3K04 Deliverable 2: Documentation

Group 33

Last Updated: 2025-11-27

Contents

List of Figures

List of Tables

1 Group Members

Table 1: Table of Group Members

Name	MacID	Student Number
Ryan Su	sur21	400507973
Cameron Lin	lin422	400535393
Braden McEachern	mceacb1	400527617
Damian Szydlowski	szydlowd	400512629
Menakan Thamilchelvan	thamilcm	400510755

2 Abbreviations

2.1 General Abbreviations

BPM - Beats Per Minute

CCS - Cardiac Conduction System

DCM - Device Controller-Monitor

GPIO - General Purpose Input Output

GUI - Graphical User Interface

PWM - Pulse Width Modulation

2.2 Bradycardia Operating Abbreviations

Table 2: Bradycardia Operating Abbreviations

Category	**Chambers Paced**	**Chambers Sensed**	**Response to Sensing**	**Rate Modulation**
Letters	O-None	O-None	O-None	R-Rate Modulation
	A-Atrium	A-Atrium	T-Triggered	
	V-Ventricle	V-Ventricle	I-Inhibited	
	D-Dual	D-Dual	D-Tracked	

3 Part 1

3.1 Introduction

It is hard to understate the importance of the human heart. The heart is the core part of the cardiovascular system; supplying nutrients and oxygen to all the cells and removing carbon dioxide, especially to vital organs such as the brain, it is imperative for it to be working flawlessly and harmoniously at all times. Unfortunately, however, cardiovascular diseases are a leading cause of death globally, many of which are caused from complications with abnormal heart rhythms. A pacemaker is an implantable device capable of sending timed electrical impulses causing contractions at appropriate intervals. Understanding the operation and design of this life saving device will aid in developing more efficient and reliable cardiac assistive technology.

The purpose of this project is to design and implement a system that operates a cardiac pacemaker under specified modes. This project will be accomplished through an understanding of embedded systems and through engineering principles of software development.

The scope of this deliverable is to design and implement the embedded pacemaker software, driver software and user interface for the DCM while updating and maintaining documentation.

3.2 Requirements

3.2.1 System Requirements

Table 3: System Requirements

ID	Requirement	Rationale
SR1	System Modes	The system shall implement AOO, VOO, VVI, and AAI pacing modes as the basic bradycardia modes required by the spec.
SR2	Hardware Hiding	The system shall use a hardware abstraction layer that maps logical control signals to GPIO pins, which improves maintainability and supports future hardware changes.
SR3	Simulink and DCM Separation	The pacing logic shall be implemented in Simulink and the DCM as a separate GUI so that embedded logic and clinical interface can evolve independently.

3.2.2 Programming Requirements

Table 4: Programming Requirements

ID	Requirement	Description / Rationale
PR1	Programmable Pulse Amplitude	The pacemaker shall generate atrial and ventricular pacing signals with amplitudes configurable by the user. Adjustable amplitudes allow tuning of pacing strength.
PR2	Programmable Pulse Width	The pacemaker shall generate atrial and ventricular signals with pulse widths configurable by the user so timing of stimulation can be adjusted.
PR3	Programmable Rate Timing	The pacemaker shall have programmable lower rate limit (LRL) and upper rate limit (URL) that control the minimum and maximum pacing rates, preventing bradycardia and tachycardia.
PR4	Programmable Refractory Periods	Atrial and ventricular pulse modes shall implement refractory periods during which sensed events are ignored to avoid double sensing and ringing.
PR5	Programmable Sensitivity	A programmable sensitivity threshold for event detection shall be adjustable from the DCM so that sensing can be adapted to patient signals and noise.
PR6	Pacing Responses	Each pacing mode shall follow the response implied by its letters: O for asynchronous pacing that ignores sensing, I for inhibited pacing, and T for triggered pacing from sensed pulses.

3.2.3 Hardware Requirements

Table 5: Hardware Requirements

ID	Requirement	Description / Rationale
HR1	Hardware Hiding Layer	A hardware abstraction layer shall map digital logic signals to analog front end hardware pins so that the pacing model does not reference physical pins directly, improving readability and portability.
HR2	Front End Enable Control	A dedicated signal shall enable or disable the front end sensing circuitry so ADCs and amplifiers can be powered only when needed, reducing noise and power consumption.

3.2.4 DCM Requirements

Table 6: DCM Requirements

ID	Requirement	Description / Rationale
DCM1	User Authentication	The DCM GUI shall provide user registration and login functionality supporting up to ten stored users to control access to the programming interface.
DCM2	Parameter Display and Editing	The DCM shall display and allow editing of pacemaker parameters such as LRL, URL, atrial and ventricular amplitude, pulse width, and sensitivity so that clinicians can program the device.
DCM3	Status Indicators	The DCM shall provide visible indicators of device connection status and communication loss so the user remains aware of system state and faults.

3.2.5 Serial Requirements

NEW SECTION

3.2.6 Egram Requirements

NEW SECTION

3.2.7 Rate Adaptivity Requirements

NEW SECTION

3.3 Design

3.3.1 Simulink Design - Deliverable 1

3.3.1.1 Overall Design

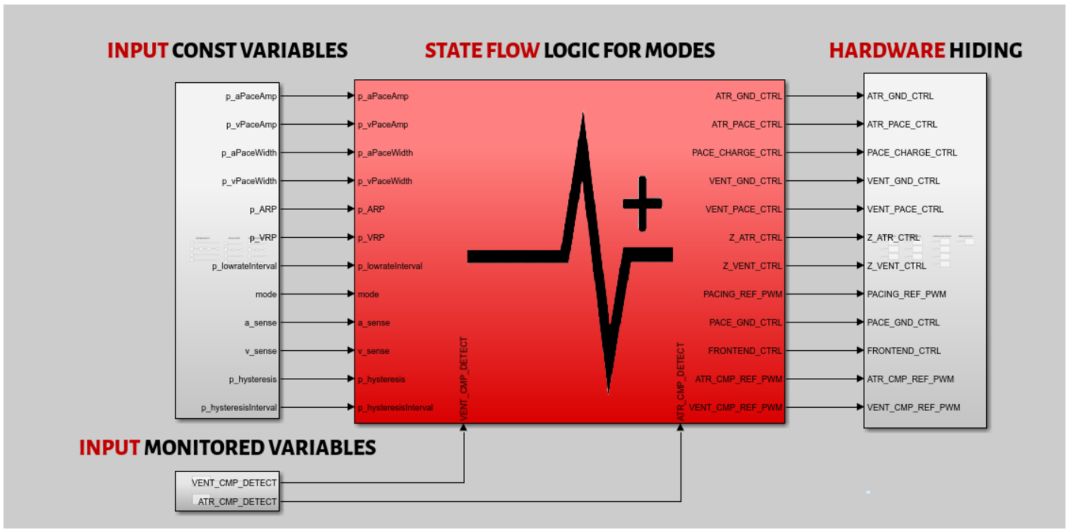


Figure 1: Overall Simulink Mapping

The Pacemaker architecure can be split up into 4 main modules, input constant variables, input monitor variables, stateflow logic for modes, and hardware hiding. Figure 1 below shows the overarcing workflow of the system:

3.3.1.2 Input Constant Variables

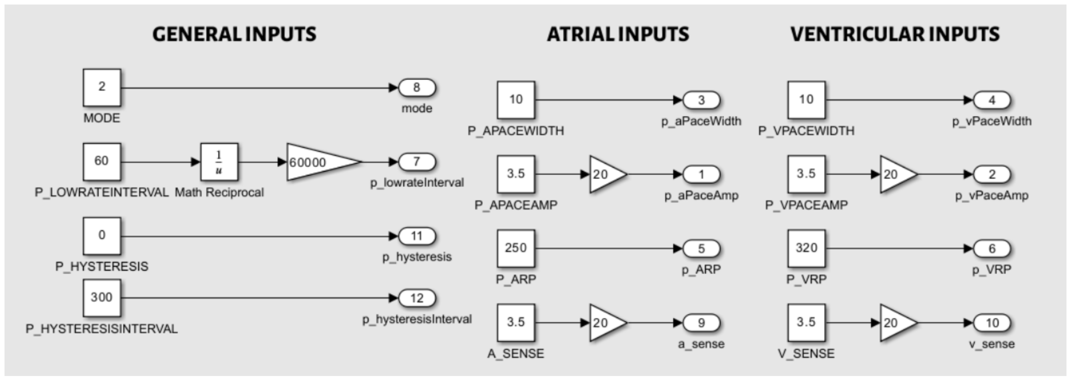


Figure 2: Constant Input Variables

In the above image, Figure 2, we find changeable variables relating to pacemaker operation. For general inputs, the changeable variables are:

- **Mode** - Refers to bradycardia operating modes, e.g AOO, VOO, AAI and VVI.
- **Low Rate Interval** - The number of generated pace pulses per minute, converted from a millisecond time period.
- **Hysteresis Pace** - When enabled, 1, a longer period is waited before pacing after sensing an event to prevent unwanted pacing pulses from ringing from an event.
- **Hysteresis Interval** - Specifies the time interval waited in the hysteresis mode in milliseconds.

The modfiable atrial variables are:

- **Pace Pulse Width** - Changes the width, length of time, of the pace pulse.
- **Pace Pulse Amplitude** - Changes the amplitude, voltage, of the pace pulse.
- **ARP (Atrial Refactory Period)** - The programmed time interval following an atrial event during which time atrial events shall not inhibit nor trigger pacing
- **Sense (Sensitivity)** - Determines the minimum value an atrial signal must be to be considered by the pacemaker.

The modifiable ventricular variables are:

- **Pace Pulse Width** - Changes the width, length of time, of the pace pulse.
- **Pace Pulse Amplitude** - Changes the amplitude, voltage, of the pace pulse.

- **VRP (Ventricle Refactory Period) -** The programmed time interval following an ventricle event during which time atrial events shall not inhibit nor trigger pacing.
- **Sense (Sensitivity) -** Determines the minimum value a ventricle signal must be to be considered by the pacemaker.

3.3.1.3 Monitored Input Variables

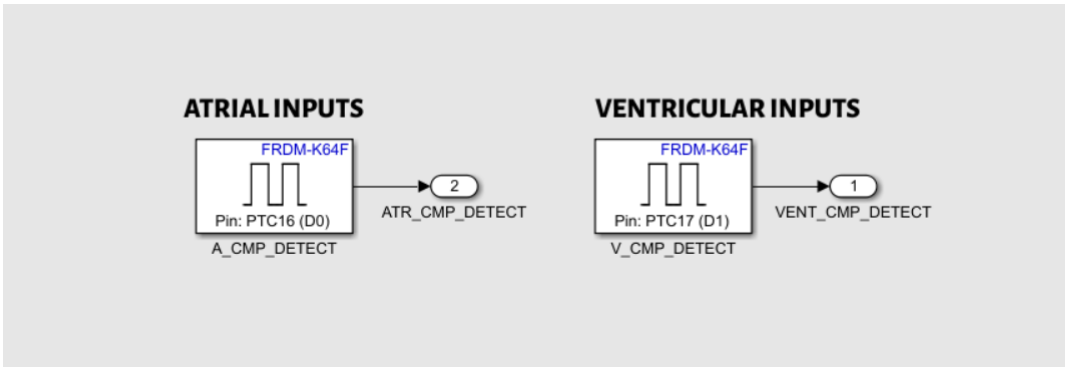
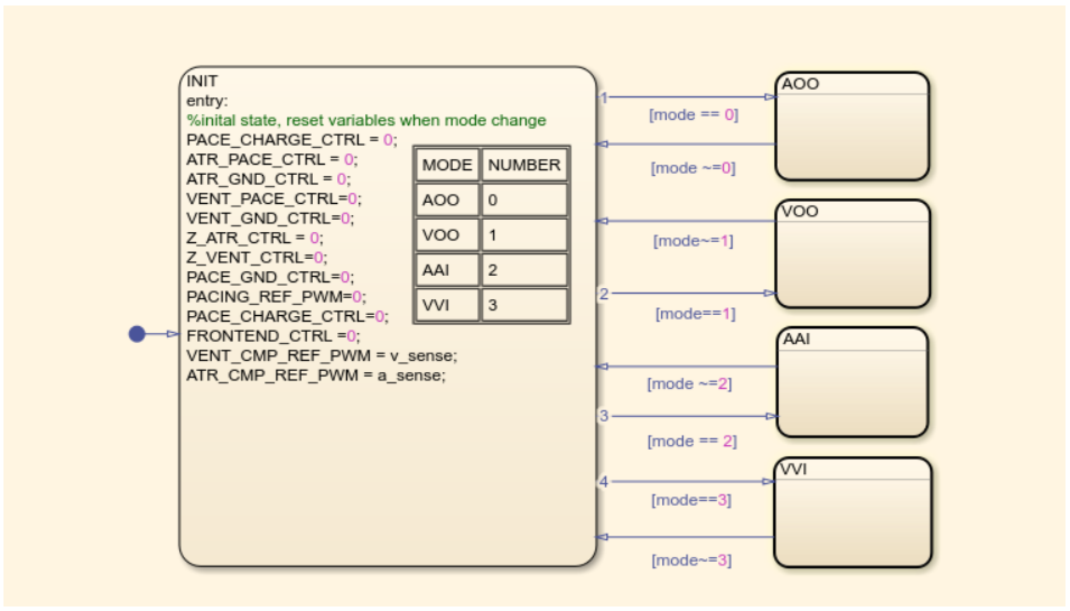


Figure 3: Monitored Input Variables

The monitored input variables can be seen in the above Figure 3. These are the atrial and ventricle detection variables. The pulses are sensed with through GPIO pins connecting to a board simulating heart conditions.

3.3.1.4 Stateflow Modules



```
INIT
entry:
%inital state, reset variables when mode change
PACE_CHARGE_CTRL = 0;
ATR_PACE_CTRL = 0;
ATR_GND_CTRL = 0;
VENT_PACE_CTRL=0;
VENT_GND_CTRL=0;
Z_ATR_CTRL = 0;
Z_VENT_CTRL=0;
PACE_GND_CTRL=0;
PACING_REF_PWM=0;
PACE_CHARGE_CTRL=0;
FRONTEND_CTRL =0;
VENT_CMP_REF_PWM = v_sense;
ATR_CMP_REF_PWM = a_sense;
```

MODE	NUMBER
AOO	0
VOO	1
AAI	2
VVI	3

Figure 4: Stateflow Modules

The stateflow diagram in Simulink is shown above. It shows the transition between each mode given the mode input shown in Figure 2. When switching between modes, a core state is returned to that resets variables to their nominal values.

3.3.1.5 AOO Stateflow Model

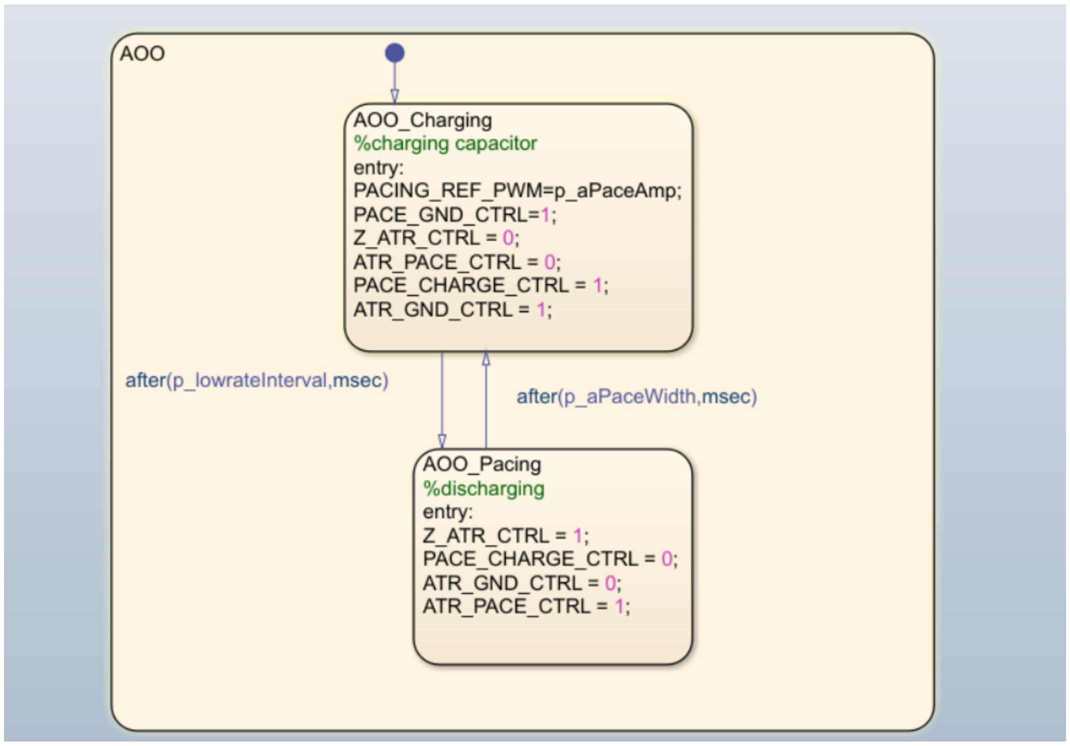


Figure 5: AOO Stateflow Model

The above stateflow, Figure 5, shows the stateflow for the AOO mode. It sets values controlling discharge and charging of the capacitor to specified nominal values.

3.3.1.6 VOO Stateflow Model

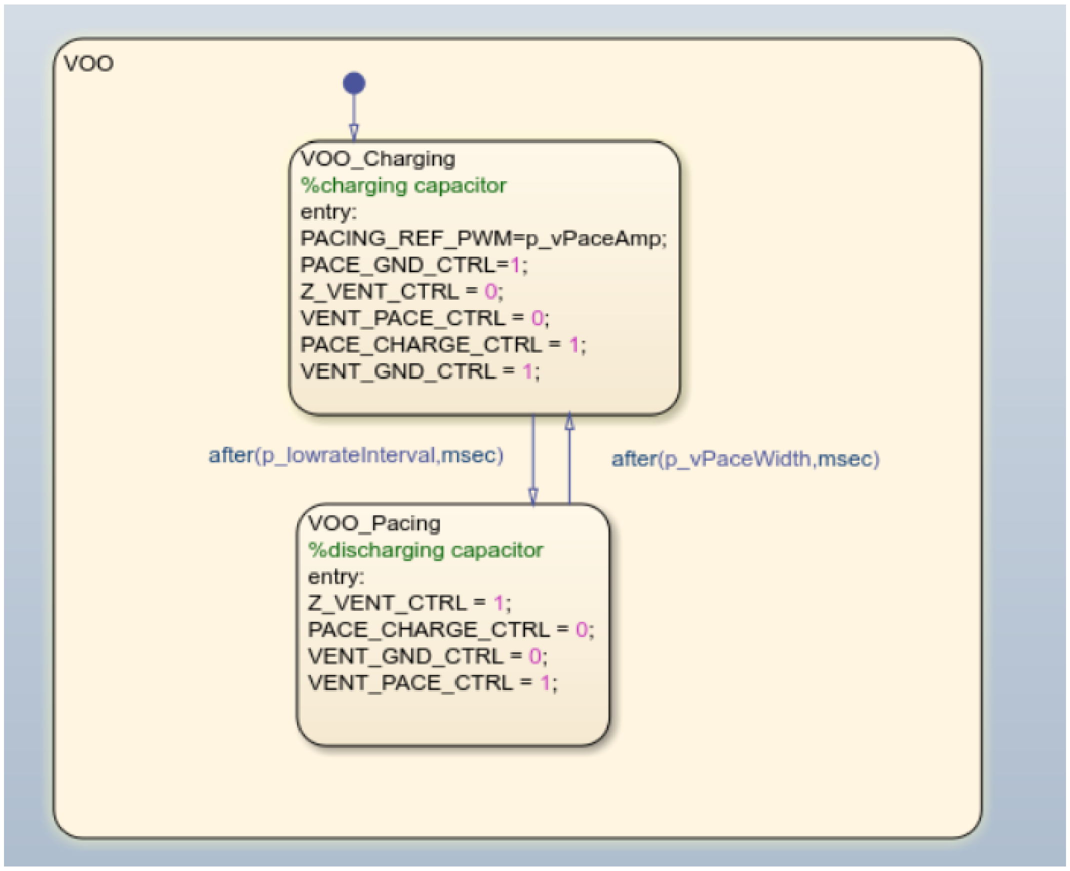


Figure 6: VOO Stateflow Model

Similar to the AOO stateflow, the above figure, Figure 6, shows the states of charging and discharging of the capacitor using specified nominal values.

3.3.1.7 VII Stateflow Model

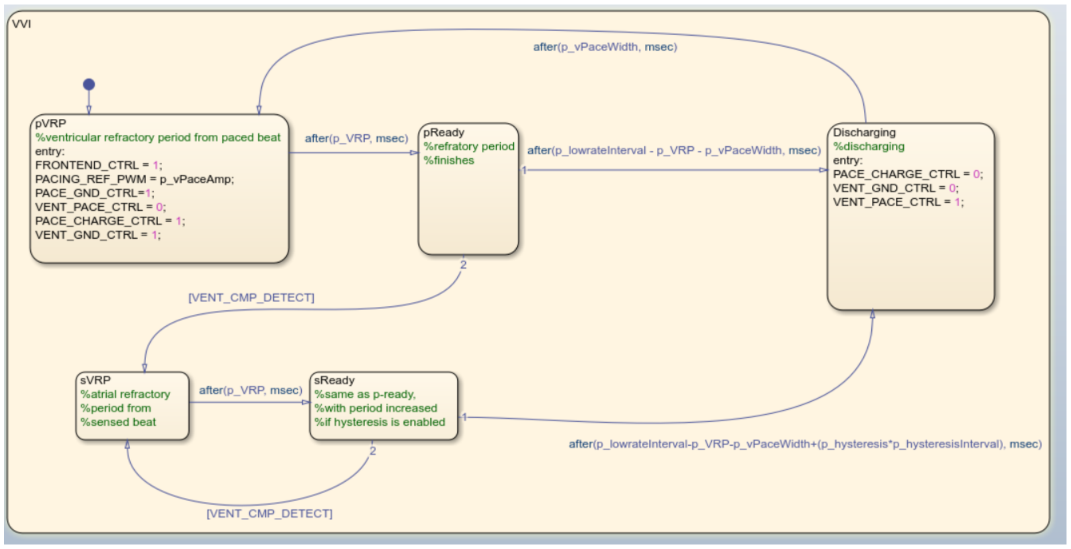


Figure 7: VVI Stateflow Model

The above stateflow, Figure 7 shows the FSM for the VVI model. The initial state, pVRP, is the state that occurs right after the pacemaker delivers a ventricle pacing pulse. A refactory period occurs during this time, where the pacemaker ignores sensor inputs to prevent sensing of its own produced pulse or electrical ringing. After a certain amount of time, the pReady state is transitioned to where sensor inputs are allowed. This allows for the pacemaker to detect natural heart rhythms and correct heart rhythms accordingly, or solely deliver pacing pulses. The last state before going to the initial state is the discharging state where the pacing pulse is produced.

3.3.1.8 AAI Stateflow Model

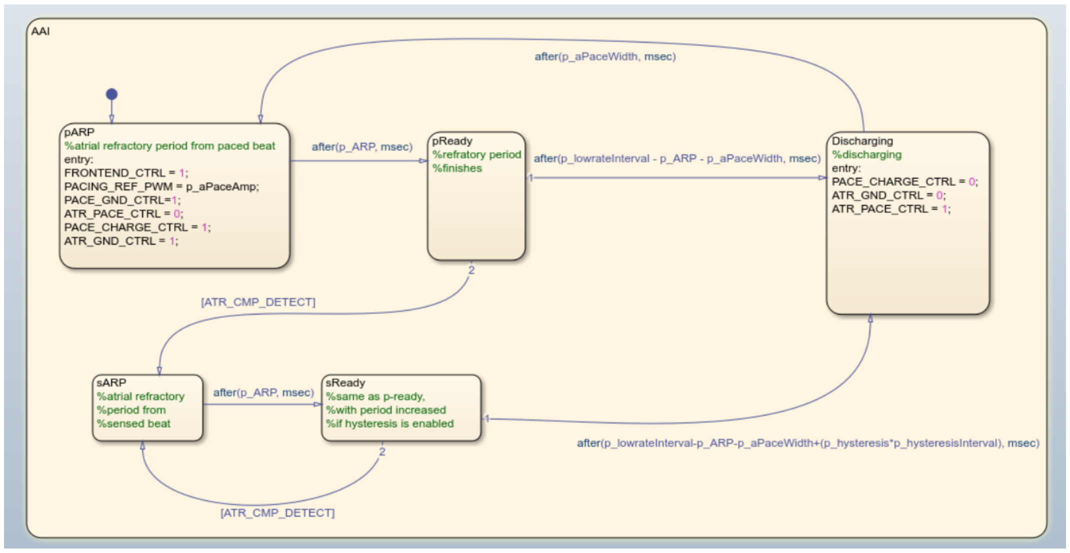


Figure 8: AAI Stateflow Model

The above stateflow, Figure 8 shows the process for the AAI mode. The initial state, pARP, is the state occurs right after the pacemaker delivers an atrial pacing pulse. During this time, the pacemaker ignores sensing events to prevent it from sensing its own delivered pulse. The next state is transitioned to after a set time period, the refactory period, if no natural heartbeat is detected after a certain time inverval, the discharging state is then transitioned to. However, if a natural heartbeat is detected, another refactory period occurs to prevent the pacemaker from sensing ringing. This state then transitions to the discharging state once an appropriate time has passed.

3.3.1.9 Hardware Hiding

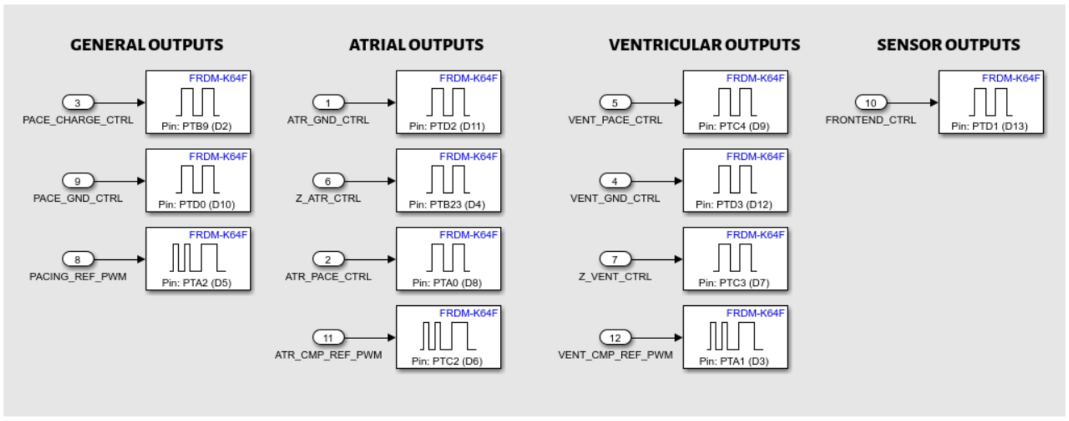


Figure 9: Hardware Hiding of Model

The above image, Figure 9 shows abstraction of the GPIO pin functions. It connects logical output signals to the pins on the FRDM-K64F. This allows for better readability, thus making the code easier to maintain, debug and safer.

3.3.2 Simulink Design - Deliverable 2

3.3.3 Simulink Design - Deliverable 2

3.3.3.1 Serial Packet Structure

With the addition of additional parameters a data conversion table is used to organize the multiple data types being sent. This can be seen in the table below:

Table 7: Parameter Settings

Parameter	**Min**	**Max**	**Inc**	**Unit**	**Type**	**Bytes**
LRL	30	175	5	BPM	Uint8	1
URL	50	175	5	BPM	Uint8	1
Max Sensor Rate	50	175	5	BPM	Uint8	1
Fixed AV Delay	70	300	10	ms	Uint8	1
Dynamic AV Delay	OFF	ON	N/A	N/A	Bool	1
Sensed AV Delay Offset	0 (lowest -10)	-100	-10	ms	Uint8	1
Atrial Amplitude	0.00	5.00	1.25	Volts	Uint8	1
Ventricular Amplitude	0.00	5.00	1.25	Volts	Uint8	1
Atrial Pulse Width	0.05 0.1	- 1.9	- 0.1	ms	Uint8	1
Ventricular Pulse Width	0.05 0.1	- 1.9	- 0.1	ms	Uint8	1
Atrial Sensitivity	0.25 1	0.75 10	0.25 0.5	mV	Uint8	1
Ventricular Sensitivity	0.25 1	0.75 10	0.25 0.5	mV	Uint8	1
VRP	150	500	10	ms	Uint8	1
ARP	150	500	10	ms	Uint8	1
PVARP	150	500	10	ms	Uint8	1
PVARP Extension	0	400	50	ms	Uint8	1
Hysteresis	OFF	ON	N/A	N/A	Bool	1
Rate Smoothing	0	21	3	N/A	Uint8	1
ATR Duration	10 20 100	- 80 2000	- 20 100	cc	Uint8	1
ATR Fallback Mode	OFF	ON	N/A	N/A	Bool	1
ATR Fallback Time	1	5	1	min	Uint8	1
Activity Threshold	V-Low, Low, Med-Low, Med, Med-High, High, V-High			N/A	Uint8	1
Reaction Time	10	50	10	sec	Uint8	1
Response Factor	1	16	1	N/A	Uint8	1
Recovery Time	2	16	1	min	Uint8	1
Mode	OFF, DDD, VDD, DDI, DOO, AOO, AAI, VOO, VVI, AAT, VVT, DDDR, VDDR, DDIR, DOOR, AOOR, AAIR, VOOR, VVIR			N/A	Uint8	1

3.3.3.2 Rate Response Curve

3.3.4 DCM Design

The design of the DCM has 3 seperate sections; the **model, view and controller.**

3.3.4.1 Model

The model section handles the applications data and user logic. It does not interact directly with the interface but only manages data and enforces user authentication. Below is an explanation of the individual Python files and their functionality:

User Model

Purpose: To manage all user account information.

Functions:

- Loads and saves login information to or from the users.json file.
- Registers new users, checking for duplicates and the max user limit.
- Authenticates logins by verifying user passwords
- Provides a method to get the amount of registered users which is displayed by the DCM.

Pacing Model

Purpose: Manages pacing parameters for all users.

Functions:

- Loads and saves settings to or from pacing_settings.json
- The data is structured as a dictionary with settings as the key and value as the value.
- Saves a dictionary of parameters for a specific user and pacemaker mode
- Retrieves the saved parameters for a user and mode

Egram Model

Purpose: Manages the capture of electrogram data

Functions:

- Adds a single data point to an internal listed
- Returns the list of captured data
- Clears the captured data to start a new session

3.3.4.2 View

The view section is what the user sees and interacts with. It is built with the customtkinter library for a more aesthetically pleasing interface. They do not have any control logic, only to display data and send user actions such as button clicks to the controller. Below are an explanation of each view:

Login View

Purpose: Provides a screen for users to login

Functions:

- Welcome class displays the login form and a register button.
- Register class displays the new user registration form and a back button.
- When a button is clicked, it collects text from the entry fields and calls a method on the controller that handles the login logic.
- The welcome view calls a function to display the current registered user count.

Main View

Purpose: Provides the main application screen after logging in

Functions:

- MainFrame is the main menu. It shows buttons for each pacing mode AOO, VOO, AAI and VVI as well as mock connection controls. It also displays the current connection status.
- DataEntry is the form for editing parameters. It displays all parameters and handles validation of data such as the input being in the correct range and is a numeric type before sending the data to the controller.

3.3.4.3 Controller

Controller

Purpose: The main application class that uses instances of the models and view frames.

Functions:

- It initializes itself and creates all view frames, UserModel and PacingModel. It then stores these in a dictionary.
- The navigation is handled by a show_frame function which requests a frame from the dictionary and uses tkinters library function, tkraise(), to bring it into view.
- Event handling is seperated into handling registration and saving settings. For For registration it takes the username and password from the view, calls the UserModel to verify them, then shows a message to confirm or deny entry. The saving of settings takes mode and data from the DataEntry view and passes them to the PacingModel to be saved.

Main

Purpose: Its purpose and sole function is to create an instance of the controller and run it in the main loop, starting the customtkinter application.

3.3.4.4 Views of DCM GUI

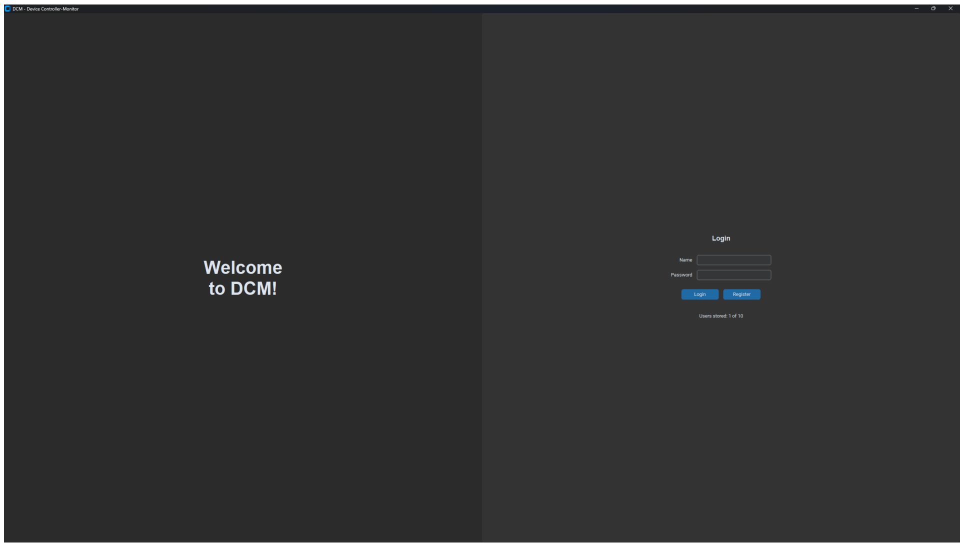


Figure 10: Login Screen of DCM

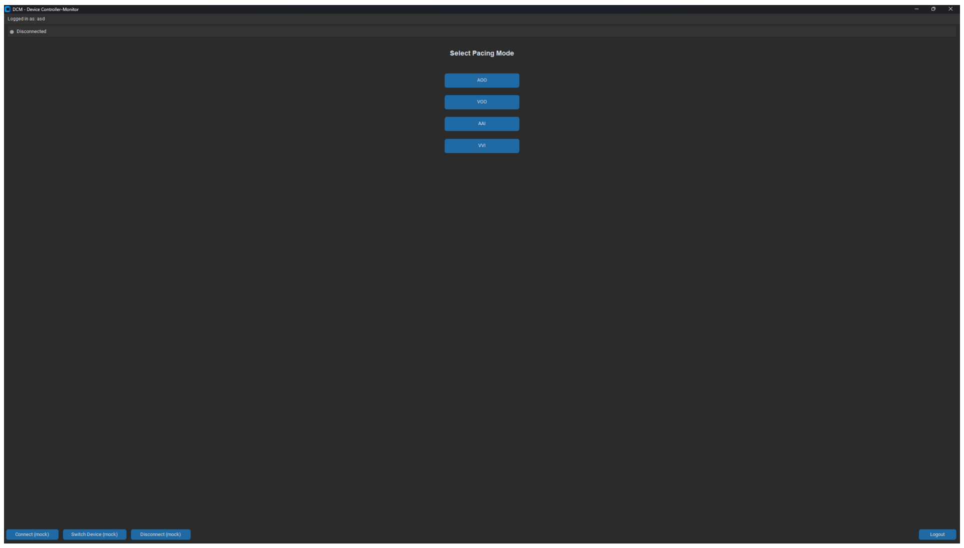
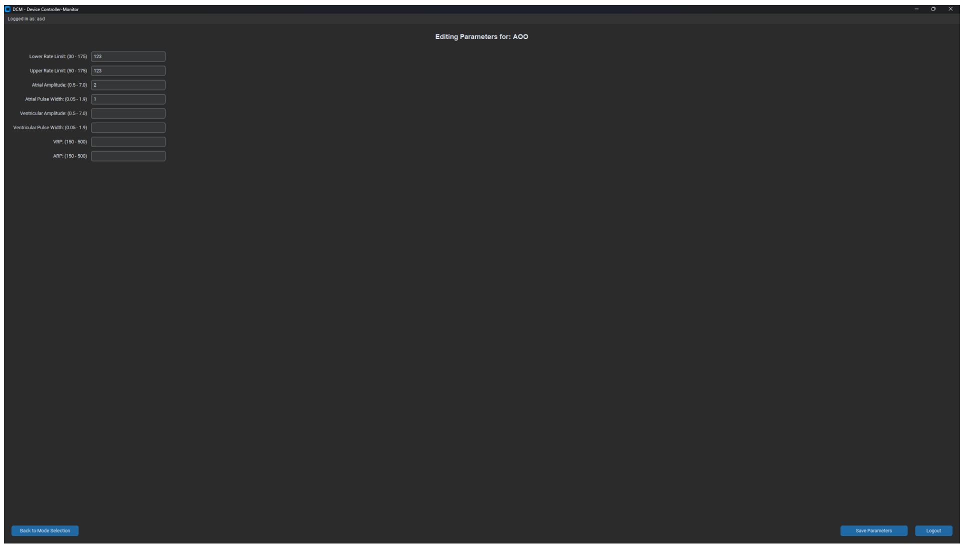


Figure 11: Main Menu of DCM

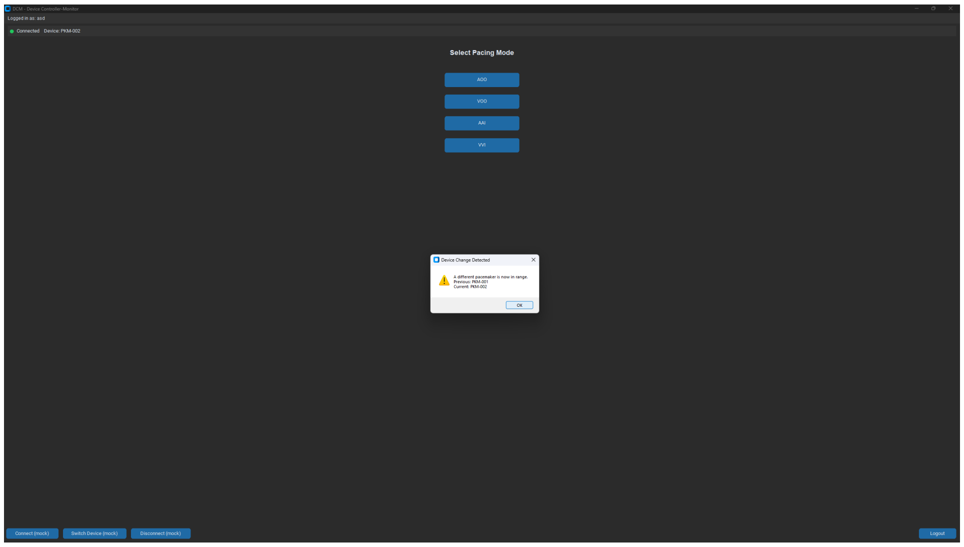

DCM - Device Controller-Monitor

Logged in as: asd

Editing Parameters for: AOO

Lower Rate Limit: (30 - 175)	123
Upper Rate Limit: (50 - 175)	123
Atrial Amplitude: (0.5 - 7.0)	2
Atrial Pulse Width: (0.05 - 1.9)	1
Ventricular Amplitude: (0.5 - 7.0)	
Ventricular Pulse Width: (0.05 - 1.9)	
VRP: (150 - 500)	
ARP: (150 - 500)	

Figure 12: Data Entry View


DCM - Device Controller-Monitor

Logged in as: asd

Connected Device: PKM-002

Select Pacing Mode

Device Change Detected

A different pacemaker is now in range.
Previous: PKM-001
Current: PKM-002

Figure 13: Switch in Device Detection from DCM

3.3.5 Assurance Cases

4 Part 2

4.1 Requirements Potential Changes

Pacing Modes: Additional pacing modes are to be developed such as AOOR, VOOR, AAIR, and VVIR.

Hardware Communication: The connection between hardware and GUI may be developed in the next deliverable. The functionality implemented in this deliverable will be used to interface with the board.

Parameter Changes: As additional pacing modes are to be developed, this in turn will bring additional parameters that will need to be added and configured.

Electrogram Functionality: The electrogram functionality will be used to store and display egram data provided by the simulating board.

4.2 Design Decision Potential Changes

Expansion of Libraries: As hardware communication will be implemented in the next iteration, hardware communication libraries such as those implementing serial communication will need to be explored.

Expansion of GUI: To accomodate for more pacing modes, the GUI will need to be modified. Furthermore, for improved quality of life features, more graphics and interactive features may be added.

DCM Architecture: The DCM system may need to be refactored to implement hardware communication.

Simulink Architecture: As 4 more additional modes are being added for a total of 8 pacing modes, the Simulink structure may need to be modified to integrate all new features more easily.

Data Verification: As data will be actively transfered from the DCM to the pacemaker hardware, data integrity further becomes more crucial to maintain. A system to verify parameters are correctly sent and saved will need to be implemented.

4.3 Module Description

Although breifly highlighted in the design section of this documentation. This section will go into more detail about the purpose of each module, key functionality, variables, and how each module interacts with one another.

As shown in Figure 1, there are 3 primary modules operating the pacemaker; **input constant variables, stateflow logic for modes, and hardware hiding.**

4.3.1 Input Constant Variables Module

This module is highlighted in the design section. It goes through all the state variables that are changeable parameters of the pacemaker. These inputs include general inputs such as the pacing mode, low rate interval, and hysteresis settings, as well as parameters for atrial and ventricle pacing. These parameters are then used in the stateflow logic module to complete pacing to user specifications.

4.3.2 Stateflow Logic

This module has many submodules which are also covered in the design section. The overall state machine controlling mode selection and reseting of variables to prevent unwanted pacing behaviour is shown in Figure 4. This module converts input parameters into raw data that can be further converted to electrical signals. This module recieves data from the input constant variables module as well as from the monitored input variables. This module then feeds into hardware hiding.

4.3.3 Hardware Hiding

Hardware hiding is also briefly covered in the design section. The purpose of this module is to convert the signals from the stateflow logic module into outputted electrical signals through pins. As shown in Figure 9, pins are mapped to certain electrical signals such as atrial outputs, ventricle outputs, front end signals, and general pin configurations such as grounding pins. This is designed to ensure coding and modules are easier to debug with higher level identification and function of each GPIO.

4.4 Testing

4.4.1 SimuLink Mode Testing

4.4.1.1 Testing of AOO

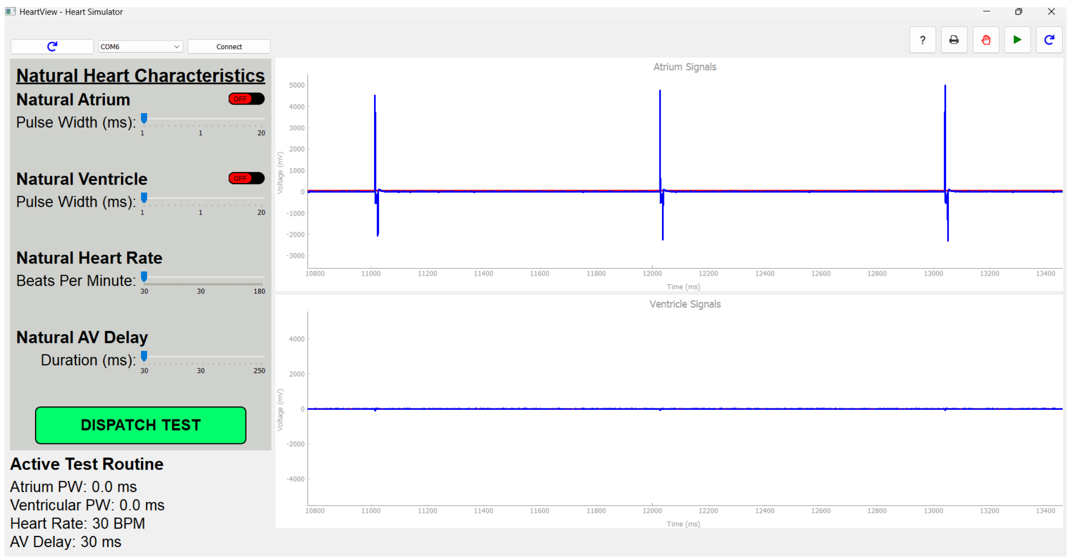
Purpose: The purpose of this test is to test basic AOO functionality.

Input Conditions: General inputs of mode = 0, AOO, and hysteresis = 0, off, and standard atrial inputs.

Expected Output: Consistent, evenly spaced pulses in the output with disregard for natural heart beats.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 10.

Result: Pass



Pulse Width (ms): 1

Pulse Width (ms): 1

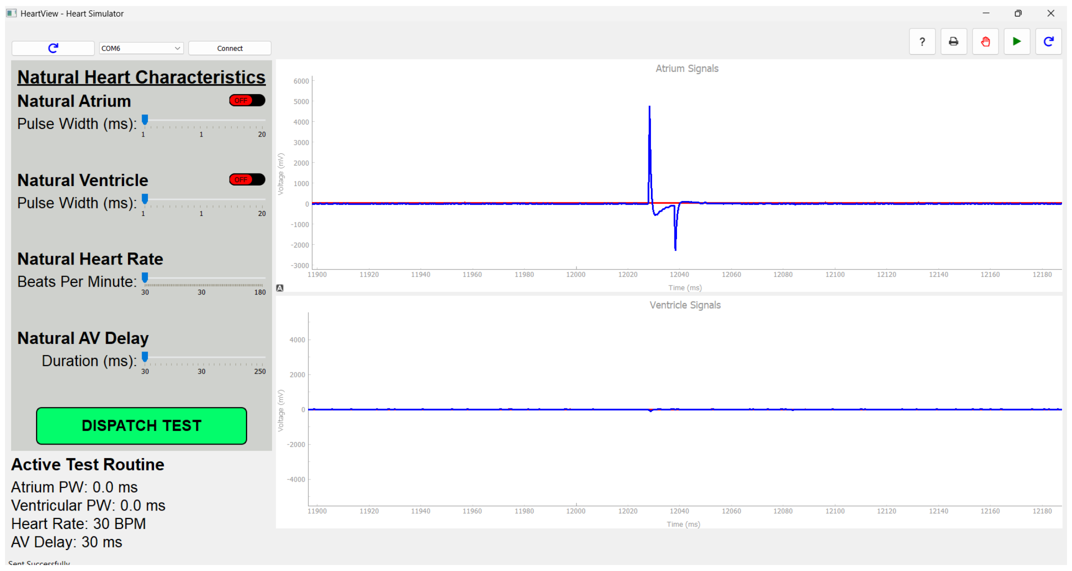
Beats Per Minute: 30

Duration (ms): 30

Active Test Routine
Atrium PW: 0.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 30 BPM
AV Delay: 30 ms

Figure 14: AOO Test

Figure 11 below shows a zoomed in view of one of the pulses in the AOO test above, Figure 10.



Active Test Routine
Atrium PW: 0.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 30 BPM
AV Delay: 30 ms

Figure 15: Close-up of AOO Pulse

4.4.1.2 Testing of VOO

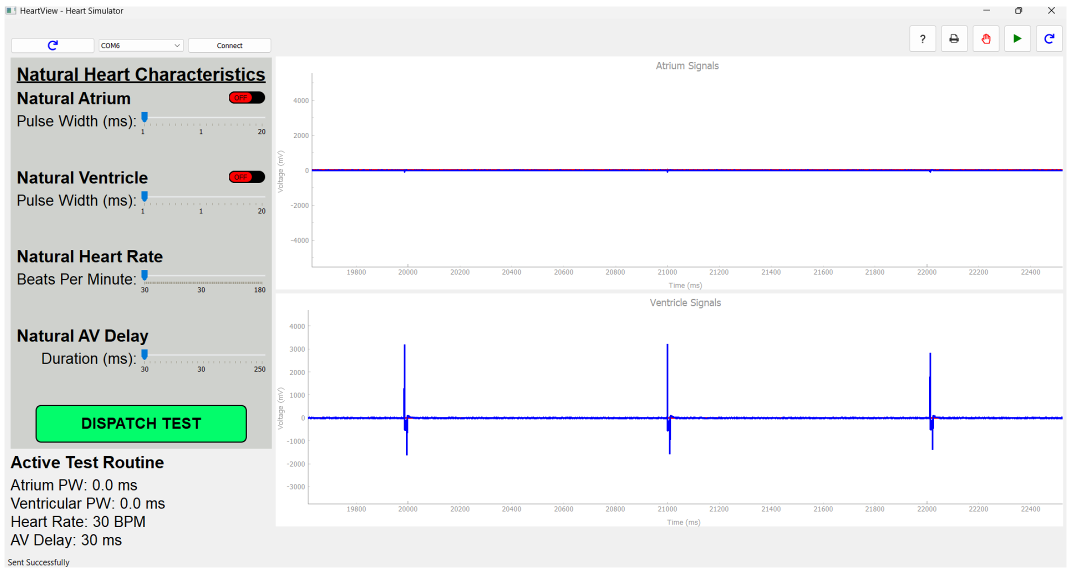
Purpose: The purpose of this test is to test basic VOO functionality.

Input Conditions: General inputs of mode = 1, VOO, and hysteresis = 0, off, and standard ventricle inputs.

Expected Output: Consistent, evenly spaced pulses in the output with disregard for natural heart beats.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 12.

Result: Pass

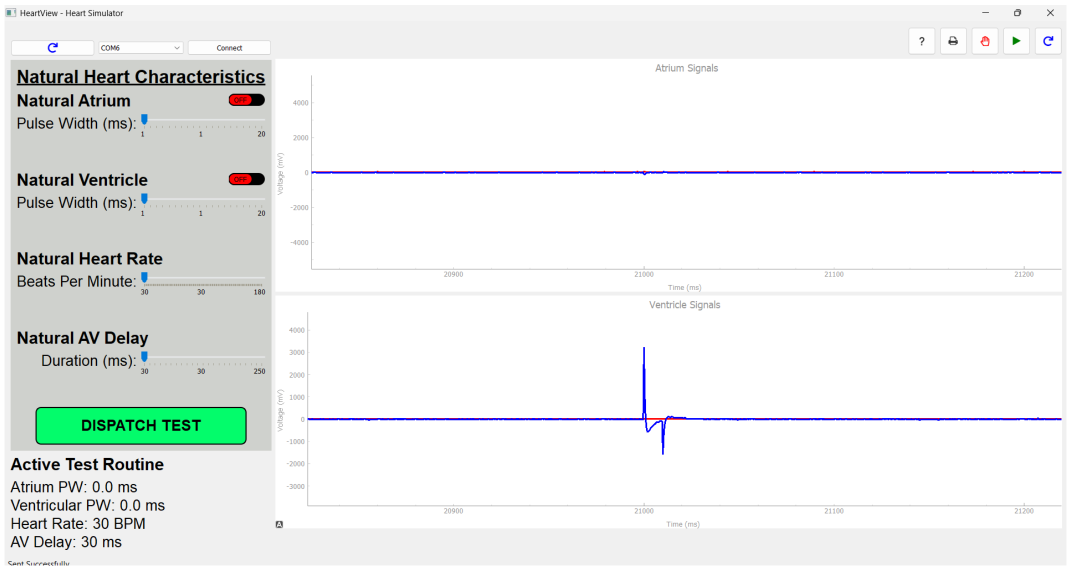


Natural Heart Characteristics
Natural Atrium
Pulse Width (ms): 1
Natural Ventricle
Pulse Width (ms): 1
Natural Heart Rate
Beats Per Minute: 30
Natural AV Delay
Duration (ms): 30

Active Test Routine
Atrium PW: 0.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 30 BPM
AV Delay: 30 ms
Sent Successfully

Figure 16: VOO Test

Figure 13 below shows a zoomed in view of one of the pulses in the VOO test above, Figure 12



Natural Heart Characteristics

Natural Atrium

Pulse Width (ms): 1

Natural Ventricle

Pulse Width (ms): 1

Natural Heart Rate

Beats Per Minute: 30

Natural AV Delay

Duration (ms): 30

Active Test Routine

Atrium PW: 0.0 ms

Ventricular PW: 0.0 ms

Heart Rate: 30 BPM

AV Delay: 30 ms

Sent Successfully

Figure 17: Close-up of VOO Pulse

4.4.1.3 Testing of AAI

No Natural Heart Rate

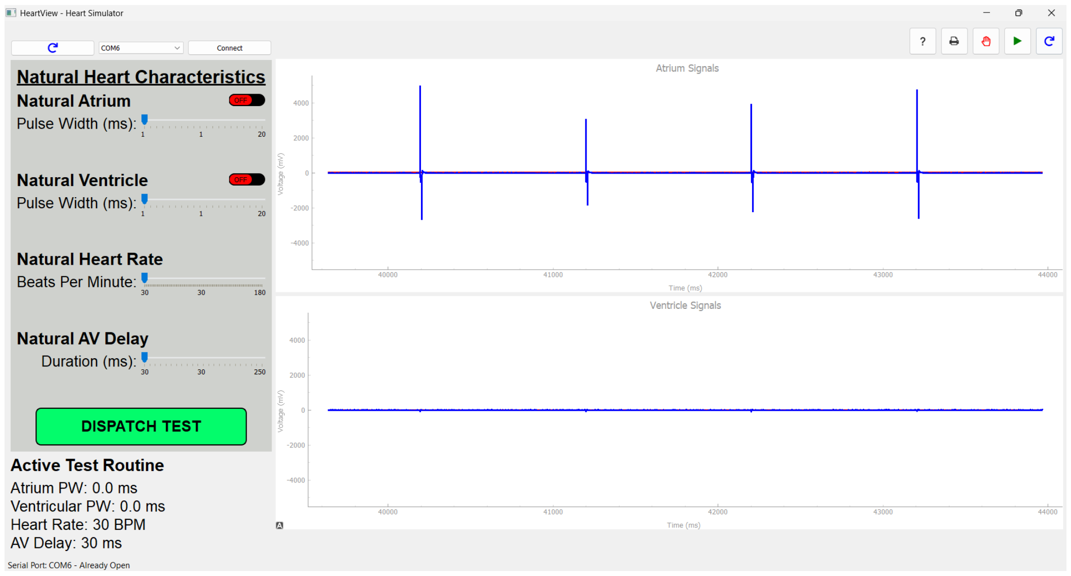
Purpose: The purpose of this test is to test basic AAI functionality with no natural heart rhythms.

Input Conditions: General inputs of mode = 2, AAI, and hysteresis = 0, off, and standard artial inputs.

Expected Output: Consistent, evenly spaced pulses in the output.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 13.

Result: Pass



Active Test Routine
Atrium PW: 0.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 30 BPM
AV Delay: 30 ms
Serial Port: COM6 - Already Open

Figure 18: AAI Test No Heart Rate

Natural Heart Rate of 45 BPM

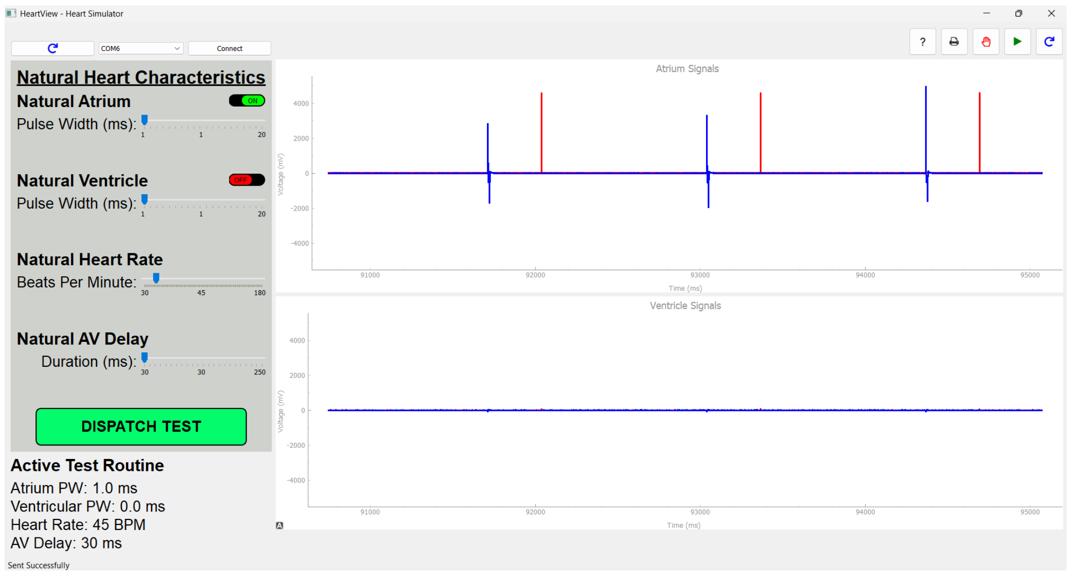
Purpose: The purpose of this test is to test basic AAI functionality at a low heart rate.

Input Conditions: General inputs of mode = 2, AAI, and hysteresis = 0, off, standard artial inputs, and monitored atrial pulses at 45 BPM.

Expected Output: The pacemaker should pulse after the refactory period expires. An output of blue pulses right before the natural red pulses is expected.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 15.

Result: Pass



Natural Heart Characteristics

Natural Atrium ON

Pulse Width (ms): 1 1 20

Natural Ventricle OFF

Pulse Width (ms): 1 1 20

Natural Heart Rate

Beats Per Minute: 30 45 180

Natural AV Delay

Duration (ms): 30 30 250

DISPATCH TEST

Active Test Routine

Atrium PW: 1.0 ms

Ventricular PW: 0.0 ms

Heart Rate: 45 BPM

AV Delay: 30 ms

Sent Successfully

Figure 19: AAI Test 45 BPM

Natural Heart Rate of 75 BPM

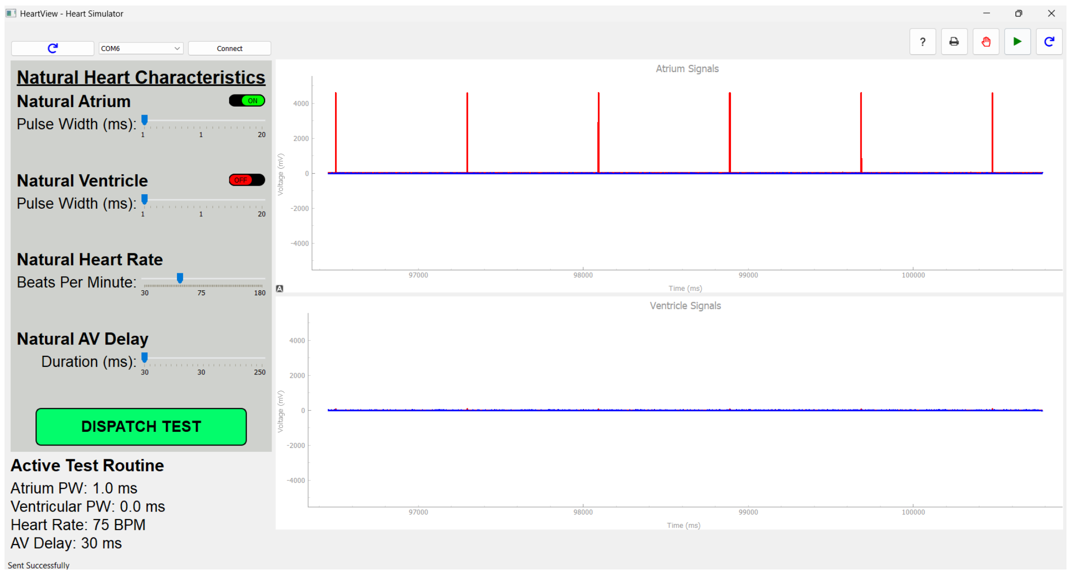
Purpose: The purpose of this test is to test basic AAI functionality at a nominal heart rate.

Input Conditions: General inputs of mode = 2, AAI, and hysteresis = 0, off, standard artial inputs, and monitored atrial pulses at 75 BPM.

Expected Output: As a nominal heart reate is being inputted, the pacemaker should not be delivering pacing pulses as the simulated heart rate is nominal and healthy.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 16.

Result: Pass



Natural Heart Characteristics

Natural Atrium ON
Pulse Width (ms): 1

Natural Ventricle OFF
Pulse Width (ms): 1

Natural Heart Rate
Beats Per Minute: 75

Natural AV Delay
Duration (ms): 30

Active Test Routine
Atrium PW: 1.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 75 BPM
AV Delay: 30 ms

Sent Successfully

Figure 20: AAI Test 75 BPM

4.4.1.4 Testing of VVI

No Natural Heart Rate

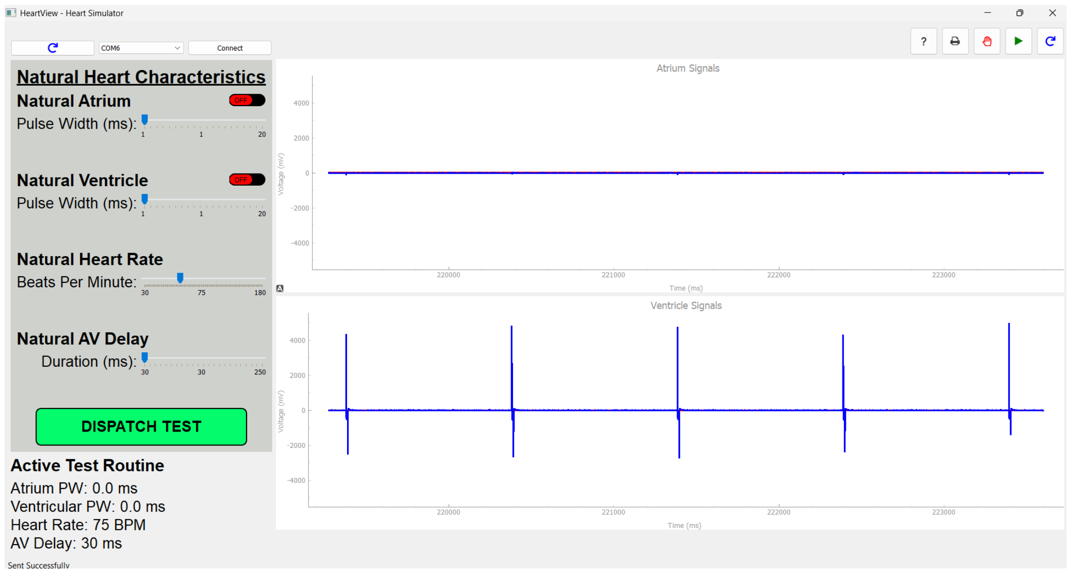
Purpose: The purpose of this test is to test basic VVI functionality with no inputted heart rate.

Input Conditions: General inputs of mode = 3, VVI, and hysteresis = 0, off, standard ventricle inputs, and monitored ventricle pulses.

Expected Output: Consistent, evenly spaced pulses in the output.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 17.

Result: Pass



Natural Heart Characteristics
Natural Atrium
Pulse Width (ms):
Natural Ventricle
Pulse Width (ms):
Natural Heart Rate
Beats Per Minute:
Natural AV Delay
Duration (ms):

Active Test Routine
Atrium PW: 0.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 75 BPM
AV Delay: 30 ms
Sent Successfully

Figure 21: VVI Test No Heart Rate

Natural Heart Rate at 45 BPM

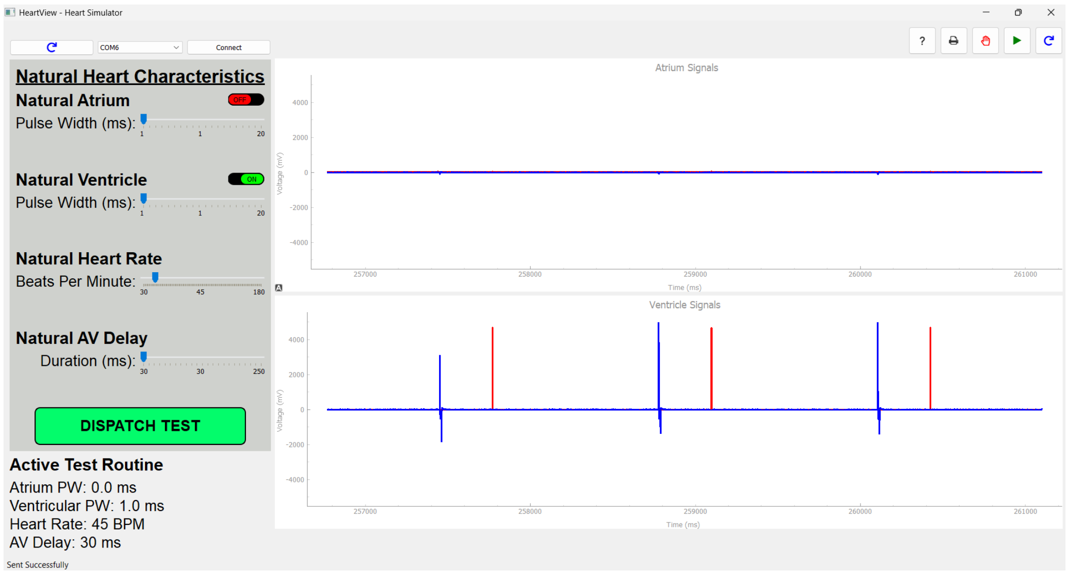
Purpose: The purpose of this test is to test basic VVI functionality with no inputted heart rate.

Input Conditions: General inputs of mode = 3, VVI, and hysteresis = 0, off, standard ventricle inputs, and monitored ventricle pulses.

Expected Output: The pacemaker should pulse after the refactory period expires. An output of blue pulses right before the natural red pulses is expected.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 18.

Result: Pass



Active Test Routine
Atrium PW: 0.0 ms
Ventricular PW: 1.0 ms
Heart Rate: 45 BPM
AV Delay: 30 ms

Sent Successfully

Figure 22: VVI Test 45 BPM

Natural Heart Rate at 75 BPM

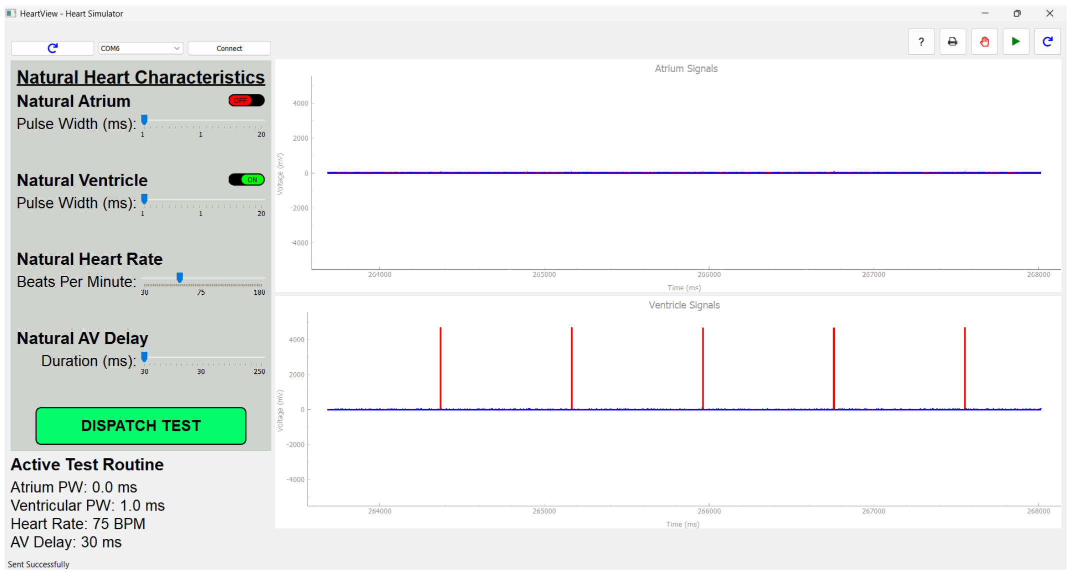
Purpose: The purpose of this test is to test basic VVI functionality with no inputted heart rate.

Input Conditions: General inputs of mode = 3, VVI, and hysteresis = 0, off, standard ventricle inputs, and monitored ventricle pulses.

Expected Output: As a nominal heart reate is being inputted, the pacemaker should not be delivering pacing pulses as the simulated heart rate is nominal and healthy.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 18.

Result: Pass



Active Test Routine
Atrium PW: 0.0 ms
Ventricular PW: 1.0 ms
Heart Rate: 75 BPM
AV Delay: 30 ms
Sent Successfully

Figure 23: VVI Test 75 BPM

4.4.1.5 Hysteresis Testing

Hysteresis Test 1 (60 BPM)

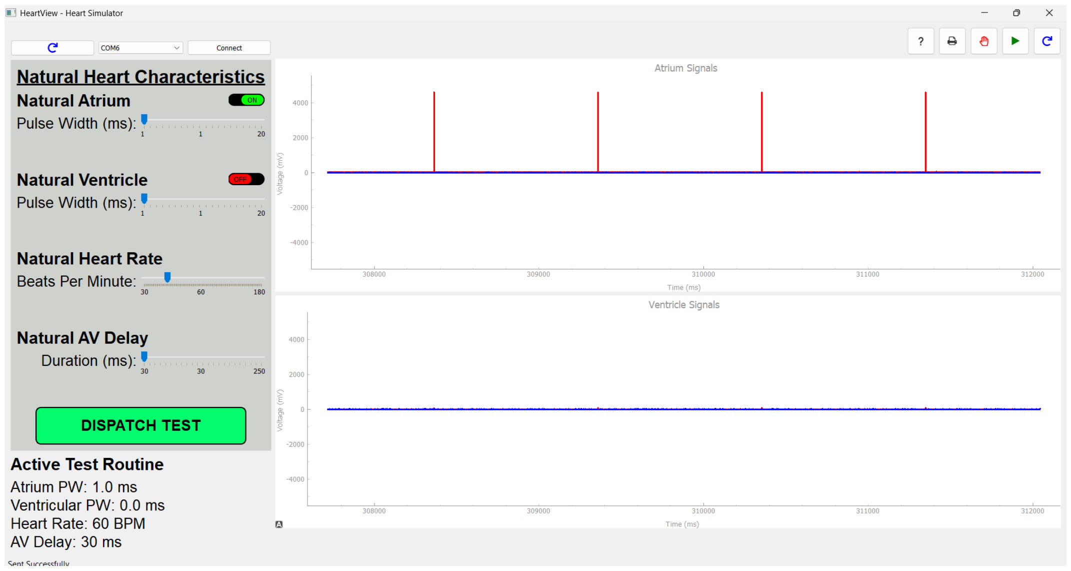
Purpose: The purpose of this test is to test basic hysteresis mode functionality.

Input Conditions: General inputs of mode = 2, AAI, and hysteresis = 1, on, standard atrium inputs, and monitored atrial pulses at 50 BPM.

Expected Output: No output of the pacemaker.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 20.

Result: Pass



Natural Heart Characteristics

Natural Atrium ON
Pulse Width (ms): 1 (1–20)

Natural Ventricle OFF
Pulse Width (ms): 1 (1–20)

Natural Heart Rate
Beats Per Minute: 60 (30–180)

Natural AV Delay
Duration (ms): 30 (30–250)

Active Test Routine
Atrium PW: 1.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 60 BPM
AV Delay: 30 ms

Sent Successfully

Figure 24: Hysterisis Test 1

Hysteresis Test 2 (50 BPM)

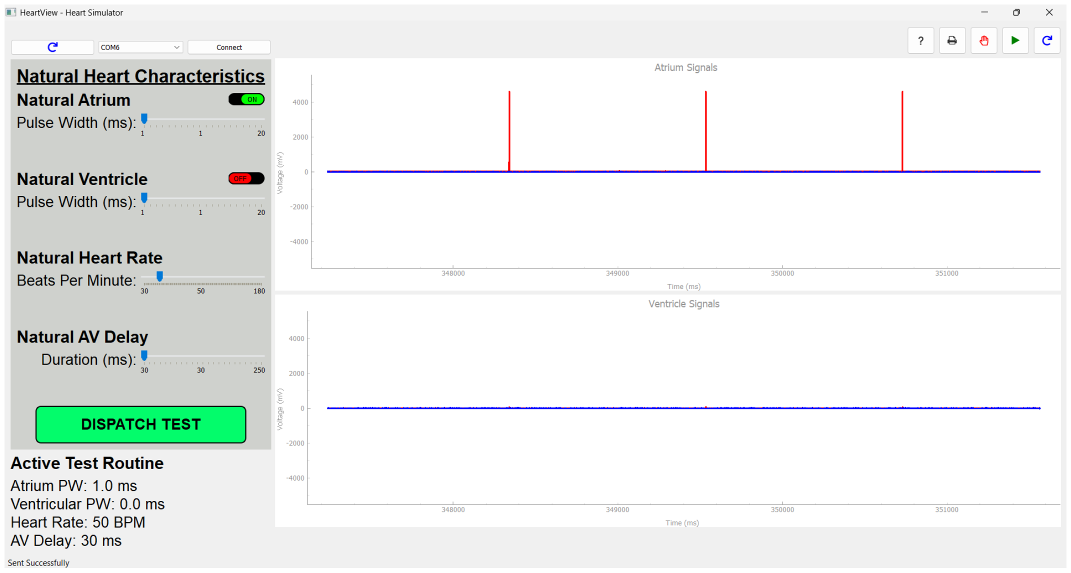
Purpose: The purpose of this test is to test basic hysteresis mode functionality.

Input Conditions: General inputs of mode = 2, AAI, and hysteresis = 1, on, standard atrium inputs, and monitored atrial pulses at 50 BPM.

Expected Output: No output of the pacemaker.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 21.

Result: Pass



Active Test Routine
Atrium PW: 1.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 50 BPM
AV Delay: 30 ms
Sent Successfully

Figure 25: Hysterisis Test 2

Hysteresis Test 3 (40 BPM)

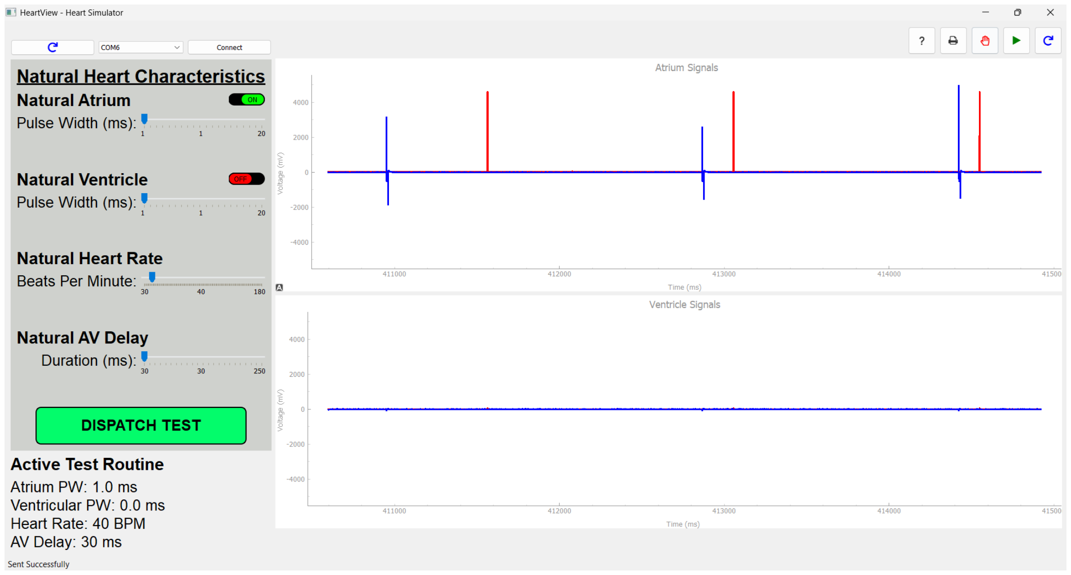
Purpose: The purpose of this test is to test basic hysteresis mode functionality.

Input Conditions: General inputs of mode = 2, AAI, and hysteresis = 1, on, standard atrium inputs, and monitored atrial pulses at 50 BPM.

Expected Output: Delayed output signals from the pacemaker.

Actual Output: Output of testing is exactly that of expected, as shown below in Figure 22.

Result: Pass



Active Test Routine
Atrium PW: 1.0 ms
Ventricular PW: 0.0 ms
Heart Rate: 40 BPM
AV Delay: 30 ms
Sent Successfully

Figure 26: Hysterisis Test 3

4.4.2 DCM Testing

4.4.2.1 Login and Registration

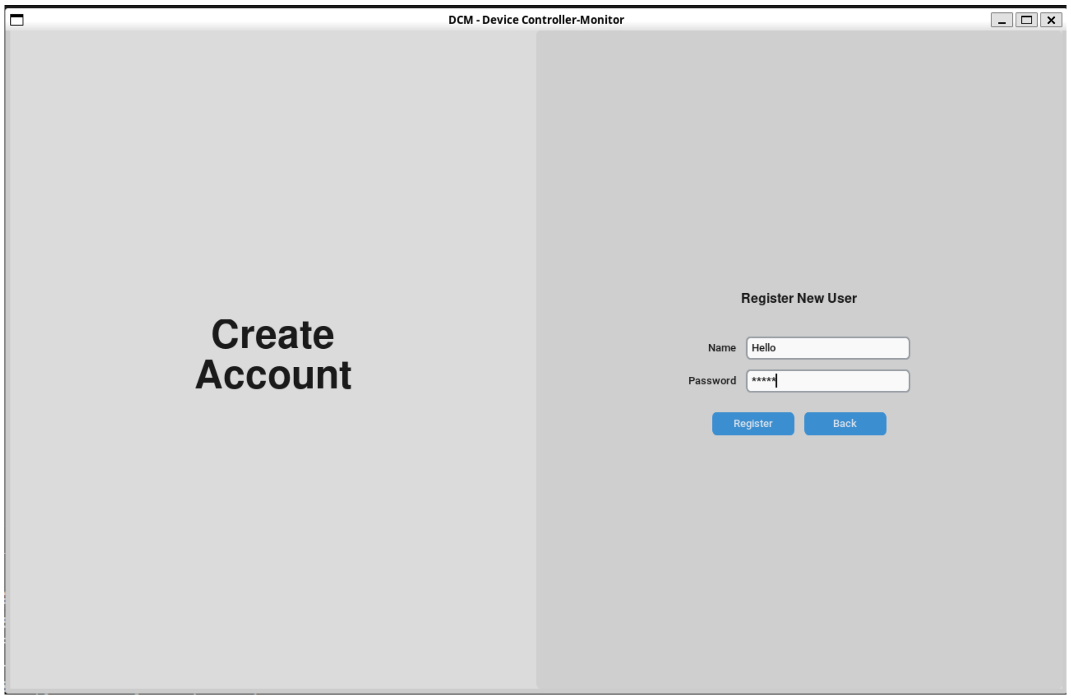
Purpose: The purpose of this test is to verify correct storage of newly registered user data and allow login.

Input Conditions: A random username and password. This will be used again in the login screen to access the DCM.

Expected Output: A window should pop up notifying the user an account has been registered. The DCM controls should be accessible after the user logs in.

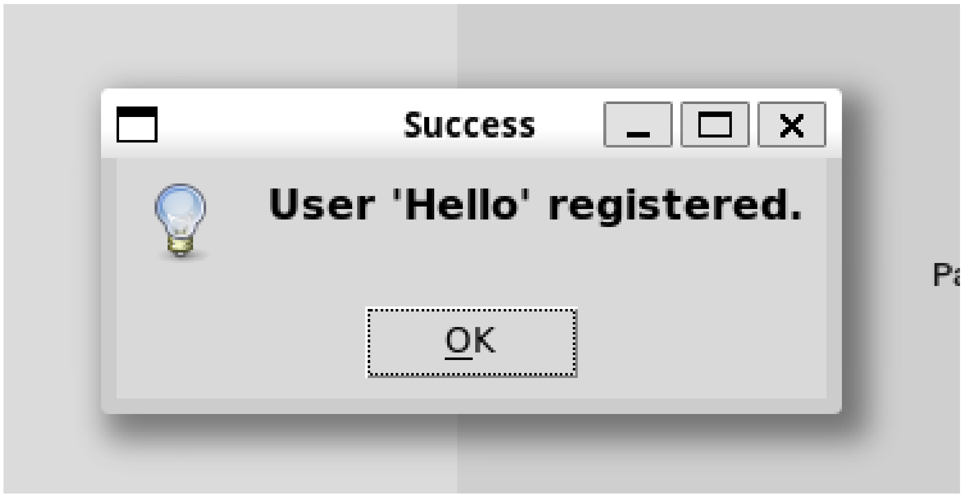
Actual Output: Dialogue is shown and the file is updated to include new user data. The user is then brought to the

Result: Pass



Name: Hello
Password: *****

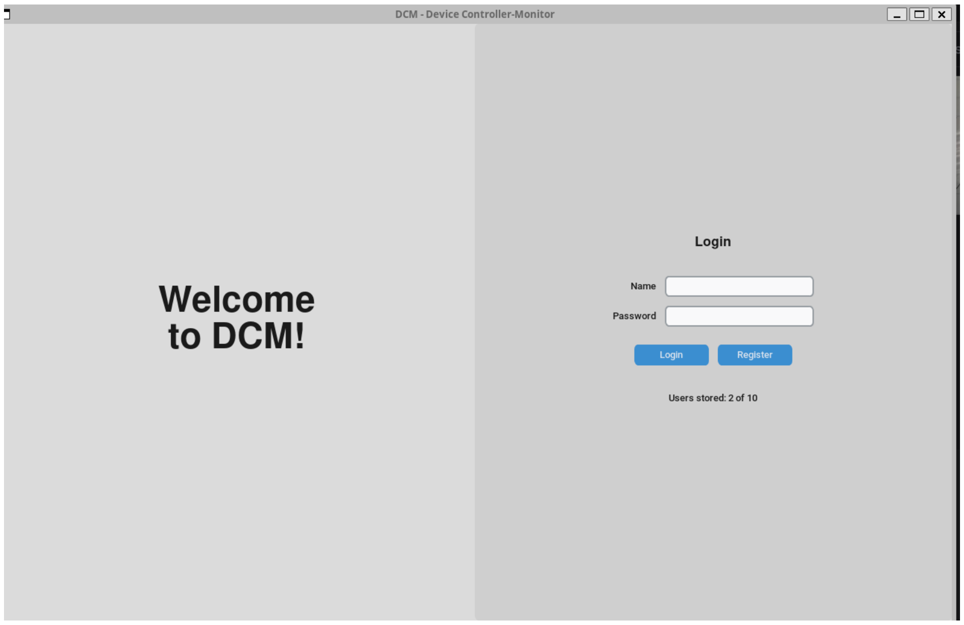
Figure 27: Registration Test


Success

User 'Hello' registered.

Pa

Figure 28: Registration Result


DCM - Device Controller-Monitor

Welcome to DCM!

Login

Name

Password

Users stored: 2 of 10

Figure 29: Login Test

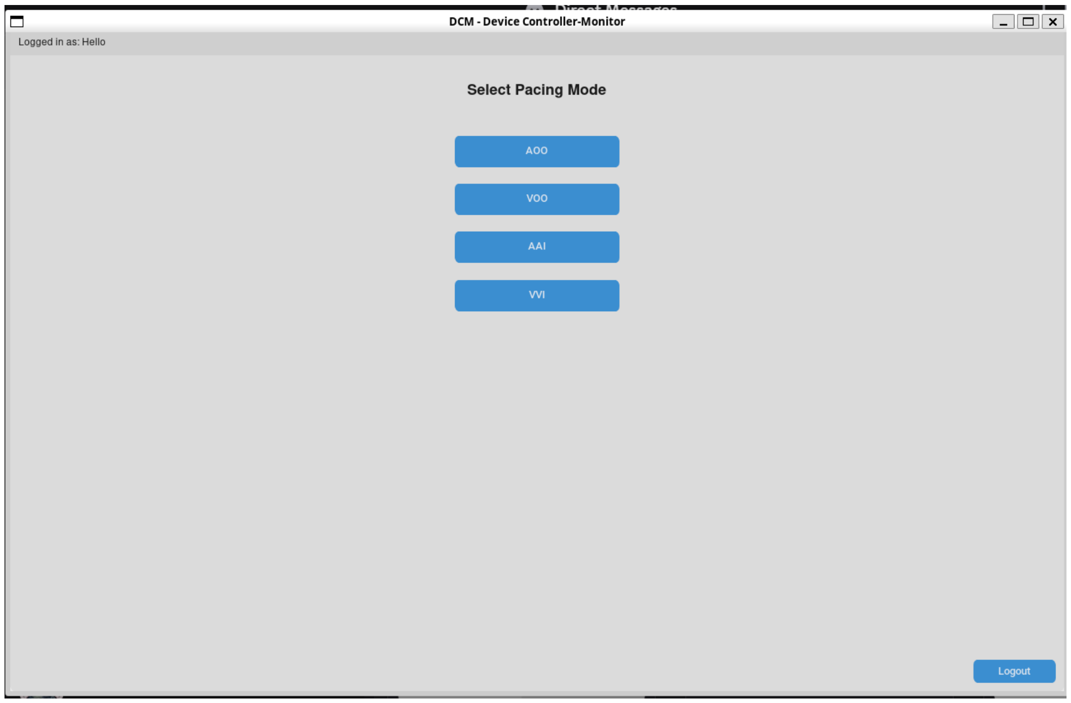


Figure 30: Login Result

4.4.2.2 Parameter Input Validation

Purpose: To enforce numeric types within an allowed range and to ensure the upper rate interval is greater than lower rate interval.

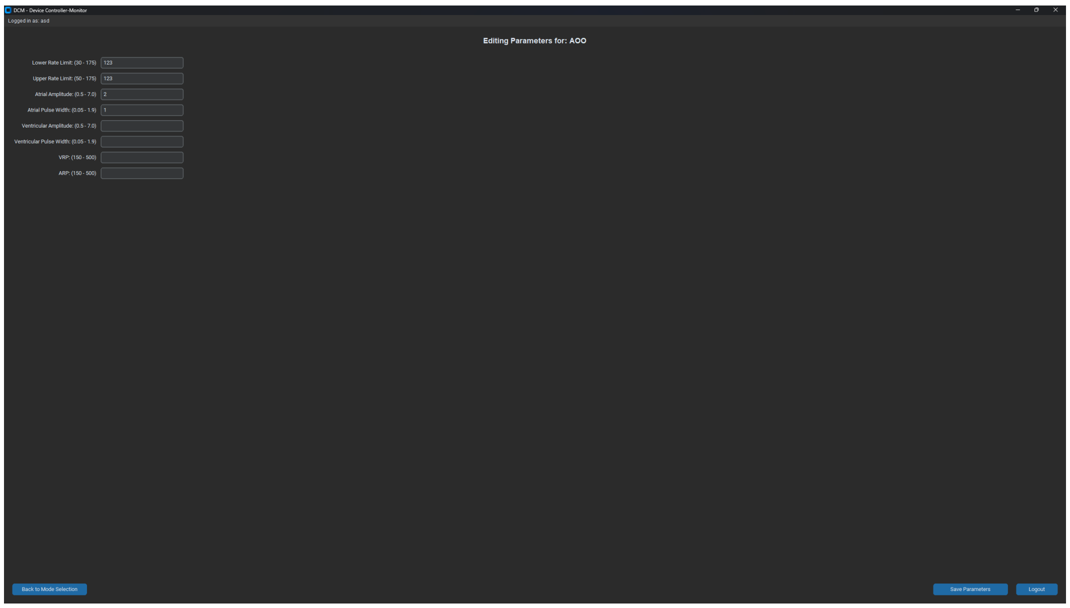
Input Conditions: Entering a non-numeric, an out of range number and an upper rate interval that is greater than lower rate interval in parameter settings.

Expected Output: Invalid input dialogue is shown and parameter changes are not saved.

Actual Output:

Result: Pass

The below figures show the general parameter page with some values inputted as well as the results of putting invalid values into parameter page.



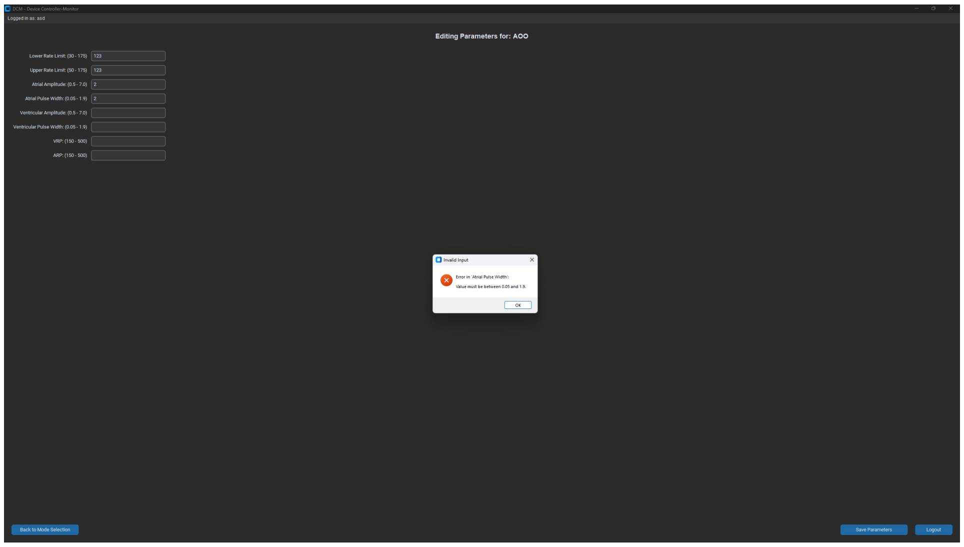
DCM - Device Controller-Monitor

Logged in as: asd

Editing Parameters for: AOO

Lower Rate Limit: (30 - 175)	123
Upper Rate Limit: (50 - 175)	123
Atrial Amplitude: (0.5 - 7.0)	2
Atrial Pulse Width: (0.05 - 1.9)	1
Ventricular Amplitude: (0.5 - 7.0)	
Ventricular Pulse Width: (0.05 - 1.9)	
VRP: (150 - 500)	
ARP: (150 - 500)	

Figure 31: DCM Parameter Page with Values Filled


DCM - Device Controller-Monitor

Logged in as: asd

Editing Parameters for: AOO

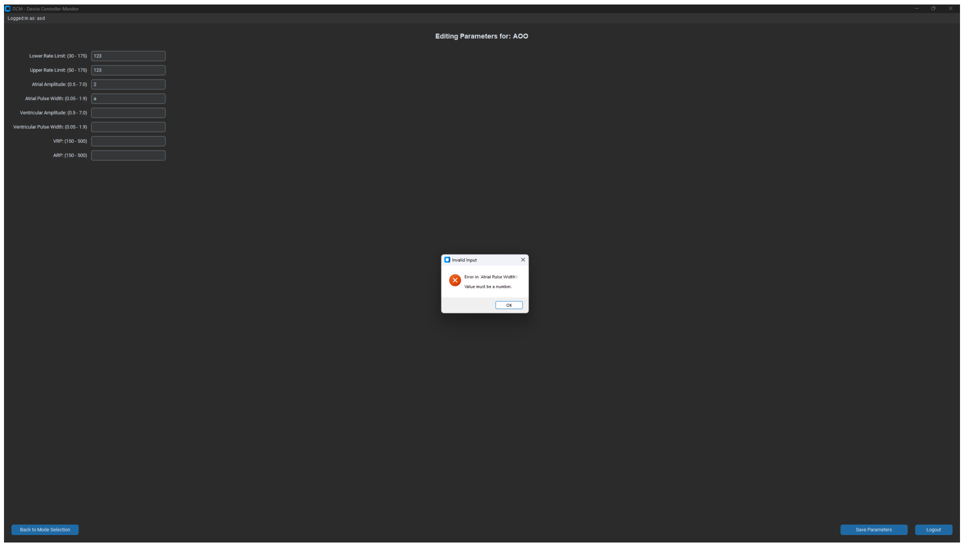
Lower Rate Limit: (30 - 175)	123
Upper Rate Limit: (50 - 175)	123
Atrial Amplitude: (0.5 - 7.0)	2
Atrial Pulse Width: (0.05 - 1.9)	2
Ventricular Amplitude: (0.5 - 7.0)	
Ventricular Pulse Width: (0.05 - 1.9)	
VRP: (150 - 500)	
ARP: (150 - 500)	

Invalid Input

Error in 'Atrial Pulse Width':

Value must be between 0.05 and 1.9.

Figure 32: DCM Parameter Error When Number Inputted is Out of Range


DCM - Device Controller-Monitor

Logged in as: asd

Editing Parameters for: AOO

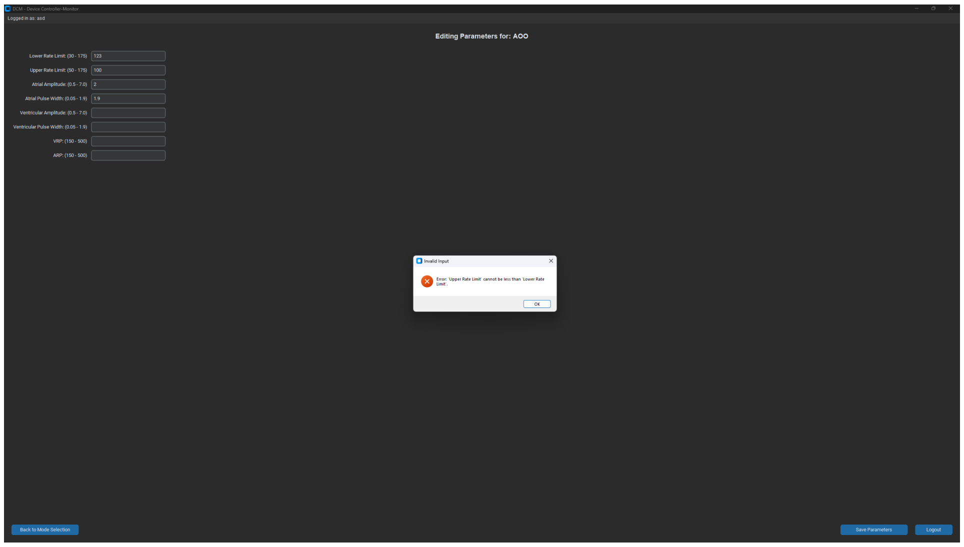
Lower Rate Limit: (30 - 175)	123
Upper Rate Limit: (50 - 175)	123
Atrial Amplitude: (0.5 - 7.0)	2
Atrial Pulse Width: (0.05 - 1.9)	a
Ventricular Amplitude: (0.5 - 7.0)	
Ventricular Pulse Width: (0.05 - 1.9)	
VRP: (150 - 500)	
ARP: (150 - 500)	

Invalid Input

Error in 'Atrial Pulse Width':

Value must be a number.

Figure 33: DCM Parameter Error When Non-Number is Inputted



DCM - Device Controller-Monitor

Logged in as: asd

Editing Parameters for: AOO

Lower Rate Limit: (30 - 175)	123
Upper Rate Limit: (50 - 175)	100
Atrial Amplitude: (0.5 - 7.0)	2
Atrial Pulse Width: (0.05 - 1.9)	1.9
Ventricular Amplitude: (0.5 - 7.0)	
Ventricular Pulse Width: (0.05 - 1.9)	
VRP: (150 - 500)	
ARP: (150 - 500)	

Invalid Input

Error: 'Upper Rate Limit' cannot be less than 'Lower Rate Limit'.

Figure 34: DCM Parameter Error When URL is Larger Than LRL

4.4.2.3 Mode Selection and Data Retrieval

Purpose: To test data storage, ensuring proper saving of user data

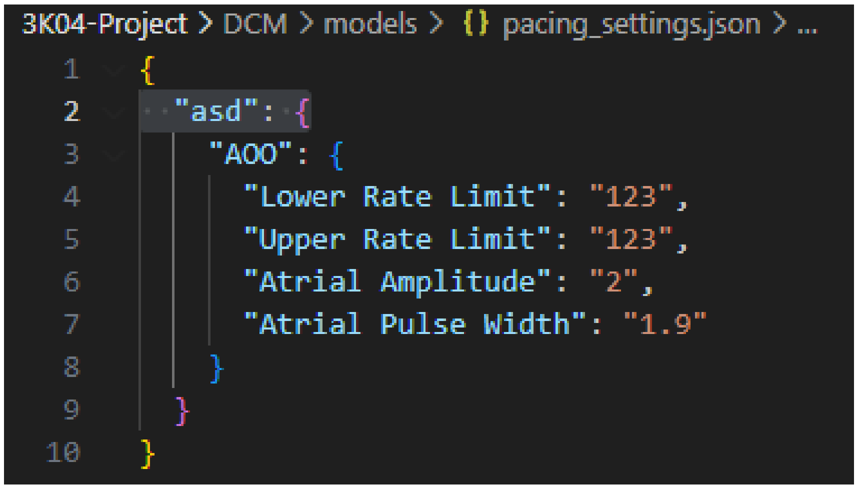
Input Conditions: Registering account, logging in, and saving parameters.

Expected Output: User data is now found in the associated JSON files.

Actual Output: Registered user data and parameters are found in their respective JSON files.

Result: Pass

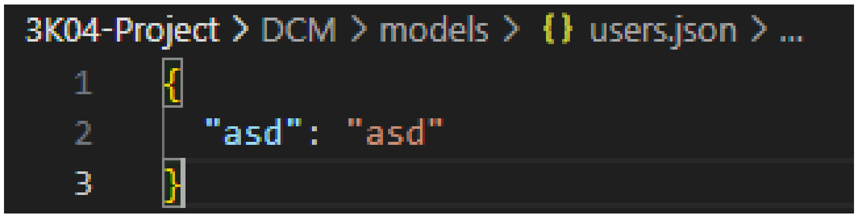
This test was done in conjunction to previous tests except with a different user registered. A user ”asd” with password ”asd” was used for faster log ins. The following images are of the JSON files and the saved parameters from the previous test.



3K04-Project > DCM > models > {} pacing_settings.json > ...

```json
{
  "asd": {
    "AOO": {
      "Lower Rate Limit": "123",
      "Upper Rate Limit": "123",
      "Atrial Amplitude": "2",
      "Atrial Pulse Width": "1.9"
    }
  }
}
```

Figure 35: Stored Parameter File



3K04-Project > DCM > models > {} users.json > ...

```json
{
  "asd": "asd"
}
```

Figure 36: Stored Users File

4.4.3 Serial Testing

4.4.4 Rate Adaptive Test

4.5 GenAI Usage

We used a Generative AI assistant to support development of the DCM. It provided starter boilerplate for a Tkinter app with a welcome screen, registration and login, JSON storage capped at ten users, which we then adapted and tested. We also used it to clarify Python functions and libraries such as Tkinter, JSON, etc. and to troubleshoot installing tkinter. We had AI to clarify comments within the code as well. All design decisions, requirements, and validation were done by our team, and we reviewed and verified all AI outputs before inclusion.