Module Interface Specification for Mechtronics Enigeering

Team 32, Wingman, SmartVault
Edward He
Erping Zhang
Guangwei Tang
Peng Cui
Peihua Jin

January 18, 2023

1 Revision History

Date	Version	Authors
2023-01-18	1.0	Edward He, Erping Zhang, Guangwei Tang, Peng Cui, Peihua Jin

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at https://github.com/Edwardhyw/smartVault/tree/main/docs/SRS

Contents

3 Introduction

The following document details the Module Interface Specifications for SmartVault, a Mechatronics system that aims to assist users in finding their belongings

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/Edwardhyw/smartVault.

4 Notation

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

The following table summarizes the primitive data types used by Mechtronics Enigeering.

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

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

5 Module Decomposition

The Whole projet is decomposed into two main part: Hardware Module and Software Module. The Hardware Module is mainly designed for motor control and Hardware-Software Communication. The Software Module is designed for main human interface. It is the main part of the design of the project. These two modules are further divided into seven parts. The first part of the decomposition is Communication Port 2 which is used for sending signals to the software part of the project. The second decomposition of Hardware Module is

Motor Control, it is used to control the real-time position of the camera so that the human body is within the center of the screen taken by the camera.
When it comes to the Software Module, The first part is Login Module, it is designed ask the user log into the running program. It can also be treated as the first barrier of protection of the user's privacy. If the person does not enter the correct username and password, he or she cannot enter the search window, which means the private information is blocked from the user. The second part is the Information Storage Module. It is used to store the screenshot of the images taken by the camera, which shown the position information of the object. The third part is Image Processing Module. It helps analyze the image to get the key information that is useful to the user. Currently this module contains human body detection method and object movement method. The forth part is Information Extraction Module. It is related to the Information Storage Module and is used to choose pictures from the database which meets the information required by the user. The fifth part decomposed from the Software Module is Communication Port 1. It can be treated as the connection bbridge between the hardware software.

Level 1	**Level 2**
Software-Module	Login Information Storage Image Processing Information Extraction Communication Port 1
Hardware-module	Communication Port 2 Motor Control

Table 1: Module Hierarchy

6 MIS of Login Module

6.1 Uses

N/A

6.2 Syntax

6.2.1 Constants

Name of Constants	Value
Username	Determined by the user.
Password	Determined by the user.
Technical Support Window Geometry	450*300
Login Window Geometry	450*300

6.2.2 Access Programs

Name	Input	Output	Description
submitact	N/A	N/A	This is used to get the input of username and password. By verifying the correctness it decides if the user can enter the search window.
tec_create	N/A	N/A	This is used to create the Technical Support Window.
login_create	N/A	N/A	This is used to create the main Login Window.

6.3 Semantics

6.3.1 State Variables

N/A

6.3.2 Environment Variables

Environment Variables	Description
Input Username	The input value in Username Box
Input Password	The input value in Password Box

6.3.3 Assumptions

N/A

6.3.4 Access Routine Semantics

User_input.get():

- transition: N/A
- output: The input value of username entered by user.
- exception: N/A

Pass_input.get()

- transition: N/A
- output: The input value of password entered by user.
- exception: N/A

Tk()

- transition: Creating a new window.
- output: The output window.
- exception: N/A

tk.label()

- transition: Adding text on the window.
- output: The text that presented on the window.
- exception: N/A

text.place()

- transition: Placing the text at desired location on the window.
- output: N/A
- exception: N/A

6.3.5 Local Functions

N/A

7 MIS of Information Storage Module

7.1 Uses

- Image Processing Module
- Information Extraction Module

7.2 Syntax

7.2.1 Constants

maxNumOfStorage: The constant is set to ensure the storage to be small and keep the data precise and clear.

7.2.2 Exported Access Programs

Name	Description
checkRecord	check object in frame is on record or needs a new entry

7.3 Semantics

7.3.1 State Variables

- objectMotion: Signal received from Image Processing Module
- photoTakenDate: Variables used to record when the picture is captured
- recordT: Record the picture and move it to the storage
- recordF: Do nothing

7.3.2 Environment Variables

N/A

7.3.3 Assumptions

- The module may not always capture the expected information. Since the product is designed to assist people to make their room organized instead of replacing them to memorize everything they lost.

- photoTakenDate: The system shall have good network signal during operation

7.3.4 Access Routine Semantics

objectMotion:

- transition: if true
- output: start checkRecord
- exception:

checkRecord:

- transition: on record
- output: update object position
- exception: N/A
- transition: not on record
- output: add a new entry and record position
- exception: N/A

7.3.5 Local Functions

- photoT(): This function will take the photo once the command is operated.
- photoNot(): This function will not take the photo even if the command is operated.

8 MIS of Information Extraction Module

8.1 Uses

information Extraction Module

8.2 Syntax

8.2.1 Constants

- byColor: Click the button and search by color.
- bySize: Click the button and search by size.
- byShape: Click the button and search by shape.
- maxLoop: To ensure the infinite loop occur, the system will stop once it reach the max limitation.

8.2.2 Access Programs

Name	Description
enterTime	User enters the last time the object used
displayConformation	window for displaying result

8.3 Semantics

8.3.1 State Variables

- timeEntered: boolean variable for whether user has enter a time or not.
- attr_color: The color of the desired item.
- attr_size: The size of the desired item.
- attr_shape: The shape of the desired item.

8.3.2 Environment Variables

searchButton: button for entering the time. User can use this button with empty message.

8.3.3 Assumptions

- The module may not always find out the exactly matched object. Since the attributes are not large enough and the training for machine learning is not mature enough to do complicated computation.

8.3.4 Access Routine Semantics

timeEntered():

- transition: if empty
- output: default value of 1, search result becomes sorting of most recent records.
- exception: N/A
- transition: if not empty
- output: send the time value to find corresponding record
- exception: N/A

searchButton():

- transition: if clicked
- output: displaceConformation
- exception: N/A

8.3.5 Local Functions

- research_color(): The function of which could search the desired item about the specific color.
- research_size(): The function of which could search the desired item about the specific size.
- research_shape(): The function of which could search the desired item about the specific shape.

9 MIS of Image Processing Module

9.1 Uses

Communication Port 1

9.2 Syntax

9.2.1 Constants

Name of Constants	Value
min_detection_confidence	0.5
Search Window Geometry	1500*800

9.2.2 Access Programs

Name	Input	Output	Description
__init__	self, window, cap, label	N/A	set the initial condition of the module
hand_detect	self	N/A	detect the hand of the human ody from the image.

9.3 Semantics

9.3.1 State Variables

Name	textbfvalue	Description
window	N/A	The Search Window
window.title	Main Window	The title of teh Search Window in the program.
hand_detect	N/A	The hand detection method.

9.3.2 Environment Variables

N/A

9.3.3 Assumptions

N/A

9.3.4 Access Routine Semantics

humanDetected():

- transition: If True

- output: start object motion detection and send signal to Communication Transmit Module
- exception: N/A

objectMotion():

- transition: If True
- output: send signal to Information Storage Module
- exception: N/A

img.resize():

- transition: Resizing the image taken by the camera.
- output: The image after resizing.
- exception: N/A

cv2.flip():

- transition: Flip the whole image taken by the camera.
- output: The output image after fliping.
- exception: N/A

cv2.cvtColor():

- transition: The color of the image is converted from BGR to RGB.
- output: The output image after color conversion.
- exception: N/A

mp_drawing.draw_landmarks():

- transition: The shape of hand is shown on image.
- output: The output image after the shape of hand has been detected in the image.
- exception: N/A

9.3.5 Local Functions

[As appropriate —SS] [These functions are for the purpose of specification. They are not necessarily something that is going to be implemented explicitly. Even if they are implemented, they are not exported; they only have local scope. —SS]

References

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

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

10 MIS of Communication Port 1 Module

10.1 Uses

Communication Port 2

10.2 Syntax

10.2.1 Constants

N/A

10.2.2 Access Programs

N/A

10.3 Semantics

10.3.1 State Variables

connectionCheck: Boolean variable for connection between software component and hardware component

10.3.2 Environment Variables

10.3.3 Assumptions

N/A

10.3.4 Access Routine Semantics

connectCheck:

- transition: if True
- output: send signal and data to Communication Port 2 Module
- exception: N/A

10.3.5 Local Functions

N/A

References

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

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

11 MIS of Communication Port 2 Module

11.1 Uses

Communication Port 2 Module

11.2 Syntax

11.2.1 Constants

N/A

11.2.2 Access Programs

N/A

11.3 Semantics

11.3.1 State Variables

connectionCheck: Boolean variable for connection between software component and hardware component

11.3.2 Environment Variables

11.3.3 Assumptions

N/A

11.3.4 Access Routine Semantics

connectCheck:

- transition: if True
- output: send signal and data to Communication Port 1 Module
- exception: N/A

11.3.5 Local Functions

N/A

References

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

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

12 MIS of Motor Control Module

12.1 Uses

Communication Port 2

12.2 Syntax

12.2.1 Constants

Table 3: Constants Variables				
Constant Name	Constant Type	Value	Units	Comment
Angle per step	float	TBD	Degree/step	This is the angle movement stepper motor will move after 1 signal
Height of the Camera	float	TBD	mm	This is the distance between the lens of camera and the bottom of the mount
Resolution	Integer	1920x1080	Pixel	This is the resolution of the camera
Arduino input voltage	float	9.0	V	This is the input voltage of the Arduino board

12.2.2 Access Programs

Name	Description
positionMotor	rotate motor to reposition camera according to the data send from the software module.

12.3 Semantics

12.3.1 State Variables

dataRecieved: Boolean variable representing whether data has received.
posistionReached: Boolean variable representing whether camera has reached desired position

12.3.2 Environment Variables

12.3.3 Assumptions

N/A

12.3.4 Access Routine Semantics

dataRecieved():

- transition: if True
- output: positionMotor
- exception: N/A

positionReached():

- transition: if True
- output: signal Communication Port 2 to send video frames to Communication Port 1
- exception: N/A

12.3.5 Local Functions

N/A

References

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

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

13 Appendix

[Extra information if required —SS]

References

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

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