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	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at [give url —SS]

[Also add any additional symbols, abbreviations or acronyms —SS]

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 —SS]

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

[You should describe your notation. You can use what is below as a starting point. —SS]

The structure of the MIS for modules comes from **?**, with the addition that template modules have been adapted from **?**. The mathematical notation comes from Chapter 3 of **?**. 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 following table is taken directly from the Module Guide document for this project.

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 Exported Constants

N/A

6.2.2 Access Programs

Name	**Description**
loginInfo	Ask user to enter username and password

6.3 Semantics

6.3.1 State Variables

N/A

6.3.2 Environment Variables

technicalSupportButton: Button for technical support and contact informations

6.3.3 Assumptions

N/A

6.3.4 Access Routine Semantics

loginInfo:

- transition: corrext username and password entered
- output: display searching interface and transition into information extraction module
- exception: N/A

technicalSupportButton

- transition: Button clicked
- output: display technical support screen
- exception: N/A

6.3.5 Local Functions

N/A

7 MIS of Information Storage Module

7.1 Uses

Image Processing Module

7.2 Syntax

7.2.1 Constants

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

7.3.2 Environment Variables

7.3.3 Assumptions

[Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate. —SS]

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

[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]

8 MIS of Information Extraction Module

8.1 Uses

information Extraction Module

8.2 Syntax

8.2.1 Constants

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.

8.3.2 Environment Variables

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

8.3.3 Assumptions

[Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate. —SS]

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

[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]

9 MIS of Image Processing Module

9.1 Uses

Communication Port 1

9.2 Syntax

9.2.1 Constants

N/A

9.2.2 Access Programs

N/A

9.3 Semantics

9.3.1 State Variables

humanDetected: Boolean variable representing whether human is detected in the frame
objectMotion: Boolen variable representing whether an object is moved by human

9.3.2 Environment Variables

N/A

9.3.3 Assumptions

[Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate. —SS]

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

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]

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

[Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate. —SS]

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

[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]

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

[Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate. —SS]

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

[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]

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

[Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate. —SS]

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

[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]

13 Appendix

[Extra information if required —SS]