SE 3KO4 - L01 G7

Assignment 2 - Pacemaker Software

Carlos Capili
Raeed Hassan
Shaqeeb Momen
Aaron Pinto
Udeep Shah

Table of Contents

1.0 Requirements

1.1 Current Requirements

The current requirements for the pacemaker software are to properly implement the behaviours for the standard AOO, VOO, AAI and VVI modes for pacemakers, given a set of user parameters by case. The specific behaviour of each mode is provided in the following sections. A table of programmable parameters can also be found below.

User Parameter	Units	Allowed values (nominal)	Description
Mode	---	AOO ,VOO, AAI, VVI	Indicates which operating mode the pacemaker should operate in
Lower Rate Limit	ppm	30 - 175ppm (60ppm)	The lowest rate that the pacemaker should force the heart to achieve. Also acts as the fixed heart rate for non-adaptive modes
Upper Rate Limit	ppm	30 - 175ppm (60ppm)	The highest rate that the pacemaker should force the heart to achieve.
Pulse Amplitude	V	0.5 - 5V (3.5V)	The amplitude of the pulse sent to the heart when pacing either the atrium or ventricle
Pulse Width	ms	0.05 - 1.9 ms (0.1 ms)	The width of the pulse sent to the heart when pacing either the atrium or ventricle
ARP	ms	150 - 500 ms (250 ms)	The period that the pacemaker will ignore sensed inputs, after a pacing or sensing a pace from the atrium
VRP	ms	150 - 500 ms	The period that the pacemaker will ignore

		(350 ms)	sensed ventricular pulses, after a pacing or sensing a pace from the ventricle
PVARP	ms	150-500 ms (250ms)	The period that the pacemaker will ignore sensed atrial paces after a ventricular pulse (used for dual sensing)
Fixed AV Delay	ms	70-300 ms (150ms)	The delay between pulsing the ventricle after an atrial pulse (used for dual pacing)
Maximum Sensor Rate	ppm	50-175 ppm (120 ppm)	The maximum rate that the accelerometer can set the pacemaker to operate at
Reaction Time	s	10-50 s (30 s)	The time required to drive the active sensor rate to the maximum sensor rate
Response Factor	---	1-16 (8)	A factor to determine the increment of change in the active sensor rate
Recovery Time	min	2-16 min (5 min)	The time required for the sensor rate to fall from maximum to the lower limit
Activity Threshold	---	V-Low, Low, Med-Low, Med, Med-High, High, V-High (Med)	A threshold to determine the level of activity required to begin changing the active rate based on accelerometer

Table 1.1: Current User Programmable Parameters

1.1.1 AOO Requirements

The AOO mode paces the atrium at a fixed interval regardless of natural heart behaviour. The rate at which pacing occurs is determined by the Lower Rate Limit. Pulse characteristics are determined by the Pulse Amplitude and Width. ARP and VRP are unused for this mode.

Condition	**Action**
60 000/(Lower Rate Limit) time has passed since last pacemaker atrial pulse	- Pulse the atrium - Reset interval timer

Table 1.1.1: Conditional Responses for AOO mode

1.1.2 VOO Requirements

The VOO mode paces the ventricle at a fixed interval regardless of natural heart behaviour. The rate at which pacing occurs is determined by the Lower Rate Limit. Pulse characteristics are determined by the Pulse Amplitude and Width. ARP and VRP are unused for this mode.

Condition	**Action**
60 000/(Lower Rate Limit) time has passed since last pacemaker ventricular pulse	- Pulse the ventricle - Reset interval timer

Table 1.1.2: Conditional Responses for AOO mode

1.1.3 AAI Requirements

The AAI mode will pace the atrium to meet the Lower Rate Limit parameter if the atrium is not naturally doing so on its own.

Condition	**Action**
60 000/(Lower Rate Limit) time has passed since last pacemaker or natural atrial pulse	- Pulse the atrium - Reset interval timer
ARP time has passed between last natural or pacemaker pulse & atrium pulses naturally	- Reset interval timer

Table 1.1.3: Conditional Responses for AAI mode

1.1.4 VVI Requirements

The VVI mode will pace the ventricle to meet the Lower Rate Limit parameter if the atrium is not naturally doing so on its own.

Condition	**Action**
60 000/(Lower Rate Limit) time has passed since last pacemaker or natural ventricular pulse	- Pulse the ventricle - Reset interval timer
VRP time has passed between last natural or pacemaker pulse & ventricle pulses naturally	- Reset interval timer

Table 1.1.4: Conditional Responses for VVI mode

1.2 Expected Changes to Requirements

In future several changes are expected to be made to our requirements, more specifically

- Adding support for full dual sensing, pacing and adaptive operation (DDDR)

1.2.1 DDDR Mode

This mode of operation is meant to emulate the pacemaker operating dual pacing and sensing operations while also using an adaptive rate. The current architecture was made in a manner to eventually support this mode by putting junctions ahead of sense state exits and new transitions exclusive to the dual operation. These exclusive transitions would bypass the normal full loop reset to force both atrial and ventricular operation to occur in a single loop. This technique has already been briefly utilized to support DOO and DOOR modes of operation.

2.0 Software Model

2.1 Architecture

The overall architecture of the pacemaker's software is composed of 4 main modules: Parameters, MainStateFlow, HardwareOutput, and Hardware input. Parameters stores and transforms any programmable parameters from the user or any software interfaces into relevant values for the main stateflow to manipulate. The MainStateFlow is the module that hosts all the logic for pacemaker decision making. This module itself does not affect any hardware, it simply reads state from input modules and outputs variable state to output modules. The modules are all loosely coupled and can be interchanged as long as the expected interface variables are properly passed between, there is no access to internals from one module to its neighbours. High Cohesion is achieved as all code written inside the module purely acts towards the purpose of that module. As an example, the MainStateFlow simply sets variables for pin states as true or false, but does not implement the block that would actually cause the board's pin to change, just the data that represents that pin state.

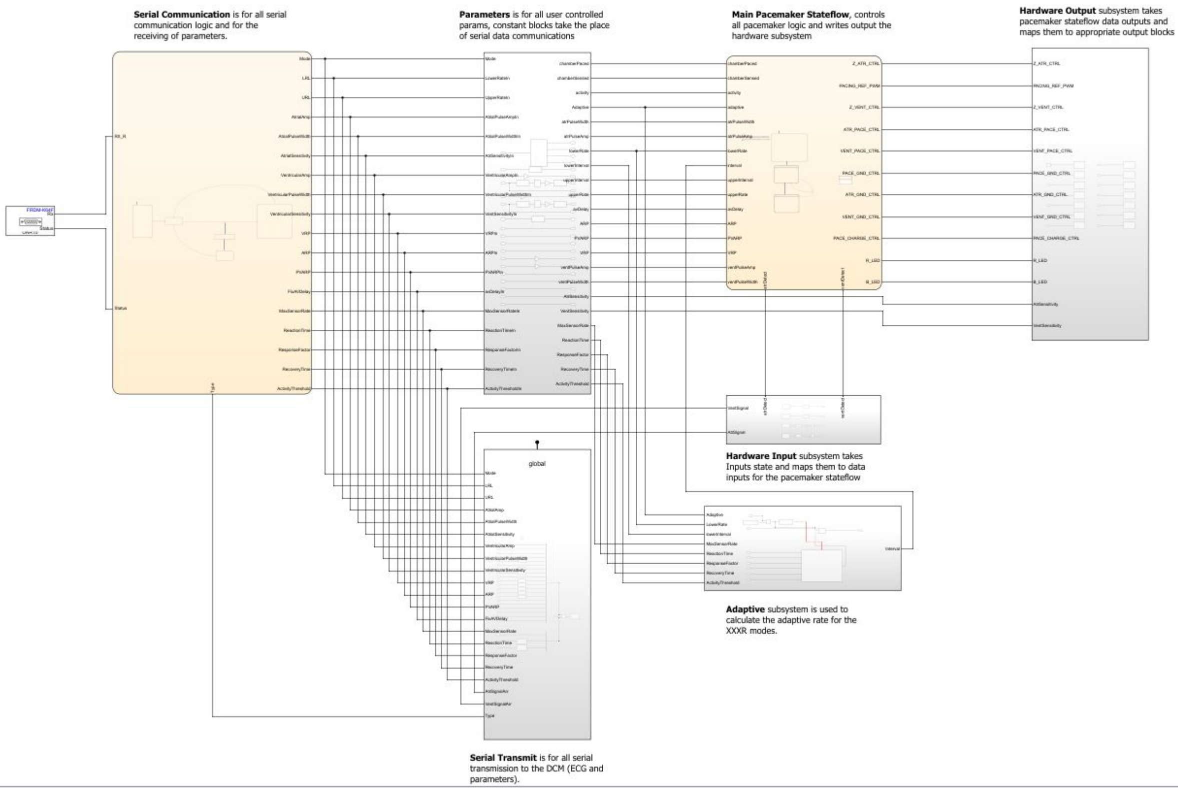


Figure 2.1: Top Level Pacemaker Architecture

2.1.1 Parameter Module

This module is a subsystem that has essential conversions and transformations on input parameters and outputs those to the main pacemaker stateflow. This module takes in data which is received from the DCM through serial communication. The reason this was done was to separate the allocation of variables used in the main stateflow from the logical aspect of the main stateflow.

Inputs Param	Transformations	Outputs
Mode	Function to split the mode up into the 4 letter code.	ChamberPaced
		chamberSensed
		activity
		Adaptive

AtrialpulseWidthIn	----	atrPulseWidth
AtrialpulseAmpIn	*100	atrpulseAmp
lowerRateIn	----	lowerRate
	60 000/lowerRateIn	lowerInterval
VentricularpulseWidthIn	----	ventPulseWidth
upperRateIn	----	upperRate
	60 000/upperRateIn	upperInterval
VentricularpulseAmpIn	*100	ventPulseAmp
avDelayIn	----	avDelay
ARPIn	----	ARP
PVARPIn	----	PVARP
VRPIn	----	VRP
ArtSensitivityIn	*20	ArtSensitivity
VentSensitivityIn	*20	VentSensitivity
MaximumSensorRateIn	----	MaximumSensorRate
ReactionTimeIn	----	ReactionTime
ResponseFactorIn	----	ResponseFactor
RecoveryTimeIn	----	RecoveryTime
ActivityThresholdIn	----	ActivityThreshold

Table 2.1.1 - Parameters module inputs and outputs

2.1.2 Main State Flow

The Main State Flow module hosts all the decision making logic for the pacemaker. It does not apply any hardware changes, it purely expects an input of what the pin states are as data from an external module and outputs other pin states as variables to another external module. At the highest level, this stateflow has 3 main states: Init, Sense and Pace.

In order to support dual-pacing (DOO,DOOR) modes, a junction was added after the event transition for PaceEnd to direct the model to enter the Sense.Ventricle state directly, instead of restarting the whole sense loop. This results in both the atrium and ventricle being paced in a single code execution loop. The new timing is handled inside the relevant Sense child states.

Source State	Events	Conditions	During Actions	Destination State
State = Init	NC	NC	INIT Z_ATR_CTRL = false; Z_VENT_CTRL = false; PACING_REF_PWM = (pulseAmp/5); ATR_PACE_CTRL = false; VENT_PACE_CTRL = false; PACE_GND_CTRL = true; ATR_GND_CTRL = false; VENT_GND_CTRL = false; PACE_CHARGE_CTRL = true;	Sense

			R_LED = false; B_LED = false;	
State = Sense	paceStart	See Table 2.1.2.1 for sense stateflow	See Table 2.1.2.1 for sense stateflow	Pace
State = Pace	paceEnd	None	See Table 2.1.2.1 for pace stateflow	Sense
		dualSense(chamberPaced, chamberSensed) && ActiveChamber == 1		Sense.Ventricle

Table 2.1.2: High Level Main Stateflow

2.1.2.1 Init

The Init state is code meant to run only on bootup of the pacemaker hardware. Its purpose is to set all the initial pin state values and then immediately enter the Sense state

2.1.2.2 Sense

The Sense stateflow is meant to handle the implementation of sensing and activity behaviours, as per set operating mode. It consists of 3 main states: Default, Atrium and Ventricle.

Default acts purely as a placeholder for any code that should be run at the beginning of each sense-activity loop and also as a decision junction to switch between the Atrium and Ventricle state charts. As is, this could be replaced with just a junction, but it was anticipated that we would need a place to run code at the beginning of each sense-activity loop, so it was left in.

Atrium/Ventricle consists of 4 possible child states: Default, Triggered, Inhibited and None. The first is similar to the default of Sense chart, but this has code in it to set the ActiveChamber to the entirety of the MainStateChart. In anticipation of having to handle dual pacing and sensing modes it was critical we have a variable to track which chamber to interact with, as it changes from atrium to ventricle in the same operation loop. The Pace stateflow also uses ActiveChamber to determine the appropriate hardware state to change. The other three states of Triggered, Inhibited and None, run the logic to implement the correct behaviour as per standard pacemaker operating modes and fire the paceStart event whenever a pace should be sent. Currently Triggered is left empty in Atrium and is not present in Ventricle, as implementing any triggered operation is not part of the present requirements.

For this stage of development, support for dual pacing was added. Due to the structure of the previous model, this mode is supported through a series of mode-specific transitions. These transitions check for different timing requirements than the standard ones to compensate for two pulses being fired in a single pulse-execution loop. A function, dualSense(a,b) was also introduced to quickly check the mode parameters for the relevant dual operation settings.

Source State	Events	Conditions	During Actions	Destination State
State = Default	False	atrSense(chamberPaced, chamberSensed) ==1	NC	Atrium
		ventSense(chamberPaced, chamberSensed) ==1	NC	Ventricle
State =	False	See Table 2.1.2.2 for Atrium	See Table	Default

Atrium		stateflow details	2.1.2.2 for Atrium Stateflow details	Ventricle
				Pace
State = Ventricle	False	See Table 2.1.2.2 for Ventricle stateflow details	See Table 2.1.2.2 for Ventricle stateflow details	Default
				Pace

Table 2.1.2.1: Sense Stateflow

Source State	Events	Conditions	During Actions	Destination State
State = Default	False	activity == Activity.DUAL_A /DUAL_V	NC	Triggered
		activity == Activity.INHIBITED_A/INHIBITED_V		Inhibited
		activity == Activity.NONE_A/ NONE_V		None
State = Triggered (Atrium Only)	False	NC	NC	NC
State = Inhibited	False	atrDetect == true	See Section 2.2.2	Sense.Default

		after(interval,msec)		Pace
State = Atrium.None	False	after(interval,msec)	See Section 2.2.1	Pace
		after((interval - avDelay - atrPulseWidth - ventPulseWidth), msec) && dualSense(chamberPaced, chamberSensed)		
State = Ventricle.None	False	after(interval,msec)		
		after(avDelay, msec) && dualSense(chamberPaced, chamberSensed)		

Table 2.1.2.2: Atrium/Ventricle Stateflow Details

To deal with future dual-pacing/sensing requirements (DDDR), this model architecture supports adding transition junctions when exiting the atrial inhibit/trigger states and directly entering the appropriate ventricle activity states. Currently they are removed for model clarity.

2.1.2.3 Pace

See section 2.2.3 Pacing for detailed information on the operation of the Pace stateflow. In short, the Pace stateflow is triggered by the paceStart event, fired whenever Sense wants to trigger a hardware change. Pace does not look at any variables or data from sense itself, it only uses the from the high state flow level.

2.1.3 Hardware Inputs

This subsystem is used to receive digital reads from the hardware and send them as an input to the main stateflow. This was done to implement separation of concerns and separate the receiving of inputs from the hardware from the logic in the main stateflow using those inputs. For a further description of inputs and outputs of this subsystem refer to the Hardware hiding document.

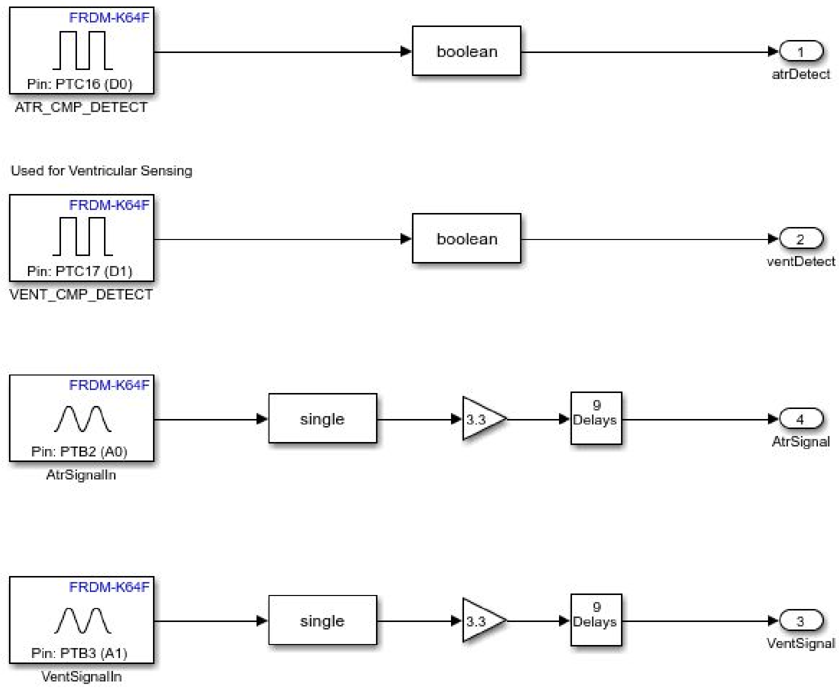


Figure 2.1.3 - Hardware Inputs Subsystem

2.1.4 Hardware Outputs

This subsystem is used to output digital writes to the hardware with values obtained from the main stateflow. This way, the software does not directly interact with the hardware output pins. For a more detailed description refer to the Hardware Hiding document.

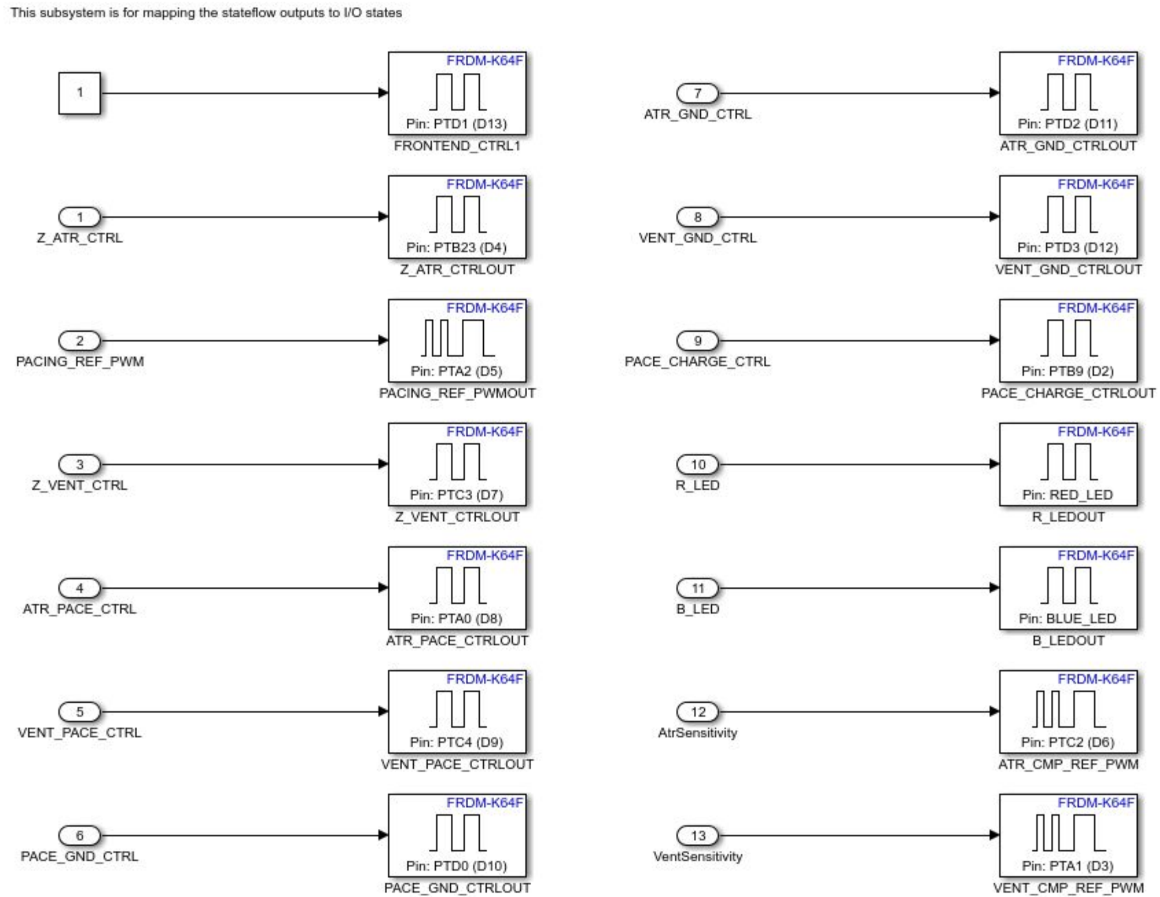


Figure 2.1.4 - Hardware Outputs Subsystem

2.1.5 Serial Communication

Serial Communication is one of the new modules that was added into the model after the first assignment. It was implemented to help the model communicate with the DCM. It is split into two different parts, one for all the receiving and logical operations of the Serial Communication, another is for all the transmission of serial data to the DCM.

2.1.5.1 Serial Communication Subsystem

This subsystem is used to receive serial communication from the DCM and then runs some logic on that received data. It is also used to set the defaults in the case that the DCM is not plugged. The received data is passed on to both the serial transmit subsystem and the parameters subsystem where it is further passed on to the main stateflow.

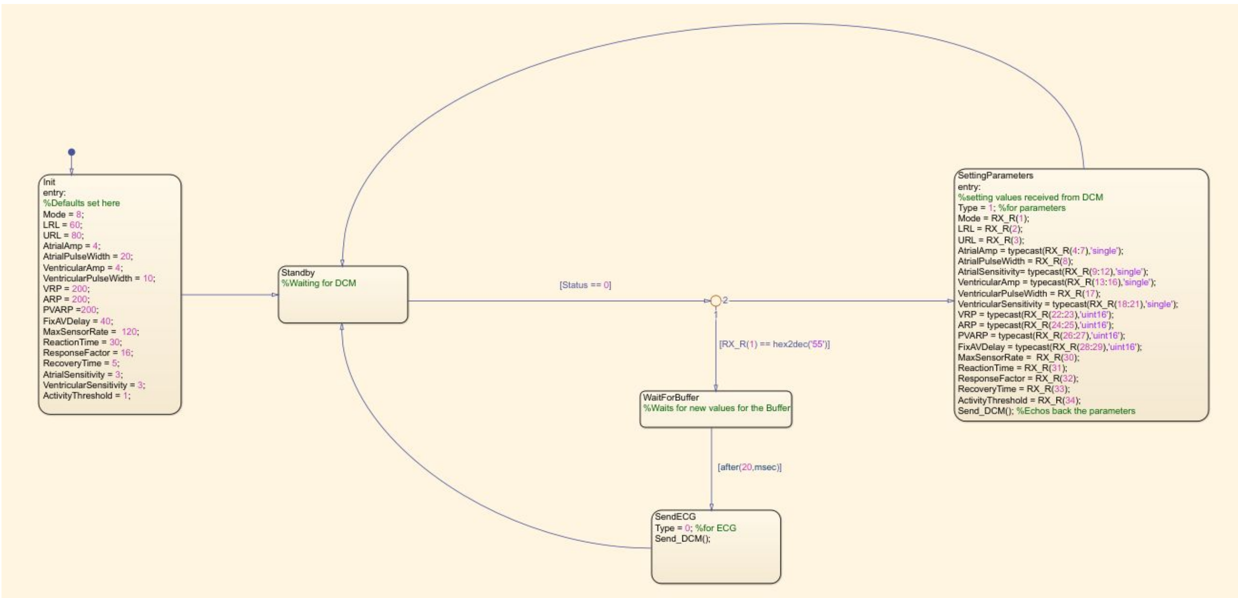


Figure 2.1.5.1 - Serial Communication Subsystem

For efficient serial communication, data types for different variables were needed to be allocated as efficiently as possibly. Most of the data types are uint8. If the range of the variable extends 255, it was given uint16. If the range of values included decimals then the single data type was chosen. The Complete list of all the variables in the received order is given below.

PARAM NAME	**DATA TYPE**
Mode (enumerated)	u_int_8 (1) -> 0:AOO, 1:AAI, 2:AOOR, 3: AAIR, 4:VOO, 5:VVI, 6:VOOR, 7:VVIR 8:DOO, 9:DOOR
Lower Rate Limit	u_int_8 (1)
Upper Rate Limit	u_int_8 (1)
Atrial Amplitude	single (4)

Atrial Pulse Width	u_int_8 (1)
Atrial Sensitivity	single (4)
Ventricular Amplitude	single (4)
Ventricular Pulse Width	u_int_8 (1)
Ventricular Sensitivity	single (4)
VRP	u_int_16 (2)
ARP	u_int_16 (2)
PVARP	u_int_16 (2)
Fixed AV Delay	u_int_16 (2)
Maximum Sensor Rate	u_int_8 (1)
Reaction Time	u_int_8 (1)
Response Factor	u_int_8 (1)
Recovery Time	u_int_8 (1)
Activity Threshold	u_int_8 (1) -> 0:V-Low, 1:Low, 2:Med-Low, 3: Med, 4:Med-High, 5:High, 6:V-High

Table 2.1.5.1 - Received serial communication variables

2.1.5.2 Serial Transmit Subsystem

This subsystem is used to transmit both the ECG data and the echo of the parameters to the DCM. This is a function call subsystem and does not run constantly and only runs when it is needed to. This subsystem switches between sending the parameters and the ECG data depending on what is requested by the DCM. All the variables either for the parameters or the ECG that have more than a byte are packed. A constant was added before both the parameter data and the ECG data to classify which of the two data is being sent.

For the parameters, the same variables are sent from table 2.1.5.1 in the same order. An additional padding is added to the end of this to make the total number of bytes equal to that of the ECG data. This is done as the switch needs to have arrays of the same size. The ECG data is directly received from the hardware inputs subsystem.

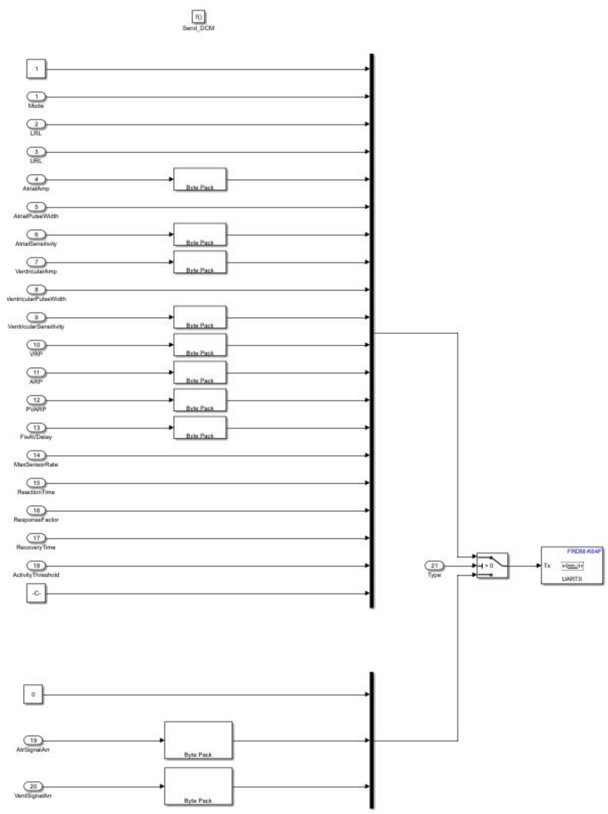


Figure 2.1.5.2 - Serial Transmit Subsystem

2.1.6 Rate Adaptive Subsystem

This subsystem tracks activity using the on-board accelerometer and changes the pacing rate according to the amount of activity sensed. In Figure 2.1.6.1 below, preprocessing is done to determine the sensor indicated rate, the rate of increase, rate of decrease and whether the activity sensed exceeds the activity threshold.

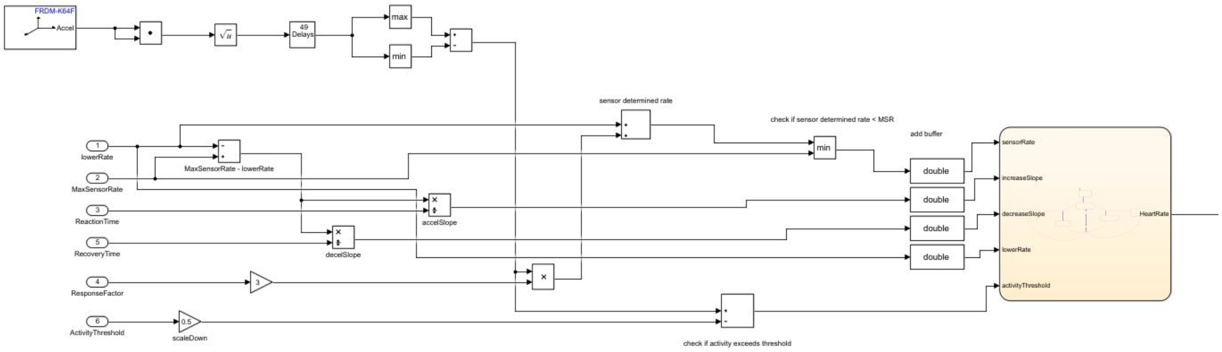


Figure 2.1.6.1 - Preprocessing inputs for Rate Adaptive stateflow

2.1.6.1 Determining Activity Level

To determine the activity level, the magnitude of the raw data from the accelerometer is taken and fed into a buffer This is to ensure each data from the accelerometer is processed and previous data points can be compared.

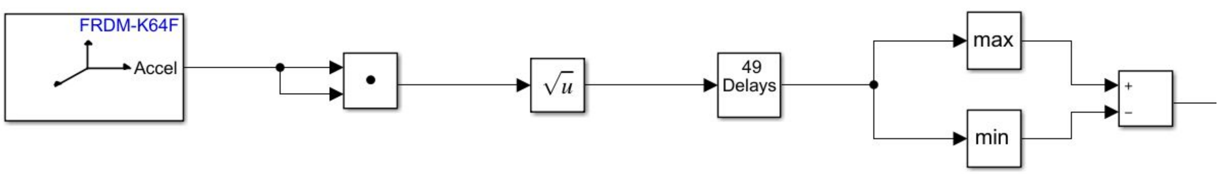


Figure 2.1.6.2 - Activity Level Preprocessing

The delay block acts as a buffer in which it holds all the previous data points . Using this method, a range of accelerometer data is processed and by taking the max and min values the activity level can be determined. For example, if the board is not shaken, the values in the buffer would remain the same, and taking the difference between the max and min values would equal to 0 representing no activity.

2.1.6.2 Rate of Increase and Rate of Decrease

The rate of increase corresponds to the rate at which the LRL will increase to the Sensor Rate. The rate of decrease is the rate at which the Sensor Rate will decrease back to the LRL. Both of these rates increase the pace rate linearly. To determine the rate of increase from the LRL, the difference between the MSR and LowerRate was taken and divided by the reaction time. The same was done for rate of decrease but with the recovery time.

2.1.6.3 Rate Adaptive Stateflow

The rate adaptive logic checks if the activity level exceeds or does not exceed the activity threshold. Then it will go to the respective state where it will increase or decrease the pacing rate linearly over a period of time until the sensor rate is achieved or the rate has returned to the LRL.

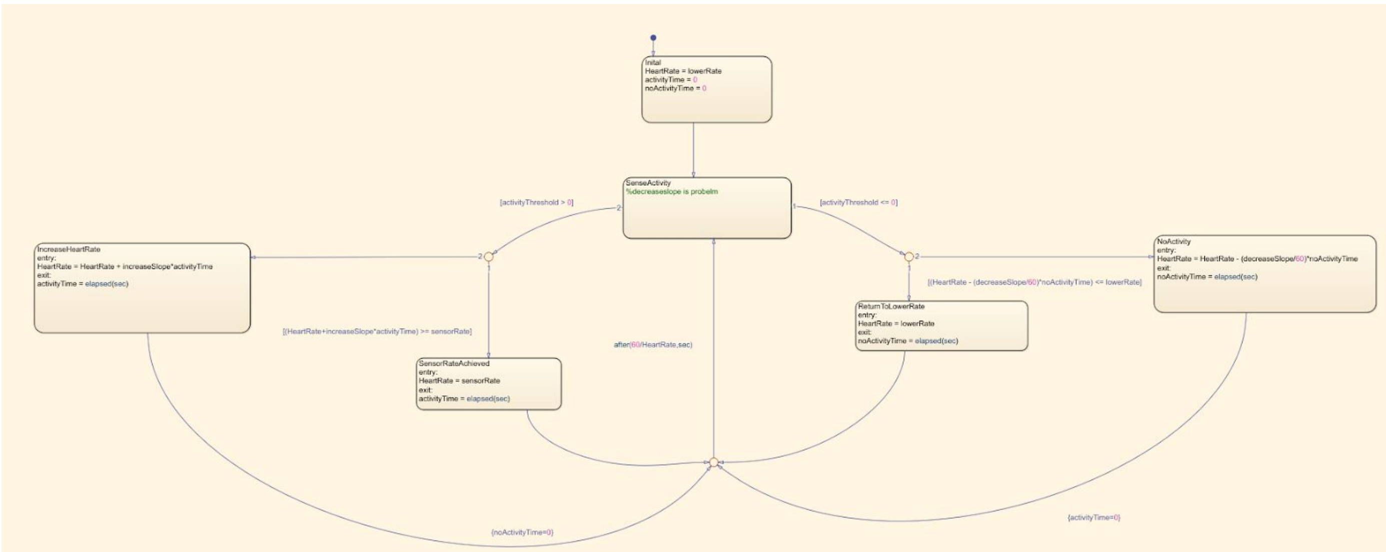


Figure 2.1.6.3 - Rate Adaptive logic

Source State	Events	Conditions	Actions	Condition Actions	Transition Actions	Destination State
Initial	NC	NONE	HeartRate = lowerRate	NC	NC	SenseActivity
State = SenseAct ivity t	NC	activityThresho ld <= 0	NC	NC	NC	NoActivity, ReturnToLowe rRate
		activityThresho ld > 0	NC	NC	NC	IncreaseHeart Rate, SensorRateAch ieved
Increase HeartRat e	NC	NONE	HeartRate = HeartRate + increaseSlope*activity Time activityTime = elapsed(sec)	NC	noActivityTim e = 0	SenseActivity
SensorRa teAchiev ed	NC	NONE	HeartRate = sensorRate activityTime = elapsed(sec)	NC	NC	SenseActivity
NoActivit y	NC	NONE	HeartRate = HeartRate - (decrease/60)*noActivi	NC	activityTime = 0	SenseActivity

			tyTime noActivityTime = elapsed(sec)			
ReturnToLowerRate	NC	NONE	HeartRate = lowerRate	NC	NC	SenseActivity

Table 2.1.6.1 Rate Adaptive Tabular Form

2.2 Individual State Description

2.2.1 AOO, VOO and DOO Timing Logic

These two pacing modes are ones which require no sensing or response to sensing. It will pace the appropriate chamber asynchronously at the lower programmed pacing rate.

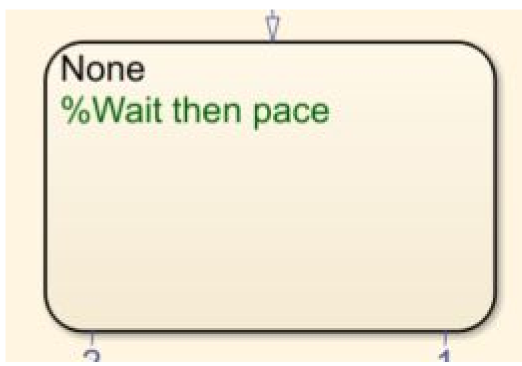


Figure 2.2.1 - Logic for AOO/VOO/DOO states

In the figure above, the state defined as *None* can be found in the atrium and ventricle states inside the Sensing state and represents the logic for AOO,VOO and DOO pacing modes. Depending on whether dual operation is active, the model will remain in this state and idle and transition out after the full interval time (single pacing) or after a short time to compensate for the rest of the pace-loop execution (dual-pacing). See table 2.1.2.2 for full details on transition logic.

2.2.2 AAI and VVI Logic

The logic for the sensing for AAI and VVI are both similar other than the pins it is detecting from.

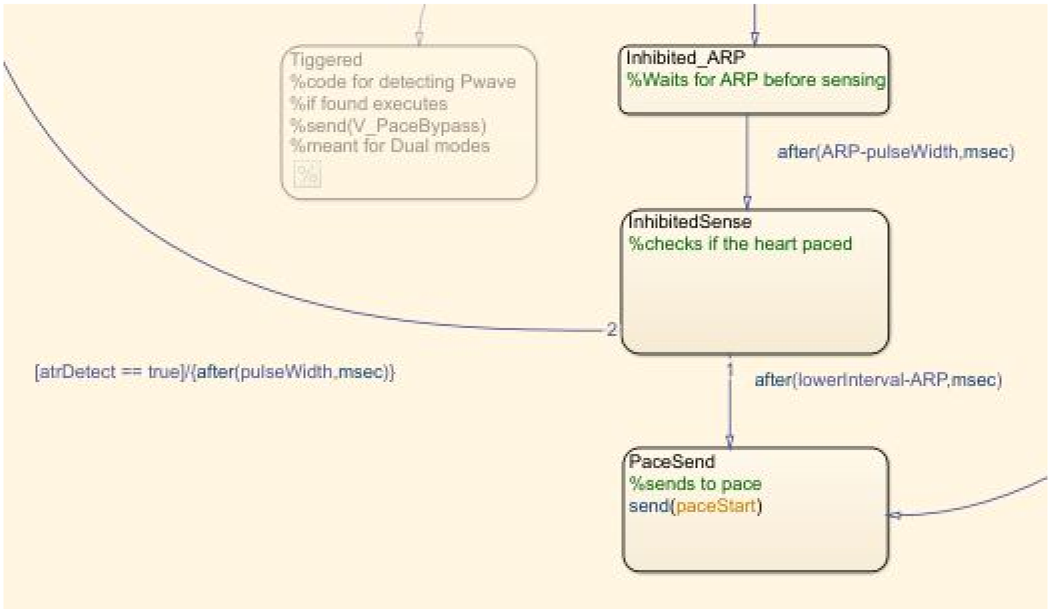


Figure 2.2.2 - Inhibition logic for atrium

Figure 2.2.2 gives a snapshot of the code that is in the Inhibited stateflow for the Atrium. For the Ventricle, atrDetect is replaced with ventDetect. First it waits for the ARP regardless of what events took place before that. After that the logic is based on which of the if statement's condition returns a 1 first. This is elaborated in the table below.

Condition	Event	Description
after(interval-ARP,msec)	Validates transition	Sends it into PaceSend state where a pace is sent into the heart.
(atrDetect/ventDetect == true)	Validates transition	Sends it back to the default in the Sense state.

Table 2.2.2 - Tabular expression for the inhibition logic for atrium

In the above table interval is the active interval, which uses the value set for the lower limit when non-adaptive or the appropriately calculated value for adaptive modes. AtrDetect gives whether the Atrium has a natural pace, this is received from the subsystem Hardware Inputs.

The initial design for the first assignment did not have ARP or VRP. This was corrected for this assignment. Also the logic was changed from hardcoded if statements to transitions with conditions.

2.2.3 Pacing

The Pace state consists of two sub-states called *Discharging* and *Charging*, an event called *chargePace* to transition from Discharging to Charging, and an event called *paceEnd* to exit the Pace state. To enter the Pace state, event *paceStart* must be sent. The sole purpose of this state is to pace the specified chamber regardless of pacing mode. For example, when AOO mode is selected, it will go through this state to pulse the atrium and immediately exits. This is also true for all pacing modes.

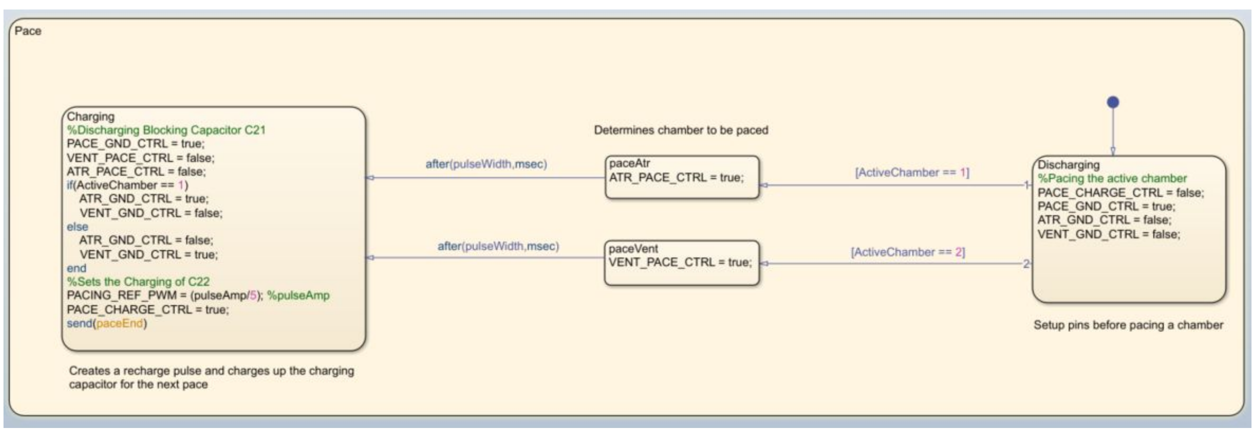


Figure 2.2.3 Pace State seen on Simulink

Source State	Events	Conditions	During Actions	Condition Actions	Transition Actions	Destination State
State = Discharging	NC	ActiveChamber == 1	PACE_CHARGE_CTRL = false PACE_GND_CTRL =	NC	NC	paceAtr

			true
			ATR_GND_CTRL = false
			VENT_GND_CTRL = false
			if(ActiveChamber == 1)
			ATR_PACE_CTRL = true
			else
			VENT_PACE_CTRL = true
			if after(pulseWidth,msec)
			SEND chargePace			

		ActiveChamber == 2		NC	NC	paceVent
State = paceAtr	NC	after(pulseWidth, msec)	ATR_PACE_CTRL = true	NC	NC	Charging
State = paceVent			VENT_PACE_CTRL = true	NC	NC	Charging

State = Charging	paceEnd	NONE	PACE_GND_CTRL = true;
			VENT_PACE_CTRL = false;
			ATR_PACE_CTRL = false;
			if(ActiveChamber == 1)
			ATR_GND_CTRL = true;
			VENT_GND_CTRL = false;
			else
			ATR_GND_CTRL = false;
			VENT_GND_CTRL = true;
			end
			%Sets the Charging of C22
			PACING_REF_PWM = (pulseAmp/5);
				NC	CALL paceEnd	Sense

			%pulseAmp PACE_CHARGE_CTRL = true; send(paceEnd)			

Table 2.2.3 Pace State Tabular Form

The function of the Discharging state is to drain the charging capacitor (C22) and pace the appropriate chamber of the heart. This is determined through the if statement by checking activeChamber. The activeChamber result triggers one of two transitions for states: paceAtr or paceVent that trigger the pins for discharging. After the relevant pulse width time has expired (set by the previous sense child state) the model will transition into the Charging state after the specified pulseWidth programmable parameter has been satisfied by sending the chargePace event.

The function of the Charging state is to ground the blocking capacitor (C21) and recharge the charging capacitor. Depending on the value of ActiveChamber the appropriate pin for the chamber will be set to an active low for the blocking capacitor to be grounded and then the charging capacitor will begin to charge. Finally, an event called paceEnd will be sent indicating a pace has successfully been produced.

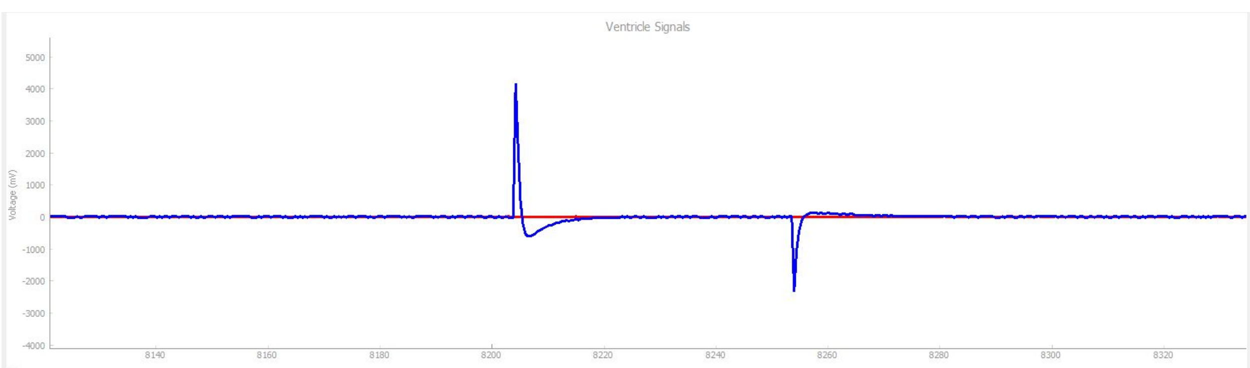


Figure 2.2.3 - Single generated pulse

The figure above represents the general shape of a single pulse. The controlling parameters of the pace pulse are the pulse amplitude and pulse width. Pulse amplitude represents the upward pulse. Changing this parameter in Simulink will affect the height of the pulse. The space between the upward pulse and the downward pulse is the pulse width. Changing this parameter will result in increasing or decreasing the distance between the two pulses.

3.0 Testing

3.1 AAI and VVI

As the logic for both AAI and VVI are both similar, the test cases for both yield similar results. There will be a total of 4 cases. This is not definitely testing every possible case but it does try to test basic and extreme conditions.

Test case I

Pacemaker Settings		HeartView Settings	
Mode	AAI	Natural Atrium Pulse Width (ms)	On 5
Pulse Width (ms)	10	Natural Ventricle	Off
Pulse Amplitude (V)	4	Natural Heart Rate (bpm)	30
Lower Rate (ppm)	60	Natural AV Delay (ms)	30

Expected Out	Actual Out
The pacemaker paces in between the heart pulses and inhibits the paces when the heart paces.	The result is as expected.
Test result - Pass	
Changes - None	

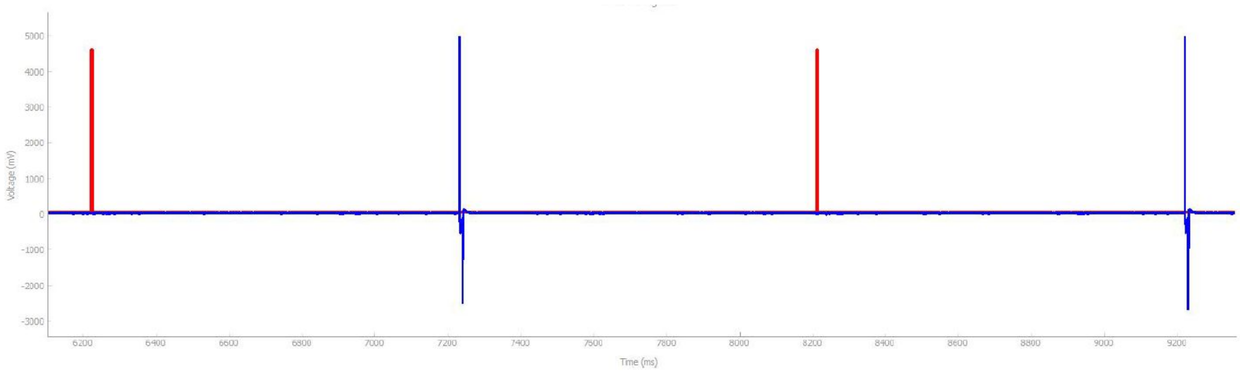


Figure 3.1.1 - Atrium signal on Heartview for Case I

Test case II

Pacemaker Settings		HeartView Settings	
Mode	AAI	Natural Atrium Pulse Width (ms)	On 5
Pulse Width (ms)	10	Natural Ventricle	Off
Pulse Amplitude (V)	4	Natural Heart Rate (bpm)	60
Lower Rate (ppm)	60	Natural AV Delay (ms)	30

Expected Out	Actual Out
The pacemaker inhibits all the pulses.	The result is as expected.
Test result - Pass	
Changes - None	

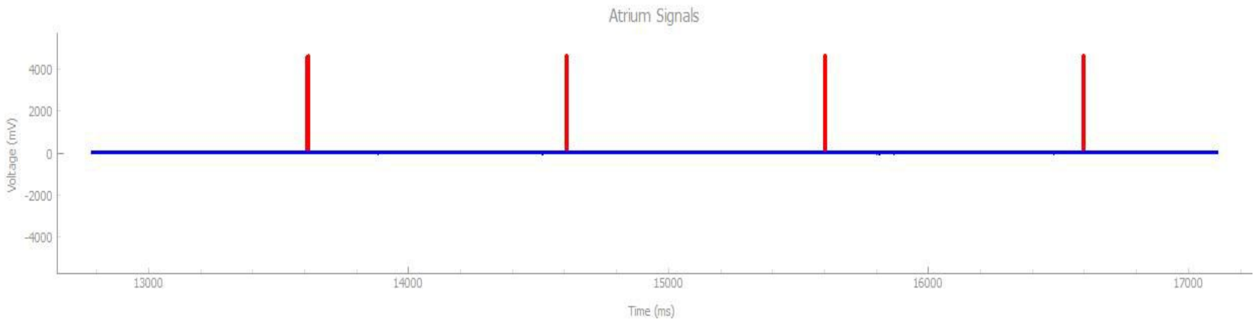


Figure 3.1.2 - Atrium signal on Heartview for Case II

Test case III

Pacemaker Settings		HeartView Settings	
Mode	AAI	Natural Atrium	Off
Pulse Width (ms)	10	Natural Ventricle	Off
Pulse Amplitude (V)	4	Natural Heart Rate (bpm)	NA
Lower Rate (ppm)	60	Natural AV Delay (ms)	NA

Expected Out	Actual Out
The pacemaker paces always with an interval of the lowerInterval.	The result is as expected.
Test result - Pass	
Changes - None	

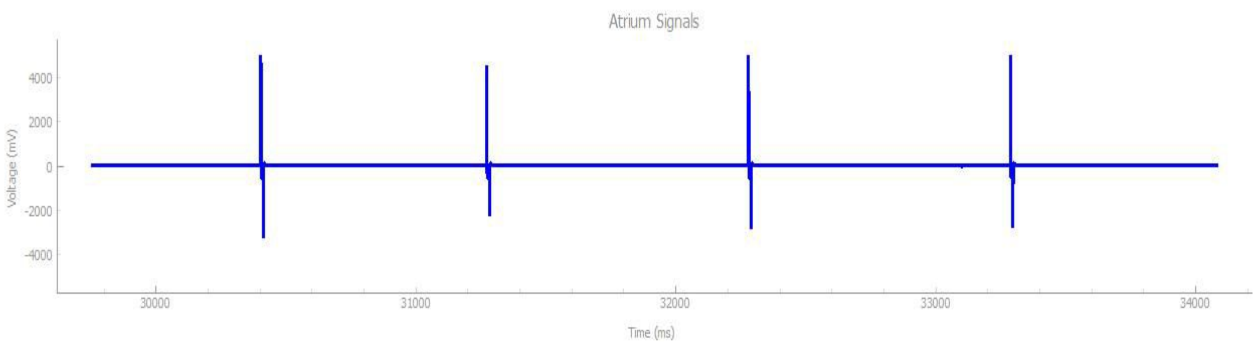


Figure 3.1.3 - Atrium signal on Heartview for Case III

Test case IV

Pacemaker Settings		HeartView Settings	
Mode	AAI	Natural Atrium Pulse Width (ms)	On 5
Pulse Width (ms)	10	Natural Ventricle	Off
Pulse Amplitude (V)	4	Natural Heart Rate (bpm)	58
Lower Rate (ppm)	60	Natural AV Delay (ms)	30

Expected Out	Actual Out
The pacemaker should inhibit all the pulses that are outside ARP.	The result is consistent with what is expected. The pacemaker does not inhibit until the natural pace comes out of ARP and then it inhibits.
Test result - Pass	
Changes - None	

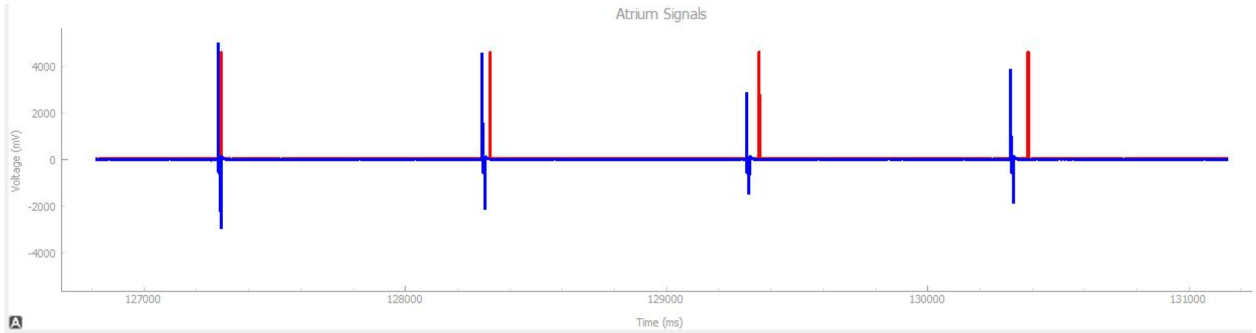


Figure 3.1.4 - Atrium signal on Heartview for Case IV

3.2 AOO and VOO

In order to verify these pacing modes correctly, 2 tests must be done. The first test is to allow the pacemaker to pace without any natural heart beats. The second is to introduce heart beats and check if it paces regardless of natural beats. The output of these pacing modes are exactly the same with only the chamber differing. The test cases below verify AOO but can be used and compared to VOO outputs.

AOO - Test Case 1

Pacemaker Settings		HeartView Settings	
Mode	AOO	Natural Atrium	Off
Pulse Width (ms)	50	Natural Ventricle	Off
Pulse Amplitude (V)	5	Natural Heart Rate (bpm)	30
Lower Rate (ppm)	60	Natural AV Delay (ms)	30

Expected Out	Actual Out
The pacemaker paces always with an interval of the Lower Rate	The result is as expected.

Test result - Pass
Changes - None

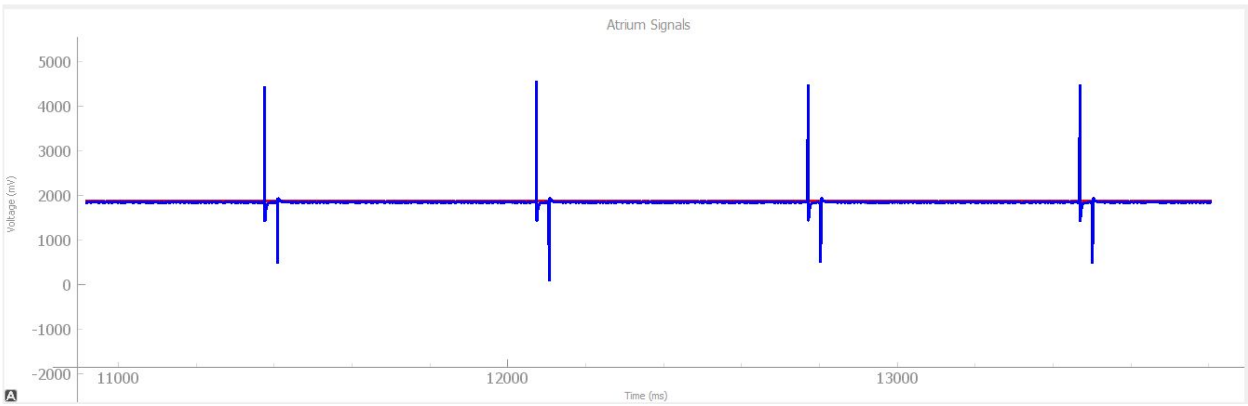


Figure 3.2.1 - Atrium signal on HeartView for AOO test case 1

AOO - Test Case 2

Pacemaker Settings		HeartView Settings	
Mode	AOO	Natural Atrium Pulse Width (ms)	On 10
Pulse Width (ms)	50	Natural Ventricle	Off
Pulse Amplitude (V)	5	Natural Heart Rate (bpm)	75
Lower Rate (ppm)	60	Natural AV Delay (ms)	30

*Regardless of HeartView settings, the pacemaker paces will remain the same.

Expected Out	Actual Out
	The result is as expected.

The pacemaker paces always with an interval of the Lower Rate regardless if the heart has natural beats	
Test result - Pass	
Changes - None	

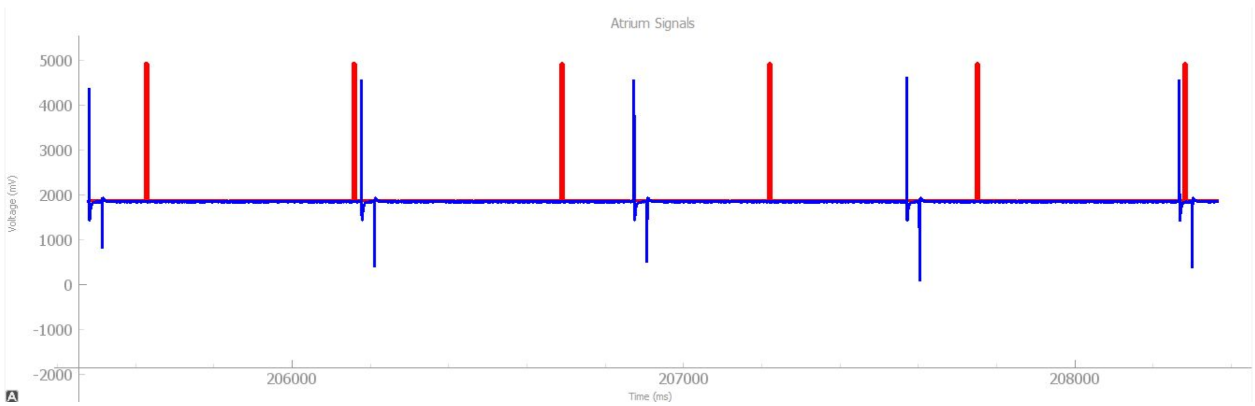


Figure 3.2.2 - Atrium signal on HeartView for AOO test case 2

3.3 DOO

In order to verify these pacing modes correctly, 4 tests must be done. The first test is to allow the pacemaker to pace without any natural heart beats. The second is to introduce heart beats and check if it paces regardless of natural beats. The third is to see if the fixed AV delay is functioning properly. The fourth to see different pulse widths for each of the chambers.

DOO - Test Case 1

Pacemaker Settings		HeartView Settings	
Mode	DOO	Natural Atrium	Off
Pulse Width (ms)	Atrial - 10 Ventricle - 10	Natural Ventricle	Off

Pulse Amplitude (V)	Atrial - 4 Ventricle - 5	Natural Heart Rate (bpm)	30
Lower Rate (ppm)	60	Natural AV Delay (ms)	30
Fixed AV Delay (ms)	150		

<u>Expected Out</u> The pacemaker paces always with an interval of the Lower Rate	<u>Actual Out</u> The result is as expected.
Test result - Pass	
Changes - None	

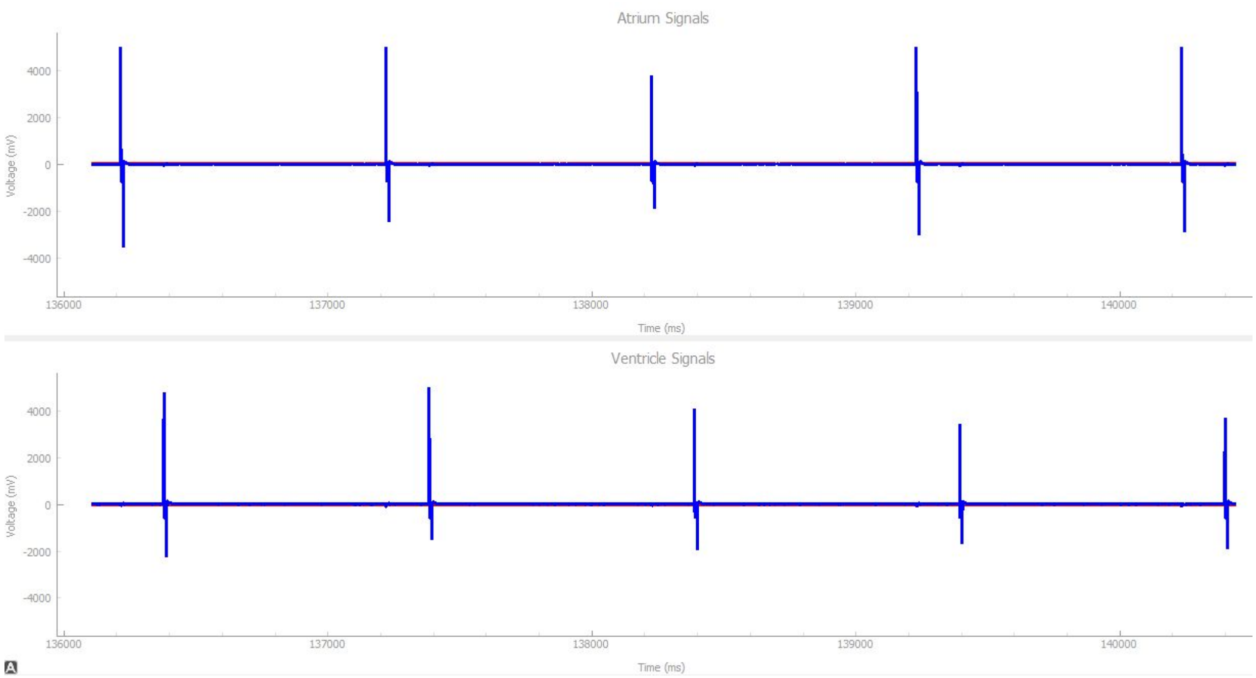


Figure 3.3.1 - Both signals on HeartView for DOO test case 1

DOO - Test Case 2

Pacemaker Settings		HeartView Settings	
Mode	DOO	Natural Atrium	On
Pulse Width (ms)	Atrial - 10 Ventricle - 10	Natural Ventricle	On
Pulse Amplitude (V)	Atrial - 4 Ventricle - 5	Natural Heart Rate (bpm)	30
Lower Rate (ppm)	60	Natural AV Delay (ms)	30
Fixed AV Delay (ms)	150		

*Regardless of HeartView settings, the pacemaker paces will remain the same.

Expected Out	Actual Out
The pacemaker paces always with an interval of the Lower Rate regardless if the heart has natural beats	The result is as expected.
Test result - Pass	
Changes - None	

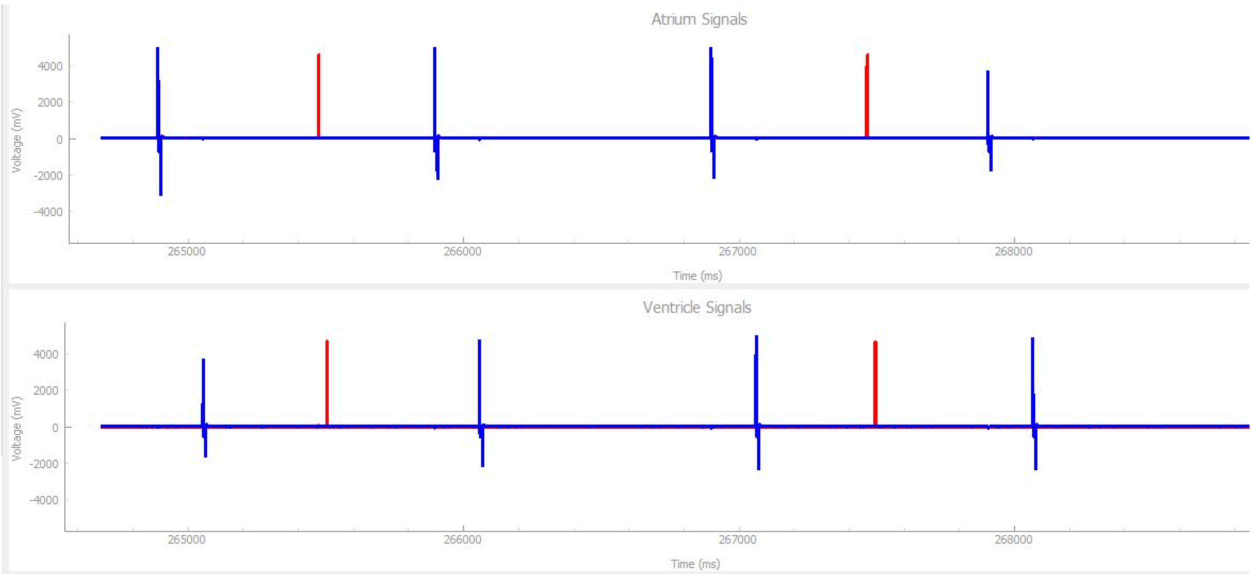


Figure 3.3.2 - Both signals on HeartView for DOO test case 2

DOO - Test Case 3

Pacemaker Settings		HeartView Settings	
Mode	DOO	Natural Atrium	Off
Pulse Width (ms)	Atrial - 10 Ventricle - 10	Natural Ventricle	Off
Pulse Amplitude (V)	Atrial - 4 Ventricle - 5	Natural Heart Rate (bpm)	30
Lower Rate (ppm)	60	Natural AV Delay (ms)	30
Fixed AV Delay (ms)	70		

Expected Out	Actual Out
The Atrium and ventricle signals should be closer as compared to test case I	The result is as expected (compared with the first test case).

Test result - Pass
Changes - None

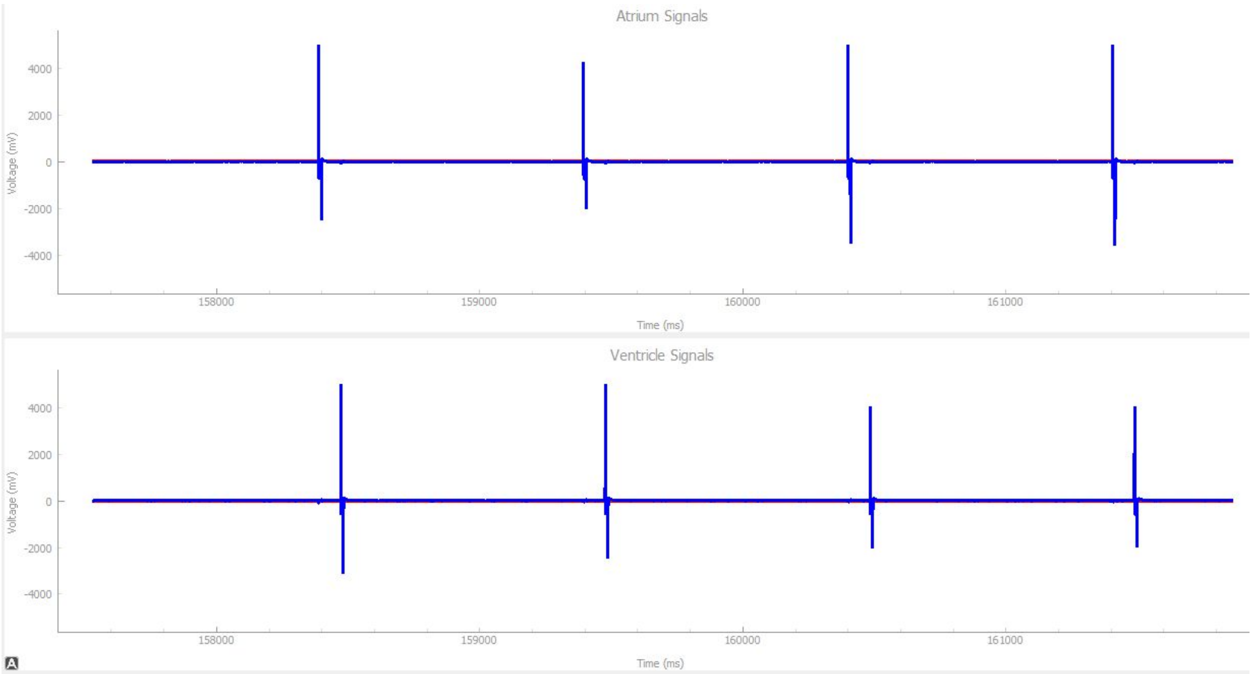


Figure 3.3.3 - Both signals on HeartView for DOO test case 3

DOO - Test Case 4

Pacemaker Settings		HeartView Settings	
Mode	DOO	Natural Atrium	Off
Pulse Width (ms)	Atrial - 10 Ventricle - 20	Natural Ventricle	Off
Pulse Amplitude (V)	Atrial - 4 Ventricle - 5	Natural Heart Rate (bpm)	30
Lower Rate (ppm)	60	Natural AV Delay (ms)	30

Fixed AV Delay (ms)	150		

Expected Out	Actual Out
<u>Expected Out</u> The Atrium and ventricle signals should have different widths, with the Ventricle being double that of Atrium	<u>Actual Out</u> The result is as expected.
Test result - Pass	
Changes - None	

Figure 3.3.4 - Both signals (zoomed) on HeartView for DOO test case 4

3.4 Adaptive Pacing Modes: AOOR, VOOR, DOOR

To verify the adaptive pacing modes correctly, 2 tests must be done. The first test is to introduce activity and observe if the pacing rate will gradually increase. The second is to introduce natural heart beats while activity is performed and observe the change of pacing while the board is shaken. The output of these pacing modes are exactly the same with only the chamber differing. The test cases below verify AOOR but can be used and compared to VOOR outputs.

AOOR - Test Case 1

Pacemaker Settings		HeartView Settings	
Mode	AOOR	Natural Atrium	Off
Pulse Width (ms)	20	Natural Ventricle	Off
Pulse Amplitude (V)	5	Natural Heart Rate (bpm)	30
Lower Rate (ppm)	20	Natural AV Delay (ms)	30

Expected Out	Actual Out
The pacemaker will increase its pacing rate when the accelerometer is shaken	The result is as expected.
Test result - Pass	
Changes - None	

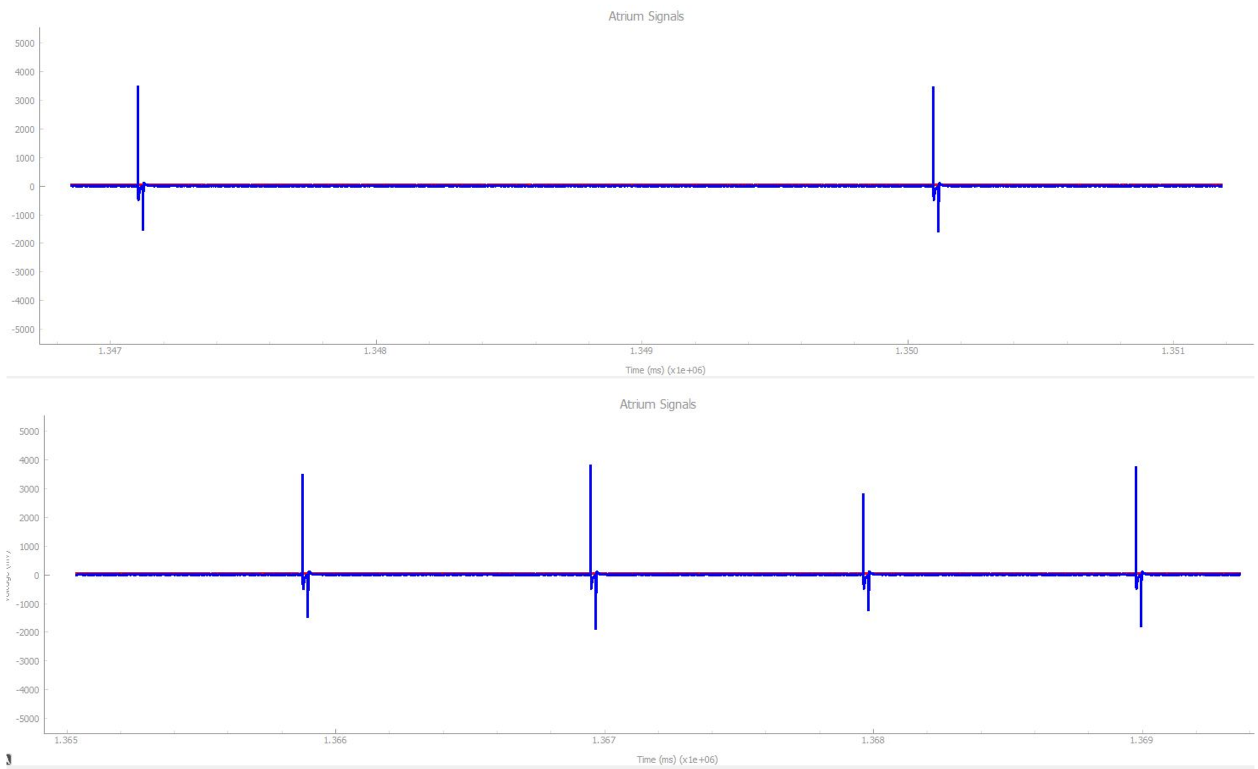


Figure 3.4.1 - Increase of pacing rate after pacemaker is shaken (Atrium)

AOOR - Test Case 2

<u>Pacemaker Settings</u>		<u>HeartView Settings</u>	
Mode	AOOR	Natural Atrium	On
Pulse Width (ms)	20	Natural Ventricle	Off
Pulse Amplitude (V)	5	Natural Heart Rate (bpm)	60
Lower Rate (ppm)	20	Natural AV Delay (ms)	30

<u>Expected Out</u>	<u>Actual Out</u>
	The result is as expected.

The pacemaker will increase its pacing rate when the accelerometer is shaken regardless of natural beats	
Test result - Pass	
Changes - None	

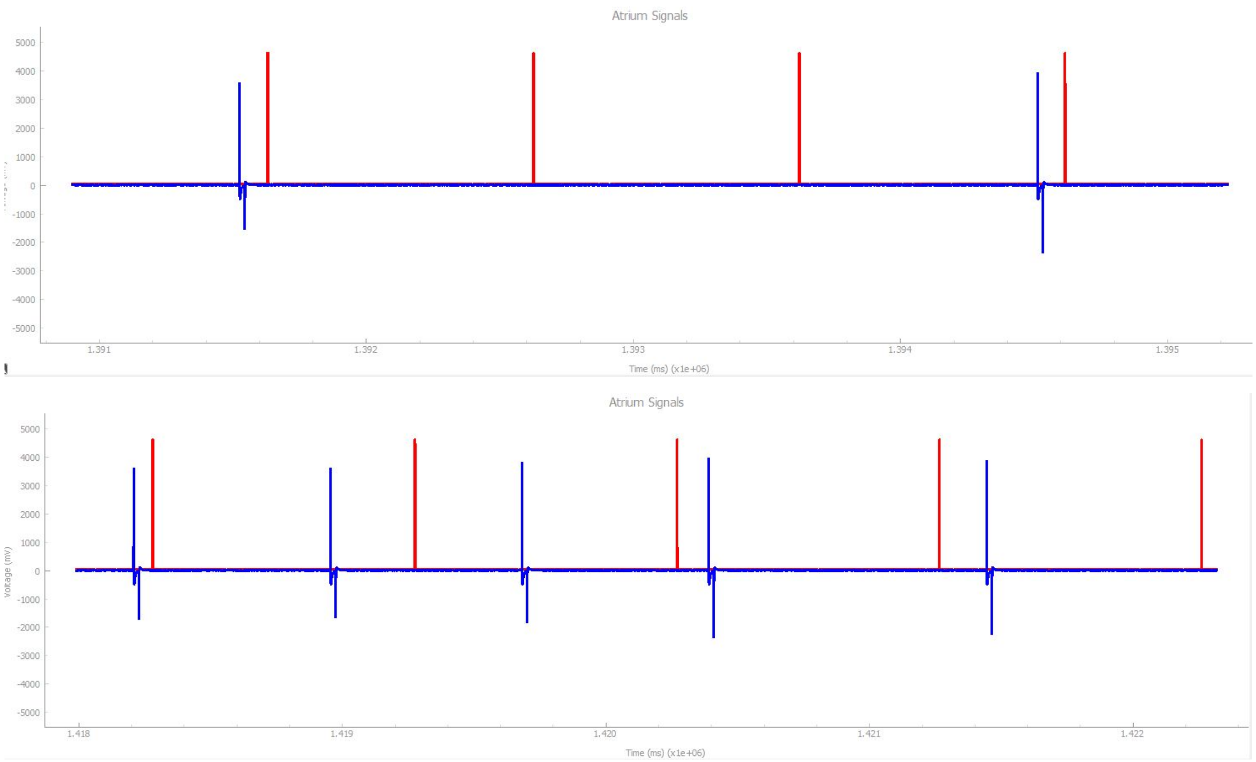


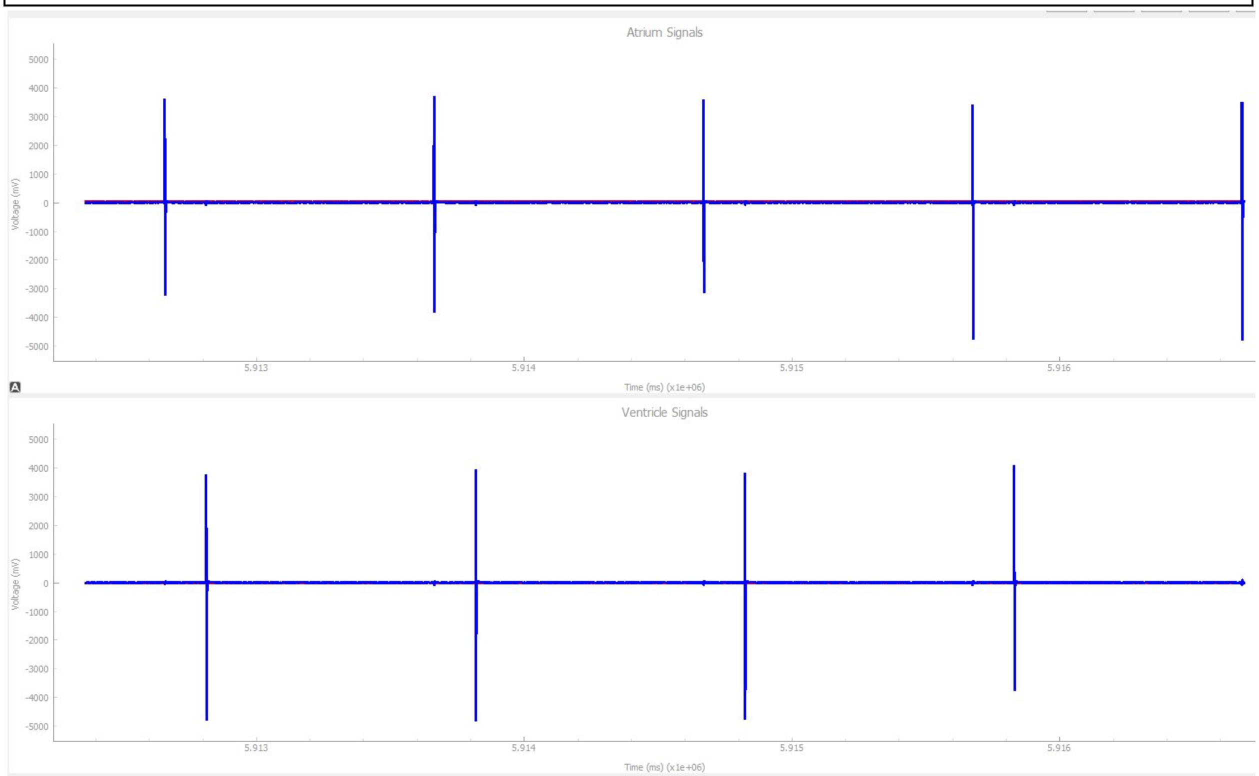
Figure 3.4.2 - Increase of pacing rate after activity is introduced with natural beats

DOOR - Test Case 2

Pacemaker Settings		HeartView Settings	
Mode	AOOR	Natural Atrium	On
Pulse Width (ms)	20	Natural Ventricle	Off
Pulse Amplitude (V)	5	Natural Heart Rate (bpm)	60

Lower Rate (ppm)	60	Natural AV Delay (ms)	30

<u>Expected Out</u>	<u>Actual Out</u>
The pacemaker will increase its pacing rate when the accelerometer is shaken regardless of natural beats	The result is as expected.
Test result - Pass	
Changes - None	



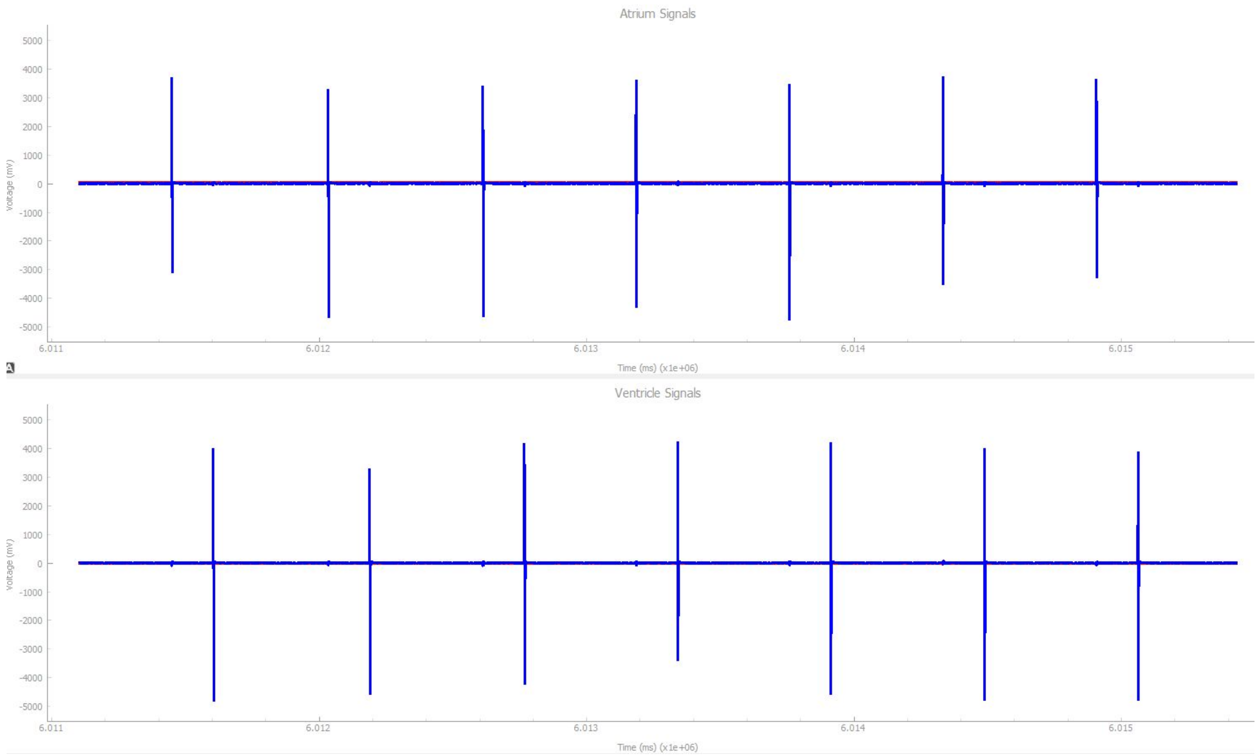


Figure 3.4.3 - DOOR before and after pacemaker is shaken

3.5 Adaptive Pacing Modes: AAIR and VVIR

To verify these modes, 2 tests must be done. The first test is to see if the pacemaker will not pace given that the Lower Rate and Natural Heart Rate are the same. When the board is shaken paces should be seen. The second test is to set the pacemaker Lower Rate higher than the Natural Heart Rate to verify if it will inhibit paces. Then when the pacemaker is shaken, it should see more paces.

AAIR - Test Case 1

Pacemaker Settings		HeartView Settings	
Mode	AAIR	Natural Atrium	On
Pulse Width (ms)	20	Natural Ventricle	Off
Pulse Amplitude (V)	10	Natural Heart Rate (bpm)	60
Lower Rate (ppm)	60	Natural AV Delay	30

		(ms)	

<u>Expected Out</u>	<u>Actual Out</u>
The pacemaker should not inhibit paces since Natural Heart Rate equals Lower Rate. When the pacemaker is shaken paces should be seen	The result is as expected.
Test result - Pass	
Changes - None	

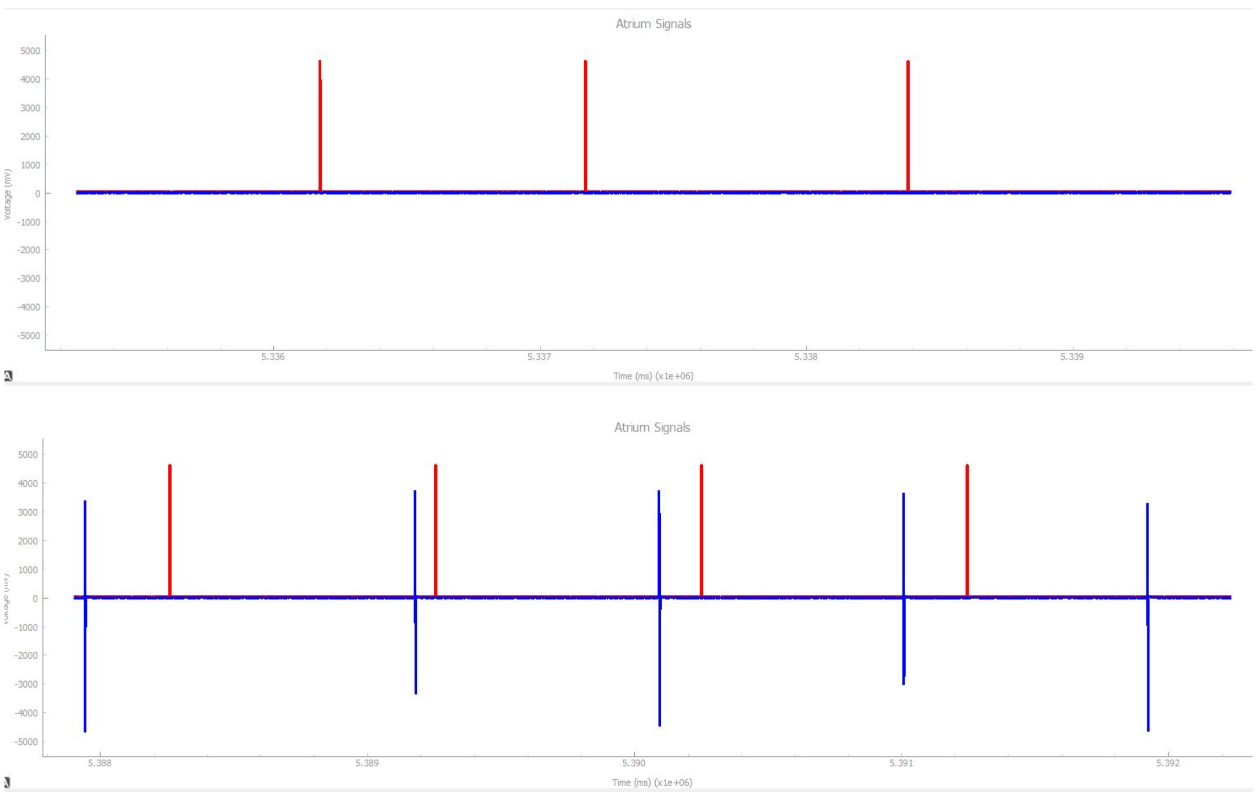
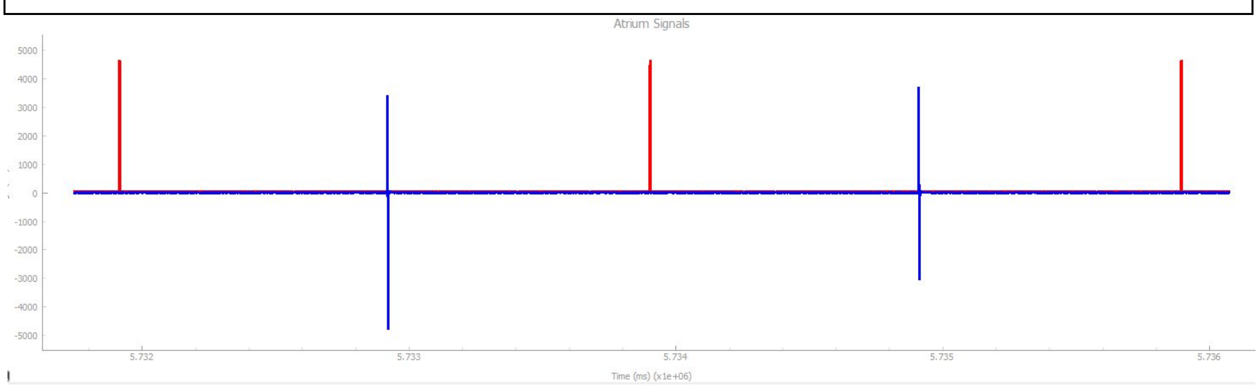


Figure 3.5.1 - Pacemaker is shaken and pace rate goes beyond 60 ppm

AAIR - Test Case 2

Pacemaker Settings		HeartView Settings	
Mode	AAIR	Natural Atrium	On
Pulse Width (ms)	20	Natural Ventricle	Off
Pulse Amplitude (V)	5	Natural Heart Rate (bpm)	30
Lower Rate (ppm)	60	Natural AV Delay (ms)	30

Expected Out	Actual Out
The pacemaker will inhibit paces since the Lower Rate is higher than the Natural Heart Rate. When the pacemaker is shaken, more paces should be seen	The result is as expected.
Test result - Pass	
Changes - None	



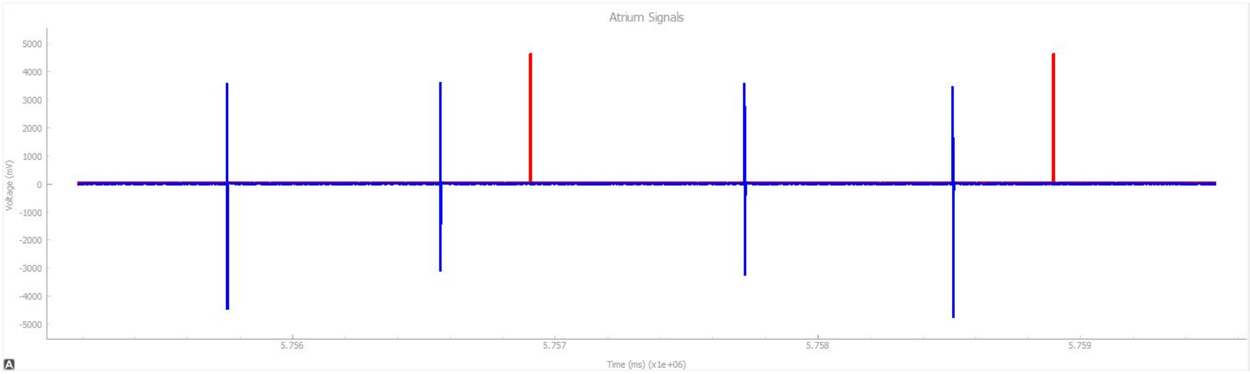


Figure 3.5.2 - Pacemaker inhibits paces(top), Pacemaker is shaken and paces more(bottom)