3K04 Deliverable 1: Documentation

Group 33

Last Updated: 2025-10-26

Contents

List of Figures

List of Tables

1 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 1: 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

- Overall system requirements (summarized from provided specification documents). It can be informal or semi-formal.
- Mode-specific requirements: AOO, VOO, AAI, VVI.

3.2.1 DCM Requirements

The user shall be capable of the following:

- Utilizing and managing windows for display of text and graphics.
- Processing user positioning and input buttons.
- Displaying all programmable parameters for review and modification.
- Visually indicating when telemetry is lost due to the device being out of range or noise.
- Vndicating when a different PACEMAKER device is approached than was previously interrogated

3.3 Design

In this section, you want to expand on the design decisions based on the requirements. You should be specific about your system design and how the various components relate together.

- System architecture (major subsystems, hardware hiding, pin mapping).
- Programmable parameters (rate limits, amplitudes, pulse widths, refractory periods, etc.).
- Hardware inputs and outputs (signals sensed, signals controlled).
- State machine design for each pacing mode (with diagrams if applicable). You can also use a tabular method.
- Simulink diagram
- Screenshots of your DCM, explaining its software structure

You should also be explicit on how your design decisions map directly to the requirements.

3.3.1 Simulink Design

3.3.1.1 Overall Design

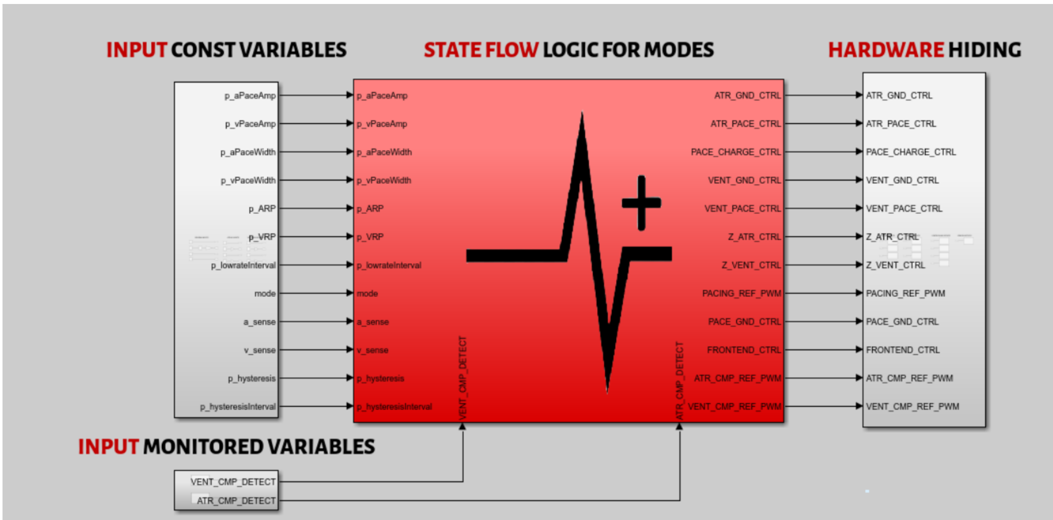

Figure 1: Overall Simulink Mapping

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

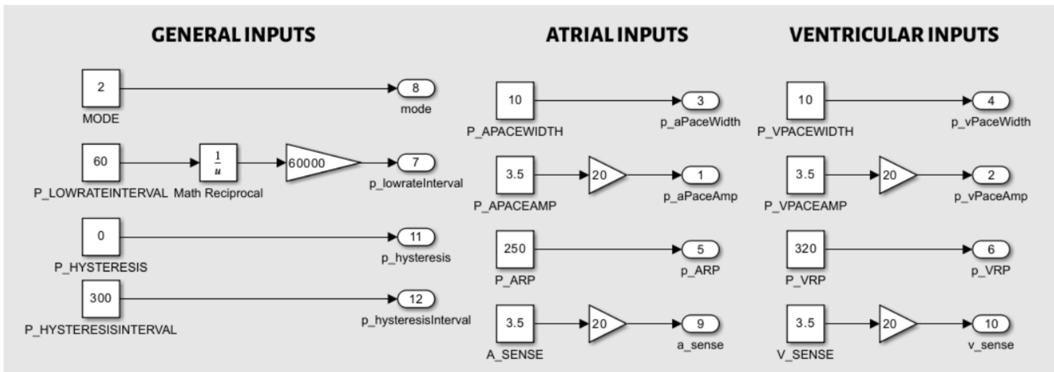

Figure 2: Constant Input Variables

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

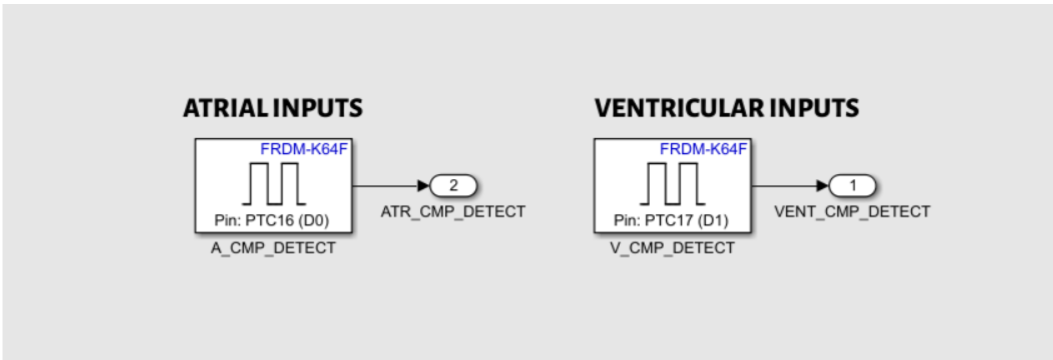

Figure 3: Monitored Input Variables

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

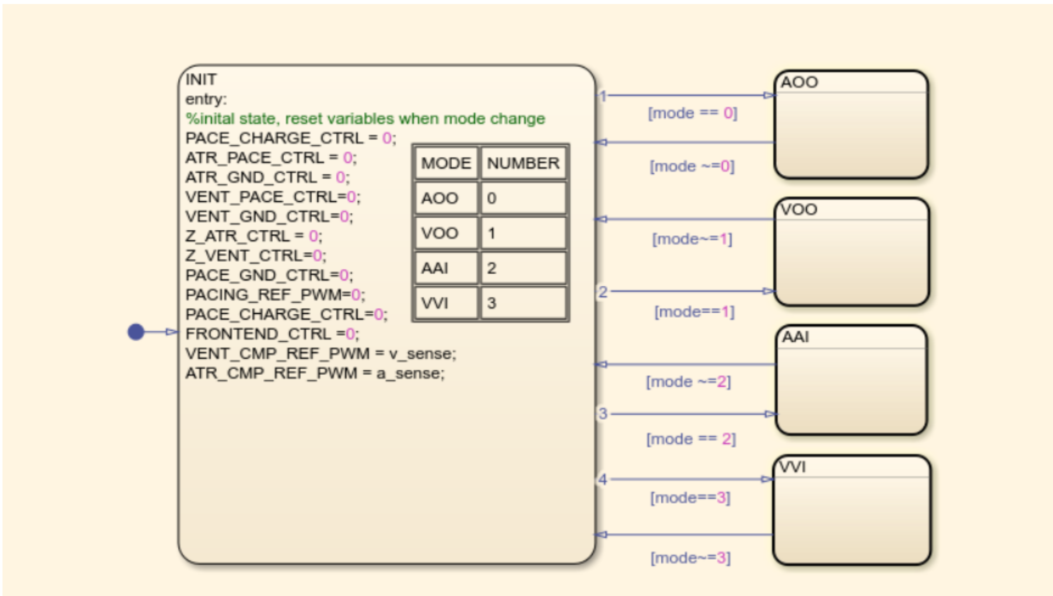

Figure 4: Stateflow Modules

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

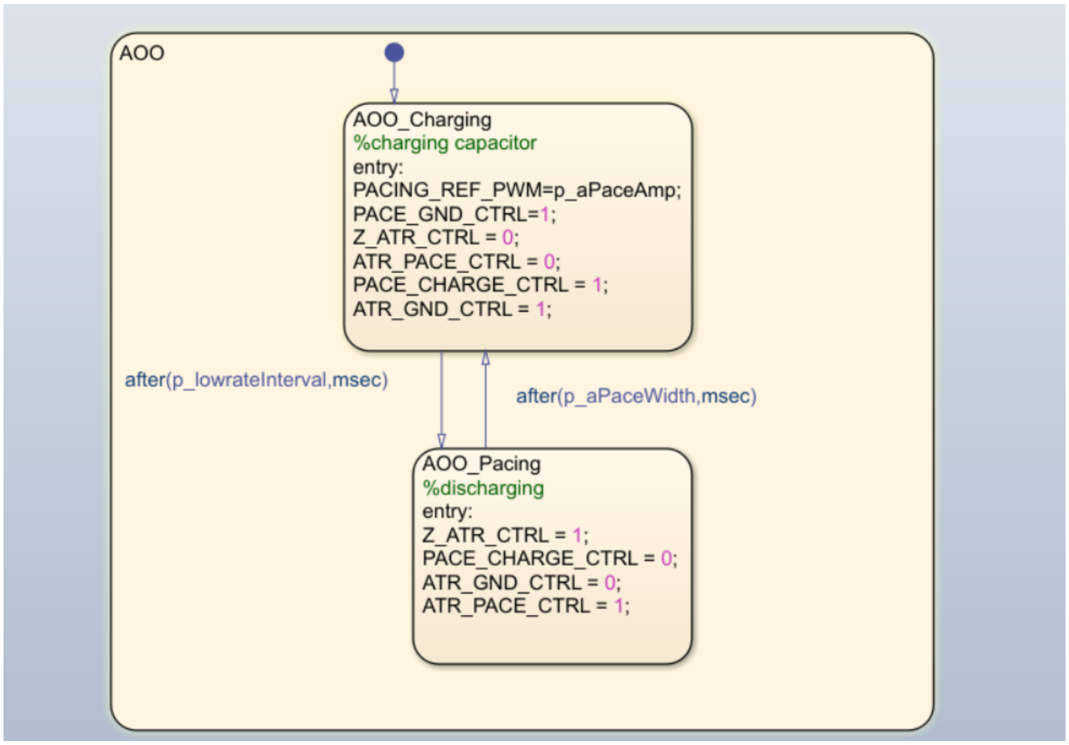

Figure 5: AOO Stateflow Model

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

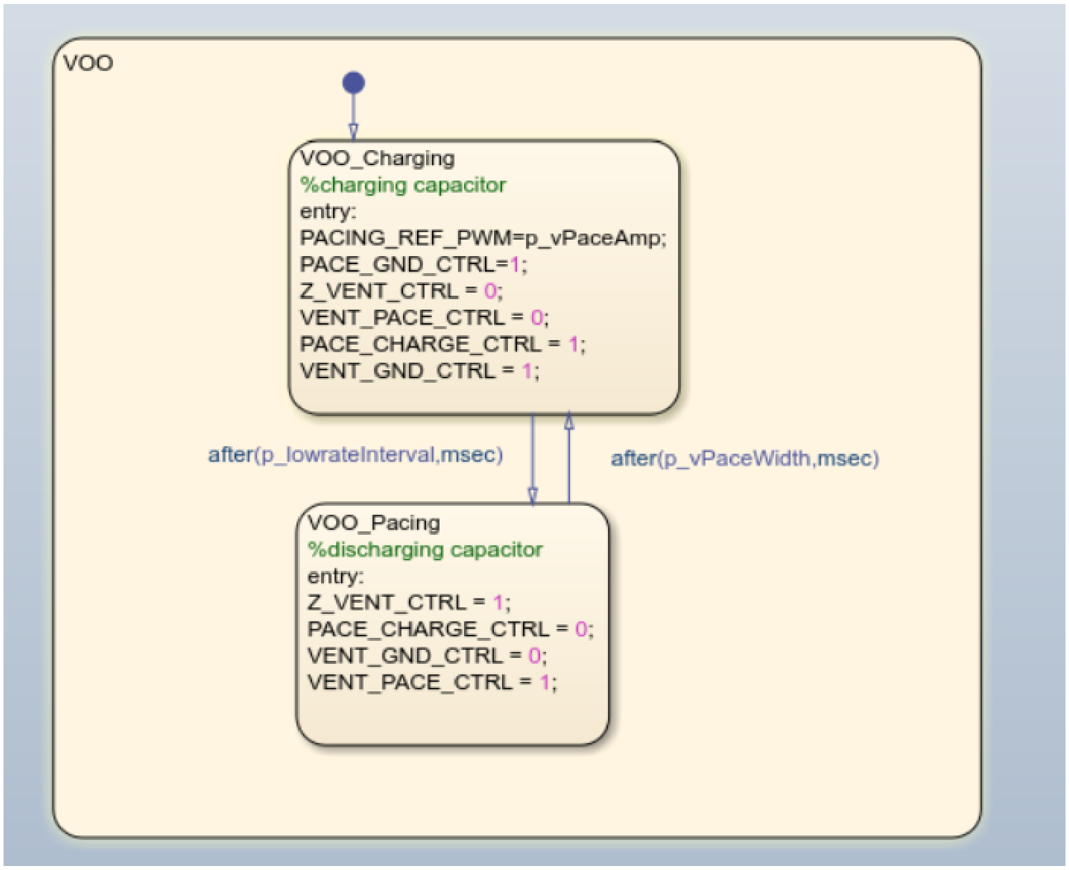

Figure 6: VOO Stateflow Model

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

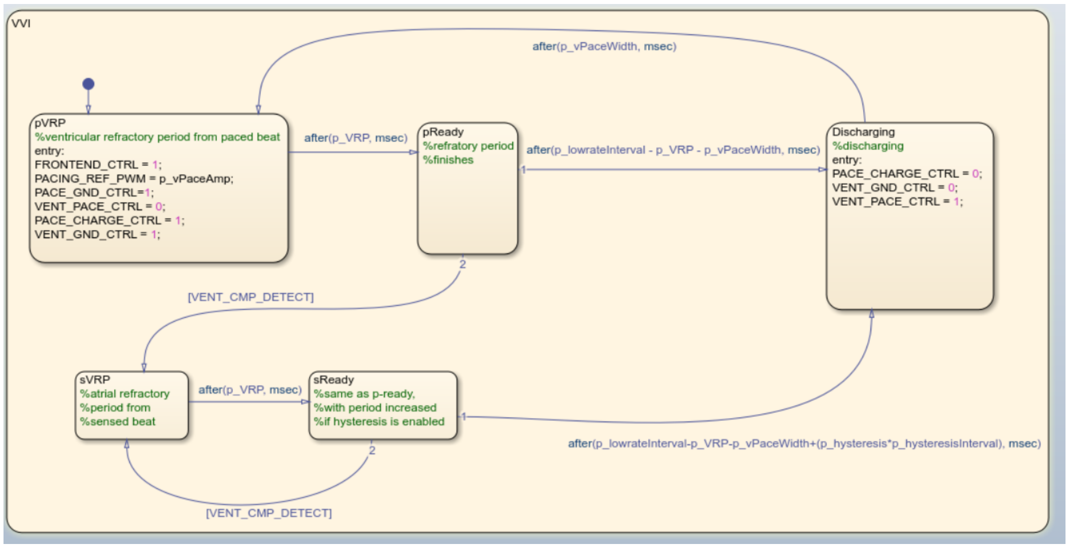

Figure 7: VVI Stateflow Model

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

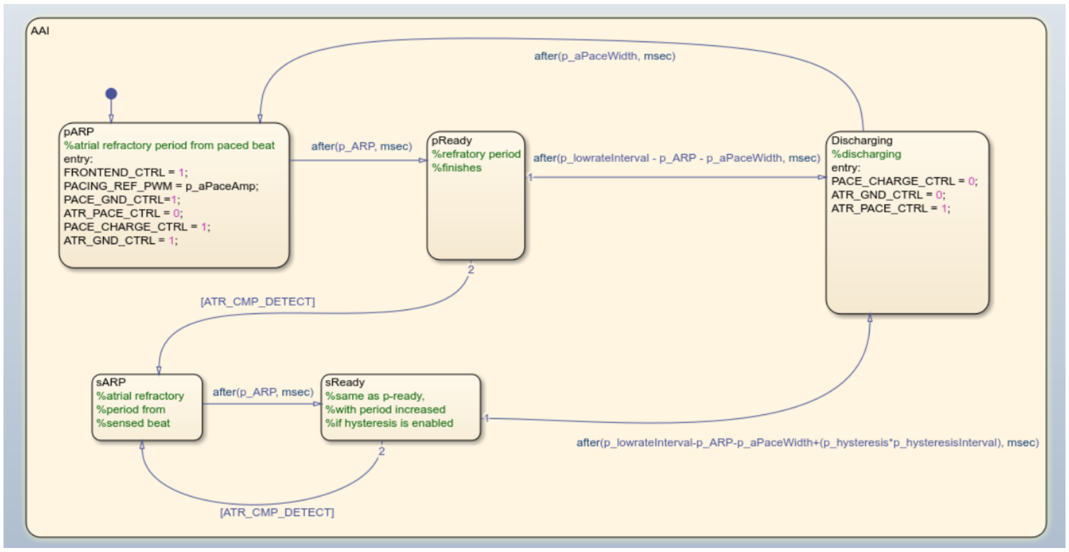

Figure 8: AAI Stateflow Model

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

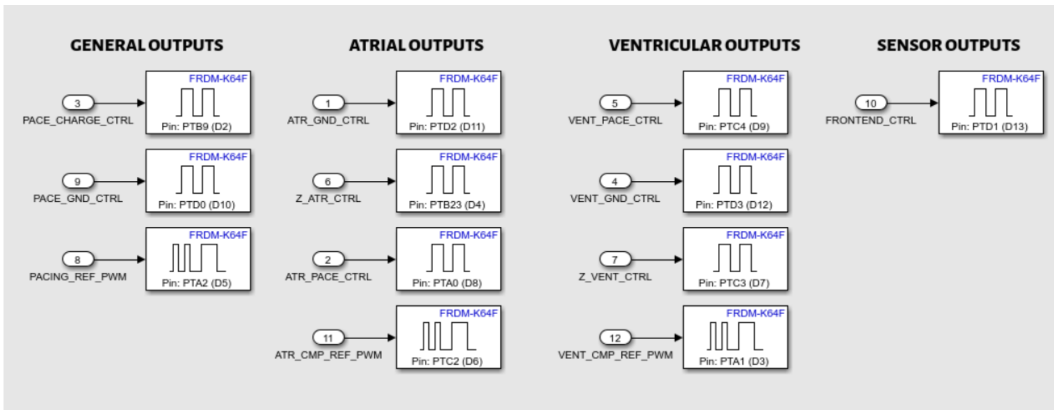

Figure 9: Hardware Hiding of Model

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 DCM Design

4 Part 2

4.1 Requirements Potential Changes

Identify which requirements may evolve in the next deliverable (e.g., adding more modes, communication, new parameters).

4.2 Design Decision Potential Changes

List design choices that may need revisiting (e.g., choice of libraries, interface design, architecture).

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, ventrical 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

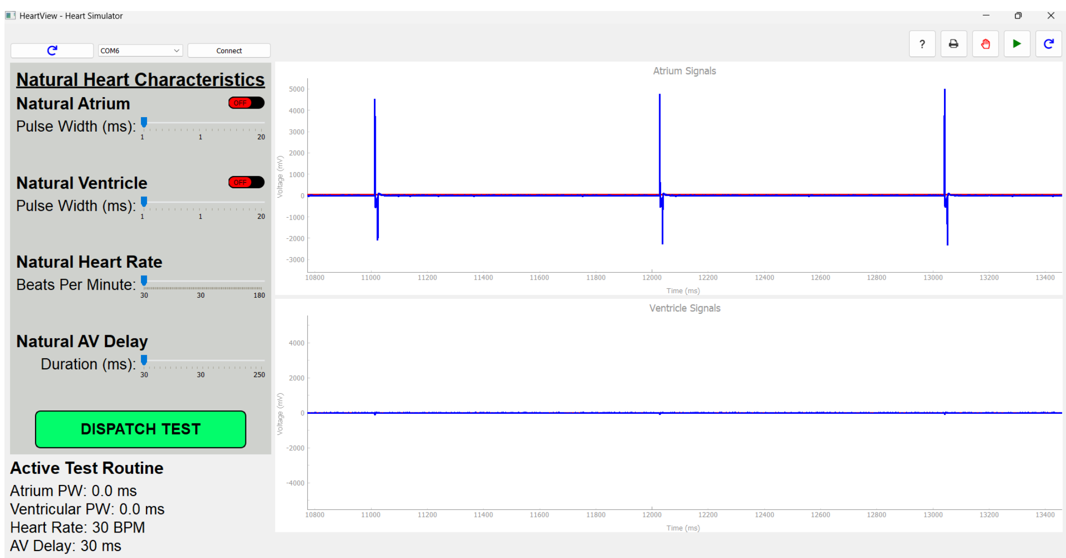

Figure 10: AOO Test

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

Figure 11: 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

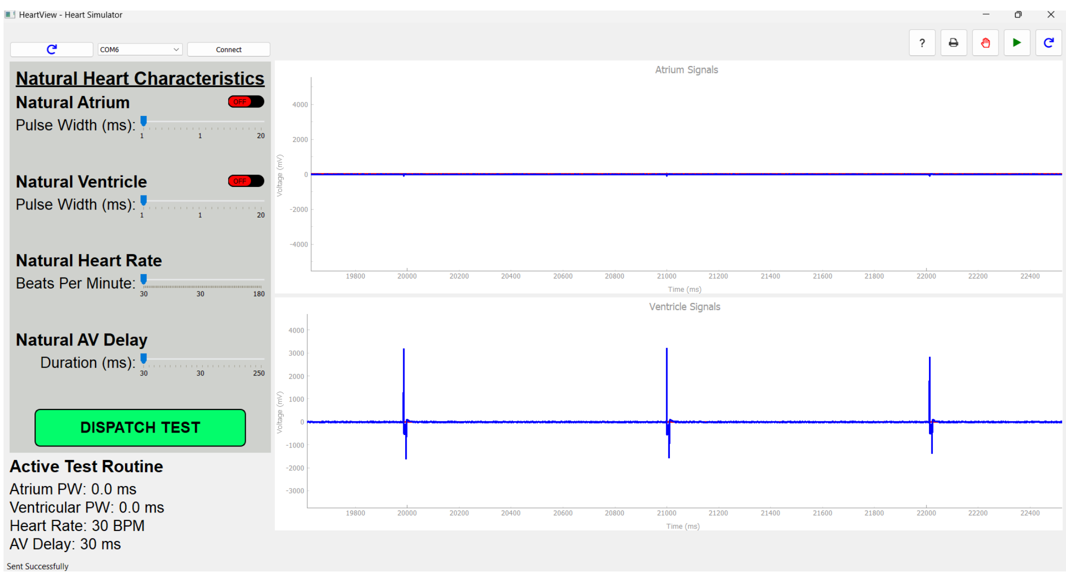

Figure 12: VOO Test

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

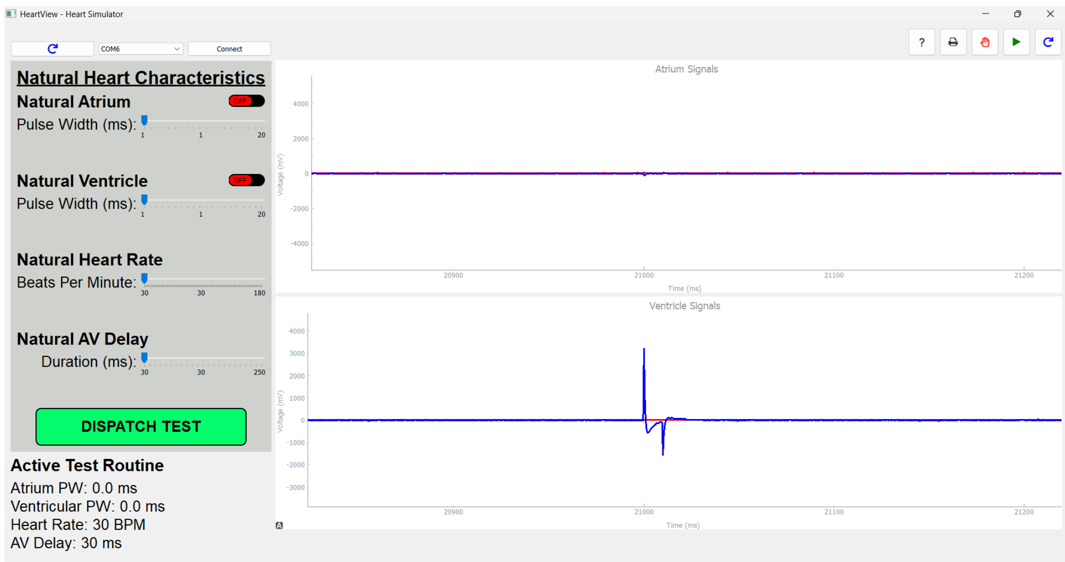

Figure 13: Close-up of VOO Pulse

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

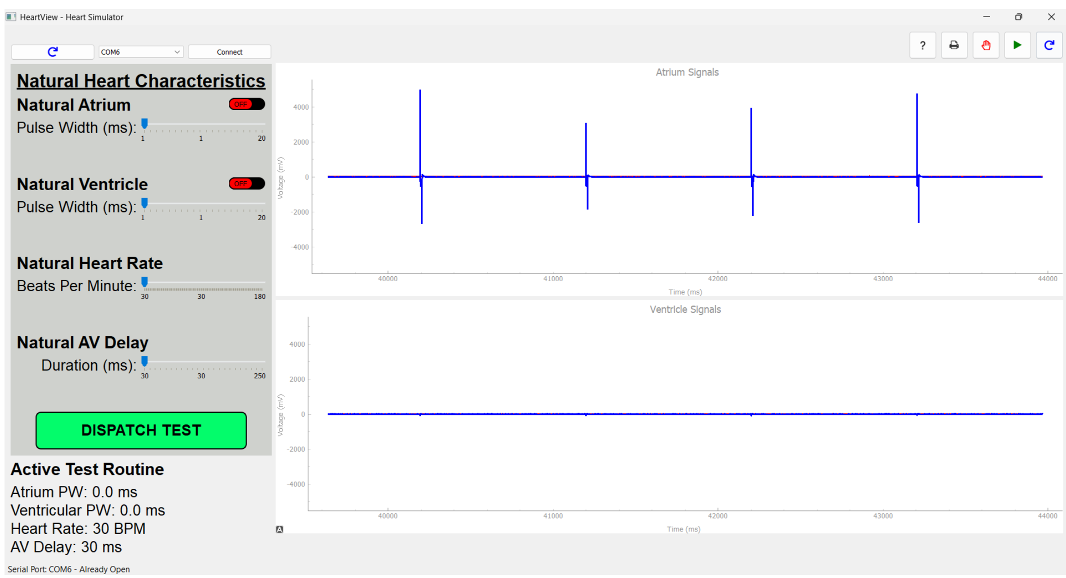

Figure 14: AAI Test No Heart Rate

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

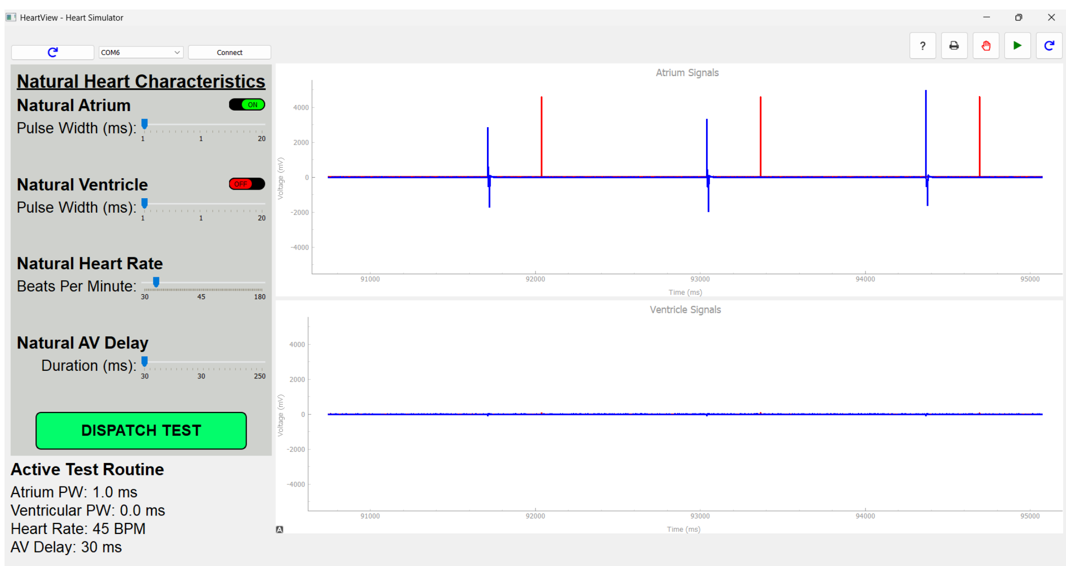

Figure 15: AAI Test 45 BPM

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

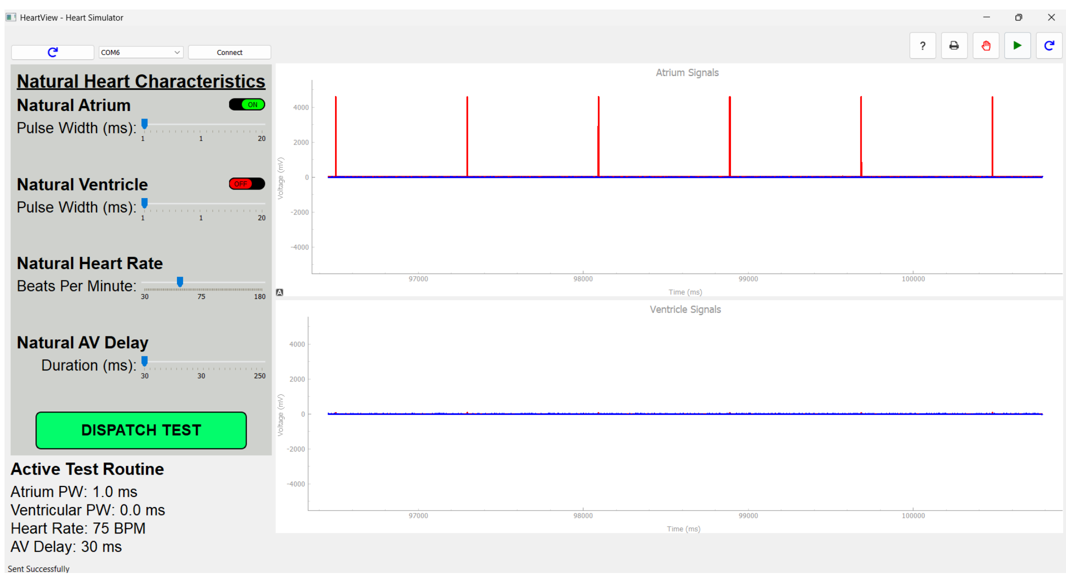

Figure 16: AAI Test 75 BPM

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

Figure 17: 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

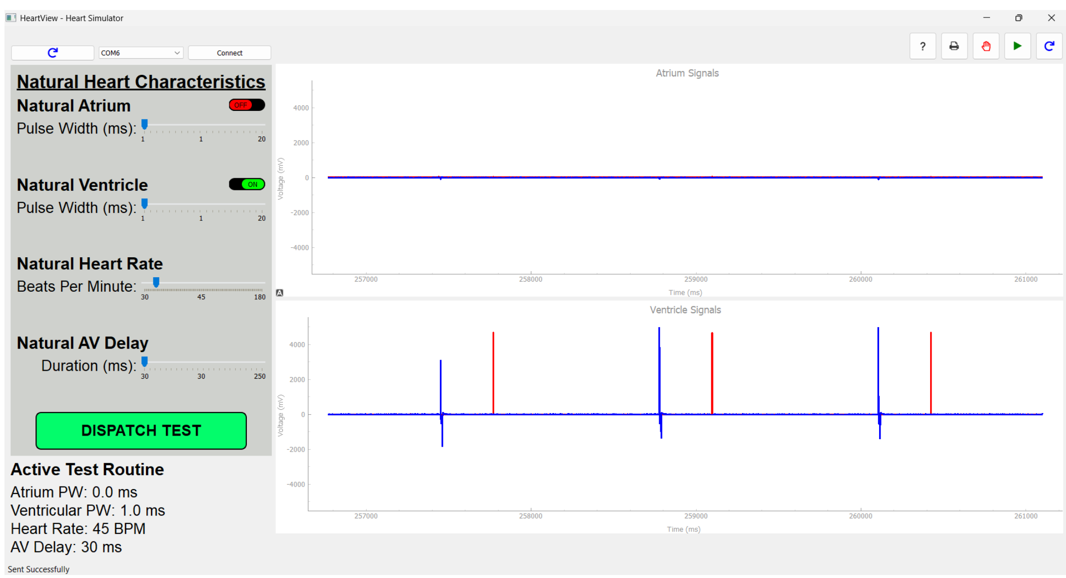

Figure 18: VVI Test 45 BPM

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

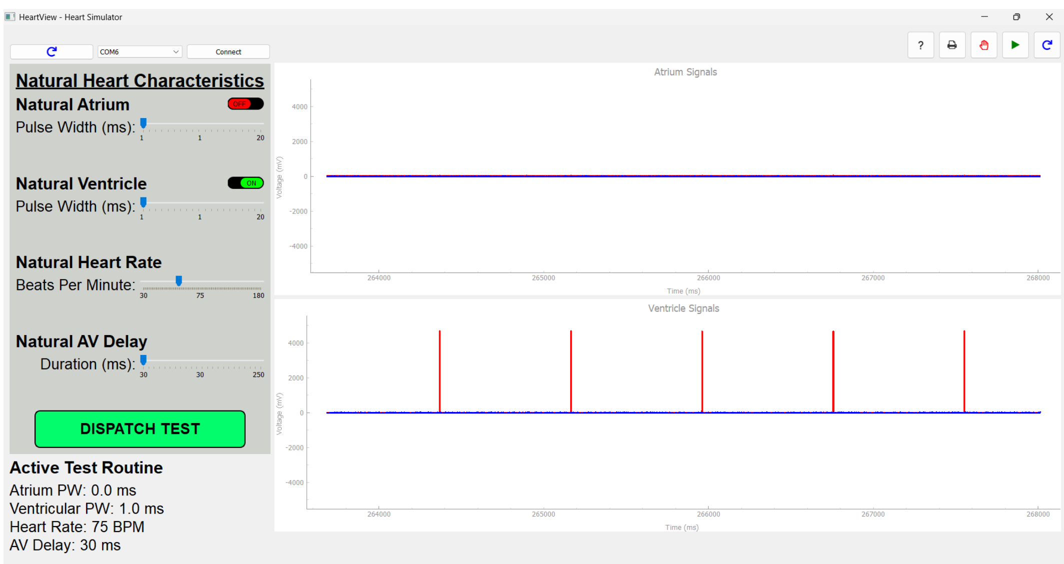

Figure 19: VVI Test 75 BPM

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

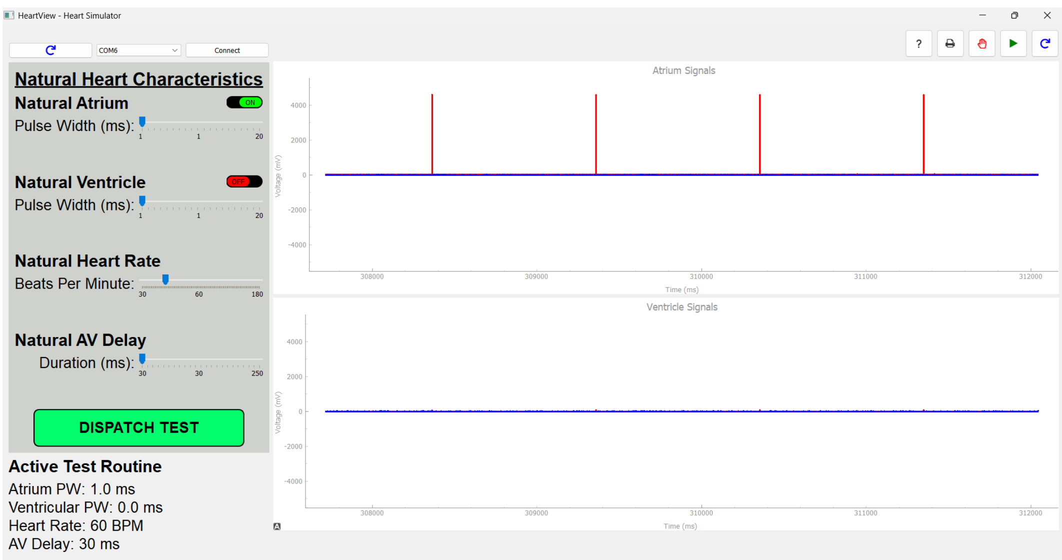

Figure 20: Hysterisis Test 1

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

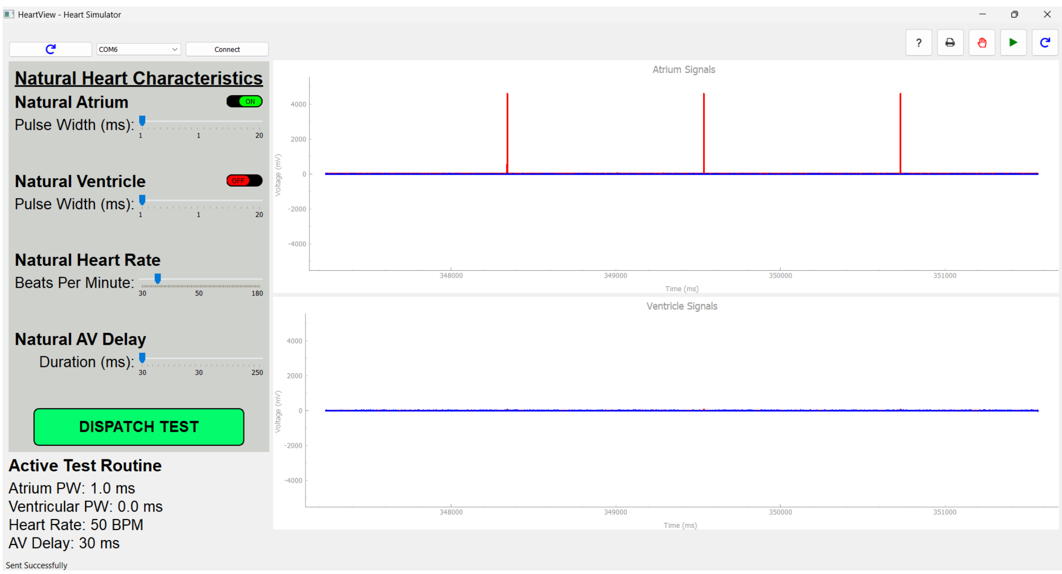

Figure 21: Hysterisis Test 2

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

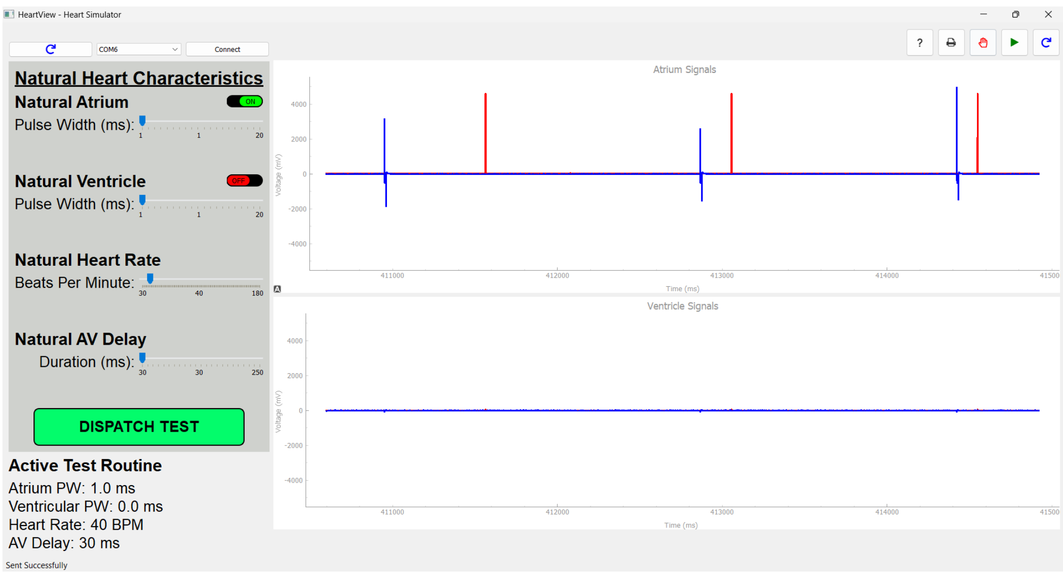

Figure 22: Hysterisis Test 3

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

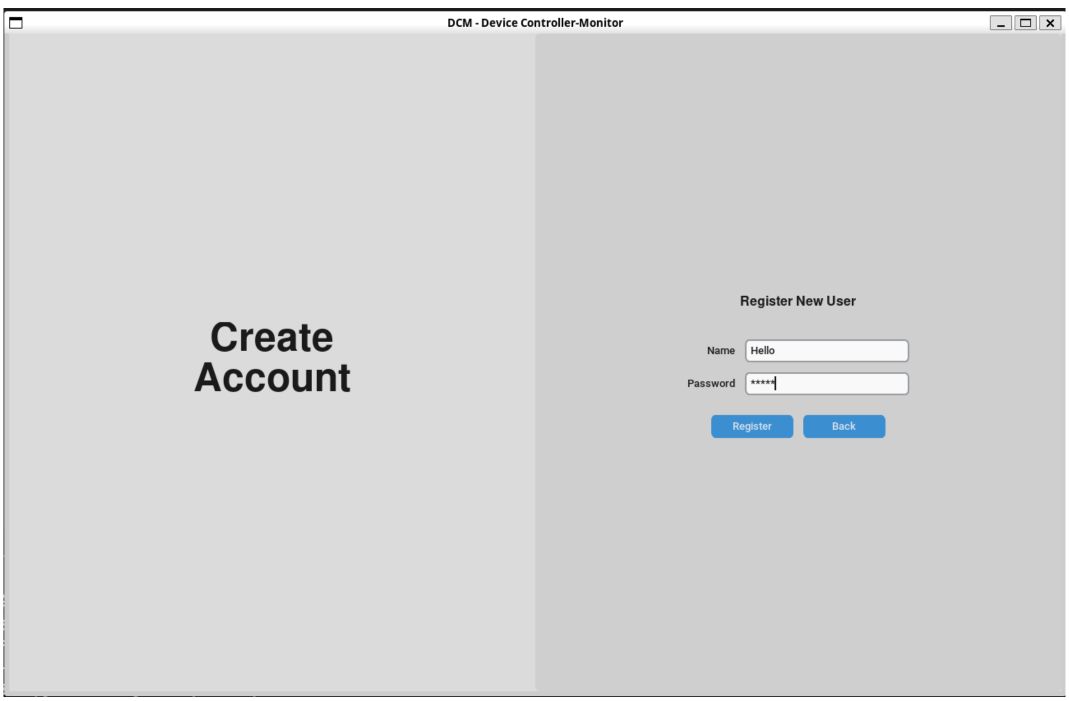

Figure 23: Registration Test

Figure 24: Registration Result

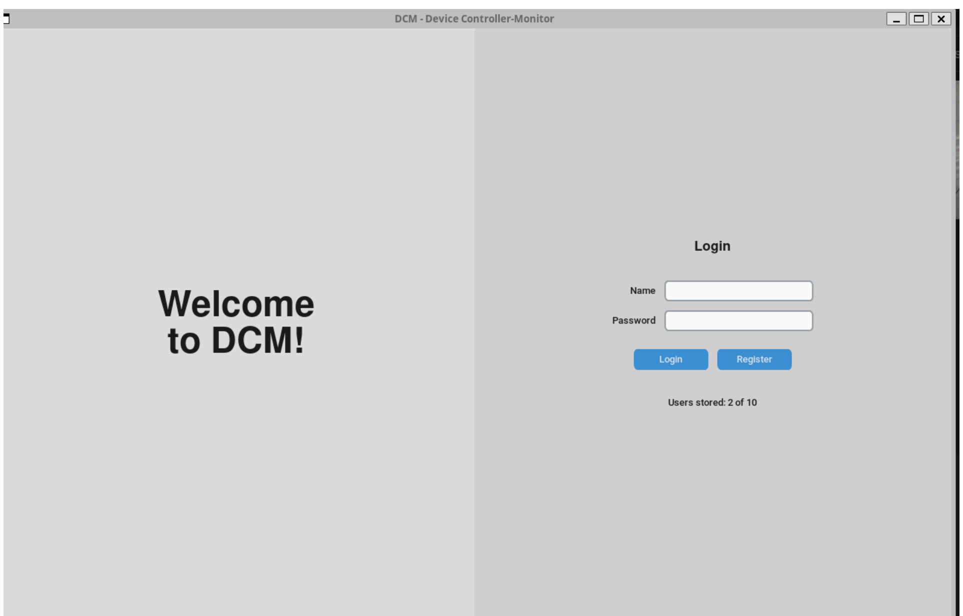

Figure 25: Login Test

Figure 26: 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

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.

5 General Notes

- This is a general outline based on the Deliverable 1 handout. You should make sure everything listed in the handout it is included.
- Use screenshots of Simulink diagrams and DCM interface where appropriate.
- Ensure the requirements are traceable to design and test cases.
- Be concise and make things clear.
- You can add other sections, and you can also decide not to use this structure, however, I am including the main general sections we will expect to see.