Module Interface Specification for Mechatronics Engineering

Team #1, Back End Developers
Jessica Bae
Oliver Foote
Jonathan Hai
Anish Rangarajan
Nish Shah
Labeeb Zaker

January 18, 2023

1 Revision History

Date	Version	Notes
January 18th, 2023	1.0	Initial Documentation

2 Symbols, Abbreviations and Acronyms

Please refer to the System Requirements Specifications document at this link for relevant symbols, abbreviations.

Contents

3 Introduction

The following document details the Module Interface Specifications for the EMAnator; the system currently being developed by the Back End Developers designed to aid in Ecological Momentary Assessment research. This document describes the various relevant details of interfacing with each module. These details include module descriptions, the uses of each module, the syntax of each module, and the semantics associated with each module.

Complementary documents include the System Requirement Specifications and Module Guide. The Back End Developers highly recommend a thorough read-through of each document prior to a reading of this document to attain the prerequisite knowledge necessary to fully understand this MIS. The System Requirements Specifications can be found at this link, and the Module Guide can be found at this link.

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 Mechatronics Engineering.

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 Mechatronics Engineering 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, Mechatronics Engineering 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**
Hardware-Hiding Module	Device Manager Data Storage Sensor Array
Behaviour-Hiding Module	Display System Prompt Generation Real Time Clock
Software Decision Module	Moving Average Algorithm Graph Plotter

Table 1: Module Hierarchy

6 MIS of Physical Design Module

6.1 Module

Physical_Design

6.2 Uses

None.

6.3 Syntax

6.3.1 Exported Constants

None.

6.3.2 Exported Access Programs

None.

6.4 Semantics

6.4.1 State Variables

None.

6.4.2 Environment Variables

None.

6.4.3 Assumptions

Will not impact the the functionality of any other components.

6.4.4 Access Routine Semantics

None.

6.4.5 Local Functions

None.

7 MIS of Battery Management Module

7.1 Module

Bat_Man

7.2 Uses

None.

7.3 Syntax

7.3.1 Exported Constants

Name	In	Out	Exceptions
Disp_Flag	-	bool	-
MPU_Flag	-	bool	-
RTC_Flag	-	bool	-
HR_Flag	-	bool	-
Touch_Flag	-	bool	-

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
Bat_State	Bat_Select	$\mathbb{Z}$ (tuple)	BED_ERR_BAT

7.4 Semantics

7.4.1 State Variables

None.

7.4.2 Environment Variables

None.

7.4.3 Assumptions

System responds instantaneously to changes in flags (exported constants).

7.4.4 Access Routine Semantics

None.

7.4.5 Local Functions

None.

8 MIS of Device Manager Module

8.1 Module

Dev_Man

8.2 Uses

Data Storage Module (Section 9)

8.3 Syntax

8.3.1 Exported Constants

None.

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
SD_Data	-	Card_IO: string	BED_ERR_SD: $\mathbb{Z}$
Host_Data	Host_IO: string	Extract_Data: string	BED_ERR_SD: $\mathbb{Z}$

8.4 Semantics

8.4.1 State Variables

None.

8.4.2 Environment Variables

Extract_Data: string which outputs to supporting software on researcher's computers.

8.4.3 Assumptions

- SD card is formatted correctly.
- SD card is inserted correctly.
- Valid connection to host computer is present.

8.4.4 Access Routine Semantics

None.

9 MIS of Data Storage Module

9.1 Module

Data_Stor

9.2 Uses

Sensor Array Module (Section 10), Real Time Clock Module (Section 13)

9.3 Syntax

9.3.1 Exported Constants

None.

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
Card_Read	-	Read_Data: string	BED_ERR_SD: $\mathbb{Z}$
Card_Write	Write_Data: $\mathbb{Z}$ (tuple)	Write_Flag: bool	BED_ERR_SD: $\mathbb{Z}$

9.4 Semantics

9.4.1 State Variables

None.

9.4.2 Environment Variables

file: A text file.

9.4.3 Assumptions

- SD card is formatted correctly.
- SD card is inserted correctly.

9.4.4 Access Routine Semantics

None.

9.4.5 Local Functions

None.

10 MIS of Sensor Array Module

10.1 Module

Sensor_Array

10.2 Uses

Data Storage Module (Section 9), Moving Average Algorithm Module (Section **??**)

10.3 Syntax

10.3.1 Exported Constants

10.3.2 Exported Access Programs

None.

10.4 Semantics

10.4.1 State Variables

pedometer_count : $\mathbb{R}$
heartrate : $\mathbb{R}$
prompt_interrupt : bool
touch_input : bool (tuple)

10.4.2 Environment Variables

None.

10.4.3 Assumptions

- All activity thresholds are provided from the configuration file.
- There is available space on the SD card.

10.4.4 Access Routine Semantics

activity_sensed():

- transition: change the state of the Prompt Generation Module in order to send a prompt to the user which will be sent to the display.

- exception: there is already a prompt being displayed to the user OR not enough time has elapsed between the prompts.

10.4.5 Local Functions

data_smoothing_filter(): the purpose of this function is to make sure that the data coming from all the sensors is smoothed, in order to prevent a prompt from being generated erroneously and distrubing the user.

11 MIS of Display System Module

11.1 Module

Disp_Sys

11.2 Uses

Prompt Generation Module (Section 12), Real Time Clock Module (Section 13)

11.3 Syntax

11.3.1 Exported Constants

None.

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
Disp_Time	-	-	BED_ERR_DISP: $\mathbb{Z}$
Disp_Prompt	Prompt: string	Response: string	BED_ERR_DISP: $\mathbb{Z}$
Switch_Window	Window: $\mathbb{Z}$	-	BED_ERR_DISP: $\mathbb{Z}$

11.4 Semantics

11.4.1 State Variables

None.

11.4.2 Environment Variables

None.

11.4.3 Assumptions

None.

11.4.4 Access Routine Semantics

None.

11.4.5 Local Functions

None.

12 MIS of Prompt Generation Module

12.1 Module

Prompt_Gen

12.2 Uses

Sensor Array Module (Section 10)

12.3 Syntax

12.3.1 Exported Constants

$max\_prompts : \mathbb{Z}$

12.3.2 Exported Access Programs

Name	In	Out	Exceptions
Access_Prompt	Prompt_: $\mathbb{Z}$	Prompt: Struct	BED_ERR_PG: $\mathbb{Z}$

12.4 Semantics

12.4.1 State Variables

None.

12.4.2 Environment Variables

None.

12.4.3 Assumptions

None.

12.4.4 Access Routine Semantics

None.

12.4.5 Local Functions

None.

13 MIS of Real Time Clock Module

13.1 Module

RTC

13.2 Uses

None.

13.3 Syntax

13.3.1 Exported Constants

None.

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
Get_DateTime	-	$\mathbb{R}$	BED_ERR_RTC: $\mathbb{Z}$

13.4 Semantics

13.4.1 State Variables

Date: string
Time: string

13.4.2 Environment Variables

None.

13.4.3 Assumptions

- Initial Date and Time is correctly set.

13.4.4 Access Routine Semantics

None.

13.4.5 Local Functions

None.

14 MIS of Parameter Selection

14.1 Module

Param_Select

14.2 Uses

Device Manager Module (Section 8)

14.3 Syntax

14.3.1 Exported Constants

None.

14.3.2 Exported Access Programs

None.

14.4 Semantics

14.4.1 State Variables

param_input: string

14.4.2 Environment Variables

file: a text file on the SD card.

14.4.3 Assumptions

- All configuration parameters within acceptable limits.

14.4.4 Access Routine Semantics

None.

14.4.5 Local Functions

None.

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.