McMaster University

MECHTRON 3K04

Group 26

Assignment 2

November 26, 2023

Simulink Documentation

This document defines the functions and operating characteristics that the system must perform and provides a description of the system's environmental performance parameters and critical interactions.

Part 1: Requirements and design

1.1 Requirements

These are all the requirements for the Simulink implementation of the pacemaker and its modes.

1.1.1 Overall Pacemaker System

The pacemaker should provide rate adaptive bradycardia pacing support, historical data on device implementation, and user diagnostics through brady analysis function. In addition, the bradycardia analysis functions should allow pacing measurements such as lead impedance, pacing threshold, P and R wave measurement, battery status, temporary brady pacing, motion sensor trending, and tests to be performed.

1.1.2 Device Operation

The proposed medical device is designed to meticulously monitor and manage a patient's heart rate, specifically focusing on the detection and provision of therapy for bradycardia. This pivotal device features programmable functionalities, with the ability to administer single-chamber, rate-adaptive pacing, and permanent state maintenance. Importantly, the device operates in tandem with the Device Controller-Monitor (DCM), facilitating programmable controls and oversight. Additionally, the device offers comprehensive historical data, including output rate histograms for both atrial and ventricular activities, as well as essential sensor output data, further enhancing its diagnostic and therapeutic capabilities.

1.1.3 Pulse Pacing (Atrial and Ventricular)

The device should result in pulses with programmable voltages and widths for the atrium and ventricle, which provide electrical heart-pacing stimulation.

1.1.3.1 Amplitude and Width

Both amplitude and width for pulse pacing should be independently programmable for each chamber.

1.1.3.2 Rate Sensing

Bipolar electrodes and a sensing circuit should operate rate sensing. Rate detection should be based on the measured cardiac cycle measurements of the sensed rhythm and assessed on an interval-by-interval basis.

1.1.4 Operation Modes

The DCM will be used to select a permanent bradycardia operating mode for the pacemaker at startup. At present, it will choose between the modes AOO, VOO, AAI, VVI, AOOR, VOOR, AAIR, and VVIR. Table 1.0 outlines the features present in each mode.

	AOO	VOO	AAI	VVI	AOOR	VOOR	AAIR	VVIR
LRL	O	O	O	O	O	O	O	O
URL	O	O	O	O	O	O	O	O
Atrial Amplitude	O		O		O		O	
Ventricular Amplitude		O		O		O		O
Atrial Pusle Width	O		O		O		O	

Ventricular Pulse Width		O		O		O		O
Atrial Sensitivity			O				O	
Ventricular Sensitivity				O				O
ARP			O				O	
VRP				O				O
Maximum Sensor Rate					O	O	O	O
Activity Threshold					O	O	O	O
Reaction Time					O	O	O	O
Response Factor					O	O	O	O
Recovery Time					O	O	O	O

Table 1: Programmable Parameters for AOO, VOO, AAI, VVI, AOOR, VOOR, AAIR, and VVIR modes

1.1.5 Bradycardia Therapy

1.1.5.1 Lower Rate Limit (LRL)

The number of generator pace pulses delivered (per minute) should be affected by following the requirements.

a) LRL should be the longest allowable definition of pacing interval.
b) The LRL interval should begin at a ventricular sensed or paced event in VII and VOO mode.
c) The LRL interval should begin at an atrial sensed or paced event in AII and AOO mode.

1.1.5.2 Upper Rate Limit (URL)

The Upper Rate Limit (URL) refers to the highest pace at which the heart's ventricles will follow the detected atrial signals. The URL interval represents the shortest duration between one ventricular event and the subsequent ventricular pacing.

1.1.5.3 Refractory Periods

a) **Ventricular Refractory Period (VRP)**: It should be the set duration of time after a ventricular event during which neither the sensing of ventricular activity will prevent nor initiate ventricular pacing.
b) **Atrial Refractory Period (ARP)**: In single-chamber atrial modes, it should be the preset time duration following an atrial event, within which atrial events will neither stop nor initiate pacing.
c) **Post Ventricular Atrial Refractory Period (PVARP)**: It must be available in modes with ventricular pacing and atrial sensing. It is a customizable period of time after a ventricular event during which an atrial heart event should neither block an atrial pace nor activate a ventricular pace.

1.1.5.4 Rate-Adaptive Pacing

The device must be able to adjust the cardiac cycle according to metabolic needs, as measured by physical exercise using an accelerometer.

a) **Maximum Sensor Rate (MSR):** MSR is the maximum pacing rate allowed due to sensor control, which should be required for rate adaptive modes and independently programmable from the URL.
b) **Activity Threshold:** The activity threshold is the value that the accelerometer sensor result must exceed before the rate of the pacemaker is affected by the activity data.

c) **Response Factor:** The accelerometer must decide the pacing rate at different levels of steady-state patient activity. Based on the corresponding patient activity, the most significant response factor setting, which means 16, should allow the largest incremental change in rate, and the smallest response factor setting, which means 1, should allow the least significant change in rate.

d) **Reaction Time:** The accelerometer must determine the increase rate in the pacing rate, and reaction time refers to the duration needed for the activity to elevate the rate from the Lower Rate Limit (LRL) to the Maximum Sensor Rate (MSR).

e) **Recovery Time:** The accelerometer must determine the increase rate in the pacing rate, and recovery time refers to the duration needed for the rate to drop from the Maximum Sensor Rate (MSR) to the Lower Rate Limit (LRL) when activity drops below the activity threshold.

1.1.6 Programmable Parameters

Parameter	Programmable Values	Increment	Nominal	Tolerance
Modes	AOO	–	DDD	–
	VOO			
	AAI			
	VVI			
	AOOR			
	VOOR			
	AAIR			
	VVIR			
Lower Rate Limit	30 – 50 ppm	5 ppm	60 ppm	± 8 ms
	50 – 90 ppm	1 ppm		
	90 – 175 ppm	5 ppm		
Upper Rate Limit	50 – 175 ppm	5 ppm	120 ppm	± 8 ms
A or V Pulse Amplitude Regulated	Off, 0.1 – 5.0 V	0.1 V	5V	± 12 %
A or V Pulse Width	1 – 30 ms	1 ms	1 ms	1 ms
Ventricular Refractory Period	150 – 500 ms	10 ms	320 ms	± 8 ms
Atrial Refractory Period	150 – 500 ms	10 ms	250 ms	± 8 ms
Maximum Sensor Rate	50 – 175 ppm	5 ppm	120 ppm	± 4 ms
A or V Sensitivity	0 – 5V	0.1V	–	± 2 %
Activity Threshold	V-Low, Low, Med-Low, Med, Med-High, High, V-High	–	Med	–
Reaction Time	10 – 50 sec	10 sec	30 sec	± 3 sec
Response Factor	1 – 16	1	8	–
Recovery Time	2 – 16 min	1 min	5 min	± 30 sec

Table 2: Programmable Parameters Values

1.1.7 Serial Communications

The pacemaker will use UART serial communication to receive parameters from the DCM and send data about the atrium and ventricles pacing to the DCM. A baud rate of 115200 is used with a parity bit to ensure correct data transmission.

1.1.7.1 Parameter Block (from DCM)

Parameters for the pacemaker's operation will be received in a 16-byte packet as outlined below. To enable them to fit into an 8-bit integer, some parameters will be scaled before transmission. The pacemaker will rescale them to their original size upon receipt.

Item	Start Byte Index	Size (B)	Scaling	Scaled Data Range
Mode	1	1	x1	0-7
LRL	2	1	x1	30-175
URL/MSR	3	1	x1	50-175
Atrial Amplitude	4	1	x10	1-50
Ventricular Amplitude	5	1	x10	1-50
Atrial Pulse Width	6	1	x1	1-30
Ventricular Pulse Width	7	1	x1	1-30
VRP	8	1	x0.1	15-50
ARP	9	1	x0.1	15-50
Activity Threshold	10	1	x1	0-6
Reaction Time	11	1	x1	10-50
Response Factor	12	1	x1	1-16
Recovery Time	13	1	x1	2-16
Reserved	14	1	--	N/A
Reserved	15	1	--	N/A
Reserved	16	1	--	N/A

Bytes 1-13 contain the parameters outlined in section 1.1.6. Bytes 14 through 16 are reserved for future use and should be set to 0x0.

1.1.7.2 ECG Block (to DCM)

The pacemaker will also periodically send the DCM information about the paces present in each heart chamber.

Item	Start Byte Index	Size (B)	Data
Atrium (natural)	1	1	Value of ATR_CMP_DETECT
Atrium (paced)	2	1	Value of ATR_PACE_CTRL
Ventricle (natural)	3	1	Value of VENT_CMP_DETECT
Ventricle (paced)	4	1	Value of VENT_PACE_CTRL
Reserved	5	1	N/A

A new packet is sent whenever a natural pace is detected or an artificial one is delivered, so one of the four data values in the packet will always be high (0x1). Byte 5 is reserved and can be used for debugging to output other parameters within the Simulink model.

1.1.8 Rate Adaptive Pacing

1.1.8.1 Open Loop Rate Adaptive Pacemaker System

The looping should start from the indicator, and when it indicates the body's needs, it should be sent through the sensor. The sensor measures patients' acceleration and produces an electrical signal. Then, the sensor results must be sent to the pacemaker's software function, which is an algorithm, and it should convert the electrical signals into the correct pacing response. Then, the microcontroller drives the output circuit b, showing the logic, and the pacemaker shield should interface with the heart.

1.2 Design decisions

One subsystem within the system's architecture is dedicated to the task of mapping input pins, which are responsible for conveying data used in the program, to their respective names as specified in Table 1 of the document entitled "Pacemaker Shield Explained." Another subsystem is employed to map output pins to their designated names from the same referenced table. This dual mapping approach serves to enhance program readability and obfuscate the underlying hardware complexities. By doing so, it facilitates the use of variable names like "ATR_CMP_DETECT" instead of generic identifiers such as "D0" throughout the codebase, thus fostering clarity and abstraction between the software and hardware components.

In pursuit of adaptability and ease of parameter adjustment, the program incorporates the usage of constants for programmable parameters. This design choice ensures that the values associated with these parameters can be modified with utmost simplicity, promoting versatility across different simulation runs. It is anticipated that in future iterations of the system, these parameter values will be dynamically supplied by the Device Controller-Monitor (DCM) through serial communication, thereby allowing for real-time adjustment and fine-tuning.

The system's operational modes are implemented using a stateflow diagram, which branches into the appropriate modes based on the input data provided. Furthermore, nested stateflows are employed to implement and manage the distinct states required for each pacing mode. This hierarchical state management approach enhances the overall system's organization and comprehensibility while ensuring the correct execution of various pacing functionalities.

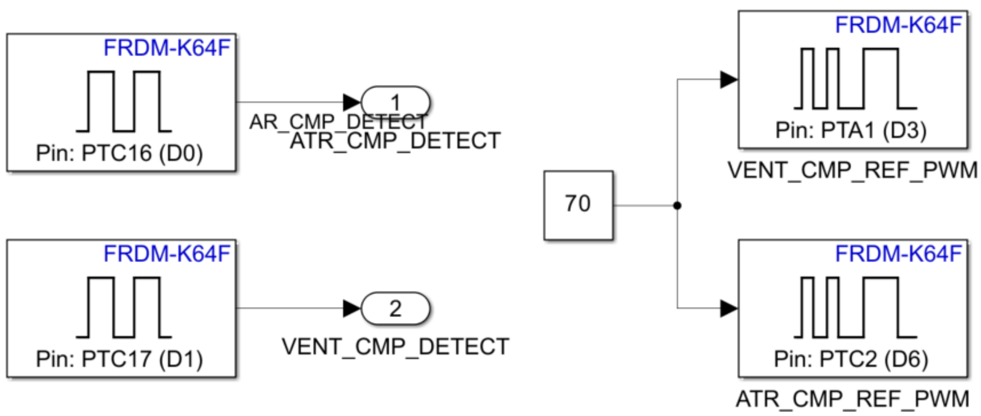
1.2.1 Serial Communication

Packets were designed to be small to promote efficiency and readability within the code. All the parameters being passed to the pacemaker were found to be within the resolution of an unsigned, 8-bit integer (0-255) and as such no byte packing or unpacking is needed. The parameters which were outside of the normal range of a byte were able to be scaled without any loss of resolution. For example, VRP and ARP both have maximal values of 500, greater than 255, but as they can only be incremented by 10, it is possible to divide this parameter by 10 to fit it into a byte, and then multiply it upon receipt to regain the exact original value.

1.3 Simulink diagrams and testing

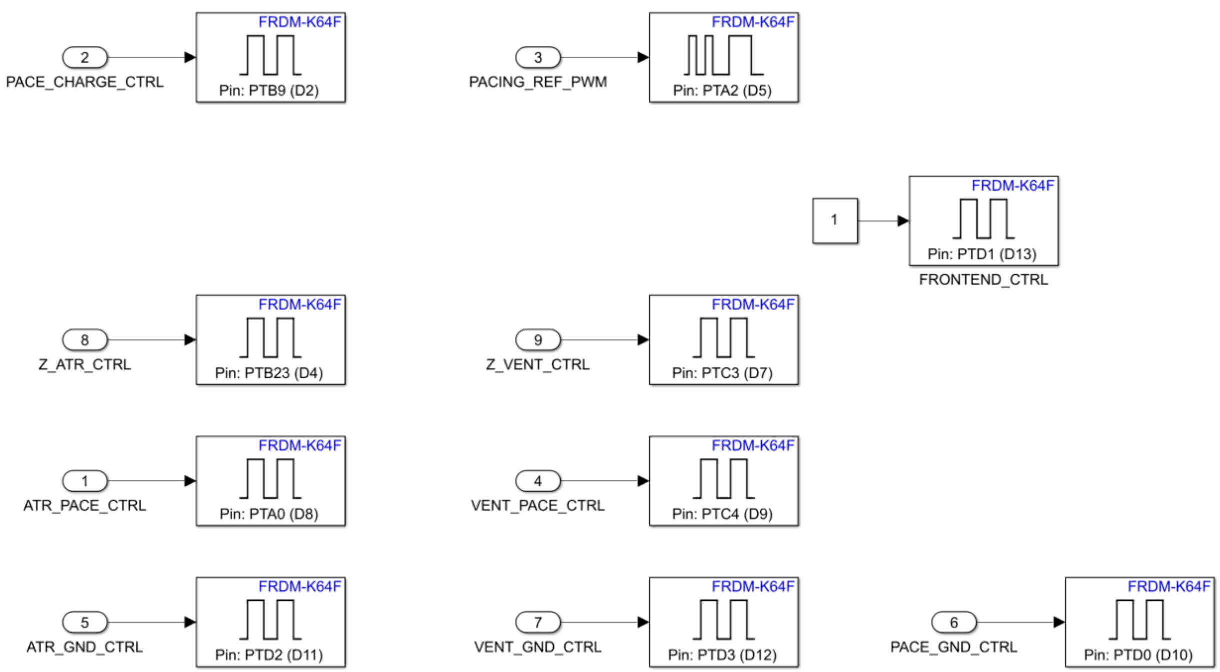
1.3.1 Simulink Diagram

1.3.1.1 Inputs



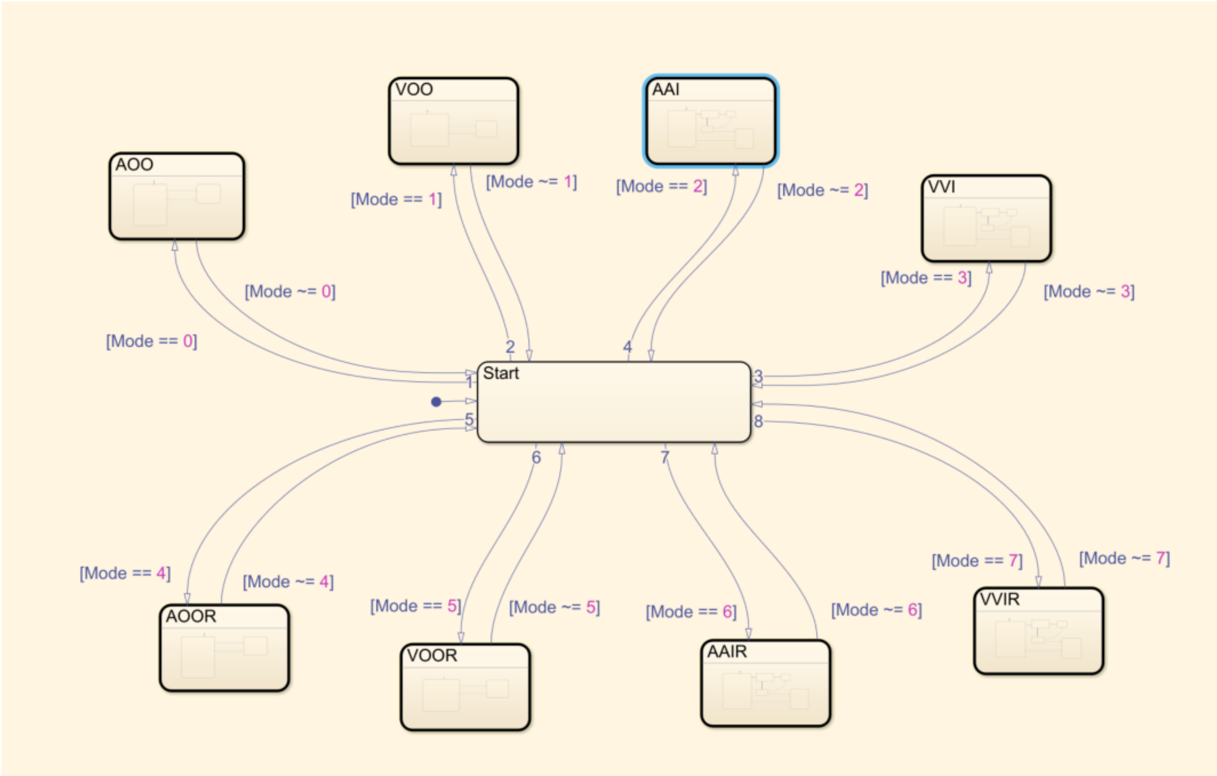
In this subsystem the digital compare pins are assigned, and the PWM threshold for their sensing is set.

1.3.1.2 Outputs

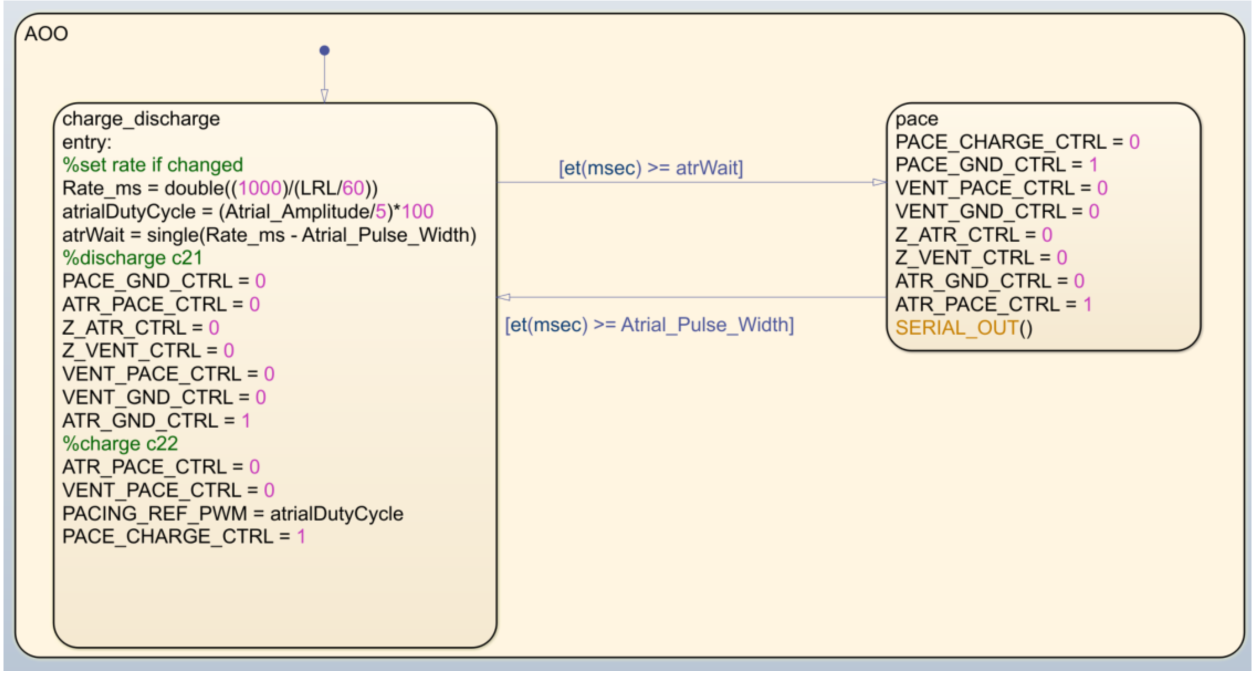


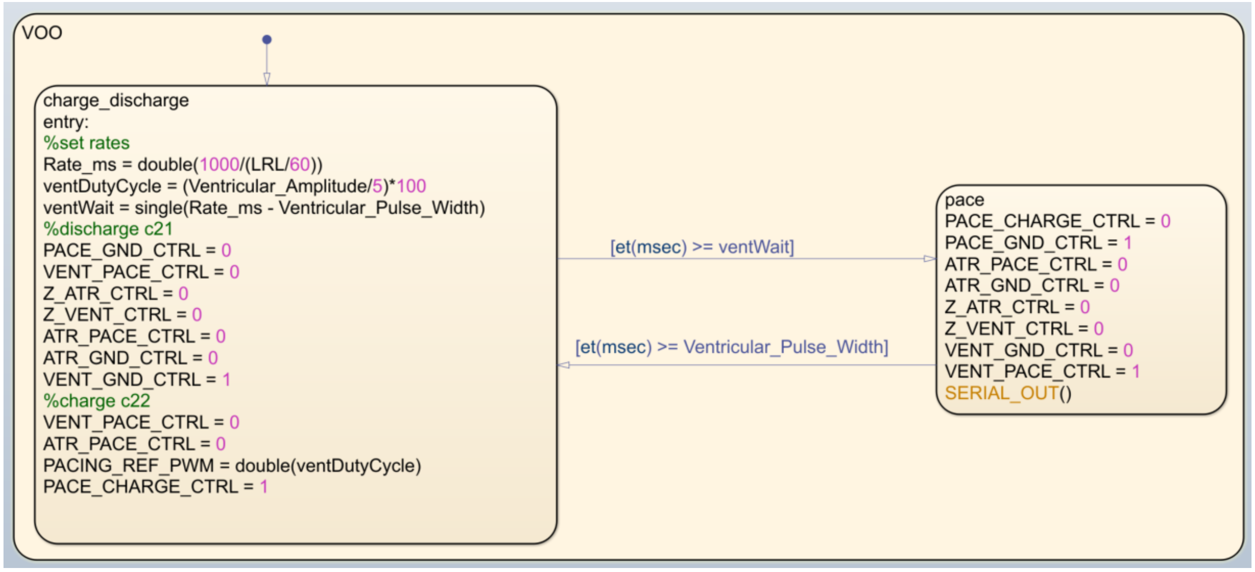
Here, outputs from the pacing stateflow are assigned to the various digital and analog inputs of the board.

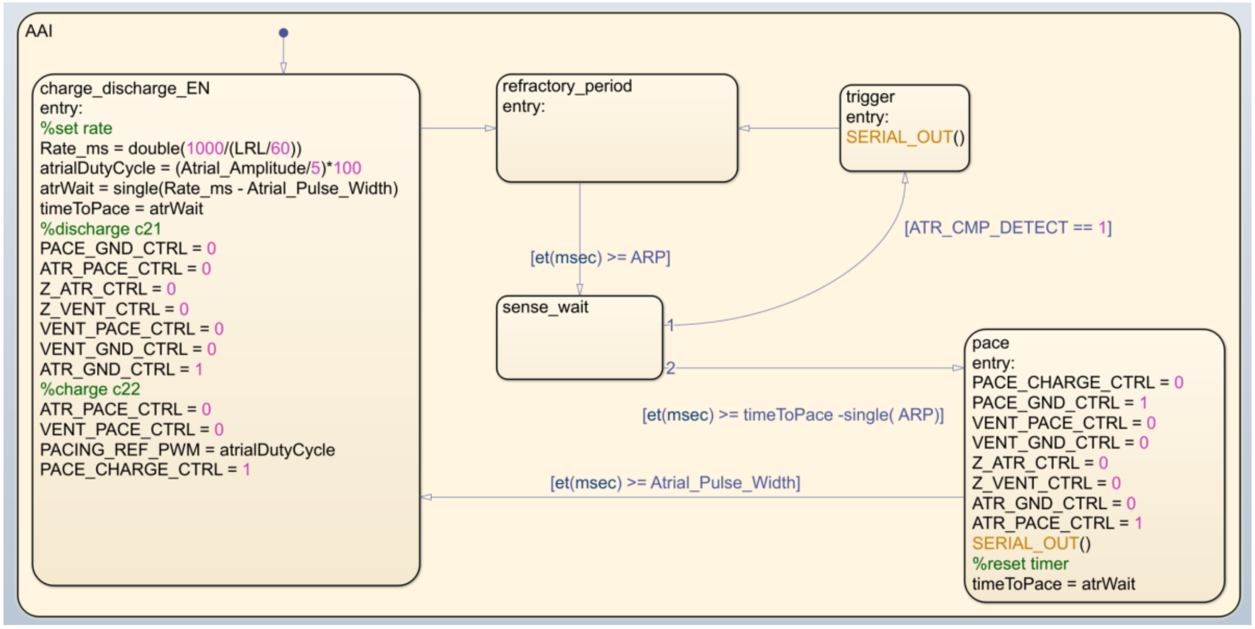
1.3.1.3 Pacing Stateflows

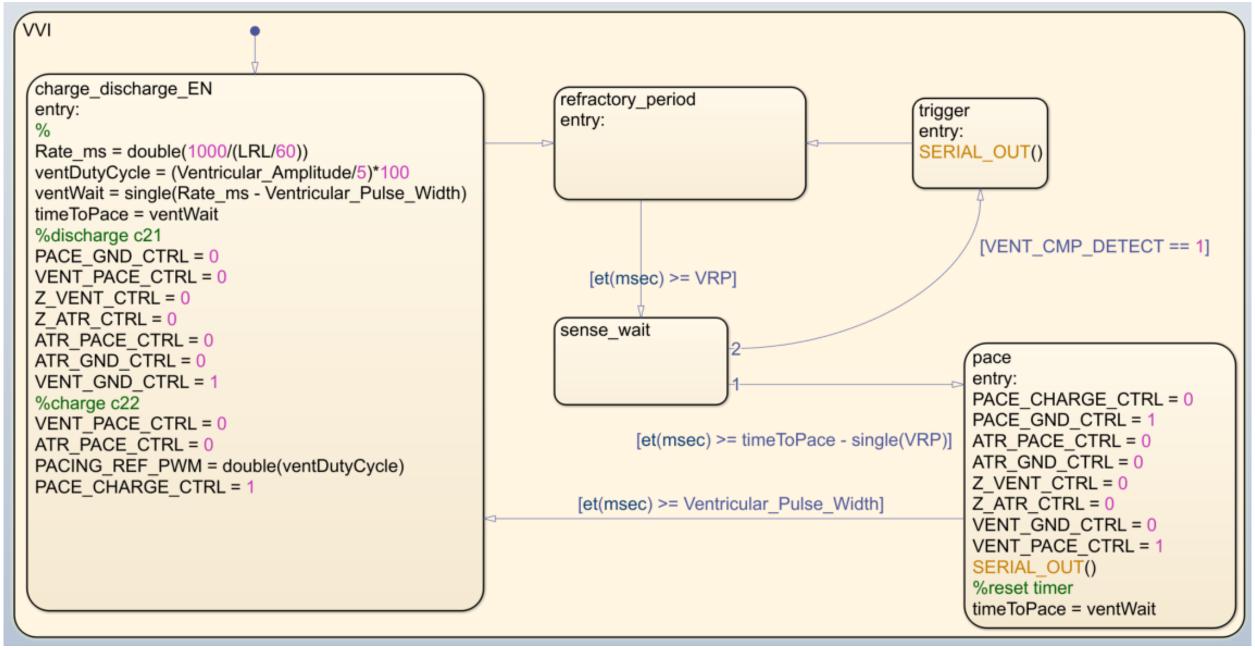


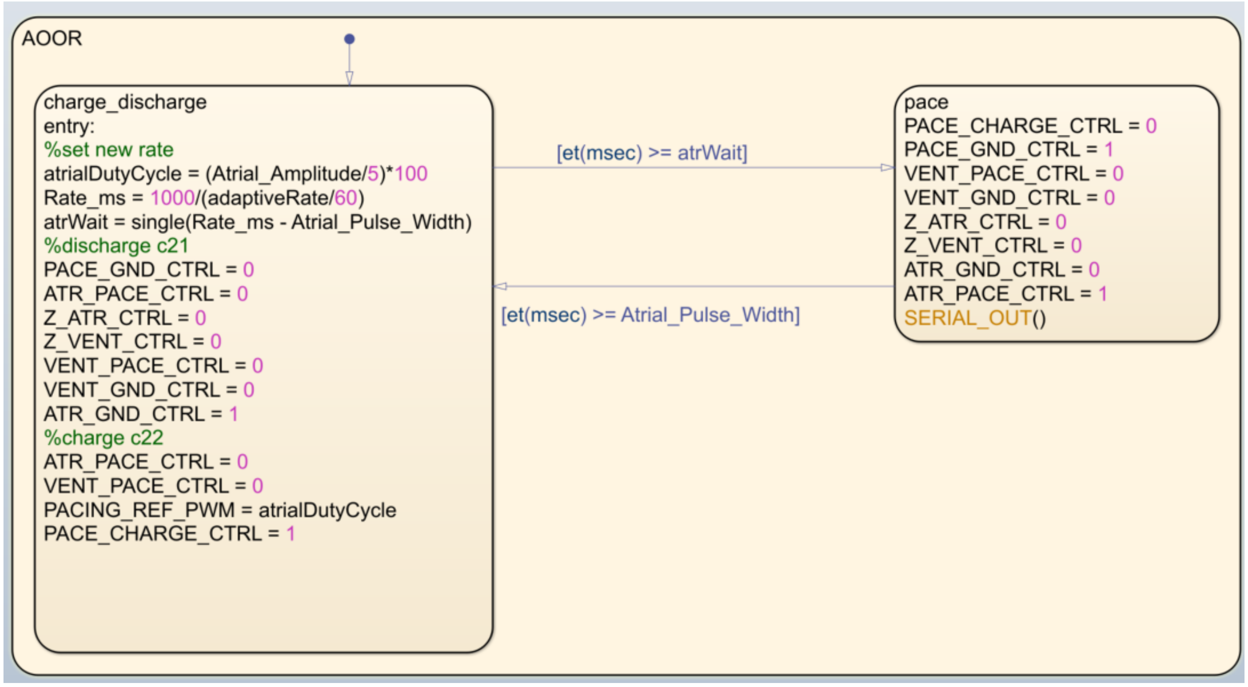
The number of pacing states has doubled to include the rate adaptive counterparts of the previously implemented modes. Additionally, due to the dynamic assignment of mode from the DCM, the pacing states can now be exited when their mode is no longer selected. Each pacing mode is below.

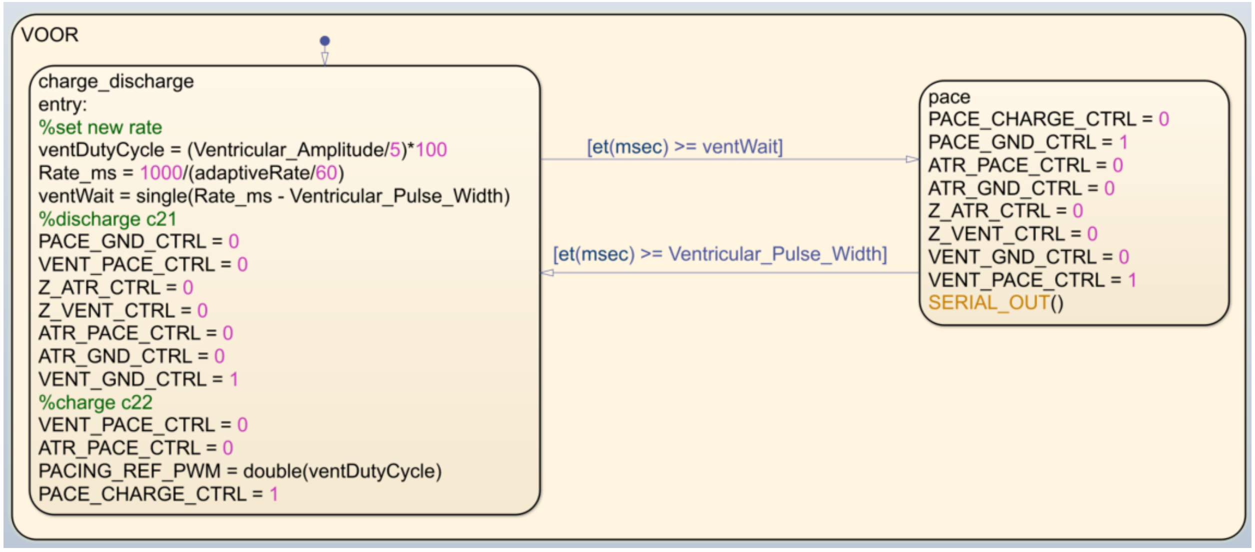


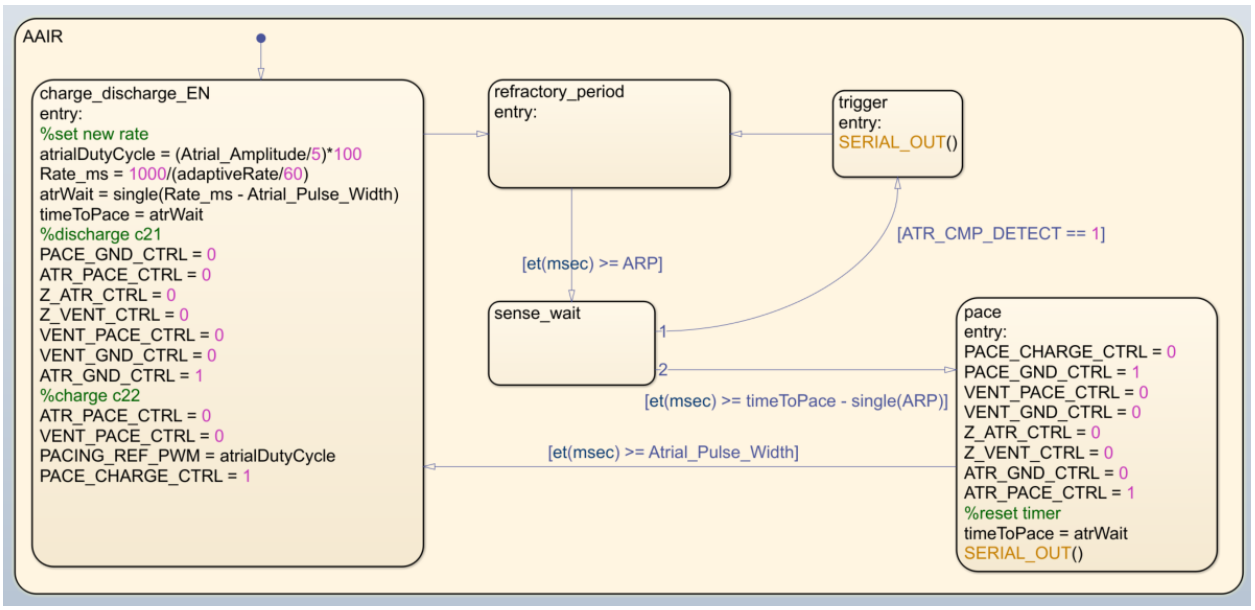

VOO
charge_discharge
entry:
%set rates
Rate_ms = double(1000/(LRL/60))
ventDutyCycle = (Ventricular_Amplitude/5)*100
ventWait = single(Rate_ms - Ventricular_Pulse_Width)
%discharge c21
PACE_GND_CTRL = 0
VENT_PACE_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
ATR_PACE_CTRL = 0
ATR_GND_CTRL = 0
VENT_GND_CTRL = 1
%charge c22
VENT_PACE_CTRL = 0
ATR_PACE_CTRL = 0
PACING_REF_PWM = double(ventDutyCycle)
PACE_CHARGE_CTRL = 1
[et(msec) >= ventWait]
[et(msec) >= Ventricular_Pulse_Width]
pace
PACE_CHARGE_CTRL = 0
PACE_GND_CTRL = 1
ATR_PACE_CTRL = 0
ATR_GND_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
VENT_GND_CTRL = 0
VENT_PACE_CTRL = 1
SERIAL_OUT()

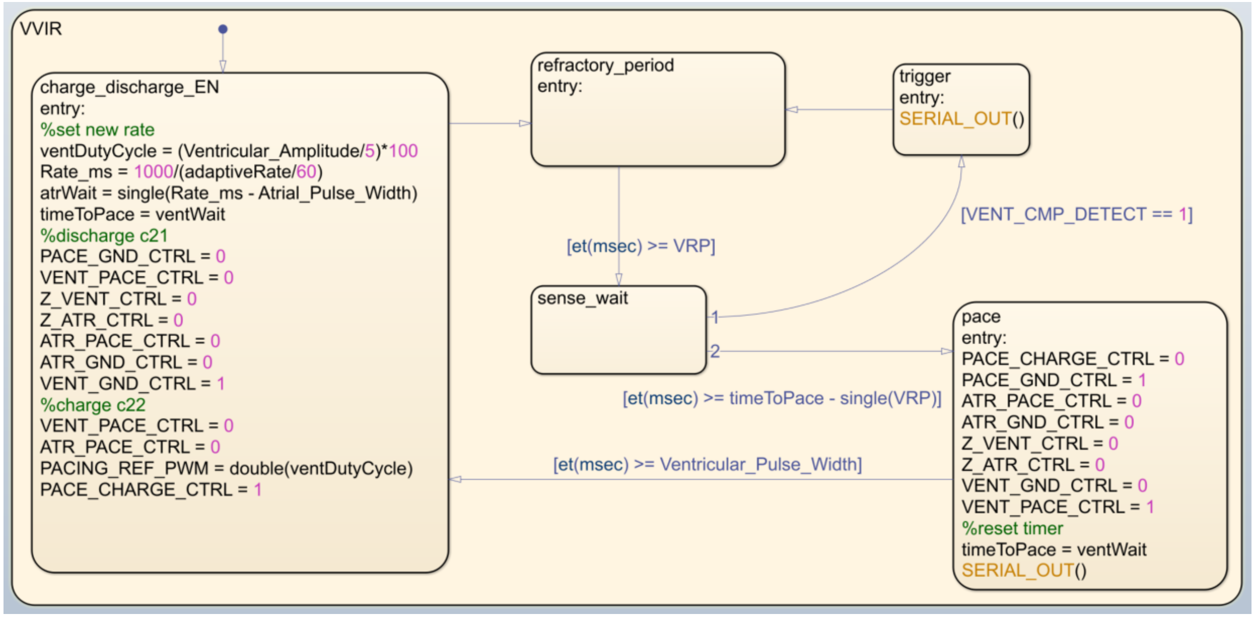

AAI
charge_discharge_EN
entry:
%set rate
Rate_ms = double(1000/(LRL/60))
atrialDutyCycle = (Atrial_Amplitude/5)*100
atrWait = single(Rate_ms - Atrial_Pulse_Width)
timeToPace = atrWait
%discharge c21
PACE_GND_CTRL = 0
ATR_PACE_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
VENT_PACE_CTRL = 0
VENT_GND_CTRL = 0
ATR_GND_CTRL = 1
%charge c22
ATR_PACE_CTRL = 0
VENT_PACE_CTRL = 0
PACING_REF_PWM = atrialDutyCycle
PACE_CHARGE_CTRL = 1
refractory_period
entry:
trigger
entry:
SERIAL_OUT()
[ATR_CMP_DETECT == 1]
[et(msec) >= ARP]
sense_wait
1
2
[et(msec) >= timeToPace -single(ARP)]
[et(msec) >= Atrial_Pulse_Width]
pace
entry:
PACE_CHARGE_CTRL = 0
PACE_GND_CTRL = 1
VENT_PACE_CTRL = 0
VENT_GND_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
ATR_GND_CTRL = 0
ATR_PACE_CTRL = 1
SERIAL_OUT()
%reset timer
timeToPace = atrWait


VVI
charge_discharge_EN
entry:
%
Rate_ms = double(1000/(LRL/60))
ventDutyCycle = (Ventricular_Amplitude/5)*100
ventWait = single(Rate_ms - Ventricular_Pulse_Width)
timeToPace = ventWait
%discharge c21
PACE_GND_CTRL = 0
VENT_PACE_CTRL = 0
Z_VENT_CTRL = 0
Z_ATR_CTRL = 0
ATR_PACE_CTRL = 0
ATR_GND_CTRL = 0
VENT_GND_CTRL = 1
%charge c22
VENT_PACE_CTRL = 0
ATR_PACE_CTRL = 0
PACING_REF_PWM = double(ventDutyCycle)
PACE_CHARGE_CTRL = 1
refractory_period
entry:
trigger
entry:
SERIAL_OUT()
[VENT_CMP_DETECT == 1]
[et(msec) >= VRP]
sense_wait
2
1
pace
entry:
PACE_CHARGE_CTRL = 0
PACE_GND_CTRL = 1
ATR_PACE_CTRL = 0
ATR_GND_CTRL = 0
Z_VENT_CTRL = 0
Z_ATR_CTRL = 0
VENT_GND_CTRL = 0
VENT_PACE_CTRL = 1
SERIAL_OUT()
%reset timer
timeToPace = ventWait
[et(msec) >= timeToPace - single(VRP)]
[et(msec) >= Ventricular_Pulse_Width]


AOOR
charge_discharge
entry:
%set new rate
atrialDutyCycle = (Atrial_Amplitude/5)*100
Rate_ms = 1000/(adaptiveRate/60)
atrWait = single(Rate_ms - Atrial_Pulse_Width)
%discharge c21
PACE_GND_CTRL = 0
ATR_PACE_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
VENT_PACE_CTRL = 0
VENT_GND_CTRL = 0
ATR_GND_CTRL = 1
%charge c22
ATR_PACE_CTRL = 0
VENT_PACE_CTRL = 0
PACING_REF_PWM = atrialDutyCycle
PACE_CHARGE_CTRL = 1
[et(msec) >= atrWait]
[et(msec) >= Atrial_Pulse_Width]
pace
PACE_CHARGE_CTRL = 0
PACE_GND_CTRL = 1
VENT_PACE_CTRL = 0
VENT_GND_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
ATR_GND_CTRL = 0
ATR_PACE_CTRL = 1
SERIAL_OUT()

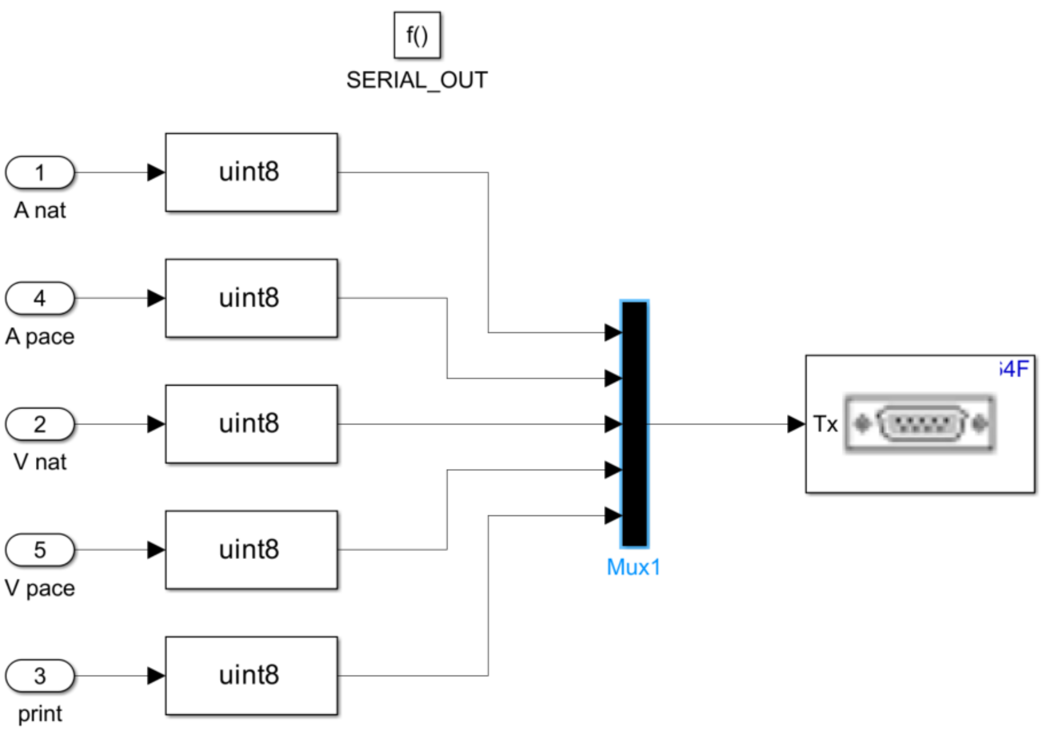

VOOR
charge_discharge
entry:
%set new rate
ventDutyCycle = (Ventricular_Amplitude/5)*100
Rate_ms = 1000/(adaptiveRate/60)
ventWait = single(Rate_ms - Ventricular_Pulse_Width)
%discharge c21
PACE_GND_CTRL = 0
VENT_PACE_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
ATR_PACE_CTRL = 0
ATR_GND_CTRL = 0
VENT_GND_CTRL = 1
%charge c22
VENT_PACE_CTRL = 0
ATR_PACE_CTRL = 0
PACING_REF_PWM = double(ventDutyCycle)
PACE_CHARGE_CTRL = 1
[et(msec) >= ventWait]
[et(msec) >= Ventricular_Pulse_Width]
pace
PACE_CHARGE_CTRL = 0
PACE_GND_CTRL = 1
ATR_PACE_CTRL = 0
ATR_GND_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
VENT_GND_CTRL = 0
VENT_PACE_CTRL = 1
SERIAL_OUT()

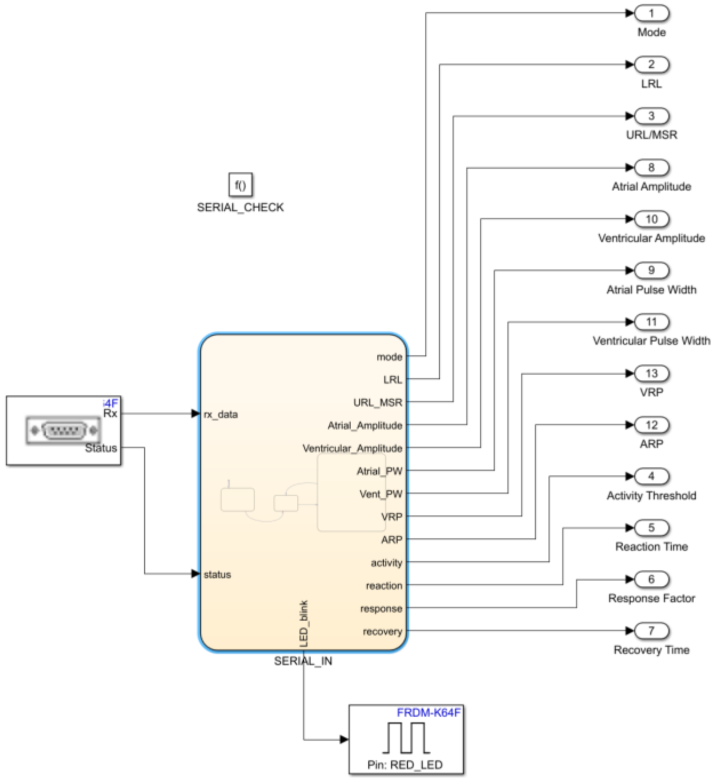

AAIR
charge_discharge_EN
entry:
%set new rate
atrialDutyCycle = (Atrial_Amplitude/5)*100
Rate_ms = 1000/(adaptiveRate/60)
atrWait = single(Rate_ms - Atrial_Pulse_Width)
timeToPace = atrWait
%discharge c21
PACE_GND_CTRL = 0
ATR_PACE_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
VENT_PACE_CTRL = 0
VENT_GND_CTRL = 0
ATR_GND_CTRL = 1
%charge c22
ATR_PACE_CTRL = 0
VENT_PACE_CTRL = 0
PACING_REF_PWM = atrialDutyCycle
PACE_CHARGE_CTRL = 1
refractory_period
entry:
trigger
entry:
SERIAL_OUT()
[ATR_CMP_DETECT == 1]
[et(msec) >= ARP]
sense_wait
1
2
[et(msec) >= timeToPace - single(ARP)]
[et(msec) >= Atrial_Pulse_Width]
pace
entry:
PACE_CHARGE_CTRL = 0
PACE_GND_CTRL = 1
VENT_PACE_CTRL = 0
VENT_GND_CTRL = 0
Z_ATR_CTRL = 0
Z_VENT_CTRL = 0
ATR_GND_CTRL = 0
ATR_PACE_CTRL = 1
%reset timer
timeToPace = atrWait
SERIAL_OUT()

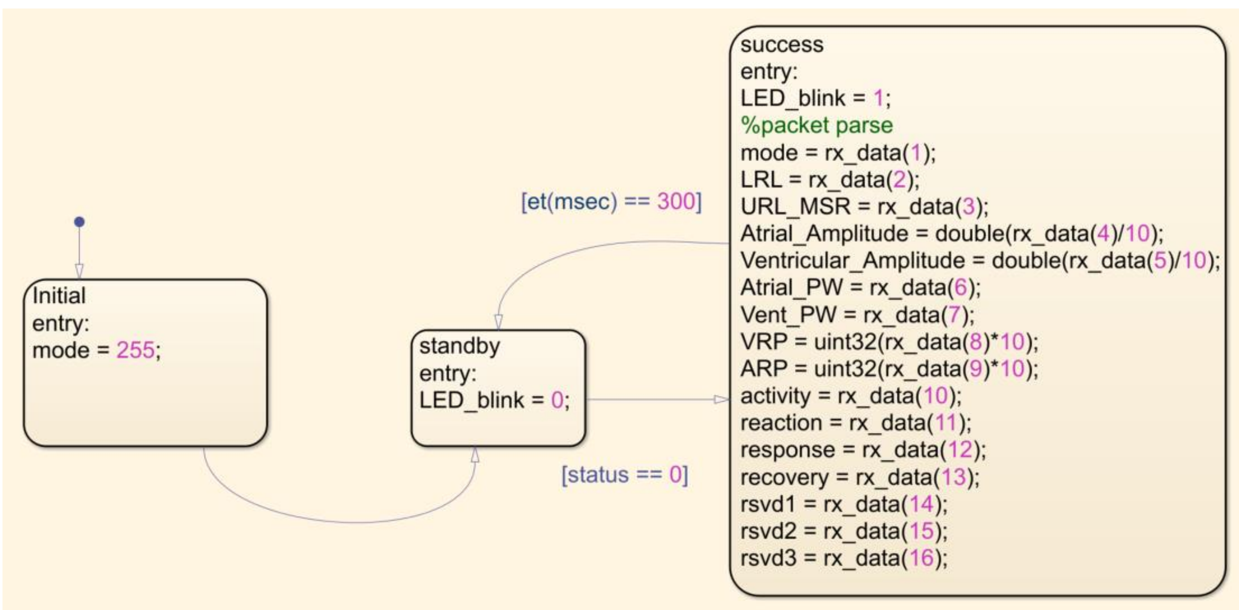


1.3.1.4 Serial Communications

This subsystem holds two self-contained subsystems to implement transmission and reception of serial data.



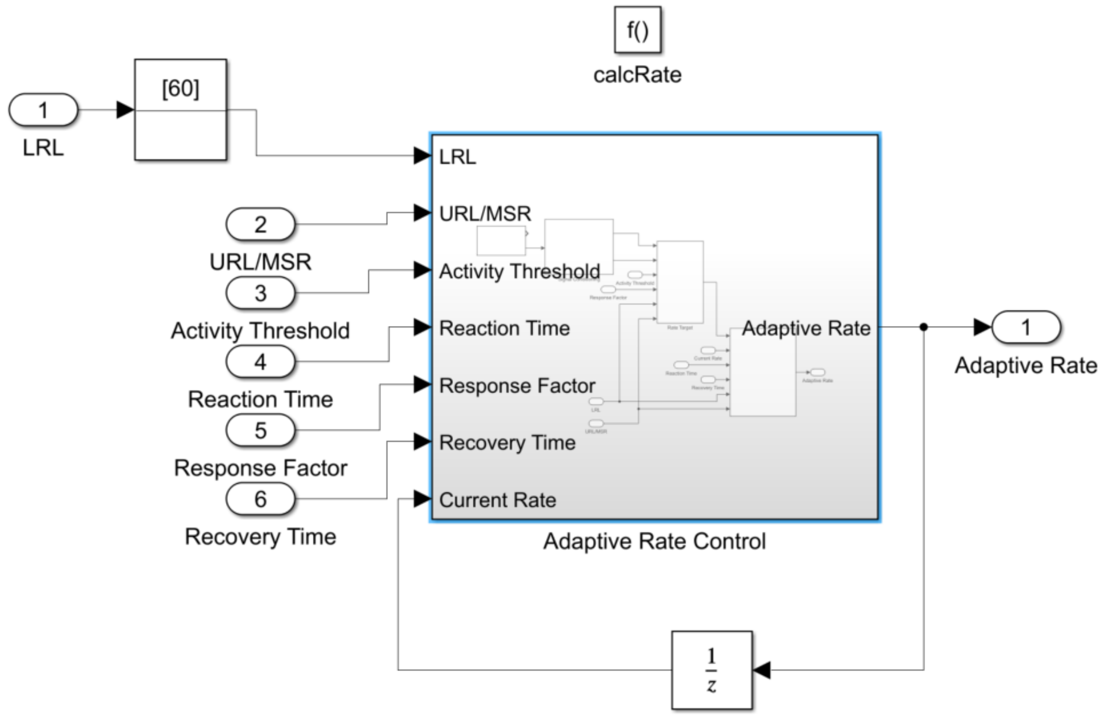




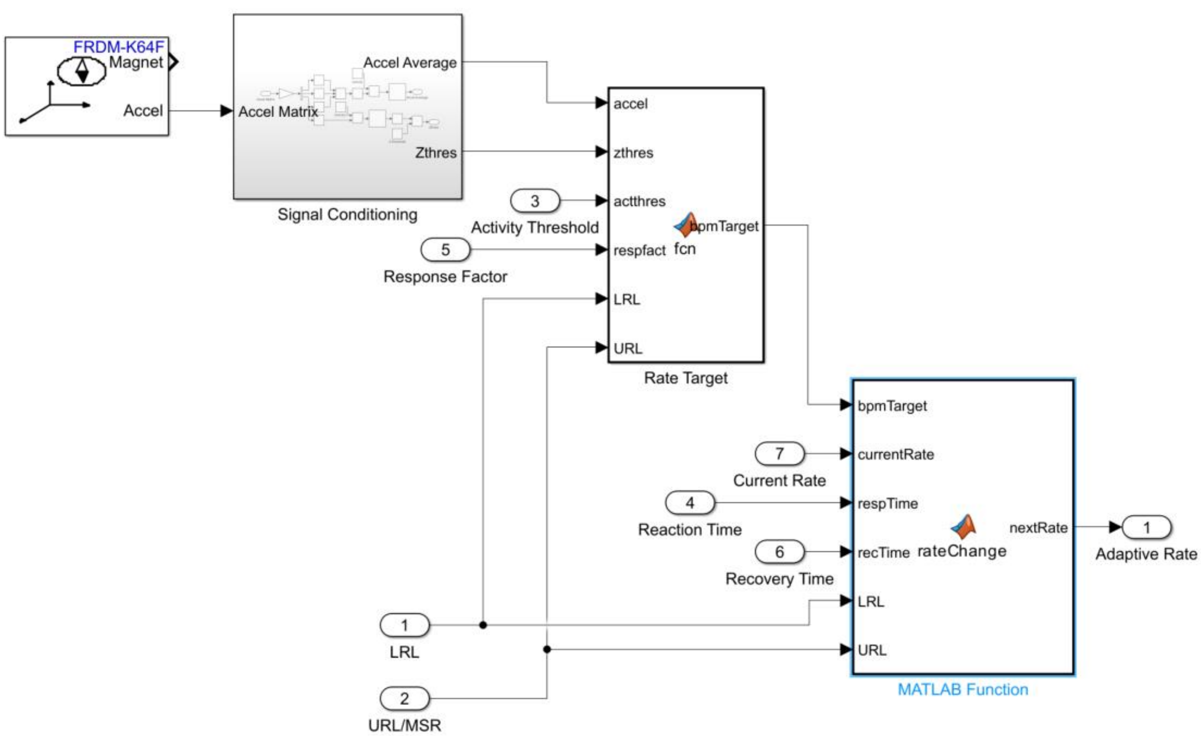
1.3.1.5 Adaptive Rate Pacing

This subsystem has another inside of it, as well as two MATLAB functions to carry out calculations.

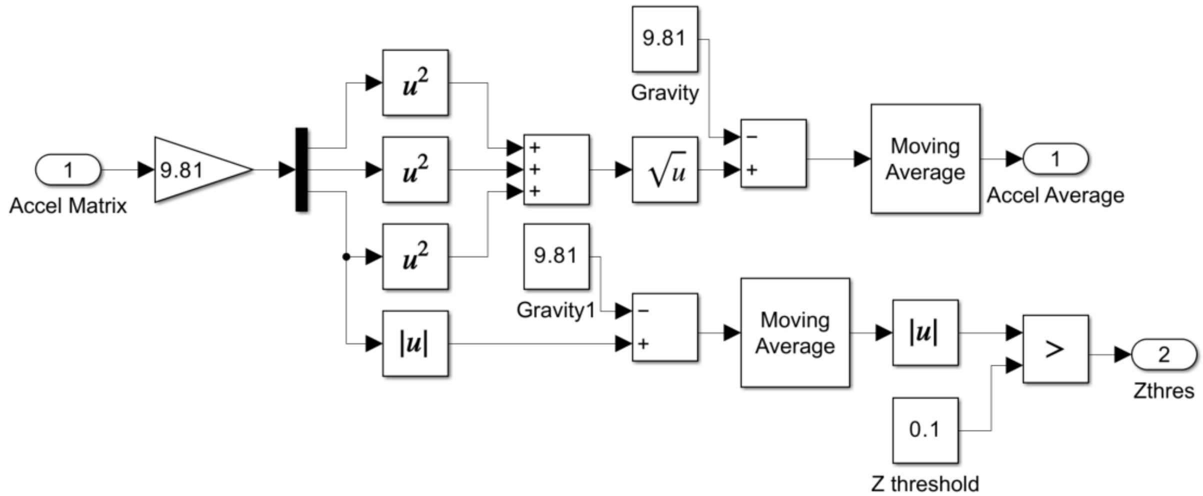
Adaptive Pacing



Adaptive Rate Control



Signal Conditioning



Rate Target

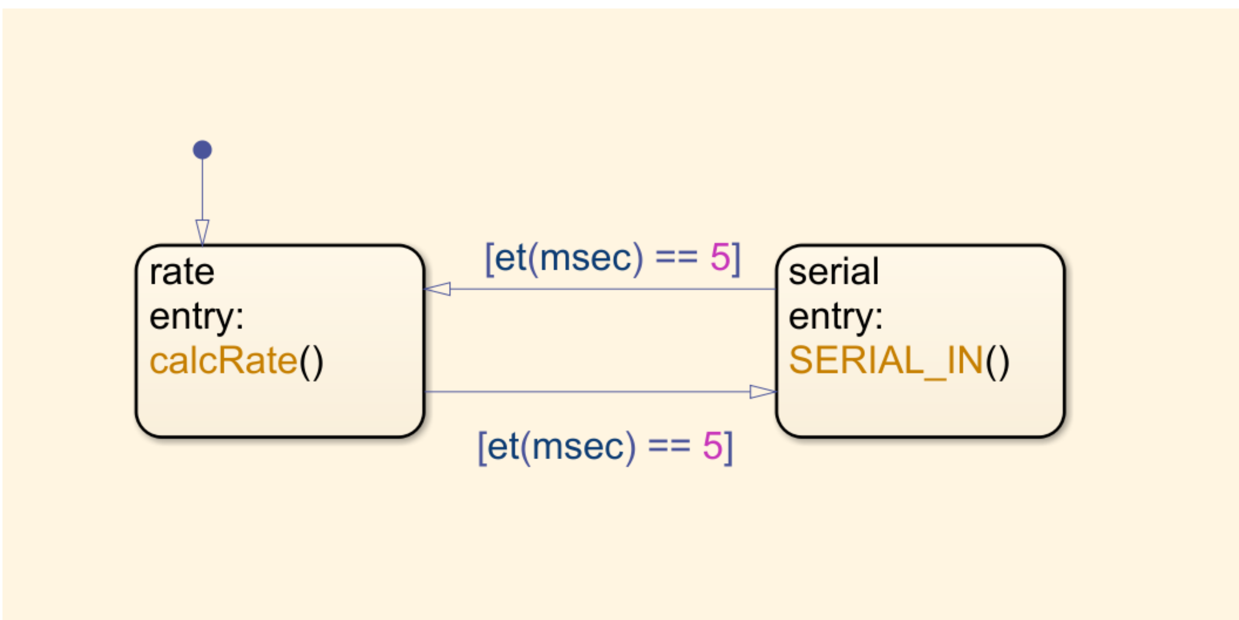
```
function bpmTarget = fcn(accel, zthres, actthres, respfact, LRL, URL)

activity = accel;

if (activity > actthres) && zthres
    diff = activity - actthres;
    bpmTarget = LRL + respfact*diff;

    if bpmTarget > URL
        bpmTarget = URL;
    elseif bpmTarget < LRL
        bpmTarget = LRL;
    end
else
    bpmTarget = LRL;
end

```

Rate Change

```
function nextRate = rateChange(bpmTarget, currentRate, respTime, recTime, LRL, URL)


if bpmTarget > currentRate
    slopemax = (URL-LRL)/(recTime*100);
    slopemin = slopemax*0.25;
else
    slopemax = (URL-LRL)/(respTime*100);
    slopemin = slopemax*0.5;
end

gap = bpmTarget - currentRate;
slope = gap/(URL-LRL) * slopemax;

if abs(slope)<slopemin && abs(gap) > 0.01
    slope = slopemin * gap/abs(gap);
end

nextRate = double(currentRate + slope);

```

1.3.1.6 Timing Stateflow

This unit was found to be necessary to keep adaptive rate pacing from interfering with serial transmission. It alternates function calls to keep them from overlapping with each other.



1.3.2 Simulink Testing

Known issue: as of Revision 2 of the pacemaker for Assignment 2, changing the pacing rate does not work. For indiscernible reasons, the pacing rate for all modes has been stuck at approximately 120 or 240 beats per minute. Non rate-adaptive tests are from Rev 1.

1.3.2.1 AOO Test

We conducted two tests to see if the AOO mode works well. First, we checked that the Pacemaker operated properly when the Natural Atrium was ON and showed the result value. Next, we tested Pacemaker's graph for the same result as the previous test case when both the Natural Atrium and Natural Ventricle were off. In AOO mode, the result value was confirmed as follows because the same value must be obtained even if both the Natural Atrium and Ventricle are turned off. The following shows the corresponding graph and the settings for Pacemaker and Heartview.

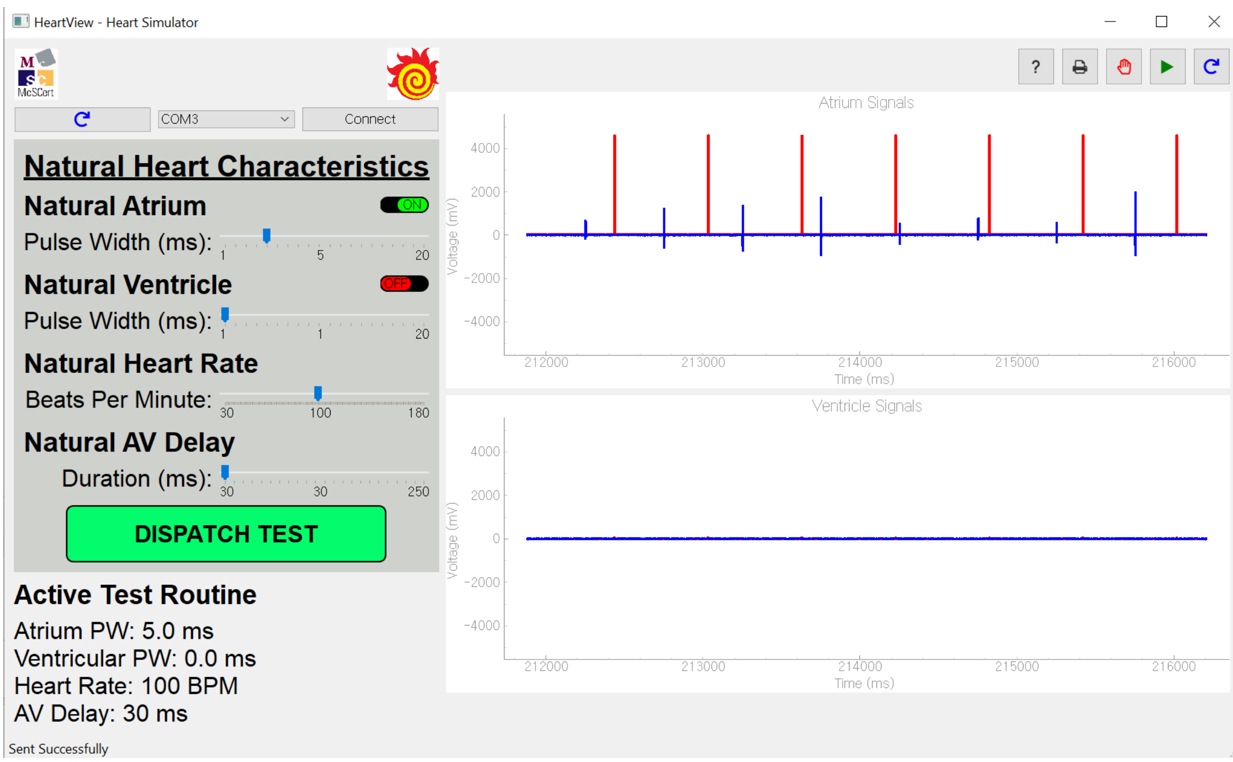


Figure 1: AOO Test with Natural Atrium ON

Pacemaker		Heartview	
Mode	Mode 0 (AOO)	Natural Atrium	ON (5ms)
Atrial Amplitude	3.3	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	100 bpm
Ventricular Amplitude	3.1	Natural AV Delay	30 ms
Ventricular Width	1		
Rate	120		
Result: Heartview result is as expected.			
Pass/Fail: Pass			

Table 4: AOO Test Overview with Natural Atrium ON

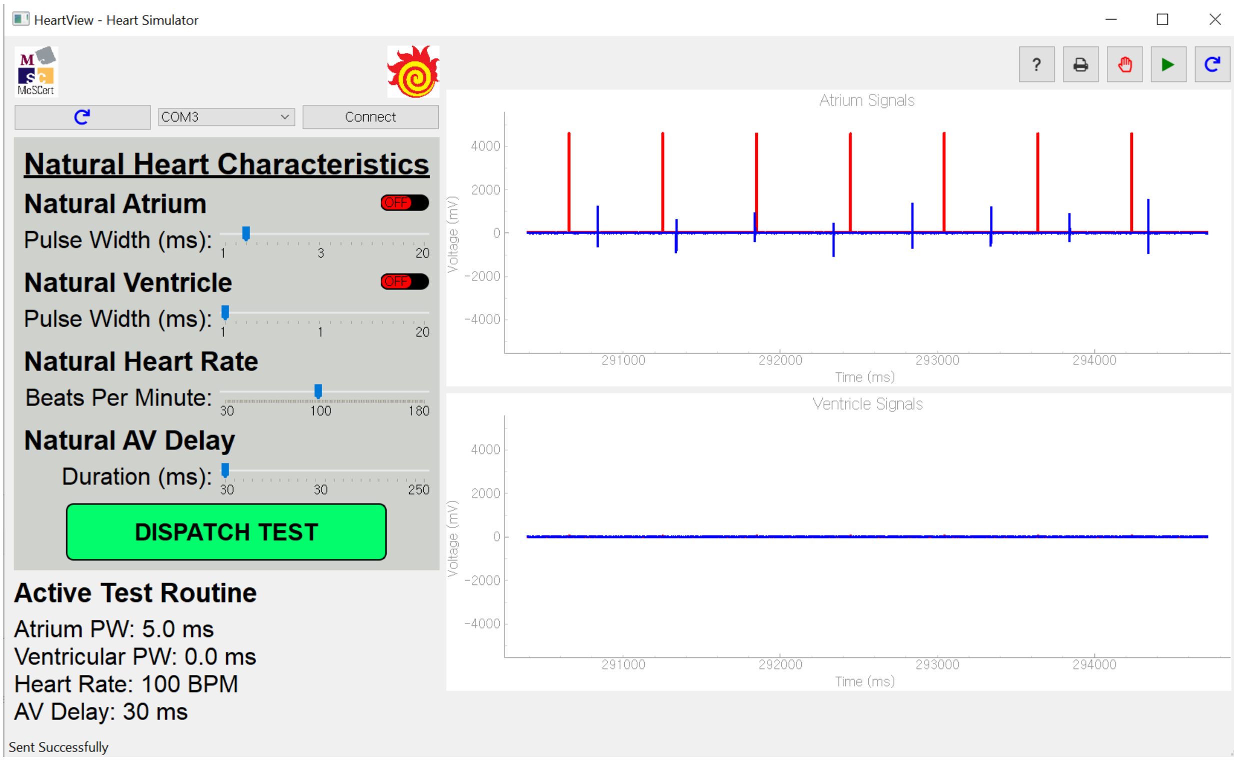


Figure 2: AOO Test with Natural Atrium and Ventricle Both OFF

Pacemaker		Heartview	
Mode	Mode 0 (AOO)	Natural Atrium	OFF
Atrial Amplitude	3.3	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	100 bpm
Ventricular Amplitude	3.1	Natural AV Delay	30 ms
Ventricular Width	1		
Rate	120		
Result: Heartview result is as expected.			
Pass/Fail: Pass			

Table 5: AOO Test Overview with Natural Atrium and Ventricle Both OFF

1.3.2.2 VOO Test

We conducted two tests to see if the VOO mode works well. First, we checked that the Pacemaker operated properly when the Natural Ventricle was ON and showed the result value. Next, we tested Pacemaker's graph for the same result as the previous test case when both the Natural Atrium and Natural Ventricle were off. In VOO mode, the result value was confirmed as follows because the same value must be obtained even if both the Natural Atrium and Ventricle are turned off. The following shows the corresponding graph and the settings for Pacemaker and Heartview.

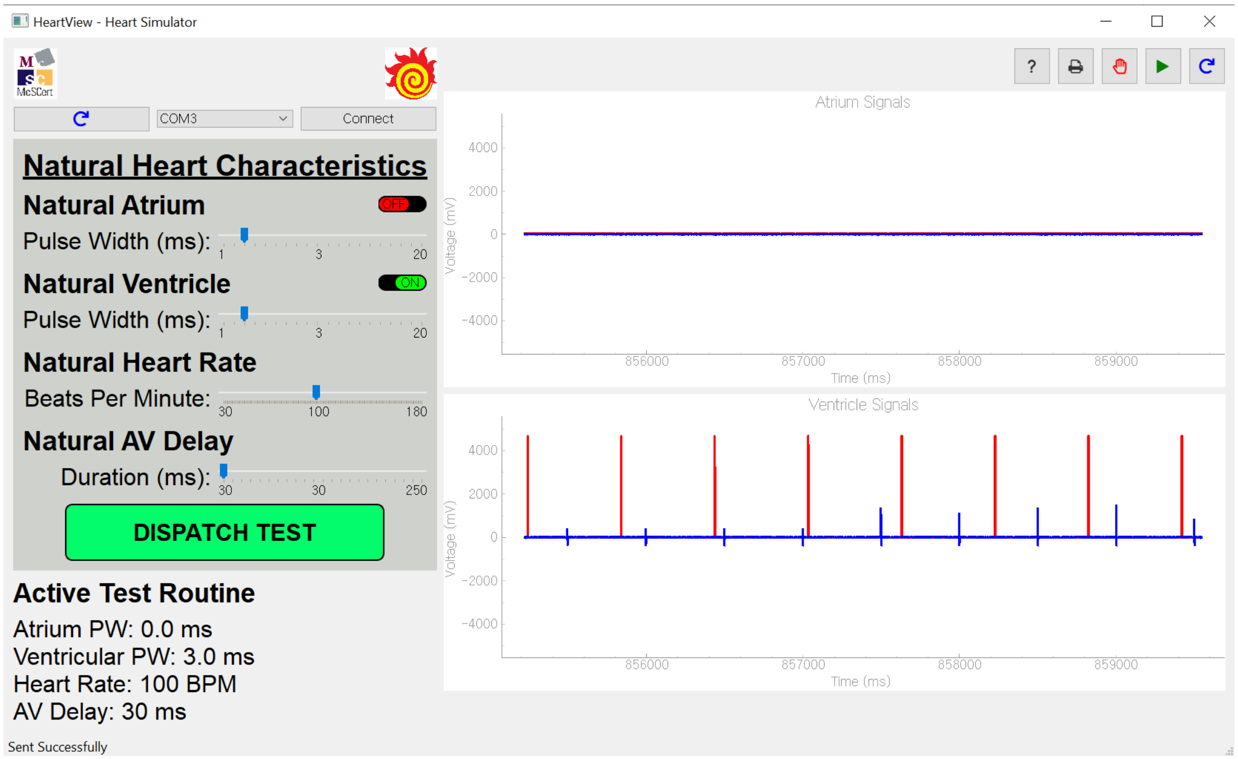


Figure 3: VOO Test with Natural Ventricle ON

Pacemaker		Heartview	
Mode	Mode 1 (VOO)	Natural Atrium	OFF
Atrial Amplitude	3.3	Natural Ventricle	ON (3ms)
Atrial Pulse Width	1	Natural Heart Rate	100 bpm
Ventricular Amplitude	3.1	Natural AV Delay	30 ms
Ventricular Width	1		
Rate	120		
Result: Heartview result is as expected.			
Pass/Fail: Pass			

Table 6: VOO Test Overview with Natural Ventricle ON

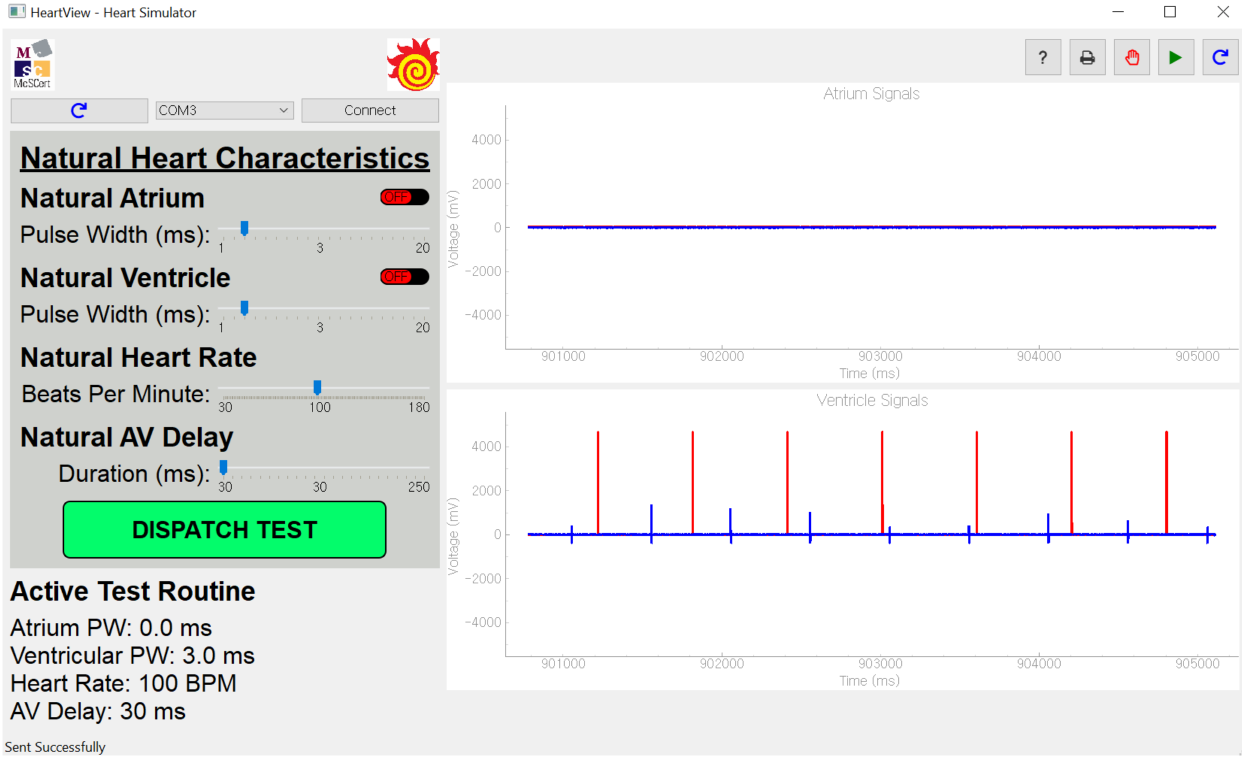


Figure 4: VOO Test with Natural Atrium and Ventricle Both OFF

Pacemaker		Heartview	
Mode	Mode 1 (VOO)	Natural Atrium	OFF
Atrial Amplitude	3.3	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	100 bpm
Ventricular Amplitude	3.1	Natural AV Delay	30 ms
Ventricular Width	1		
Rate	120		
Result: Heartview result is as expected.			
Pass/Fail: Pass			

Table 7: VOO Test Overview with Natural Atrium and Ventricle Both OFF

1.3.2.3 AAI Test

We conducted two tests to see if the AAI mode works well. First, when the Natural Atrium is ON, and the Natural Heart Rate exceeds 60 bpm, we check that the Pacemaker operates properly and shows the result value. Next, when the Natural Atrium is ON and the Natural Heart Rate is less than 60 bpm, the Pacemaker operates properly, and we check the result value. In the first test case, as a Natural Heart Rate, we used 126 bpm; in the second test case, we used 30 bpm. The following shows the corresponding graph and the settings for Pacemaker and Heartview.

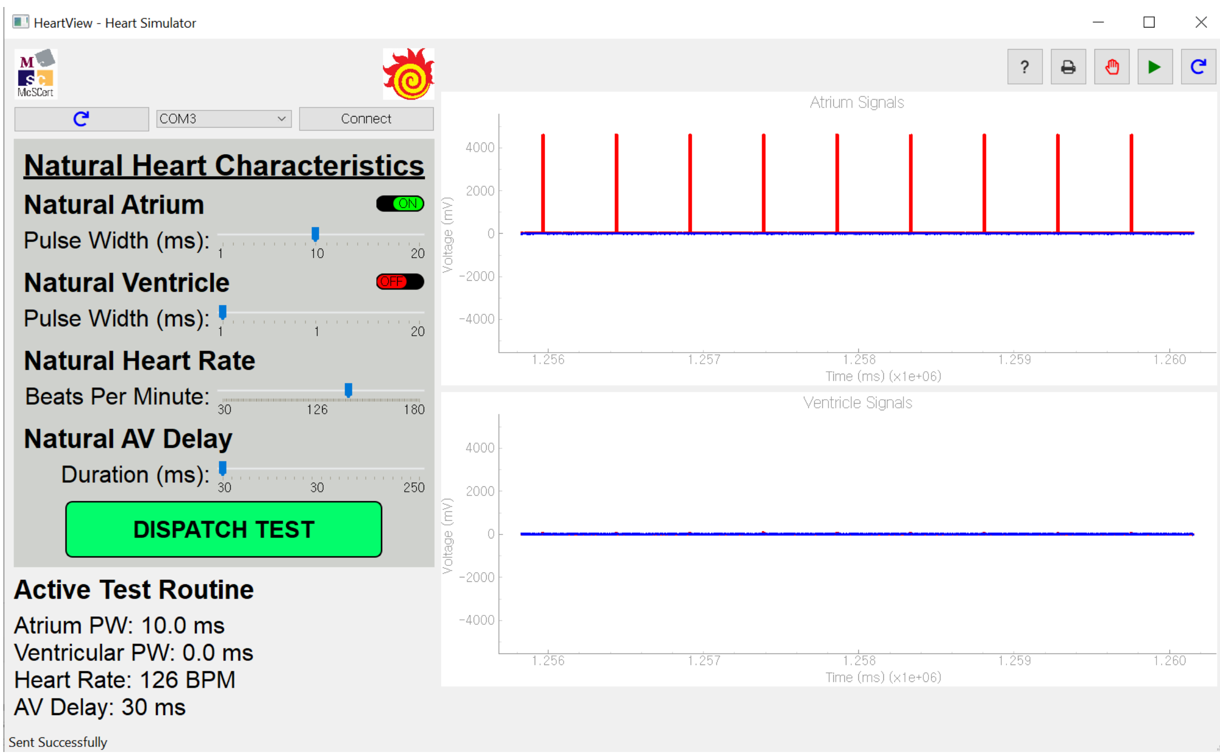


Figure 5: AAI Test with Natural Heart Rate at 126 BPM

Pacemaker		Heartview	
Mode	Mode 2 (AAI)	Natural Atrium	ON (10 ms)
Atrial Amplitude	3.3	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	126 bpm
Ventricular Amplitude	3.1	Natural AV Delay	30 ms
Ventricular Width	1		
Rate	120		
Result: Heartview result is as expected.			
Pass/Fail: Pass			

Table 8: AAI Test Overview with Natural Heart Rate at 126 BPM

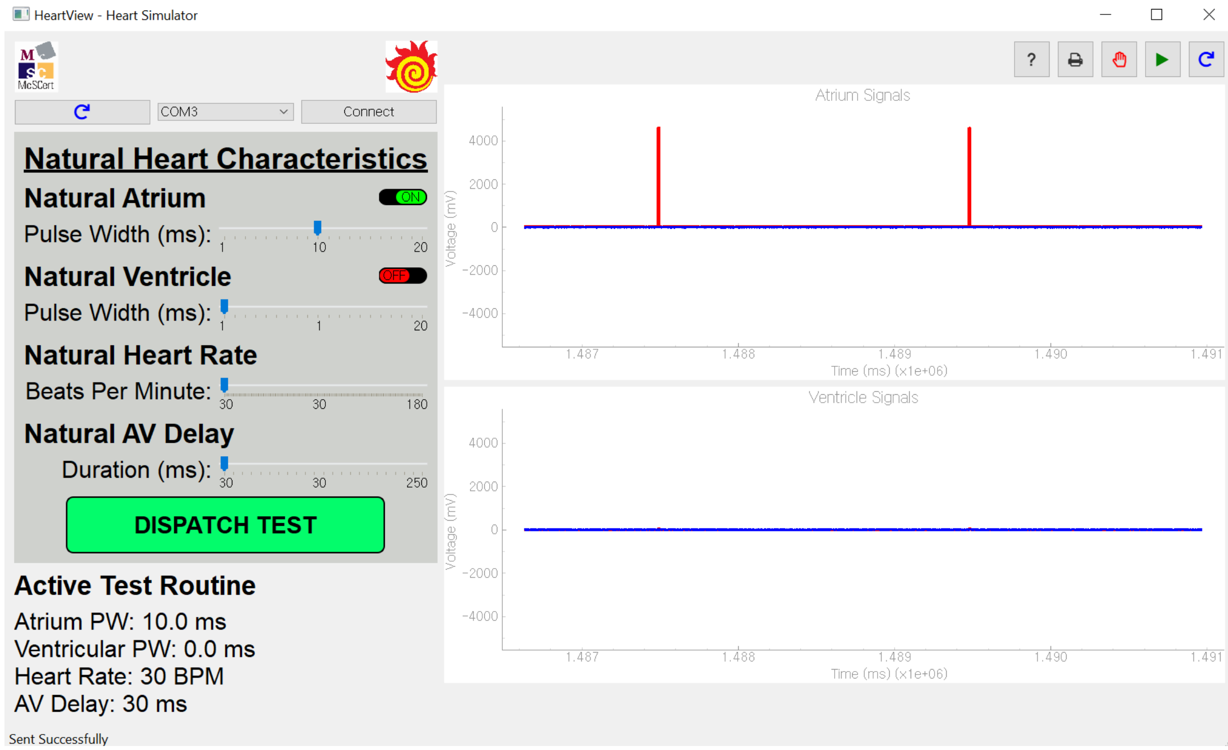


Figure 6: AAI Test with Natural Heart Rate at 30 BPM

Pacemaker		Heartview	
Mode	Mode 2 (AAI)	Natural Atrium	ON (10 ms)
Atrial Amplitude	3.3	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	30 bpm
Ventricular Amplitude	3.1	Natural AV Delay	30 ms
Ventricular Width	1		
Rate	120		
Result: Heartview result is as expected.			
Pass/Fail: Pass			

Table 9: AAI Test Overview with Natural Heart Rate at 30 BPM

1.3.2.4 VVI Test

We conducted two tests to see if the VVI mode works well. First, when the Natural Ventricle is ON, and the Natural Heart Rate exceeds 60 bpm, we check that the Pacemaker operates properly and shows the result value. Next, when the Natural Ventricle is ON and the Natural Heart Rate is less than 60 bpm, the Pacemaker operates properly, and we check the result value. In the first test case, as a Natural Heart Rate, we used 118 bpm; in the second test case, we used 30 bpm. The following shows the corresponding graph and the settings for Pacemaker and Heartview.

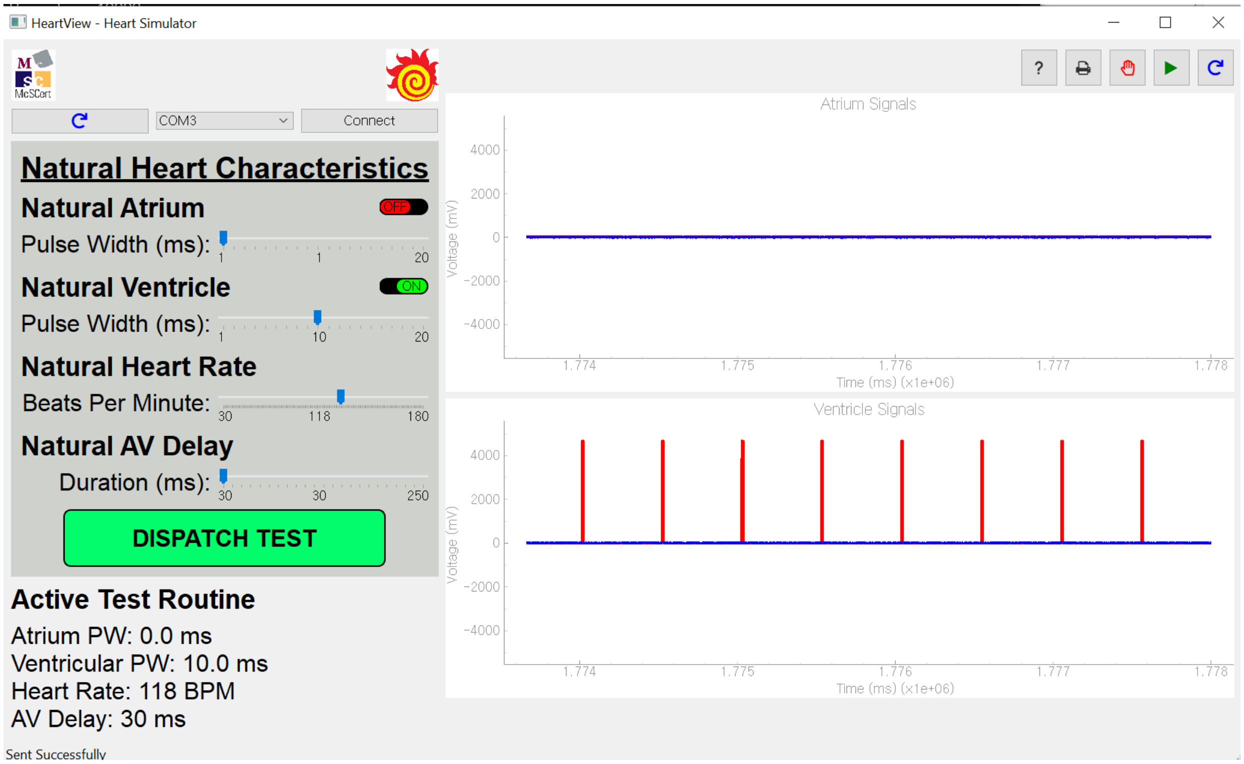


Figure 7: VVI Test with Natural Heart Rate at 118 BPM

Pacemaker		Heartview	
Mode	Mode 3 (VVI)	Natural Atrium	OFF
Atrial Amplitude	3.3	Natural Ventricle	ON (10 ms)
Atrial Pulse Width	1	Natural Heart Rate	118 bpm
Ventricular Amplitude	3.1	Natural AV Delay	30 ms
Ventricular Width	1		
Rate	120		
Result: Heartview result is as expected.			
Pass/Fail: Pass			

Table 10: VVI Test Overview with Natural Heart Rate at 118 BPM

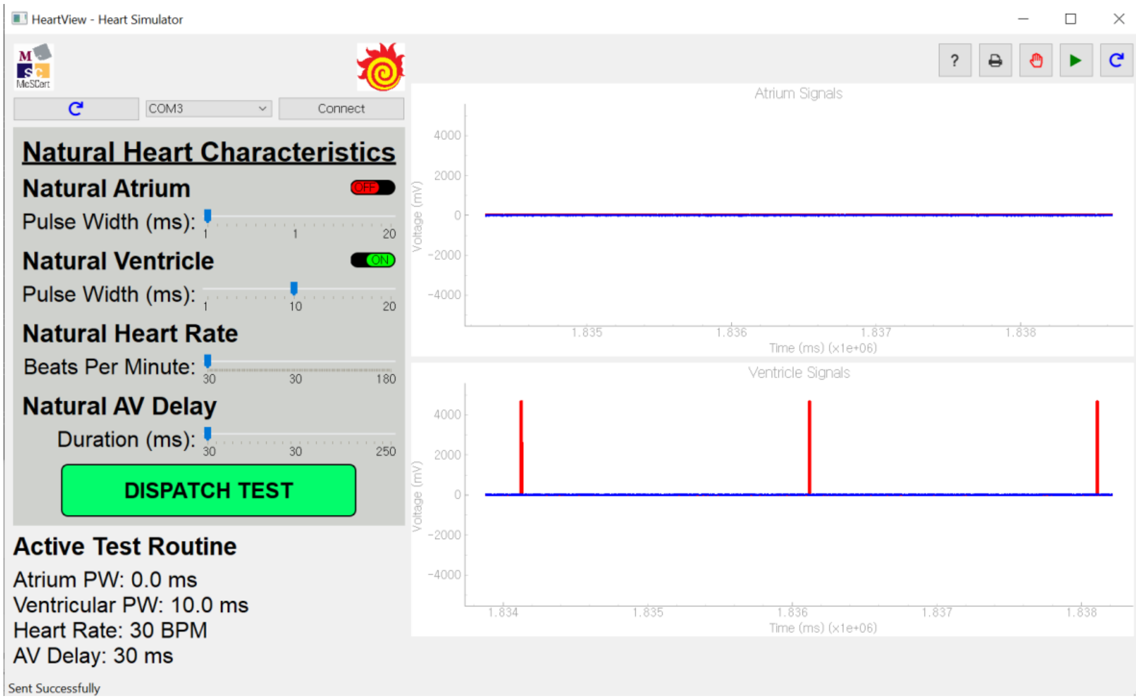


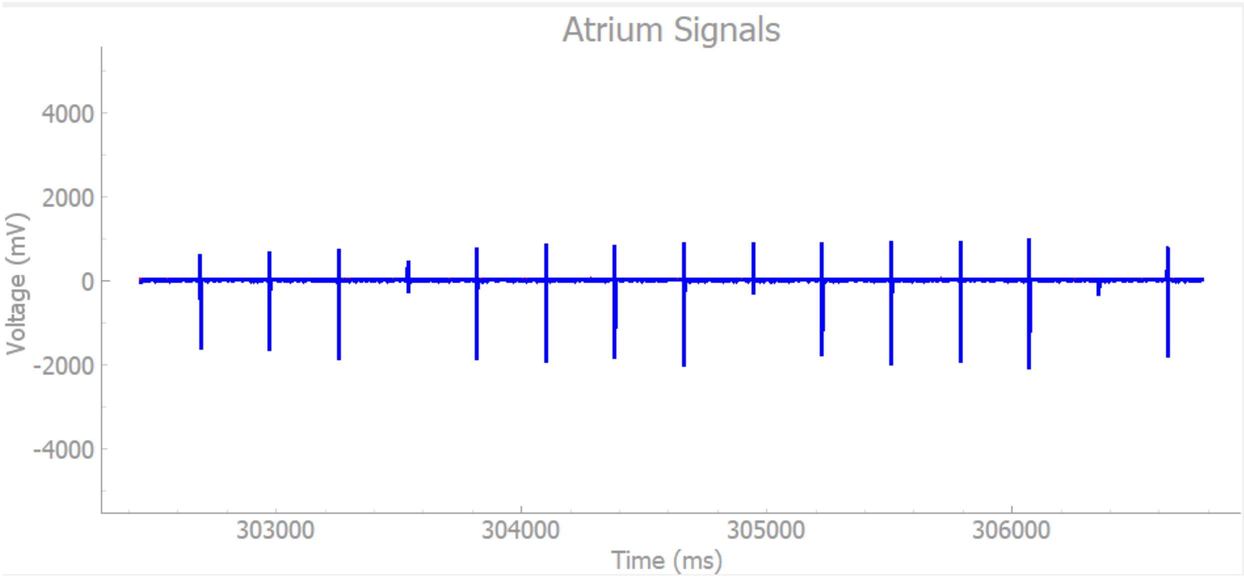
Figure 21.0 VVI Test with Natural Heart Rate at 30 bpm

Pacemaker		Heartview	
Mode	Mode 3 (VVI)	Natural Atrium	OFF
Atrial Amplitude	3.3	Natural Ventricle	ON (10 ms)
Atrial Pulse Width	1	Natural Heart Rate	30 bpm
Ventricular Amplitude	3.1	Natural AV Delay	30 ms
Ventricular Width	1		
Rate	120		
Result: Heartview result is as expected.			
Pass/Fail: Pass			

Table 11: VVI Test Overview with Natural Heart Rate at 30 BPM

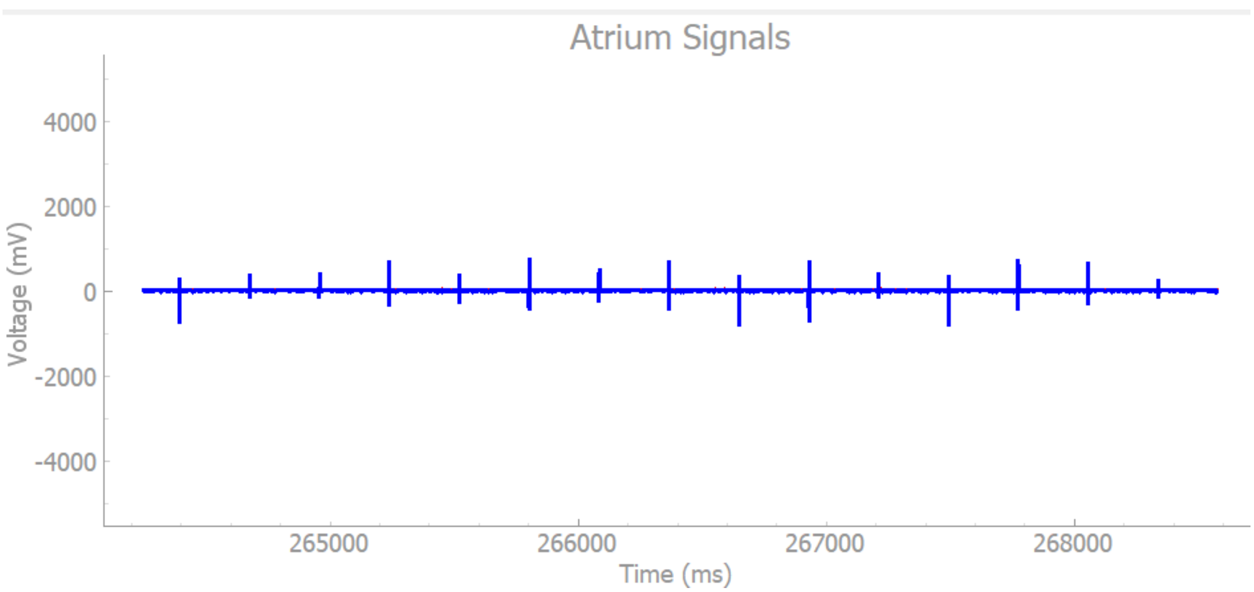
1.3.2.5 AOOR Test

We conducted two tests to see if the AOOR mode works well. First, we checked that the Pacemaker operated properly when the Natural Atrium was ON and the Natural Heart Rate was 30 bpm, and we checked the change of pacing while shaking the board. Next, we checked that the Pacemaker operated properly when the Natural Atrium was OFF, and the Natural Heart Rate was 30 bpm, and amplitude was set to 2V. The pacing rate and resulting magnitude was noted.



DCM Settings		Heartview	
Mode	AOOR	Natural Atrium	ON
Atrial Amplitude	5	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	30 bpm
URL/MSR	120	Natural AV Delay	30 ms
LRL	30		
Activity Threshold	Low/0		
Reaction Time	10		
Response Factor	16		
Recovery Time	16		
Result: Pacemaker stuck pacing at 200BPM, no apparent change with activity			
Pass/Fail: Fail			

Table 12: AOOR Test 1

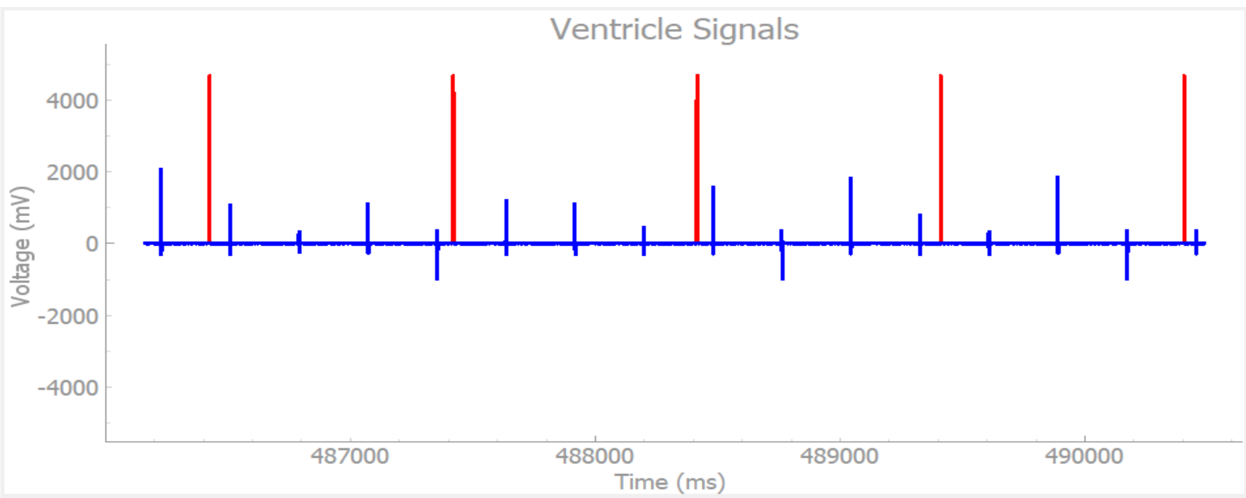


DCM Settings		Heartview	
Mode	AOOR	Natural Atrium	OFF
Atrial Amplitude	2	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	30 bpm
URL/MSR	120	Natural AV Delay	30 ms
LRL	30		
Activity Threshold	Low/0		
Reaction Time	10		
Response Factor	16		
Recovery Time	16		
Result: Amplitude lower, unexpected pacing still present			
Pass/Fail: Fail			

Table 13: AOOR Test 2

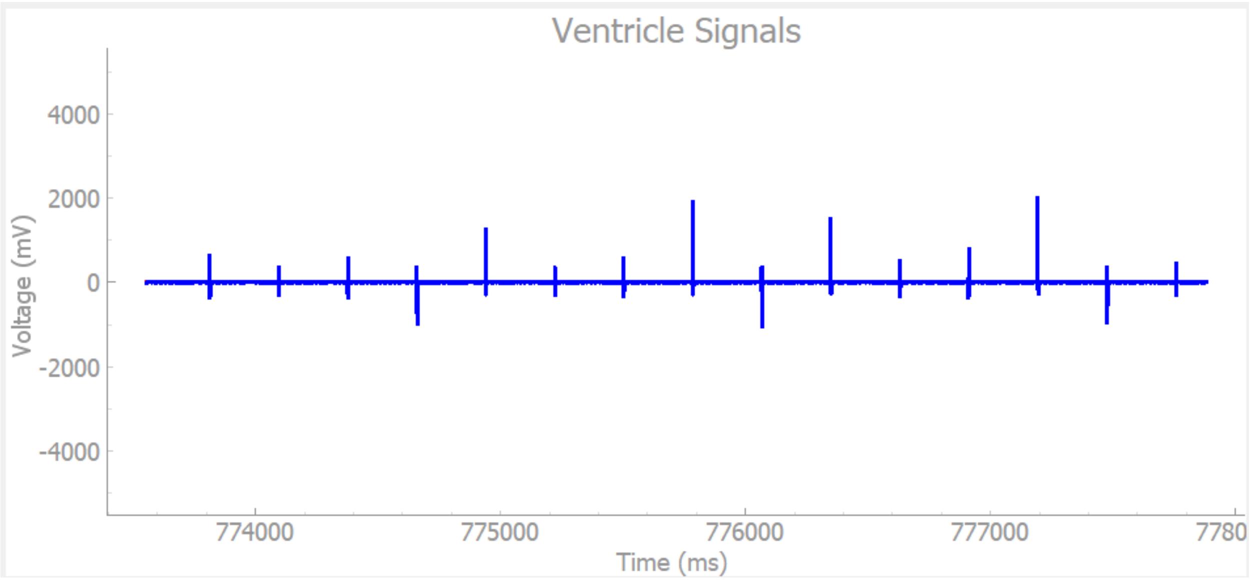
1.3.2.6 VOOR Test

We conducted two tests to see if the VOOR mode works well. First, we checked that the Pacemaker operated properly when the Natural Ventricle was ON and the Natural Heart Rate was 60 bpm, and we checked the pacing change while shaking the board. Next, we checked that the Pacemaker operated properly when the Natural Ventricle was OFF. We checked the behaviour of adaptive pacing with nominal parameter inputs. The following shows the corresponding graph and the settings for Pacemaker and Heartview.



Pacemaker		Heartview	
Mode	VOOR	Natural Atrium	OFF
URL/MSR	120	Natural Ventricle	ON
LRL	30	Natural Heart Rate	60 bpm
Ventricular Amplitude	5	Natural AV Delay	30 ms
Ventricular Pulse Width	1		
Activity Threshold	Low/0		
Reaction Time	10		
Response Factor	16		
Recovery Time	16		
Result: Abnormal pacing rate still present, no apparent adaptation.			
Pass/Fail: Fail			

Table 14: VOOR Test 1

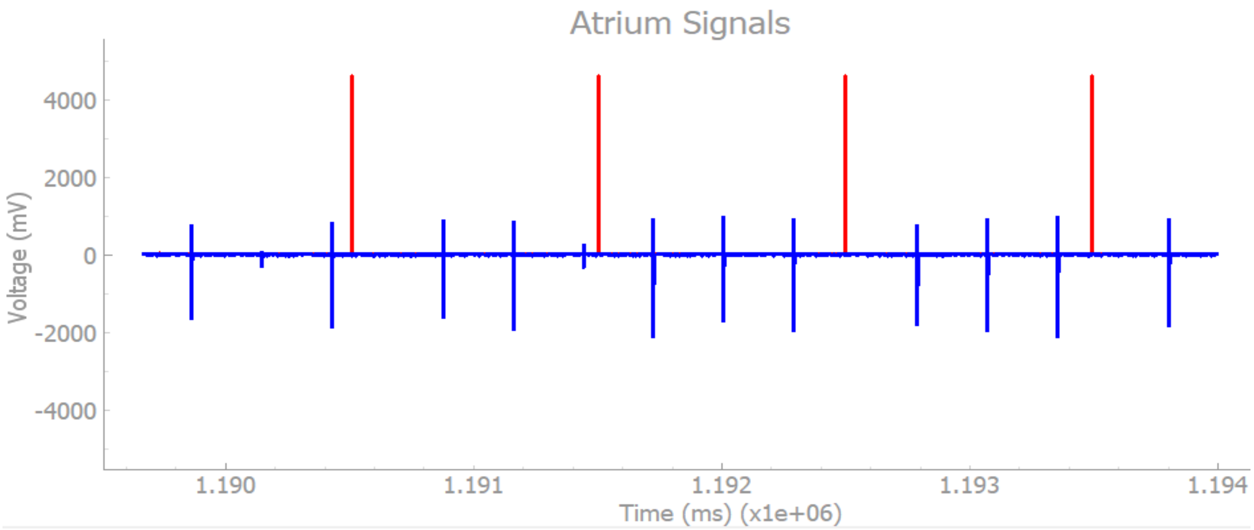


Pacemaker		Heartview	
Mode	VOOR	Natural Atrium	OFF
URL/MSR	120	Natural Ventricle	OFF
LRL	30	Natural Heart Rate	30 bpm
Ventricular Amplitude	5	Natural AV Delay	30 ms
Ventricular Width	1		
Activity Threshold	Med/3		
Reaction Time	30		
Response Factor	8		
Recovery Time	8		
Result: Abnormal pacing, no reaction			
Pass/Fail: Fail			

Table 15: VOOR Test 2

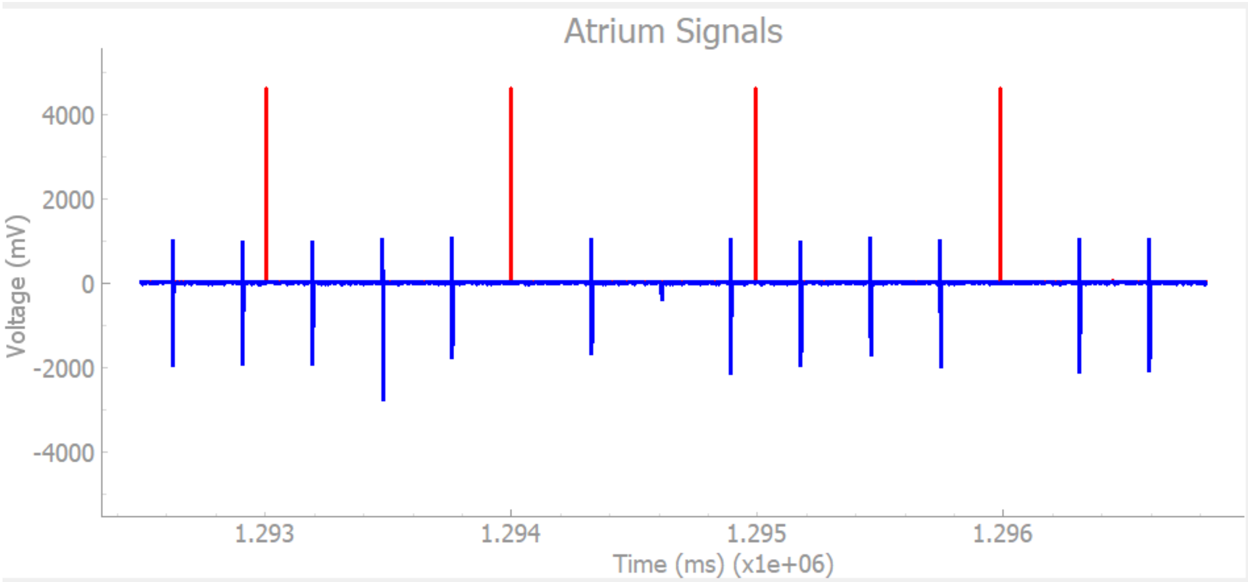
1.3.2.7 AAIR Test

We conducted two tests to see if the AAIR mode works well. The pacemaker was checked with nominal values for all fields in both, except using the extreme edges of the ARP parameter, with 150 in test one and 500 in test two.



Pacemaker		Heartview	
Mode	AAIR	Natural Atrium	ON
Atrial Amplitude	5	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	60 bpm
URL/MSR	120	Natural AV Delay	30 ms
LRL	30		
ARP	150		
Activity Threshold	Med/3		
Reaction Time	30		
Response Factor	8		
Recovery Time	8		
Result: Abnormally high rate still present, but paces inhibited as expected			
Pass/Fail: Fail			

Table 16: AAIR Test 1

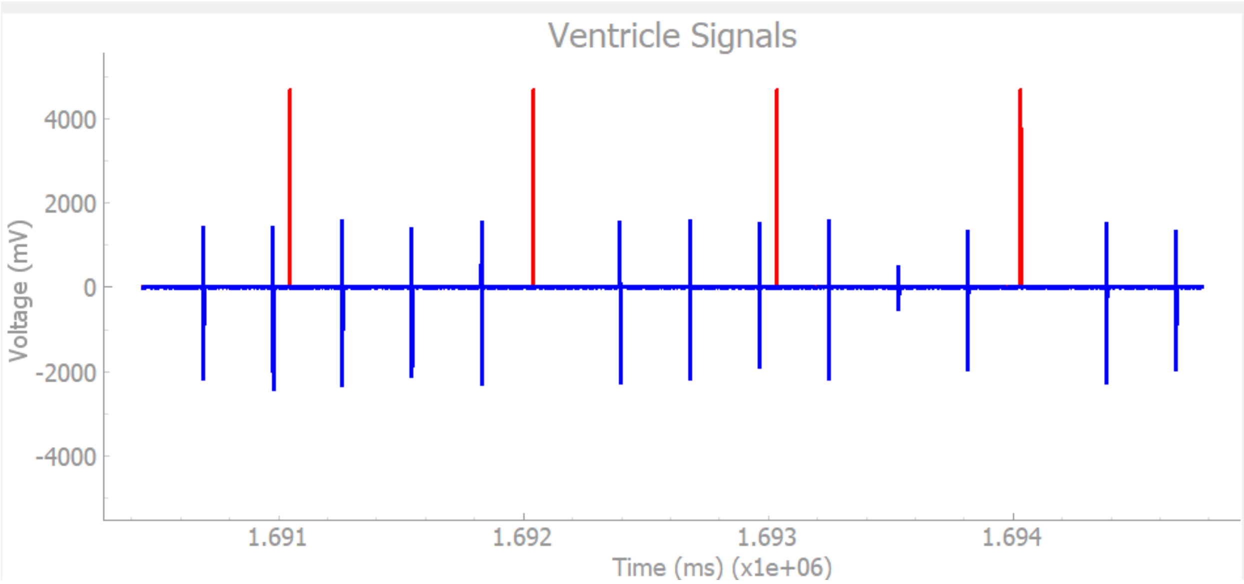


Pacemaker		Heartview	
Mode	AAIR	Natural Atrium	ON
Atrial Amplitude	5	Natural Ventricle	OFF
Atrial Pulse Width	1	Natural Heart Rate	60 bpm
URL/MSR	120	Natural AV Delay	30 ms
LRL	30		
ARP	500		
Activity Threshold	Med/3		
Reaction Time	30		
Response Factor	8		
Recovery Time	8		
Result: Less impedance of paces, rate still high			
Pass/Fail: Fail			

Table 17: AAIR Test 2

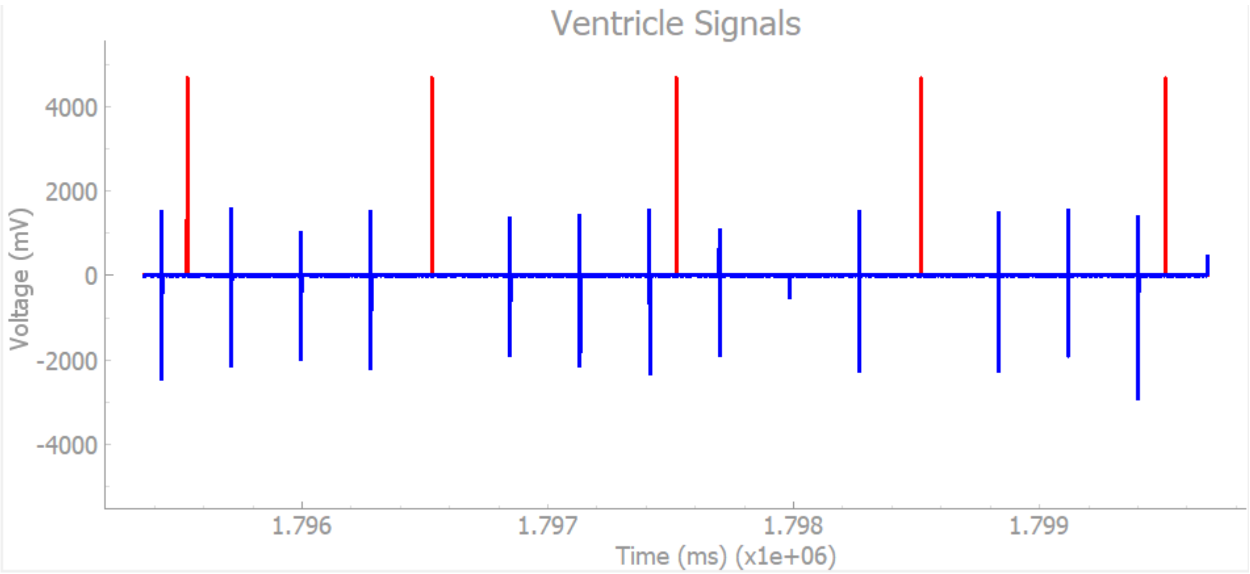
1.3.2.8 VVIR Test

Similar tests were conducted for VVIR as for AAIR, except with more changes to the rate adaptive parameters.



Pacemaker		Heartview	
Mode	VVIR	Natural Atrium	OFF
URL/MSR	175	Natural Ventricle	ON
LRL	30	Natural Heart Rate	60 bpm
Ventricular Amplitude	5	Natural AV Delay	30 ms
Ventricular Width	1		
VRP	150		
Activity Threshold	VHigh/6		
Reaction Time	20		
Response Factor	16		
Recovery Time	2		
Result: Inhibition present, pacing rate not adapting or even at any number written anywhere in the code			
Pass/Fail: Fail			

Table 18: VVIR Test 1



Pacemaker		Heartview	
Mode	VVIR	Natural Atrium	OFF
URL/MSR	175	Natural Ventricle	ON
LRL	30	Natural Heart Rate	60 bpm
Ventricular Amplitude	5	Natural AV Delay	30 ms
Ventricular Width	1		
VRP	500		
Activity Threshold	Low/1		
Reaction Time	40		
Response Factor	15		
Recovery Time	5		
Result: No change from last test outside of lessened inhibition			
Pass/Fail: Fail			

Table 19: VVIR Test 2

Part 2: Future Flexibility and Modules

2.1 Requirements likely to change

In the future, dual pacing modes could be added. Subsystems are used to encapsulate the different operating modes to make it easier to add new ones. Further features similar to the adaptive rate pacing would be implemented in further subsystems.

2.2 Design decisions likely to change

The exact algorithm to calculate adaptive rate pacing could change if we were interested in furthering this project.

2.3 MIS and MID

2.3.1 Module Interface Specification

Describes the purpose, public functions, and black-box behaviour of each function.

2.3.1.1 Input Pins Subsystem

Maps MCU input pins to more descriptive internal variables to enable hardware hiding. Includes outputs pins relevant to the two inputs.

Input	Type	Output	Type
PTC16 (D0)	bool	ATR_CMP_DETECT	bool
PTC17 (D1)	bool	VENT_CMP_DETECT	bool
ATR_CMP_REF_PWM	int	PTA1 (D3)	PWM duty cycle
VENT_CMP_REF_PWM	int	PTC2 (D6)	PWM duty cycle

Table 20: MCU input pins

2.3.1.2 Output Pins Subsystem

Maps internal variables to MCU output pins to enable hardware hiding.

Input	Type	Output	Type
PACE_CHARGE_CTRL	bool	PTB9 (D2)	bool
Z_ATR_CTRL	bool	PTB23 (D4)	bool
PACING_REF_PWM	single	PTA2 (D5)	single
Z_VENT_CTRL	bool	PTC3 (D7)	bool
ATR_PACE_CTRL	bool	PTA0 (D6)	bool
VENT_PACE_CTRL	bool	PTC4 (D9)	bool
PACE_GND_CTRL	bool	PTD0 (D10)	bool
ATR_GND_CTRL	bool	PTD2 (D11)	bool
VENT_GND_CTRL	bool	PTD3 (D12)	bool

Table 21: MCU output pins

2.3.1.3 Main Stateflow

Allows the program to branch to different operating modes. Organizes sub-stateflows and is extensible for future modes.

Input	Type	Output	Type
ATR_CMP_DETECT	bool	PACE_CHARGE_CTRL	bool
VENT_CMP_DETECT	bool	Z_ATR_CTRL	bool
Mode	int8	PACING_REF_PWM	single

LRL	single	Z_VENT_CTRL	bool
Atrial_Amplitude	single	ATR_PACE_CTRL	bool
Atrial_Pulse_Width	single	VENT_PACE_CTRL	bool
Ventricular_Amplitude	single	PACE_GND_CTRL	bool
Ventricular_Pulse_Width	single	ATR_GND_CTRL	bool
ARP	single	VENT_GND_CTRL	bool
VRP	single	SERIAL_OUT()	function call
adaptiveRate	single		

Table 22: Programmable Parameters in Main Stateflow

2.3.1.4 AOO Stateflow

Implements the AOO operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create paces.

Input	Type	Output	Type
Atrial_Amplitude	single	PACE_CHARGE_CTRL	bool
Atrial_Pulse_Width	single	Z_ATR_CTRL	bool
Rate_ms	single	PACING_REF_PWM	single
		Z_VENT_CTRL	bool
		ATR_PACE_CTRL	bool
		VENT_PACE_CTRL	bool
		PACE_GND_CTRL	bool
		ATR_GND_CTRL	bool
		VENT_GND_CTRL	bool
		SERIAL_OUT()	function call

Table 23: Programmable Parameters in AOO Stateflow

2.3.1.5 VOO Stateflow

Implements the VOO operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create paces.

Input	Type	Output	Type
Ventricular_Amplitude	single	PACE_CHARGE_CTRL	bool
Ventricular_Pulse_Width	single	Z_ATR_CTRL	bool
Rate_ms	single	PACING_REF_PWM	single
		Z_VENT_CTRL	bool
		ATR_PACE_CTRL	bool
		VENT_PACE_CTRL	bool
		PACE_GND_CTRL	bool
		ATR_GND_CTRL	bool
		VENT_GND_CTRL	bool
		SERIAL_OUT()	function call

Table 24: Programmable Parameters in VOO Stateflow

2.3.1.6 AAI Stateflow

Implements the AAI operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create and respond to paces.

Input	Type	Output	Type
Atrial_Amplitude	single	PACE_CHARGE_CTRL	bool
Atrial_Pulse_Width	single	Z_ATR_CTRL	bool
Rate_ms	single	PACING_REF_PWM	single
ATR_CMP_DETECT	bool	Z_VENT_CTRL	bool
ARP	single	ATR_PACE_CTRL	bool
		VENT_PACE_CTRL	bool
		PACE_GND_CTRL	bool
		ATR_GND_CTRL	bool
		VENT_GND_CTRL	bool
		SERIAL_OUT()	function call

Table 25: Programmable Parameters in AAI Stateflow

2.3.1.7 VVI Stateflow

Implements the VVI operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create and respond to paces.

Input	Type	Output	Type
Ventricular_Amplitude	single	PACE_CHARGE_CTRL	bool
Ventricular_Pulse_Width	single	Z_ATR_CTRL	bool
Rate_ms	single	PACING_REF_PWM	single
VENT_CMP_DETECT	bool	Z_VENT_CTRL	bool
VRP	single	ATR_PACE_CTRL	bool
		VENT_PACE_CTRL	bool
		PACE_GND_CTRL	bool
		ATR_GND_CTRL	bool
		VENT_GND_CTRL	bool
		SERIAL_OUT()	function call

Table 26: Programmable Parameters in VVI Stateflow

2.3.1.8 AOOR Stateflow

Implements the AOOR operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create paces.

Input	Type	Output	Type
Atrial_Amplitude	single	PACE_CHARGE_CTRL	bool
Atrial_Pulse_Width	single	Z_ATR_CTRL	bool
Rate_ms	single	PACING_REF_PWM	single
adaptiveRate	single	Z_VENT_CTRL	bool
		ATR_PACE_CTRL	bool
		VENT_PACE_CTRL	bool
		PACE_GND_CTRL	bool
		ATR_GND_CTRL	bool
		VENT_GND_CTRL	bool
		SERIAL_OUT()	function call

Table 23: Programmable Parameters in AOOR Stateflow

2.3.1.9 VOOR Stateflow

Implements the VOOR operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create paces.

Input	Type	Output	Type
Ventricular_Amplitude	single	PACE_CHARGE_CTRL	bool
Ventricular_Pulse_Width	single	Z_ATR_CTRL	bool
Rate_ms	single	PACING_REF_PWM	single
adaptiveRate	single	Z_VENT_CTRL	bool
		ATR_PACE_CTRL	bool
		VENT_PACE_CTRL	bool
		PACE_GND_CTRL	bool
		ATR_GND_CTRL	bool
		VENT_GND_CTRL	bool
		SERIAL_OUT()	function call

Table 24: Programmable Parameters in VOOR Stateflow

2.3.1.10 AAIR Stateflow

Implements the AAIR operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create and respond to paces.

Input	Type	Output	Type
Atrial_Amplitude	single	PACE_CHARGE_CTRL	bool
Atrial_Pulse_Width	single	Z_ATR_CTRL	bool
Rate_ms	single	PACING_REF_PWM	single
ATR_CMP_DETECT	bool	Z_VENT_CTRL	bool
ARP	single	ATR_PACE_CTRL	bool
adaptiveRate	single	VENT_PACE_CTRL	bool
		PACE_GND_CTRL	bool
		ATR_GND_CTRL	bool
		VENT_GND_CTRL	bool
		SERIAL_OUT()	function call

Table 25: Programmable Parameters in AAIR Stateflow

2.3.1.11 VVIR Stateflow

Implements the VVIR operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create and respond to paces.

Input	Type	Output	Type
Ventricular_Amplitude	single	PACE_CHARGE_CTRL	bool
Ventricular_Pulse_Width	single	Z_ATR_CTRL	bool
Rate_ms	single	PACING_REF_PWM	single
VENT_CMP_DETECT	bool	Z_VENT_CTRL	bool
VRP	single	ATR_PACE_CTRL	bool
adaptiveRate	single	VENT_PACE_CTRL	bool
		PACE_GND_CTRL	bool
		ATR_GND_CTRL	bool
		VENT_GND_CTRL	bool
		SERIAL_OUT()	function call

Table 26: Programmable Parameters in VVIR Stateflow

2.3.1.12 SERIAL_OUT()

Exists alongside SERIAL_IN() within the Serial Communications subsystem. Transmits information over the serial bus, primarily pacing details.

Input	Type	Output	Type
SERIAL_OUT()	function call	Serial transmission	5 byte uint8 array
A nat	bool		
A pace	bool		
V nat	bool		
V pace	bool		
print	N/A		

2.3.1.13 SERIAL_IN()

Exists alongside SERIAL_OUT() within the Serial Communications subsystem. Receives information from the serial bus.

Input	Type	Output	Type
SERIAL_CHECK()	function call	LED_blink	bool
rx_data	16 byte uint8 array	mode	uint8
status	uint8	LRL	uint8
		URL/MSR	uint8
		Atrial_Amplitude	double
		Ventricular_Amplitude	double
		Atrial_PW	uint8
		Vent_PW	uint8
		VRP	uint32
		ARP	uint32
		activity	uint8
		reaction	uint8
		response	uint8
		recovery	uint8

2.3.1.14 Adaptive Rate Control

Implements adaptation of the heart rate based on activity and other listed parameters.

Input	Type	Output	Type
CALC_RATE()	function call	adaptiveRate	single
LRL	uint8		
URL/MSR	uint8		
Activity Threshold	uint8		
Reaction Time	uint8		
Response Factor	uint8		
Recovery Time	uint8		
FXOS8700 6-Axes Sensor	3 byte double vector		

2.3.1.15 Timing block
Triggers the SERIAL_IN() and Adaptive Rate Pacing subsystems.

Input	Type	Output	Type
		calcRate	Function call
		SERIAL_IN()	Function call

2.3.2 Module Internal Design
Describes the state variables, private functions, and internal behaviour of all functions.

2.3.2.1 Input Pins Subsystem
Assigns the input from pin D0 to the boolean variable ATR_CMP_DETECT, and the input from pin D1 to the boolean variable VENT_CMP_DETECT. Assigns a duty cycle of 70 to both ATR and VENT _CMP_REF_PWM pins.

2.3.2.2 Output Pins Subsystem
Assigns the boolean variables PACE_CHARGE_CTRL, Z_ATR_CTRL, Z_VENT_CTRL, ATR_PACE_CTRL, VENT_PACE_CTRL, PACE_GND_CTRL, ATR_GND_CTRL, and VENT_GND_CTRL to pins D2, D4, D7, D8, D9, D10, D11, and D12 respectively as digital outputs. Assigns PACING_REF_PWM to pin D5 as a digital PWM output.

2.3.2.3 Main Stateflow
Creates the state variable Rate_ms. This is a single that holds the conversion of the input variable Rate from paces per minute to milliseconds. Then, based on the value of the int8 Mode, it branches into either AOO, VOO, AAI, VVI, AOOR, AAIR, VOOR, or VVIR if the value is 0, 1, 2, 3, 4, 5, 6, or 7 respectively. If the value is 255 then no mode is chosen, and the pacemaker will not operate.

2.3.2.4 AOO Stateflow
Defines the variable of type single atrialDutyCycle, converting Atrial_Amplitude into a PWM signal referenced from 5V. Also defines atrWait using Rate_ms and Atrial_Pulse_Width to designate how long to wait in between pulses. Then enters two looping states, charge_discharge and pace. The former assigns all the necessary output variables to discharge C21 and charge C22 (as outlined in Pacemaker Shield Explained) for pacing the atrium. After spending time atrWait in this state, it will then move to the state pace, where C22 will be discharged into C21 through the atrium using the output values as defined in Pacemaker Shield Explained. After Atrial_Pulse_Width milliseconds have passed, it will return to charge_discharge. Calls SERIAL_OUT() when delivering a pace.

2.3.2.5 VOO Stateflow
This acts identically to AOO, except all references to the atrium are replaced with the ventricle, and vice versa. Local state variables are ventDutyCycle and ventWait.

2.3.2.6 AAI Stateflow
Defines the variable of type single atrialDutyCycle, converting Atrial_Amplitude into a PWM signal referenced from 5V. Also defines atrWait using Rate_ms and Atrial_Pulse_Width to designate how long to wait in between pulses. Assigns the state variable timeToPace to be equal to atrWait upon entry, then moves to state charge_discharge. This state performs the same assignments as in AOO, but then immediately moves to state sense_wait. This state decrements timeToPace by 1 millisecond every millisecond whenever timeToPace is positive and ATR_CMP_DETECT is 0. If ATR_CMP_DETECT is

1, then state refractory_period is entered, which increases timeToPace by ARP milliseconds. Sense_wait is then re-entered. If in sense_wait timeToPace ever falls below or equal to 0, then it will enter the state pace. This state performs the same as in AOO, except it also assigns timeToPace to be equal to atrWait. Pace is exited after Atrial_Pulse_Width milliseconds and moves back to charge_discharge. Calls SERIAL_OUT() when delivering a pace or sensing a natural pace.

2.3.2.7 VVI Stateflow
This acts identically to AAI, except all references to the atrium are replaced with the ventricle, and vice versa. Local state variables are ventDutyCycle, ventWait, and timeToPace. Note that ARP is replaced with VRP.

2.3.2.8 AOOR Stateflow
Operates identically to the AOO stateflow except that the pacing rate is updated to reflect adaptiveRate each pacing cycle.

2.3.2.9 VOOR Stateflow
Operates identically to the VOO stateflow, except that the pacing rate is updated to reflect adaptiveRate every pacing cycle.

2.3.2.10 AAIR Stateflow
Operates identically to the AAI stateflow, except that the pacing rate is updated to reflect adaptiveRate every pacing cycle.

2.3.2.11 VVIR Stateflow
Operates identically to the VVI stateflow, except that the pacing rate is updated to reflect adaptiveRate every pacing cycle.

2.3.2.12 SERIAL_OUT()
Casts all input variables into uint8, muxes them into an array for transmission over serial.

2.3.2.13 SERIAL_IN()
A stateflow is used to set the default mode value and blink the LED. On entry, mode is set to 255. Then state standby is entered, setting LED_blink to 0. If status becomes zero, state success is entered. LED_blink is set to 1, and rx_data is stripped and rescaled into the parameters used internally, as defined above. After 300 milliseconds state standby is re-entered.

2.3.2.14 Adaptive Rate Control
A default value of 60 for LRL is established for safety reasons. State variable Current Rate is defined as Adaptive Rate delayed by a unit delay (1/z). First the input signal from the accelerometer is conditioned. The double array is split into its three components, converted from g into m/s^2, squared, summed, then square rooted. 9.81 is subtracted to remove acceleration due to gravity, then a moving average is taken and assigned to state variable Accel Average. The magnitude of z is also split out, with 9.81 subtracted from it, and a separate moving average taken. The magnitude of this average is then compared to a threshold (0.1) to give the boolean state variable Zthres. Next the reate target is calculated. If Accel Average > activity and Zthres = True, the target is set to the LRL plus the difference between the activity and the threshold times the response factor. Checks are in place to keep this target within LRL and URL/MSR. If the threshold is not reached the rate is kept at LRL. The rate change is then found, with a maximum and minimum allowable slope based on the reaction and response time. The next adaptive rate is incremented by the slope selected, scaled by the gap.

2.3.2.15 Timing block
Contains two states, alternating between them every 5ms. State rate calls calcRate(), and state serial calls SERIAL_IN().

Part 3: Assurance Case
Designing a pacemaker using Simulink requires a variety of safety and functional aspects that ensure that the device operates reliably and securely. The assurance case for this design is that the pacemaker typically meets all the required safety and performance requirements. The assurance case may include the following lists.

3.1 Description

3.1.1 Functional Safety Requirements
The pacemaker is a built-in system designed to control the heart rate to allow mode adjustments to manage the patient's heart rate. It consists of a detection module, a control unit, and a heart rate control module, generates a stable pacing pulse, and detects the correct speed.

3.1.2 Risk Analysis and Mitigation
The device must constantly detect abnormal heartbeat and provide a heartbeat stimulus within a safe voltage limit. It can use duplicate detection algorithms for inaccurate detection.

3.1.3 Reliability and Performance
The pacemaker operates reliably under a variety of operating conditions. It can set AOO, VOO, AAI, VVI, AOOR, VOOR, AAIR, and VVIR modes to operate appropriately for those modes.

3.1.4 Compliance with medical standards
To meet the standards for identifying relevant medical device standards, we designed the design based on the PACEMAKER document provided by McMaster. In addition, it is necessary to check whether it conforms to the standards for implantable pacemakers.

3.1.5 Software Validation
To ensure that the software works as intended, the verification process was verified through unit testing, integration testing, and system testing. The simulation was run by setting various heart conditions.

3.1.6 User Safety Features
Safety devices such as alarms should be designed and implemented to protect patients in the future. User safety modes such as automatic speed control and emergency mode are required.

3.1.7 Battery life management
To ensure patient safety, it seems necessary to design a safety device that optimizes battery use and analyzes life. When it comes to battery failure, additional designs should be designed to monitor battery life.

3.1.8 Monitoring after deployment
A plan will be needed for performance monitoring and software updates after the pacemaker is deployed.

3.1.9 Patient Data Security

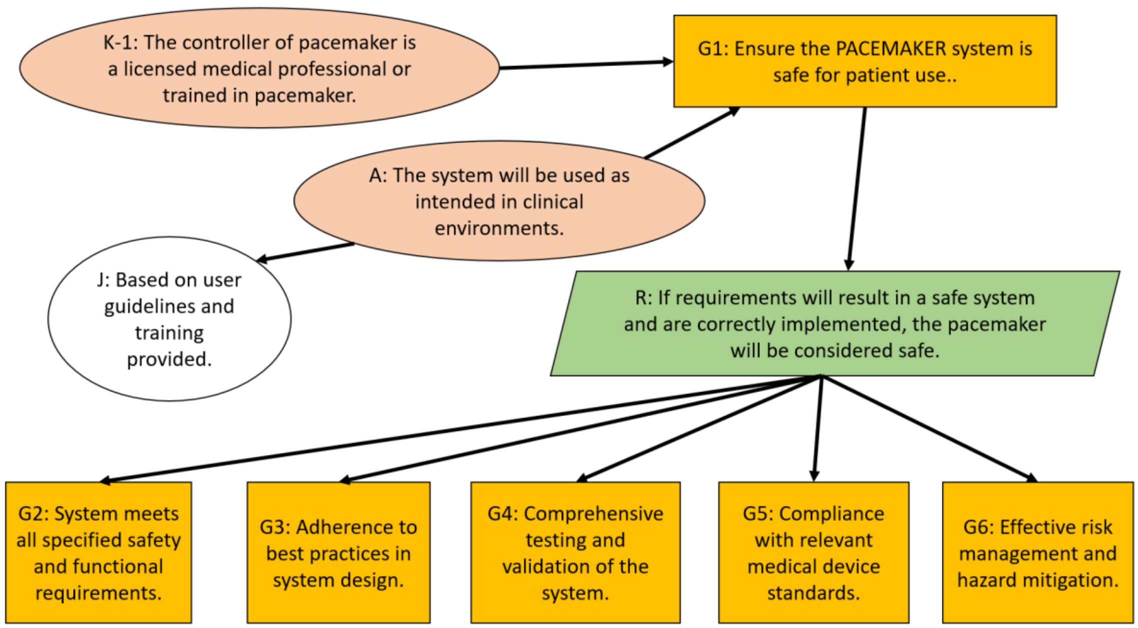
A plan seems to be needed to ensure the privacy and data security of patients handled by the pacemaker.

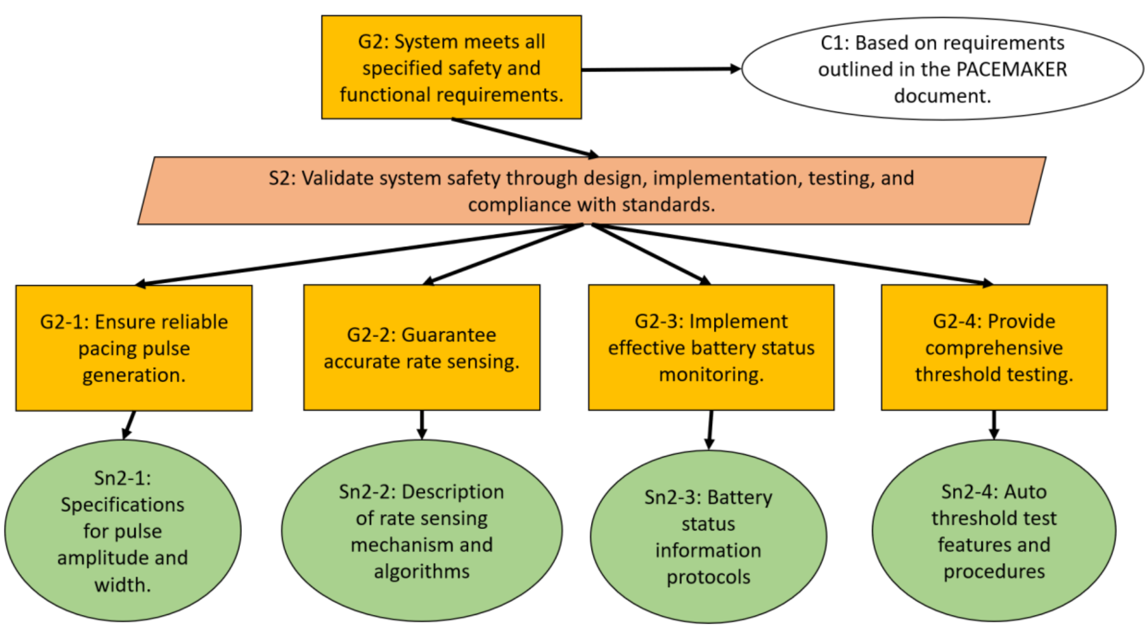
3.1.10 Ergonomics

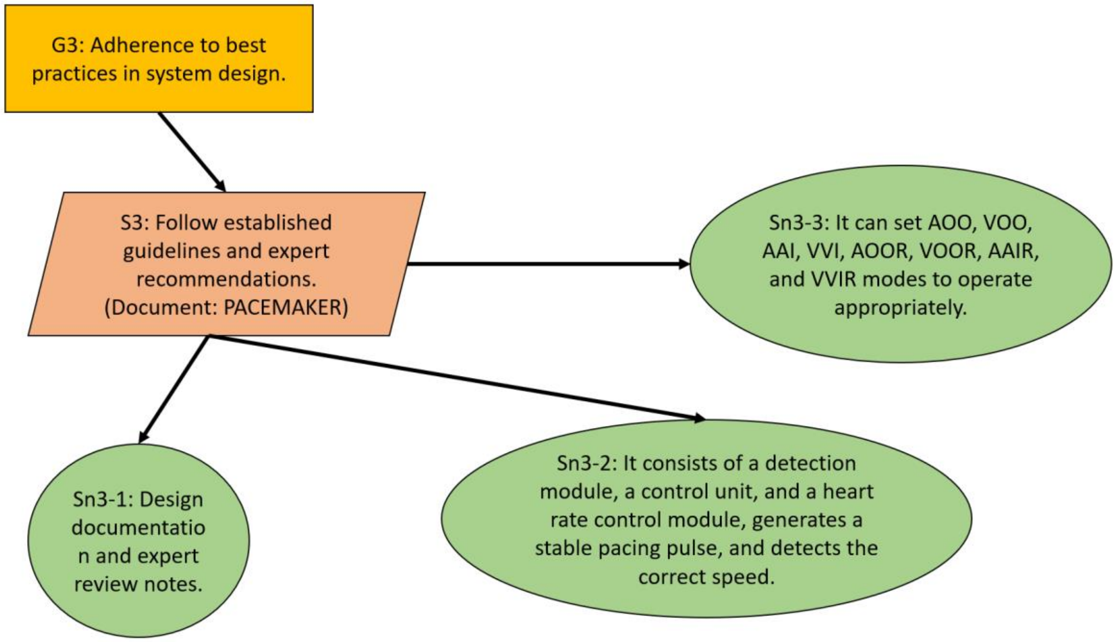
It seems necessary to consider user interfaces or interactions with external components.

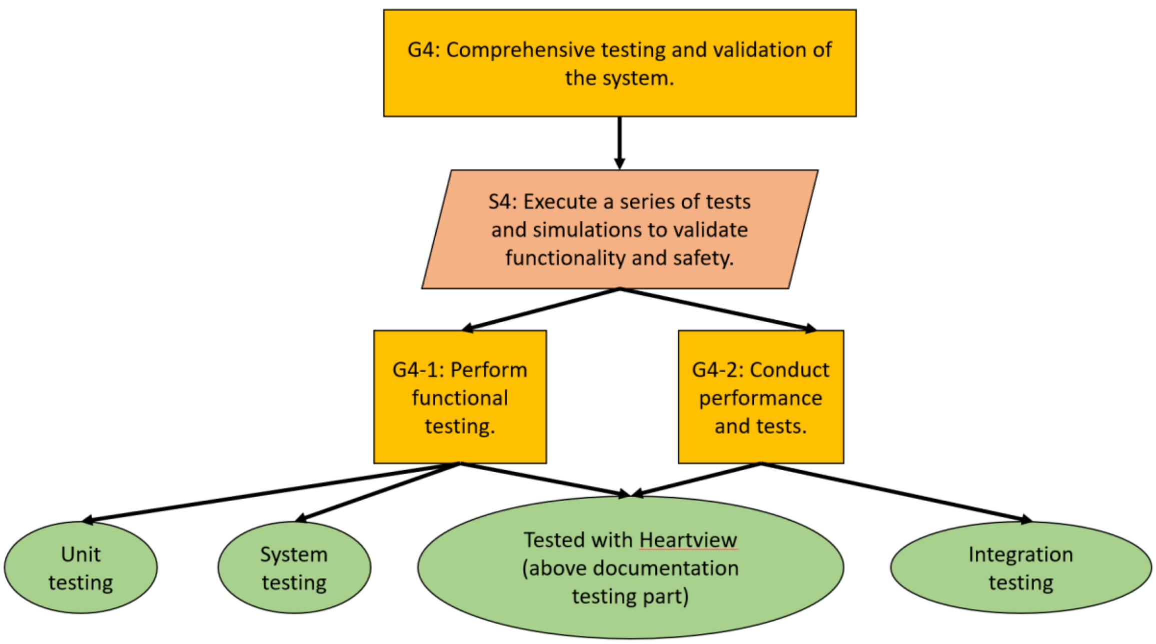
3.2 Assurance Case: GSN Format

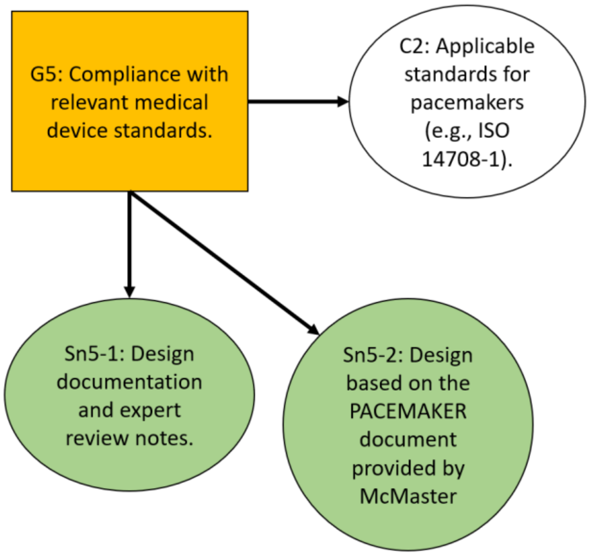
Following assurance case is using the Goal-Structured Notation (GSN) format.

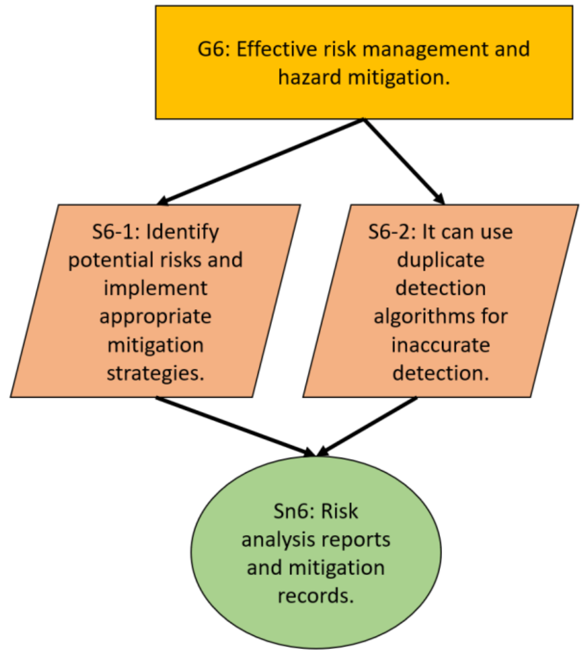





G3: Adherence to best practices in system design.
S3: Follow established guidelines and expert recommendations. (Document: PACEMAKER)
Sn3-3: It can set AOO, VOO, AAI, VVI, AOOR, VOOR, AAIR, and VVIR modes to operate appropriately.
Sn3-1: Design documentatio n and expert review notes.
Sn3-2: It consists of a detection module, a control unit, and a heart rate control module, generates a stable pacing pulse, and detects the correct speed.


G4: Comprehensive testing and validation of the system.
S4: Execute a series of tests and simulations to validate functionality and safety.
G4-1: Perform functional testing.
G4-2: Conduct performance and tests.
Unit testing
System testing
Tested with Heartview (above documentation testing part)
Integration testing


G5: Compliance with relevant medical device standards.
C2: Applicable standards for pacemakers (e.g., ISO 14708-1).
Sn5-1: Design documentation and expert review notes.
Sn5-2: Design based on the PACEMAKER document provided by McMaster


G6: Effective risk management and hazard mitigation.
S6-1: Identify potential risks and implement appropriate mitigation strategies.
S6-2: It can use duplicate detection algorithms for inaccurate detection.
Sn6: Risk analysis reports and mitigation records.