Assignment 2 - Pacemaker

SFWRENG 3K04: Group 10

Rutvi Patel - 400260481
Arnav Arora - 400227929
Priya Patel - 400262251
Shiv Thakar - 400247588
Jil Shah - 400252316
Yuvraj Bal - 400247720

1.0 Requirements

1.1 Modes

The requirements for the Pacemaker Design is to include eleven different modes: AOO, VOO, AAI, VVI, DOO, AOOR, VOOR, AAIR, VVIR, and DOOR. For the project, all the modes were implemented using the requirements outlined in Table 1.

Table 1: *Mode Requirements*

MODE	Description	Explanation
AOO	Atrial Asynchronous Pacing	● Atrial pacing ● No sensing ● Atrial asynchronous pacing at lower programmed pacing rate ● Pacing is determined by Lower Rate Limit and it occurs at a constant rate regardless of the natural heart rate ● Pulse amplitude and pulse width determine the characteristics
VOO	Ventricle Asynchronous Pacing	● Ventricular Pacing ● No Sensing ● Ventricular Asynchronous Pacing at lower programmed pacing rate. ● Pacing is determined by Lower Rate Limit and it occurs at a constant rate regardless of the natural heart rate ● Pulse amplitude and pulse width determine the characteristics
AAI	Atrial Demand Pacing	● Atrial pacing ● Atrial sensing ● Intrinsic P wave inhibits atrial pacing
VVI	Ventricle Demand Pacing	● Ventricular pacing ● Ventricular sensing ● Senses intrinsic QRS inhibits ventricular pacing
DOO	Dual asynchronous pacing	● Pacing in atrium and ventricle ● Intrinsic P wave and QRS do not affect pacing ● Asynchronous pacing (always pace at lower pacing rate)
AOOR	Atrial Rate Adaptive	● Atrial pacing

	Pacing	• No sensing • Atrial asynchronous pacing at lower programmed pacing rate • Pacing is determined by Lower Rate Limit and it occurs at a constant rate regardless of the natural heart rate • Pulse amplitude and pulse width determine the characteristics • Pacing should change based on the accelerometer activity
VOOR	Ventricle Rate Adaptive Pacing	• Ventricular Pacing • No Sensing • Ventricular Asynchronous Pacing at lower programmed pacing rate. • Pacing is determined by Lower Rate Limit and it occurs at a constant rate regardless of the natural heart rate • Pulse amplitude and pulse width determine the characteristics • Pacing should change based on the accelerometer activity
DOOR	Dual Rate Adaptive Pacing	• Pacing in atrium and ventricle • Intrinsic P wave and QRS do not affect pacing • Asynchronous pacing (always pace at lower pacing rate) • Pacing should change based on the accelerometer activity
AAIR	Atrial Rate Adaptive Demand Pacing	• Atrial pacing • Atrial sensing • Intrinsic P wave inhibits atrial pacing • Pacing should change based on the accelerometer activity • When the lower rate limit is equal to the natural heart rate, the pacemaker does not inhibit any heart paces and will only pace when the activity of the heart changes.
VVIR	Ventricle Rate Adaptive Demand Pacing	• Ventricular pacing • Ventricular sensing • Senses intrinsic QRS inhibits ventricular pacing • Pacing should change based on the accelerometer activity • When the lower rate limit is equal to the natural

		heart rate, the pacemaker does not inhibit any heart paces and will only pace when the activity of the heart changes.

Table 2 below identifies the boolean value for the pace control variables. Depending on the type of pacing required for the mode, the ATR_PACE_CTRL and VENT_PACE_CTRL are set to ON or OFF.

Table 2: *Boolean Value for Pace Control Variables for Each Mode*

ATR_PACE_CTRL	VENT_PACE_CTRL	Result
OFF	OFF	No pacing
ON	OFF	Atrial Pacing
OFF	ON	Ventricular Pacing
ON	ON	Parallel Connection (Not Useful)

1.2 Parameters

Based on the Programmable Parameters table found in the *'PACEMAKER'* document, the unique parameters were identified for the Pacemaker Design implementation and the ranges for all variables. The Programmable Parameters specified in Table 3 are the ones that are being transmitted and received from DCM and Simulink. The specific variable types, number of bytes and the starting byte is specified below and was used for determining the bit values and types on both ends.

Table 3: *Types and Related Byte Allocation for the Parameters*

Parameter	Type	# of Bytes	Starting Byte
dec(22) hex(16)	Byte	1	0
dec(18) hex(12)	Byte	1	1
Mode	unit16	2	2
Lower Rate	unit16	2	4
Upper Rate	unit16	2	6
Atrial Amplitude	double	4	8
Atrial Pulse Width	unit16	2	16

Ventricular Amplitude	double	4	18
Ventricular Pulse Width	unit16	2	26
VRP	unit16	2	28
ARP	unit16	2	30
AV Delay	unit16	2	32
Reaction Time	unit16	2	34
Recovery Time	unit16	2	36
Activity Threshold	double	8	38
Maximum Sensor Rate	unit16	2	46
Rate Smoothing	unit16	2	48
Ventricular Sensitivity	double	4	50
Atrial Sensitivity	double	4	58
Response Factor	unit16	2	66
Total # of Bytes		**52**	

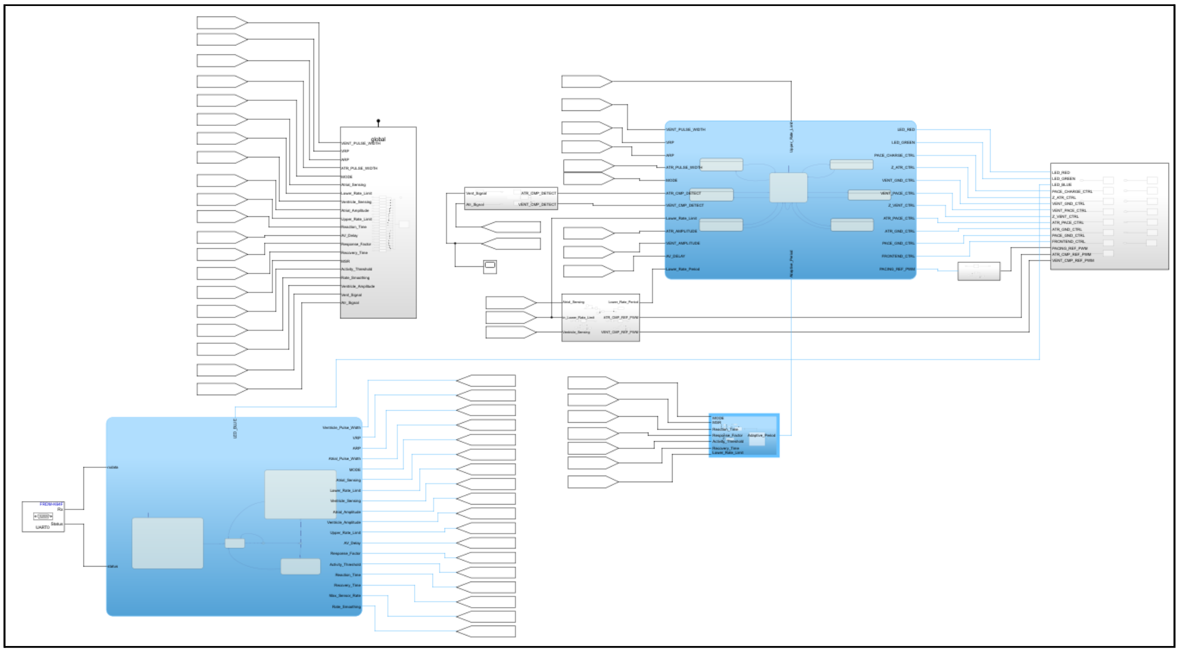
1.3 Requirement Changes

Requirement changes that are expected for the future include the implementation of the DDDR mode with the proper AV delay. For DDDR, there is dual pacing, dual sensing, and rate adaptive which would be a requirement change that can be implemented as it is similar to DOOR. In addition, a requirement change could be to inhibit only ventricular pacing when the push button is pressed and when push button is released the ventricular pacing will appear.

2.0 Design Decisions

To implement the Pacemaker on Simulink, the system was broken down into different modules: Hardware Inputs, Hardware Outputs, Input Conversions, Pacemaker Stateflow, Serial Communication Receiving, Serial Communication Transmitting and Rate Adaptive as shown in Figure 1. A combination of subsystems were used for hardware-hiding to secure any changes that are made within each of the modules. The detailed description for each of the modules can be found below.

Figure 1: *Simulink Overview*



2.1 Serial Communication Receiving (COM_IN)

In Simulink, the Serial Communication Receiving is a chart that is used to communicate with the Device Controller Monitor (DCM) as shown in Figure 2 and 3. The range for the values are determined by the Programmable Parameters found in the ‘PACEMAKER’ document. The implementation uses a serial receive UART block that sends RX data and status to the chart. Within the chart, there are four states. The first state sets the initial parameters for a default mode of 10, and this does not correspond to any of the other 9 modes that have been implemented. The second state is a standby mode which waits for the data to be received from the UART serial monitor. In order to transition into the next state, a hexadecimal 16 and 12 must be received from the DCM. Once the data is received, the type and its corresponding variable name is set for each value. Then only if a hexadecimal 16 and 16 are received from the serial monitor the parameters are sent back.

Figure 2: *Serial Communication Receiving Overview*

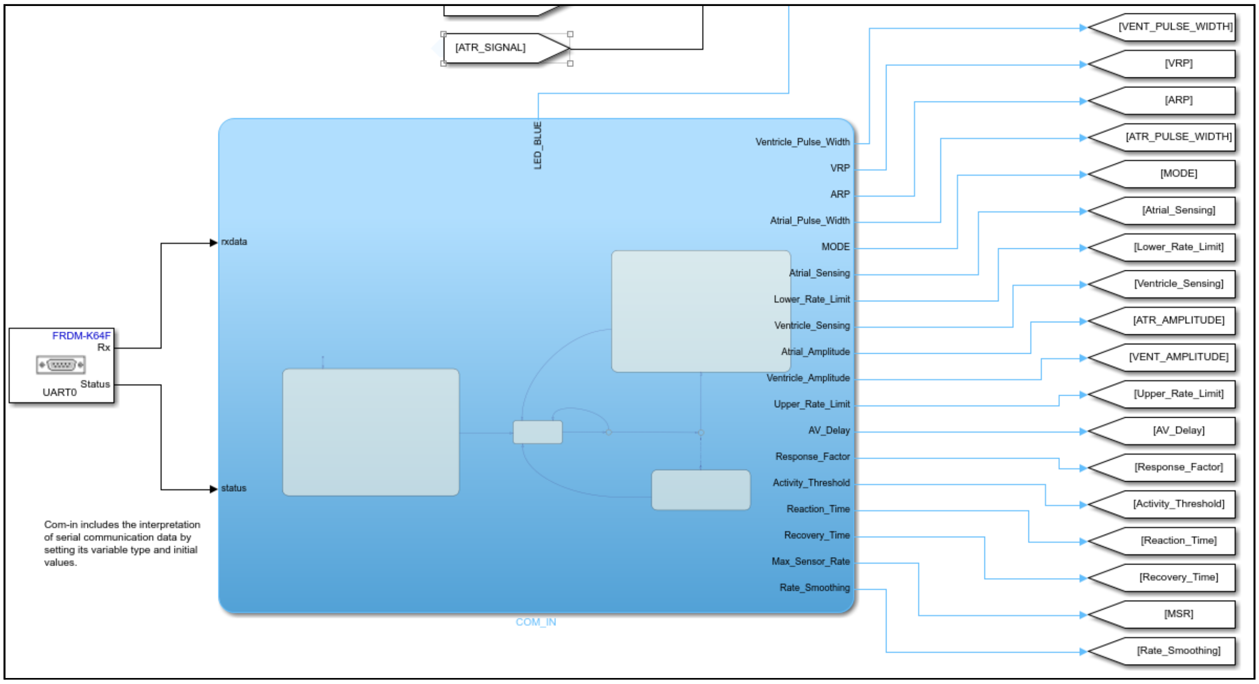
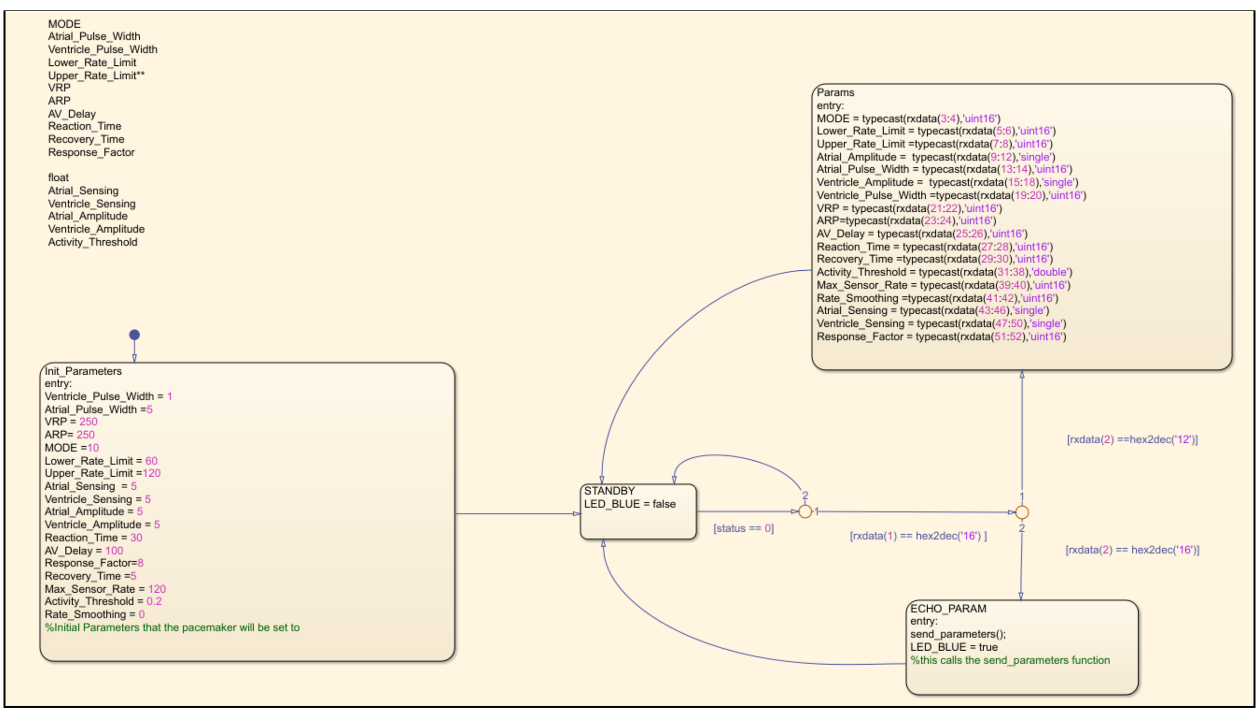


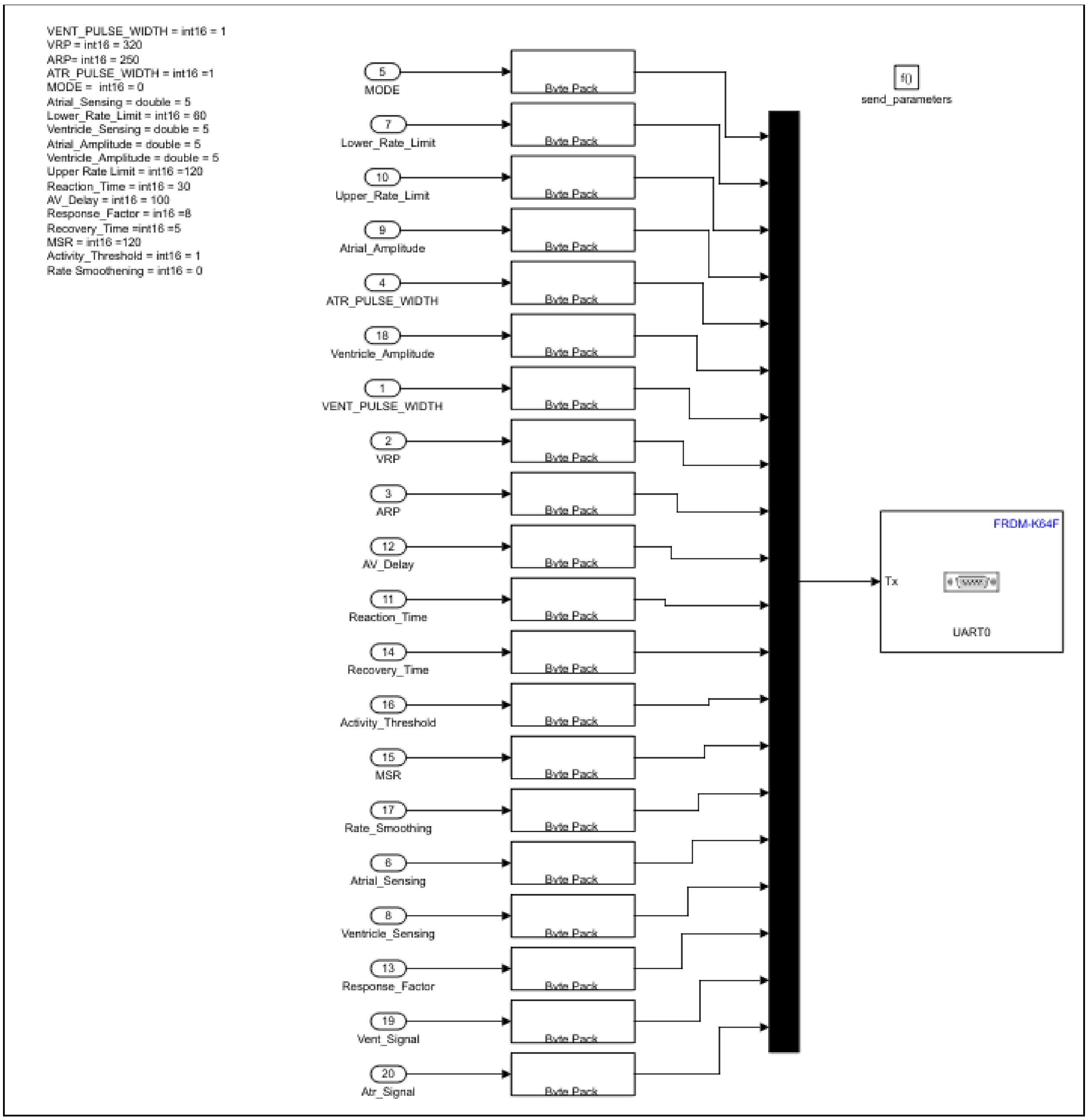
Figure 3: *Serial Communication Receiving*



2.2 Serial Communication Transmitting (COM_OUT)

For Simulink, the Serial Communication Transmitting is a subsystem that is used to package the parameters using a demultiplexer and byte packaging. This information is then sent to the DCM using UART serial communication. The subsystem was connected to a function called ‘Send_Parameters’ which is used in the system as shown in figure 4.

Figure 4: *Serial Communication Transmitting*



2.3 Rate Adaptive

The Rate Adaptive subsystem is used to specify the pacing rate at a given time based on the activity level detected. The rate adaptive calculation is only performed for modes that range between 4-10. The input parameters for the rate calculation are Mode, Maximum Sensory Rate (MSR), Reaction Time, Response Factor, Activity Threshold, Recovery Time, Lower Rate Limit (LRL) as shown in figure 5. Inside the rate adaptive subsystem, there are three subsystems that determine the activity level, target rate and then output a final Adaptive Period. The individual subsystems are explained below.

Figure 5: *Rate Adaptive Overview*

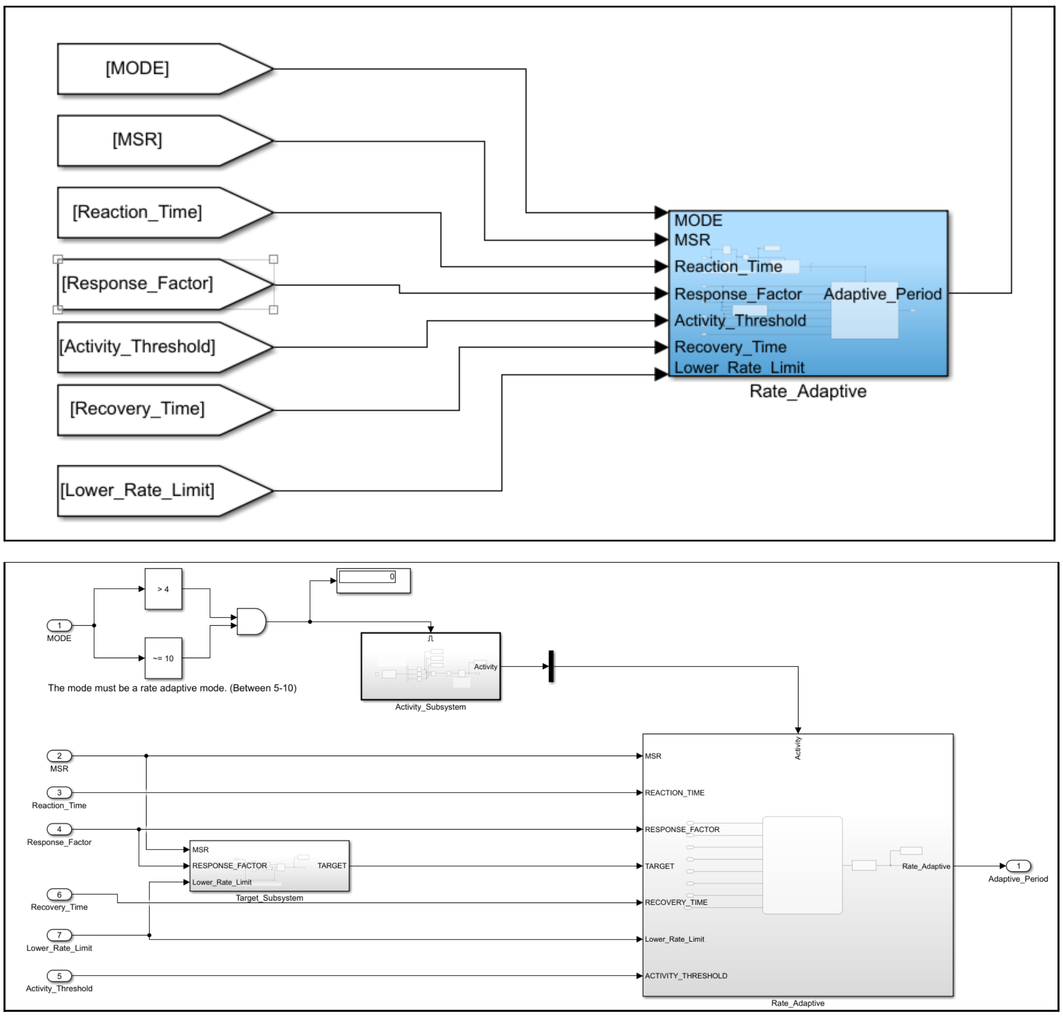
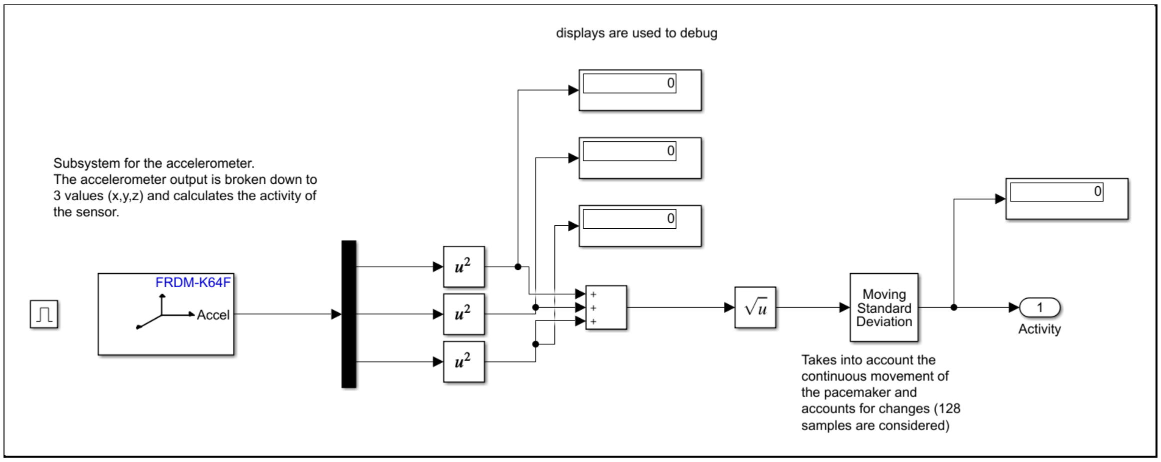


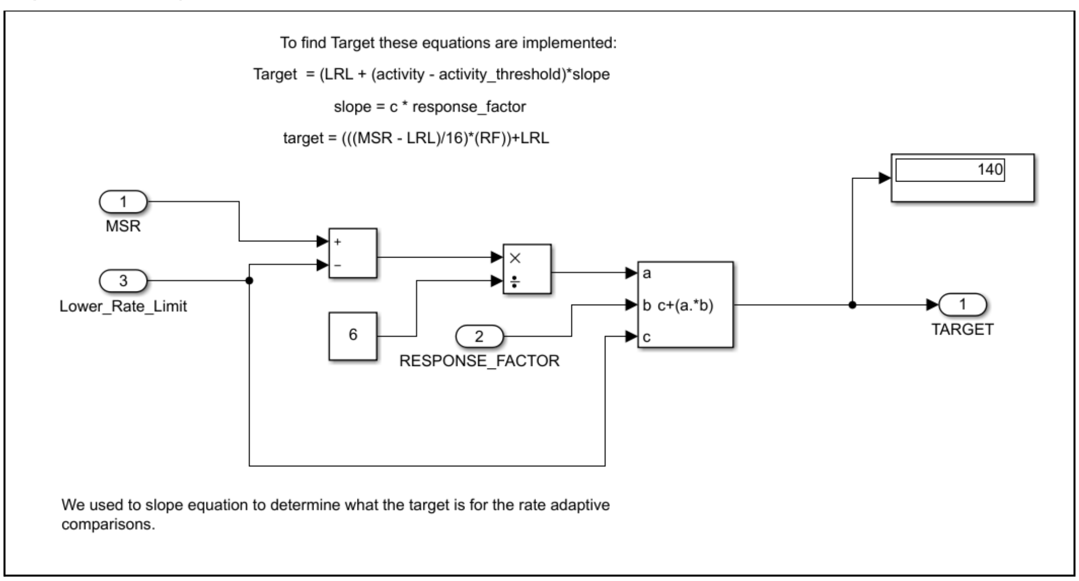
Figure 6 below is a subsystem for the accelerometer that is used to detect and continuously determine the activity level. The accelerometer receives the sensory data from the xyz plane and then uses a moving standard deviation block to account for continuous movements within an input signal over-time independently and an activity level is outputted.

Figure 6: *Accelerometer Activity Calculation*



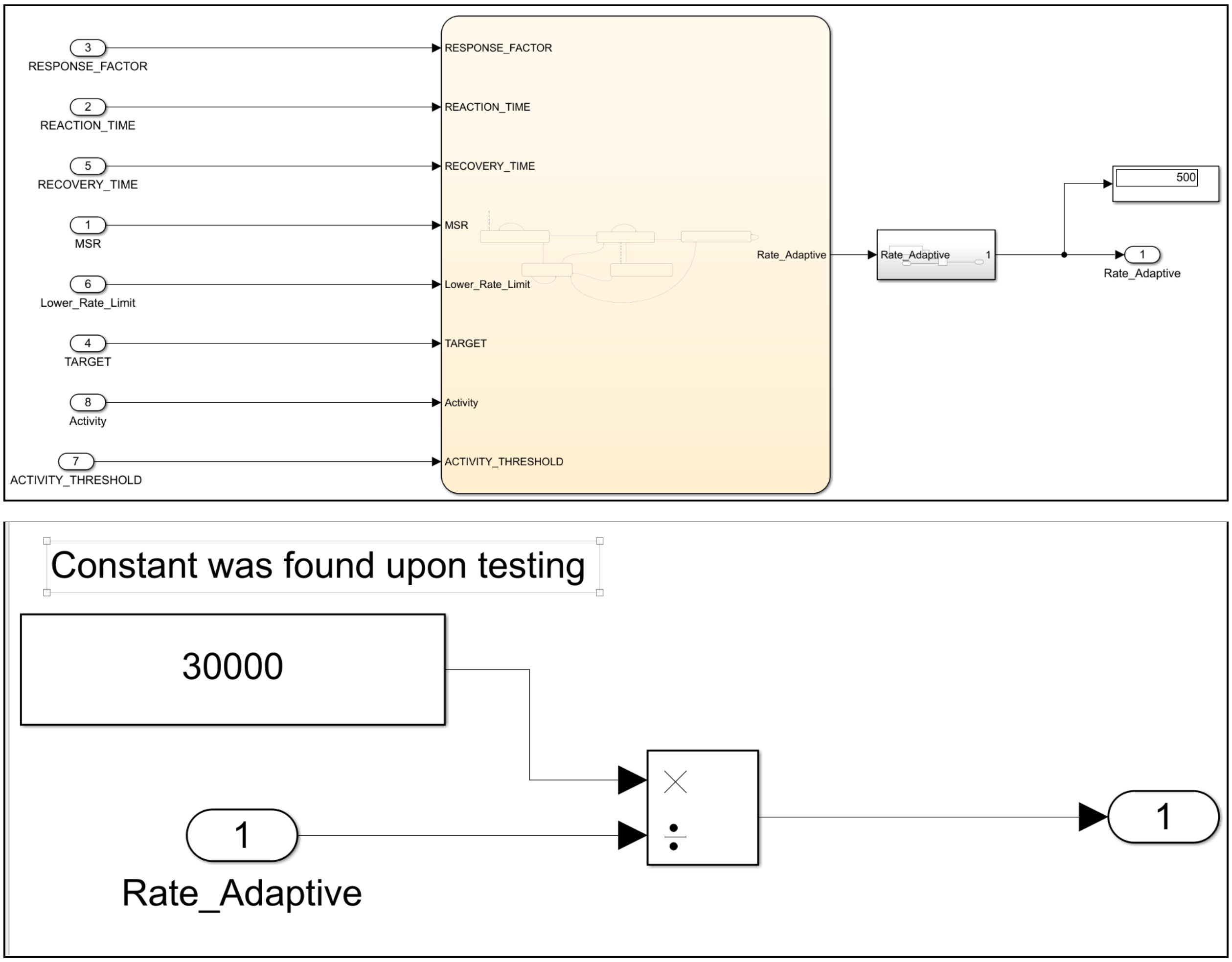
The subsystem block shown below is used to output a target rate. The target rate is time independent and is calculated by subtracting the lower rate limit from the maximum sensory rate. The result is then multiplied by a constant of 16 which is set as ‘a’ as shown in the figure 7. Finally, the value of ‘a’ is multiplied by the response factor and added to the lower rate limit. If the activity level is less than or equal to the activity threshold, then the target is set to LRL. If the activity level is greater than the threshold then the target is a function of the response factor and activity f(RF, activity).

Figure 7: *Target Rate Calculation*



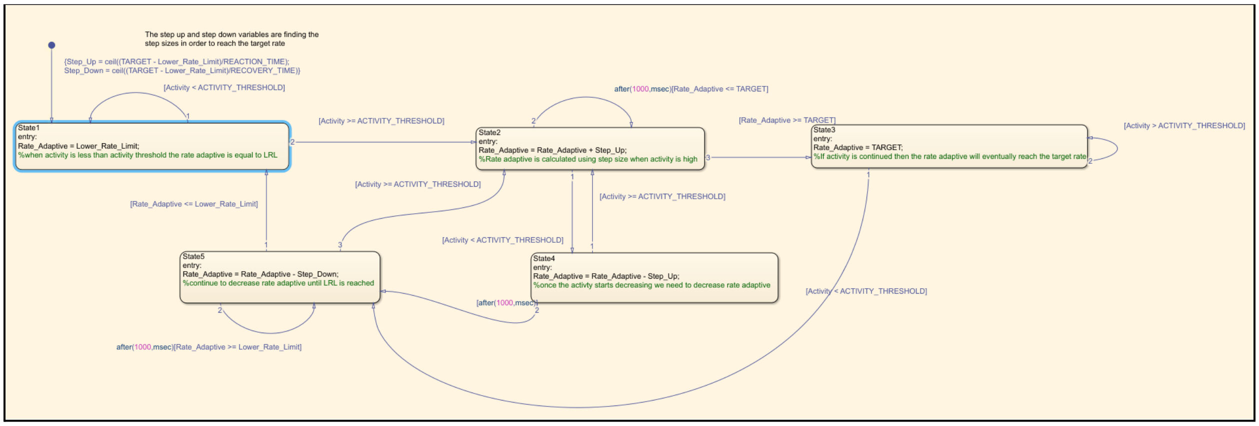
The rate adaptive sub-system takes in all of the necessary input variables into the chart as shown in figure 8. The output of the chart is the adaptive rate. As the output ranges between 50-80, we implemented an equation that divides a constant by the rate adaptive chart value and sends this converted rate modulated period to the pacemaker stateflow.

Figure 8: *Rate Adaptive Calculation*



In figure 9, the adaptive rate is determined from the activity level. Initially, the step up and step down values were calculated which adjust the step sizes to achieve the target rate. There are 5 states which are finding the adaptive rate based on the activity level and activity threshold. In state 1 we set the adaptive rate to the lower rate limit which will stay that way when the activity is less than the activity threshold or when adaptive rate is less than or equal to the lower rate limit and will eventually return back to this state once the activity level starts decreasing. In state 2 the adaptive rate is incremented using step up which will happen once the activity is greater than the activity threshold or when the adaptive rate is less than the target rate. In state 3 the adaptive rate is set to the target rate which happens once the adaptive rate is greater than or equal to the target. In state 4 the adaptive rate is decremented using step up which will happen once activity is less than activity threshold and it is state 2. In state 5 the adaptive rate is decremented using step down which happens when activity is less than activity threshold.

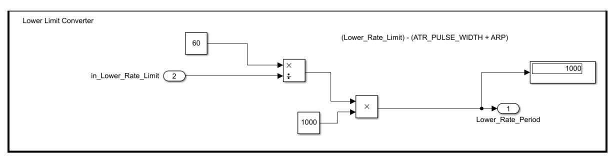
Figure 9: *Rate Adaptive Stateflow*

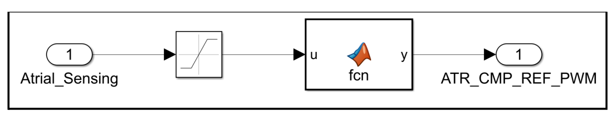


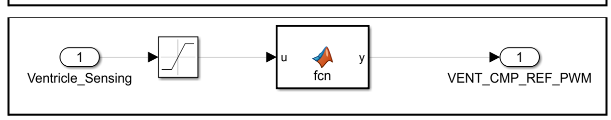
2.4 Input Conversion

The input conversion is a subsystem used to convert some of the inputs to its ideal unit required for the Pacemaker. Each of the conversions performed for the Pacemaker are outlined in Table 4.

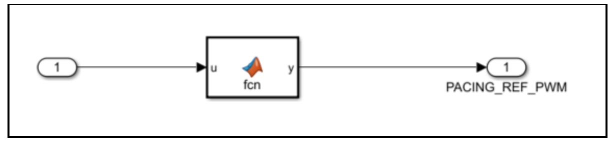
Table 4: *Input Conversion*

<table>
<tr><td>Lower Rate Limit
The conversion for Lower Rate Limit (LRL) is done using a MATLAB function. The LRL input in milliseconds is converted to a rate by dividing the input by 60 and then multiplying it by 1000.</td><td>


</td></tr>
<tr><td>Atrial and Ventricle Sensing
The saturation block is used to pull up or pull down the input voltage which is then passed into a MATLAB function for conversion. The MATLAB function converts the input signal by dividing it by 5 and then multiplying by 100. The same process is used for both atrial and ventricle sensing. The output is then stored in ATR_CMP_REF_PWM or VENT_CMP_REF_PWM accordingly.</td><td>




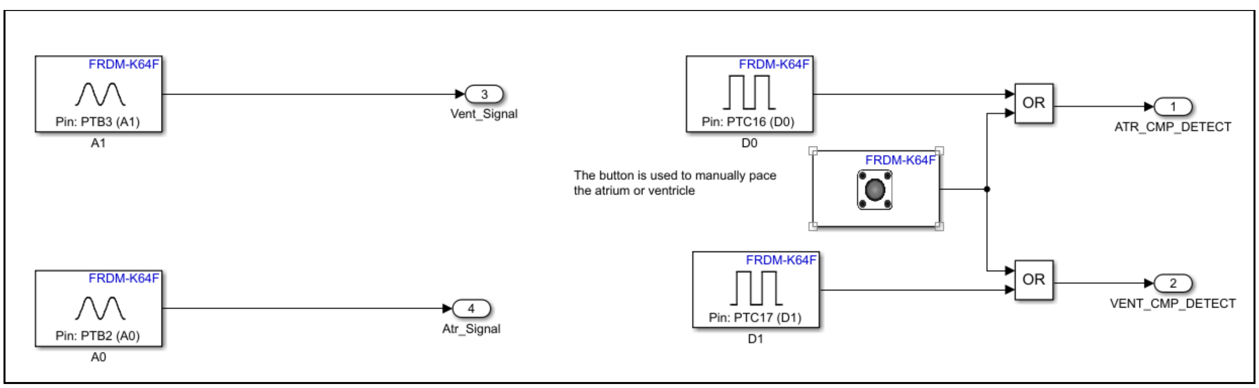

```
function y = fcn(u)
y = u./5.*100;
```
</td></tr>
</table>

Atrial and Ventricle Amplitude Selection
The function converts the amplitude based on the maximum voltage (i.e. 5V). The function divides the input by 5, and then is multiplied by 100 which is stored in PACING_REF_PWM. Inside each mode, the corresponding amplitude is assigned. For example, for the implementation of AOO, AAI, AOOR, and AAIR the Atrial Amplitude is set. For VOO, VVI, VOOR, and VVIR the Ventricle Amplitude is assigned.
	
	`function y = fcn(u)`
	`y = u./5.*100;`

2.5 Hardware Outputs

The hardware output is a subsystem that is used to detect if a pulse is being received from the atrium and the ventricle using a digital read input pin. ATR_CMP_DETECT stores the value as a boolean if a signal is detected in the atrium and VENT_CMP_DETECT stores the value as a boolean if a signal is detected in the ventricle. As shown in Figure 10, pins D0 and D1 identify the corresponding signal as inputs since the signals are being sent into the chart. Additionally, a button was implemented to manually pace the atrium or ventricle.

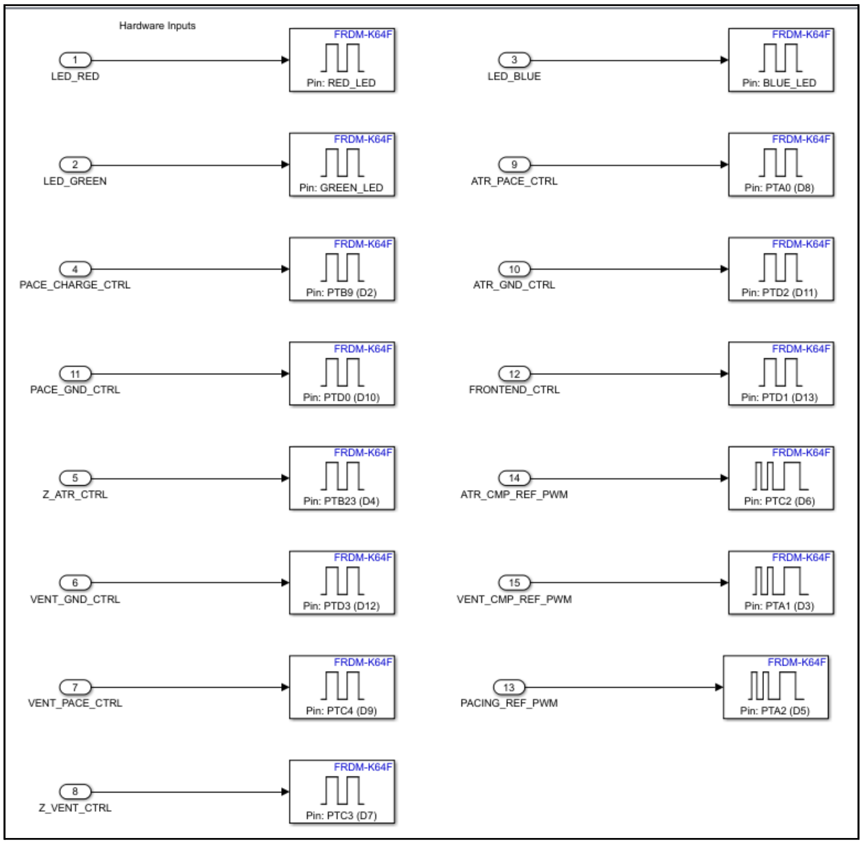
Figure 10: *Simulink Input Pin Assignment*



2.6 Hardware Inputs

The hardware inputs allocate all of the variable names to its corresponding digital write output pin as outlined in the *'pacemaker_shield_explained'* document and is shown in Figure 11. The inputs are determined from the Pacemaker Stateflow and the Input Conversions. The hardware inputs are changed depending on what the variables are set to in the previous modules. For instance, the LED_RED and LED_GREEN change depending on the boolean value True or False. The digital outputs are either PWM Outputs or Digital Write. The PWM Output accepts any values from 0 to 100 and is used to control the duty cycle of the pulse, and the Digital Write block internally converts the value to a boolean.

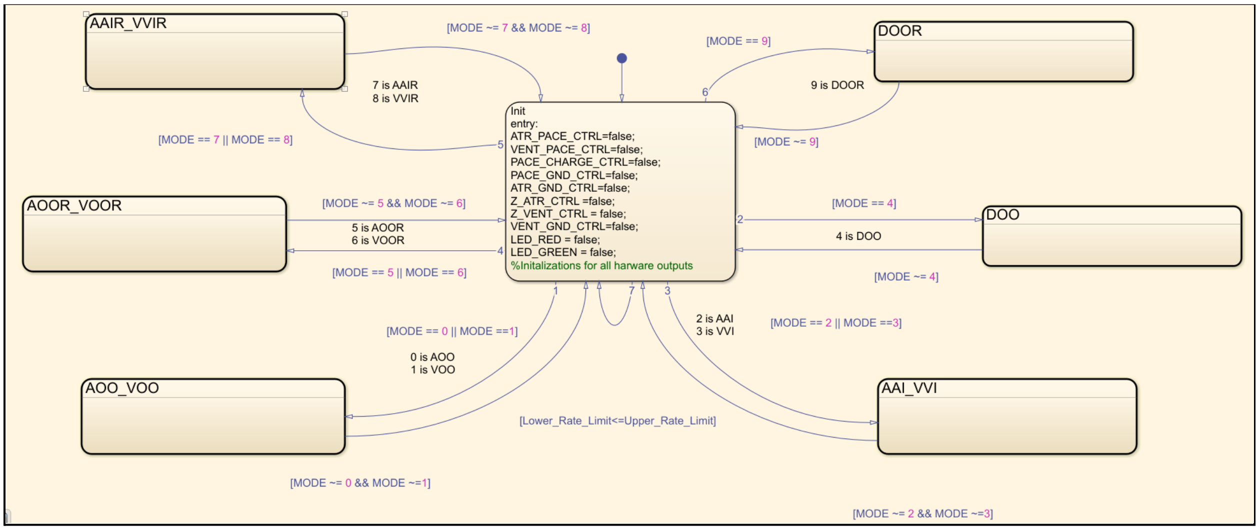
Figure 11: *Simulink Output Pin Assignment*



2.7 Pacemaker Stateflow

The pacemaker stateflow is implemented using a state chart and sub-charts as shown in figure 12. All of the variables in the state chart were treated as inputs from previous modules explained above. The state chart was broken down into seven different states: Initial State, AOO & VOO State, AAI & VVI State, DOO state, AOOR & VOOR state, AAIR & VVIR, and DOOR state outlined below. Inside the individual states for each mode, the PACING_REF_PWM is assigned to ATR_AMPLITUDE or VENT_AMPLITUDE depending on the mode.

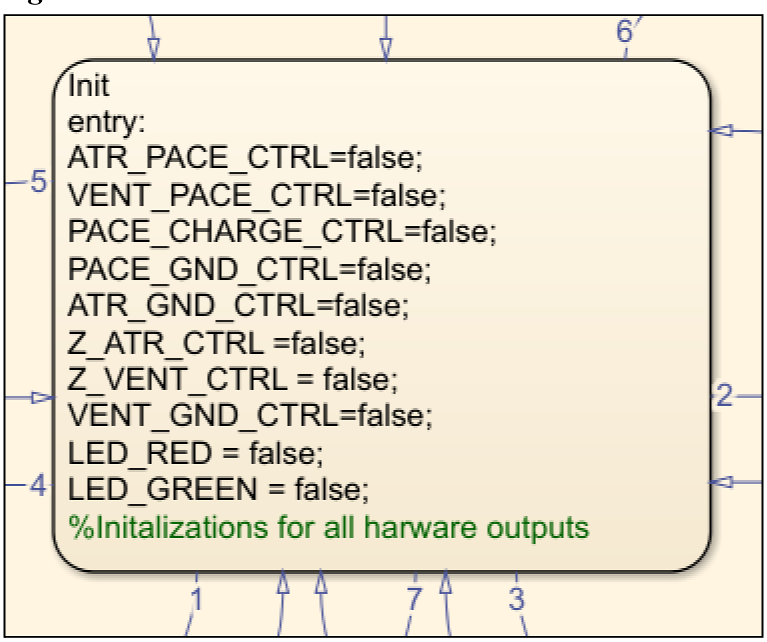
Figure 12: *Modes Implementation Overview*



2.5.1 Initial State

The initial state or the default state sets the variables to its initial condition which is false. The state transitions into the AOO/VOO, AAI/VVI, DOO, AOOR/VOOR, AAIR/VVIR and DOOR mode depending on the value set by the DCM. For example, if the mode is 0 or 1, it enters the AOO/VOO sub-chart as shown in Figure 13.

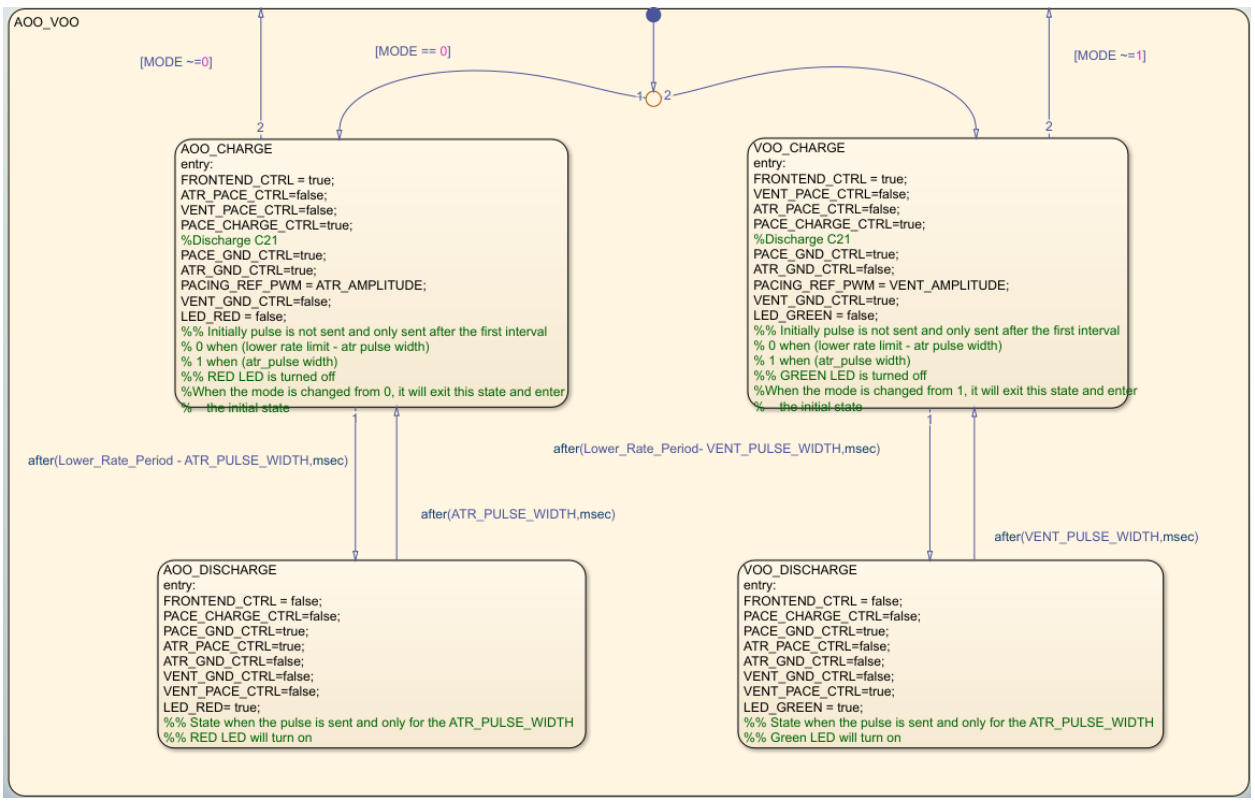
Figure 13: *Initial State*



2.5.2 AOO and VOO

The AOO mode is the Atrial Asynchronous Pacing which identifies the chambers being paced in the atrium and disregards sensing or response to sensing. The VOO mode functions similarly but is the Ventricle Asynchronous Pacing. The AOO and VOO modes are implemented using identical methods. There are two states for each mode. The AOO_CHARGE state is used for charging and discharging the capacitors C22 and C21 respectively, while the AOO_DISCHARGE is used for AV pacing. As shown in Figure 14, for the AOO_CHARGE state the conditions ATR_GND_CTRL and VENT_GND_CTRL are set to True and False respectively. The state transitions to the AOO_DISCHARGE after a delay of (Lower_Rate_Limit - ATR_PULSE_WIDTH). Inside that state, ATR_PACE_CTRL is set to True and VENT_PACE_CTRL is set to false. Then, it returns back to AOO_CHARGE after a delay of the ATR_PULSE_WIDTH. The VOO mode is implemented similarly, while changing the conditions from True to False and vice versa.

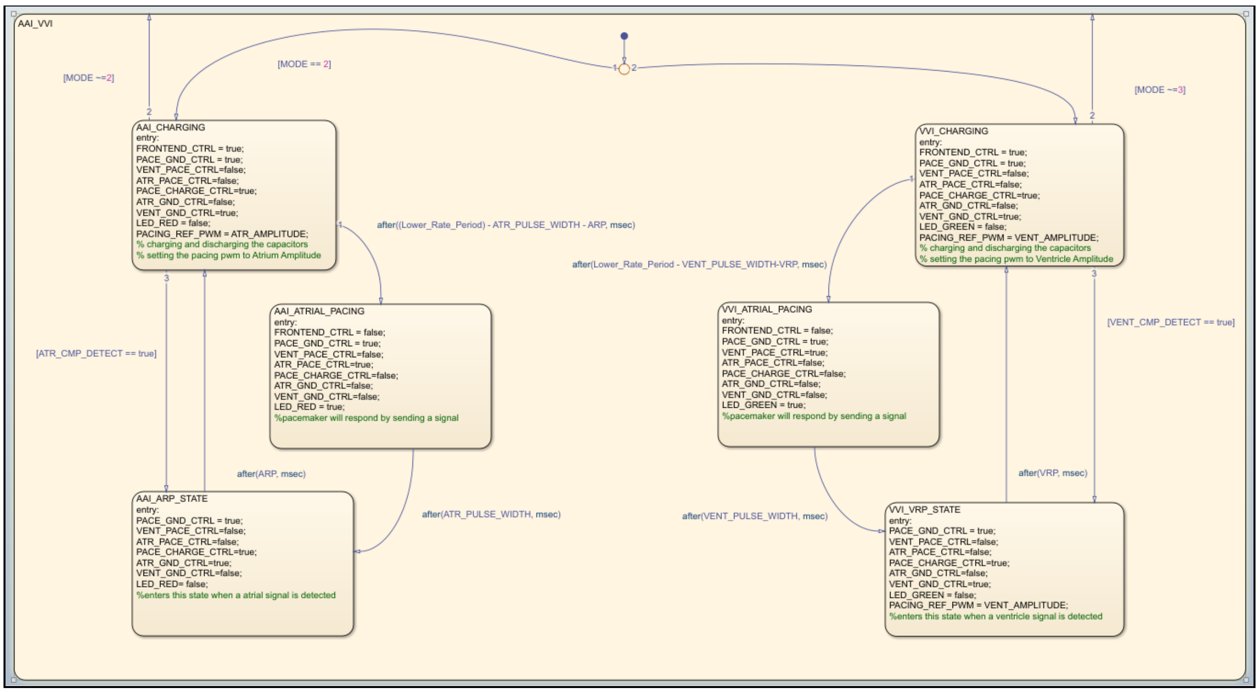
Figure 14: *AOO and VOO Mode*



2.5.3 AAI and VVI

The AAI mode is the Atrial Demand Pacing which identifies the atrium chamber being paced, the atrium chamber being sensed and inhibits a response to sense. The VVI mode functions similarly, but is the Ventricle Demand Pacing. The AAI and VVI modes are implemented in identical ways. As shown in Figure 15, there are three states for each mode. The initial state charges and discharges the capacitors, and then transitions into the AAI_ARP_STATE when a signal from the heart has been detected. Then the pacemaker sends a response for the heart signal being detected using the next two states. It transitions from state 1 to state 2 and then back where it charges and discharges the capacitors again. Then, it moves to AAI_ATRIAL_PACING state where the pacemaker will respond by sending a signal. The VVI mode is implemented similarly, while changing the values for the conditions and the delays.

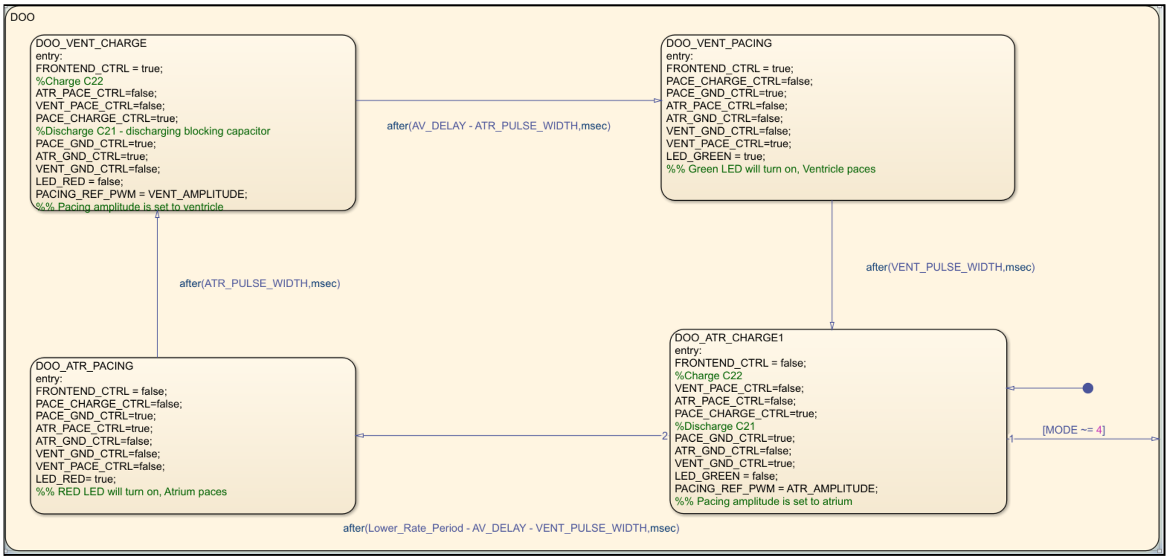
Figure 15: *AAI and VVI Mode*



2.5.4 DOO

The DOO mode is the Dual Pacing Mode which identifies the atrium and ventricle being paced and inhibits a response to sense. The DOO Mode is a combination of the AOO and VOO mode and uses the AV delay to determine the next pulse and change the amplitude. As shown in Figure 16, there are four states for the mode. The initial state is the DOO_VENT_CHARGE which charges and discharges the capacitors C22 and sets the PACING_REF_PWM to the ATR_AMPLITUDE. Similarly, the DOO_ATR_CHARGE state charges and discharges the capacitors but sets the PACING_REF_PWM to VENT_AMPLITUDE. Then it transitions into the different states using the delay conditions. The period accounts for the AV delay which is the distance between the atrial pulse and the ventricle pulse.

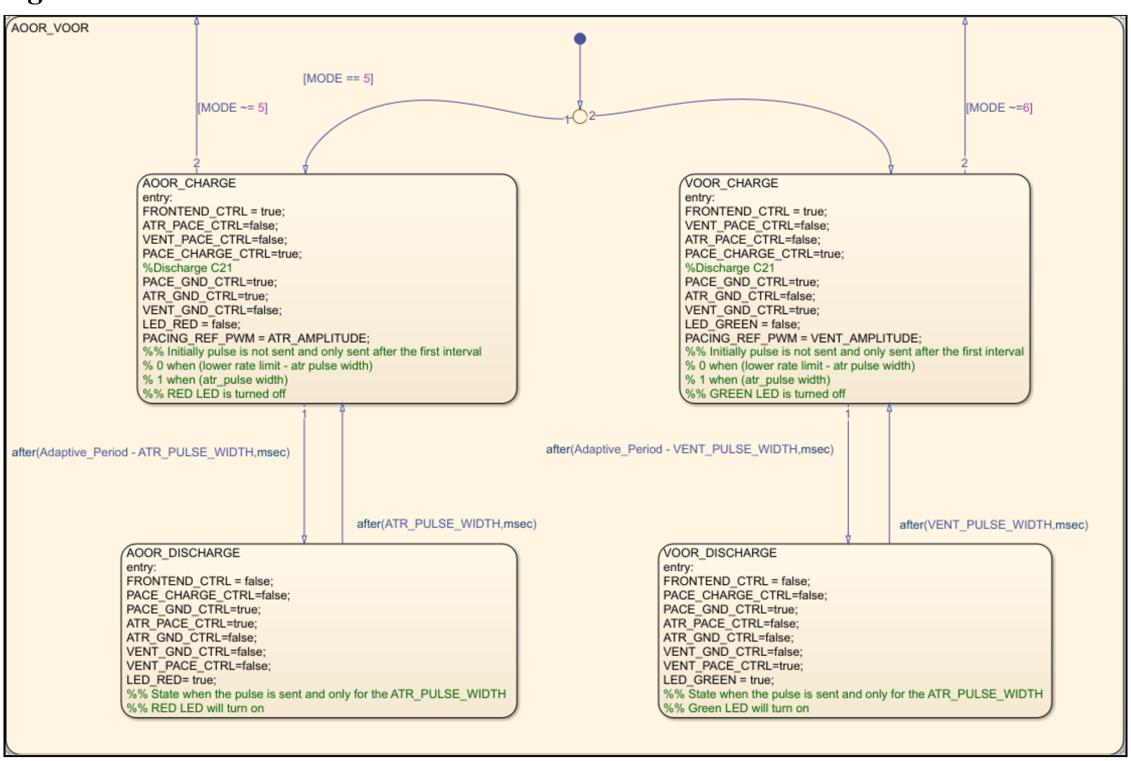
Figure 16: *DOO Mode*



2.5.5 AOOR and VOOR

The AOOR and VOOR mode use the same implementation logic as the AOO and VOO mode; however, it also accounts for the adaptive period as shown in figure 17. The rate is used to continuously recalculate the transition and the timing between the pulses depending on the activity that is detected.

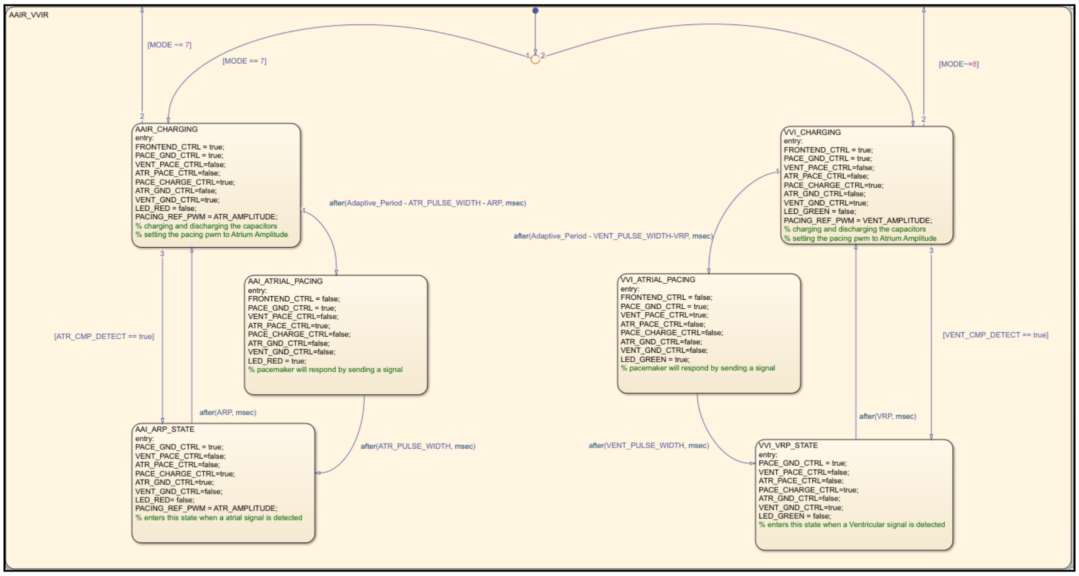
Figure 17: *AOOR and VOOR Mode*



2.5.6 AAIR and VVIR

The AAIR and VVIR mode use the same implementation logic as the AAI and VVI mode; however, it also accounts for the adaptive period as shown in figure 18. The rate is used to continuously recalculate the transition and the timing between the pulses depending on the activity that is detected.

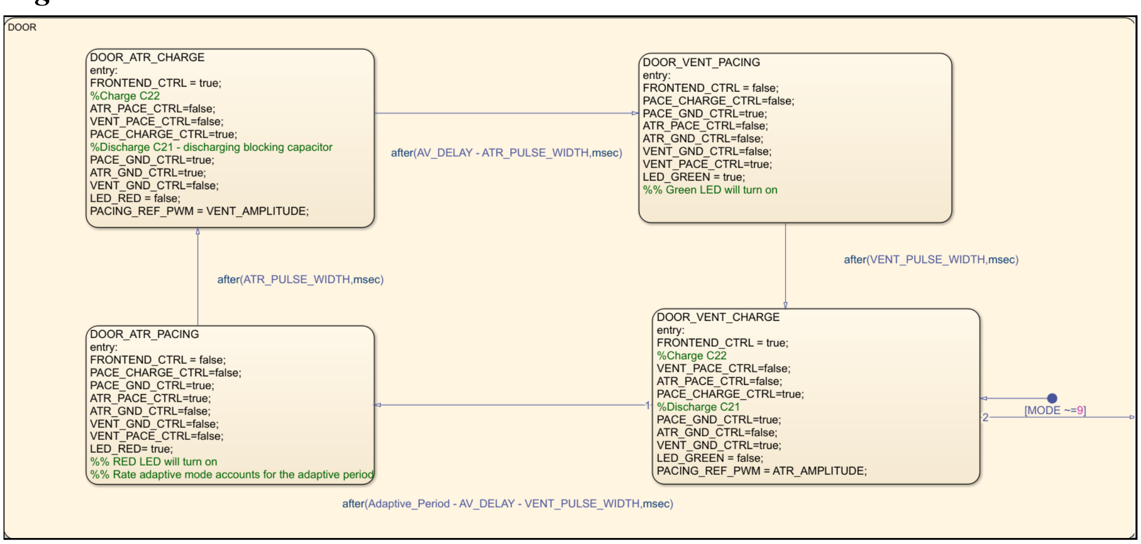
Figure 18: *AAIR and VVIR Mode*



2.5.7 DOOR

The DOOR mode uses the same implementation logic as the DOO mode; however, it also accounts for the adaptive period as shown in figure 19. The rate is used to continuously recalculate the transition and the timing between the pulses depending on the activity that is detected.

Figure 19: *DOOR Mode*



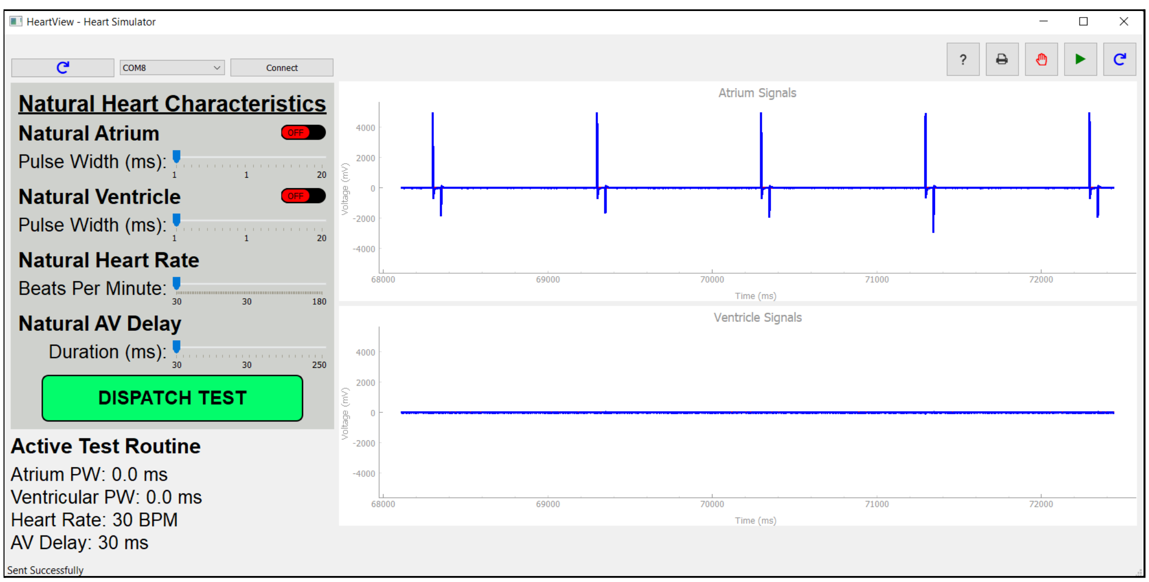
2.6 Design Decisions Changes

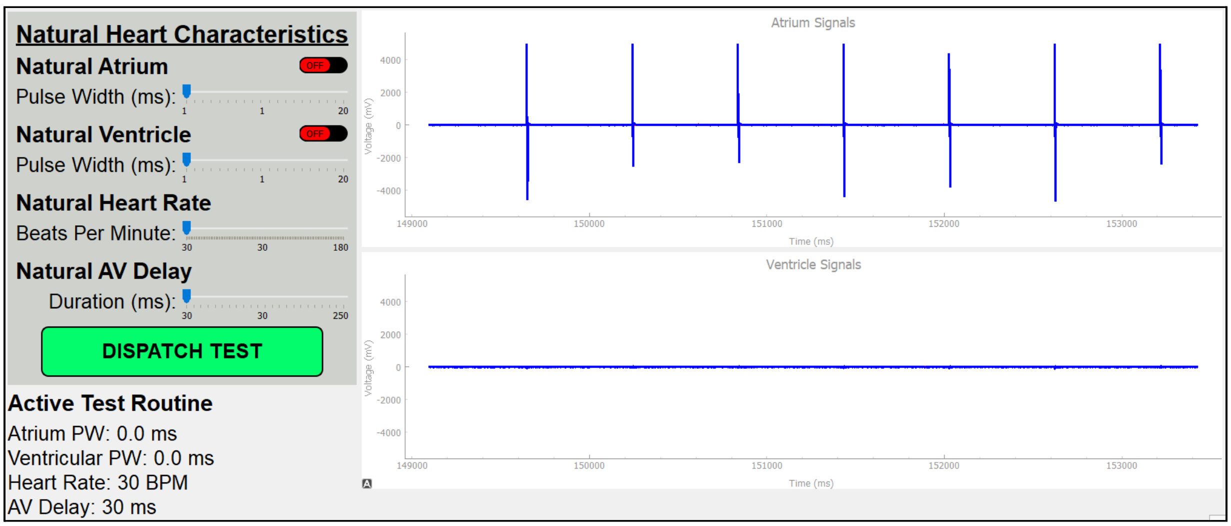
The design changes that are likely to be implemented include simplifying the sub-charts. For each mode, the states will be combined into one state and if statements will be used to ensure that it is pacing, inhibiting, and sensing for the correct mode. Another design decision change that could be implemented is changing the units for recovery time inside simulink from minutes to seconds. This will be done by adding a conversion block for the input of recovery time in which the DCM passed parameter will be multiplied by 60. The multiplication of 60 ensures that the parameter is converted from minutes to seconds to reflect the implementation in the rate adaptive stateflow. This design decision will allow the paces to recover back to the original pace when the activity is not occurring at a faster rate making the hardware more efficient. Also, the variable types should be changed where a lower range of numbers can be used to decrease the number of bytes allocated to make the implementation of serial communication memory efficient.

3.0 Testing

3.1 AOO and VOO Mode

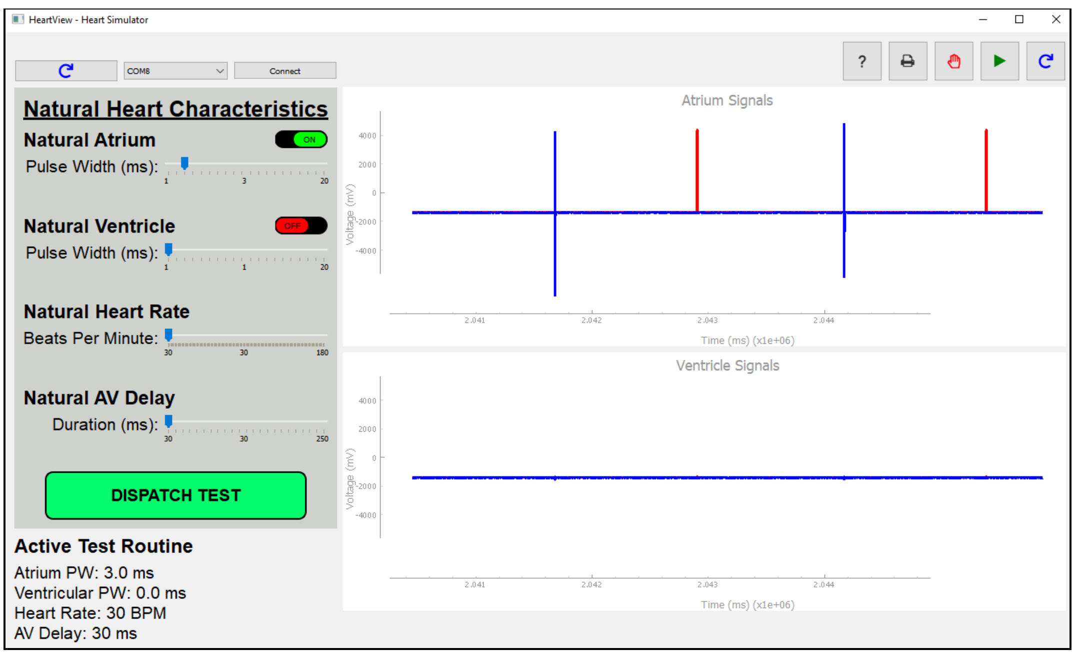
TEST CASE 1	
PACEMAKER	**HEARTVIEW**
Lower Rate Limit (ppm): 60 Amplitude (V): 5 Pulse Width (ms): 20	Natural Atrium: OFF Natural Ventricle: OFF Natural Heart Rate (bpm): 30 Natural AV Delay (bpm): 30
EXPECTED OUTPUT	
In the AOO/VOO Mode, the pacemaker will be pacing at a rate based on the Lower Rate Limit regardless of the Natural Heart Rate. As seen in the test cases below, the AOO Mode functions correctly and meets the requirements for the mode. The VOO mode functions similarly for the ventricle.	
ACTUAL OUTPUT → RESULT: PASS	

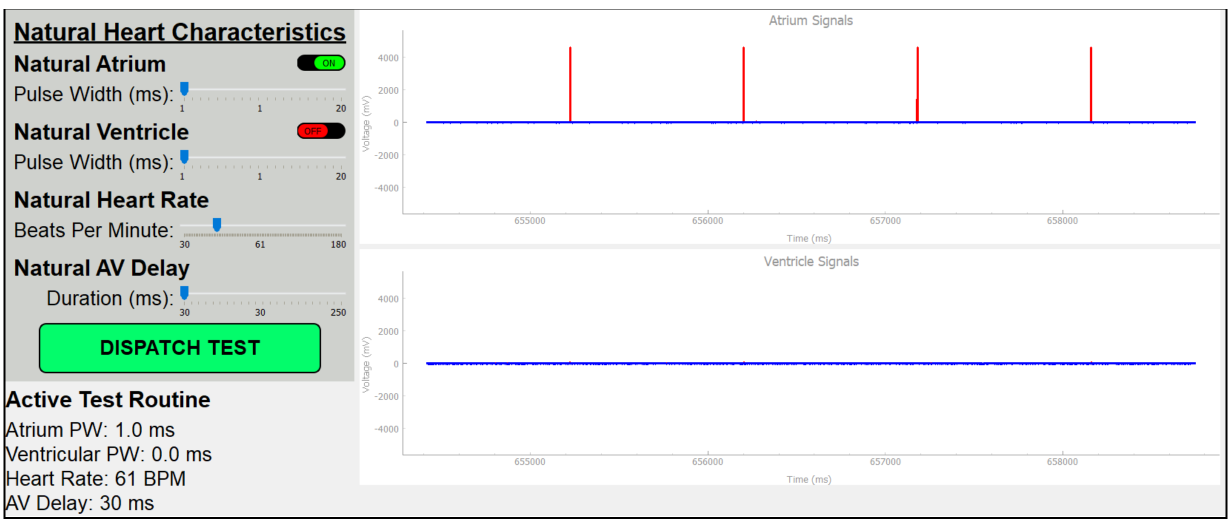


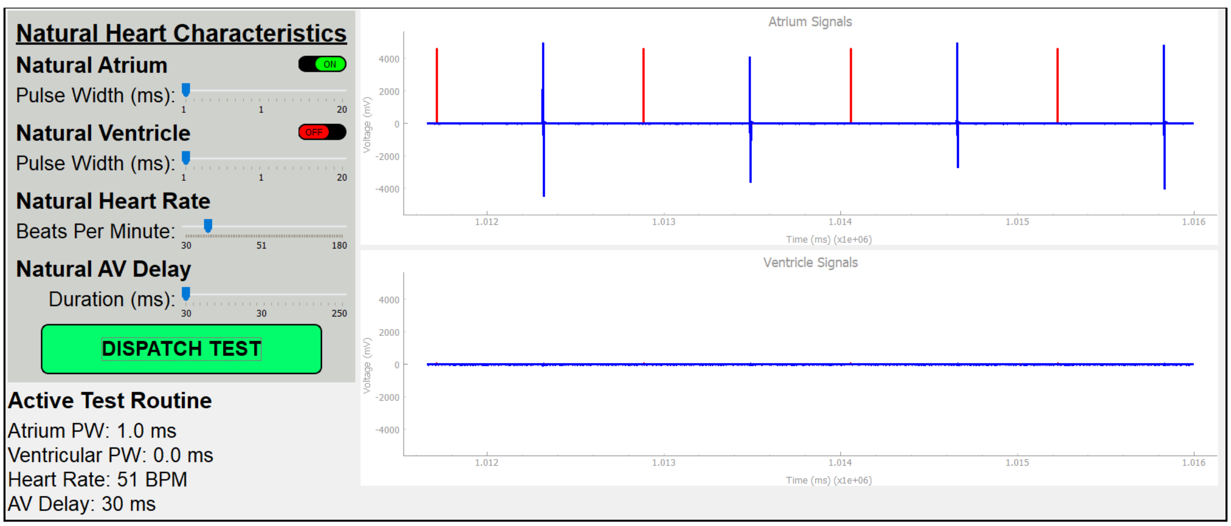
<table>
<tr><th colspan="2">TEST CASE 2</th></tr>
<tr><th>PACEMAKER</th><th>HEARTVIEW</th></tr>
<tr><td>Lower Rate Limit (ppm): 100
Amplitude (V): 5
Pulse Width (ms): 20</td><td>Natural Atrium: OFF
Natural Ventricle: OFF
Natural Heart Rate (bpm): 30
Natural AV Delay (bpm): 30</td></tr>
<tr><th colspan="2">EXPECTED OUTPUT</th></tr>
<tr><td colspan="2">The AOO/VOO mode works similarly as explained above. As seen below, the AOO Mode functions correctly for a higher Lower Rate Limit. The VOO mode functions similarly for the ventricle.</td></tr>
<tr><th colspan="2">ACTUAL OUTPUT → RESULT: PASS</th></tr>
<tr><td colspan="2">
</td></tr>
</table>

3.2 AAI and VVI Mode

TEST CASE 1:	
PACEMAKER	**HEARTVIEW**
Lower Rate Limit (ppm): 60 Amplitude (V): 5 Pulse Width (ms): 1	Natural Atrium: ON • Pulse Width (ms): 3 Natural Ventricle: OFF Natural Heart Rate (bpm): 30 Natural AV Delay (bpm): 30
EXPECTED OUTPUT	
In the AAI/VVI Mode, the pacemaker will be pacing at a rate based on the Lower Rate Limit and the Natural Heart Rate. The Pacemaker for the AAI mode paces between each heart pulse as it inhibits the pace, this occurs when the natural heart rate is less than the lower rate limit. Depending on the Lower Rate Limit and the Natural Heart Rate, the output changes. As seen below, the AAI Mode functions correctly and meets the requirements for the mode. The VVI mode functions for the ventricle.	
ACTUAL OUTPUT → RESULT: PASS	



<table>
<tr><th colspan="2">TEST CASE 2</th></tr>
<tr><td>PACEMAKER</td><td>HEARTVIEW</td></tr>
<tr><td>Lower Rate Limit (ppm): 60
Amplitude (V): 5
Pulse Width (ms): 1</td><td>Natural Atrium: ON
• Pulse Width (ms): 1
Natural Ventricle: OFF
Natural Heart Rate (bpm): 61
Natural AV Delay (bpm): 30</td></tr>
<tr><td colspan="2">EXPECTED OUTPUT</td></tr>
<tr><td colspan="2">The AAI/VVI mode works similarly as explained above. Since the natural heart rate is greater than the lower rate limit, the pacemaker does not send a pace. As seen below, the AAI Mode functions correctly and meets the requirements for the mode. The VVI mode functions for the ventricle.</td></tr>
<tr><td colspan="2">ACTUAL OUTPUT → RESULT: PASS</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

<table>
<tr><th colspan="2">TEST CASE 3:</th></tr>
<tr><th>PACEMAKER</th><th>HEARTVIEW</th></tr>
<tr><td>Lower Rate Limit (ppm): 100
Amplitude (V): 5
Pulse Width (ms): 1</td><td>Natural Atrium: ON
• Pulse Width (ms): 1
Natural Ventricle: OFF
Natural Heart Rate (bpm): 51
Natural AV Delay (bpm): 30</td></tr>
<tr><th colspan="2">EXPECTED OUTPUT</th></tr>
<tr><td colspan="2">The AAI/VVI mode works similarly as explained above. As seen below, the AAI Mode functions correctly for a higher Lower Rate Limit. The VVI mode functions similarly for the ventricle.</td></tr>
<tr><th colspan="2">ACTUAL OUTPUT → RESULT: PASS</th></tr>
<tr><td colspan="2">
</td></tr>
</table>

3.3 DOO Mode

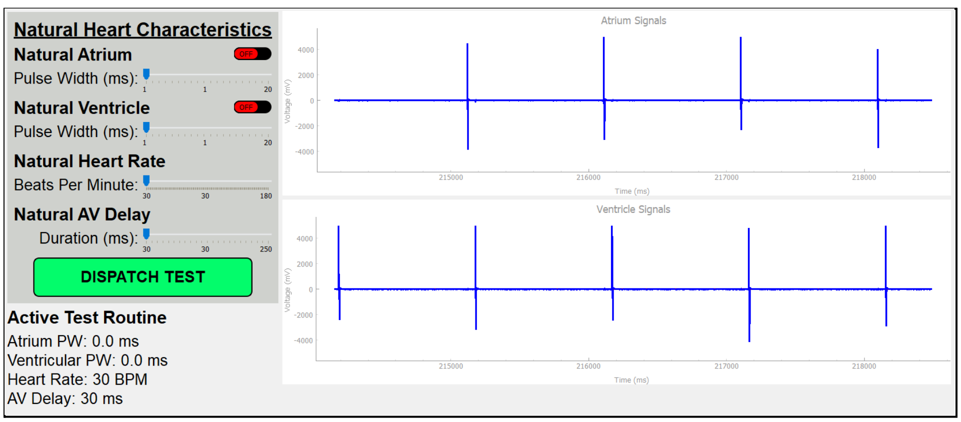
TEST CASE 1:

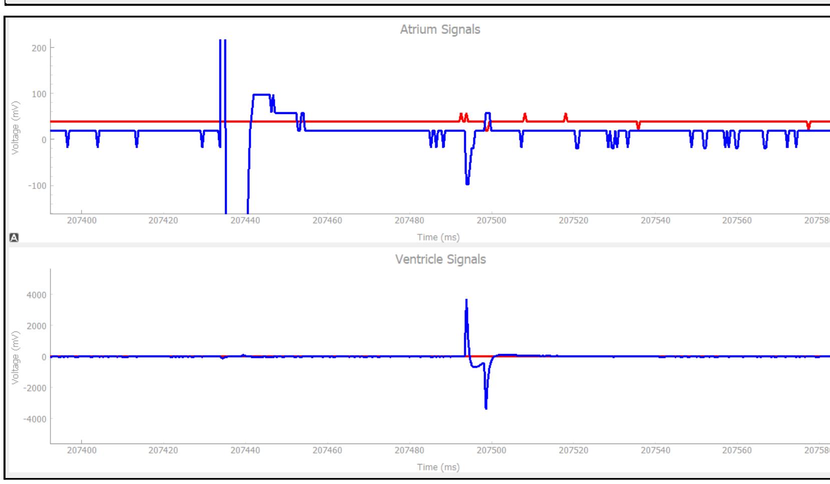
PACEMAKER	HEARTVIEW
Lower Rate Limit (ppm): 60 Amplitude (V): 5 Pulse Width (ms): 20 AV Delay (ms): 60	Natural Atrium: OFF Natural Ventricle: OFF Natural Heart Rate (bpm): 30 Natural AV Delay (bpm): 30

EXPECTED OUTPUT

The DOO mode is the Dual Pacing Mode which identifies the atrium and ventricle being paced. The DOO Mode is a combination of the AOO and VOO mode and uses the AV delay to determine the next pulse and changes the amplitude. As seen in the output below, the atrium and ventricle pulses are delayed by 60ms which is what the AV delay is set to demonstrating that the test case works correctly.

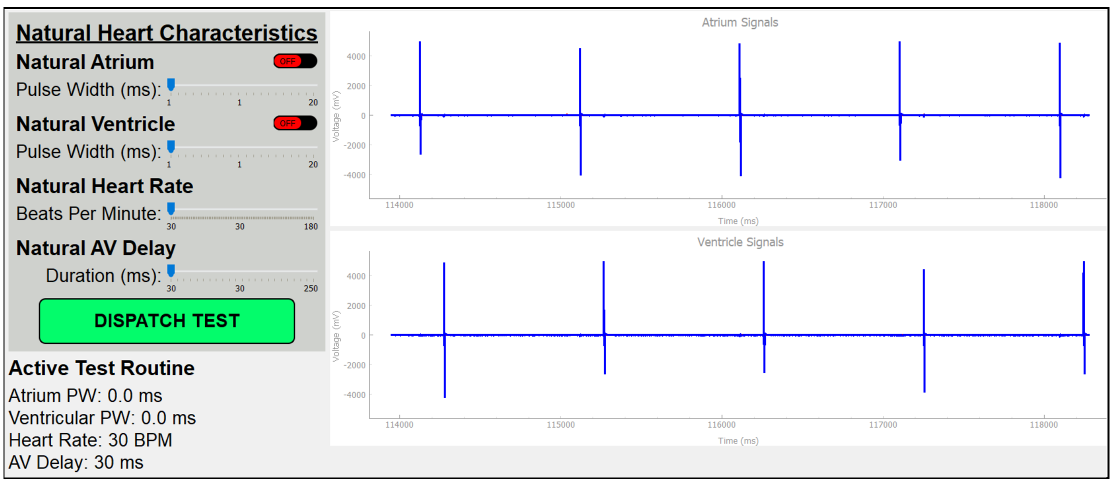
ACTUAL OUTPUT → RESULT: PASS

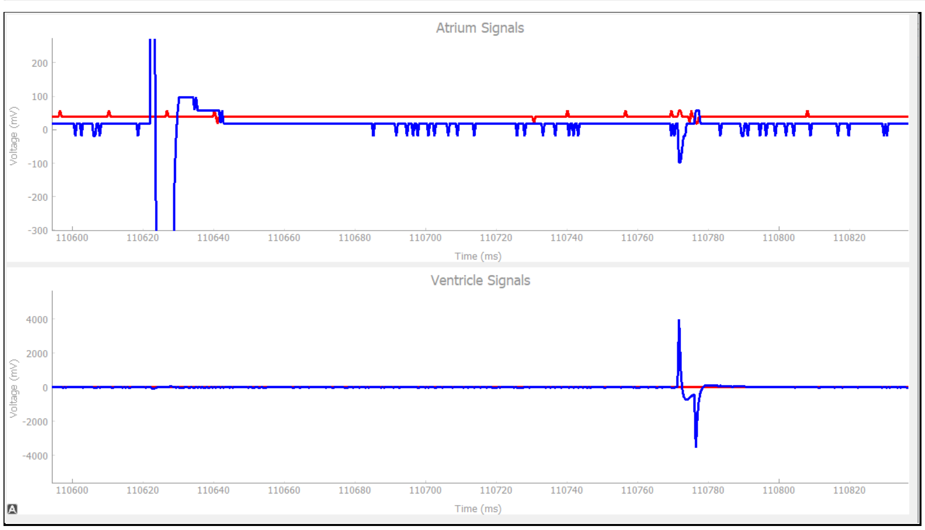




TEST CASE 2:	
PACEMAKER	**HEARTVIEW**
Lower Rate Limit (ppm): 60 Amplitude (V): 5 Pulse Width (ms): 20 AV Delay (ms): 100	Natural Atrium: OFF Natural Ventricle: OFF Natural Heart Rate (bpm): 30 Natural AV Delay (bpm): 30
EXPECTED OUTPUT	
The DOO mode works similarly as explained above. As seen in the output below the atrium and ventricle pulses are delayed by an AV delay of 100 ms which is shown in the second image.	

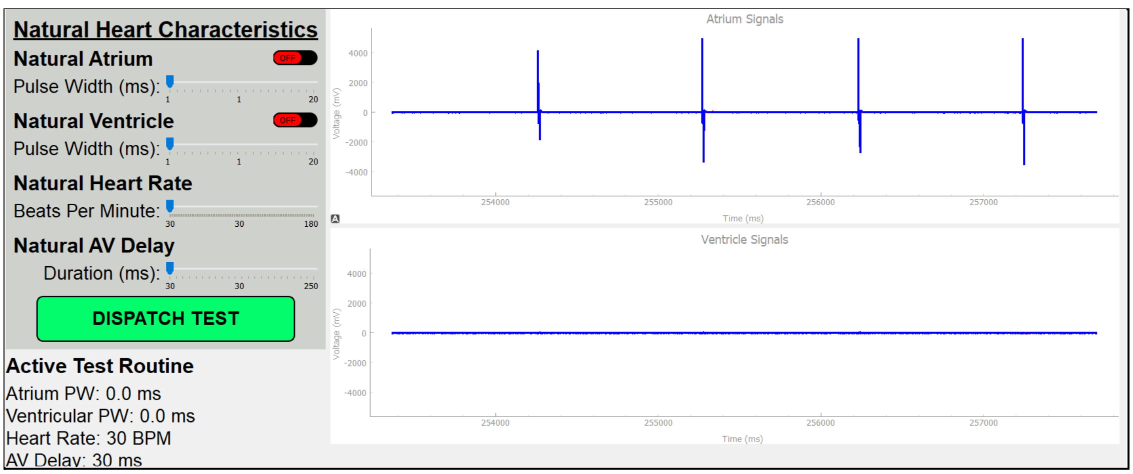
ACTUAL OUTPUT → RESULT: PASS

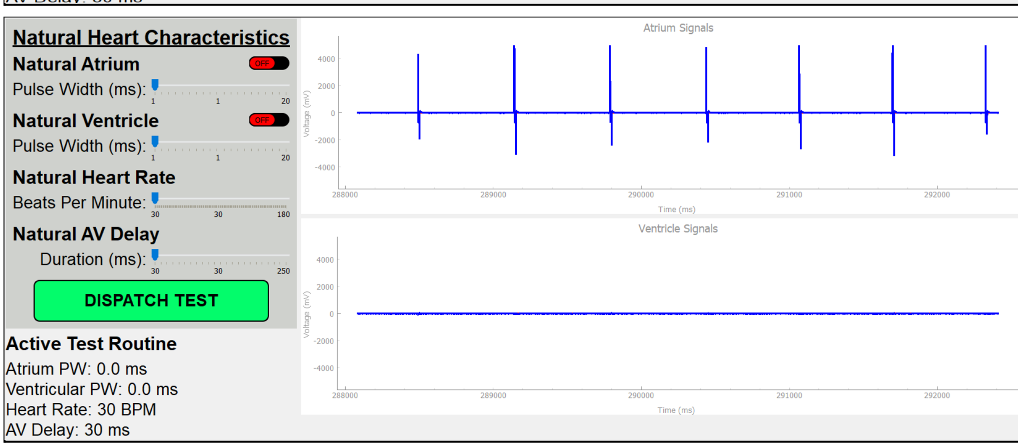




3.4 AOOR and VOOR Mode

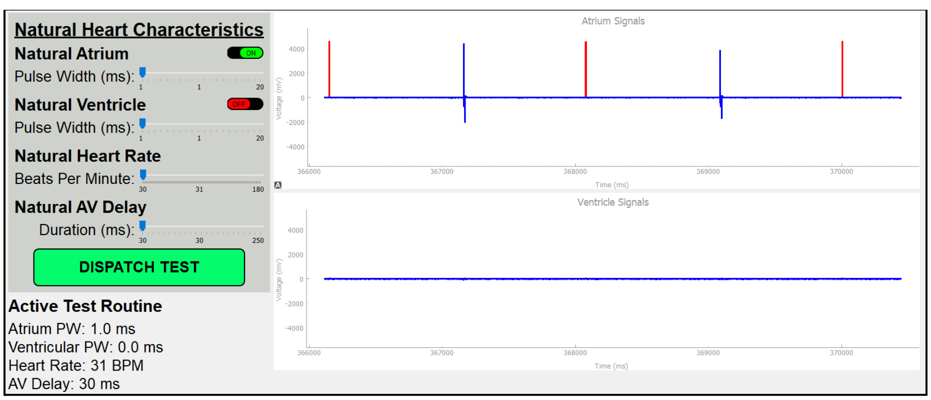
TEST CASE 1:	
PACEMAKER	**HEARTVIEW**
Lower Rate Limit (ppm): 60 Amplitude (V): 5 Pulse Width (ms): 20	Natural Atrium: OFF Natural Ventricle: OFF Natural Heart Rate (bpm): 30 Natural AV Delay (bpm): 30
EXPECTED OUTPUT	
In the AOOR/VOOR Mode, the pacemaker will be pacing at a rate based on the Lower Rate Limit regardless of the Natural Heart Rate. The pacing becomes faster as the pacemaker is being shaken. As seen below, the AOOR Mode functions correctly and meets the requirements for the mode. The VOOR mode functions similarly for the ventricle.	
ACTUAL OUTPUT → RESULT: PASS	

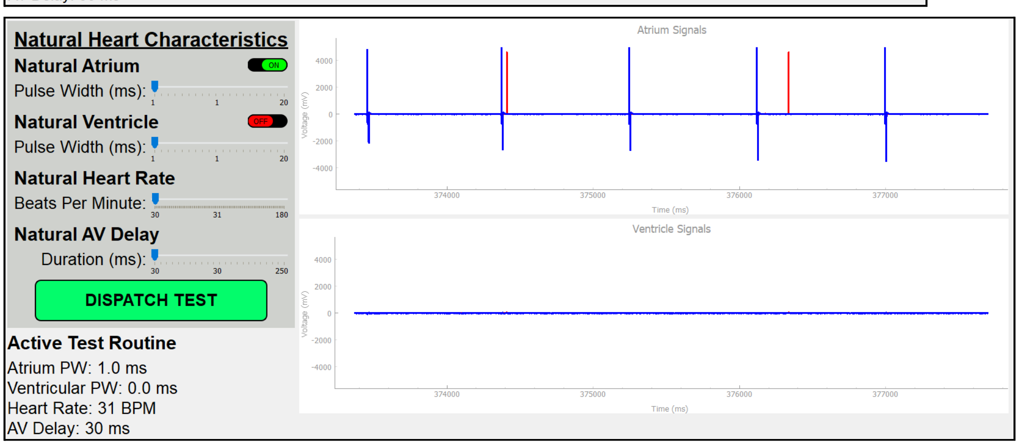




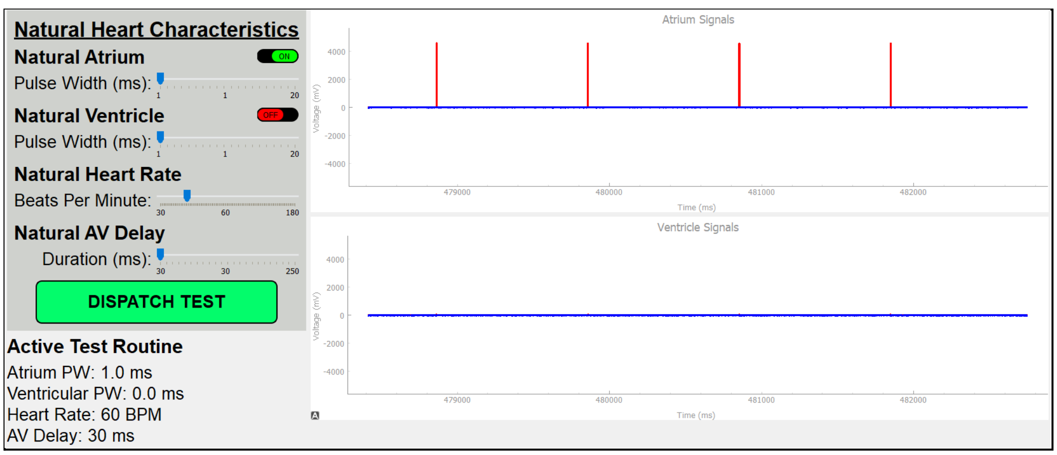
3.5 AAIR and VVIR Mode

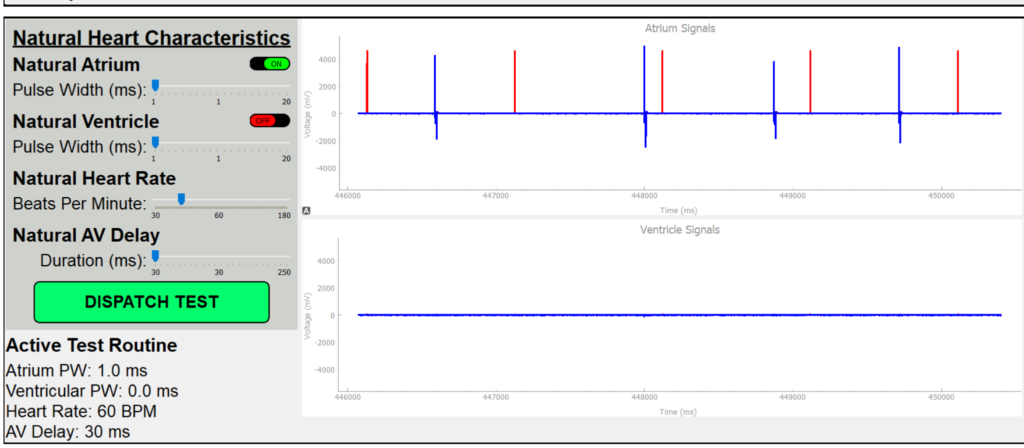
<table>
<tr><td colspan="2">TEST CASE 1:</td></tr>
<tr><td>PACEMAKER</td><td>HEARTVIEW</td></tr>
<tr><td>Lower Rate Limit (ppm): 30
Amplitude (V): 5
Pulse Width (ms): 20</td><td>Natural Atrium: ON
• Pulse Width (ms): 1
Natural Ventricle: OFF
Natural Heart Rate (bpm): 60
Natural AV Delay (bpm): 30</td></tr>
<tr><td colspan="2">EXPECTED OUTPUT</td></tr>
<tr><td colspan="2">In the AAIR/VVIR Mode, the pacemaker will be pacing at a rate based on the Lower Rate Limit and the Natural Heart Rate. The Pacemaker for the AAIR mode paces between each heart pulse as it inhibits the pace for natural heart rates lower than the lower rate limit. The pacing becomes faster as the pacemaker is being shaken. As seen below, the AAI Mode functions correctly and meets the requirements for the mode. The VVI mode functions for the ventricle.</td></tr>
<tr><td colspan="2">ACTUAL OUTPUT → RESULT: PASS</td></tr>
<tr><td colspan="2">





</td></tr>
</table>

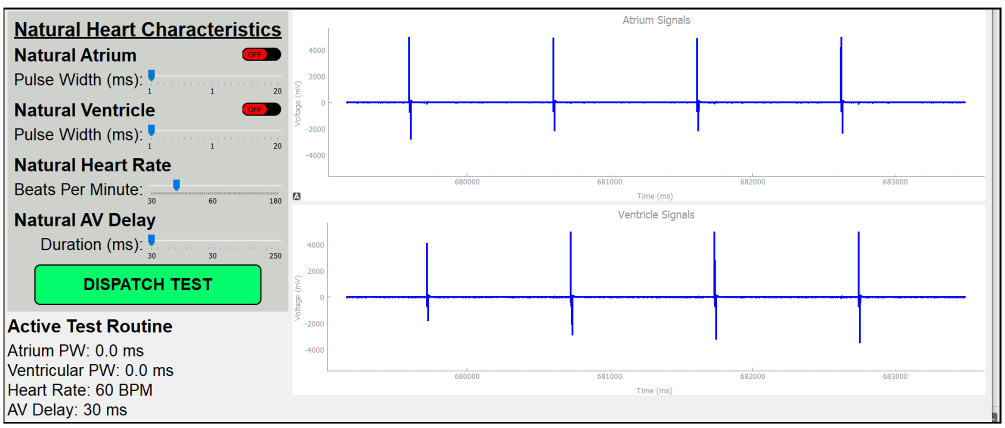
<table>
<tr><th colspan="2">TEST CASE 2:</th></tr>
<tr><th>PACEMAKER</th><th>HEARTVIEW</th></tr>
<tr><td>Lower Rate Limit (ppm): 60
Amplitude (V): 5
Pulse Width (ms): 20</td><td>Natural Atrium: ON
• Pulse Width (ms): 1
Natural Ventricle: OFF
Natural Heart Rate (bpm): 60
Natural AV Delay (bpm): 30</td></tr>
<tr><th colspan="2">EXPECTED OUTPUT</th></tr>
<tr><td colspan="2">The AAIR/VVIR mode works similarly as explained above. Since the natural heart rate is equal to the lower rate limit, the pacemaker does not send a pace. Once activity is detected the pacing appears and becomes faster as the pacemaker is being shaken. As seen below, the AAIR Mode functions correctly and meets the requirements for the mode. The VVIR mode functions for the ventricle.</td></tr>
<tr><th colspan="2">ACTUAL OUTPUT → RESULT: PASS</th></tr>
<tr><td colspan="2">




</td></tr>
</table>

3.6 DOOR Mode

TEST CASE 1:	
PACEMAKER	**HEARTVIEW**
Lower Rate Limit (ppm): 60 Amplitude (V): 5 Pulse Width (ms): 20 AV Delay (ms) : 60 ms	Natural Atrium: OFF Natural Ventricle: OFF Natural Heart Rate (bpm): 60 Natural AV Delay (bpm): 30
EXPECTED OUTPUT	
The DOOR mode is the Dual Pacing Mode which identifies the atrium and ventricle being paced. The DOOR Mode is a combination of the AOOR and VOOR mode and uses the AV delay to determine the next pulse and changes the amplitude. As seen in the output below, the atrium and ventricle pulses are delayed by 60ms. Once activity is detected the pacing becomes faster as the pacemaker is being shaken.	
ACTUAL OUTPUT → RESULT: PASS	



4.0 References

Meyer, G., 2020. *Functionality and Circuitry Explanation: Pacemaker Microcontroller Shield*. Hamilton: McMaster University, pp.4-6.

2007. *Pacemaker System Specification*. Boston Scientific.

McMaster University. (n.d.). *Pacing Codes and Modes Concepts* [PPT]. Hamilton.