recognition result.
- Users can see your hand gestures on LCD.
- Users can check the result of the current hand sign.
- Users can edit the current translated content.
- Users receive the recognition result via text or sound shown from LCD.

1.3.2.4 Learning Hand Sign

- Users can choose words that they want to learn which existed in the system.
- Users can see images, which expresses the hand gesture.
- User's hand signs can be practiced and checked by following some steps of the system.
- Users receive the current recognized result of the hand sign via text or sound.

1.3.3 Boundaries of the System

1.3.3.1 The restrictions

- The system language is Vietnamese.
- Hand sign language the system supports is Vietnamese sign language.
- The system just recognizes no motion hand signs.
- The system requires users must use supported accessories.
- The system requires users must provide a stable environment in room with sufficient light and a background is not complex on color, especially, no color close to skin color. If the background is not suitable, user must use the background which is provided by the system
- User must wear the accessory, which is provided by the system such as black bracelet.
- The system must be fixed during the working process.

1.3.3.2 The components of the system:

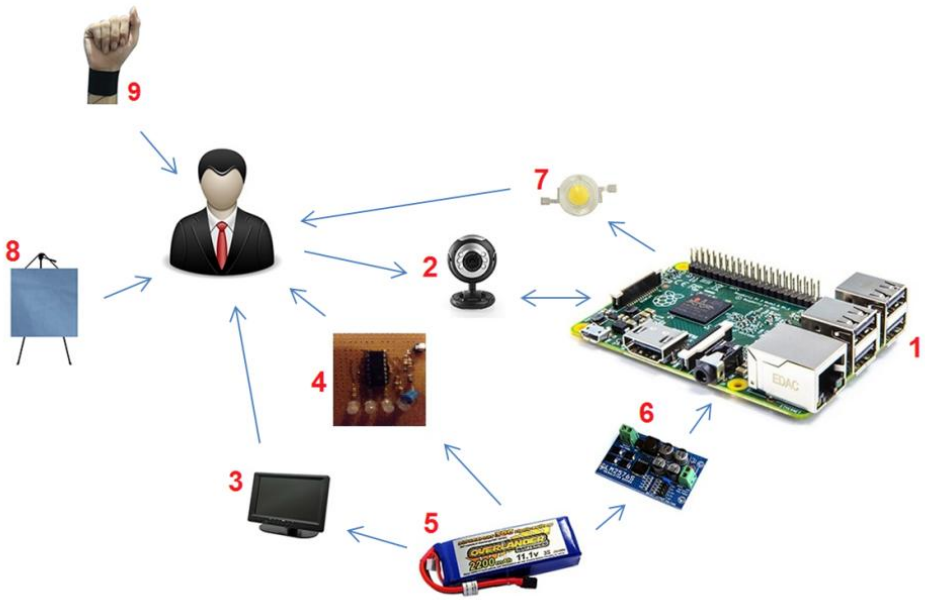


Figure 1: Components of the system

No.	Component's Name
1	Raspberry
2	Camera
3	LCD
4	Battery Capacity Display Circuit
5	Lipo Battery
6	LM2576ADJ - 3A UNI REG Board
7	LED
8	Accessory: fixed background
9	Accessory: black bracelet

Table 2: Component's Name

1.3.4 Development Environment

1.3.4.1 Hardware requirements

- Four laptops are used for development the system. These are setup Ubuntu 14.04 operating system.
- Raspberry Pi B2 is used to process as central processing unit.
- Micro SD card 16GB: is used to setup Raspbian operating system
- Cable HDMI to HDMI: connect Raspberry to LCD
- Cable is connection between laptop and raspberry pi B2.
- Keyboard, mouse, and usb wifi are used to setup operating system and necessary environments for raspberry pi B2.
- Backup flash memory: a backup solution when problems with operating system. This memory must be setup similar to main flash memory.
- LIPO 3 cells battery (12V – 2700mA): power for the system can works.
- Logitech C270 Webcam: is used to capture images.
- 7 inch HD TFT Color Monitor LCD: is used to show the interface of functions and the recognized results.
- 1 Led (1W 350mA): is used to balance light.
- LM2576ADJ-Board - UNI Regulator Board: conversion from higher to lower DC voltages
- TL084 + zener 5.1v are used to monitor battery capacity.
- Accessories: black bracelet and fixed background.

1.3.4.2 Software requirements

- Operating system and platform for deployment and development: Ubuntu 14.04 for laptop and Raspbian for Raspberry PI B2.
- Remmina Remote Desktop Client: application for remoting to work on raspberry.
- QT Creator version 4: is to develop C++ application and Linux GUI.
- OpenCV 2.4.9 library: support image processing.

- LIBSVM library version 3.20: support SVM for recognition.
- BCM2835 1.45 library: allowing access to the GPIO pins on the 40 pin IDE plug on the RPi board
- Espeak 1.48.04 library: open source software speech synthesizer for English and other languages, for Linux and Windows
- SQLite 3: software creates and manages the system database.
- Software Ideas Modeler: application for creating models and diagrams.
- Microsoft Office 2010: is used to write documents and assign tasks.
- Githup and TortoiseSVN and Rabbit VCS: used for source control.
- Skype: used for communication and meeting.

2. Project organization

2.1 Software Process Model

2.1.1 Overall Description

Scrum is an agile methodology that can be applied to nearly any project; however, the Scrum methodology is most commonly used in software development. The Scrum process is suited for projects with rapidly changing or emergent requirements. Scrum software development progresses via a series of iterations called sprints, which last from one to four weeks. In the agile Scrum world, a sprint-planning meeting is described in terms of the desired outcome (a commitment to a set of features to be developed in the next sprint) instead of a set of Entry criteria, Task definitions, Validation criteria, Exit criteria. The Scrum model suggests each sprint begins with a brief planning meeting and concludes with a review. These are the basics of Scrum project management.

2.1.2 Scrum Development Model

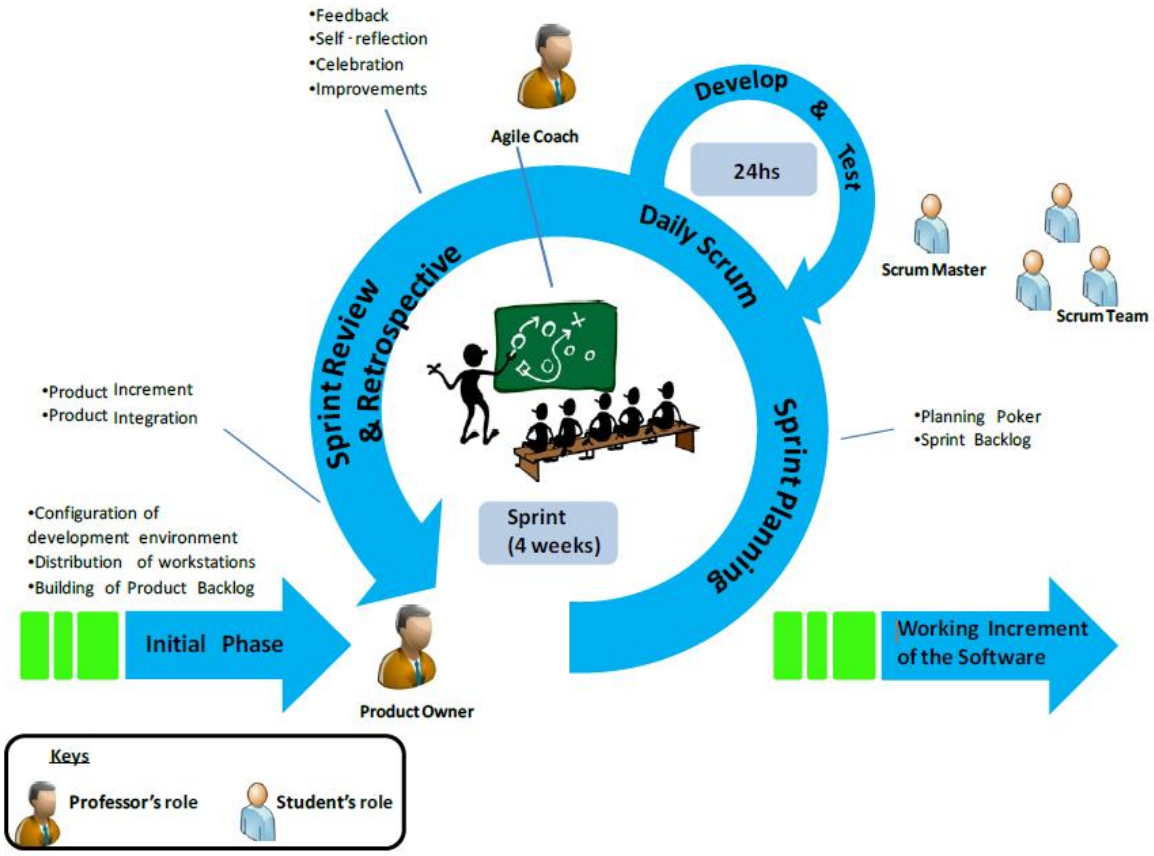


Figure 2: Scrum Development Model

For more information:

https://www.scrum.org/Resources/What-is-Scaled-Scrum

2.1.3 Reasons for Choosing

Project is developed under scrum model. We choose this model because the scope of the project is not fixed when the requirement changes day by day. Products are created quickly. Therefore, the development team can easy to change if the wrong direction. Degree of cooperation between the members is set to high.

2.2 Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Đỗ Đức Minh Quân	Scrum Master/Product Owner	• Defining user requirements • Specifying business • Control the development process • Give advices on techniques, solutions and business analysis support

2	Nguyễn Đức Lợi	Scrum Master/Product Owner	• Give advices on techniques, solutions and business analysis support
3	Nguyễn Hữu Kỳ Long	Team Leader, BA, DEV, Tester	• Managing process • Clarifying requirements • Researching solutions and techniques • Assigning task for members • Reviewing the result of task of members. • Editing documents and reports • Reviewing documents and reports • Developing the system software • Reviewing the system hardware • Coding • Creating test plan. • Testing
4	Nguyễn Đình Tân	Team Member, BA, DEV, Tester	• Clarifying requirements • Researching solutions and techniques • Designing database • Preparing documents and reports • Reviewing documents and reports • Developing the system software • Reviewing the system hardware • Coding • Testing
5	Lê Phương Bình	Team Member, BA, DEV, Tester	• Clarifying requirements • Preparing documents and reports • Reviewing documents and reports • Developing the system hardware • Reviewing the system software • Coding • Testing

6	Nguyễn Xuân Ý	Team Member, BA, DEV, Tester	• Clarifying requirements • Editing documents and reports • Reviewing documents and reports • Developing the system hardware • Developing the system software • Coding • Testing

Table 3: Roles and Responsibilities Details

2.3 Tools and Techniques

- Front-end and back-end IDE:
  - QT Creator version 4
- Front-end technology:
  - QT Linux GUI version 4
- Back-end library:
  - OPENCV library version 2.4.9
  - LIBSVM library version 3.20
  - Espeak library version 1.48.04
  - BCM2835 library version 1.45
- Managing database:
  - SQLite 3
- Connecting to Raspberry PI B2:
  - Remote Desktop Connection Program of Ubuntu 14.04
- Managing the project:
  - SVN tortoise version 1.8.11
  - Rabbit VCS
- Managing documents, reports, models and diagrams:
  - Software Ideas Modeler version 7.70.5385.38708
  - Microsoft Office 2010

3. Project Management Plan

3.1 Product Backlog

ID	Theme	User Type	Wants to...	So that...	Priority	Sprint
1	Detection	User	keep track their hand gesture	can see his/her hand in the screen	Very High	1
2	Device	User	the system is a portable system	move the device easily and use it more flexibly	High	1
3	Recognition	User	recognize the hand signs	express the same meaning to the partner can understand	Very High	2
4	Recognition	User	receive the recognition result via text and sound	express the translated content in a clearly way	Medium	2
5	Detection	User	control the system functions by hand gesture	perform and move beetween the system functions	High	3
6	Power	User	know remaining of battery capacity	can monitor the use of device	Medium	3
7	Recognition	User	increase the accuracy of the recognition result	raise the reliability of the translated content	Very High	4
8	Learning	User	learn the hand sign language	learn new signs or pratice his/her signs	High	4
9	Device	User	turn on/off the system	can turn on/off the device according to the demand	Medium	4
10	Device	User	the system is boxed firm, compact	the component are protected against bumps	Medium	5
11	Recognition	User	the system reliable operation	no error occurs when using	High	5
12	User manual	User	know how to install and use the system	easy to use, repair	High	5

Table 4: Product Backlog Details

3.2 Sprint Backlog

Place at folder ScrumBacklog in Github

3.3 All Meeting Minutes

Place at folder ScrumBacklog in Github

4. Coding Convention

General view of C++ Programming Style put into practice in the project

- Naming Conventions
  - Variable names must be in mixed case starting with lower case.
  - Named constants must be all uppercase using underscore to separate words.
  - Names representing methods or functions must be verbs and written in mixed case starting with lower case.
  - Plural form should be used on names representing a collection of objects
  - The prefix is should be used for Boolean variables and methods
- Include Files and Include Statements
  - Header files must contain an include guard
  - Include statements should be sorted and grouped
  - Include statements must be located at the top of a file only
- Variables
  - Class variables should never be declared public
  - C++ pointers and references should have their reference symbol next to the type rather than to the name
- Conditionals
  - Complex conditional expressions must be avoided
  - The conditional should be put on a separate line
  - Executable statements in conditionals must be avoided
- Comments
  - Use // for all comments, including multi-line comments
  - Comments should be included relative to their position in the code
  - Class and method header comments should follow the JavaDoc conventions

References

C++ Programming Style Guidelines, Version 4.9, January 2011, Geotechnical Software Services, Copyright © 1996 – 2011
http://geosoft.no/development/cppstyle.html

C. Report No. 3 Software Requirement Specification

1. User Requirement Specification

The system is not only reserved for mute person but also everyone who wants to learn sign language. Therefore, we have determined the requirement from these users:

- Recognize his or her hand signs to text and sound: users want devices that can recognize exactly their hand signs. Then, the device must show recognition results via text on screen and emit pronunciation of this word via speaker.
- Learn the way expressing hand signs: there still are many hand signs that users do not know exactly, they want a device that can help them practice these signs. The system should have images that can describe clearly the way expressing hand sign for user can follow. In addition, the system should have practice function for user practice.
- Controlling the system by hand gesture: users want to perform the operations of the system through his or her hand gesture without electricity devices.
- The system is portable: Users can easily move the system. They expect the system can work at many places, and it still works during a power outage.
- System's power must be controlled: Users can know the remaining battery capacity to monitor the use of equipment. Moreover, they can charge the battery when the battery is low.
- System should be easy to use as the electricity systems people use in daily live: Users can turn on/off the system safely without prejudice to the durability of the equipment.

2. System Requirement Specification

2.1 External Interface Requirement

External interface is concerned with designing interactive products to support the way people communicate and interact in their everyday and working lives. The products must be usability means easy to learn, effective to use and provide an enjoyable experience.

2.1.1 User Interface

- The GUI should be simple, clear, intuitive, and reminiscent.
- The interface is accessible, easy to use, and efficient.
- The interface should meet some criteria such as direct manipulation, device actions, information processing approach, visual features, …
- Each screen has fully instructions of the function implementation. Besides that, it still provides error, success, or implementation notification.

2.1.2 Hardware Interface

- The system must design hardware interface similar to the standard electricity system for anyone can use.

- Provide fully devices of a portable system.
- The system needs to be designed suitable for capturing the hands with an appropriate height, and a width for people can watch the LCD.
- The provided devices should be easy to replace.
- Electricity devices should be packaged in the safety way.

2.1.3 Software Interface

- Linux GUI for Raspbian Operating System.
- The interface must be responsive for LCD 7-inch with the resolution 1280 * 720.

2.2 System Overview Use Case

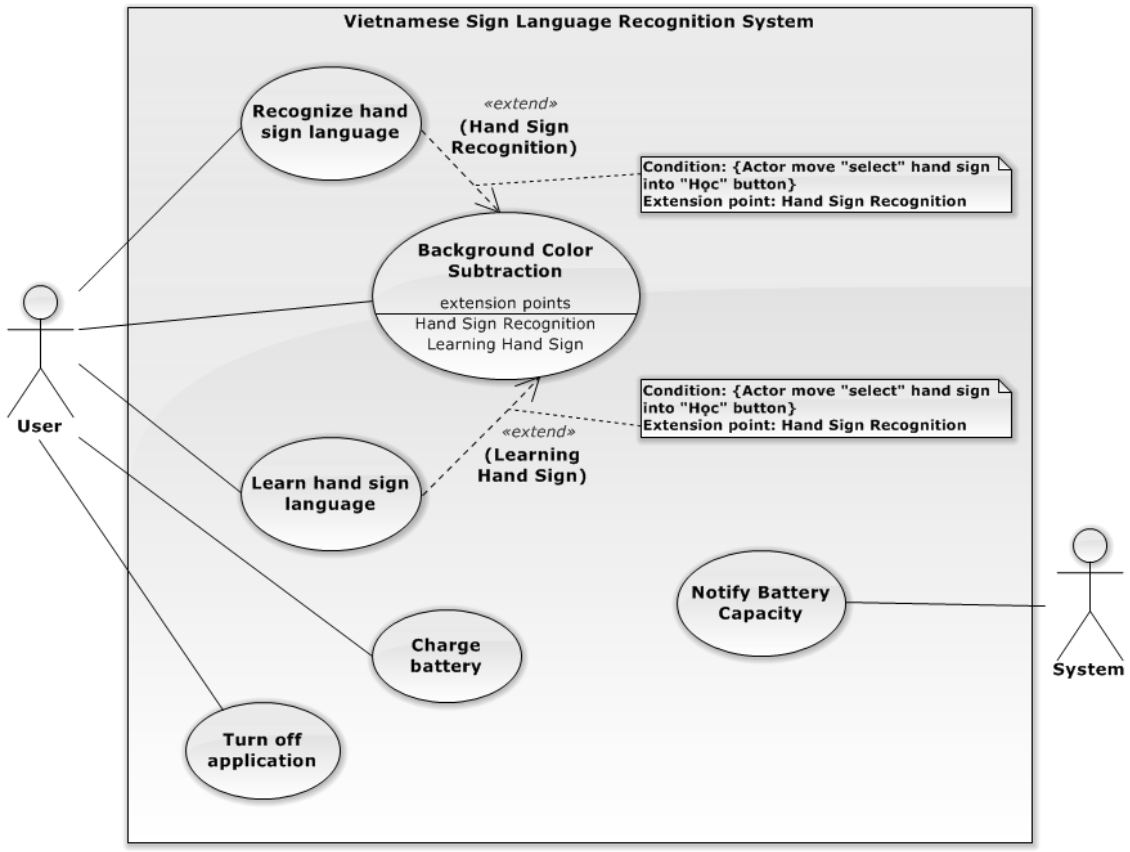


Figure 3: System Overview Use Case

2.3 List of Use Case

2.3.1 Background Color Subtraction

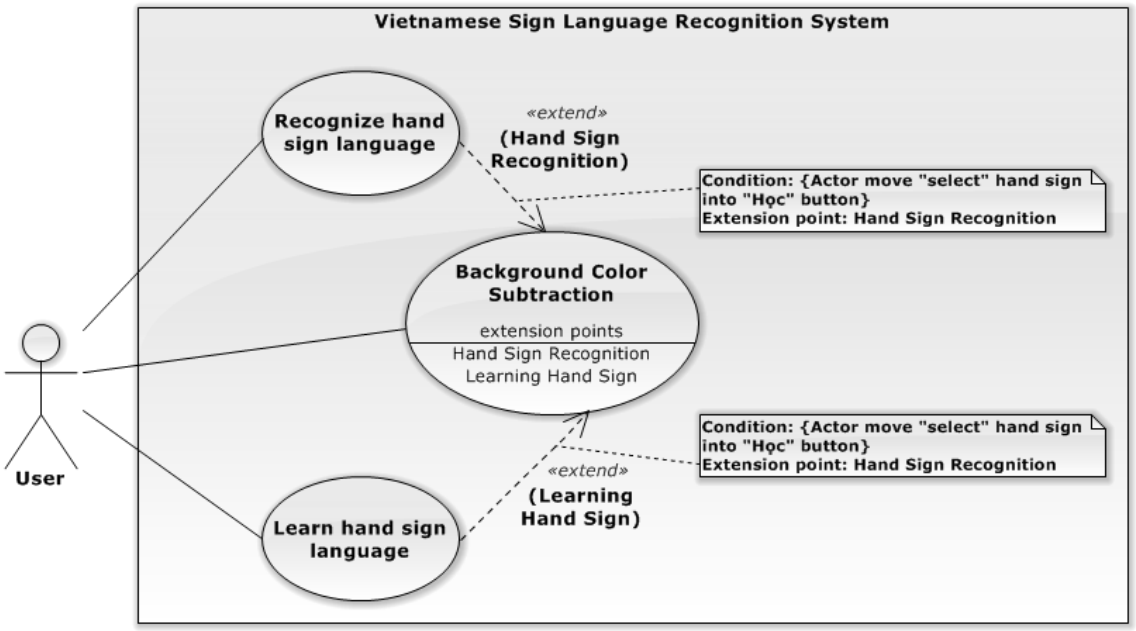


Figure 4: Background Color Subtraction use case diagram

Use Case Specification

USE CASE - 1 SPECIFICATION			
Use-case No.	VSLR001	**Use-case Version**	2.0
Use-case Name	Background Color Subtraction		
Author	Nguyễn Hữu Kỳ Long		
Date	31/05/2015	**Priority**	High

Actor

- User

Summary

- The use case describes the way subtracting background color and checking the result of background color subtraction.

Goal

- The system can detect and keep track of the hands.

Triggers

- User turns on the switch button on the system hardware.

Preconditions

- The system is turned on.

Post Conditions

- **On Success**: The system navigates to the interface to select functions.

Main Success Scenario

Step	Actor Action	System Response
1	- User turns on the switch button on the system hardware.	- The system displays notify “Người dùng vui lòng di chuyển ra khỏi vùng theo dõi.” requiring users move out of the camera area. - The system shows the images captured from camera on the interface inside the groupbox “Hình Ảnh Bàn Tay”. - The system shows the count down time and counts down from 5 by a second inside the groupbox “Thời Gian” - The system shows the message “Đang tiến hành" inside the groupbox “Kết Quả Kiểm Tra”
2	User keeps the background fixed and waits for the count down time is 0.	- The system displays notify “Vui lòng điều chỉnh bàn tay của bạn theo kí hiệu ‘kiểm tra’ trong hướng dẫn.” requiring users show the “testing” hand sign inside the camera area. - The system shows the images subtracting background color on the interface for users inside the groupbox “Hình Ảnh Bàn Tay”. - The system shows the count down time and counts down from 5 by a second inside the groupbox “Thời Gian” - The system shows the message “Đang tiến hành" inside the groupbox “Kết Quả Kiểm Tra”
3	User shows the right “testing” sign inside the camera.	- The system continues counting down timer inside the groupbox “Thời Gian” - The system shows a message “Thành Công” in the groupbox “Kết Quả Kiểm Tra” [Alternative No.1]
4	User waits for count down time is 0.	- The system navigates to the interface for user selects function. [Alternative No.2]

Alternative Scenario

No	Actor Action	System Response

1	User shows the wrong “testing” signs or the background user selected is not good.	- The system continues counting down. - The system shows a message “Đang tiến hành”.
2	The user “testing” sign is not recognized.	- The system shows message “Thất bại” - The system backs to the Step No.1

Exceptions

No	Actor Action	System Response

Relationships

- “Background Color Subtraction” use case is conditionally extended by “Recognize Hand Sign Language” use case and “Learn Hand Sign Language” in extension points “Hand Sign Recognition” and “Learning Hand Sign” respectively.

Business Rules

- The background color subtraction will executes immediately when the application starts working.
- The images captured from camera will shows continuously for users can follow.
- The count down time informs users of the system is working and will finish in five seconds.
- The recognizing “testing” hand sign step is to check whether background is good to recognize and the recognition will execute continuously in five seconds.
- Users can get the “Thành công” message immediately when the “testing” hand sign recognition is successful and the recognition will stop. However, the count down time is still continues down to zero and then the system will move to the function interface.
- After five seconds, if the “testing” sign recognition is unsuccessful, the system will backs to the first step in five seconds.

2.3.2 Recognize Hand Sign Language

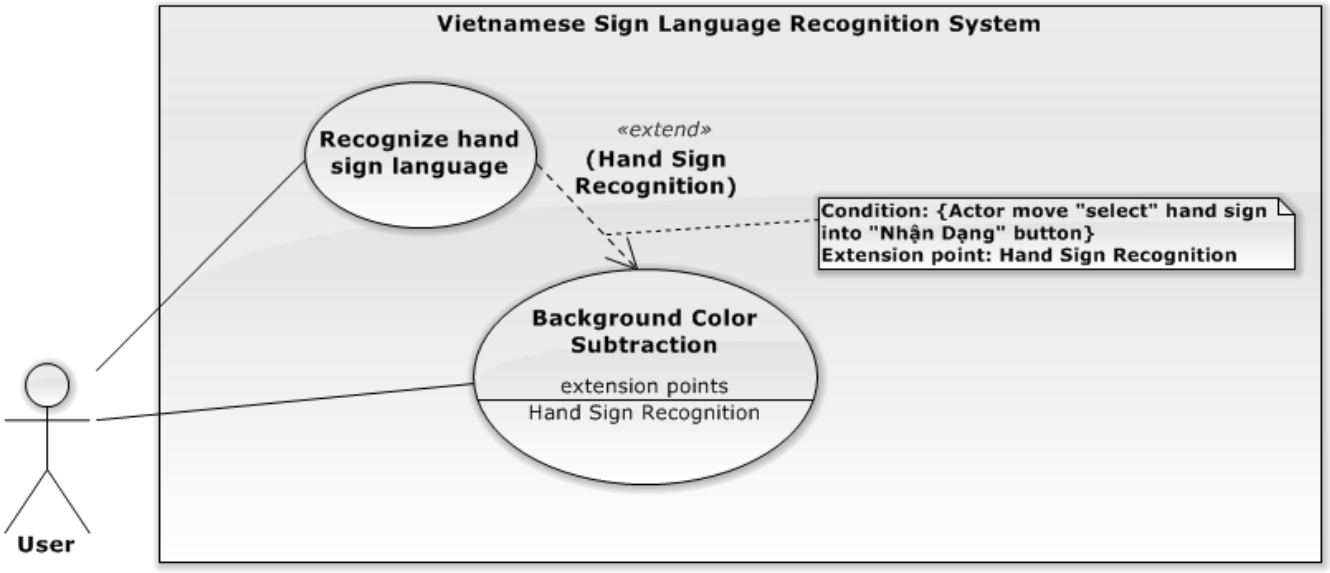


Figure 5: Recognize Hand Sign Language use case diagram

Use Case Specification

USE CASE - 2 SPECIFICATION			
Use-case No.	VSLR002	**Use-case Version**	2.0
Use-case Name	Recognize Hand Sign Language		
Author	Nguyễn Hữu Kỳ Long		
Date	31/05/2015	**Priority**	High

Actor

- User

Summary

- The use case describes the way recognizing hand signs captured by camera.

Goal

- Recognize hand signs and translate them to the same meaning content with the kind of sound and text.

Triggers

- User shows the specific “select” hand sign on the “Nhận dạng” function area that is drawn on the images shown continuously to user.

Preconditions

- Background color subtraction is successful.

Post Conditions

- **On Success**: The translated content shows on the screen and speaker of LCD.

Main Success Scenario

Step	Actor Action	System Response
1	- User completes background color subtraction step	- The system shows a notify “Hãy chọn chức năng mong muốn bằng cách đưa ký hiệu hình bên vào vùng chức năng đó!” in the groupbox “Thông Báo” - The analyzed images show on the interface continuously. - Two white “Nhận Dạng” and “Học” area are drawn inside analyzed images. - The system shows an image guiding users to select function.
2	- User shows “select” hand sign inside “Nhận dạng” area.	- The system shows the hand sign recognition interface - The analyzed images show on the interface continuously. - A notify “Hệ thống sẽ lưu lại kết quả nhận dạng sau 3 giây” is shown in groupbox “Thông Báo” - Countdown time is shown from 3 in groupbox “Thời Gian” - The system shows two groupbox “Nội dung toàn bộ” and “Kết Quả Hiện Tại” with empty content.
3	User shows the hand sign through camera	- Countdown time counts down by second. - The system shows messages containing the recognition result of the current hand sign continuously in the group “Kết Quả Hiện Tại”. [Alternative No.1]
4	User waits for the counting down counts to 0.	- The entire translated content will be updated and shown in the group “Nội Dung Toàn Bộ” - The system backs to step No.3. [Alternative No.2] [Alternative No.3] [Alternative No.4] [Alternative No.5]

Alternative Scenario

No	Actor Action	System Response
1	Cannot detect the hand inside the camera area	- Countdown time counts down by second. - The system shows a message “Không tìm thấy bàn tay!” in the

		group “Kết Quả Hiện Tại”.
2	Detect the “end” hand sign inside the camera area.	- The system navigates to the interface at the step No.1.
3	Detect the “speak” hand sign inside the camera area.	- The system reads the whole content result via LCD speaker. - After that, the whole content will be clear. - The system backs to step No.3
4	No hand inside the camera area	- The system remains the whole content without updating. - The system backs to step No.3
5	Detect the “clear” hand sign inside the camera area.	- The system deletes the last hand sign in the whole content. - The system backs to step No.3

Exceptions

No	Actor Action	System Response

Relationships

- “Recognize hand sign language” use case is an optional extension of “background color subtraction”

Business Rules

- The images captured from camera will be analyzed be system and then shown on the interface for users can keep track their hands.
- The recognition result of the current hand sign will be shown as soon as possible for users can check the hand sign.
- The last recognition result of the hand sign will update to the entire translated content every 3 seconds.
- Every 3 seconds, if the hand sign is recognized with “speak” hand sign, the system will speak the whole content via speaker before clearing the content.
- When the message “Không tìm thấy bàn tay!” is shown, that means the system does not detect user’s hands on the captured image.
- Every 3 seconds, if the system does not detect user’s hands, it will not update the whole content.
- Recognizing works continuously until the user shows “end” hand sign through camera.
- The recognition result is always gotten from the system database.

2.3.3 Learn Hand Sign

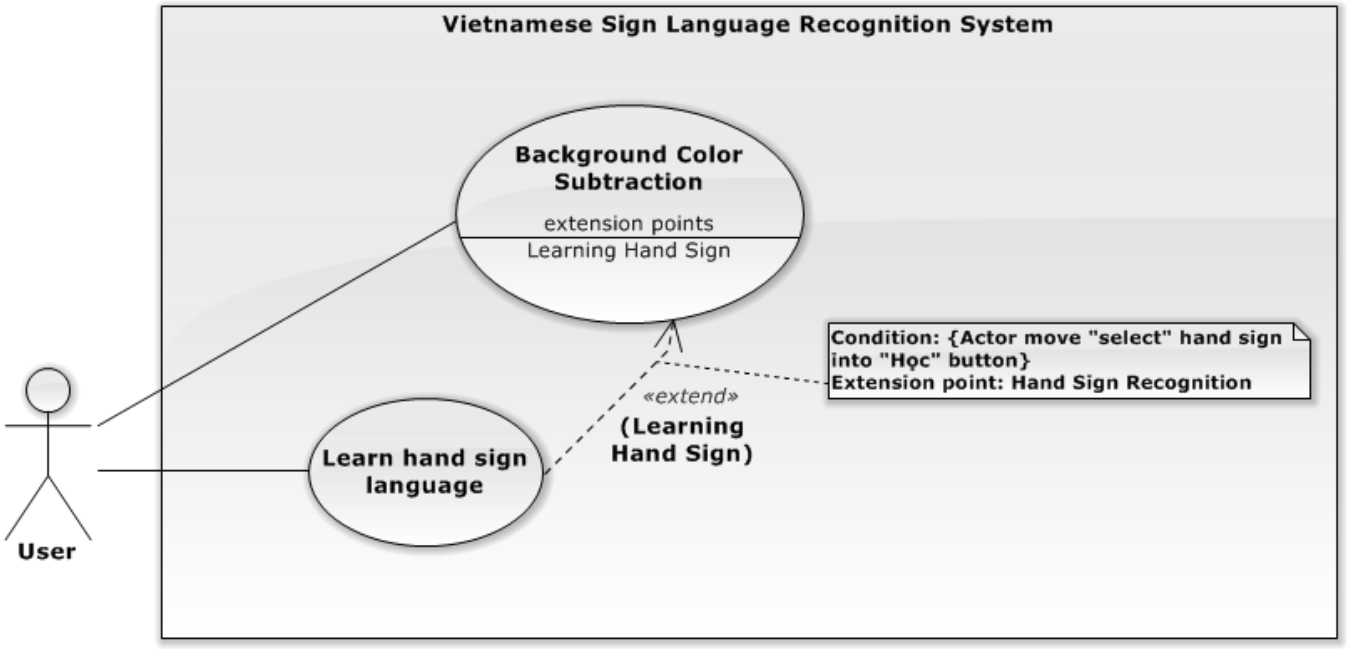


Figure 6: Learn sign use case diagram

Use Case Specification

USE CASE - 3 SPECIFICATION			
Use-case No.	VSLR003	**Use-case Version**	2.0
Use-case Name	Learn Hand Sign		
Author	Nguyễn Đình Tân		
Date	31/05/2015	**Priority**	Medium

Actor

- User

Summary

- The use case describes the way practicing a hand sign.

Goal

- It is to help user training his or her hand gesture more accurately.

Triggers

- User shows the specific "select" hand sign on the "Học" function area that is drawn on the images shown continuously to user.

Preconditions

- Background color subtraction is successful.

Post Conditions

- **On Success**: The system shows the image describing the selected word and the mean of the hand sign, which is captured

Main Success Scenario:

Step	Actor Action	System Response
1	- User completes background color subtraction step.	- The system shows a notify "Hãy chọn chức năng mong muốn bằng cách đưa ký hiệu hình bên vào vùng chức năng đó!" in the groupbox "Thông Báo" - The analyzed images show on the interface continuously. - Two white "Nhận Dạng" and "Học" area are drawn inside analyzed images. - The system shows an image guiding users to select function.
2	- User shows "select" hand sign inside "Học" area.	- The system shows the list of words on the interface in the group box "Hướng Dẫn". - The analyzed images show on the interface continuously. - A notify "Hãy đưa kí hiệu trong hướng dẫn vào vùng mũi tên lên xuống để thay đổi từ được chọn " is shown in the groupbox "Thông Báo" - Two white "Lên" and "Xuống" area were drawn on these images showing on the interface.
3	User shows the hand gesture through camera.	- The system returns the recognition result on text of the current hand sign continuously in the group "Kết Quả Hiện Tại". - The system backs to step No.3 [Alternative No.1] [Alternative No.2] [Alternative No.3] [Alternative No.4]

Alternative Scenario

No	Actor Action	System Response
1	Detect the "select" hand sign inside the "Lên" area drawn on the interface.	- The system will move the selection up to upper word in the list of words. - The system shows the image describing the selected word.
2	Detect the "select" hand sign inside the "Xuống" area drawn on the interface.	- The system will move the selection down to lower word in the list of

		words. - The system shows the image describing the selected word.
3	Detect the “end” hand sign through camera.	- The system navigates to the interface at the step No.1.
4	Detect no hand inside the camera area.	- The system shows a message “Không tìm thấy bàn tay!” in the group box “Kết Quả Hiện Tại”. - The system backs to step No.3

Exceptions:

No	Actor Action	System Response

Relationships

- “Learn hand sign language” use case is an optional extension of “background color subtraction”

Business Rules

- List of words is always loaded from the system database.
- The word “A” will be selected first.
- The image corresponding to the selected word will be loaded and shown on the interface.
- Recognizing processes continuously until the “end” hand sign is recognized.
- Users just select one of two functions.
- When the message “Không tìm thấy bàn tay!” is shown, that means the system does not detect user’s hands on the captured image.
- The “Lên” rectangle is to move selecting up one-step in the list of words.
- The “Xuống” rectangle is to move selecting down one-step in the list of words.
- The recognition result of the current hand sign will be shown as soon as possible for users can check the hand sign.

2.3.4 Charge Battery

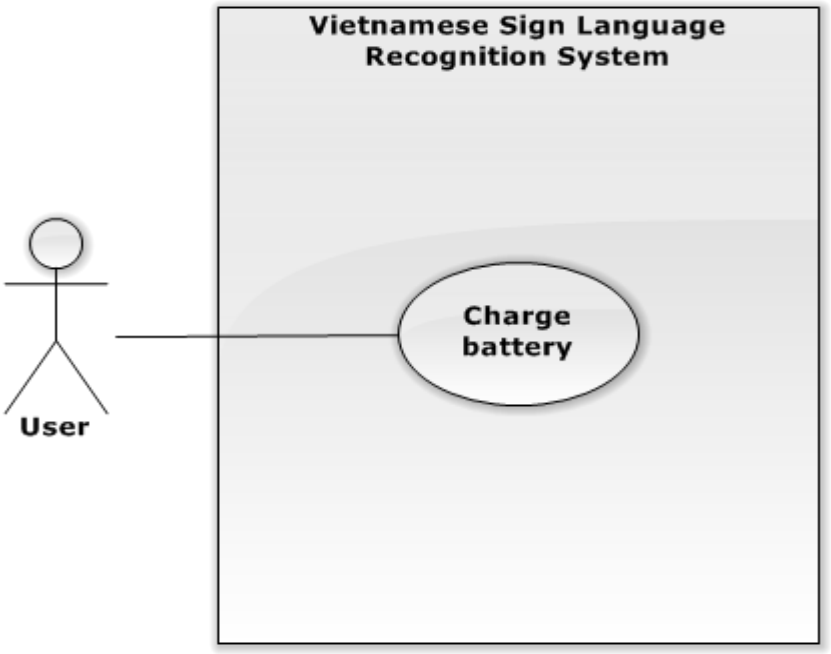


Figure 7: Charge Battery use case diagram

Use Case Specification

USE CASE -4 SPECIFICATION			
Use-case No.	VSLR004	**Use-case Version**	2.0
Use-case Name	Charge Battery		
Author	Lê Phương Bình		
Date	31/05/2015	**Priority**	High

Actor

- User

Summary

- The use case describes users how to know to charge battery.

Goal

- It is to help the system has enough power to operate.

Triggers

- User connect B3AC charger to AC power source 220V.

Preconditions

- N/A

Post Conditions

- **On Success**: The battery charger's LEDs will be bright with green color.

Main Success Scenario:

Step	Actor Action	System Response

1	- User connects LIPO B3AC charger to AC power source 220V.	- The charger's LEDs indicator is bright with red color. [Alternative No.1] [Alternative No.2]
2	- User waits for LIPO battery is charged full. [Alternative No.3]	- Three Charger's LEDs are bright with green color.

Alternative Scenario

No	Actor Action	System Response
1	- Battery is still full.	- The charger's LEDs indicator is bright with green color.
2	- Battery is not ready to charge.	- The charger's LEDs indicator is flickering with red and green color.
3	- User interrupts LIPO B3AC charger connection to AC power source 220V.	- The charger's LEDs indicator is dark.

Exceptions:

No	Actor Action	System Response

Relationships

- N/A

Business Rules

- LIPO battery has three cells and these cells will be charged simultaneously.
- Red color indicates that its cell is charging.
- When the cell is charged full, red color will change to green color.
- When the LEDs indicator is flickering, that means battery maybe is out of order or battery is not connected to the charger.

2.3.5 Notify Battery Capacity

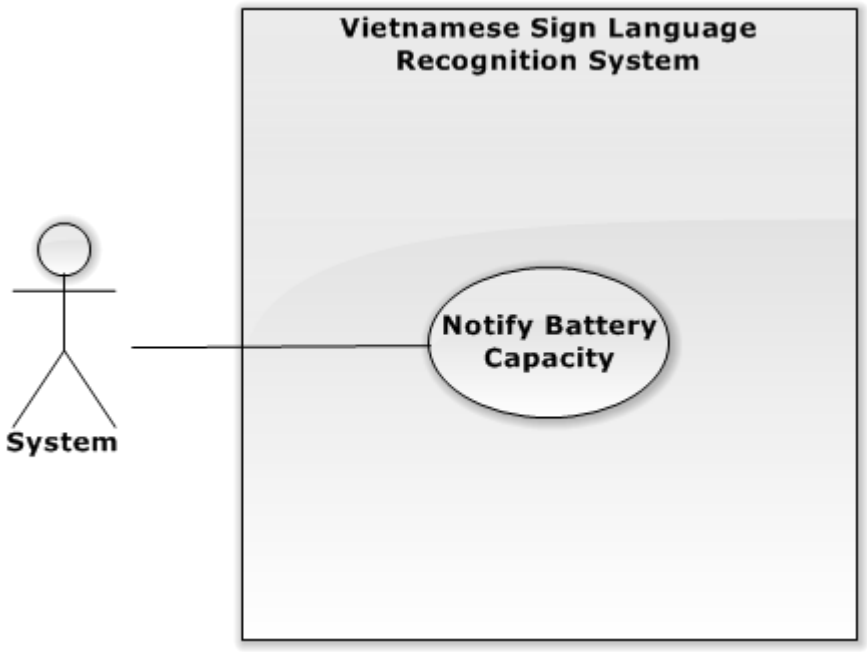


Figure 8: Notify Battery Capacity use case diagram

Use Case Specification

USE CASE - 6 SPECIFICATION			
Use-case No.	VSLR005	**Use-case Version**	2.0
Use-case Name	Notify Battery Capacity		
Author	Nguyễn Xuân Ý		
Date	31/05/2015	**Priority**	High

Actor

- System

Summary

- The use case shows the way system notifies user the battery capacity when system is working.

Goal

- Notify users of the battery capacity can use the system in an appropriate way.

Triggers

- Users switch on the system.

Preconditions

- The system is ON.

Post Conditions

- **On Success**: The battery capacity will shows continuously at the LEDs indicator on the system box and on the interface.

Main Success Scenario

Step	Actor Action	System Response
1	System checks the battery capacity is higher than 75%. [Alternative No.1] [Alternative No.2] [Alternative No.3] [Alternative No.4]	- System shows four bright leds on the box. - System shows the “full” battery image on the top right of the application interface.

Alternative Scenario

No	Actor Action	System Response
1	System checks the battery capacity is between 75% and 50%.	- System shows three bright leds on the box. - System shows the “75%” battery image on the interface.
2	System checks the battery capacity is between 50% and 25%.	- System shows two bright leds on the box. - System shows the “50%” battery image on the interface.
3	System checks the battery capacity is under 25%	- System shows one bright leds on the box. - System shows the “25%” battery image on the interface. - System shows a dialog message “Bin yếu vui lòng tắt hệ thống và cắm sạc. Thông báo sẽ được tự động tắt.” within a countdown time from 5 on the system application interface in five seconds.
4	System checks the battery capacity is empty.	- System shows no bright leds on the box. - System shows the “empty” battery image on the interface. - System closes the application. - System shut down operation system.

Exceptions

No	Actor Action	System Response

Relationships

- N/A

Business Rules

- System will check how much the battery capacity is every five minutes.
- System shows the battery capacity by the electricity devices such as LEDs on the system hardware.
- The notify dialog will show every five minutes if the system checks that battery capacity is under 25% or empty.
- The notify dialog closes automatically after five seconds.
- A countdown time will be shown on the notify dialog.
- Actually, checking battery capacity is to check the voltage level of battery. Battery Capacity Display circuit will check the voltage level of battery continuously and notify on LEDs indicator and it returns 5 levels of battery capacity to the system application every five minutes.
- The battery capacity is higher than 75% if the voltage is higher than 12V.
- The battery capacity is between 75% and 50% if the voltage is lower than or equal 12V and higher than 11.3V.
- The battery capacity is between 50% and 25% if the voltage is lower than or equal 11.3V and higher than 10.8V.
- The battery capacity is lower than 25% if the voltage is lower than or equal 10.8V and 9.9V.
- The battery capacity is empty if the voltage is lower than 9.9V, and that not mean battery is totally out of capacity. However, this capacity is not enough for the system work correctly, so we should turn off the application to protect life span of battery and other devices.

2.3.6 Turn off application

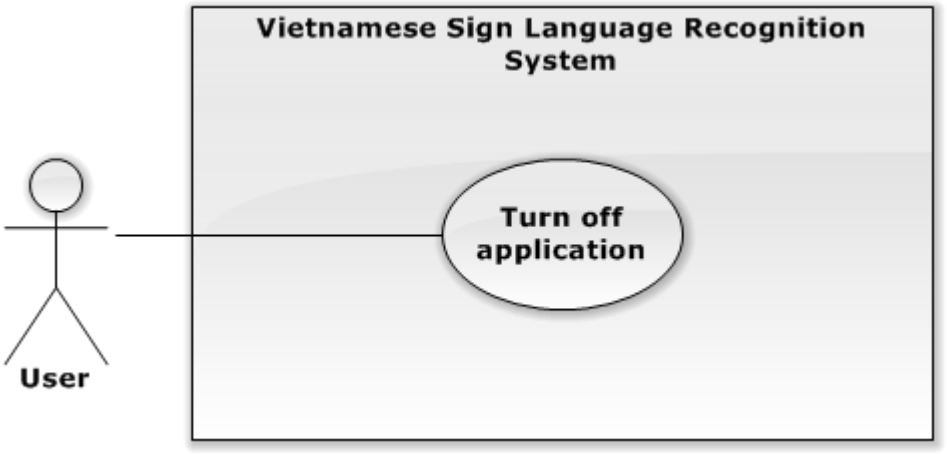


Figure 9: Turn off application use case diagram

Use Case Specification

USE CASE - 6 SPECIFICATION			
Use-case No.	VSLR006	**Use-case Version**	2.0
Use-case Name	Turn off application		
Author	Nguyễn Đình Tân		
Date	31/05/2015	**Priority**	High

Actor

- User

Summary

- The use case shows the way user can turn off application when system is working.

Goal

- User can safely shutdown application

Triggers

- Users press turn off switch the system.

Preconditions

- The system is ON.

Post Conditions

- **On Success**: System will close application is working.

Main Success Scenario

Step	Actor Action	System Response
1	User press black "Turn off" button on the hardware.	- System closes the application. - System shut down Operation System.

Alternative Scenario

No	Actor Action	System Response

Exceptions

No	Actor Action	System Response

Relationships

- N/A

Business Rules

- The system will response immediately when user press the button "Turn off" on the hardware.
- User should press button before switching button "Power" OFF on the hardware to protect the system's operating system and the system's application.

3. System Attribute

3.1 Usability

The system should be designed for everyone can use easily in controlling and GUI operations.

3.1.1 Graphic User Interface

The system musts show all instructions, notifications and operations in Vietnamese.

3.1.2 Usability

User just needs to read the user manual that is enclosed with the system for using in the first time. The attached manual guide must be clear. User can read and do by themselves.

3.1.3 Hardware controlling

User can control the device very easily as well as using any electronic device in the daily live.

3.2 Reliability

- The database should be constructed on Vietnamese sign language.
- The system uses "Support Vector Machine" library to recognize hand sign language and OpenCV library to process image.
- The system is using Raspberry PI B2 to process that is popular board in the world.

- The system carried out the recognition experiment (reference to Recogntion Expermient section in Report 5)

3.3 Availability

The system runs continuously about 3 hours with LIPO 2700mAh battery. That means it is safe to user.

3.4 Security

N/A

3.5 Maintainability

- Electronic devices in the system are common so when any electronic equipment, which is attached with the system, is out of ordered, it is so easy to change or to fix at any electronic store.
- The system can be extended in the future.

3.6 Portability

- The system supplies the LIPO battery as power source in which user can use for 3- 5 hours without charging. In addition, the system also provides LIPO B3AC charger for users.
- The system provides a circuit monitoring LIPO battery for users.

3.7 Performance

The system uses Raspberry PI 2 with RAM 1GB as central unit processing, so that the system can recognize one hand sign in 1 to 3 seconds and hand sign recognize can be performed continuously.

D. Report No. 4 Software Design Description

1. Design Overview

- This document describes the technical and user interface design of VSLR System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
- The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
- The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
- The database design describes the relationships between entities and details of each entity.
- Document overview:
    - Section 2: gives an overall description of the system architecture design.
    - Section 3: gives component diagrams that describe the connection and integration of the system.
    - Section 4: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
    - Section 5: gives the user interface.
    - Section 6: describe a fully attributed ERD.
    - Section 7: explain algorithm to process image.

2. System Architectural Design

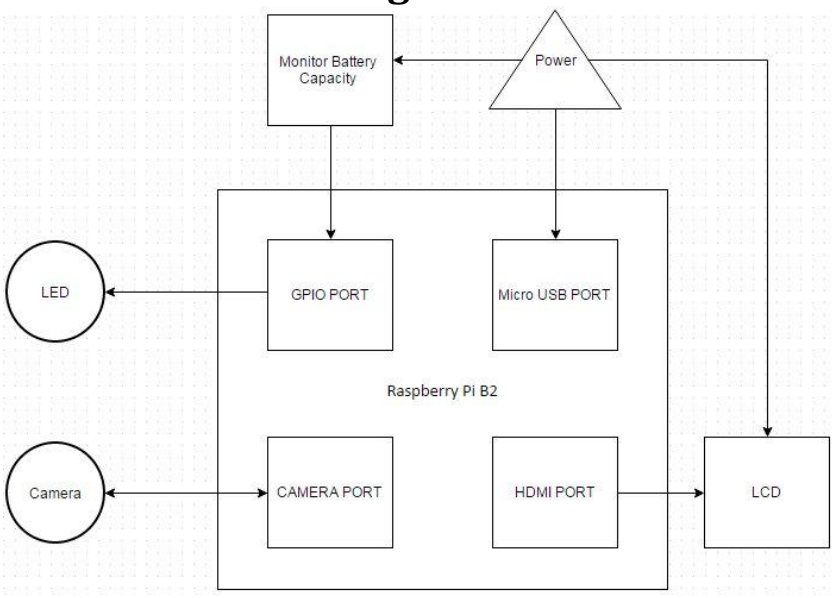


Figure 10: VSLR System Architectural

- **Raspberry PI B2** plays the role as central processing unit. It controls other components, which connects to it through several kinds of port to perform various works.

- **Power** part is where to adjust the power supply for Raspberry PI B2 and LCD to work.
- **Camera** part and **LCD** part takes responsibility for interacting with users, one for getting input and one for outputting result respectively.
- **Monitor Battery Capacity** is a part, which keeps control of power in the system and notify to users.
- **LED** part is to support light in the real environment.

3. Component Diagram

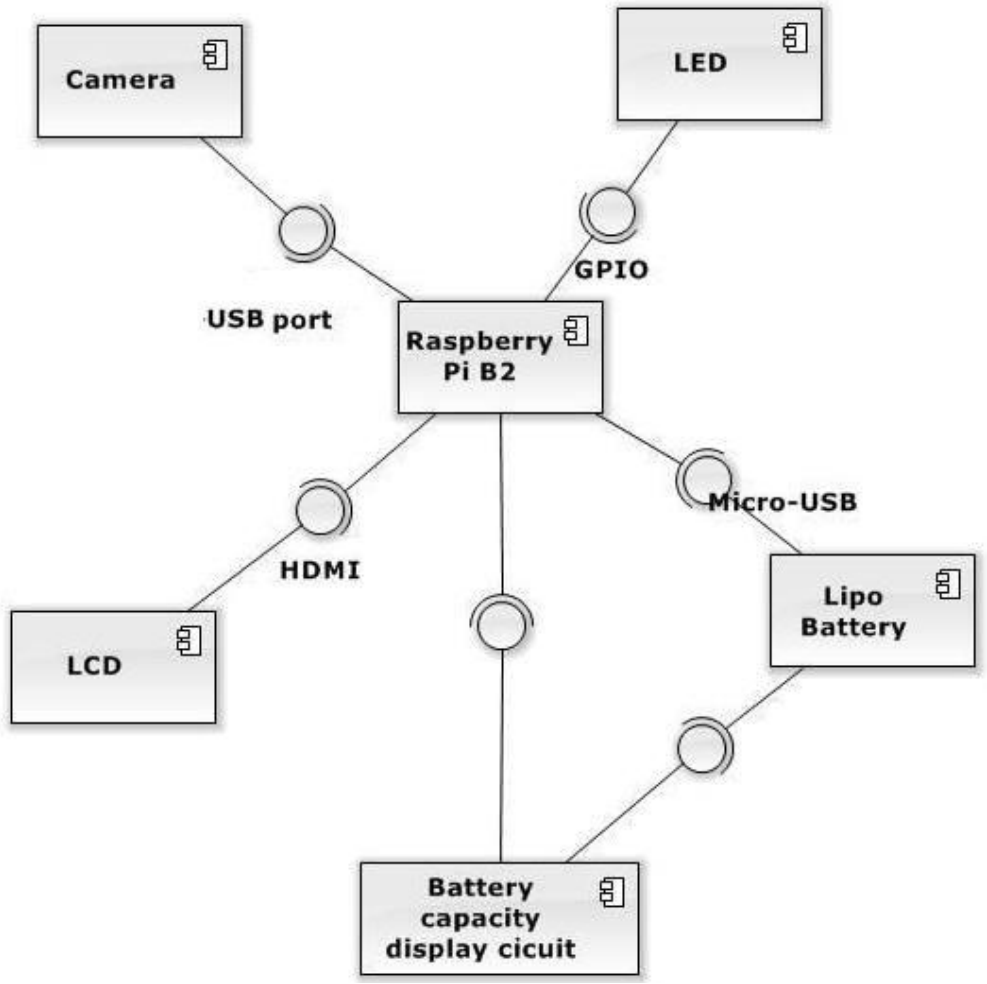


Figure 11: Component Diagram

Component Dictionary	
Component Name	**Description**
Raspberry Pi B2	Central controller
Camera	Capturing the image from users
LED	Supporting light
Lipo Battery	Provide energy for the system works.
Battery Capacity Display Circuit	Display the current battery capacity for users.
LCD	Display interface of application for users.

Table 5: Component Dictionary

4. Detailed Description

4.1 Software Detailed Description

4.1.1 Class Diagram

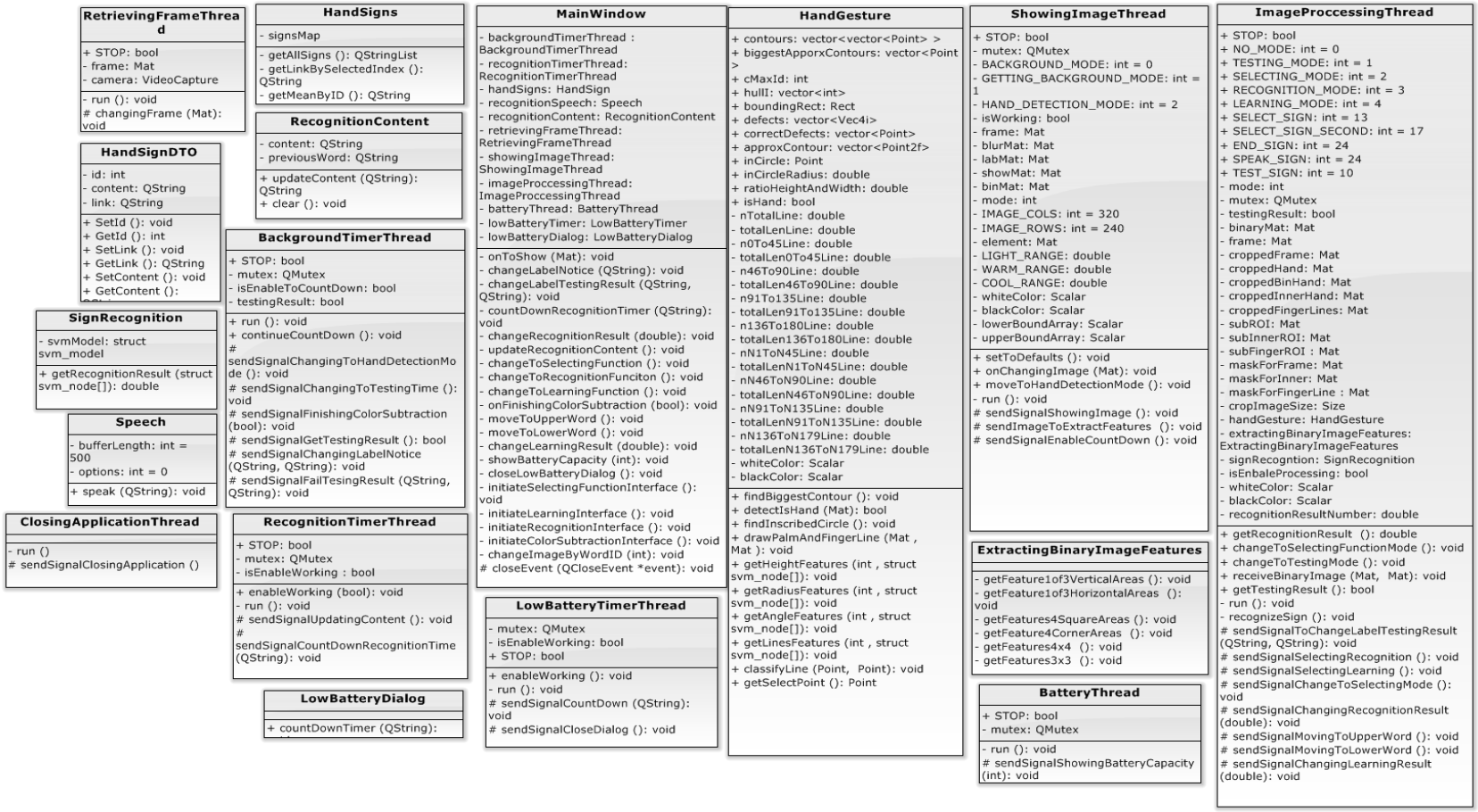


Figure 12: Class Diagram

Class Dictionary : Class Description	
Class Name	Description
HandSignDTO	This is data transfer object class of table HandSign encapsulating HandSign attributes.
HandSigns	This class contains list of hand sign loaded from database and methods for getting attributes of a specific hand sign.
BackgroundTimerThread	This class inherits QThread class that is used for sending signals to notify, count down timer and move between steps in background color subtraction phase. This thread will stop if the phase testing background color subtraction succeeds.
RetreivingFrameThread	This class inherits QThread class that is used for activing camera and then retrieving images captured from camera continuously. This thread will run until the application stops.
ShowingImageThread	This class inherits QThread class that is used for getting image from RetreivingFrameThread when it finish , processing these image to subtract background color which will show on the interface. This thread will run until the application stops.
ImageProcessingThread	This class inherits QThread class that receives image subtracted background color to detect hands, create the binary image containing hand features and then extract the features from these image. This thread receives signals asking to perform image processing for testing background subtraction, learning function and recognition function. This thread just is enabling to process when it receives image from ShowingImageThread. This thread will run until the application stops.
HandGesture	This class encapsulates attributes, methods to create binary images containing hand features and methods to output features related to height, hand palm and finger lines.

ExtractingBinaryImage	This class provides methods, which outputs histogram features of a binary image.
SignRecognition	This class encapsulates attributes and methods to perform SVM algorithm for hand sign recognition.
Speech	This class encapsulates attributes and methods to initiate some parameters for using espeak library to read content in sound via LCD speaker.
RecognitionContent	This class encapsulates attributes and methods to manage the whole recognition content for hand sign language recognition function.
RecognitionTimerThread	This class inherits QThread class that is used for implementing the real time timer to manage hand sign recognition function. This thread will start when the application runs but it is just enable to process when the recognition function is selected. This thread will stop until the application stops.
BatteryThread	This class inherits QThread class that will run during the application works to check battery capacity and send signals notifying user.
LowBatteryDialog	This class is used for initiating dialog interface, which shows low battery announcement.
LowBatteryTimerThread	This class inherits QThread processes as real time timer to manage the time showing LowBatteryDialog.
MainWindow	This class is the main UI thread, which manages the application interfaces and creates communications between thread objects.

Table 6: Class Dictionary & Class Description

4.1.2 Class Diagram Explanation

4.1.2.1 HandSigns

Attribute

Attribute	Type	Visibility	Description
signsMap	QMap<int, HandSignDTO>	Private	This map contains hand sign records loaded from HandSign table.

Table 7: Attribute of HandSigns

Method

Method	Return type	Visibility	Description
getAllSigns	QStringList	Private	Returns all records signsMap hold.
getLinkBySelectedIndex	QString	Private	Return record by index user select in combobox.
getMeanByID	QString	Private	Return record by recognition result returned from SVM.

Table 8: Method of HandSigns

4.1.2.2 HandSignDTO

Attribute

Attribute	Type	Visibility	Description
Id	Int	Private	Unique identifier of a word and recognition result number.
Content	Qstring	Private	Content of a particular sign.
Link	Qstring	Private	Path to directory contain image of hand sign.

Table 9: Attribute of HandSignDTO

Method

Method	Return type	Visibility	Description
SetId	Void	Public	Set value of attribute id
GetId	Int	Public	Get attribute id value
SetLink	Void	Public	Set value of attribute link
GetLink	Qstring	Public	Get attribute link value
SetContent	Void	Public	Set value of attribute content
GetContent	Qstring	Public	Get attribute content value

Table 10: Method of HandSignDTO

4.1.2.3 BackgroundTimerThread

Attribute

Attribute	Type	Visibility	Description
STOP	bool	Public	This variable is used for to stop the thread before the application is closed.
Mutex	QMutex	Private	It is to protect section of code so that only one thread can access at a time.
isEnableToCountDown	bool	Private	This variable is to check whether this thread continues counting down timer.
testingResult	bool	Private	This variable is to held background subtraction testing result and it is condition to stop this thread.

Table 11: Attribute of BackgroundTimerThread

Method

Method	Return type	Visibility	Description
Run	void	Private	This is starting point for the thread and this method implements function of the thread.
sendSignalChangingToHandDetectionMode	void	Private	This is signal that is sent Image Processing thread to move to hand detection step.
sendSignalChangingToTestingTime	void	Private	This is signal which is sent Image Processing thread to move to background subtraction-testing step.
sendSignalFinishingColorSubtraction	void	Private	This is signal which is sent with testing result to notify main thread.
sendSignalGetTestingResult	bool	Private	This is signal which is sent Image Processing thread to get background subtraction testing result and stop testing step.
sendSignalChangingLabelNotice	void	Private	This is signal which is sent with notify content to main thread.
sendSignalFailTesingResult	void	Private	This is signal which is sent to notify main thread of testing result.
continueCountDown	void	Private	This is function that is called in response to signal which enables to continue counting down timer.

Table 12: Method of BackgroundTimerThread

4.1.2.4 SignRecognition

Attribute

Attribute	Type	Visibility	Description
svmModel	struct svm_model	Private	SVM will load the model file to this struct.

Table 13: Attribute of SignRecognition

Method

Method	Return type	Visibility	Description
getRecognitionResult	double	Public	This method receives the hand sign features, then uses SVM library to predict recognition result and return it.

Table 14: Method of SignRecognition

4.1.2.5 Speech

Attribute

Attribute	Type	Visibility	Description
bufferLength	Int	Private	The length in mS of sound buffers passed to the SynthCallback function.
Options	Int	Private	Allow espeakEVENT_PHONEME events.

Table 15: Attribute of Speech

Method

Method	Return type	Visibility	Description
Speak	Void	Public	This is function to speech a string.

Table 16: Method of Speech

4.1.2.6 RecognitionTimerThread

Attribute

Attribute	Type	Visibility	Description
STOP	bool	Public	This variable is used for to stop the thread before the application is closed.
Mutex	QMutex	Private	It is to protect section of code so that only one thread can access at a time.
isEnableWorking	bool	Private	This variable is to check whether the thread can process functions.

Table 17: Attribute of RecognitionTimerThread

Method

Method	Return type	Visibility	Description
enableWorking	void	Public	This is function that is called in response to signal which enables to perform functions.
Run	void	Private	This is starting point for the thread and this method implements function counting down timer.
sendSignalUpdatingContent	void	Protected	This is signal which is sent to notify main thread of performing updating recognition content.
sendSignalCountDownRecognitionTime	void	Protected	This is signal which is sent main thread the real time timer.

Table 18: Method of RecognitionTimerThread

4.1.2.7 RetrievingFrameThread

Attribute

Attribute	Type	Visibility	Description
STOP	bool	Public	This variable is used for to stop the thread before the application is closed.
frame	Mat	Private	This is object holding image captured from camera.
camera	VideoCapture	Private	This is object which actives camera working and retrieves image captured from camera to frame object.

Table 19: Attribute of RetrievingFrameThread

Method

Method	Return type	Visibility	Description
run	Void	Private	This is starting point for the thread and this method implements retrieving images continuously.
changingFrame	Void	Protected	This is signal which is sent to notify Showing Image thread that new image is retrieved.

Table 20: Method of RetrievingFrameThread

4.1.2.8 RecognitionContent

Attribute

Attribute	Type	Visibility	Description
content	QString	Private	This is the whole recognition content will be updated every 3 seconds.
previousWord	QString	Private	This object holds the last updated hand sign.

Table 21: Attribute of RecognitionContent

Method

Method	Return type	Visibility	Description
updateContent	Qstring	Public	This method receives new hand sign recognition result and then performs updating content function.
clear	Void	Public	This method is used for clear the current content.

Table 22: Method of RecognitionContent

4.1.2.9 LowBatteryTimerThread

Attribute

Attribute	Type	Visibility	Description
STOP	bool	Public	This variable is used for to stop the thread before the application is closed.
mutex	QMutex	Private	It is to protect section of code so that only one thread can access at a time.
isEnableWorking	bool	Private	This variable is to check whether the thread can process functions.

Table 23: Attribute of LowBatteryTimerThread

Method

Method	Return type	Visibility	Description
run	void	Private	This is starting point for the thread and this method implements function counting down timer of Low Battery Dialog
sendSignalCountDown	void	Protected	This is signal which is sent main thread the real time timer.
sendSignalCloseDialog	void	Protected	This is signal which is sent main thread to close Low Battery Dialog.
enableWorking	Void	Public	This is function that is called in response to signal which enables to perform functions.

Table 24: Method of LowBatteryTimerThread

4.1.2.10 BatteryThread

Attribute

Attribute	Type	Visibility	Description
STOP	bool	Public	This variable is used for to stop the thread before the application is closed.
mutex	QMutex	Private	It is to protect section of code so that only one thread can access at a time.

Table 25: Attribute of BatteryThread

Method

Method	Return type	Visibility	Description
Run	void	Private	This is starting point for the thread and this method implements retrieving images continuously.
sendSignalShowingBatteryCapacity	void	Protected	This is signal which is to notify main thread of current battery capacity every 5 minutes.

Table 26: Method of BatteryThread

4.1.2.11 LowBatteryDialog

Attribute

Attribute	Type	Visibility	Description

Table 27: Attribute of LowBatteryDialog

Method

Method	Return type	Visibility	Description
countDownTimer	Void	Public	This method is used for changing the timer which is shown on the low battery dialog interface.

Table 28: Method of LowBatteryDialog

4.1.2.12 HandGesture

Attribute

Attribute	Type	Visibility	Description
Contours	vector<vector<Point> >	Public	This holds sets of points as the contours of possible hands.
cMaxId	int	Public	This variable is to determines which contour is of right hand.
boundingRect	Rect	Public	This is rectangle bounding the contour of the right hand.
approxContour	vector<Point2f>	Public	This is contour of the hand in the form of set of float points after it is approximated polygonal curves.
biggestApporxContour	vector<Point >	Public	This is contour of the hand in the form of set of integer points after it is approximated polygonal curves.
hullI	vector<int>	Public	This is convex hull output.
Defects	vector<Vec4i>	Public	This is convexity defects output.
correctDefects	vector<Point>	Public	This is convexity defects output after verifying which is correct.
inCircle	Point	Public	This is center location of hand palm.
inCircleRadius	double	Public	This is radius length of hand palm.
ratioHeightAndWidth	double	Public	This is height feature output of the hand sign.

isHand	bool	Public	This is output of detecting the hands.
nTotalLine	double	Private	This variable holds the number of finger lines.
totalLenLine	double	Private	This variable hold the total length of all finger lines.
n0To45Line	double	Private	This variable holds the number of finger lines belongs to 0 to 45 degree category.
totalLen0To45Line	double	Private	This variable hold the total length of all finger lines belongs to 0 to 45 degree category.
n46To90Line	double	Private	This variable holds the number of finger lines belongs to 46 to 90 degree category.
totalLen46To90Line	double	Private	This variable hold the total length of all finger lines belongs to 46 to 90 degree category.
n91To135Line	double	Private	This variable holds the number of finger lines belongs to 91 to 135 degree category.
totalLen91To135Line	double	Private	This variable hold the total length of all finger lines belongs to 91 to 135 degree category.
n136To180Line	double	Private	This variable holds the number of finger lines belongs to 136 to 180 degree category.
totalLen136To180Line	double	Private	This variable hold the total length of all finger lines belongs to 136 to 180 degree category.
nN1ToN45Line	double	Private	This variable holds the number of finger lines belongs to -45 to -1 degree category.
totalLenN1ToN45Line	double	Private	This variable hold the total length of all finger lines belongs to -45 to -1 degree category.
nN46ToN90Line	double	Private	This variable holds the number of finger lines belongs to -46 to -90 degree category.
totalLenN46ToN90Line	double	Private	This variable hold the total length of all finger lines belongs to -46 to -90 degree category.
nN91ToN135Line	double	Private	This variable holds the number of finger lines belongs to -91 to -135 degree category.
totalLenN91ToN135Line	double	Private	This variable hold the total length of all finger lines belongs to -91 to -135 degree category.

nN136ToN179Line	double	Private	This variable holds the number of finger lines belongs to -136 to -179 degree category.
totalLenN136ToN179Line	double	Private	This variable hold the total length of all finger lines belongs to -136 to -179 degree category.
whiteColor	Scalar	Private	White color is used to draw binary images.
blackColor	Scalar	Private	Black color is used to draw binary images.

Table 29: Attribute of HandGesture

Method

Method	Return type	Visibility	Description
findBiggestContour	void	Public	This method is to find which contour is right.
detectIsHand	bool	Public	This method is to detect the biggest contour is contour of hand or not.
findInscribedCircle	void	Public	This method is to specify the hand palm.
drawPalmAndFingerLine	void	Public	This method is to draw two binary images such as hand palm and finger line.
getHeightFeatures	void	Public	This method returns the height feature of hand.
getRadiusFeatures	void	Public	This method calculates and returns radius features of hand palm.
getAngleFeatures	void	Public	This method calculates and returns angle features of finger lines.
getLinesFeatures	void	Public	This method calculates and returns classifier features of finger lines.
classifyLine	void	Public	This method classifies finger lines to 8-degree categories.
getSelectPoint	Point	Public	This method is to find the highest point of select hand sign and return it.

Table 30: Attribute of HandGesture

4.1.2.13 ShowingImageThread

Attribute

Attribute	Type	Visibility	Description
STOP	bool	Public	This variable is used for to stop the thread before the application is closed.
Mutex	QMutex	Private	It is to protect section of code so that only one thread can access at a time.
BACKGROUND_MODE	int	Private	This is a constant static variable holds value of background mode.
GETTING_BACKGROUND_MODE	int	Private	This is a constant static variable holds value of step sampling background.
HAND_DETECTION_MODE	int	Private	This is a constant static variable holds value of mode detecting hand.

isWorking	bool	Private	This variable is to check whether this thread is working.
Frame	Mat	Private	This is image received from Retrieving Image thread.
blurMat	Mat	Private	This is image after blurring image process.
labMat	Mat	Private	This is image after converting BGR image into LAB image.
showMat	Mat	Private	This is image subtracted background and it is used for showing on the interface.
binMat	Mat	Private	This is binary image subtracted background.
Mode	int	Private	This variable holds value of current mode.
IMAGE_COLS	int	Private	This is constant static variable holds image's width value.
IMAGE_ROWS	int	Private	This is constant static variable holds image's height value.
Element	Mat	Private	This is structuring element is for morphological transformations.
LIGHT_RANGE	double	Private	This is constant static variable holds range value of lightness.
WARM_RANGE	double	Private	This is constant static variable holds range value of warm color-opponent.
COOL_RANGE	double	Private	This is constant static variable holds range value of cool color-opponent.
whiteColor	Scalar	Private	White color is used to draw binary images.
blackColor	Scalar	Private	Black color is used to draw binary images.
lowerBoundArray	Scalar	Private	This array holds lower boundary of every single pixels.
upperBoundArray	Scalar	Private	This array holds upper boundary of every single pixels.

Table 31: Attribute of ShowingImageThread

Method

Method	Return type	Visibility	Description
setToDefaults	void	Public	This method sets this thread back to first step.
onChangingImage	void	Public	This is function that is called in response to signal, which receives new image captured from camera.
moveToHandDetectionMode	void	Public	This is function that is called in response to signal, which changes current mode to hand detection mode.
Run	void	Private	This is starting point for the thread and this method implements processing these image to subtract background color which will show on the interface.
sendSignalShowingImage	void	Protected	This is signal which is to send images subtracted background color to main

			thread can show on the interface.
sendImageToExtractFeatures	void	Protected	This is signal which is to send images subtracted background color to Image Processing thread can extract features.
sendSignalEnableCountDown	void	Protected	This is signal which is to enable Timer thread to continue working.

Table 32: Method of ShowingImageThread

4.1.2.14 ExtractingBinaryImageFeatures

Attribute

Attribute	Type	Visibility	Description

Table 33: Attribute of ExtractingBinaryImageFeatures

Method

Method	Return Type	Visibility	Descrition
getFeature1of3VerticalAreas	void	Private	This method extracts 3 vertical area features of binary image.
getFeature1of3HorizontalAreas	void	Private	This method extracts 3 horizontal area features of binary image.
getFeatures4SquareAreas	void	Private	This method extracts 4 square area features of binary image.
getFeature4CornerAreas	void	Private	This method extracts 4 triangle area features of binary image.
getFeatures4x4	void	Private	This method extract 16 square area features of binary image.
getFeatures3x3	void	Private	This method extract 9 square area features of binary image.

Table 34: Method of ExtractingBinaryImageFeatures

4.1.2.15 ImageProcessingThread

Attribute

Attribute	Type	Visibility	Description
STOP	bool	Public	This variable is used for to stop the thread before the application is closed.
NO_MODE	int	Public	This is a constant static variable holds value of no mode.
TESTING_MODE	int	Public	This is a constant static variable holds value of testing

			background subtraction mode.
SELECTING_MODE	int	Public	This is a constant static variable holds value of selecting function mode.
RECOGNITION_MODE	int	Public	This is a constant static variable holds value of hand sign language recognition mode.
LEARNING_MODE	int	Public	This is a constant static variable holds value of hand sign language learning mode.
SELECT_SIGN	int	Public	This is a constant static variable holds recognition result of “select” hand sign.
SELECT_SIGN_SECOND	int	Public	This is a constant static variable holds recognition result of “select” hand sign.
END_SIGN	int	Public	This is a constant static variable holds recognition result of “end” hand sign.
SPEAK_SIGN	int	Public	This is a constant static variable holds recognition result of “speak” hand sign.
TEST_SIGN	int	Public	This is a constant static variable holds recognition result of “test” hand sign.
Mode	int	Private	This variable holds value of current mode.
testingResult	bool	Private	This is testing background color subtraction result at testing mode.
binaryMat	Mat	Private	This is hand binary image received from Showing Image thread.
Frame	Mat	Private	This is image received from Showing Image thread which is subtracted background color.
croppedFrame	Mat	Private	This is hand images cropped from images subtracted background.
croppedHand	Mat	Private	This is cropped images which contains hand after adjusting size.
croppedBinHand	Mat	Private	This is cropped binary images, which contains hand after adjusting size.
croppedInnerHand	Mat	Private	This is cropped binary images, which contains hand

			palm after adjusting size.
croppedFingerLines	Mat	Private	This is cropped binary images, which contains finger lines after adjusting size.
subROI	Mat	Private	This is region of hand image is to hold hand images cropped.
subInnerROI	Mat	Private	This is region of hand palm image is to hold hand palm cropped.
subFingerROI	Mat	Private	This is region of finger lines image is to hold finger line images cropped.
maskForFrame	Mat	Private	This is a mask which is used for cropping hand images.
maskForInner	Mat	Private	This is a mask which is used for cropping hand palm images.
maskForFingerLine	Mat	Private	This is a mask which is used for cropping finger line images.
cropImageSize	Size	Private	This is common size for every cropped images.
handGesture	HandGesture	Private	This is object processing images to create binary images and output features related to height, hand palm and finger lines.
extractingBinaryImageFeatures	ExtractingBinaryImageFeatures	Private	This is object processing binary images to output histogram features.
signRecogntion	SignRecognition	Private	This object is used for recognize hand sign.
isEnbaleProcessing	bool	Private	This variable is to check whether this thread can process images.
whiteColor	Scalar	Private	White color is to draw binary images containing features.
blackColor	Scalar	Private	Black color is to draw binary images containing features.
recognitionResultNumber	double	Private	This is recognition result predicted by SVM.

Table 35: Attribute of ImageProcessingThread

Method

Method	Return type	Visibility	Description
getRecognitionResult	double	Public	This method return hand sign recognition result.
recognizeSign	void	Private	This method implements steps to recognize hand sign.
changeToSelectingFunctionMode	void	Public	This method is to change current mode to selecting function mode.
changeToTestingMode	void	Public	This method is to change current mode to selecting function mode.
receiveBinaryImage	void	Public	This is function that is called in response to signal, which receives new image subtracted background color.
getTestingResult	bool	Public	This is function that is called in response to signal, which return testing background subtraction result.
Run	void	Private	This is starting point for the thread and this method perform image processing for testing background subtraction, learning function and recognition function.
sendSignalToChangeLabelTestingResult	void	Protected	This is signal which is sent testing background color subtraction to main thread can update the interface.
sendSignalSelectingRecognition	void	Protected	This is signal which is sent notify main thread of changing current mode to recognition mode.
sendSignalSelectingLearning	void	Protected	This is signal which is sent notify main thread of changing current mode to learning mode.
sendSignalChangeToSelectingMode	void	Protected	This is signal which is sent notify main thread of changing current mode to selecting function mode.
sendSignalChangingRecognitionResult	void	Protected	This is signal which is sent notify main thread of outputting recognition result at recognition mode.
sendSignalMovingToUpperWord	void	Protected	This is signal which is sent notify main thread of “Lên” area is selected at learning mode.
sendSignalMovingToLowerWord	void	Protected	This is signal which is sent notify main thread of “Xuống” area is selected atlearning mode.
sendSignalChangingLearningResult	void	Protected	This is signal which is sent notify main thread of outputting recognition result at learning mode.

Table 36: Method of ImageProcessingThread

4.1.2.16 ClosingApplicationThread

Method

Method	Return type	Visibility	Description
Run	void	Private	This is starting point for the thread and this method implements function checking whether button is pressed continuously.
sendSignalClosingApplication	void	Protected	This is signal which is sent main thread to stop all threads and then close the application.

Table 37: Method of ClosingApplicationThread

4.1.2.17 MainWindow

Attribute

Attribute	Type	Visibility	Description
backgroundTimerThread	BackgroundTimerThread	Private	This object is to manage background color subtraction phase.
recognitionTimerThread	RecognitionTimerThread	Private	This object is to manage timer at recognition mode.
handSigns	HandSign	Private	This object is to output hand signs content.
recognitionSpeech	Speech	Private	This object is to speak recognition content.
recognitionContent	RecognitionContent	Private	This object is to manage the whole recognition content at recognition mode.
retrievingFrameThread	RetrievingFrameThread	Private	This is thread object which retrieves images captured from camera continuously.
showingImageThread	ShowingImageThread	Private	This is thread object which subtracts background color.
imageProccessingThread	ImageProccessingThread	Private	This is thread object which processes image to recognize hand sign during the application runs.
batteryThread	BatteryThread	Private	This is thread object which outputs battery capacity every 5 minutes.
lowBatteryTimerThread	LowBatteryTimerThread	Private	This object is to manage timer of Low Battery Dialog.
lowBatteryDialog	LowBatteryDialog	Private	This is UI thread of Low Battery Dialog.

Table 38: Attribute of MainWindow

Method

Method	Return type	Visibility	Description
onToShow	void	Private	This is function that is called in response to signal, which shows images on the interface.
changeLabelNotice	void	Private	This is function that is called in response to signal, which updates notify on the interface.
changeLabelTestingResult	void	Private	This is function that is called in response to signal, which updates testing background color subtraction on the interface.
countDownRecognitionTimer	void	Private	This is function that is called in response to signal, which updates real time timer on the interface at recognition mode.
changeRecognitionResult	void	Private	This is function that is called in response to signal, which updates recognition result on the interface at recognition mode.
updateRecognitionContent	void	Private	This is function that is called in response to signal, which updates new recognition result to the whole content and shows it on the interface at recognition mode.
changeToSelectingFunction	void	Private	This is function that is called in response to signal, which implements steps to change to selecting function mode.
changeToRecognitionFunciton	void	Private	This is function that is called in response to signal, which implements steps to change to recognition mode.
changeToLearningFunction	void	Private	This is function that is called in response to signal, which implements steps to change to learning mode.
onFinishingColorSubtraction	void	Private	This is function that is called in response to signal, which checks the testing background subtraction result.
moveToUpperWord	void	Private	This is function that is called in response to signal, which moves selection to upper word at learning mode.
moveToLowerWord	void	Private	This is function that is called in response to signal, which moves selection to lower word at learning mode.
changeLearningResult	void	Private	This is function that is called in response to signal, which updates recognition result on the interface at laerning mode.
showBatteryCapacity	void	Private	This is function that is called in response to signal, which shows battery

			capacity images on the interfaces.
closeLowBatteryDialog	void	Private	This is function that is called in response to signal, which closes low battery dialog.
initiateSelectingFunctionInterface	void	Private	This methods initiates components of selecting function mode on the interface.
initiateLearningInterface	void	Private	This methods initiates components of learning mode on the interface.
initiateRecognitionInterface	void	Private	This methods initiates components of recognition mode on the interface.
initiateColorSubtractionInterface	void	Private	This methods initiates components of selecting function mode on the interface.
changeImageByWordID	void	Private	This is function that is called in response to signal, which updates image of selected word on the interface at learning mode.
closeEvent	void	Protected	This is an override method which stops threads which is still working before closing application.

Table 39: Method of MainWindow

4.1.3 Sequence Diagram

4.1.3.1 Subtract Background Color

Summary: This diagram shows how the system sample background color, and test background color subtraction

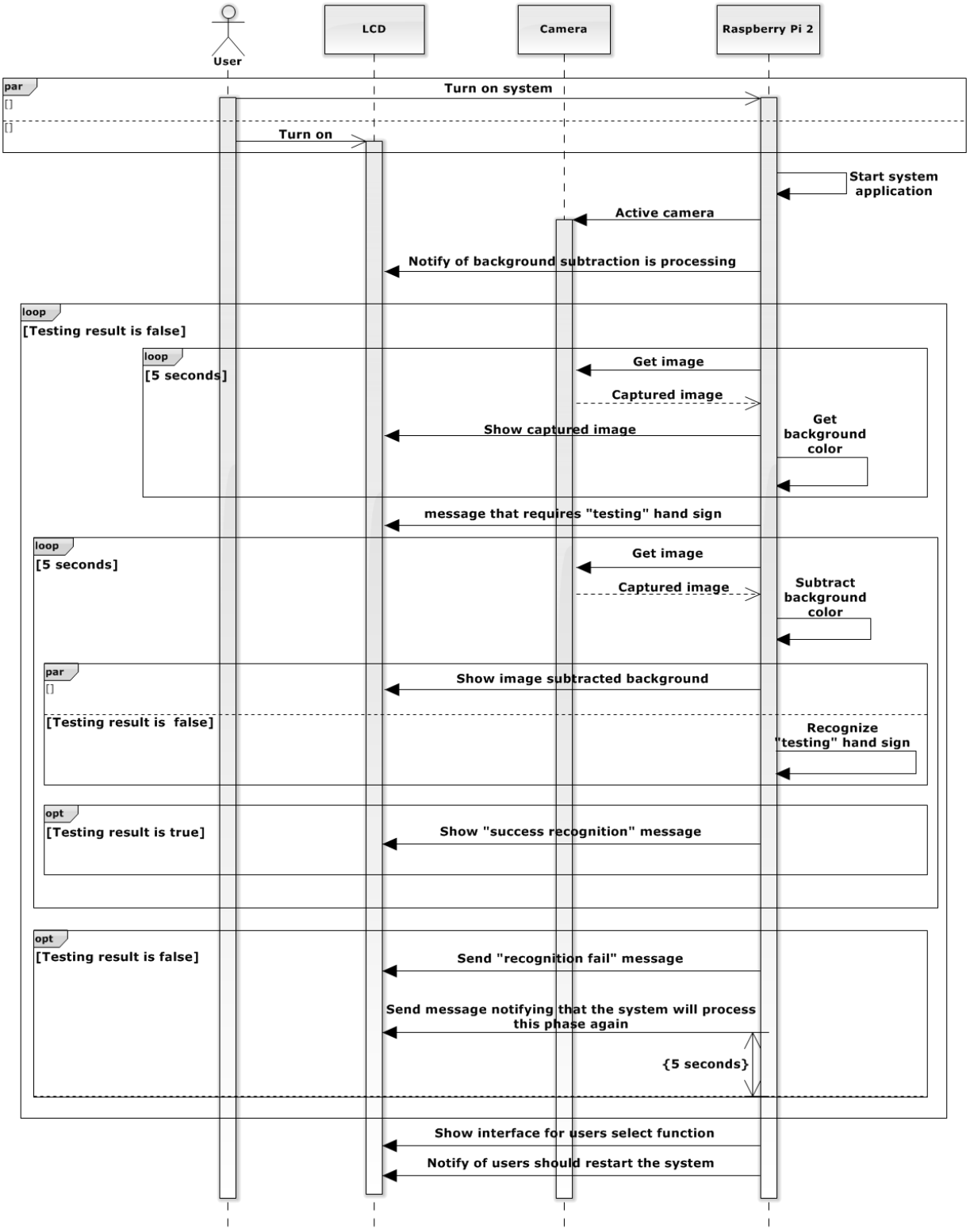


Figure 13: Subtract Background Color Sequence Diagram

4.1.3.2 Recognize Hand Sign Language

Summary: This diagram shows how users recognize their hand signs.

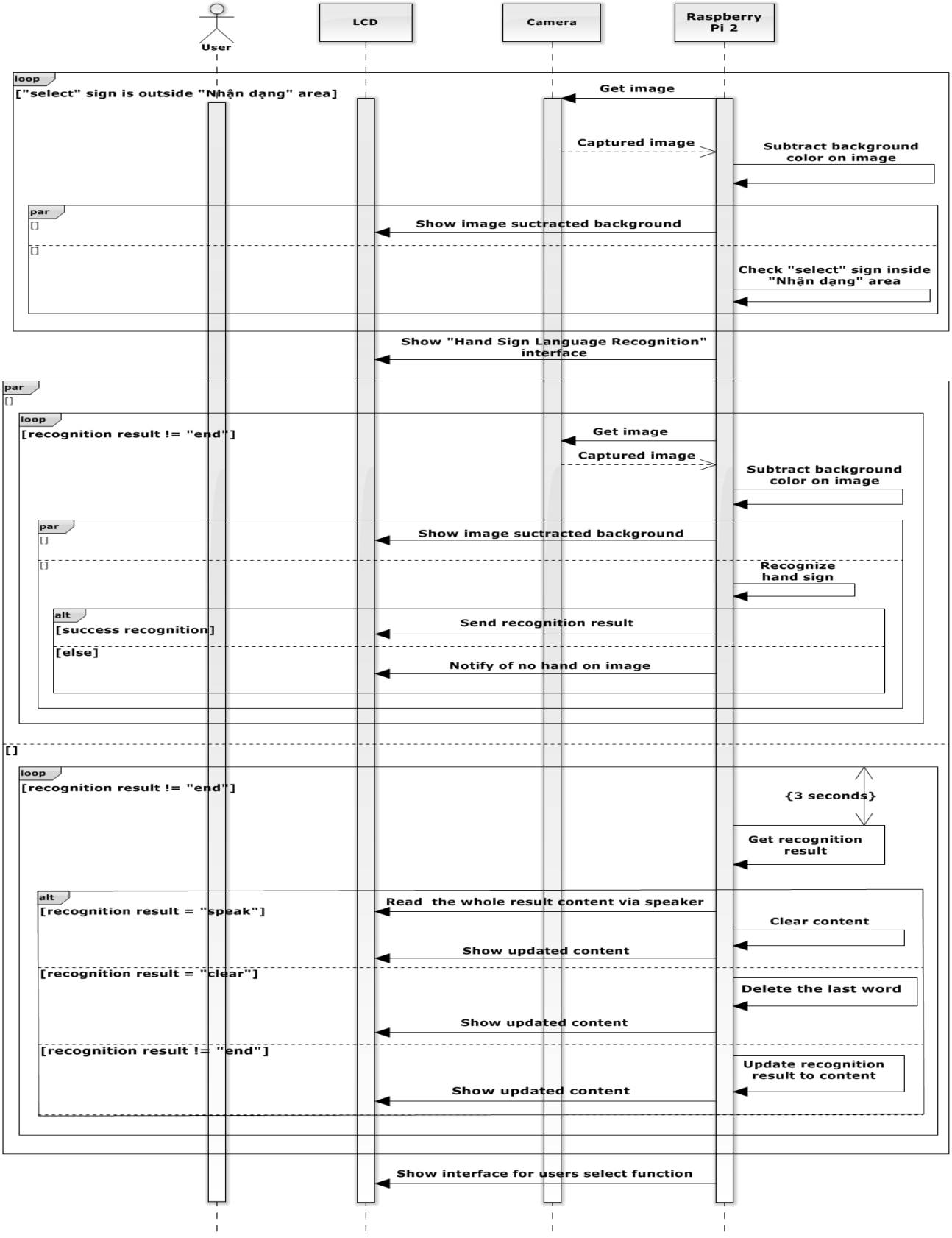


Figure 14: Recognize Hand Sign Language Sequence Diagram

4.1.3.3 Learn Hand Sign Language

Summary: This diagram shows how users learn hand signs.

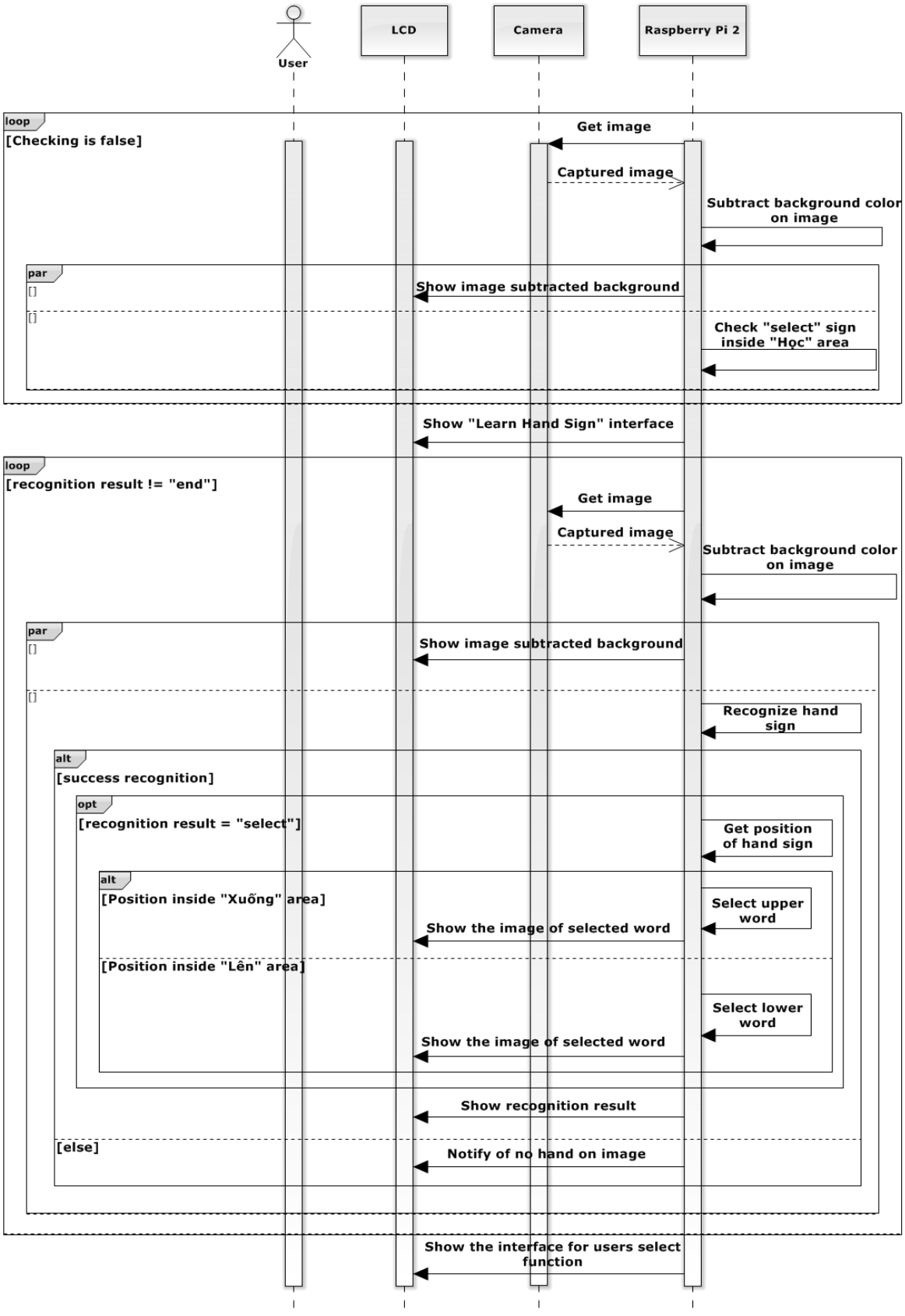


Figure 15: Learn Hand Sign Language Sequence Diagram

4.1.3.4 Notify Battery Capacity

Summary: This diagram shows how system notifies users of battery capacity

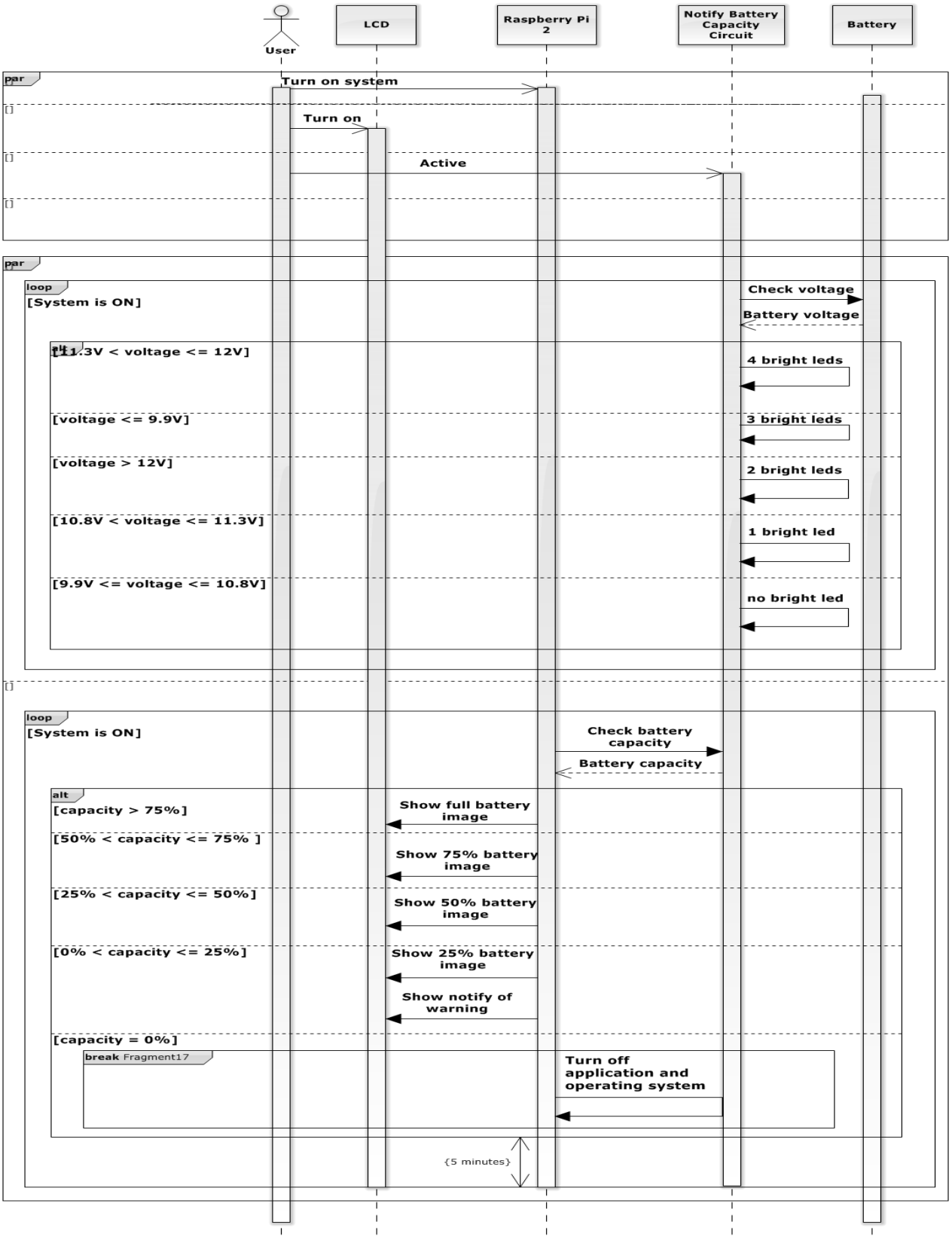


Figure 16: Notify Battery Capacity Sequence Diagram

4.1.3.5 Charge Battery

Summary: This diagram shows the way users charge Lipo battery

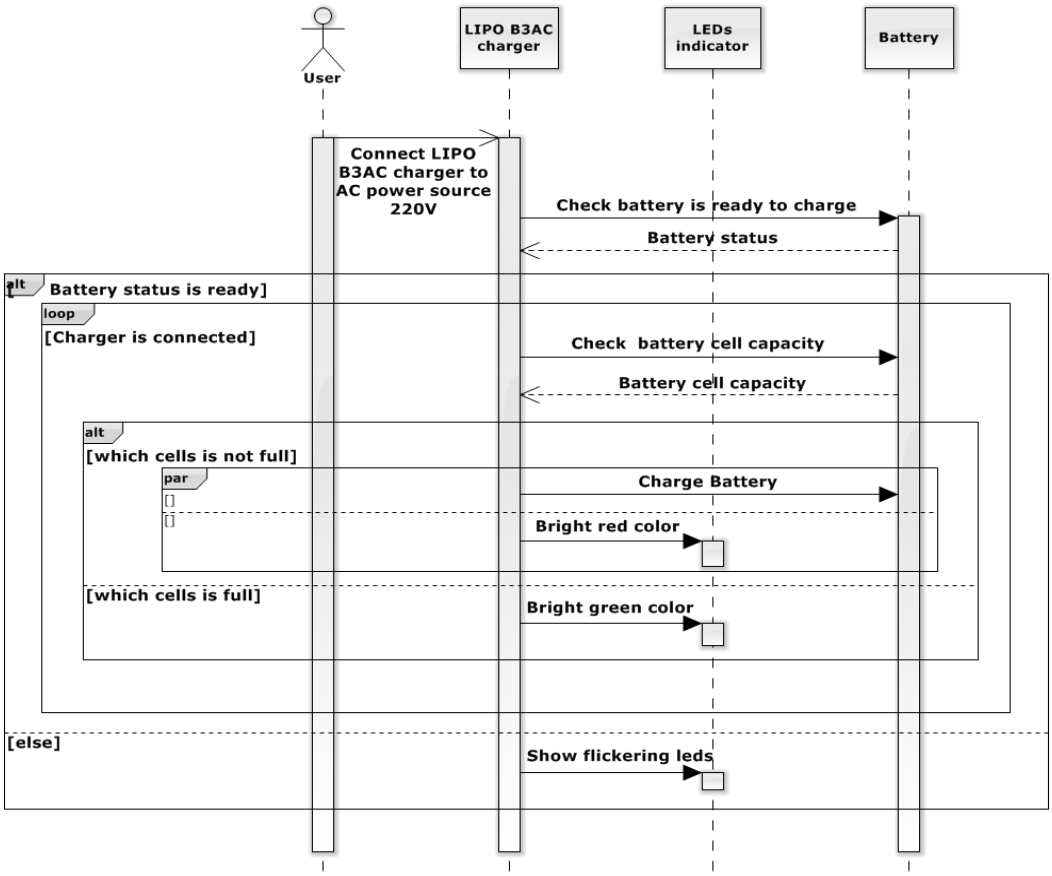


Figure 17: Charge Battery Sequence Diagram

4.1.3.6 Turn off Application

Summary: This diagram shows the way users turn off application when the system is running.

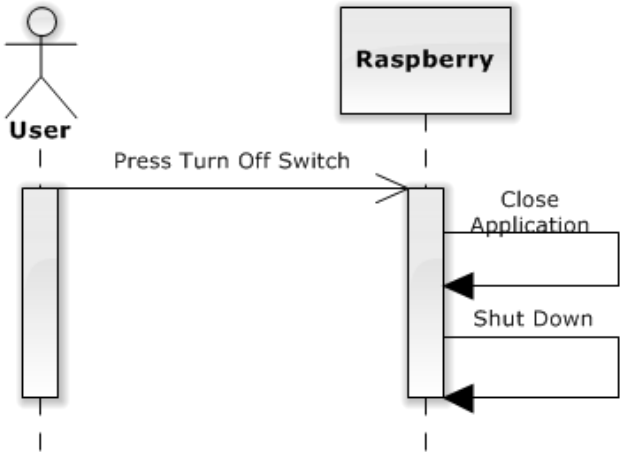


Figure 18: Turn off Application Sequence Diagram

4.2 Hardware Detailed Description

4.2.1 Raspberry PI B2

Figure 19: Raspberry Pi B2 Kit

The Raspberry Pi is a low cost, credit card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. A capable little device enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It is capable of doing everything you would expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

FOR

- Quad-core CPU
- Backwards compatible
- More RAM
- Will fit existing cases

AGAINST

- Could prove intimidating for Linux newcomers
- No micro-USB adapter included

Raspberry Pi B2 specification:

- SoC: Broadcom 2836
- CPU: Quad-core ARM7 800MHz
- GPU: Videocore IV 250MHz
  RAM: 1GB
- GPIO: 40 pin
- Ports: 4x USB 2.0, 100BaseT Ethernet, HDMI and MicroSD card
- Size: 85.60 × 56.5mm (about 3.2 x 2.1-inch)

4.2.2 Power

4.2.2.1 LIPO Battery

Raspberry use input is 5v and 1A so; we do not use battery AA or AAA for Raspberry. We choose lipo battery use for Raspberry Pi B2 kit because lipo battery supplies 11.1v and 1A.

Figure 20: Lipo Battery

4.2.2.2 Voltage Regulator Circuit

However, Raspberry use input is 5v and 1A so, we need a circuit transformer convert from 11.1v to 5v, that reason why we choose LM2576ADJ - 3A UNI REG Board.

Figure 21: LM2576ADJ - 3A UNI REG Board

Overview:

- UNI-REG board allows changing voltages from 7-23V AC (or 9-32V DC) to 5V, 4V, 3.3V, 2.7V or 1.8V.
- Circuit Board using LM2576 - Step-Down Voltage Regulator
- On-board screw-terminals are available for easy connection.

Description:

LM2576ADJ - UNI REG Board uses voltage regulator IC provides functional step-down (buck) switching regulator, capable of responding and changing load voltage lines are excellent. This is the ideal motherboard for projects requiring high voltage switching from lower to AC (DC).

Circuit Board accepts 7-23V AC input voltage (9-32V DC or), and stable output 5V, 4V, 3.3V, 2.7V or 1.8V DC, suitable for most electronics projects. The output voltage is selected via a jumper on the board. Compact and affordable, this board is perfect for use when switching power supplies are needed for your embedded project.

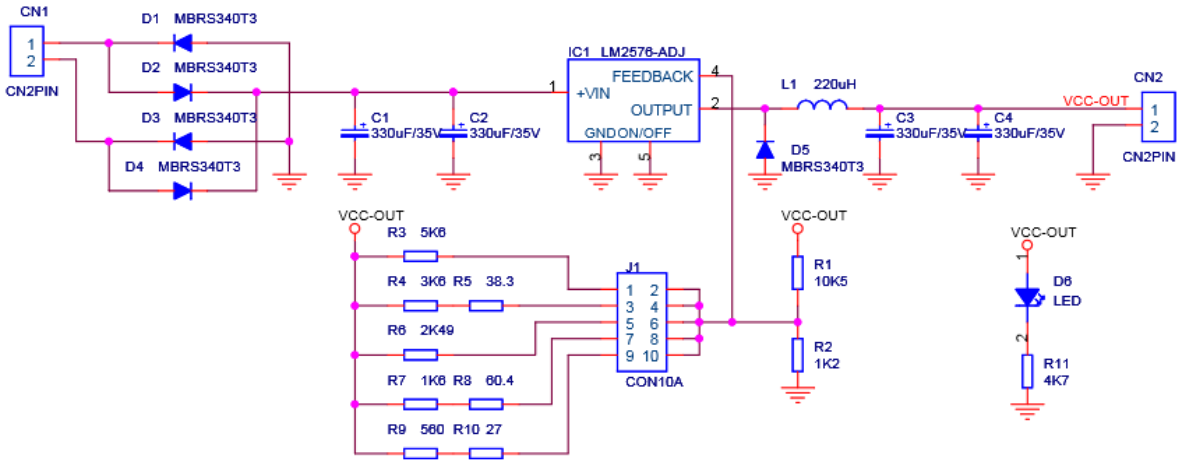


Figure 22: Principle of LM2576ADJ - 3A UNI REG Board

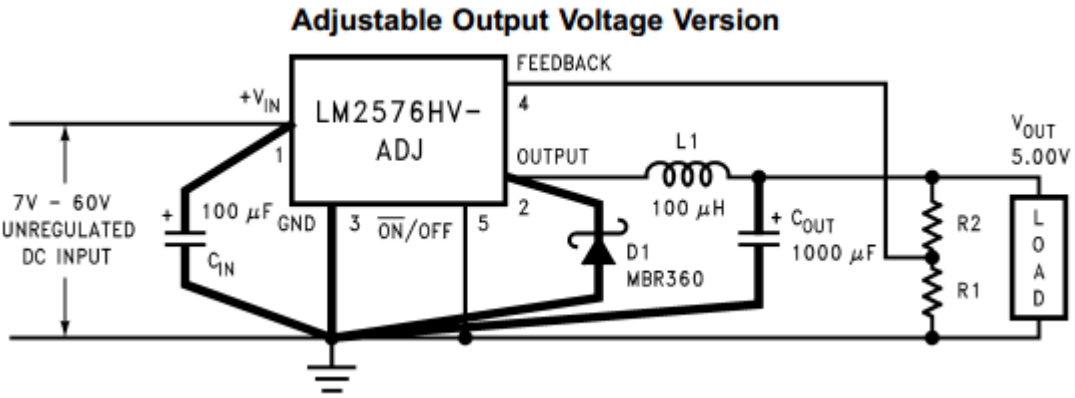


Figure 23: IC LM2576HV-ADJ

$V_{OUT} = V_{REF} (1+ R_2/R_1)$

$R_2 = R_1 (V_{OUT}/V_{REF} - 1)$

Where $V_{REF} = 1.23v$, R_1 between 1k and 5k

4.2.2.3 LIPO Battery Charger – ImaxRC B3AC Compact

Figure 24: LIPO Battery Charger

Overview:

- Input voltage: AC 110V-240V
- Balance charge current: 850mA
- Display: Green & Red LED
- Case: Plastic
- Max charge power: 10W

ImaxRC B3 Compact Charger Operation

- Please firstly connect the B3 Compact Charger to power, while the three power LEDs will turn into green, which indicating that the charger is ready to work.
- Secondly, please connect the battery pack to the charger balance port, while the three power LEDs will all turn into red and charging process begins.
- When the three power LEDs all turn green, the charging process is finished.
- If the three power LEDS is flickering with green and red color, that means LIPO battery is not connected to charger.

4.2.3 The Battery Capacity Display Circuit

Components of the circuit:

- Battery lipo
- TL084
- 1 diode zener 5.1v (1N4733)
- 3 resistors 220 ohm
- 2 resistors 3.3k ohm
- 2 resistors 2.2k ohm
- 5 resistors 1k ohm
- 4 LED

4.2.3.1 TL084

The TL084 JFET-input operational amplifier family is designed to offer a wider selection than any previously developed operational amplifier family. Each of these JFET-input operational amplifiers incorporates well-matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit. The devices feature high slew rates, low input bias and offset currents, and low offset voltage temperature coefficient.

Figure 25: TL084

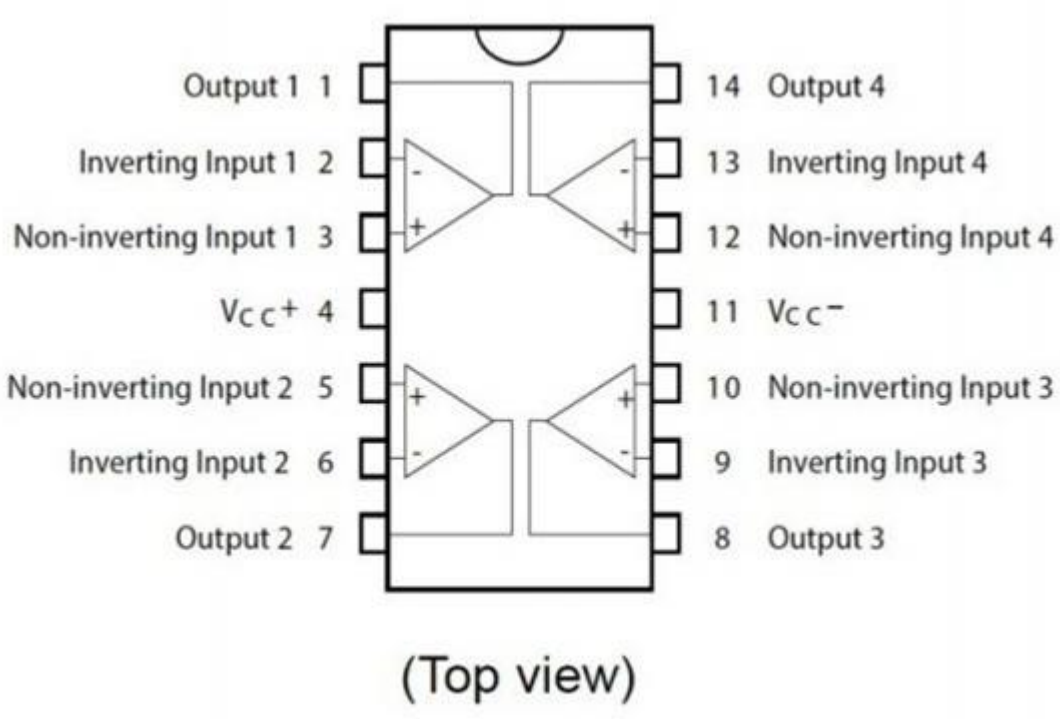


Figure 26: TL084 pin

4.2.3.2 Diode zener

Zener diode, also known as voltage regulator diode, is a semiconductor diode work in reverse polarity mode on the breakdown voltage (breakdown). This voltage is also called Zener voltage avalanche or cascade (avalanche). At that voltage value little changed. It was built so that the reverse polarity, the Zener

diodes will pin a fixed voltage level nearly equal to the value indicated on the diode, do Stabilizers of circuit.

When biased diodes Zener diodes operate like normal. When the polarity invert, at first only a small electric current through the diode truth, but if the voltage is increased to a value inversely adaptation: Vnguoc = Vz (Vz: Zener voltage), the current through diodes increase, but the voltage between the two ends of the diodes hardly change, so-called Zener effect.

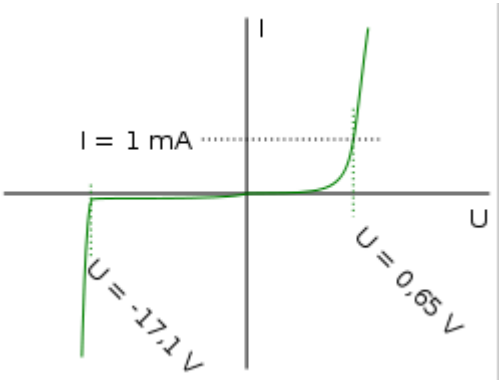


Figure 27: Current-voltage characteristic of a Zener diode with a breakdown voltage of 17 volts. Notice the change of voltage scale between the forward biased (positive) direction and the reverse biased (negative) direction.

Figure 28: Diot Zener 5.1 V

4.2.3.3 Principles of circuit

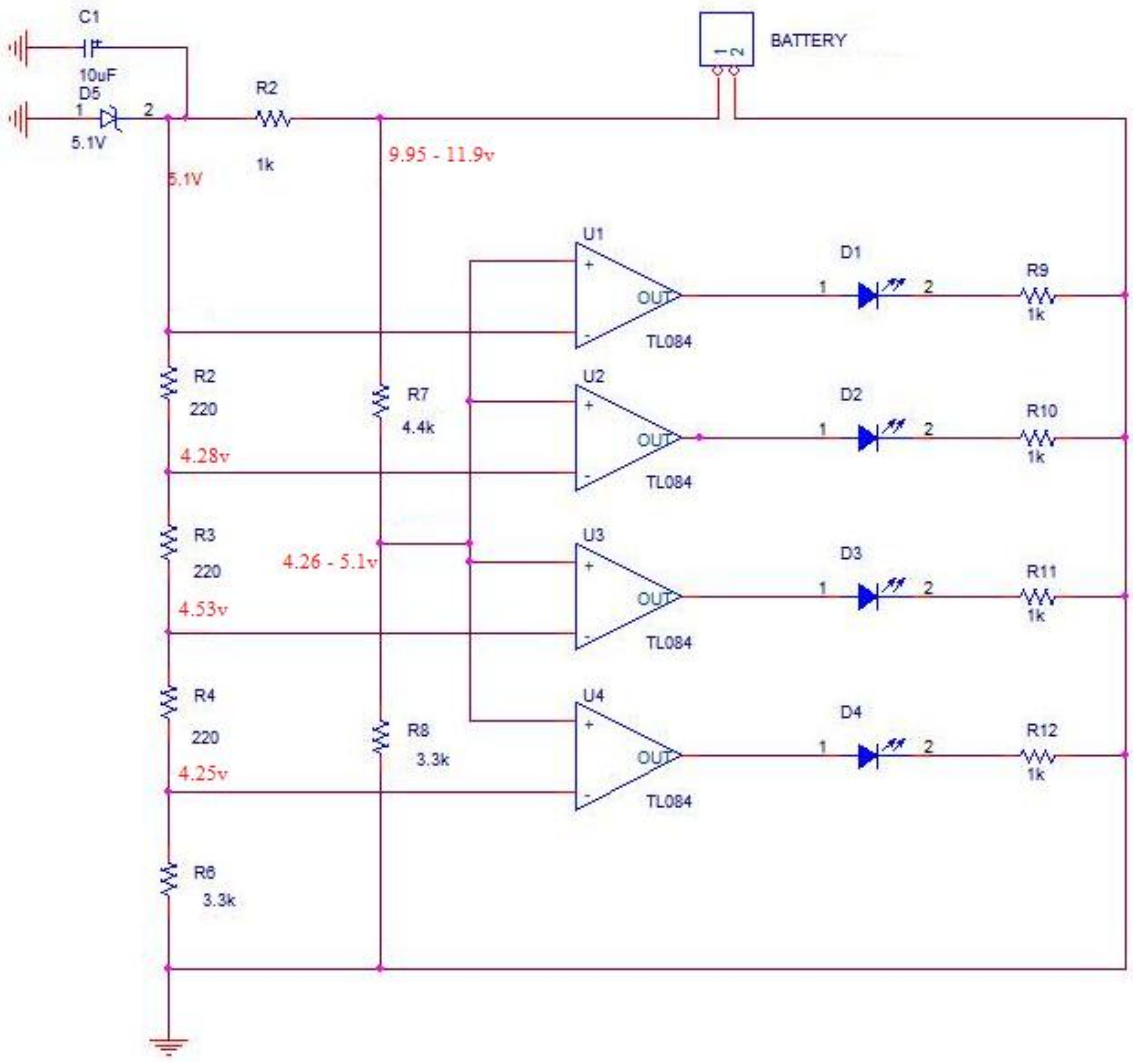


Figure 29: Principle of Monitor Battery Capacity

4.2.4 Camera – Logitech HD Webcam C270

Figure 30: Camera – Logitech HD Webcam C270

Overview:

- HD video calling (1280 x 720 pixels)
- Video capture: Up to 1280 x 720 pixels
- Logitech Fluid Crystal™ Technology
- Photos: Up to 3.0 megapixels
- Hi-Speed USB 2.0 certified

We choose Camera – Logitech HD Webcam C270 because it meets our needs toi.no have high-resolution cameras and image processing at high speed. Besides, low price and high reliability, we chose this device.

It communicates with Raspberry via a USB connector and the driver has been installed. Therefore, we are very easy to use it.

4.2.5 LCD - Tontec® 7 Inch HD TFT Color Monitor

Figure 31: LCD - Tontec® 7 Inch HD TFT Color Monitor

Overview:

- Screen size: 7 inch Display, ration 16:9/4:3 adjustable.
- Display Component: Color TFT-LCD
- Resolution:1280*720 Pixels
- Supports adjust the Brightness, contrast, color.
- Input Signal: PC(VGA) ,HDMI
- Power: DC 5 - 12V
- Consumption: 6W
- Operate mode: the key operation and remote control
- Automatically displays blue screen when no signal
- Product Dimensions: 179*120*20mm (L*W*D)

We use Tontec® 7 Inch High Resolution 1280*720 because it meets our needs. It has a high resolution, compact size and mode power consumption less. Therefore, it is suitable for a portable system. Besides, it supports HDMI to connect easily with Raspberry.

4.2.6 Components Connection

4.2.6.1 Overview Components Connection

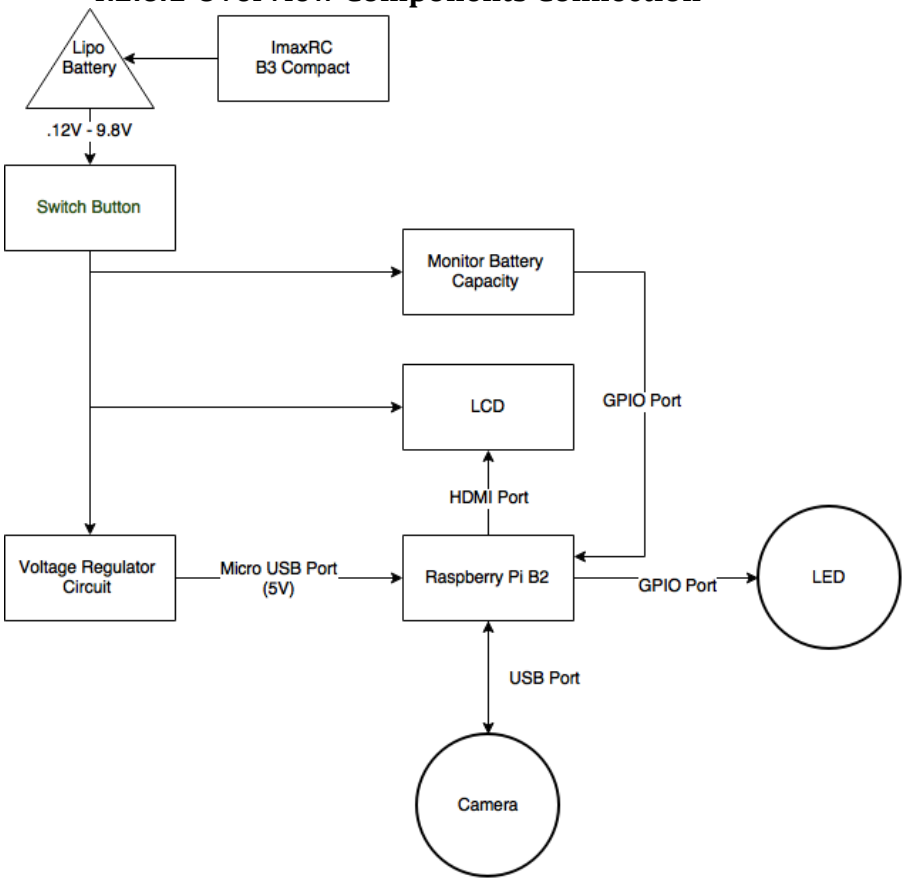


Figure 32: Overview Components Connection

Components Connection Detail		
Device 1	**Device 2**	**Connection**
ImaxRC B3 Compact	Lipo Battery	Jack charge
Lipo Battery	Voltage Regulator Circuit	Screw-terminals
Monitor Battery Capacity	Raspberry Pi B2	GPIO4, GPIO17, GPIO27, GPIO22
LCD	Raspberry Pi B2	HDMI Port
LED	Raspberry Pi B2	Pin1, Pin6
Camera	Raspberry Pi B2	USB Port
Voltage Regulator Circuit	Raspberry Pi B2	Micro-USB Port

Table 40: Components Connection Detail

4.2.6.2 Connection Beetween Monitoring Battery Capacity and Raspbbery PI B2

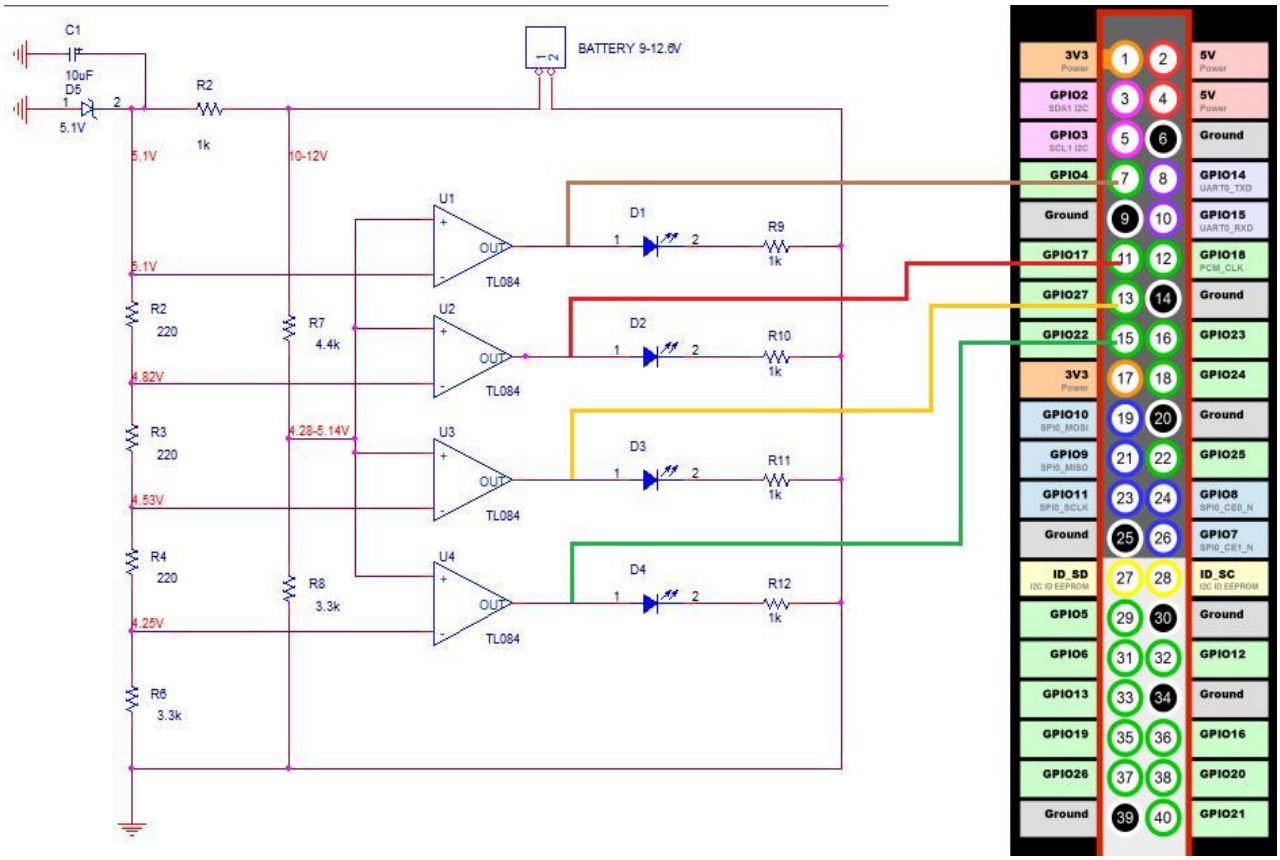


Figure 33: Connection between Monitoring Battery Capacity and Raspberry PI B2

GPIO4, GPIO17, GPIO27, GPIO22 are configured as input pins. We use BCM2835 library to read the input value of the four pins. The battery capacity is calculated by how many feet the output signal at a high level.

4.2.6.3 Connection between LED And Raspberry PI B2

Figure 34: Connection between LED luxeon 1W 350mA and Raspberry Pi B2

4.2.6.4 Connection between LIPO Battery and Charger

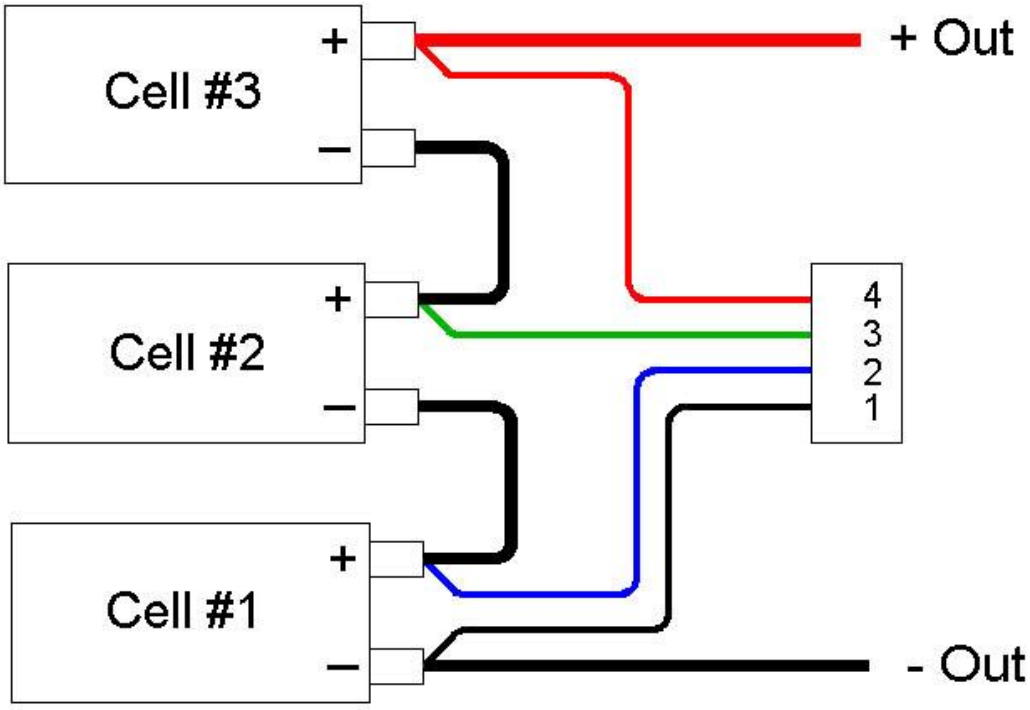


Figure 35: Connection Lipo Battery with Charger

Lipo battery consists of 3 cells, which are connected in series with each other. When they are charged simultaneously charging together as above picture, we call such a charger is charging balance.

5. User Interface Design

5.1 Software Interface

5.1.1 Subtract Background Color

Figure 36: Subtract Background Color

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	BatteryCapacity	Image describing current battery capacity	Yes	Yes	QLabel	QImage	N/A
2	NotifyMessage	Notify of current situation	Yes	Yes	QLabel	QString	N/A
3	Timer	Countdown timer of current function	Yes	Yes	QLabel	QString	N/A
4	CapturedImage	Image captured from camera	Yes	Yes	QLabel	QImage	N/A
5	TestingStatus	Current status of background subtraction step	Yes	Yes	QLabel	QString	N/A

Table 41: Fields of Subtract Background Color

5.1.2 Recognize Hand Sign Language

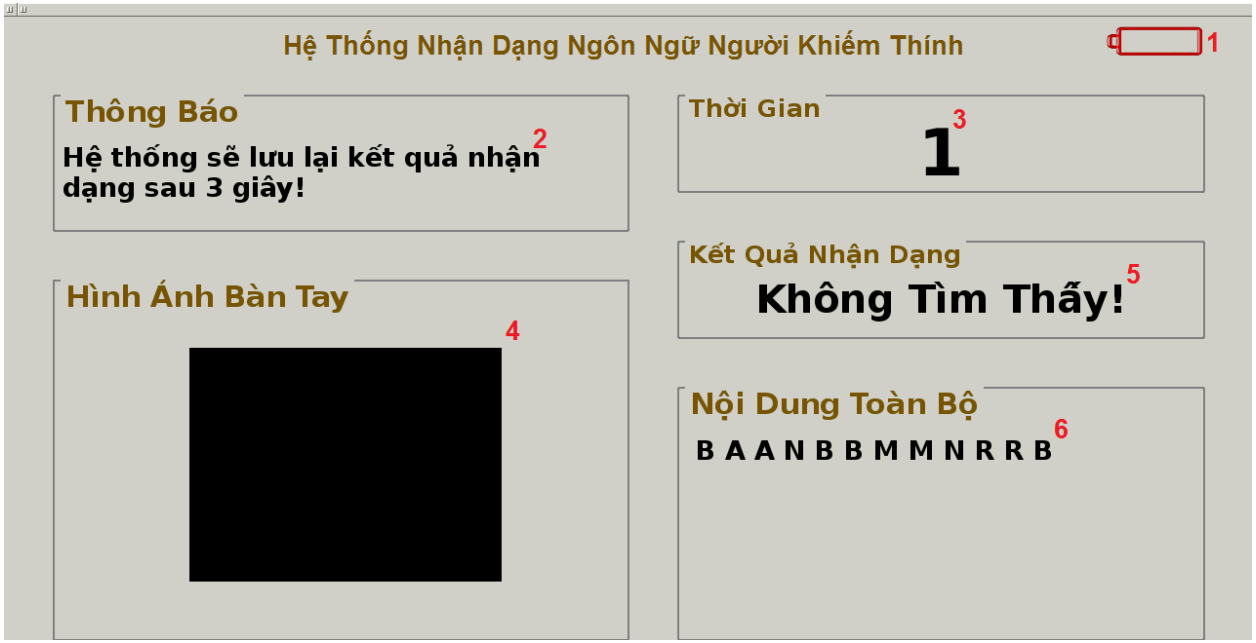


Figure 37: Recognize Hand Sign Language

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	BatteryCapacity	Image describing current battery capacity	Yes	Yes	QLabel	QImage	N/A
2	NotifyMessage	Notify of current situation	Yes	Yes	QLabel	QString	N/A
3	Timer	Countdown timer of current function	Yes	Yes	QLabel	QString	N/A
4	CapturedImage	Image captured from camera	Yes	Yes	QLabel	QImage	N/A
5	RecognitionResult	The result of current hand sign captured from camera	Yes	Yes	QLabel	QString	N/A
6	EntireContent	The whole content of recognition process	Yes	Yes	QLabel	QString	N/A

Table 42: Fields of Recognize Hand Sign Language

5.1.3 Learn Hand Sign Language

Figure 38: Learn Hand Sign Language

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	BatteryCapacity	Image describing current battery capacity	Yes	Yes	QLabel	QImage	N/A
2	NotifyMessage	Notify of current situation	Yes	Yes	QLabel	QString	N/A
3	RecognitionResult	The result of current hand sign captured from camera	Yes	Yes	QLabel	QString	N/A
6	CapturedImage	Image captured from camera	Yes	Yes	QLabel	QImage	N/A
7	ListHandSigns	List of hand signs which is loaded from database	Yes	Yes	QListWidget	QStringList	N/A
8	HandSignImage	Image describing gesture of selected hand sign	Yes	Yes	QLabel	QImage	N/A

Table 43: Fields of Learn Hand Sign Language

No	Function	Description	Validation	Outcome
4	Select Hand Sign	Move the selection to the upper hand sign.	N/A	Upper hand sign is selected and its image is shown.
5	Select Hand Sign	Move the selection to the lower hand sign.	N/A	Lower hand sign is selected and its image is shown.

Table 44: Buttons/Hyperlinks of Learn Hand Sign Language

5.1.4 Selecting Function Interface

Figure 39: Selecting Function Interface

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	BatteryCapacity	Image describing current battery capacity	Yes	Yes	QLabel	QImage	N/A
2	NotifyMessage	Notify of current situation	Yes	Yes	QLabel	QString	N/A
5	CapturedImage	Image captured from camera	Yes	Yes	QLabel	QImage	N/A
6	InstructionImage	Image guides the way selecting function	Yes	Yes	QLabel	QImage	N/A

Table 45: Fields of Selecting Function Interface

No	Function	Description	Validation	Outcome
3	Select Function	Select hand sign recognition function.	N/A	Interface of hand sign recognition function is initiated
4	Select Function	Select learning hand sign language function	N/A	Interface of learning hand sign language function is initiated

Table 46: Buttons/Hyperlinks of Selecting Function Interface

5.1.5 Notify Low Battery Dialog

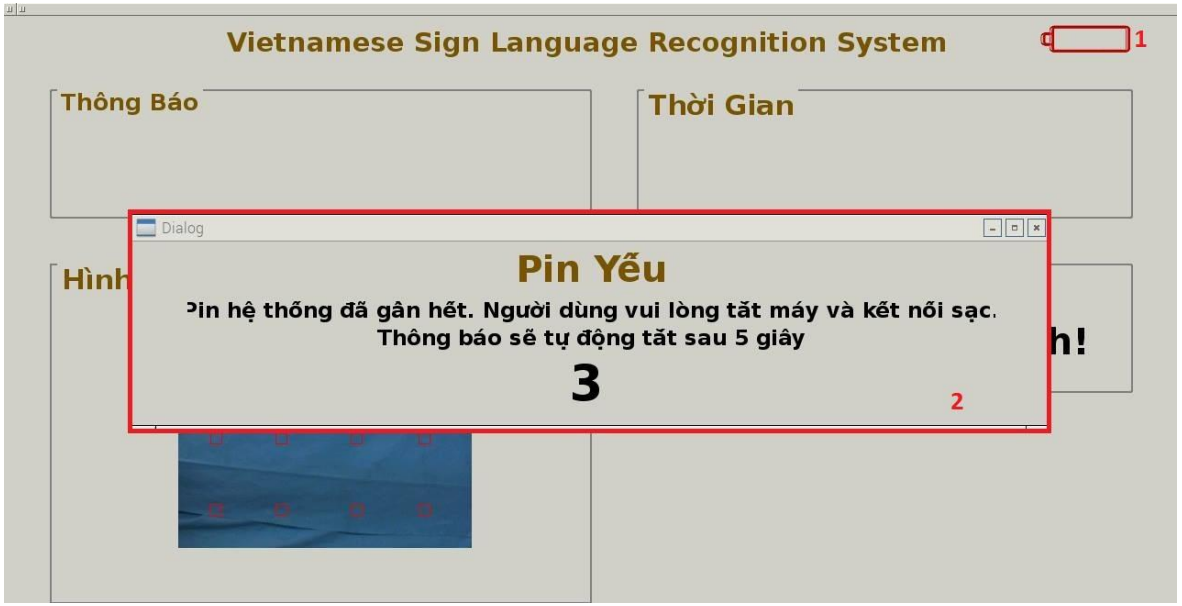


Figure 40: Notify Low Battery Dialog Interface

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	BatteryCapacity	Image describing current battery capacity	Yes	Yes	QLabel	QImage	N/A
2	NotifyDialog	Notify low battery	Yes	Yes	Dialog	QString	N/A

Table 47: Fields of Notify Low Battery Dialog Interface

5.2 Hardware Interface

Figure 41: Front side and above side of box

Figure 42: Right side of box

No.	Name	Purpose
1	LCD	Display interface of application
2	Camera	Capture image
3	LED	Balance light
4	Indicator LED	Display battery capacity
5	Charging jack	Charge battery
6	Switch	Turn off application
7	On/off Switch	Turn on/off power

Table 48: Elements of Hardware Interface

6. Database Design

6.1 Logical Diagram

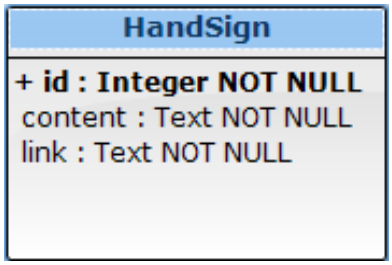


Figure 43: Logical Diagram

6.2 Data Dictionary

Entity name	Attributes	Description	Domain	Null
HandSign	id	Unique identifier of the hand sign.	Integer	No
	content	The translated content of the handsign	Text	No
	link	The image link leads to images describing the hand sign.	Text	No

Table 49: Data Dictionary

7. Algorithms

7.1 Background Subtraction

7.1.1 Definition

Background subtraction is the way that the system keeps only the hand on images captured from camera, and then the system can detect and keep track the hand through camera.

7.1.2 Define Problem

Images captured from camera that is quite complex in color and these maybe contains the user's hand or not so, we need to remove the colors not belong to the hand for detecting and recognizing.

7.1.3 Goal

- This solution must produce an image containing only the hands within black background and a binary image.

- Example:
  - Image containing the hands

Figure 44: The first goal of background subtraction algorithm

  - Binary Image containing contour

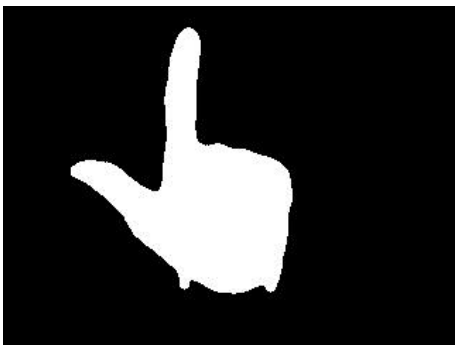


Figure 45: The second goal of background subtraction algorithm

7.1.4 Solution

To solve this problem, we need to have a background image as a sample, then we will remove the colors belongs to the sample from the following images captured from camera.

We should follow these steps to process background subtraction:

Step 1: Sampling background image

- Specify sample background color:
  - Capture an image as background sample.

Figure 46: Background image

- Blur background image to smooth and balance color.

Figure 47: Background image after blur

- Convert BGR background sample image to LAB sample image.

Figure 48: Background image after convert to Lab color space

- Get all color pixels from lab sample image.

- Define color range of LAB sample color:
  - With lightness dimension of LAB color space, the range is
    [Value – 40, value + 20]
  - With A and B for the color-opponent dimensions, the range is
    [Value – 15, value + 15]

Step 2: Process the following image to subtract background color

- Create a binary image with black background.
- Preprocess image
  - Blur image

Figure 49: Hand image after blur

- Convert to LAB image

Figure 50: Hand image after convert to Lab color space

- Check color pixels of the following images is between sample color range:
  - With every single color pixel, if lightness dimension (L) and color-opponent dimensions (A, B) is in color range of background that means this pixel does not belong to hand.
- Remove the color pixels is between color range:
  - Change all pixels not belong to hand to black color.

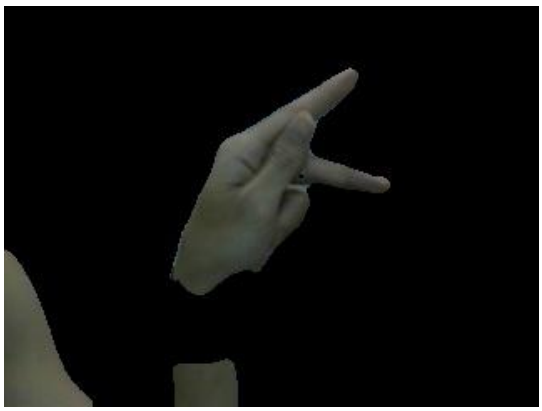


Figure 51: Hand image after subtract background

- Remove noises and smooth the contour:
  - After the previous step, we have a binary image containing some noises and the contour is not smooth so we need to remove them by morphology transformations.
- Find the hand in binary image:
  - Now we have a binary image containing not only hand but also arm or something else so we need to find out the hand by finding the biggest contour.

Figure 52: Image only contain hand

7.1.5 Definitions in image processing

7.1.5.1 RGB Image

RGB image is an image in which each color pixel is described based on the RGB color model. A color in the RGB color model is described by indicating how much of each of the red, green, and blue is included. The color is expressed as an RGB triplet (r, g, b), each component of which can vary from zero to a defined maximum value. If all the components are at zero the result is black; if all are at maximum, the result is the brightest representable white.

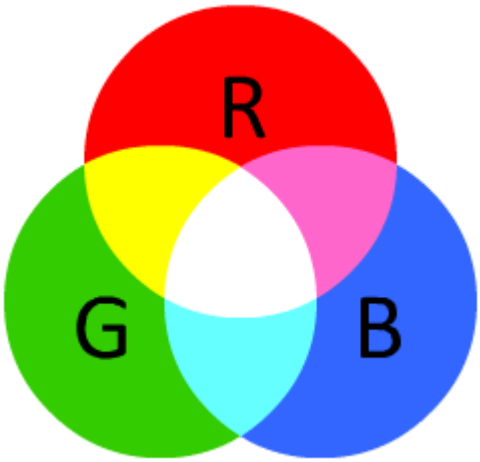


Figure 53 : Three spaces of RGB image

These ranges may be quantified in several different ways:

- From 0 to 1, with any fractional value in between. This representation is used in theoretical analyses, and in systems that use floating-point representations.
- Each color component value can also be written as a percentage, from 0% to 100%.
- In computers, the component values are often stored as integer numbers in the range 0 to 255, the range that a single 8-bit byte can offer. These are often represented as either decimal or hexadecimal numbers.
- High-end digital image equipment are often able to deal with larger integer ranges for each primary color, such as 0..1023 (10 bits), 0..65535 (16 bits) or even larger, by extending the 24-bits (three 8-bit values) to 32-bit, 48-bit, or 64-bit units (more or less

independent from the particular computer's word size).

For example, brightest saturated **red** is written in the different RGB notations as:

Notation	RGB triplet
Arithmetic	(1.0, 0.0, 0.0)
Percentage	(100%, 0%, 0%)
Digital 8-bit per channel	(255, 0, 0) or sometimes
#FF0000 (hexadecimal)	

Table 50: Brightest saturated red in the different RGB notations

7.1.5.2 LAB Image

LAB image is an image in which each color pixel is described based on the Lab color space model.
In this model, color space have one channel for Luminance (lightness) (L) and two color channels (a and b). The a axis extends from green (-a) to red (+a) and the b axis from blue (-b) to yellow (+b). The brightness (L) increases from the bottom to the top of the three-dimensional model.

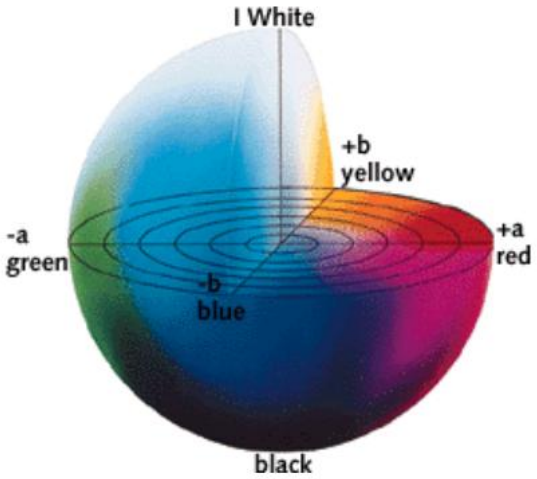


Figure 54: Three-dimensional model of LAB image

One of the most important attributes of the Lab image is device independence. This means that the colors are defined independent of their nature of creation or the device they are displayed on.

7.1.5.3 Binary Image

Binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a

binary image are black represented by 0 and white represented by 1, though any two colors can be used.

Figure 55: Example of binary image

Binary images are used in many applications because they are the simplest to process.

7.1.5.4 Contour

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity.

The contour pixels are generally a small subset of the total number of pixels representing a pattern. Therefore, the amount of computation is greatly reduced when we run feature extracting algorithms on the contour instead of on the whole pattern. The contours are a useful tool for shape analysis and object detection and recognition.

For example, below picture contains three contours, which is red color.

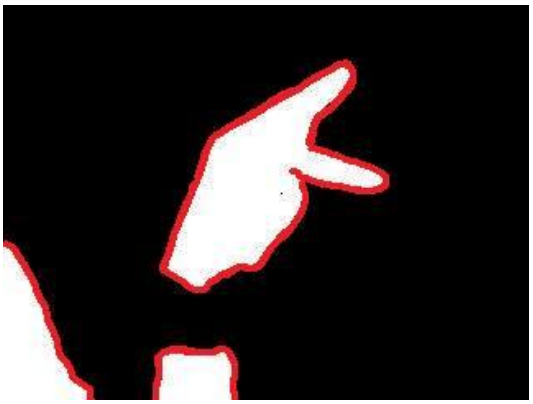


Figure 56: Examle of contour

7.1.6 Used Image Processing Algorithm

7.1.6.1 Blur Image – Gaussian Blur Algorithm

Gaussian Blur common mechanic

First, we choose a kernel size

- A kernel size is a matrix with width and height can differ but they must be positive and odd.
- Why kernel width and height both must be odd? Because we can get a center location in the matrix.
- For example:

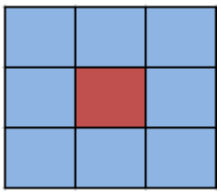


Figure 57: kernel size 3x3

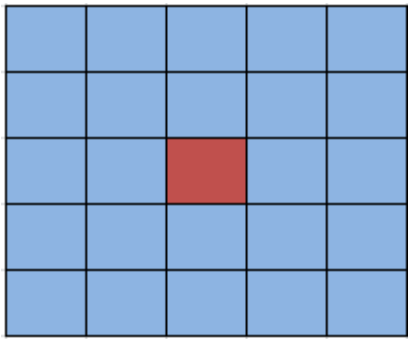


Figure 58: kernel size 5x5

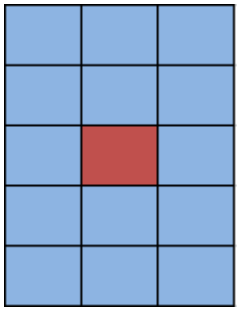


Figure 59: kernel size 3x5

Move the kernel size on the image matrix all over pixels. At single movement, the kernel's center point will take the average value of its surrounding points.

- Movements

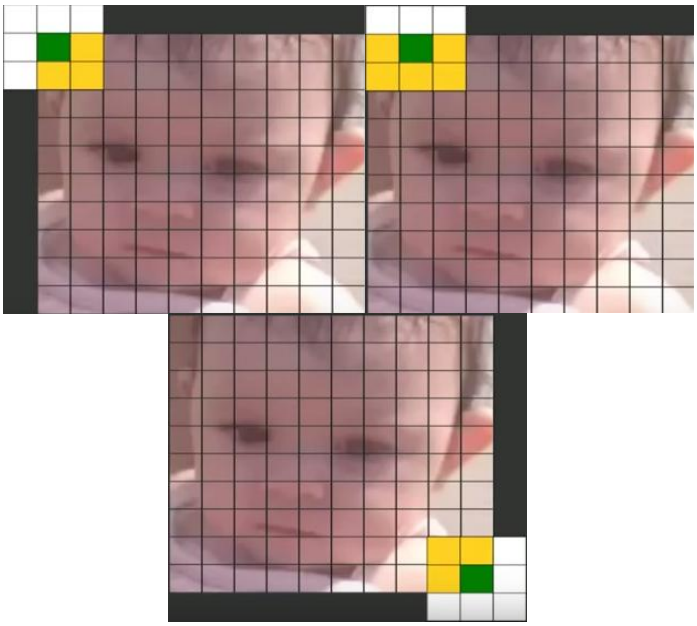


Figure 60: Kernel size movements

- At single movement

Assume we have a value matrix taken from the kernel size 3x3

1	1	1
1	2	1
1	1	1

Figure 61: Value matrix taken from the kernel size 3x3

On the above matrix, the center point value is 2 and its surrounding points are 1. The center point will take the average value of its neighbor, it will be 1.

1	1	1
1	1	1
1	1	1

Figure 62: Value matrix after applying Gaussian blur algorithm

The kernel size is large, the blur effect is strong.

Figure 63: Original RGB Image

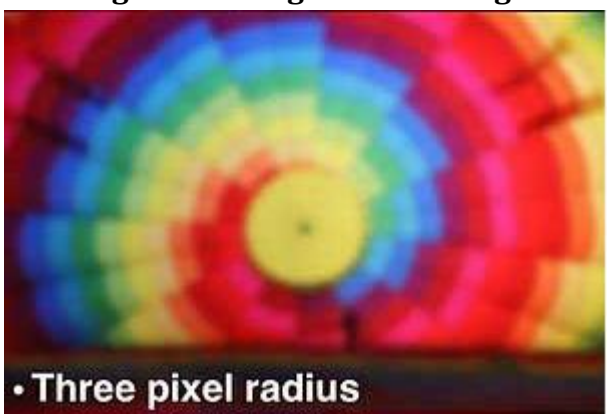


Figure 64: RGB Image applied Guassian Blur with kernel size 7x7

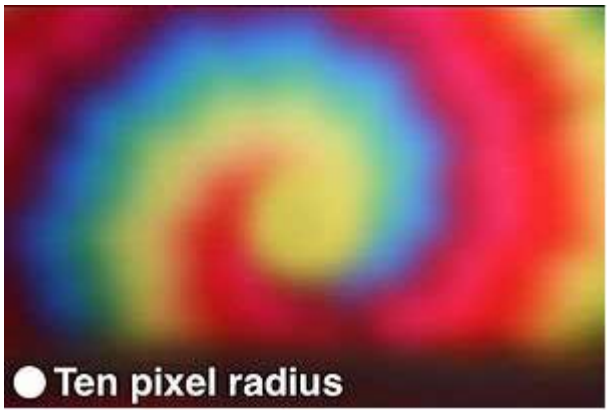


Figure 65: RGB Image applied Guassian Blur with kernel size 21x21

Weighted Matrix

To preserve boundaries and and edges better, we have a rule, the closer the points in distance, the larger the weight. In other words, the original's pixel must receive the heaviest weight and its neighboring pixels receive smaller weights as their distance increases.

Gaussian provides a formula to produce weighted matrix:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-x^2/2\sigma^2}$$

where x, y is the distance from the center point in x-axis and y-axis respectively, and σ is standard deviation.
Standard deviation is computed from size of kernel size as sigma = 0.3 * ((ksize - 1) * 0.5 -1) + 0.8 or we can choose a specific standard deviation.
For example

- We will calculate the weight matrix 3x3 with standard deviation = 1.5
  - The coordinate of the center point

-1,1	0,1	1,1
-1,0	0,0	1,0
-1,-1	0,-1	1,-1

Figure 66: the coordinate of the center point with matrix 3x3

  - Apply the above formula for each point

0.0453542	0.0566406	0.0453542
0.0566406	0.0707355	0.0566406
0.0453542	0.0566406	0.0453542

Figure 67: the weighted 3x3 matrix before divide the sum of the weight

  - The above 9 values will divide the sum of the weight of these 9 points which is 0.4787147

0.0947416	0.118318	0.0947416
0.118318	0.147761	0.118318
0.0947416	0.118318	0.0947416

Figure 68: the final weighted 3x3 matrix

- Apply this weight matrix to calculate the gaussian blur
  - Assume we have a matrix value (3x3) of grayscale image

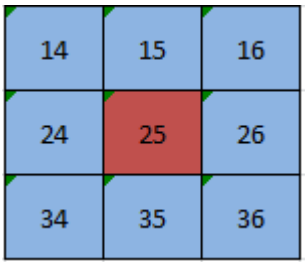


Figure 69: The 3x3-matrix value of grayscale image

- Each point multiplies its weight value (above weighted matrix)

14x0.0947416	15x0.118318	16x0.0947416
24x0.118318	25x0.147761	26x0.118318
34x0.0947416	35x0.118318	36x0.0947416

Figure 70: Multiplying the matrix value with the weighted matrix

- The final result

1.32638	1.77477	1.51587
2.83963	3.69403	3.07627
3.22121	4.14113	3.4107

Figure 71: Matrix values after applying Gaussian blur

7.1.6.2 Convert RGB image to LAB image

In case of 8-bit and 16-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range.RGB values in a particular set of primaries can be transformed to CIE via a 3x3-matrix transform

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

CIE LAB is based directly on CIE XYZ and is an attempt to linear the perceptibility of color differences. The non-linear relations for L*, a*, and b* are intended to mimic the logarithmic response of the eye. Coloring information

is referred to the color of the white point of the system, subscript n

$$X \leftarrow X/X_n, \text{where} X_n = 0.950456$$
$$Z \leftarrow Z/Z_n, \text{where} Z_n = 1.088754$$
$$L \leftarrow \begin{cases} 116 * Y^{1/3} - 16 & \text{for } Y > 0.008856 \\ 903.3 * Y & \text{for } Y \leq 0.008856 \end{cases}$$
$$a \leftarrow 500(f(X) - f(Y)) + delta$$
$$b \leftarrow 200(f(Y) - f(Z)) + delta$$
$$f(t) = \begin{cases} t^{1/3} & \text{for } t > 0.008856 \\ 7.787t + 16/116 & \text{for } t \leq 0.008856 \end{cases}$$
$$delta = \begin{cases} 128 & \text{for 8-bit images} \\ 0 & \text{for floating-point images} \end{cases}$$

This outputs $0 \leq L \leq 100, -127 \leq a \leq 127, -127 \leq b \leq 127$. The values are then converted to the destination data type 8-bit images

$$L \leftarrow L * 255/100, \quad a \leftarrow a + 128, \quad b \leftarrow b + 128$$

Example:

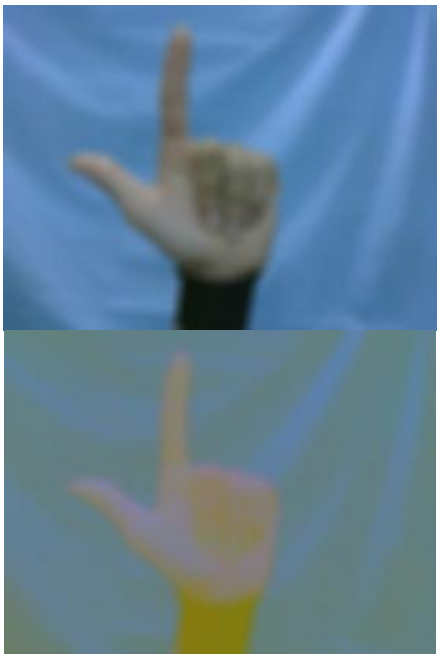


Figure 72: Example of converting RGB image to LAB image

7.1.6.3 Morphological operations - Opening on binary images

Morphological operations are affecting the form, structure or shape of an object. A morphological operation on a binary image creates a new binary image

in which the pixel has a non-zero value only if the test is successful at that location in the input image.

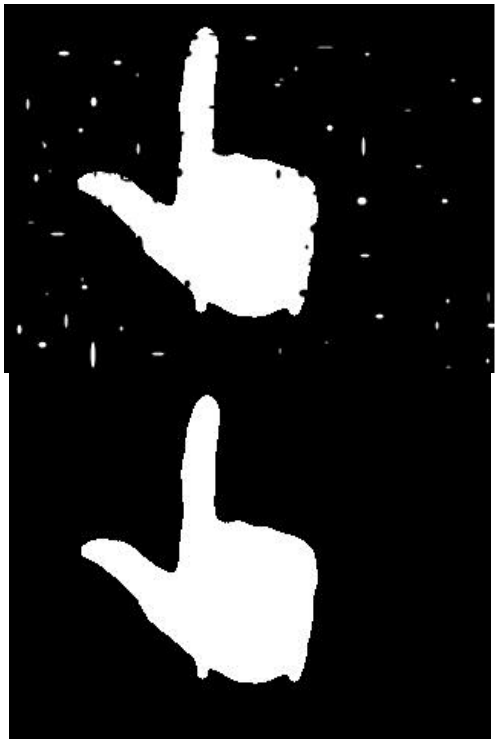


Figure 73: Example of Morphological operations – Opening

Morphological operations have 2 principal: *dilation* and *erosion.* These operations can be customized by the selection of the *structuring element*, which determines exactly small binary image will be dilated or eroded. Structuring element can be a small matrix of pixels, each with a value of zero or one:

- The matrix dimensions specify the *size* of the structuring element.
- The pattern of ones and zeros specifies the *shape* of the structuring element.
- An *origin* of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

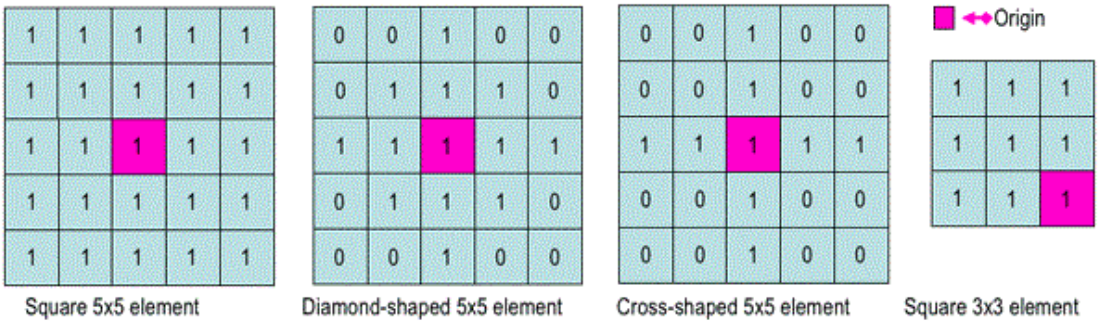


Figure 74: Some structuring element

Dilation allows objects to expand, thus potentially filling in small holes and connecting disjoint objects. The dilation process is performed by laying the structuring element B on the image A **(Notation A⊕ B*)*** and move it across each pixel of binary image. When move to a pixel, this pixel is origin. Dilation operation will perform:

1. If the origin of the structuring element coincides with a 'white' pixel in the image, there is no change; move to the next pixel.

2. If the origin of the structuring element coincides with a 'black' in the image, make black all pixels from the image covered by the structuring element.

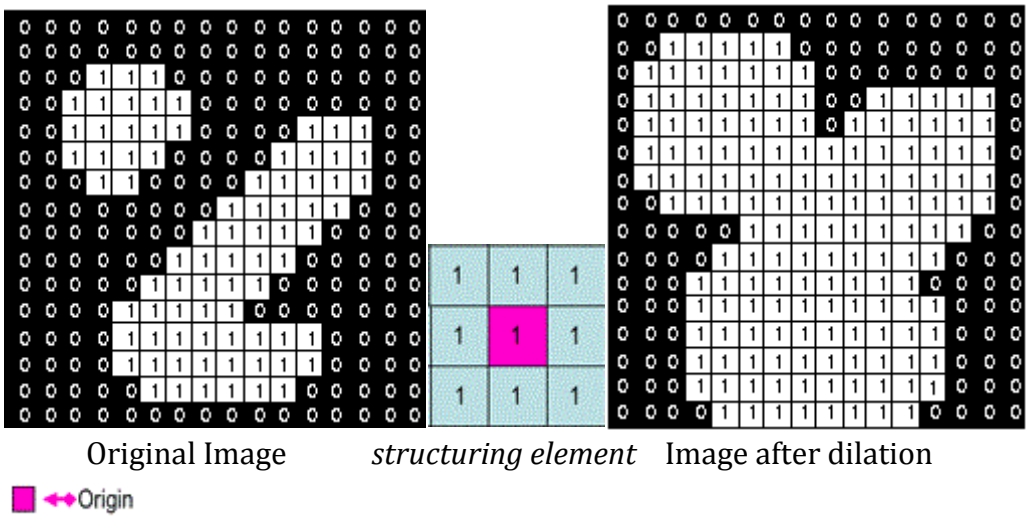
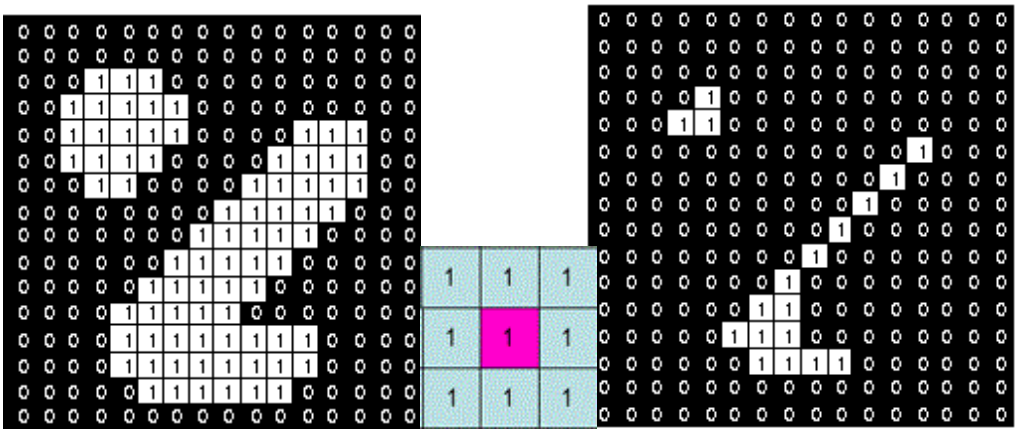


Figure 75: Illustration of the dilation process

Erosion shrinks objects by etching away (eroding) their boundaries. The *erosion* process is similar to dilation **(Notation *A* Θ *B*)** but it performs these steps:

1. If the origin of the structuring element coincides with a 'white' pixel in the image, there is no change; move to the next pixel.

2. If the origin of the structuring element coincides with a 'black' pixel in the image, and at least one of the 'black' pixels in the structuring element falls over a white pixel in the image, then change the 'black' pixel in the image (corresponding to the position on which the center of the structuring element falls) from 'black' to a 'white'.



Original Image *structuring element* Image after Erosion

Figure 76: Illustration of the erosion process

These two basic operations, dilation and erosion, can be combined into more complex sequences. *Opening* consists of an erosion followed by a dilation and can be used to eliminate all pixels in regions that are too small to contain the structuring element. **Notation: $A \circ B = (A \ominus B) \oplus B$**

7.1.6.4 Find Contour

Apply Theo Pavlidis' Algorithm:

Given a image digital have group of black pixels, on a background of white pixels locate a black pixel and declare it as your "start" pixel.

You can choose any black boundary pixel to be your start pixel as long as when you are initially standing on it, your left adjacent pixel is NOT black. In other words, you should make sure that you enter the start pixel in a direction which ensures that the left adjacent pixel to it will be white ("left" here is taken with respect to the direction in which you enter the start pixel).

We have are a bug (ladybird) standing on the **start** pixel as in Figure below.

Throughout the algorithm, the pixels, which interest you at any time, are the 3 pixels in front of you i.e. **P1, P2** and **P3** shown in ***Figure 77***. (We will define **P2** to be the pixel right **in front** of you, **P1** is the pixel adjacent to **P2** from the left and **P3** is the right adjacent pixel to **P2**).

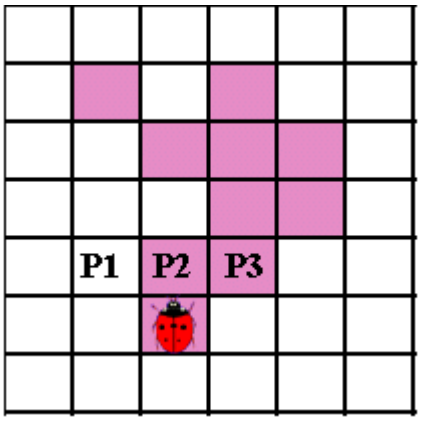


Figure 77: Starting pixel of find contour

The most important thing in Pavlidis' algorithm is your "sense of direction". The left and right turns you make are with respect to your current positioning, which depends on the way you entered the pixel you are standing on. In example , we choose direction from the bottom up which you enter the **start** pixel.

First, check pixel P1. If P1 is black, then declare P1 to be your current boundary pixel and move one step forward followed by one step to your current left to land on P1. (The order in which you make your moves is very important)

Figure 78 below demonstrates this case. The path you should follow in order to land on P1 is drawn in blue.

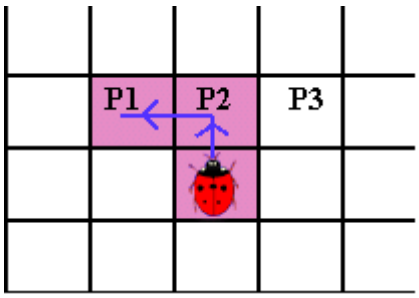


Figure 78: Demonstrating the path to P1

Only if P1 is white proceed to check P2...

If P2 is black, then declare **P2** to be your current boundary pixel and **move one-step forward** to land on **P2**.

Figure 79 below demonstrates this case. The path you should follow in order to land in **P2** is drawn in blue.

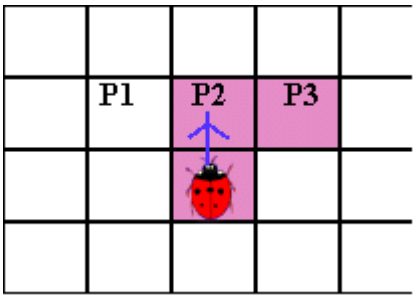


Figure 79: demonstrating the path to P2

Only if both P1 and P2 are white proceed to check P3...
If **P3** is black, then declare **P3** to be your current boundary pixel and move one-step to your right followed by one-step to your current left as demonstrated in Figure 80 below

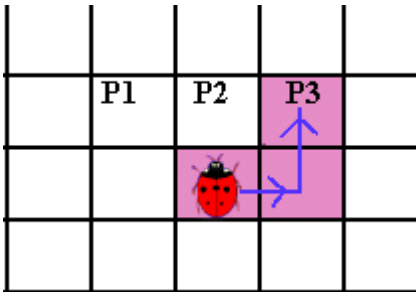


Figure 80: Demonstrating the path to P3

If all 3 pixels in front of you are white: Then, you rotate (while standing on the current boundary pixel) 90 degrees clockwise to face a new set of 3 pixels in front of you. Afterwards you do the same check on these new pixels as you have done before.
If all of these 3 pixels are still white: then rotate again through 90 degrees clockwise while standing on the same pixel.
You can rotate 3 times (each through 90 degrees clockwise) before checking out the whole Moore neighborhood of the pixel. If you rotate 3 times without finding any black pixels, this means that you are standing on an isolated pixel i.e. not connected to any other black pixel.
The algorithm terminates when:

- As mentioned above, the algorithm will allow you to rotate 3 times (each through 90 degrees clockwise) after which it will terminate and declare the pixel an isolated one, OR
- It visiting the start pixel a third time

7.1.7 Disadvantages

- This algorithm requires the background is not changed after sampling background color and the light must be stable because if background changes or light changes, it makes background color is out of sample range, which is defined after sampling.
- This algorithm just works correctly if the background color is not close to hand skin color.
- Moreover, this algorithm requires users to use accessory such as black bracelet to separate hand and arm.

7.1.8 Flow chart

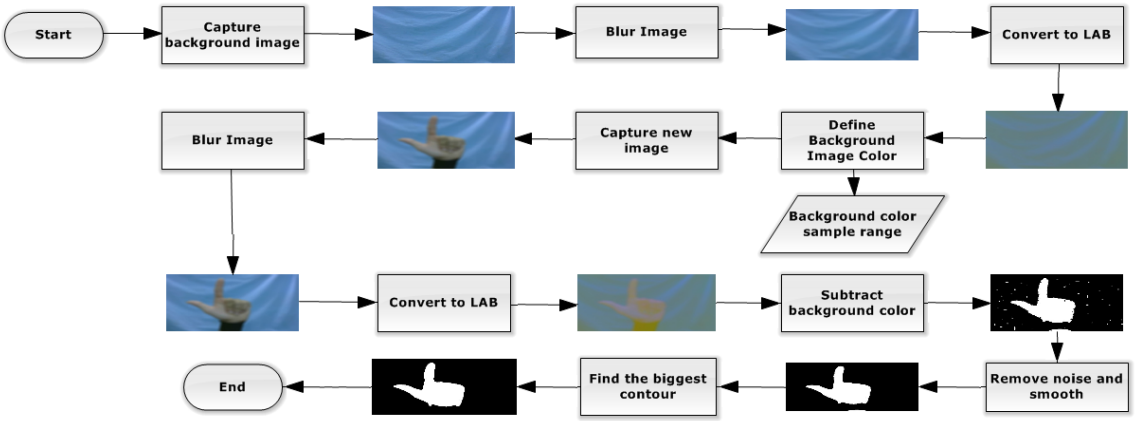


Figure 81: Flow of background subtraction algorithm

7.2 Features Extraction

7.2.1 Definition

Features extraction is the way to specify 81 features of hand for recognition.

7.2.2 Define Problem

At this time, we just have an image containing only hand and a binary image containing the hand contour and we need to find the best features from these two images.

7.2.3 Goal

- This solution must produce three binary images containing 81 features for hand sign recognition.
- Example:

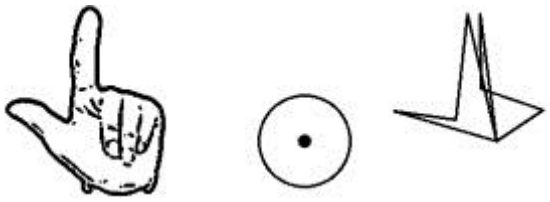


Figure 82: The goal of features extraction

7.2.4 Solution

To solve this problem, we need to convert the BGR image containing hand, which is produced from the background color subtraction into binary images, which can describe most of features.

We should follow the following steps to extract the features of hand:

- Convert the BGR image containing the hand to binary image containing features:
  - To process it we will apply the Adaptive Gaussian Threshold algorithm.

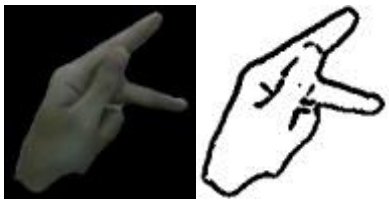


Figure 83: Example BGR image and binary image

- Produce the binary image containing the hand palm from the binary image containing contour:
  - We will find the point inside the contour and calculate the nearest distance from the point to the contour, and then we find the farthest point from those nearest distance.
  - From the farthest point and the distance, we can draw a circle similar to the hand palm

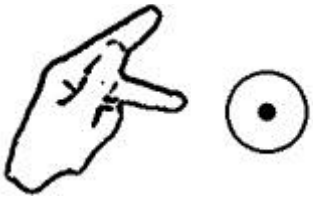


Figure 84: Example hand image and the hand palm

- Produce the binary image containing the finger lines from the binary image containing contour:
  - First, we find the convex hull of the contour
  - After that, we find the convexity defects of the contour from the convex hull.
  - From these convexity defects we can draw the finger lines

Figure 85: Example hand image and the finger lines

- Calculate the features from these binary images:
  - Calculate zoning features

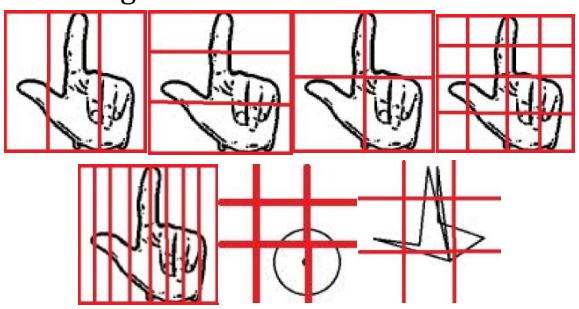


Figure 86: Zoning features

  - Calculate the ratio between height and width of hand

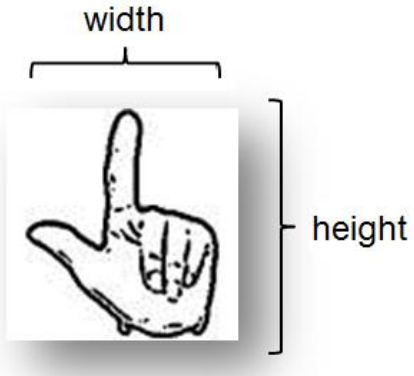


Figure 87: height and width of hand

  - Calculate the ratio between radius of palm and size of hand

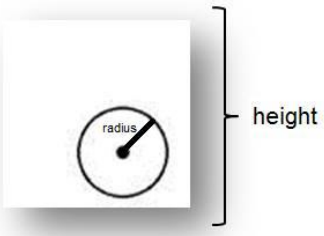


Figure 88: Radius of palm and size of hand

  - Calculate the degree features of hand fingers

7.2.5 Definitions in image processing

7.2.5.1 Convex Hull

Convex is a shape or set if for any two points that are part of the shape, the whole connecting line segment is also part of the shape. For any subset of the plane (set of points, rectangle, simple polygon), its convex hull is the smallest convex set that contains that subset

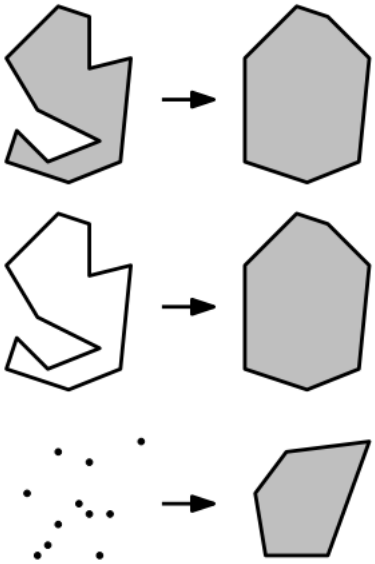


Figure 89: Example of convex hull

7.2.5.2 Zoning features

Zoning features are efficient statistical features that provide low complexity for different characters of an object. They are defined by the density of black pixels in several zones we divided in an image-containing object. Zoning features is good that depends on zones we selected to calculate the density of black pixels.
For an example of 4x4 zones:

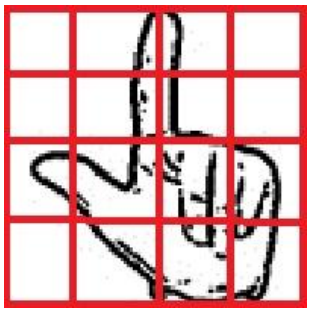


Figure 90: Example of 4x4 zones

7.2.6 Used Image Processing Algorithm

7.2.6.1 Convert to binary image - Adaptive Gaussian Threshold

This algorithm applies an adaptive threshold to a matrix for transforming a grayscale image to binary image.
According to the formulae:
Where x, y is the position of a pixel in the matrix and T(x,y) is a threshold calculated individually for every single pixel.

A transformation is good that depends on how the threshold is calculated.
We apply the method using Weight Matrix (explained in Gaussian Blur Algorithm), the threshold is a weighted sum of the MATRIX_SIZE x MATRIX_SIZE neighboring pixel of the origin pixel.
This algorithm is a useful technique to draw features of images. For example, we have an RGB image containing hand with black background.

Figure 91: RGB image containing hand with black background

After image processing, we have a binary image containing features of the hand.

Figure 92: binary image after converting to binary image by adaptive gaussian

7.2.6.2 Find the convex hull

Implement Melkman's algorithm 1987
It takes the first three vertices (starting anywhere) and sets up the current convex hull, i.e. the triangle formed. In the picture below, they happen to form a right turn (clockwise). Vertices are stored in a double ended queue (deque) : (bottom) 3-1-2-3 (top). Notice that if we take the last three vertices of the bottom in the order 2,1,3 we get a left turn. If we take the last three vertices from the top, in the order 1,2,3 we get a right turn. This is a property that we wish to maintain: in general as we read the deque from bottom to top, we get the hull in clockwise order, and as we read from top to bottom we get a counterclockwise order.
Now the next vertex, V, could be in the red/green/blue/yellow regions

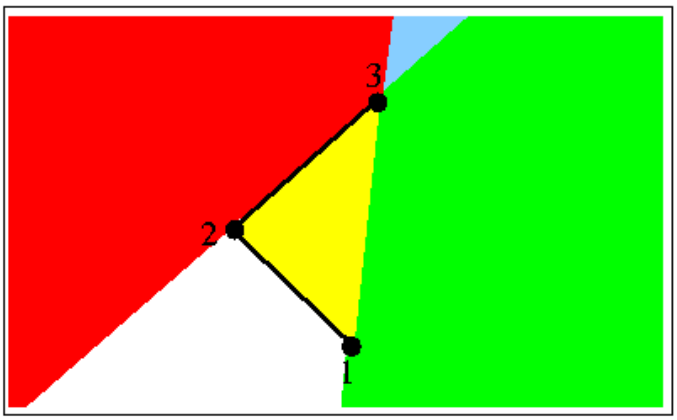


Figure 93: Example of the steps finding convex hull

If V is in the yellow region, ignore it and all following vertices until one emerges into the other regions. Call the emerging vertex V. If V is not yellow, we must add it to the deque on both sides, because it will be on the current hull. However we must ensure that we preserve our clockwise/counterclockwise property if this is to be done: If V is in the red region, then 2, 3, V form a left turn. Backtrack/delete vertices from the top of the deque (i.e. 3, then maybe 2, etc), until a right turn is formed by the last 3 vertices.

If V is in the green region, then 1, 3, V form a right turn. Backtrack/delete vertices from the bottom of the deque (i.e. 3, then maybe 1, etc), until a left turn is formed by the last 3 vertices. Notice that this case is symmetric to the red region.

If blue, follow the instructions for both red and green.

Now the deque structure is correct and we can process the next point.

The figure below shows these regions in a more general case. Vertices on the hull have circles on them. (N) is the last vertex added. The next vertex could be anywhere in the colored regions. Note that you can find these regions by looking at N, and its neighbors on the hull, which are conveniently represented at the two ends of the deque

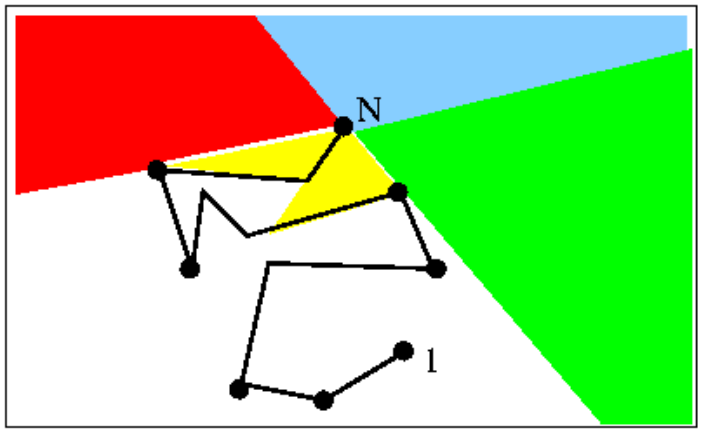


Figure 94: Example of the steps finding convex hull

7.2.6.3 Find the convexity defects

After find convex hull of shape, we have blue points as result.

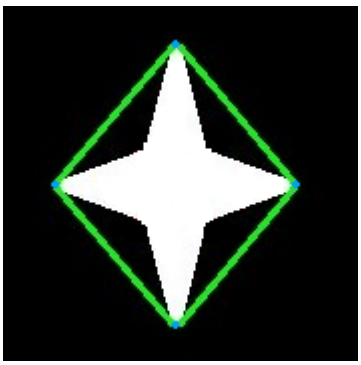


Figure 95: Example of finding the convexity defects

Then, we can find the convexity defects (red point) of this shape by:

Move each point (red point) on contour of shape and measure distance from this point to green line connect two blue points.

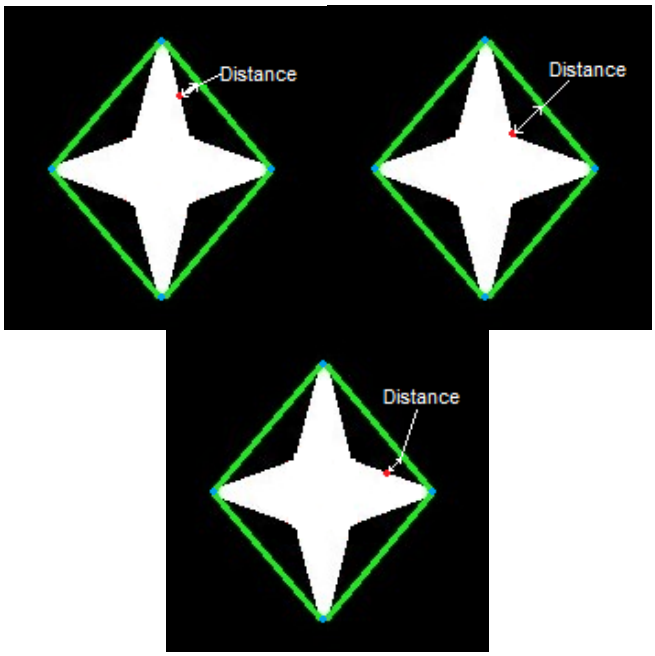


Figure 96: Example of finding the convexity defects

The convexity defect is the farthest point to green line. With two blue points, we just find out one convexity defect point.

7.2.6.4 Resize Image - Nearest Neighbor Image Scaling

The principle in image scaling is to have a reference image and using this image as the base to construct a new-scaled image. The constructed image will be smaller, larger, or equal in size depending on the scaling ratio.

We say: w1 and h1 are the width and height of an origin image, whereas w2 and h2 are the width and height of new image. Calculating the ratio for both horizontal and vertical plane is given by,

$$\left.\begin{aligned} x\_ratio &= \frac{w_1}{w_2} \\ y\_ratio &= \frac{h_1}{h_2} \end{aligned}\right\} w_2, h_2 \neq 0$$

In new image, value of each pixel Y(x_2, y_2) determined by the value of the pixel X(x_1, y_1) where :

$$\begin{cases} x_1 = rounded\ down\ of\ x_2 \times x\_ratio \\ y_1 = rounded\ down\ of\ y_2 \times y\_ratio \end{cases}$$

1	0	1
1	0	1
0	1	0

Origin image 3X3

1	1	0	0	1	1
1	1	0	0	1	1
1	1	0	0	1	1
1	1	0	0	1	1
0	0	1	1	0	0
0	0	1	1	0	0

new image 6X6

Figure 79: Illustration of the resizing 200%

Original image

Scale 200%

Figure 97: Example of Image Scaling

7.2.6.5 Calculate the degree features of hand fingers

The goal of this algorithm is to define the degree category of each finger line. The degree category is the degree between the line and the x-axis and we will have eight categories:

- Category 1: The degree is between 0 and 45
- Category 2: The degree is between 46 and 90
- Category 3: The degree is between 91 and 135
- Category 4: The degree is between 136 and 180

- Category 5: The degree is between -45 and -1
- Category 6: The degree is between -90 and -46
- Category 7: The degree is between -135 and -91
- Category 8: The degree is between -179 and -136

For example, below picture we have four lines:

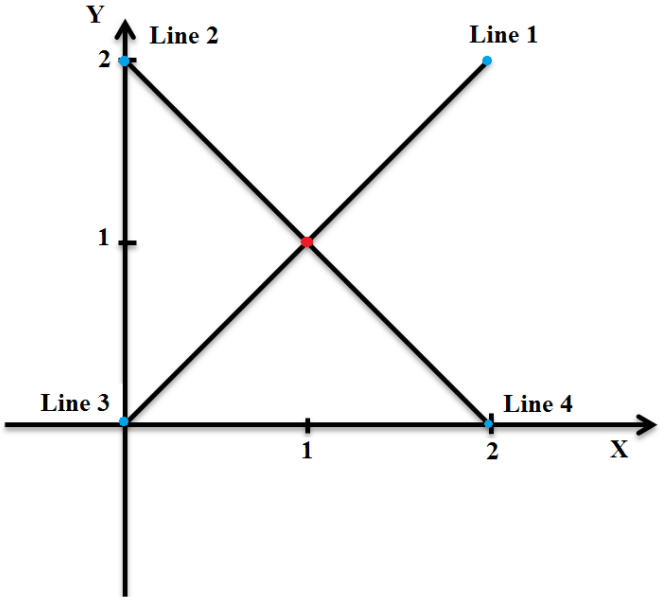


Figure 98: The example of degree features extraction

Where the red point is the origin point and the blue point is the distance point.

Then, calculate the degree and classify them.

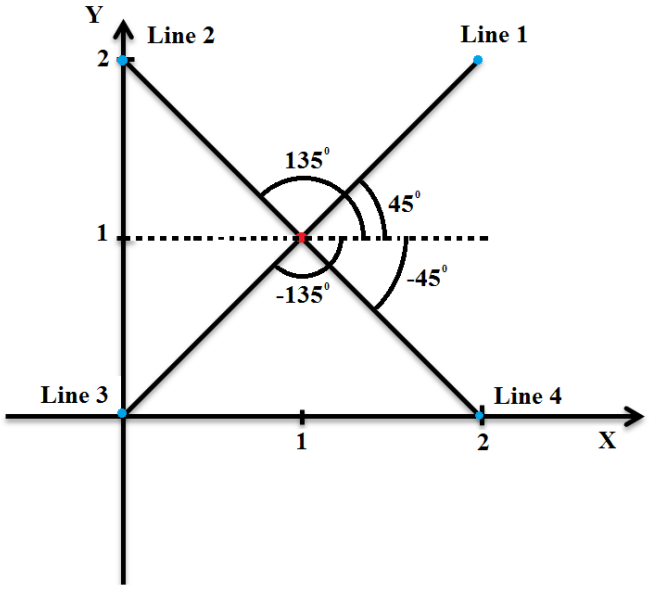


Figure 99: The example of degree features extraction

After classify the line into these eight categories, the feature is the number of lines of each category.

7.2.7 Flow chart

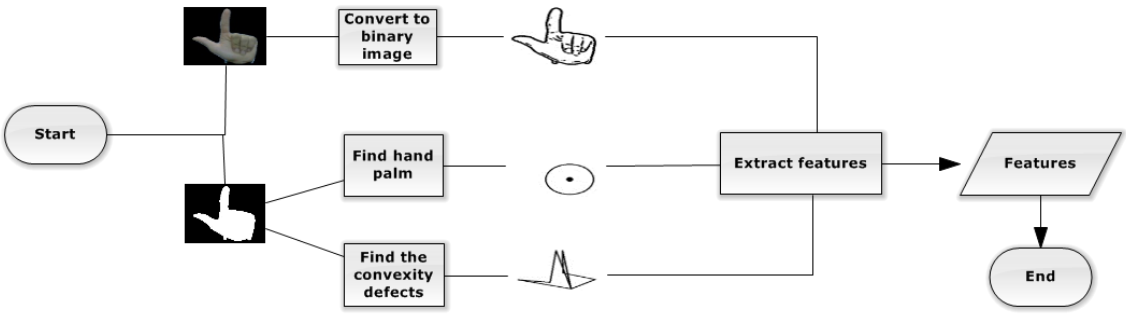


Figure 100: The flow of features extraction algorithm

7.3 Support Vector Machine

7.3.1 Definition

Support Vector Machine is a useful technique for data classification that analyzes data and recognizes patterns.

7.3.2 Define problem

Now we have sets of features of the hand signs and we must rely on those sets to recognize them. Therefore, we use library for Support Vector Machine (LIBSVM) to produce a model by basing on the training hand sign feature data and then rely on the model to predict the target values of the other hand sign features.

7.3.3 Goal

This library for Support Vector Machine algorithm builds a model that assigns new examples into specify category and when new hand sign are mapped into that same space and predicted to belong to a category.

7.3.4 Support Vector Machine Algorithm

A Support Vector Machine is a discriminative classifier formally defined by constructing set of hyperplanes in a space has high dimension or infinite dimension. Furthermore, the algorithm should output a hyperplane, which is optimal to categorize new examples. An optimal hyperplane is a hyperplane has the largest distance to the nearest training data point of classes.

For example:

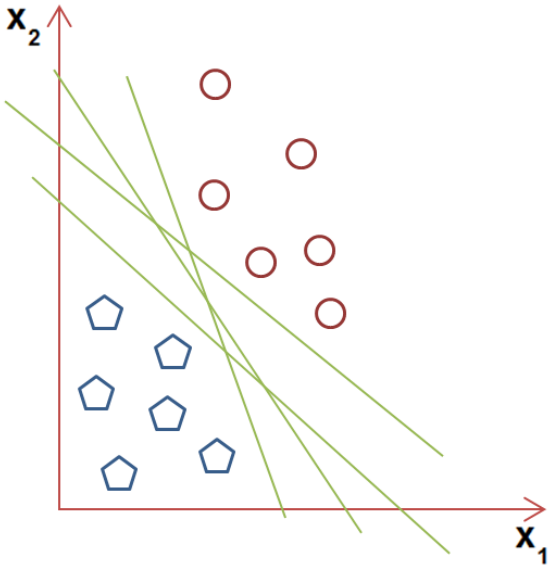


Figure 101: Example of linear SVM technique

- In the above picture, we see that there are many lines can resolve the problem but which is better than others are.
- The algorithm defines a criterion to estimate the worth of these lines that is the one that represents the largest separation between two set of points, so we choose the optimal separating hyperplane maximizes the distance from it to the nearest data point on each side.

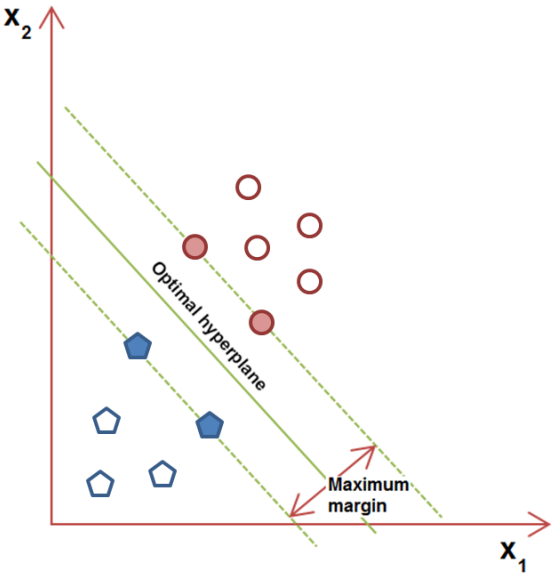


Figure 102: Example of linear SVM technique

The above example is one technique of Linear SVM. Besides that, there is another technique supported by SVM that is nonlinear

classification. This is the way to create nonlinear hyperplanes by applying the kernel trick to margin hyperplanes.
We chose the nonlinear SVM technique because it can map samples into a higher dimensional space and the number of features and sample is quite small. The following image is an example of nonlinear SVM technique:

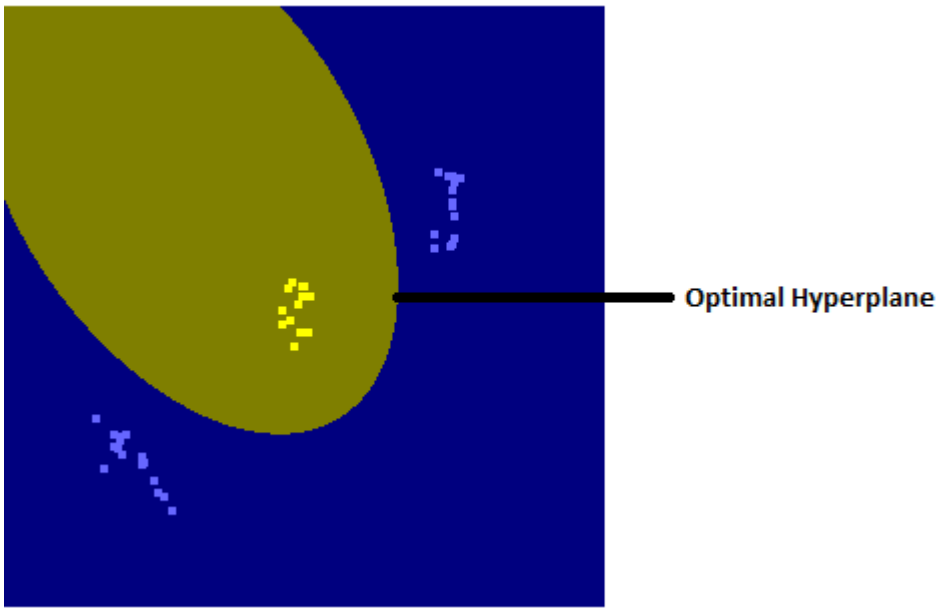


Figure 103: Example of non-linear SVM technique

With multiclass, SVM algorithm aims a common method that is reducing the single multiclass problem into binary classifiers which separate between one class and the remains. That is called one-versus-all. The following example:

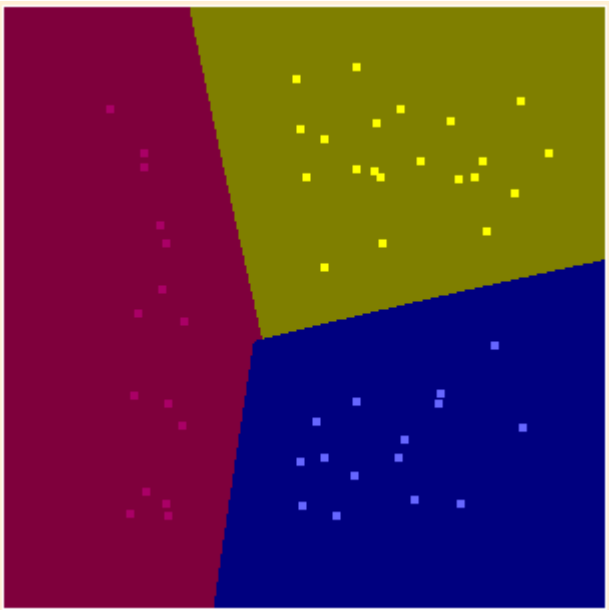


Figure 104: Example of non-linear SVM technique

Besides that, to have a good SVM model that relies on the number of sample, how many cases the samples can cover and the most important condition is the features we select.

7.4 LIPO Battery Capacity Calculation

7.4.1 Definition

The system uses LIPO battery to supply power for the system can work as a portable system. To satisfy the criterion of a portable system we need to notify user of the remaining capacity of battery.

7.4.2 Define problem

According to measurement of LIPO battery capacity in the real environment, the range of capacity LIPO battery can supply the system works correctly with power that is between 9.9V and 12.2V. Therefor, we need to find out the way can compute the current voltage of LIPO battery.

7.4.3 Goal

Reconstruct a circuit can compute five levels of voltage such as the voltage higher than 12V, the voltage lower than or equal 12V and higher than 11.3V, the voltage lower than or equal 11.3V and higher than 10.8V, the voltage lower than or equal 10.8V and higher than 9.9V, the voltage lower than 9.9V

7.4.4 Solution

As we, all know, to measure battery capacity battery, we need to rely on its voltage. Over the course of research, we will use 5.1V zener diode and chip TL084CN, these are 2 the main component of the circuit. 5.1V zener diode voltages used as a benchmark to compare the voltage of the battery. Chip TL084CN is responsible for comparing the voltage with the voltage zener diodes standard. Besides, we will use resistor to form a potentiometer circuit.

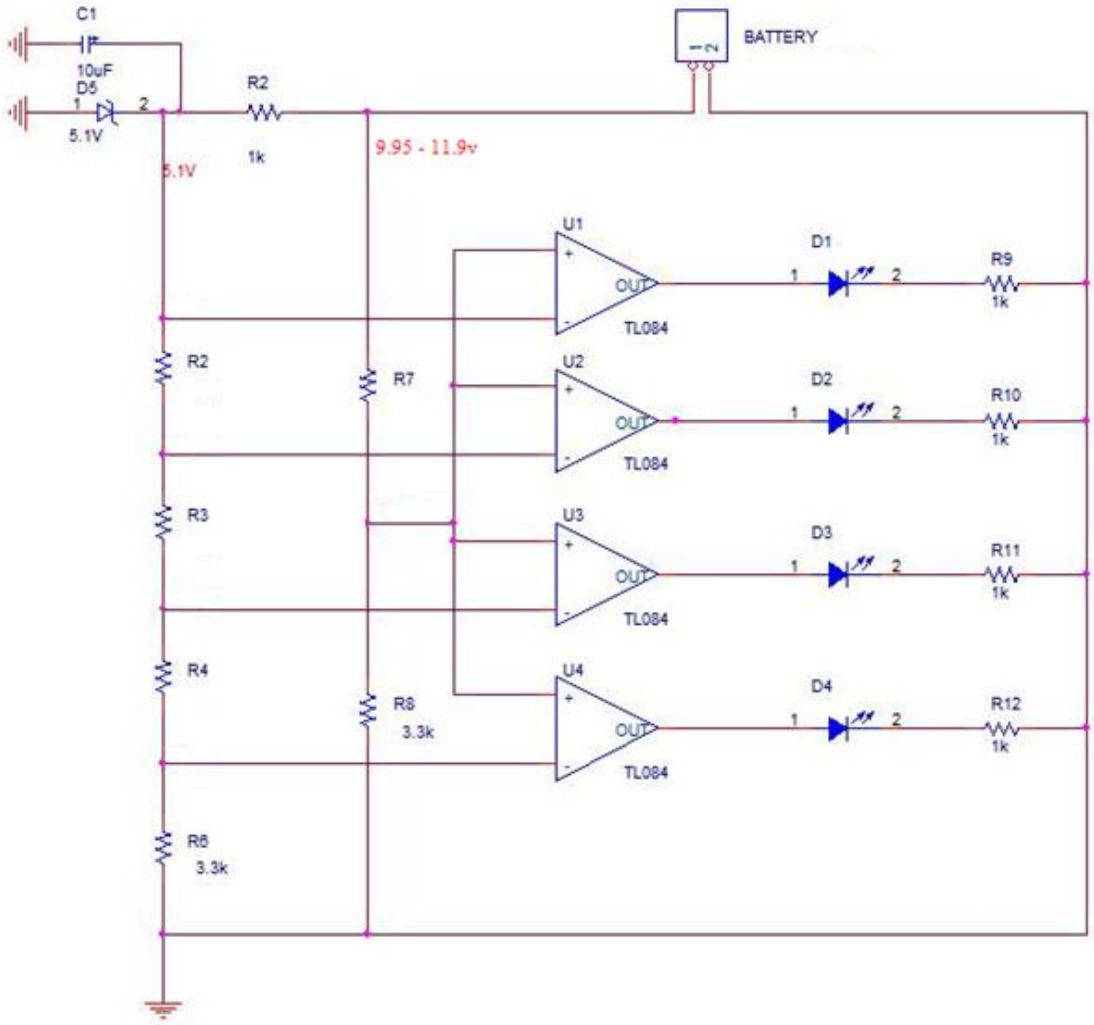


Figure 105: Principle of Monitor Battery Capacity

Because we have high input, voltage of the zener is 5.1, when the battery voltage is highest 11.9v and we have the formula is:

$$\frac{U_8}{R_8} = \frac{U_7}{R_7} = \frac{U_{battery}}{R_7 + R_8}$$

$$=> U_8 = \frac{U_{battery} \times R_8}{R_7 + R_8}; U_8 = 5.1, U_{Battery\,max} = 11.9, R_8 = 3300$$

$$=> R_7 = 4400$$

When the battery voltage is lowest 9.95V, we have U_8 is:

$$U_8 = \frac{U_{battery} \times R_8}{R_7 + R_8}, U_{Battery\,min} = 9{,}95$$

$$=> U_8 = 4{,}26\ (V)$$

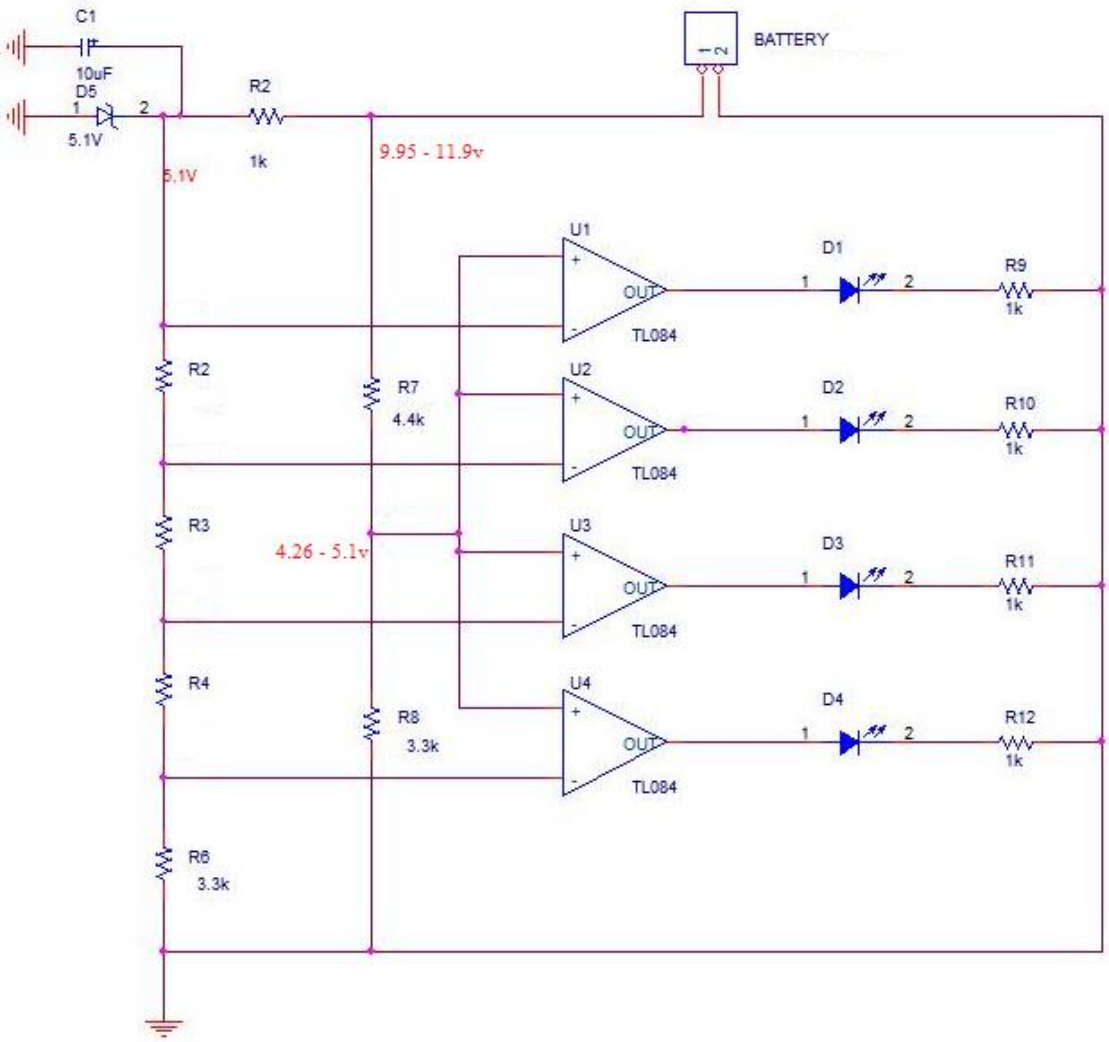


Figure 106: Calculator U_8 and R_7

Because the lowest voltage of U_8 is 4,26V, so we have the voltage on U_6 must of 4.25 to the battery expires, the leds are off.

$$U_6 = \frac{U_{zener} \times R_6}{R_6 + (R_2 + R_3 + R_4)}; U_6 = 4{,}25, U_{zener} = 5{,}1$$

$$=> R_2 + R_3 + R_4 = 660\ (ohm)$$

$$=> R_2 = R_3 = R_4 = \frac{660}{3} = 220\ (ohm)$$

Thus, we have:

$$U_2 = \frac{U_{zener} \times R_2}{R_6 + (R_2 + R_3 + R_4)}$$

$$=> U_2 = 0.2833\ (V)$$

$$=> U_{346} = 4.28(V)$$

$$U_3 = \frac{U_{zener} \times R_3}{R_6 + (R_2 + R_3 + R_4)}$$

$$=> U_3 = 0.2833\ (V)$$

$$=> U_{46} = 4.53(V)$$

$$U_4 = \frac{U_{zener} \times R_4}{R_6 + (R_2 + R_3 + R_4)}$$

$$=> U_4 = 0.2833\ (V)$$

$$=> U_6 = 4.25(V)$$

Finally, we get a complete circuit Battery Capacity Display Circuit.

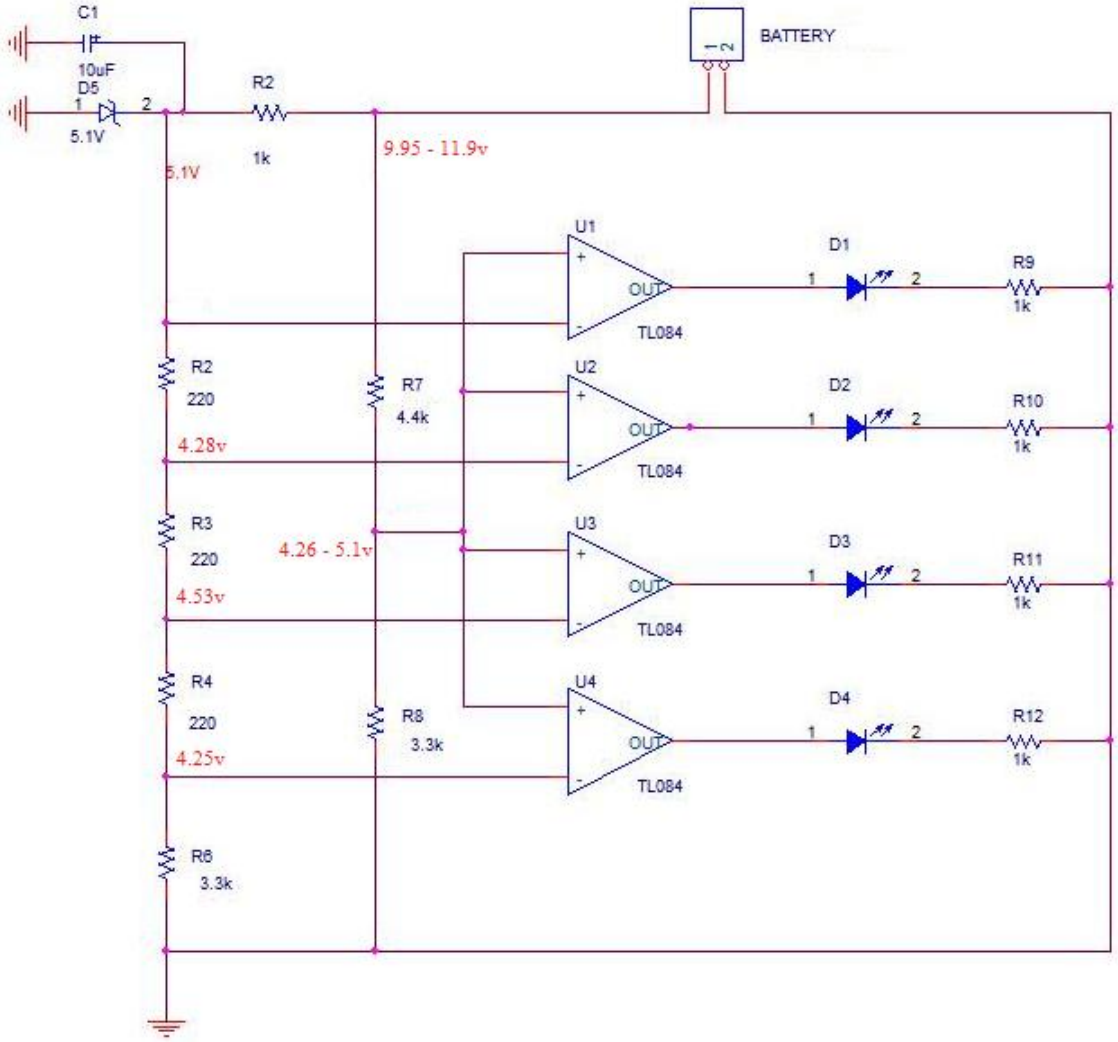


Figure 107: Calculator U_2, U_3, U_4, U_6 and R_2, R_3, R_4, R_6

E. Report No.5 System Implementation & Test

1. Introduction

1.1 Overview

This section provides in detail all necessary information about implementation information and testing procedure of VSLR includes test plans, test cases, test procedures and test result.

1.2 Test Approach

1.2.1 Method

- *Black box testing*: We examine the functionality of the system without peering into its internal structures or workings. This testing can dominate integration testing as well.

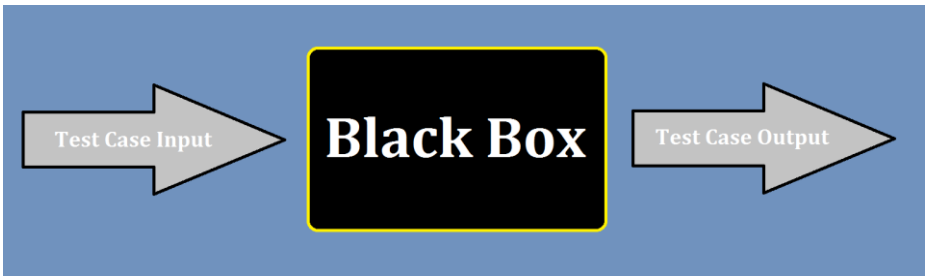


Figure 108: Black box testing

1.2.2 Goal

- To validate that the application works as the user will be operating it, and then find out incorrect or missing functions, interface errors, behavior and performance errors.

2. Database Relationship Diagram

2.1 Physical Diagram

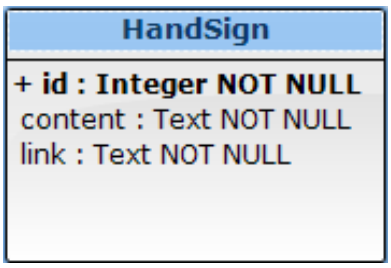


Figure 109: Physical Database Diagram

2.2 Data Dictionary

Entity Data dictionary: describe content of all entities	
Entity Name	Description
HandSign	Describe the hand sign words in the system.

Table 51: Describe content of all entities

Entity name	Attributes	Description	Domain	Null
HandSign	id	Unique identifier of the hand sign.	Integer	No
	content	The translated content of the handsign	Text	No
	link	The image link leads to images describing the hand sign.	Text	No

Table 52: Describe attributes of HandSign entities

3. Test Plan

3.1 Test items

We have a main test phase: Integration test:

- Integration Testing: We test the integration of the code modules developed and interaction with hardware. The integration testing starts at the bottom level. Each component at lower hierarchy is tested individually; then the components that rely upon these are tested.

3.2 Features to be tested

Integration Test includes the following features:

- Background Color Subtraction
- "Selecting Function" function
- "Hand Sign Language Recognition" function
- "Learning Hand Sign Language" function
- "Charging Battery" function
- "Monitoring Battery Capacity" function
- "Turn off application" function

3.3 Features not to be tested

N/A

3.4 Environmental needs

- A complete system with fully devices and functions
- An environment with stable light and background is not complex in color.

3.5 Test case pass/fail criteria

- Every test case must describe what expected outputs are to pass that specific test.
- Test coverage must be at least 90%.
- All test cases must pass.

4. Integration Test Specifications

4.1 "Background Color Subtraction" Test

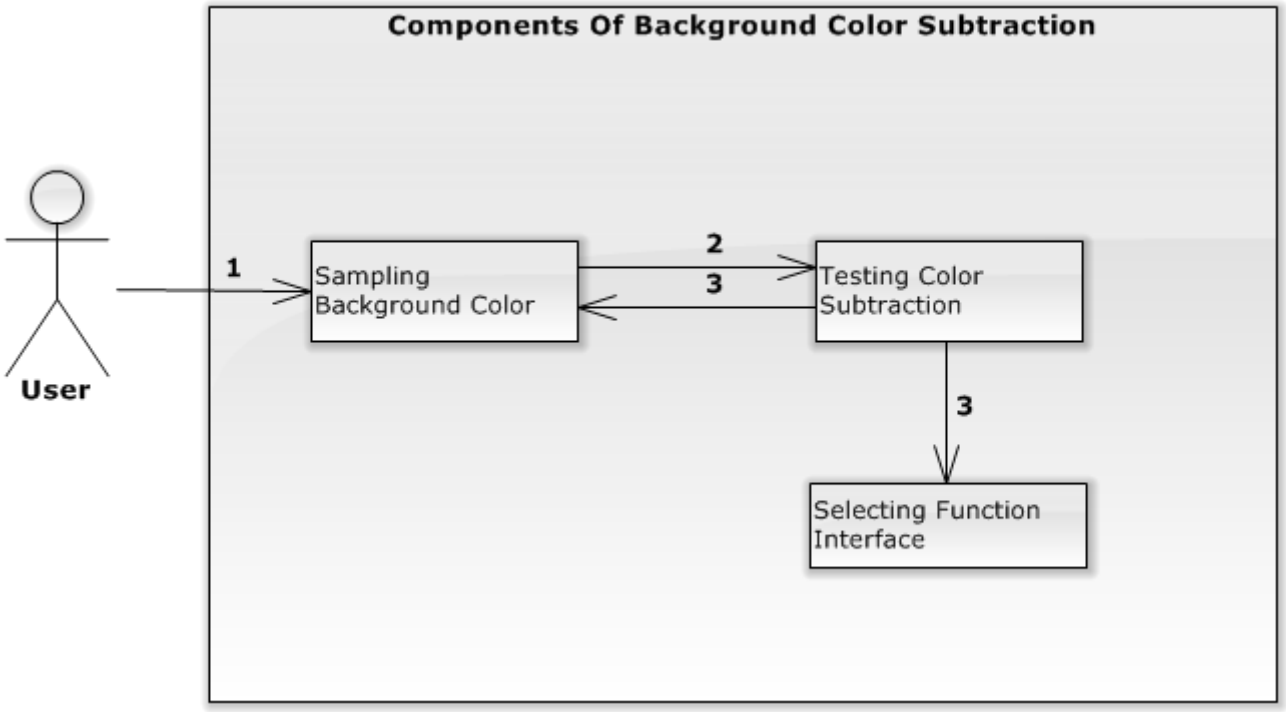


Figure 110: Components of the Background Color Subtraction

4.1.1 Integration test case

ID	Test Item(s)	Input specification	Expected Output	Condition
BCSTC01	Display Screen to Sampling Background Color	Switch on	Displays notify “Người dùng vui lòng di chuyển ra khỏi vùng camera đang theo dõi”.	N/A
			Shows the images captured from camera on the interface for users.	
			Shows the count down time and counts down from 5 by a second.	
			Show the message “Đang tiến hành".	
BCSTC02	Background Color -> Testing Color Subtraction	The background is fixed and user waiting for the countdown time counts to 0.	A notify “Vui lòng điều chỉnh bàn tay của bạn theo kí hiệu “kiểm tra” trong hướng dẫn” is shown.	Test case BCSTC01 is executed
			Shows the images subtracting background color on the interface for users	
			Countdown time is shown by seconds from 5.	
			Shows the message “Đang tiến hành"	
BCSTC03	Background Color -> Testing Color Subtraction	The background is continuously changing and user waiting for the countdown time counts to 0.	A notify “Vui lòng điều chỉnh bàn tay của bạn theo kí hiệu “kiểm tra” trong hướng dẫn” is shown.	Test case BCSTC01 is executed
			Shows the images subtracting background color on the interface for users	
			Countdown time is shown by seconds from 5.	

			Shows the message “Đang tiến hành"	
BCSTC04	Testing Color Subtraction	Showing right “testing” hand sign inside camera area	Counting down continues. Show a message “Thành Công”.	Test case BCSTC03 or BCSTC04 is executed
BCSTC05	Testing Color Subtraction	Showing wrong “testing” hand sign inside camera area	Counting down continues Show a message “Đang tiến hành”.	Test case BCSTC03 or BCSTC04 is executed
BCSTC06	Testing Color Subtraction	Don’t show hand sign	Counting down continues Show a message “Đang tiến hành”.	Test case BCSTC03 or BCSTC04 is executed
BCSTC07	Testing Color Subtraction	Showing right “testing” hand sign outside camera area	Counting down continues Show a message “Đang tiến hành”.	Test case BCSTC03 or BCSTC04 is executed
BCSTC08	Testing Color Subtraction	Showing wrong “testing” hand sign outside camera area	Counting down continues Show a message “Đang tiến hành”.	Test case BCSTC03 or BCSTC04 is executed
BCSTC09	Testing Color Subtraction -> Sampling Background Color	Waiting for the countdown time counts to 0.	Shows message “Thất bại” A notify “Người dùng vui lòng di chuyển ra khỏi vùng camera đang theo dõi” is shown. Show the images captured from camera on the interface for users . Shows the count down time and counts down from 5 by a second	Test case BCSTC06 or BCSTC07 or BCSTC08 or BCSTC09 is executed

			Show the message “Đang tiến hành".	
BCSTC10	Testing Color Subtraction -> Selecting Function Interface	Waiting for the countdown time counts to 0.	Selecting Function Interface is shown	Test case BCSTC05 is executed

Table 53: Background Color Subtraction Intergration Test Case

4.1.2 Integration test procedure

ID	Purpose	Procedure Steps	Executed By	Result	Test Date	Note
BCSITP01	Test the success background color subtraction flow can work correctly	1. Execute test case BCSTC01 2. Execute test case BCSTC02 3. Execute test case BCSTC04 4. Execute test case BCSTC10	LongNHK	Pass	11/08/2015	
BCSITP02	Test the fail background color subtraction work flow correctly: show wrong “test” hand sign inside camera area, background is fixed	1. Execute test case BCSTC01 2. Execute test case BCSTC02 3. Execute test case BCSTC05 4. Execute test case BCSTC09	LongNHK	Pass	11/08/2015	
BCSITP03	Test the fail background color subtraction work flow correctly: show right “testing” hand sign inside camera area, background is continuously changing	1. Execute test case BCSTC01 2. Execute test case BCSTC03	LongNHK	Pass	11/08/2015	

		3. Execute test case BCSTC04 4. Execute test case BCSTC09				
BCSITP04	Test the fail background color subtraction work flow correctly : background is continuously changing, show wrong “test” hand sign inside camera area	1. Execute test case BCSTC01 2. Execute test case BCSTC03 3. Execute test case BCSTC05 4. Execute test case BCSTC09	LongNHK	Pass	11/08/2015	
BCSITP05	Test the fail background color subtraction work flow correctly: don’t show hand sign, background is fixed	1. Execute test case BCSTC01 2. Execute test case BCSTC02 3. Execute test case BCSTC06 4. Execute test case BCSTC09	LongNHK	Pass	11/08/2015	
BCSITP06	Test the fail background color subtraction work flow correctly: don’t show hand sign, background is continuously changing	1. Execute test case BCSTC01 2. Execute test case BCSTC03 3. Execute test case BCSTC06 4. Execute test case BCSTC09	LongNHK	Pass	11/08/2015	
BCSITP07	Test the fail background color subtraction work flow correctly: show right “testing” hand sign outside camera area, background is fixed	1. Execute test case BCSTC01 2. Execute test case BCSTC02	LongNHK	Pass	11/08/2015	

		3. Execute test case BCSTC07 4. Execute test case BCSTC09				
BCSITP08	Test the fail background color subtraction work flow correctly: show right “testing” hand sign outside camera area, background is continuously changing	1. Execute test case BCSTC01 2. Execute test case BCSTC03 3. Execute test case BCSTC07 4. Execute test case BCSTC09	LongNHK	Pass	11/08/2015	
BCSITP09	Test the fail background color subtraction work flow correctly: show wrong “testing” hand sign outside camera area, background is fixed	1. Execute test case BCSTC01 2. Execute test case BCSTC02 3. Execute test case BCSTC08 4. Execute test case BCSTC09	LongNHK	Pass	11/08/2015	
BCSITP10	Test the fail background color subtraction work flow correctly: show wrong “testing” hand sign outside camera area, background is continuously changing	1. Execute test case BCSTC01 2. Execute test case BCSTC03 3. Execute test case BCSTC08 4. Execute test case BCSTC09	LongNHK	Pass	11/08/2015	

Table 54: Background Color Subtraction Intergration Test procedure

4.2 “Hand Sign Language Recognition” Test

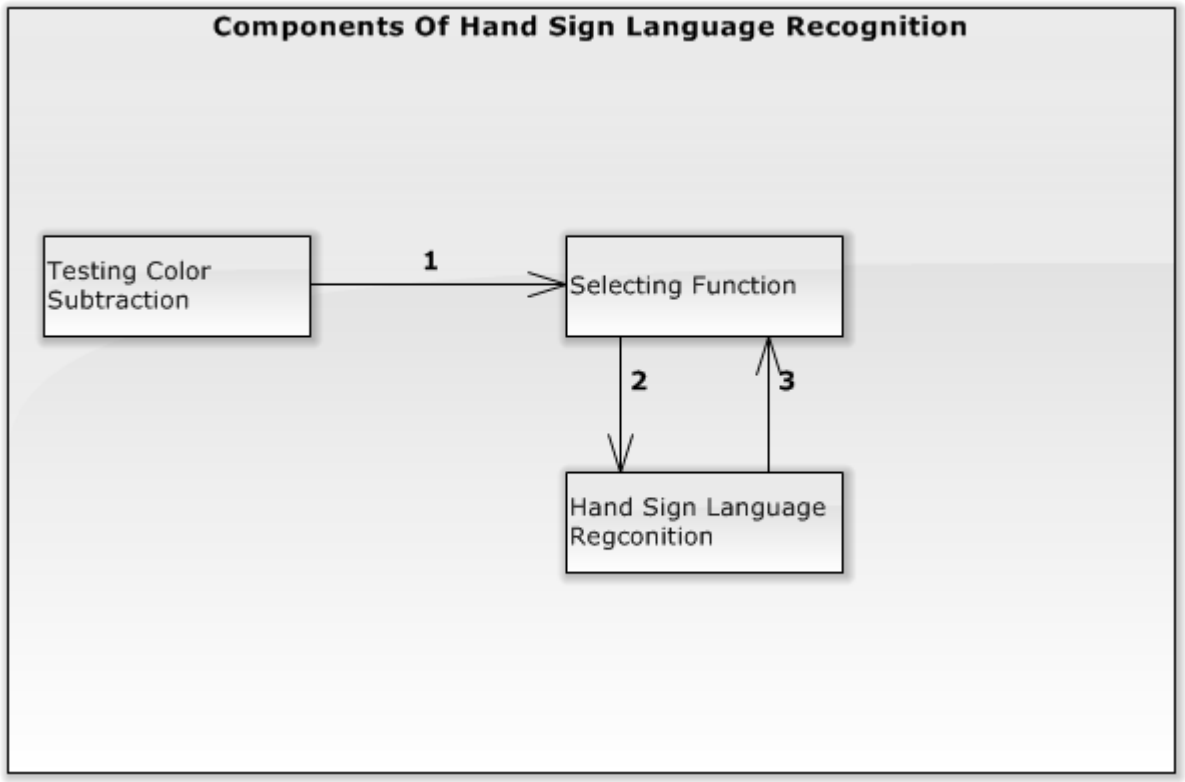


Figure 111: Components of the Hand Sign Language Recognition

4.2.1 Test Case Specification

ID	Test Item(s)	Input Specification	Output Specification	Condition
HSRTC01	Testing Color Subtraction -> Selecting Function	N/A	Show a notify " Hãy chọn chức năng mong muốn bằng cách đưa ký hiệu hình bên vào vùng chức năng đó" in the groupbox "Thông Báo"	Background Color Subtraction is executed
			The analyzed images show on the interface continuously.	
			Two white "Nhận Dạng" and "Học" area are drawn inside analyzed images	
			System shows an image guiding users to select function	
HSRTC02	Selecting Function	Show "select" hand sign outside "Nhận dạng" area	Nothing change	Test case HSRTC01 is executed
HSRTC03	Selecting Function	Show hand sign different "select" hand sign inside "Nhận dạng" area	Nothing change	Test case HSRTC01 is executed
HSRTC04	Selecting Function	Show hand sign different "select" hand sign outside "Nhận dạng" area	Nothing change	Test case HSRTC01 is executed
HSRTC05	Selecting Function -> Hand Sign Recognition	Show "select" hand sign inside "Nhận dạng" area.	The system shows the hand sign recognition interface	Test case HSRTC01 is executed
			The analyzed images show on the interface continuously	
			A notify "Hệ thống sẽ lưu lại kết quả nhận dạng sau 3 giây" is shown in groupbox "Thông Báo"	

			Countdown time is shown from 3 in groupbox “Thời Gian” The system shows two groupbox “Nội dung toàn bộ” and “Kết Quả Hiện Tại” with empty content.	
HSRTC06	Hand Sign Language Recognition	Showing the “A” hand sign through camera	Countdown time counts down by second. Groupbox “ Kết Quả Hiện Tại ” is shown with the result content “A” below.	Test case HSRTC05 is executed
HSRTC07	Hand Sign Language Recognition	Don’t show hand sign through camera	Countdown time counts down by second. Show a message “Không tìm thấy bàn tay!” in the group “Kết Quả Hiện Tại”.	Test case HSRTC05 is executed
HSRTC08	Hand Sign Language Recognition	User keeps the “A” hand sign through camera and waiting for the countdown time counts to 0.	Groupbox “ Nội Dung Toàn Bộ ” is shown with the result content “A”	Test case HSRTC06 is executed
HSRTC09	Hand Sign Language Recognition	Showing the “C” hand sign through camera and keeps when the countdown time counts to 0.	Groupbox “ Nội Dung Toàn Bộ ” is shown with the result content “C”	Test case HSRTC06 is executed
HSRTC010	Hand Sign Language Recognition	Showing the “speak” hand sign through camera and keeps when the countdown time counts to 0.	Groupbox “ Nội Dung Toàn Bộ ” is still empty	Test case HSRTC06 is executed
HSRTC11	Hand Sign Language Recognition	Showing the “B” hand sign through camera	Countdown time counts down by second. Groupbox “ Kết Quả Hiện Tại ” is shown with the result content “B”	Test case HSRTC08 is executed
HSRTC12	Hand Sign Language Recognition	User keeps the “B” hand sign through camera and waiting for the countdown time counts to 0.	Groupbox “ Nội Dung Toàn Bộ ” is shown with the result content “A B”	Test case HSRTC10 is executed
HSRTC13	Hand Sign Language	Show “speak” hand sign inside the	Reads “A” via LCD speaker	Test case

	Recognition	camera area and waiting for the countdown time counts to 0.	Groupbox " Nội Dung Toàn Bộ " is clear	HSRTC08 is executed
HSRTC14	Hand Sign Language Recognition	No hand inside the camera area and waiting for the countdown time counts to 0	Countdown time counts down by second.	Test case HSRTC06 is executed
			Show a message "Không tìm thấy bàn tay!" in the group "Kết Quả Hiện Tại".	
HSRTC15	Hand Sign Language Recognition	Show "end" hand sign then show "speak" hand sign inside the camera area and waiting for the countdown time counts to 0	Reads "A" via LCD speaker	Test case HSRTC08 is executed
			Groupbox " Nội Dung Toàn Bộ " is clear	
HSRTC16	Hand Sign Language Recognition -> Selecting Function	Show "speak" hand sign then show "end" hand sign inside the camera area and waiting for the countdown time counts to 0	Selecting Function interface is displayed	Test case HSRTC08 is executed
HSRTC17	Hand Sign Language Recognition -> Selecting Function	Showing the "end" hand sign through camera and waiting for the countdown time counts to 0.	Selecting Function interface is displayed	Test case HSRTC08 is executed

Table 55: Hand Sign Language Recognition Intergration Test Case

4.2.2 Integration Test Procedure

ID	Purpose	Procedure Steps	Executed By	Result	Test Date	Note
HSRTP01	Test fail “Selection Function”: “select” hand sign outside “Nhận dạng” area	1. Execute test case HSRTC01 2. Execute test case HSRTC02	TanND	**Pass**	11/08/2015	
HSRTP02	Test fail “Selection Function”: hand sign different “select” hand sign inside “Nhận dạng” area	1. Execute test case HSRTC01 2. Execute test case HSRTC03	TanND	**Pass**	11/08/2015	
HSRTP03	Test fail “Selection Function”: different “select” hand sign outside “Nhận dạng” area	1. Execute test case HSRTC01 2. Execute test case HSRTC04	TanND	**Pass**	11/08/2015	
HSRTP04	Test the flow Hand Sign Recognition can work correctly show the “AB” content in groupbox “ Nội Dung Toàn Bộ ”	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC06 4. Execute test case HSRTC08 5. Execute test case HSRTC12	TanND	**Pass**	11/08/2015	
HSRTP05	Test the flow Hand Sign Recognition can work correctly show and speak the “A” content	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC06 4. Execute test case HSRTC08 5. Execute test case HSRTC13	TanND	**Pass**	11/08/2015	
HSRTP06	Test the flow Hand Sign Recognition can work correctly show and speak the “A” content	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC06 4. Execute test case HSRTC08 5. Execute test case HSRTC15	TanND	**Pass**	11/08/2015	
HSRTP07	Test the flow Hand Sign Recognition can work correctly show the “C” content in groupbox “ Nội Dung Toàn Bộ ”	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC06 4. Execute test case HSRTC09	TanND	**Pass**	11/08/2015	
HSRTP08	Test the flow Hand Sign Recognition can work correctly : show Selecting	1. Execute test case HSRTC01 2. Execute test case HSRTC05	TanND	**Pass**	11/08/2015	

	Function interface	3. Execute test case HSRTC17				
HSRTP09	Test the flow Hand Sign Recognition can work correctly : show Selecting Function interface	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC16	TanND	**Pass**	11/08/2015	
HSRTP10	Test the flow Hand Sign Recognition can work correctly : show Selecting Function interface	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC06 4. Execute test case HSRTC17	TanND	**Pass**	11/08/2015	
HSRTP11	Test the flow Hand Sign Recognition can work correctly : show Selecting Function interface	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC06 4. Execute test case HSRTC16	TanND	**Pass**	11/08/2015	
HSRTP12	Test fail "Hand Sign Recognition" with No hand inside the camera area	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC06 4. Execute test case HSRTC14	TanND	**Pass**	11/08/2015	
HSRTP13	Test fail "Hand Sign Recognition" with No hand inside the camera area	1. Execute test case HSRTC01 2. Execute test case HSRTC05 3. Execute test case HSRTC07	TanND	**Pass**	11/08/2015	

Table 56: Hand Sign Language Recognition Intergration Test Procedure

4.3 “Learning Hand Sign Language” Test

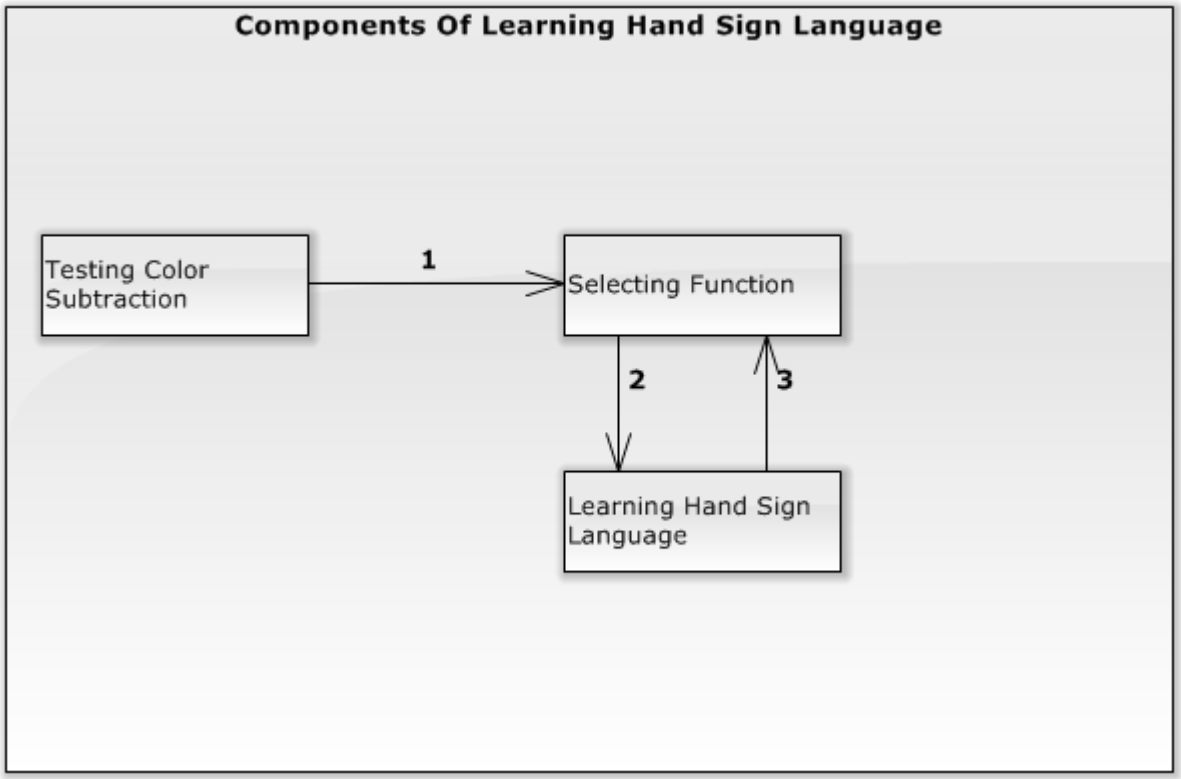


Figure 112: Components of the Learning Hand Sign Language

4.3.1 Test case specification

ID	Test Item(s)	Input Specification	Output Specification	Condition
LHSTC01	Testing Color Subtraction -> Selecting Function	N/A	Show a notify " Hãy chọn chức năng mong muốn bằng cách đưa ký hiệu hình bên vào vùng chức năng đó" in the groupbox "Thông Báo"	Background Color Subtraction is executed
			The analyzed images show on the interface continuously.	
			Two white "Nhận Dạng" and "Học" area are drawn inside analyzed images	
			System shows an image guiding users to select function	
LHSTC02	Selecting Function	Show "select" hand sign outside " Học " area	Nothing change	Test case LHSTC01is executed
LHSTC03	Selecting Function	Show hand sign different "select" hand sign inside " Học " area	Nothing change	Test case LHSTC01is executed
LHSTC04	Selecting Function	Show hand sign different "select" hand sign outside " Học " area	Nothing change	Test case LHSTC01is executed
LHSTC05	Selecting Function -> Learning Hand Sign Language	N/A	Show the list of words on the interface in the group box "Hướng Dẫn"	Test case LHSTC04 is executed
			The analyzed images show on the interface continuously	
			The "A" word is selected first	
			Image describing "A" hand sign is displayed	
			A notify "Hãy đưa kí hiệu trong hướng dẫn vào vùng mũi tên lên xuống để thay đổi từ được	

			chọn ” is shown in the groupbox “Thông Báo” Two white “Lên” and “Xuống” area were drawn on these images showing on the interface.	
LHSTC06	Learning Hand Sign Language	Move the “select” hand sign into the “Lên” square area.	The “A” word in the list is still selected Image describing “A” hand sign is still displayed Group box “Kết Quả Hiện Tại” with no result content below	Test case LHSTC05 is executed
LHSTC07	Learning Hand Sign Language	Move the “select” hand sign into the “Xuống” square area.	The “B” word in the list is selected Image describing “B” hand sign is displayed Group box “Kết Quả Hiện Tại” with no result content below	Test case LHSTC05 or LHSTC06 is executed
LHSTC08	Learning Hand Sign Language	Move the “select” hand sign outside the “Lên” and “Xuống” square area.	The “B” word in the list is still selected Image describing “B” hand sign is still displayed Group box “Kết Quả Hiện Tại” with result “N” content below	Test case LHSTC07 is executed
LHSTC09	Learning Hand Sign Language	Show “E” hand sign outside the “Lên” and “Xuống” square area.	The “B” word in the list is still selected Image describing “B” hand sign is still displayed Group box “Kết Quả Hiện Tại” with result “E” content below	Test case LHSTC07 is executed
LHSTC10	Learning Hand Sign Language	Show “E” hand sign inside the “Lên” square area.	The “B” word in the list is still selected Image describing “B” hand sign is still displayed Group box “Kết Quả Hiện Tại” with result “E” content below	Test case LHSTC07 is executed

LHSTC11	Learning Hand Sign Language	Show “E” hand sign inside the “Xuống” square area.	The images containing only the hand on the The “B” word in the list is still selected Image describing “B” hand sign is still displayed Group box “Kết Quả Hiện Tại” with result “E” content below	Test case LHSTC07 is executed
LHSTC12	Learning Hand Sign Language	Move the “select” hand sign into the “Lên” square area.	The “A” word in the list is selected Image describing “A” hand sign is displayed Group box “Kết Quả Hiện Tại” is empty	Test case LHSTC07 is executed
LHSTC13	Learning Hand Sign Language	Showing the “end” hand sign outside of the two square areas “Lên ” and “Xuống”	Selecting Function interface is displayed	Test case LHSTC05 is executed
LHSTC14	Learning Hand Sign Language -> Selecting Function	Showing the “end” hand sign inside of the two square areas “Lên”	Selecting Function interface is displayed	Test case LHSTC05 is executed
LHSTC15	Learning Hand Sign Language -> Selecting Function	Showing the “end” hand sign inside of the two square areas “Xuống”	Selecting Function interface is displayed	Test case LHSTC05 is executed
LHSTC16	Learning Hand Sign Language	Don’t show hand inside the camera area	Shows a message “Không tìm thấy bàn tay!” in the group box “Kết Quả Hiện Tại”	

Table 57: Learning Hand Sign Language Intergration Test Case

4.3.2 Integration Test Procedure TP4

ID	Purpose	Procedure Steps	Executed By	Result	Test Date	Note
LHSTP01	Test fail “Selection Function”: “select” hand sign outside “Học” area	1. Execute test case LHSTC01 2. Execute test case LHSTC02	LongNHK	**Pass**	11/08/2015	
LHSTP02	Test fail “Selection Function”: hand sign different “select” hand sign inside “Nhận dạng” area	1. Execute test case LHSTC01 2. Execute test case LHSTC03	LongNHK	**Pass**	11/08/2015	
LHSTP03	Test fail “Selection Function”: different “select” hand sign outside “Nhận dạng” area	1. Execute test case LHSTC01 2. Execute test case LHSTC04	LongNHK	**Pass**	11/08/2015	
LHSTP04	Test success Learning Hand Sign Language: select “B” word below the “A” word	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC07	LongNHK	**Pass**	11/08/2015	
LHSTP05	Test success Learning Hand Sign Language: select “A” word above the “B” word	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC07 4. Execute test case LHSTC12	LongNHK	**Pass**	11/08/2015	
LHSTP06	Test success Learning Hand Sign Language: select “Lên” when “A” word is selected in the top	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC06	LongNHK	**Pass**	11/08/2015	
LHSTP07	Test success Learning Hand Sign Language: checking “E” hand sign with result “E”	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC07 4. Execute test case LHSTC09	LongNHK	**Pass**	11/08/2015	
LHSTP08	Test success Learning Hand Sign Language: checking “E” hand sign with result “E”	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC07 4. Execute test case LHSTC10	LongNHK	**Pass**	11/08/2015	
LHSTP10	Test success Learning Hand Sign Language: checking “E” hand sign with result “E”	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC07 4. Execute test case LHSTC11	LongNHK	**Pass**	11/08/2015	

LHSTP11	Test success finish Learning Hand Sign Language	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC13	LongNHK	**Pass**	11/08/2015	
LHSTP12	Test success finish Learning Hand Sign Language	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC07 4. Execute test case LHSTC15	LongNHK	**Pass**	11/08/2015	
LHSTP13	Test success finish Learning Hand Sign Language	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC07 4. Execute test case LHSTC14	LongNHK	**Pass**	11/08/2015	
LHSTP14	Test fail Learning Hand Sign Language	1. Execute test case LHSTC01 2. Execute test case LHSTC05 3. Execute test case LHSTC16	LongNHK	**Pass**	11/08/2015	

Table 58: Learning Hand Sign Language Intergration Test Procedure

4.4 "Monitor Battery Capacity" Test

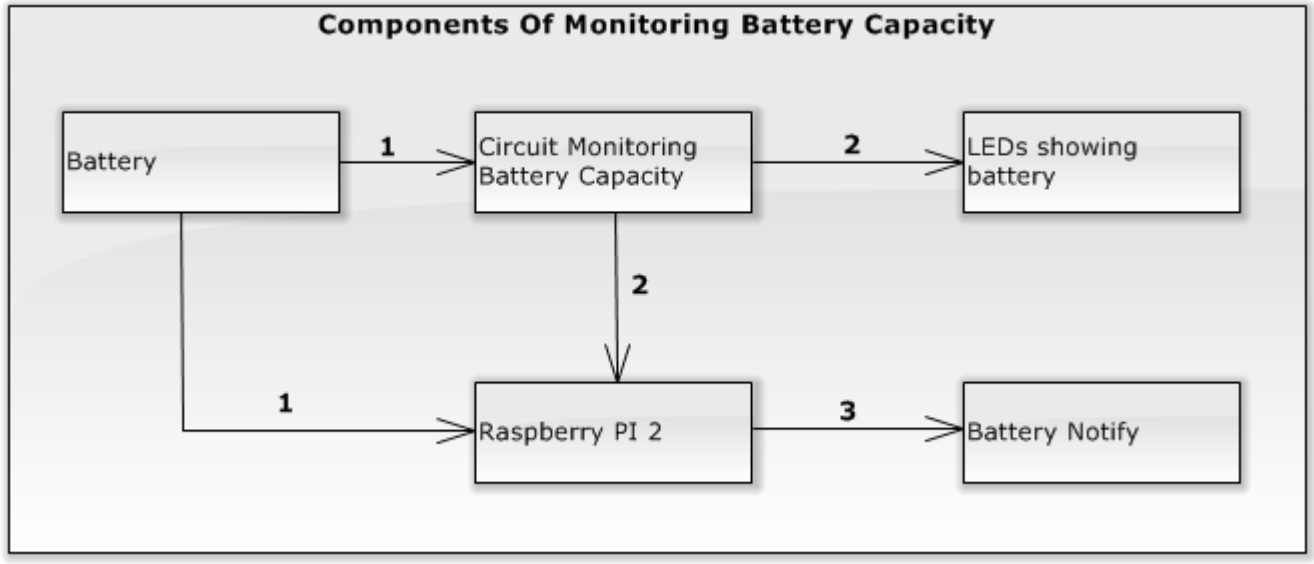


Figure 113: Components of the Monitoring Battery Capacity

4.4.1 Integration test case

ID	Test Item(s)	Input Specification	Output Specification	Condition
MBCTC01	Battery -> Battery Capacity Display Circuit -> LEDs showing battery	Switch on	LED on circuit is bright.	The battery voltage > 9.8V
MBCTC02	Battery -> Battery Capacity Display Circuit -> LEDs showing battery	The battery voltage is higher 12V	4 LEDs on circuit is bright. The application interface shows 100% battery image.	Test case MBCTC01 is exectuted
MBCTC03	Battery -> Battery Capacity Display Circuit -> LEDs showing battery	The battery voltage is lower than or equal 12V and higher than 11.3V	3 LEDs on circuit is bright. The application interface shows 75% battery image.	Test case MBCTC01 is exectuted
MBCTC04	Battery -> Battery Capacity Display Circuit -> LEDs showing battery	The battery voltage is lower than or equal 11.3V and higher than 10.8V	2 LEDs on circuit is bright. The application interface shows 50% battery image	Test case MBCTC01 is exectuted
MBCTC05	Battery -> Battery Capacity Display Circuit -> LEDs showing battery	The battery voltage is lower than or equal 10.8V and higher than 9.9V	1 LEDs on circuit is bright. The application interface shows 25% battery image	Test case MBCTC01 is exectuted
MBCTC06	Battery -> Battery Capacity Display Circuit -> LEDs showing battery	The battery voltage is lower than 9.9V	4 LEDs on circuit is off. The application and Raspberry PI B2 is shutdown.	Test case MBCTC01 is exectuted
MBCTC07	Battery -> Battery Capacity Display Circuit -> Raspberry PI B2	N/A	LED on Raspberry is bright.	Test case MBCTC01 is executed
MBCTC08	Raspberry PI B2 -> Low Battery Notify	The battery voltage is lower than or equal	Low Battery Notify “Bin yếu vui lòng tắt hệ thống	Test case MBCTC02 is executed

		10.8V and higher than 9.9V	và cắm sạc. Thông báo sẽ được tự động tắt." is shown on the system interfaces. Countdown time is shown in Low Battery Notify by seconds from 3.	
MBCTC09	Raspberry PI 2 -> Low Battery Notify	Waiting for the countdown time counts to 0.	Low Battery Notify is hide	Test case MBCTC03 is executed

Table 59: Monitor Battery Capacity Intergration Test Case

4.4.2 Integration test procedure

ID	Purpose	Procedure Steps	Executed By	Result	Test Date	Note
MBCTP01	Test the flow Monitoring Battery Capacity works correctly with low battery.	1. Execute test case MBCTC01 2. Execute test case MBCTC07 3. Execute test case MBCTC05 4. Execute test case MBCTC08 5. Execute test case MBCTC10	BinhLP	**Pass**	11/08/2015	
MBCTP02	Test the flow Monitoring Battery Capacity works correctly with normal battery.	1. Execute test case MBCTC01 2. Execute test case MBCTC07 3. Execute test case MBCTC04 4. Execute test case MBCTC09 5. Execute test case MBCTC10	BinhLP	**Pass**	11/08/2015	
MBCTP01	Test the displaying battery capacity on LEDs.	1. Execute test case MBCTC01 2. Execute test case MBCTC02	BinhLP	**Pass**	11/08/2015	
MBCTP01	Test the displaying battery capacity on LEDS.	1. Execute test case MBCTC01 2. Execute test case MBCTC03	BinhLP	**Pass**	11/08/2015	
MBCTP01	Test the displaying battery capacity on LEDs.	1. Execute test case MBCTC01 2. Execute test case MBCTC04	BinhLP	**Pass**	11/08/2015	

Table 60: Monitor Battery Capacity Intergration Test procedure

4.5 “Charging Battery” Test

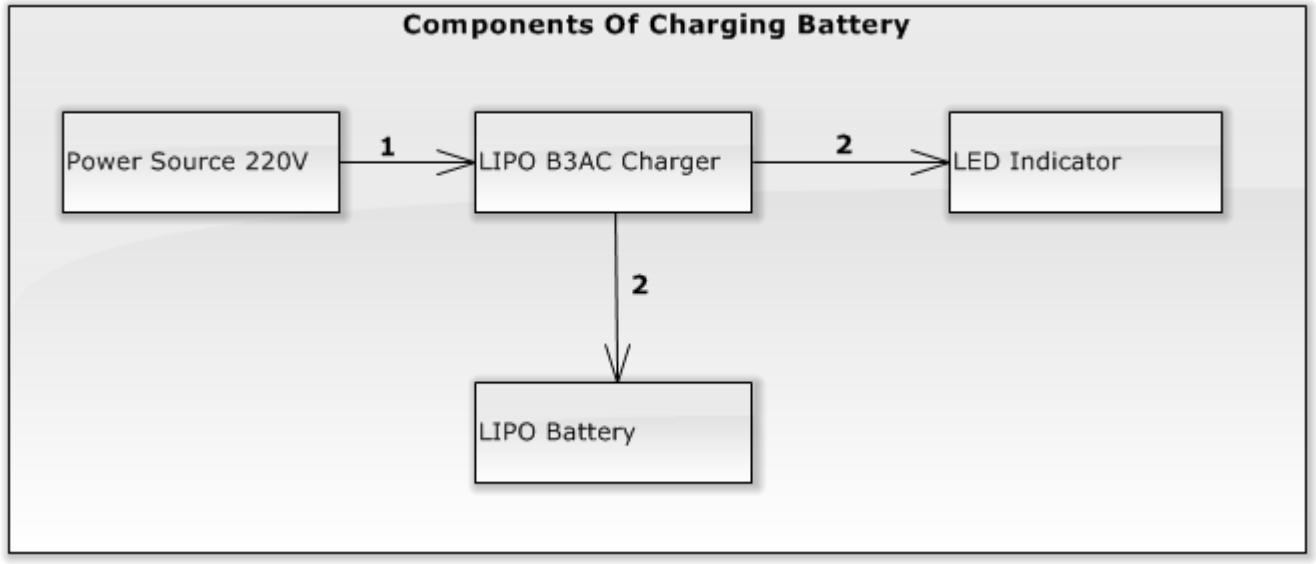


Figure 114: Components of the charging battery

4.5.1 Integration test case

ID	Test Item(s)	Input Specification	Output Specification	Condition
CBTC01	220V power source -> LIPO B3AC charger -> Led Indicator	Connect LIPO B3AC charger to power source	The charger's LEDs indicator is flickering with green and red color.	The system is OFF.
CBTC02	LIPO B3AC charger -> LIPO Battery	Connect LIPO B3AC charger to LIPO Battery has the voltage lower 9.8V.	The charger's LEDs indicator is bright with red color.	Test case CBTC01 is executed.
CBTC03	LIPO B3AC charger -> LIPO Battery	Waiting for about 6 hours.	Three Charger's LEDs are bright with green color.	Test case CBTC02 is executed.

Table 61: Charging Battery Intergration Test Case

4.5.2 Integration test procedure

ID	Purpose	Procedure Steps	Executed By	Result	Test Date	Note
CBIPT01	Test the charging battery flow can succeed and work stability.	1. Execute test case CBTC01 2. Execute test case CBTC02 3. Execute test case CBTC03	YNX	Pass	11/08/2015	
CBIPT02	Test the battery charger can handle the execption of connection between charger and battery is disconnected.	1. Execute test case CBTC01 2. Execute test case CBTC02 3. Execute test case CBTC01	YNX	Pass	11/08/2015	

Table 62: Charging Battery Intergration Test procedure

4.6 Turn off Application Test

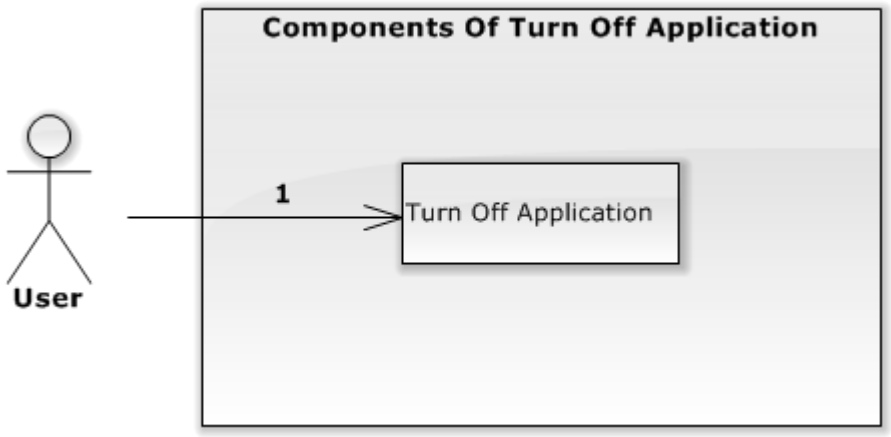


Figure 115: Components of the Turn Off Application

4.6.1 Integration Test Case

ID	Test Item(s)	Input Specification	Output Specification	Condition
TOAC01	Turn off Application	Press "Turn Off" switch	Application is closed	The system is ON.
			Shut down Operation System of Operation System.	

Table 63: Turn off Application Intergration Test Case

4.6.2 Integration Test Procedure

ID	Purpose	Procedure Steps	Executed By	Result	Test Date	Note
TOAIPT01	Test the shutdown application flow can succeed and work stability.	1. Execute test case TOATC01	YNX	Pass	11/08/2015	

Table 64: Turn off Application Intergration Test Procedure

5. Recognition Experiment

5.1 Perform Experiment

Place at Experiment folder in Github

5.2 Statistical data of recognition result in percentage

	A	B	C	D	DD	E	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	backhand	SAC	HUYEN	RAU	MU
A	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
B	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
C	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
D	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
DD	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
E	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
G	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
H	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
I	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
K	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
L	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
M	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
N	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	50%	0%	0%	0%	50%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
O	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
P	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Q	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
R	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
S	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
T	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%
U	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
V	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
X	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
Y	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%
backhand	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
SAC	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%
HUYEN	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%
RAU	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
MU	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%

Table 65: Statistical data of recognition result in percentage

F. Report No. 6 Software User's ManualProblem Definition

1. Installation Guide

1.1 Setting up environment

The following requirements in hardware and software must be set up to system can works.

1.1.1 Hardware requirements

- Raspberry Pi B2 Kit is used to process as central processing unit
- Micro SD card 16GB is used to setup Raspbian operating system.
- Logitech C270 webcam is used to capture images.
- 7 INCH TFT COLOR MONITOR LCD is used to show the interface of application.
- LIPO 3 cells battery (12V - 2.2 A) is power source of the system.
- UNI regulator board: LM2576ADJ - Board
- B3AC compact charger is charger of LIP0 3 cells battery.
- Monitor Battery Capacity circuit, which is constructed by the system.
- HDMI to HDMI cable is connection between Raspberry PI B2 Kit and LCD.
- 1 LED luxeon (1W 350mA) is used to balance light.

1.1.2 Hardware connection

1.1.2.1 For LCD

Figure 116: Connection LCD to raspberry and Lipo Battery

- Connect Lipo battery: plug to DC 12V port on LCD
- Connect to Raspberry: plug HDMI cable from HDMI port on LCD to HDMI port on Raspberry

1.1.2.2 For Monitor Battery Capacity Circuit

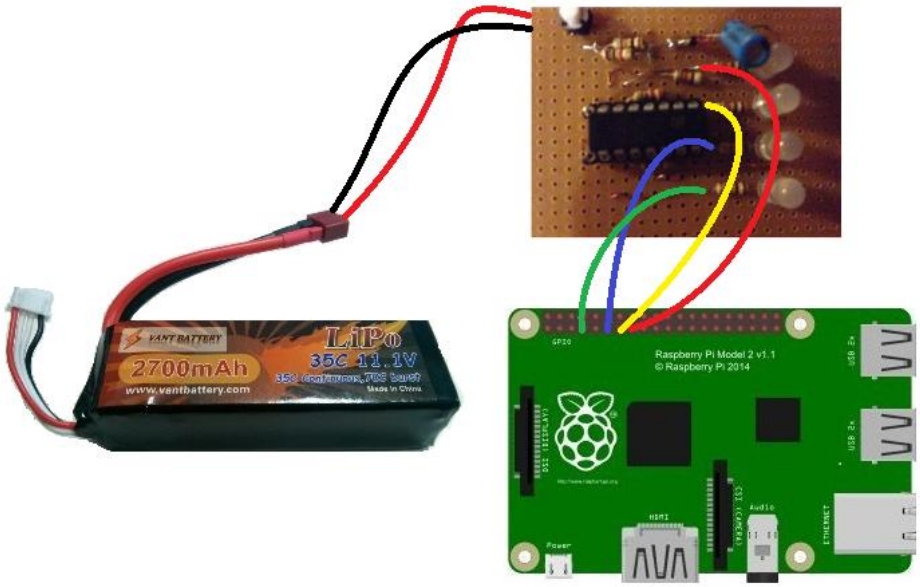


Figure 117: Connection Monitor Battery Capacity Circuit to raspberry and Lipo Battery

- Connect to Lipo Battery: Red wire to anode (+) of battery (red wire). Black wire to cathode (-) of battery (black wire).
- Connect to Raspberry: 4 wires connect to PIN No.7, PIN No.11, PIN No.13, PIN No.15

1.1.2.3 For Webcam

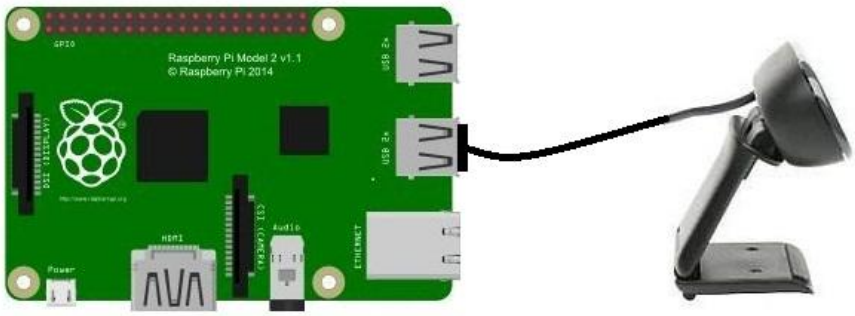


Figure 118: Connection Webcam to raspberry

- Connect to Raspberry: plug wire of Webcam to USB port on Raspberry

1.1.2.4 Power for Raspberry

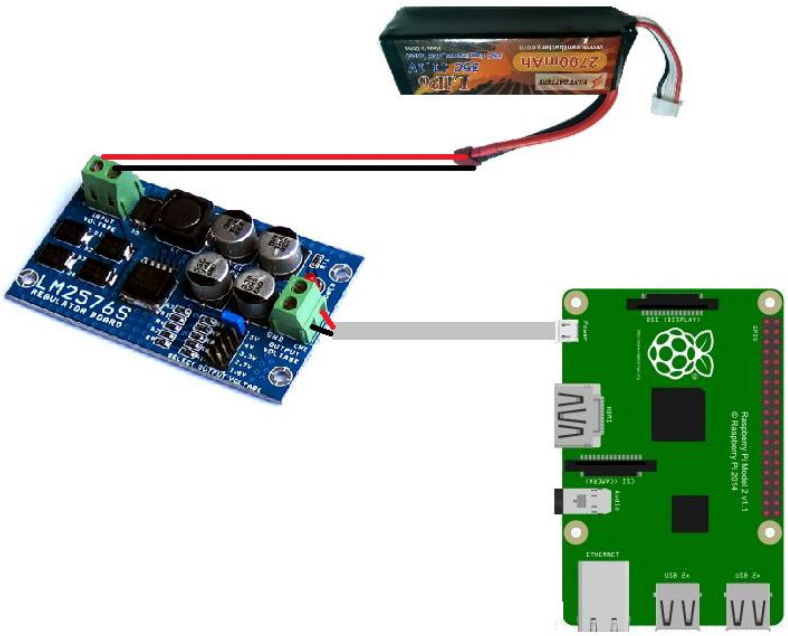


Figure 119: Connection LM2596ADJ-Board to raspberry and battery

- Connect LM2596ADJ-Board to Lipo Battery: connect anode (+) of battery (red wire) to Vccin (+) Pin of LM2596ADJ-Board, connect cathode (-) of battery (black wire) to Vccin (-) Pin of LM2596ADJ-Board
- Connect LM2596ADJ-Board to Raspberry: connect to Micro USB port of Raspberry.

1.1.2.5 For switch "Shutdown application"

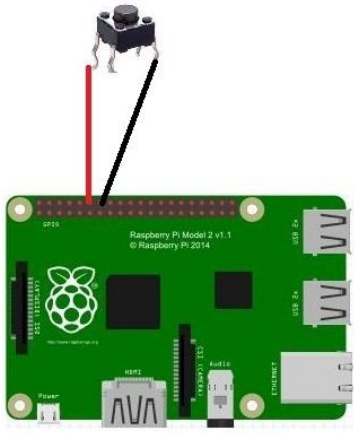


Figure 120: Connection switch "Shutdown Application" to raspberry

Connect to Raspberry: connect to Pin No 14, Pin No 16 on Raspberry.

1.1.2.6 For Led luxeon 1W 350mA

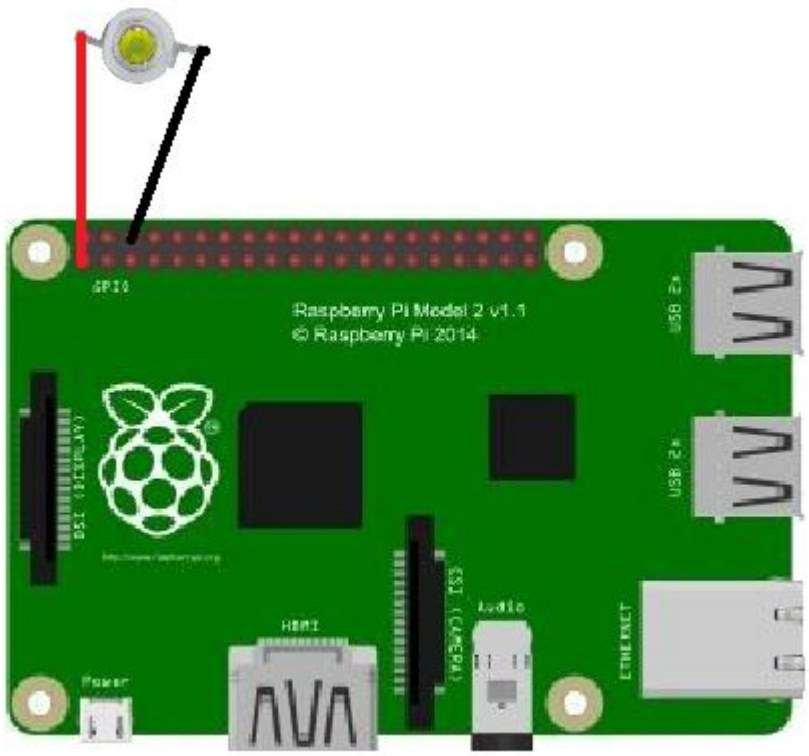


Figure 121: Connection Led luxeon 1W 350mA to raspberry

- Connect anode (+) of Led to Pin No. 1 of raspberry, cathode (-) of led to Pin No. 6 of raspberry

1.1.2.7 All connection of system

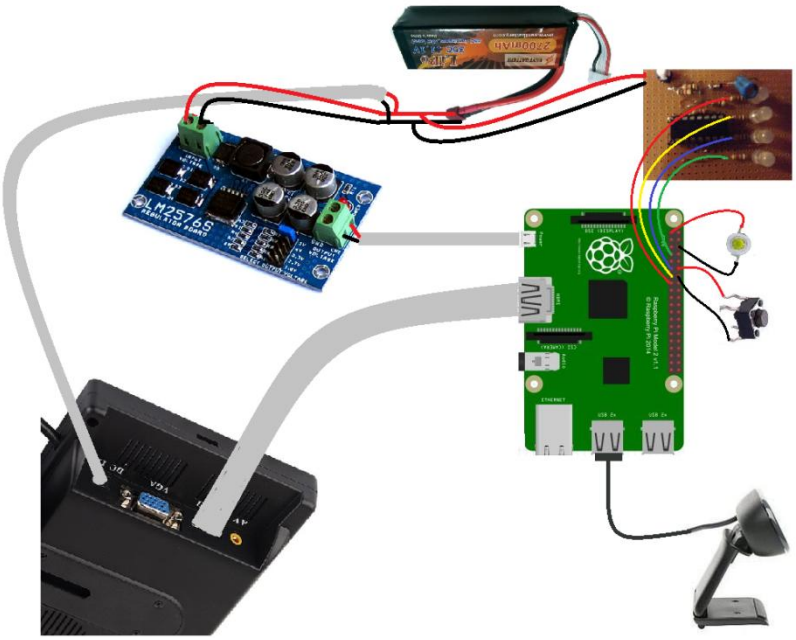


Figure 122: Connection of system

1.1.3 Configure Raspberry PI B2

1.1.3.1 Install operating system

- Download RASPBIAN operating system from this webpage https://www.raspberrypi.org/downloads/
- Install operating system images by following instructions in this webpage https://www.raspberrypi.org/documentation/installation/installing-images/README.md

1.1.3.2 Make Raspberry PI B2 use full solution of the monitor

- Follow the instructions in this webpage http://www.webtechgadgetry.com/2013/12/make-raspberry-pi-use-full-resolution-monitor/

1.1.3.3 Configure LCD settings

Prevent the monitor screen from going to blank

- Firstly, install xset, a lightweight application that controls power-saving settings.

Execute below command in terminal

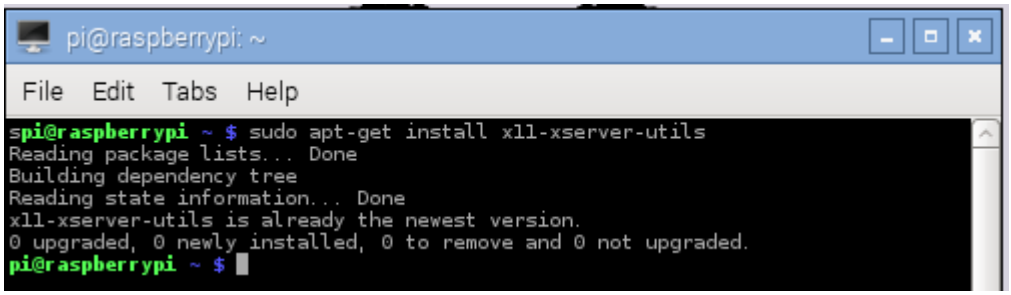


Figure 123: Install xset

- Enable to configure power-saving setting at the time Raspberry PI B2 starts up.
- Execute below command to edit "autostart" file containing commands, which will be executed at startup.

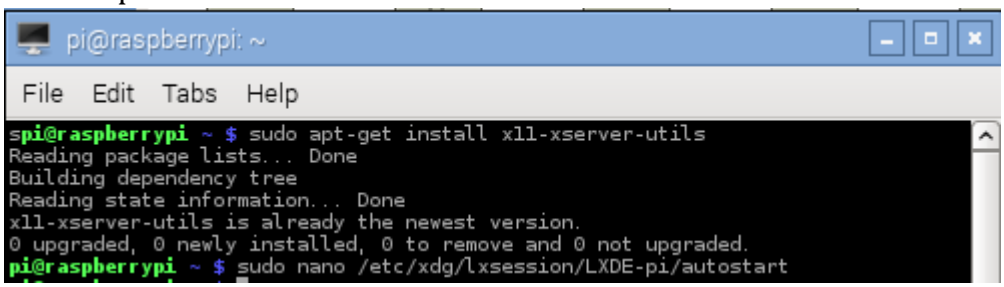


Figure 124: Edit "autostart" file containing commands

Comment line @xscreensaver -no-splash by adding # in the start line.

Add settings to this file by adding below lines at the end of file

@/usr/bin/xset s off

@/usr/bin/xset s noblank
@/usr/bin/xseet -dpms

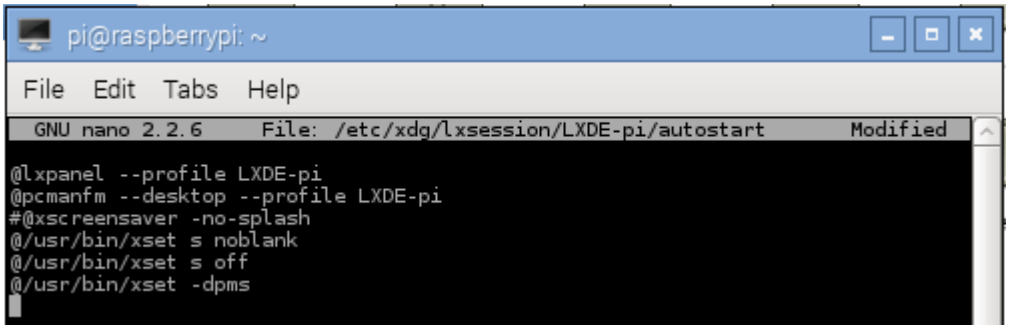


Figure 125: "autostart" file

- Ctrl + X and then type Y to save this file.

1.1.3.4 Configure webcam settings

Disable auto balancing exposure and light

- Firstly, we need to install uvcdynctrl, a application controls webcam settings.
  Execute below command to install uvcdynctrl

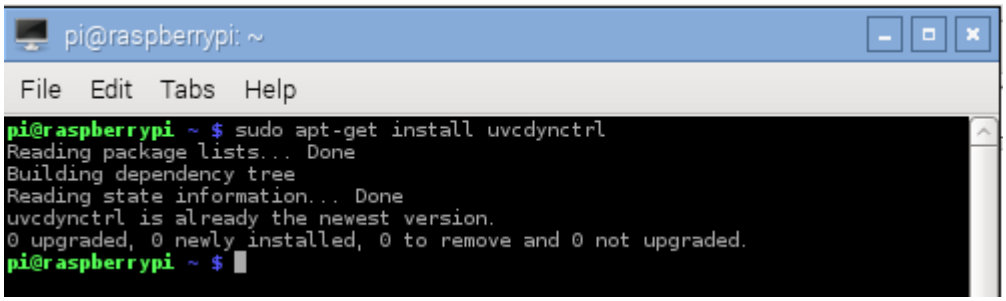


Figure 126: Install uvcdynctrl

- Change setting exposure and auto_white_balance to manual.
  Execute below command to edit "autostart" file containing commands, which will be executed at startup.

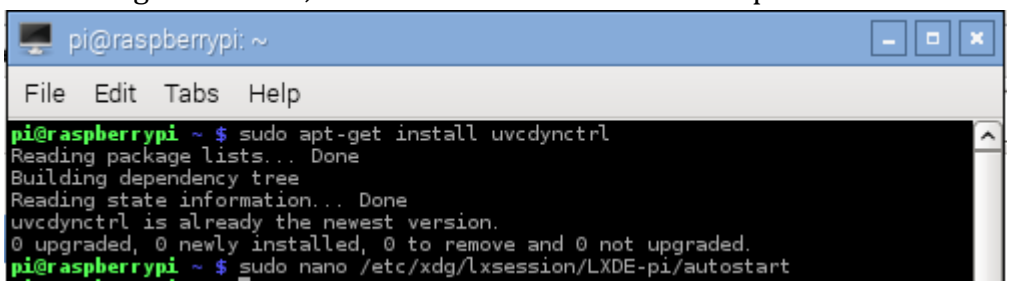


Figure 127: Edit "autostart" file containing commands

Add settings to this file by adding below lines at the end of file

@/usr/bin/uvcdynctrl -s "Exposure, Auto" 1

@/usr/bin/uvcdynctrl -s "White Balance Temperature, Auto" 0

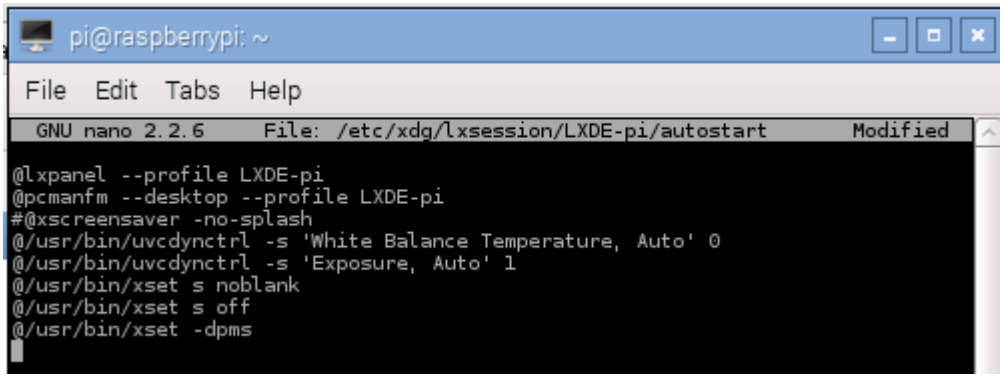


Figure 128: Add settings to "autostart" file

Press Ctrl + X and type Y to save this file

1.1.3.5 Enable Raspberry PI B2 to run application automatically at startup

- Enable Raspberry PI B2 to auto start the desktop

Step 1: Open a terminal and run below command to open Raspberry PI configuration

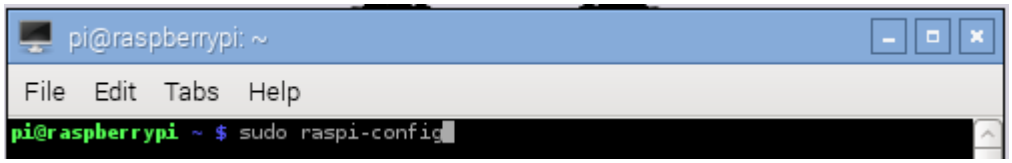


Figure 129: Raspberry PI B2configuration Command

Step 2: Select Enable Boot to Desktop/Scratch from the menu and press Enter

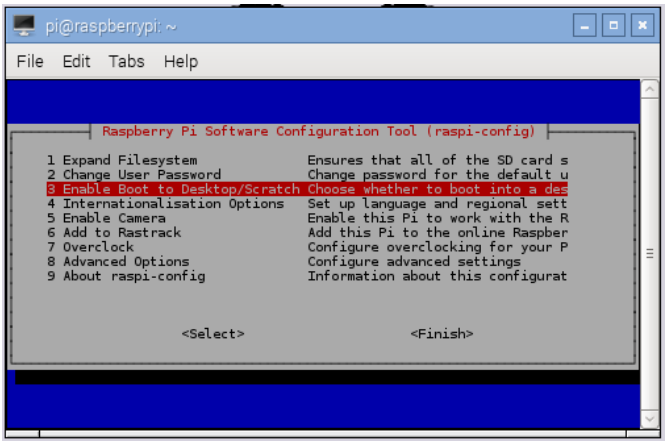


Figure 130: Raspberry PI B2configuration menu

Step 3: Select Desktop Login as user pi at the Graphical Desktop.

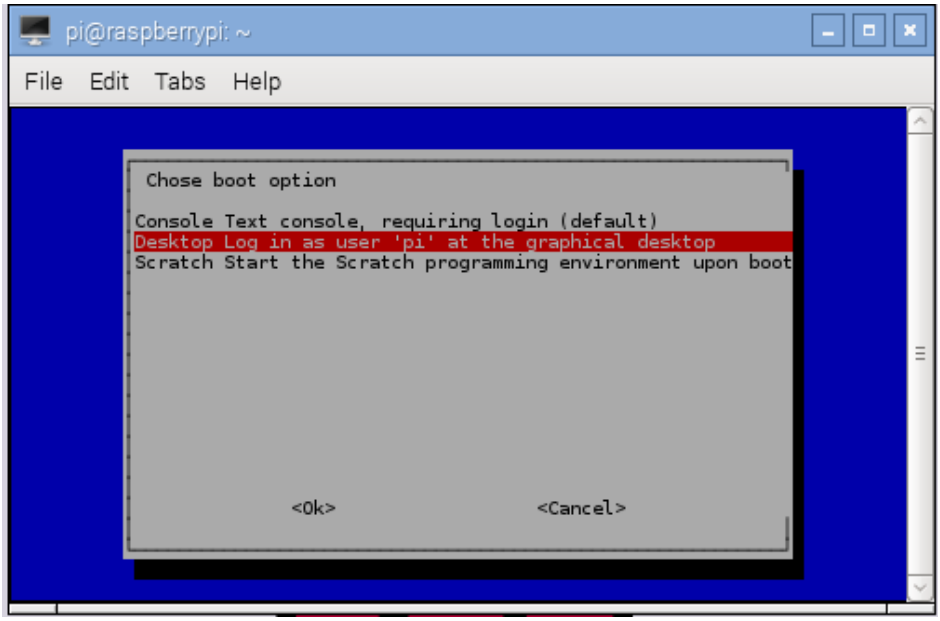


Figure 131: Boot option

Step 4: Select <Finish> and Enter, then select <Yes> to reboot.

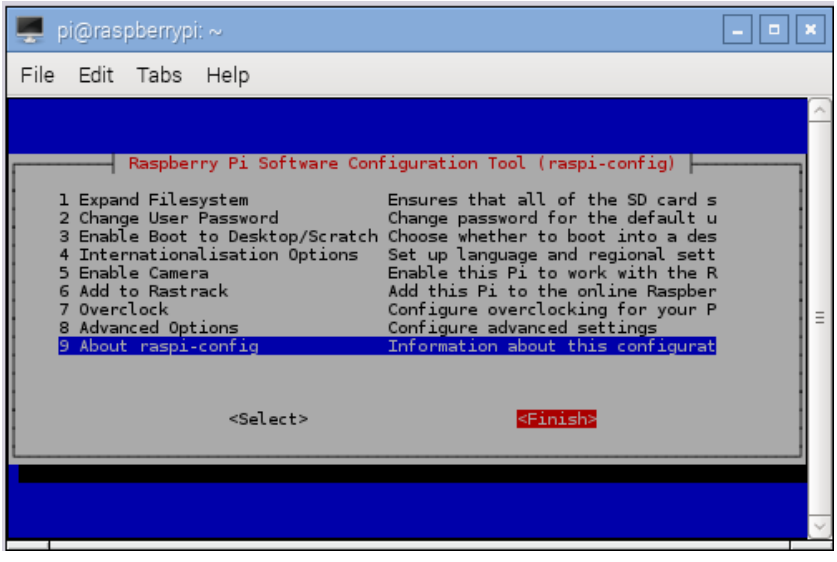


Figure 132: Raspberry PI B2 configuration menu

- Enable the root user can run application at startup

Step 1: Open terminal and execute command "sudo nano /etc/xdg/lxsession/LXDE-pi/autostart" to edit "autostart" file containing commands which will be executed at startup

Figure 133: Edit "autostart" file containing commands

Step 2: Add above lines to this file
@/usr/bin/xhost +
@/usr/bin/xhost + localhost
@/usr/bin/sudo /home/pi/SRC/MainCapstoneProject-build/MainCapstoneProject &

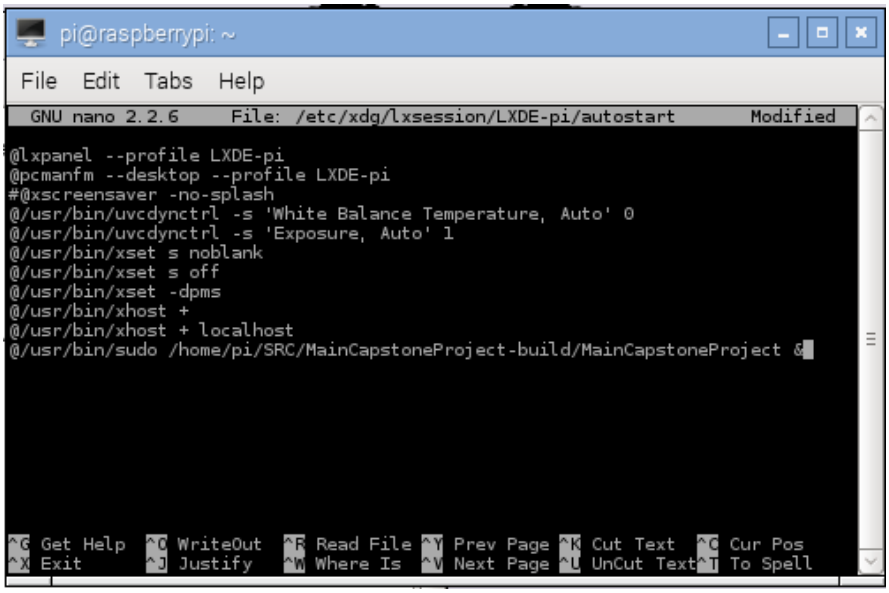


Figure 134: "autostart" file

Step 3: Press Ctrl + X and type Y to save this file.

1.1.4 Software requirements

- Install Opencv library version 2.4.9 on Raspberry PI B2 by following the instructions in this webpage http://daveaubin.com/index.php/how-to-build-and-install-opencv-2-4-9-on-raspberry-pi/
- Install bcm2835 library version 1..45 on Raspberry PI B2
  Step 1: Download bcm2835 from http://www.airspayce.com/mikem/bcm2835/bcm2835-1.45.tar.gz
  Step 2: Install bcm2835 library by following the steps at "Installation" section in this webpage http://www.airspayce.com/mikem/bcm2835/
  Step 3: Enable Raspberry PI 2 can work with bcm2835 library by following the steps at "Raspberry Pi 2 (RPI2)" section in this webpage http://www.airspayce.com/mikem/bcm2835/

- Install espeak library version 1.48.04 on Raspberry PI 2

Step 1: Install development library for portable audio I/O from this link

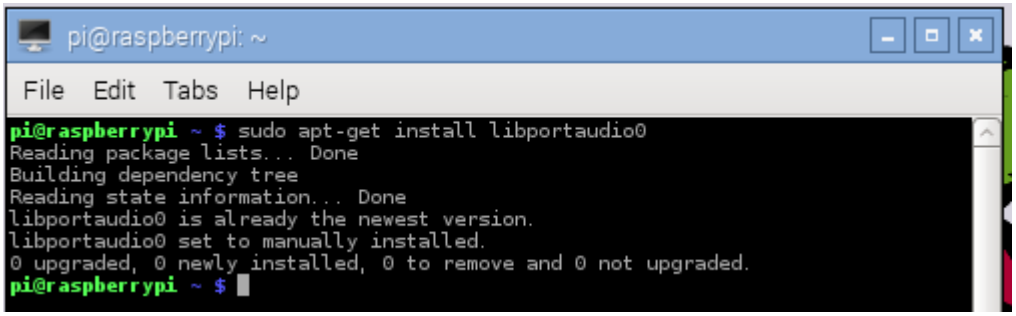


Figure 135: Install library for portable audio I/O command

Step 2: Download espeak library version 1.48.04 from this link http://sourceforge.net/projects/espeak/files/espeak/espeak-1.48/espeak-1.48.04-source.zip and uncompress it.

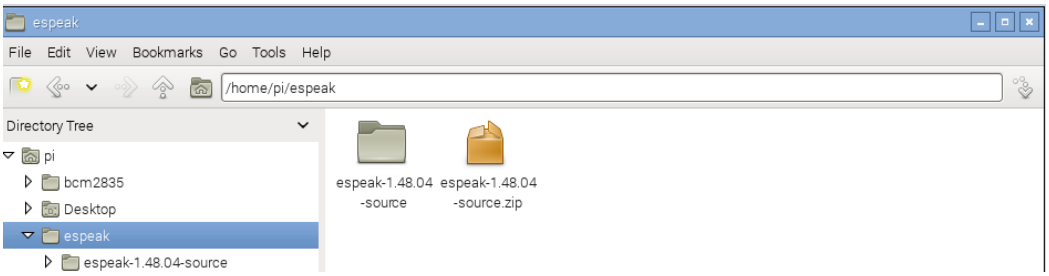


Figure 136: Espeak Directory

Step 3: Execute the following commands to install espeak library

```
cd espeak-1.48.04-source/src
make
sudo cp espeak speak /usr/bin
cd ../
sudo cp -r espeak-data /usr/share
```

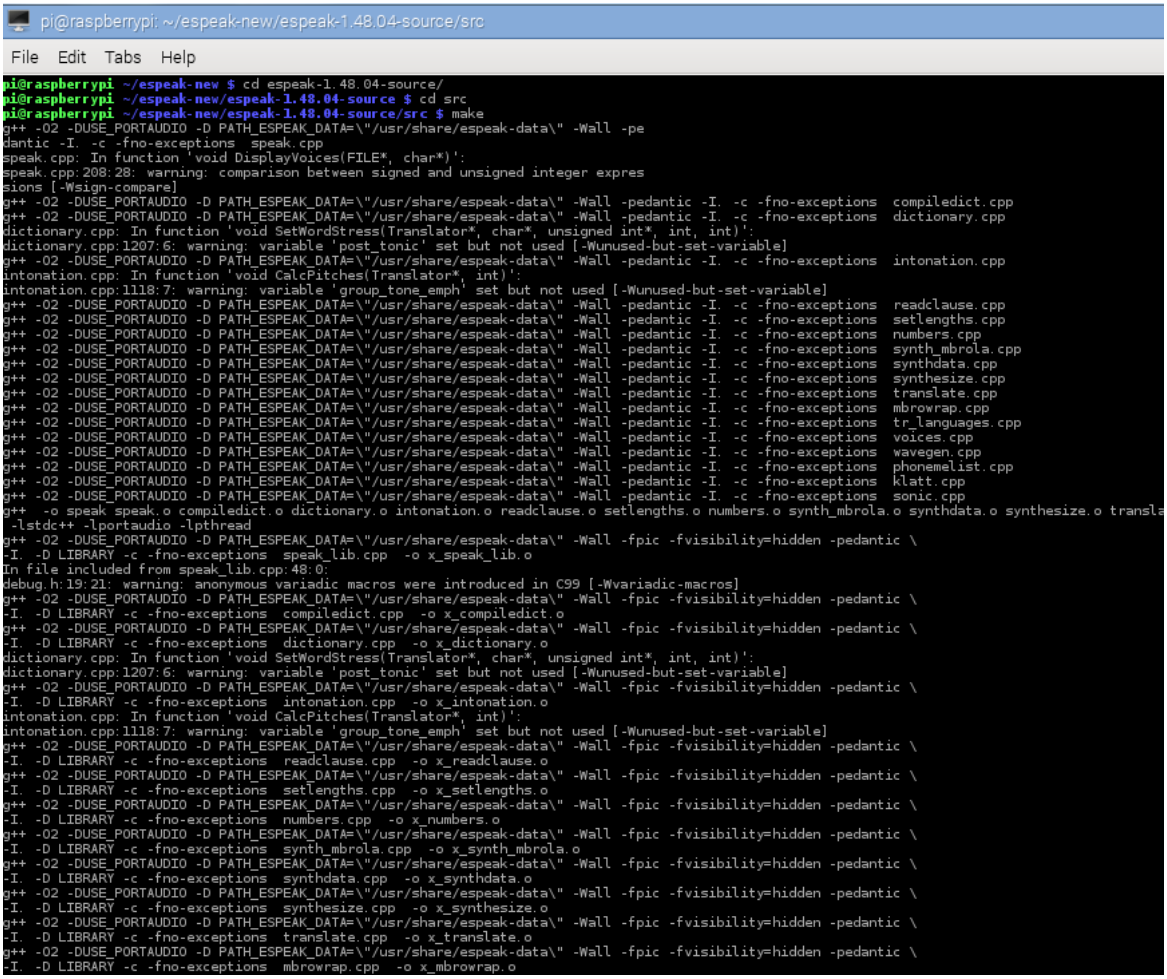


Figure 137: Install espeak library command

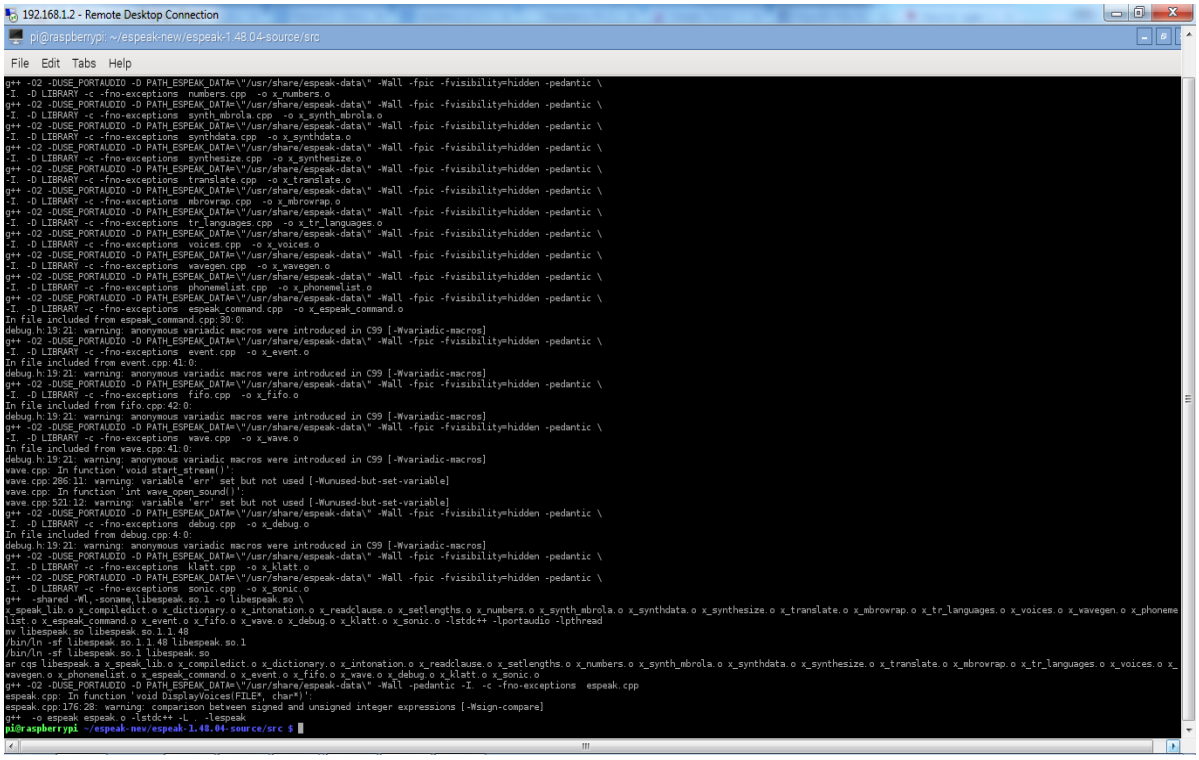


Figure 138: Install espeak library command

- Install QT 4 on Raspbbery PI B2 by following the instructions at "Description" section in this webpage http://www.engineersgarage.com/embedded/raspberry-pi/how-to-install-qt-in-raspberry-pi#

1.2 Deployment at Raspberry PI B2

Step 1: Prepare a deploy.zip file, which contains one "Database" folder and one "MainCapstoneProject" folder on the laptop.

```
nickseven@nickseven:~$ scp /home/nickseven/nhom05/deploy.zip pi@192.168.1.2:/hom
e/pi/SRC
```

Figure 139: Copy this .zip file to Raspberry PI B2 command

Step 2: Copy this .zip file to Raspberry PI B2 at location /home/pi/SRC via scp

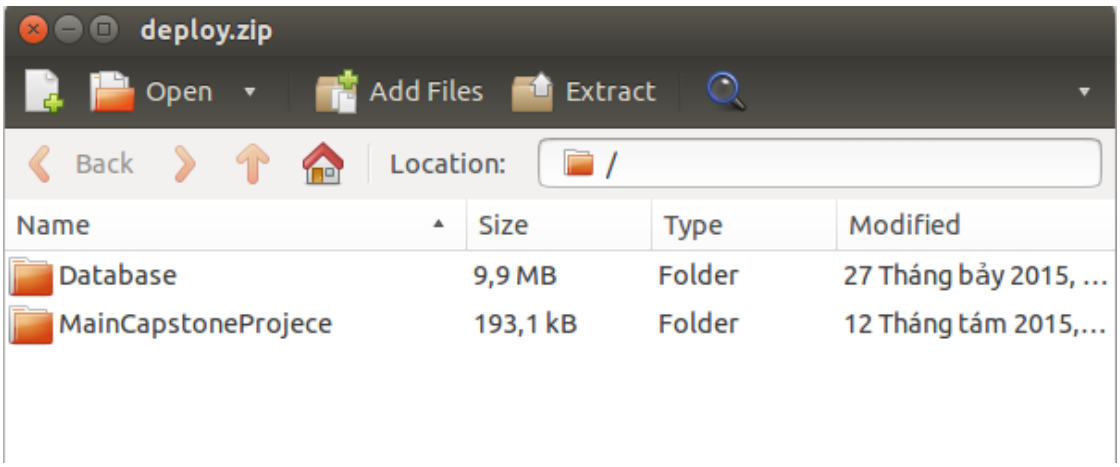


Figure 140: deploy.zip

Step 3: Uncompress this file by the following command on Raspbbery PI 2

cd /home/pi/SRC

unzip project.zip

Step 4: Reboot Raspberry PI B2

2. User Guide

2.1 Instructions for use in hardware

2.1.1 Turn On - Off System

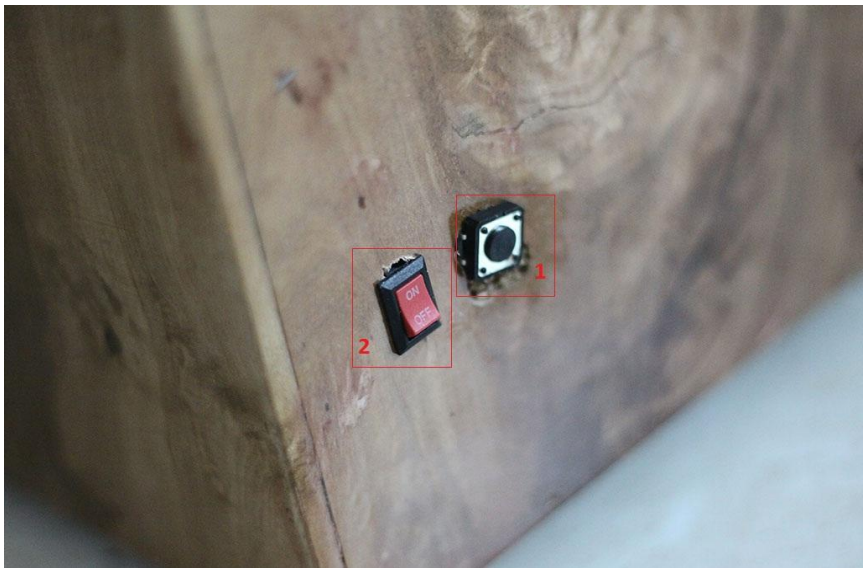


Figure 141: Right side of the box

Step	Description
1	Switch ON button switch number 2 in the above picture to turn on system.

Table 66: Turn on- off system Steps

2.1.2 Turn off application

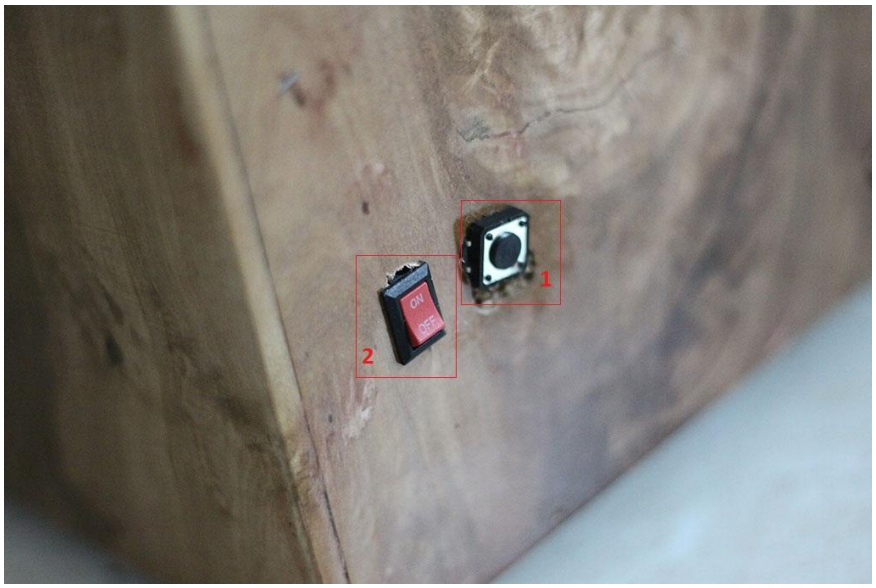


Figure 142: Right side of the box

Step	Description
1	Press black button number 1 in the above picture to turn off application running.

Table 67: Turn off application Steps

2.1.3 Monitor Battery Capacity by Led Indicator

Figure 143: Above side of the box

Step	Description
1	Watch number of bright led on the above side of the hardware box.

Table 68: Monitor Battery Capacity Step

2.1.4 Show hand sign, view application interface

Figure 144: Front side of the box

Step	Description
1	Users show hand sign front of webcam number 2 in the above picture.
2	Users watch application's interface on LCD number 1 in the above picture to keep track of the system process.

Table 69: Show hand sign, view application interface guide

2.2 Instructions for use in software

2.2.1 Conditions for the system to work

User must use bracelet, which is supported by the system, for the system work correctly.

The restrictions in environment:

- Light: must be stable and not changed during the system works.
- Background: must be not complex in color, an especially color which is similar to hand skin color, and background must be fixed.

2.2.2 Special hand signs for use

Users should know the following hand signs to work with the system:

- "Test" hand sign: is used for testing the process "Background Color Subtraction"

Figure 145: "Test" hand sign

- "Select" hand sign: is used for selecting functions "Học" and "Nhận Dạng"

Figure 146: "Select" hand sign

- "End" hand sign: is used for ending function and then navigating to the interface for selecting function.

Figure 147: "End" hand sign

- "Speak" hand sign: is trigger to the system speak the recognition content in the function hand sign language recognition.

Figure 148: "Speak" hand sign

- "Clear" hand sign: is used for clear the last word of recognition content in the function hand sign language recognition.

Figure 149: "Clear" hand sign

2.2.3 Functions

2.2.3.1 Background Color Subtraction

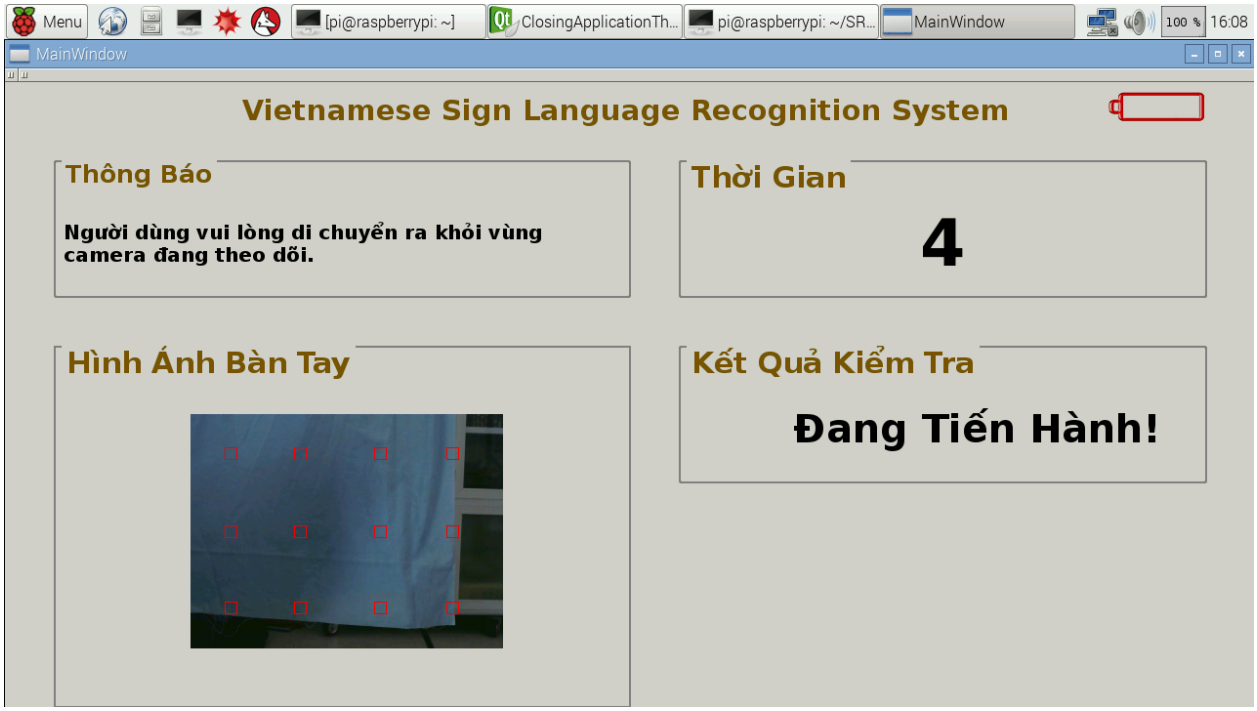


Figure 150: Background Color Subtraction Interface

Step	Description
1	User must wear the accessory such as black bracelet.
2	Turn on power by switch button
3	Users must move outside of the camera area when receiving the notify “Người dùng vui lòng di chuyển ra khỏi vùng camera theo dõi” in the group box “Thông Báo”
4	Users show “test” hand sign over camera area when receiving the notify “Vui lòng điều chỉnh bàn tay của bạn theo kí hiệu kiểm tra trong hướng dẫn.”
5	Users wait for the testing result in the group box “Đang Tiến Hành!”

Table 70: Background Color Subtraction Description

2.2.3.2 Selecting Function

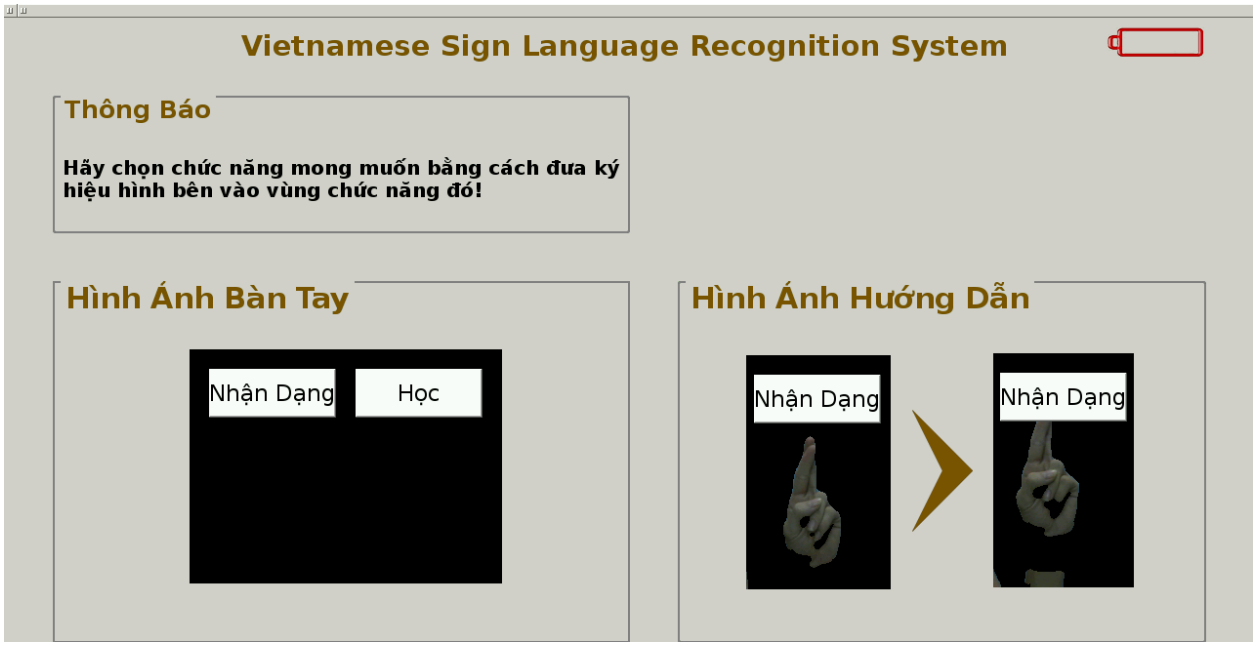


Figure 151: Selecting Function Interface

Step	Description
1	User must wear the accessory such as black bracelet.
2	The system shows the interface for users select function after users finish the phase "Background Color Subtraction"
3	Users show the "select" sign over camera so that users can see your hand in the group "Hình Ảnh Bàn Tay"
4	Move the "select" sign inside the white button "Nhận Dạng" to select function "Hand Sign Language Recognition", otherwise to select function "Learning Hand Sign Language", users should move the "select" sign inside the "Học" area.
5	Users keep the "select" sign inside the white areas until the system navigate to the interface of selected function.

Table 71: Selecting Function Description

2.2.3.3 Hand Sign Language Recognition

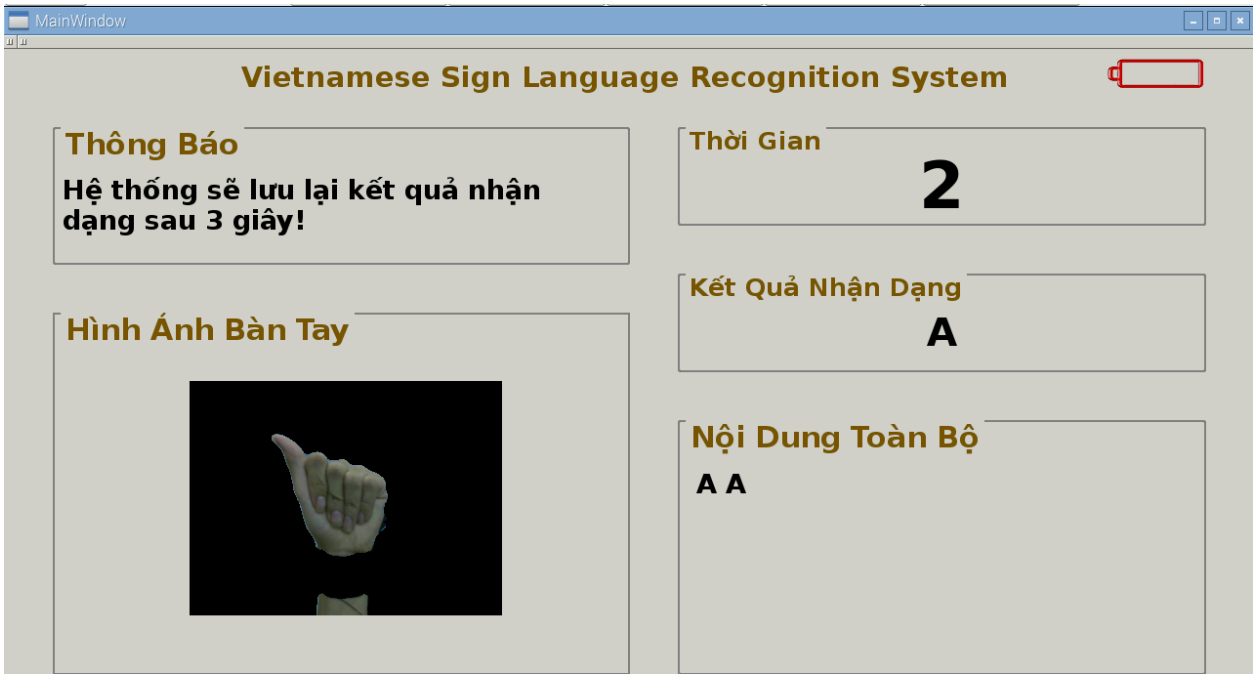


Figure 152: Hand Sign Language Recognition Interface

Step	Description
1	User must wear the accessory such as black bracelet.
2	Users show the hand sign over camera area so that can see your hand in the group "Hình Ảnh Bàn Tay"
3	Users can check the real time recognition result in the group box "Kết Quả Nhận Dạng"
4	Users keep the hand sign until the countdown time is 0 to update the recognition result to the whole content.
5	Users show "speak" hand sign to enable the system to speak the whole content. Note: If users want to clear the previous word of the whole content, users show the "clear" hand sign.
6	Users show "end" hand sign to end this function.

Table 72: Hand Sign Language Recognition Description

2.2.3.4 Learning Hand Sign Language

Figure 153: Learning Hand Sign Language Interface

Step	Description
1	Users must wear the accessory such as black bracelet.
2	Users select the another hand sign by moving the “select” hand sign inside the “Lên” area and “Xuống” area to select upper hand sign and lower hand sign in the list of hand signs inside groupbox “Bảng Ngôn Ngữ Câm” respectively.
3	‘Users show the hand sign as the image shown in the group “Bảng Ngôn Ngữ Câm”
4	Users can check the recognition result inside the groupbox “Kết Quả Nhận Dạng”
5	Users show “end” hand sign to end this function.

Table 73: Learning Hand Sign Language Description

2.2.3.5 Battery Capacity Notify

Figure 154: Battery capacity at the top right corner

Users can check the battery capacity at the top right corner of every interface.

Figure 155: Notify Low Battery Dialog

Users will receive the notify dialog which lets know of the low battery when the battery capacity is under 25%.

Note: The system will shutdown automatically when the battery is nearly out of capacity

G. Appendix

Scaled Professional Scrum. Available at https://www.scrum.org/Resources/What-is-Scaled-Scrum.

Spreadsheet for Sprint backlog & burndown in Agile/Scrum. Available at https://drive.google.com/previewtemplate?id=0ApJnj7ySevnidDFJVXN5eUZLakFBbkhvSkoxbk1ET3c&mode=public

Geotechnical Software Services. C++ Programming Style Guidelines (2011). Available at http://geosoft.no/development/cppstyle.html

Technische Informatica, Eindhoven University of Technology, Eindhoven. Integration Test Plan (29 May 2006). Available at http://wwwis.win.tue.nl/2R690/projects/spingrid/itp.pdf

Raspberry PI 2 Model B. Available at https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

Turnigy 2700mAh 3S 20C Lipo Pack (Suitable for Quanum Nova, Phantom, QR X350). Available at http://hobbyking.com/hobbyking/store/__56095__Turnigy_2700mAh_3S_20C_Lipo_Pack_Suitable_for_Quanum_Nova_Phantom_QR_X350_.html

Texas Instruments. LM2576/LM2576HV Series SIMPLE SWITCHER® 3A Step-Down Voltage Regulator (April 2013). Available at http://www.national.com/ds/LM/LM2576.pdf

ImaxRC B3 Compact charger. Available at http://www.imaxrc.com/B3-Compact.html

TL084, TL084A, TL084B. General purpose JFET quad operational amplifiers. Datasheet — production data. Available at http://www.st.com/web/en/resource/technical/document/datasheet/CD00000493.pdf

1N4728A - 1N4764A Zenners. Available at http://pdf1.alldatasheet.com/datasheet-pdf/view/94866/FAIRCHILD/1N4733A.html

Logitech HD Webcam C270. Available at http://www.logitech.com/en-us/product/hd-webcam-c270

LCD - Tontec® 7 Inch HD TFT Color Monitor. Available at http://goodmonitor.elektroshop91.com/items/Amazon@1/216.html

LM2576ADJ-Board. Available at http://www.tme.vn/Product.aspx?id=1092#page=pro_info

Gaussian Blur Algorithm. Available at http://www.pixelstech.net/article/1353768112-Gaussian-Blur-Algorithm

Wikipedia. Gaussian blur. Available at https://en.wikipedia.org/wiki/Gaussian_blur

OpenCV. Image Thresholding. Adaptive Thresholding. Available at http://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html#gsc.tab=0

Chih-Chung Chang and Chih-Jen Lin. LIBSVM -- A Library for Support Vector Machines. Available at https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html?js=1#svm-toy-js

Chih-Chung Chang and Chih-Jen Lin. A Library for Support Vector Machines (March 4, 2003). Available at http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf

Wikipedia. Support vector machine. Available at https://en.wikipedia.org/wiki/Support_vector_machine

OpenCV. Introduction to Support Vector Machines. Available at http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

Wikipedia. RGB color model. Available at https://en.wikipedia.org/wiki/RGB_color_model

Explanation of the LAB Color Space . Available at http://www.aces.edu/dept/fisheries/education/pond_to_plate/documents/ExplanationoftheLABColorSpace.pdf

Radu Gabriel Danescu. Morphological operations on binary images. Available at http://users.utcluj.ro/~rdanescu/PI-L7e.pdf

Eugene Vishnevsky. RGB to XYZ & XYZ to RGB , XYZ to CIE L*a*b* (CIELAB) & CIELAB to XYZ. Available at http://www.cs.rit.edu/~ncs/color/t_convert.html

Theo Pavlidis. Algorithms for Graphics and Image Processing in 1982

Melkman. Melkman 1987 convex hull algorithm for simple polygons. Available at http://cgm.cs.mcgill.ca/~athens/cs601/Melkman.html

Nearest Neighbor Image Scaling. Available at http://tech-algorithm.com/articles/nearest-neighbor-image-scaling/
Wikipedia. Convex hull. Available at https://en.wikipedia.org/wiki/Convex_hull