Assignment #2 - Pacemaker

Group 14
Usman Minhas
Farhan Rouf
Muhtasim Rahman
Noor Alam Mohammad
Cameron Beneteau
Calvin Suen

TABLE OF CONTENTS

Part 1: Requirements

1.1 Requirements

The current requirements are to implement pacemaker modes VOO, VVI, AOO, AAI, DOO, and the rate adaptive versions of all the modes previously listed. Specific information regarding the modes and their logic is provided in sections 1.1.1 to 1.1.6.

Table 1.1: User Parameters [2]

Parameter	Units	Values (Default)	Description
Mode	N/A	AOO, VOO, AAI, VVI, DOO, AOOR, VOOR, AAIR, VVIR, DOOR (VOO)	Indicates the pacemaker mode in use
Lower Rate Limit (LRL)	ppm	30-175 (60 ppm)	The lowest rate (ppm) that the pacemaker should maintain the heart at
Upper Rate Limit (URL)	ppm	50-175 (120 ppm)	The highest rate that the heart can reach with regulation from the pacemaker
Maximum Sensor Rate (MSR)	ppm	50-175 (120 ppm)	The highest rate that the heart can reach in rate adaptive mode when activity is detected (URL takes priority if lower than MSR)
AV Delay	ms	70-300 (150 ms)	The required time delay from an atrial event to a ventricular event
Ventricular Amplitude	V	0.5-7 (3.5V)	The amplitude of the pulse waveform delivered in the ventricle
Atrial Amplitude	V	0.5-7 (3.5V)	The amplitude of the pulse waveform delivered in the atrium
Ventricular Pulse Width	ms	0.1-1.9 (0.4 ms)	The period between the start and end of a pulse delivered to the ventricle
Atrial Pulse Width	ms	0.1-1.9 (0.4 ms)	The period between the start and end of a pulse delivered to the atrium
Ventricular Sensitivity	V	0.1-5 (3.0 V)	The amplitude necessary to sense a ventricular event
Atrial Sensitivity	V	0.1-5 (3.0 V)	The amplitude necessary to sense a atrial event
Ventricular	ms	150-500	The period that the pacemaker will

Refractory Period (VRP)		(320 ms)	ignore any activity in the ventricle after sensing a natural pace or delivering an artificial pace
Atrial Refractory Period (ARP)	ms	150-500 (250 ms)	The period that the pacemaker will ignore any activity in the atrium after sensing a natural pace or delivering an artificial pace
PVARP	ms	150-500 (250 ms)	The period following a ventricular event when an atrial cardiac event shall not trigger a ventricular pace or inhibit an atrial pace
Activity Threshold	N/A	V-Lo1, Low, Med-Low, Med, Med-High, High, V-High (Med)	The threshold that the pacemaker's accelerometer output must exceed to affect the pacemaker's rate
Reaction Time	sec	10-50 (30 sec)	The time required to transition from the LRL to the MSR in rate adaptive modes
Response Factor	N/A	1-16 (8)	Factor that determines the change in rate of the adapted rate in rate adaptive modes
Recovery Time	min	2-16 (5 min)	The time required to transition from the MSR to the LRL in rate adaptive modes

1.1.1 VOO

The purpose of VOO is to pace the heart at the specified LRL regardless of the heart's natural pace. The height of the pulse is determined by Ventricular Pulse Amplitude and the width by Ventricular Pulse Width. The Lower Rate Interval (LRI) is the minimum time between each pace and is calculated by $\frac{60000}{Lower\ Rate\ Limit}$ and has units of ms.

Table 1.1.1: VOO Logic

Condition	Response
The Lower Rate Interval time period has passed since the last pacemaker pace	The ventricle is pulsed and the timer begins again

1.1.2 VVI

The purpose of VVI is to ensure the heart is pacing at the specified LRL while also taking into account the heart's natural pulses. If the heart is pacing at or above LRL, no artificial pace will be delivered. The height of the pulse is determined by Ventricular Pulse Amplitude and the width by Ventricular Pulse

Width. The Lower Rate Interval (LRI) is the minimum time between each pace and is calculated by $\frac{60000}{Lower\ Rate\ Limit}$ and has units of ms.

Table 1.1.2: VVI Logic

Condition	Response
The Lower Rate Interval time period has passed since the last natural or artificial pace in the ventricle	The ventricle is pulsed and the LRI begins timing again. The pacemaker enters the VRP
VRP has passed since the last ventricular event and a ventricular event has been detected	Reset LRI and the pacemaker enters VRP

1.1.3 AOO

The purpose of AOO is to pace the heart at the specified LRL regardless of the heart's natural pace. The height of the pulse is determined by Atrial Pulse Amplitude and the width by Atrial Pulse Width. The Lower Rate Interval (LRI) is the minimum time between each pace and is calculated by $\frac{60000}{Lower\ Rate\ Limit}$ and has units of ms.

Table 1.1.3: AOO Logic

Condition	Response
The Lower Rate Interval time period has passed since the last pacemaker pace	The atrium is pulsed and the timer begins again

1.1.4 AAI

The purpose of AAI is to ensure the heart is pacing at the specified LRL while also taking into account the heart's natural pulses. If the heart is pacing at or above LRL, no artificial pace will be delivered. The height of the pulse is determined by Atrial Pulse Amplitude and the width by Atrial Pulse Width. The Lower Rate Interval (LRI) is the minimum time between each pace and is calculated by $\frac{60000}{Lower\ Rate\ Limit}$ and has units of ms.

Table 1.1.4: AAI Logic

Condition	Response
The Lower Rate Interval time period has passed since the last natural or artificial pace in the atrium	The atrium is pulsed and the LRI begins timing again. The pacemaker enters the ARP
ARP has passed since the last ventricular event and an atrial event has been detected	Reset LRI and the pacemaker enters ARP

1.1.5 DOO

The purpose of DOO is to pace both the atrium and ventricle regardless of the heart's natural paces. The height of the pulse is determined by Atrial Pulse Amplitude and the width by Atrial Pulse Width.The height of the pulse is determined by Ventricular Pulse Amplitude and the width by Ventricular Pulse Width. The Lower Rate Interval (LRI) is the minimum time between each pace and is calculated by $\frac{60000}{Lower\ Rate\ Limit}$ and has units of ms. The AV Delay is the required waiting period to pace the ventricle after an atrial pace.

Table 1.1.5: DOO Logic

Condition	Response
The Lower Rate Interval time period has passed since the pace	The atrium is paced, LRI is reset and the AV Delay period begins
The AV Delay time period has passed since the last atrial pace	The ventricle is paced and the AV Delay is reset

1.1.6 Rate Adaptive

The purpose of rate adaptive modes (XXXR) is to adjust the rate at which the heart is paced based on activity input from the accelerometer. When activity is detected, a new rate known as the Adapted Rate (ppm) is calculated. The calculation is explained in section 2.1.2.2. This adapted rate is converted to an interval in ms using $\frac{60000}{Adapted\ Rate}$. This Adapted Rate Interval replaces the Lower Rate Interval in all the previous modes to pace at the required rate.

1.2 Expected Changes

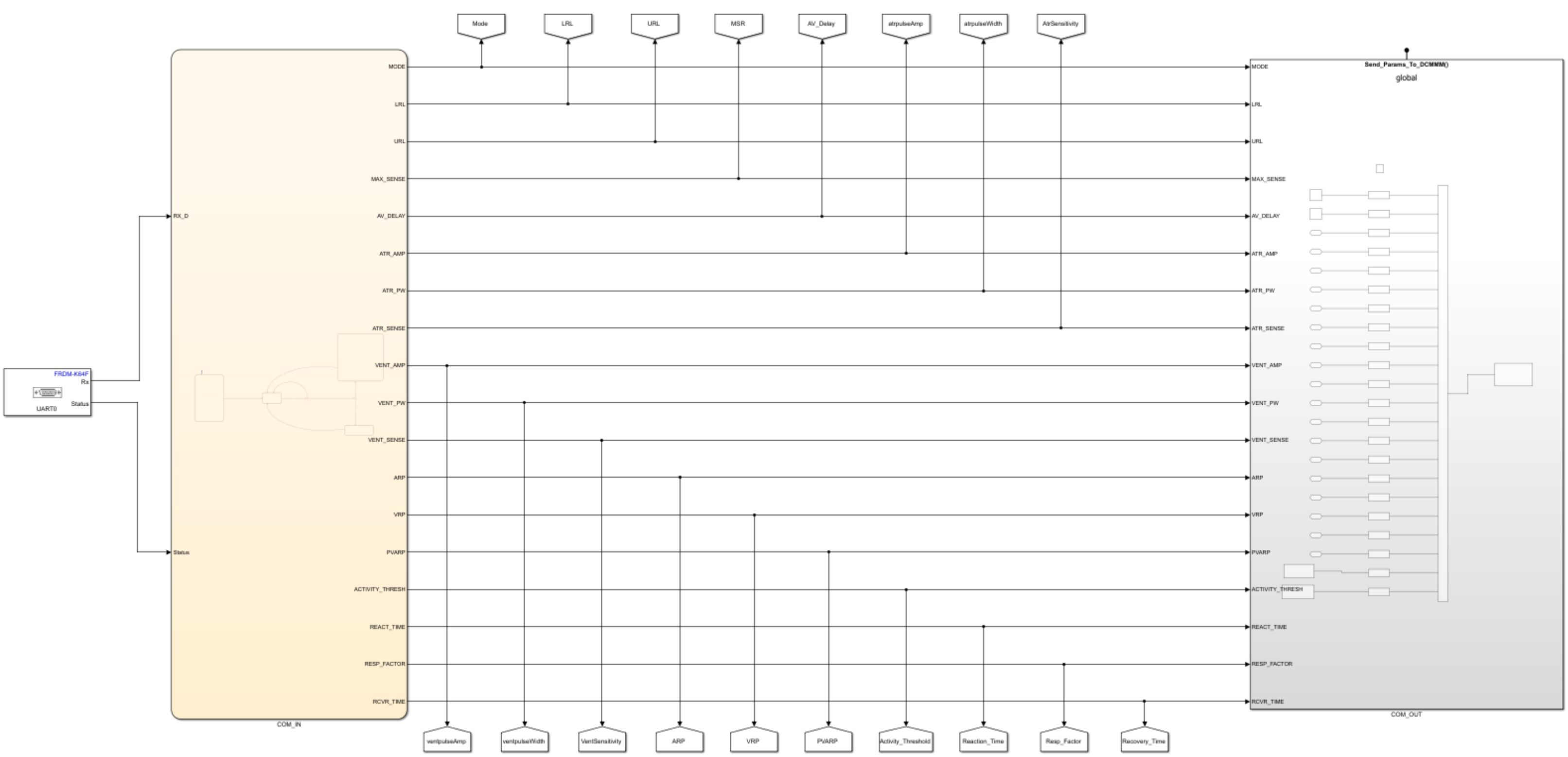
Possible future changes include adding DDD and DDDR modes for dual pacing and sensing. Other changes include improving efficiency by decreasing the size of data types used.

Part 2: Simulink Model

2.1 State Flows & Subsystems

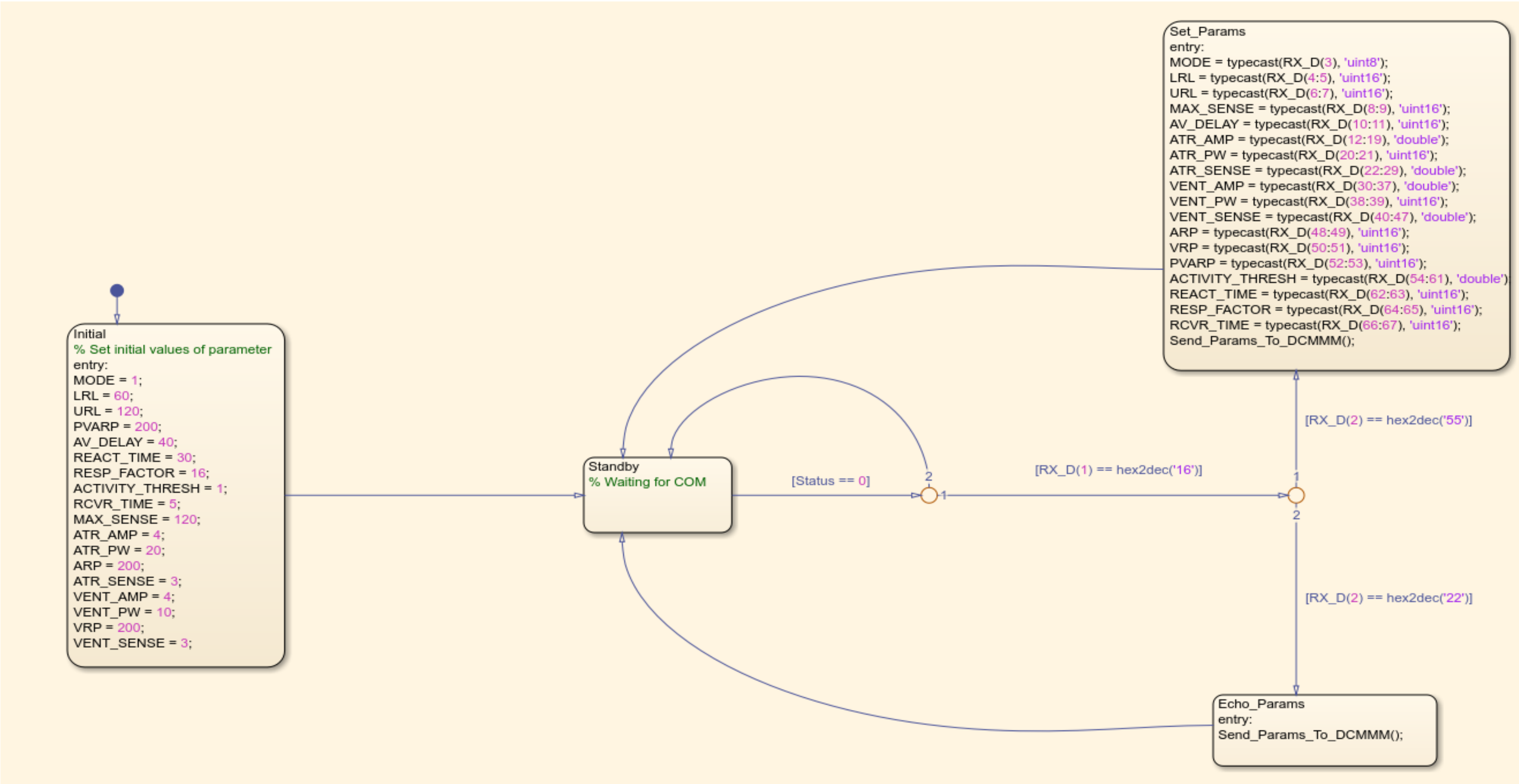
2.1.1 Serial Communication

On the left side you can see the COM_IN stateflow which receives serial data and on the right you can see the COM_OUT stateflow which uses the Send_Params_to_DCMM() function to send serial data out.



2.1.1.1 Serial Communication In

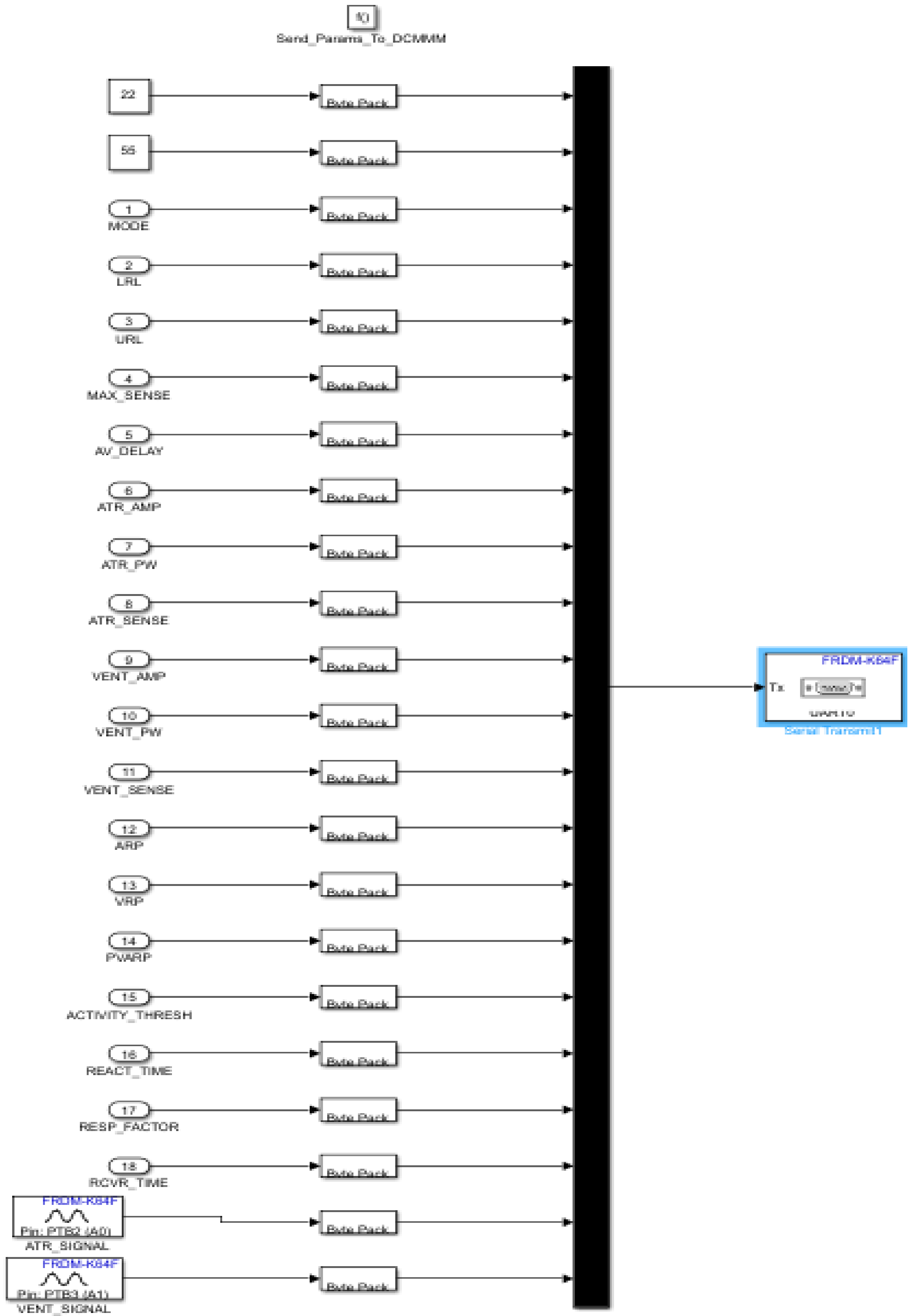
The COM_IN stateflow receives data packets from the DCM via the serial port and then runs the following stateflow logic on the data packet.



The data types were decided through initially starting with the smallest possible data types. Then, during testing these data types returned various errors in the simulink. After a lot of deliberation, a decision was made to make all data types included in calculations doubles or uint16 to completely avoid errors.

Data Type	INPUT PARAMETERS
uint8	MODE
uint16	LRL, URL, MAX_SENSE, AV_DELAY, ATR_PW, VENT_PW, ARP, VRP, PVARP, REACT_TIME, RESP_FACTOR, RCVR_TIME
double	ATR_AMP, ATR_SENSE, VENT_AMP, VENT_SENSE, ACTIVITY_THRESH

2.1.1.2 Serial Communication Out



The COM_OUT function Send_Params_To_DCMMM() takes in the values that were input by the DCM as well as the Atrial and Ventricular signals and sends them back to the DCM. The data types are the same as those in the COM_IN stateflow and the ATR_SIGNAL and VENT_SIGNAL are both doubles. The ATR_SIGNAL and VENT_SIGNAL are signals from the heart that are used to generate the electrocardiogram in the DCM.

2.1.2 Main State Flow

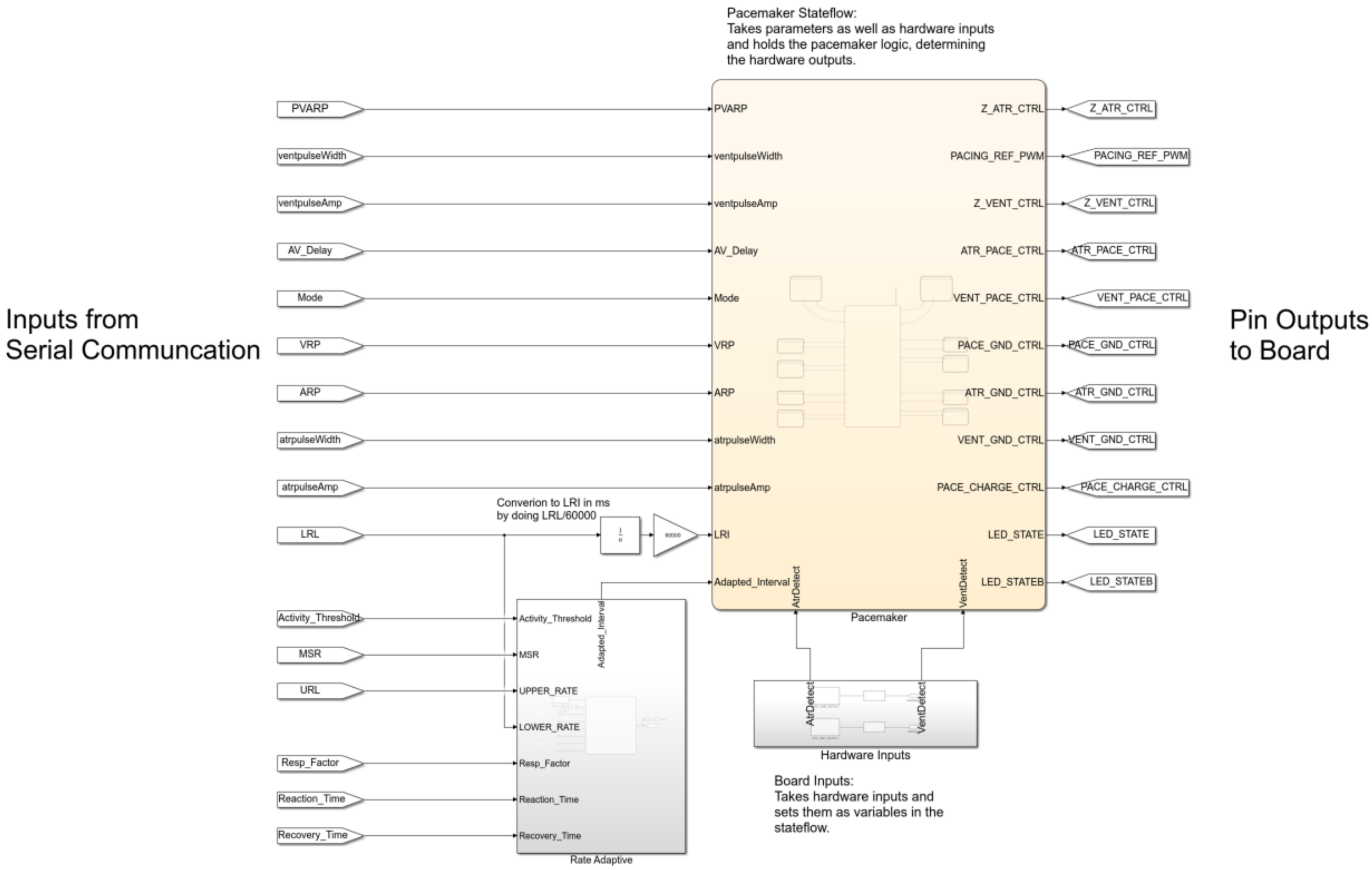


Figure 2.1.2 Pacemaker Subsystem

2.1.2.1 Input Parameters

Direct Inputs

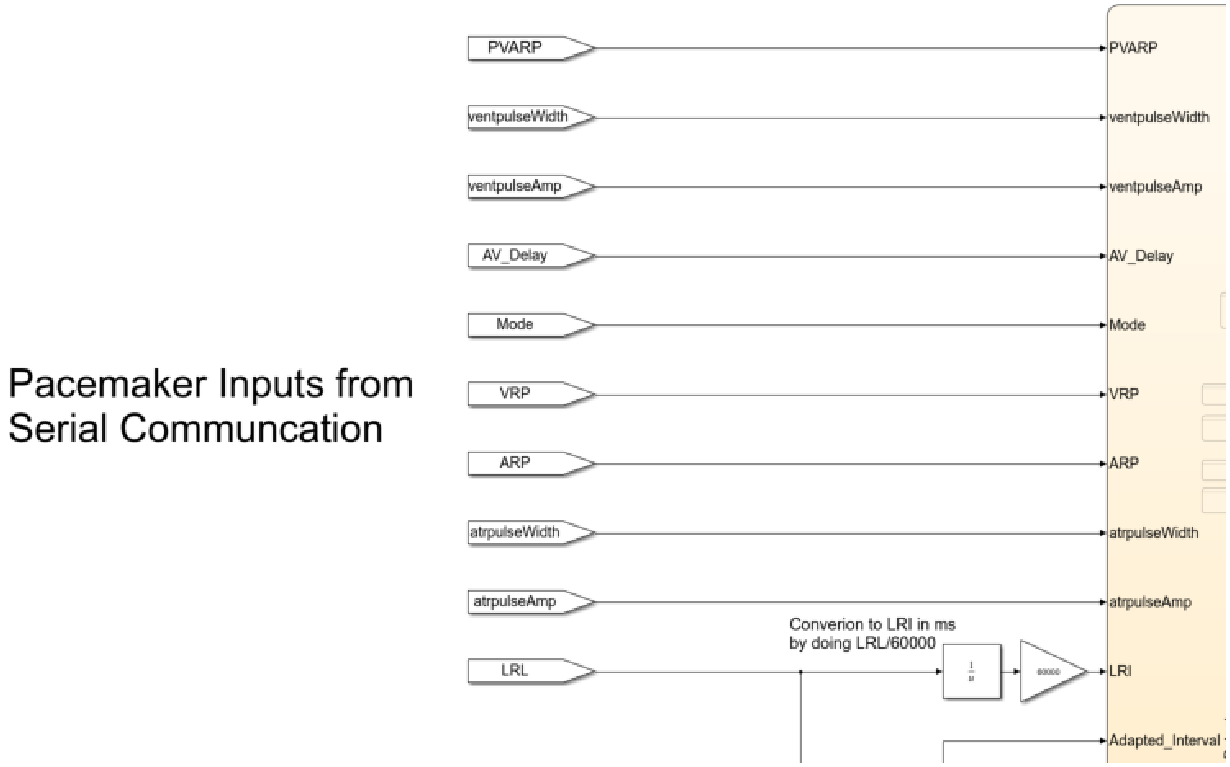


Figure 2.1.2.1.1 Direct Inputs

The following is a table explaining the parameters used in the pacemaker state flow:

Table 2.1.2.1.1: Pacemaker Input Variables

Parameter	Description	Data Type
PVARP	Time period following a ventricle event when an atrial event shall not trigger a ventricular pace	Inherit: Same as specified in 2.1.1.1 Serial Communication In
ventpulseWidth	The width of a ventricular pulse	Inherit: Same as specified in 2.1.1.1 Serial Communication In
ventpulseAmp	The amplitude of the ventricular pulse	Inherit: Same as specified in 2.1.1.1 Serial Communication In
AV_Delay	The period that must be waited after an atrial pace before pacing the ventricle	Inherit: Same as specified in 2.1.1.1 Serial Communication In
Mode	A numerical value to determine pacemaker mode	Inherit: Same as specified in 2.1.1.1 Serial Communication In
VRP	The required waiting period after a	Inherit: Same as specified in

	ventricle pulse or pace before sensing again	2.1.1.1 Serial Communication In
ARP	The required waiting period after a atrial pulse or pace before sensing again	Inherit: Same as specified in 2.1.1.1 Serial Communication In
atrpulseWidth	The width of an atrial pulse	Inherit: Same as specified in 2.1.1.1 Serial Communication In
atrpulseAmp	The width of a ventricular pulse	Inherit: Same as specified in 2.1.1.1 Serial Communication In
LRL	The Lower Rate Interval in ppm. The formula $\frac{60000}{Lower\ Rate\ Limit}$ calculates the LRI (Lower Rate Limit) in ms, which is inputted into the pacemaker	Double
Adapted_Interval	The adapted time period calculated in the Rate Adaptive subsystem in ms. Refer to 2.1.2.2 Rate Adaptive	Double

Rate Adaptive inputs

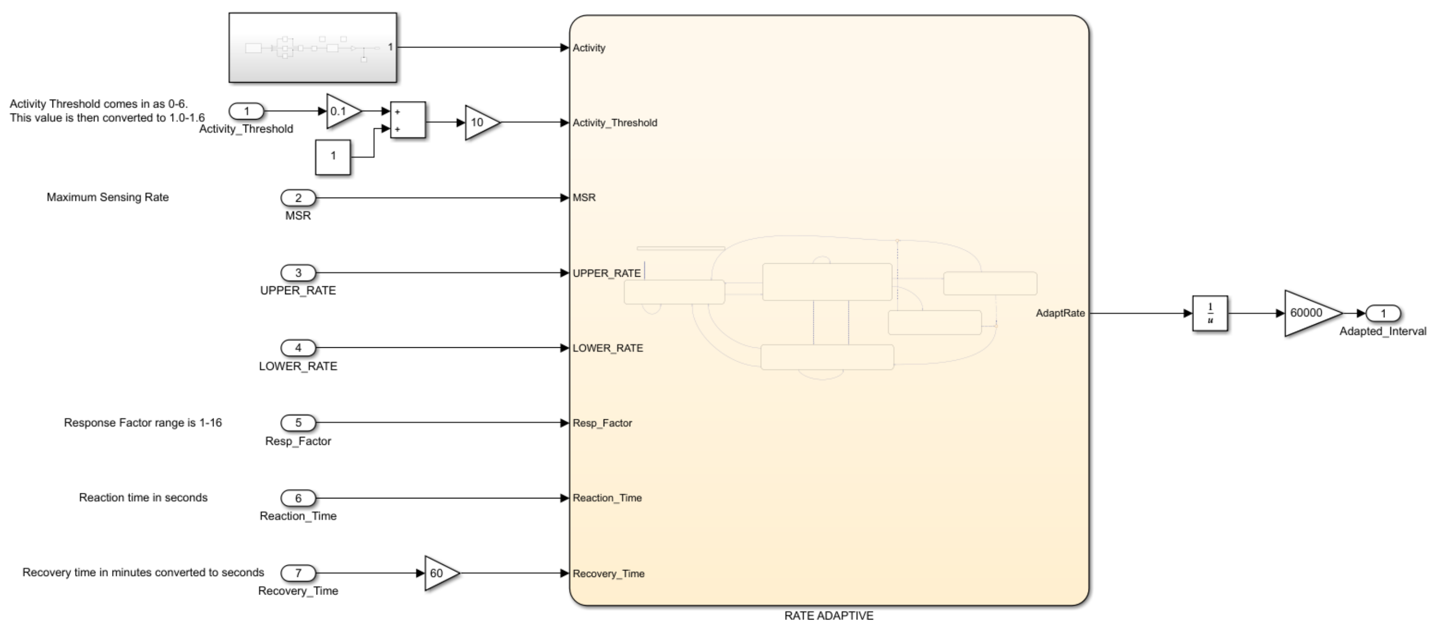


Figure 2.1.2.1.2 Rate Adaptive Inputs

Table 2.1.2.1.2: Rate Adaptive Input Variables

Input	Description	Data Type
Activity_Threshold	Inputs come from the DCM as 0-6 and converted to 1.0-1.6	double
MSR	Maximum Sensor Rate	uint16
UPPER_RATE	Upper Rate Limit	uint16
LOWER_RATE	Lower Rate Limit	uint16
Resp_Factor	Response factor	uint16
Reaction_Time	Time taken to go from LOWER_RATE to MSR	uint16
Recovery_Time	Time taken to go from MSR to LOWER_RATE	uint16

Hardware Inputs

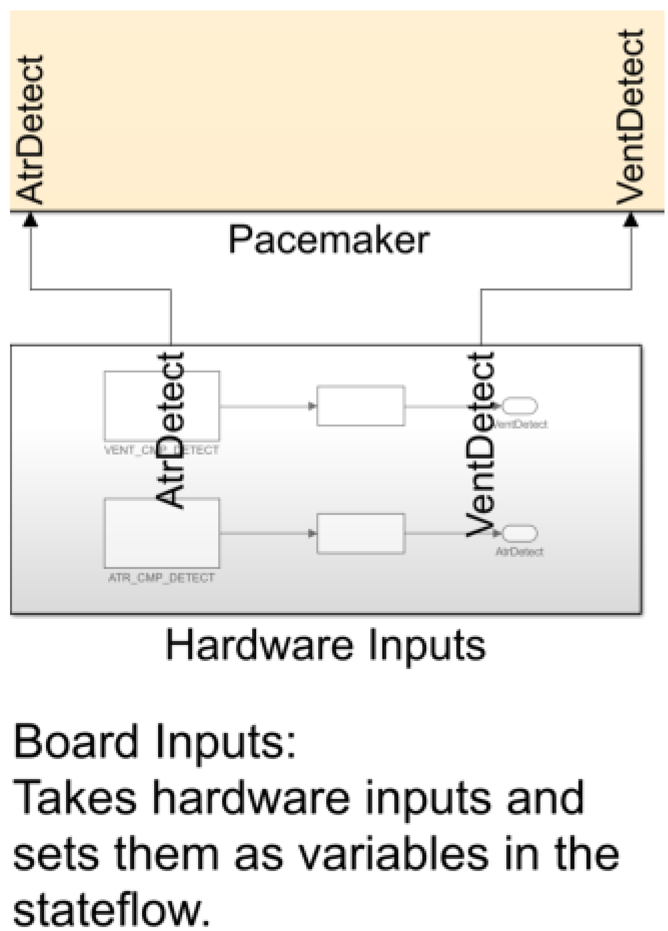


Figure 2.1.2.1.3 Hardware Inputs Subsystem

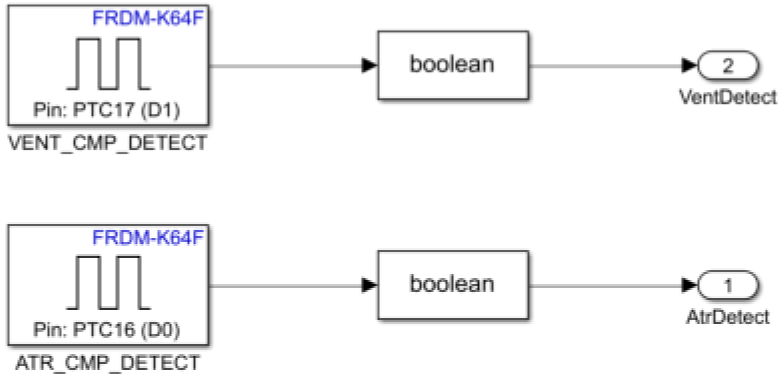
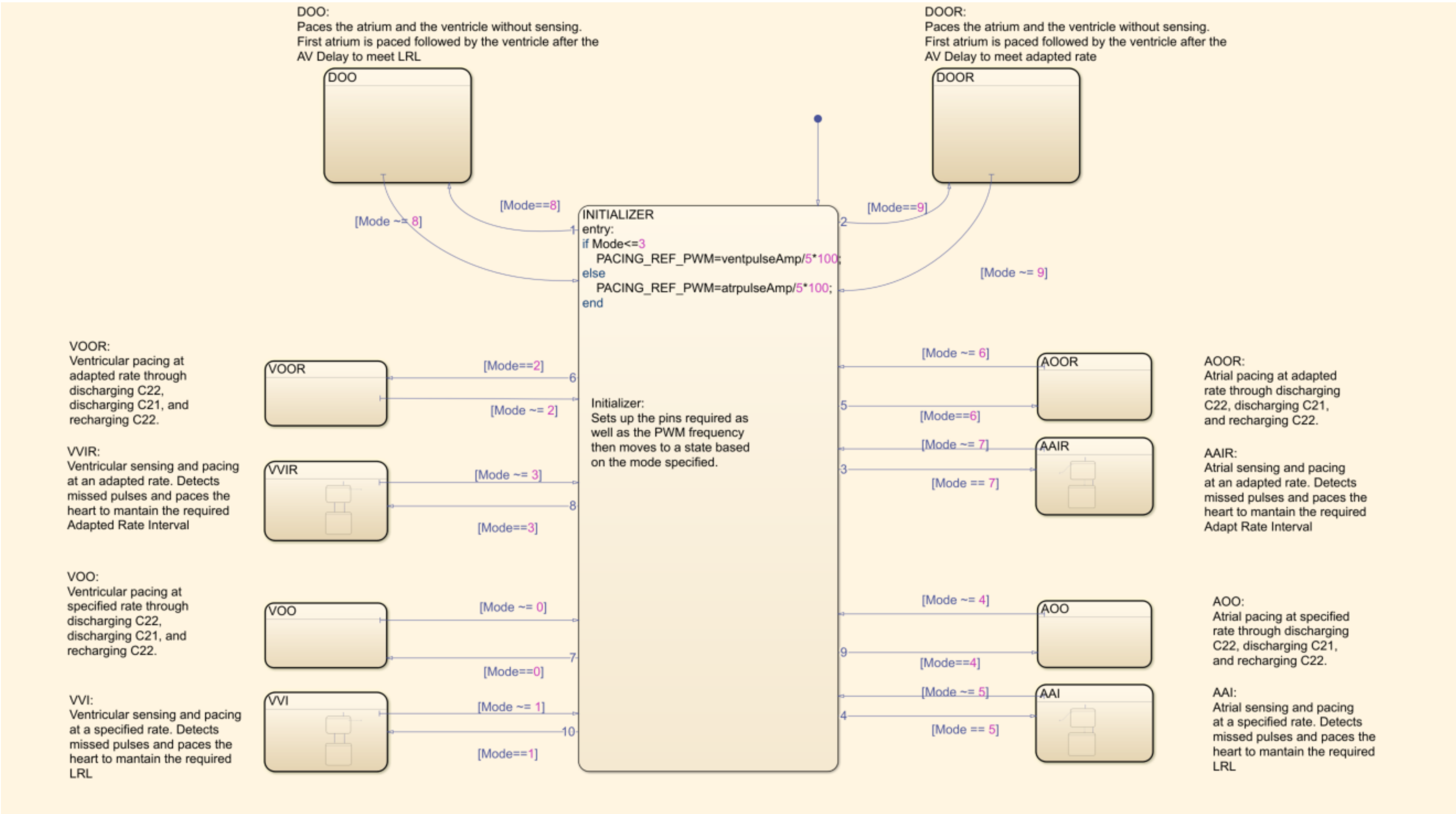


Figure 2.1.2.1.4 Hardware Input Pins

Table 2.1.2.1.2: Hardware Input Variables [1]

Parameter	Description	Data Type
AtrDetect	Detects atrial events	Boolean
VentDetect	Detects ventricular events	Boolean



Seen above is the overall state flow with each mode and an initializer. The benefit behind having each mode in an independent state is that it allows modularity. If one mode requires changes or optimizations it can be easily done without damaging the logic of all unchanged modes.

Table 2.1.2.1.2: General Pacemaker FSM

State	Description	Entry	Exit
INITIALIZER	The purpose of this state is to initialize the PACING_REF_PWM pin with the correct duty cycle using the general formula $\frac{pulseAmp}{5\,V} * 100$ which calculates a value in duty cycle. ventpulseAnp is used for VOO, VVI, VOOR, VVIR (Modes <= 3) and atrpulseAmp is used for AOO, AAI, AOOR, AAIR, DOO, DOOR (Modes>3).	Default Or Change in Mode	Mode == 0-9
VOO	Contains logic for VOO	Mode == 0	Mode != 0
VVI	Contains logic for VVI	Mode == 1	Mode != 1

VOOR	Contains logic for VOOR	Mode == 2	Mode != 2
VVIR	Contains logic for VVIR	Mode == 3	Mode != 3
AOO	Contains logic for AOO	Mode == 4	Mode != 4
AAI	Contains logic for AAI	Mode == 5	Mode != 5
AOOR	Contains logic for AOOR	Mode == 6	Mode != 6
AAIR	Contains logic for AAIR	Mode == 7	Mode != 7
DOO	Contains logic for DOO	Mode == 8	Mode != 8
DOOR	Contains logic for DOOR	Mode == 9	Mode != 9

2.1.2.2 Rate Adaptive

Subsystems

The purpose of the Rate Adaptive Subsystem is to monitor the level activity the user is currently undergoing in order to adjust the pacing rate accordingly. The level of activity is measured using the three-axis accelerometer on the pacemaker. The figure below shows the inputs to the rate adaptive stateflow.

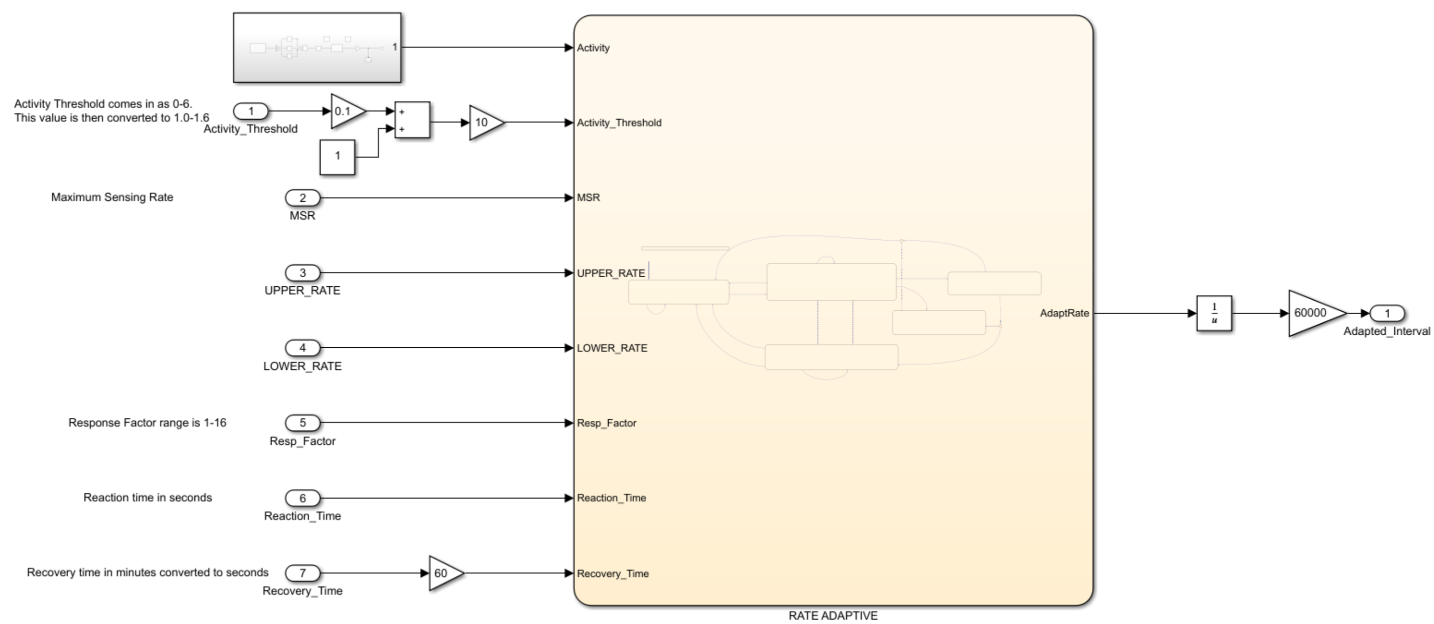


Figure 2.1.2.2.1 Rate Adaptive Subsystem

The post-processing on the rate adaptive stateflow converts the output of the rate adaptive from units of Pulses per minute to units of time (milliseconds), which is shown in the figure below:

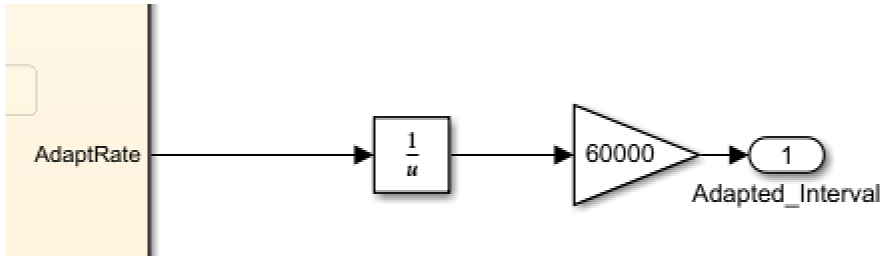


Figure 2.1.2.2.2 Post-processing Logic

2.1.2.2.1 Activity Level Subsystem

The level of activity is determined through the accelerometer data from the pacemaker hardware; which first goes through a preprocessing subsystem to ensure we convert the raw acceleration vector to the absolute magnitude of the raw acceleration data using the following formula $|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$.

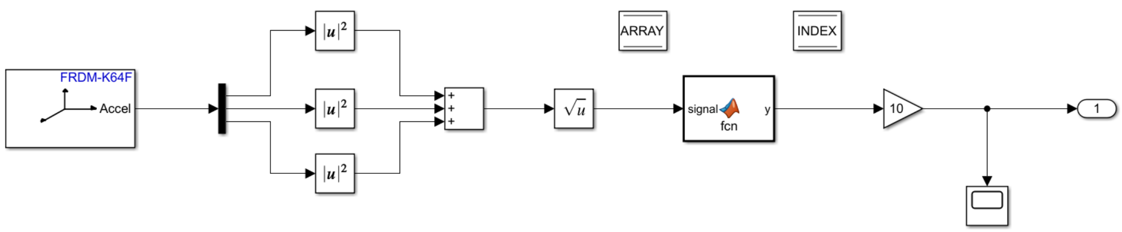


Figure 2.1.2.2.3 Pre-Processing on Accelerometer data

Additionally, we have a moving average calculating function that calculates the average of the last 50 inputs stored in an array. The function takes an input signal and provides an output, *y.*

```
function y = fcn(signal)
global INDEX;
INDEX = mod(INDEX + 1, 50) + 1;
global ARRAY;
ARRAY(INDEX) = signal;
y = mean(ARRAY);
```

Figure 2.1.2.2.4 Matlab Script for Running Average

2.1.2.2.2 Rate Adaptive Stateflow

The purpose of the rate adaptive stateflow is to determine if the current activity level exceeds the activity threshold. If the Adaptive Rate is less than Target Rate or less than or equal to the Lower Rate Limit the pacing rate would increase or decrease linearly until the Adaptive rate either reaches the Target Rate or the Lower Rate Limit. The figure below displays all the states in the stateflow and each connection:

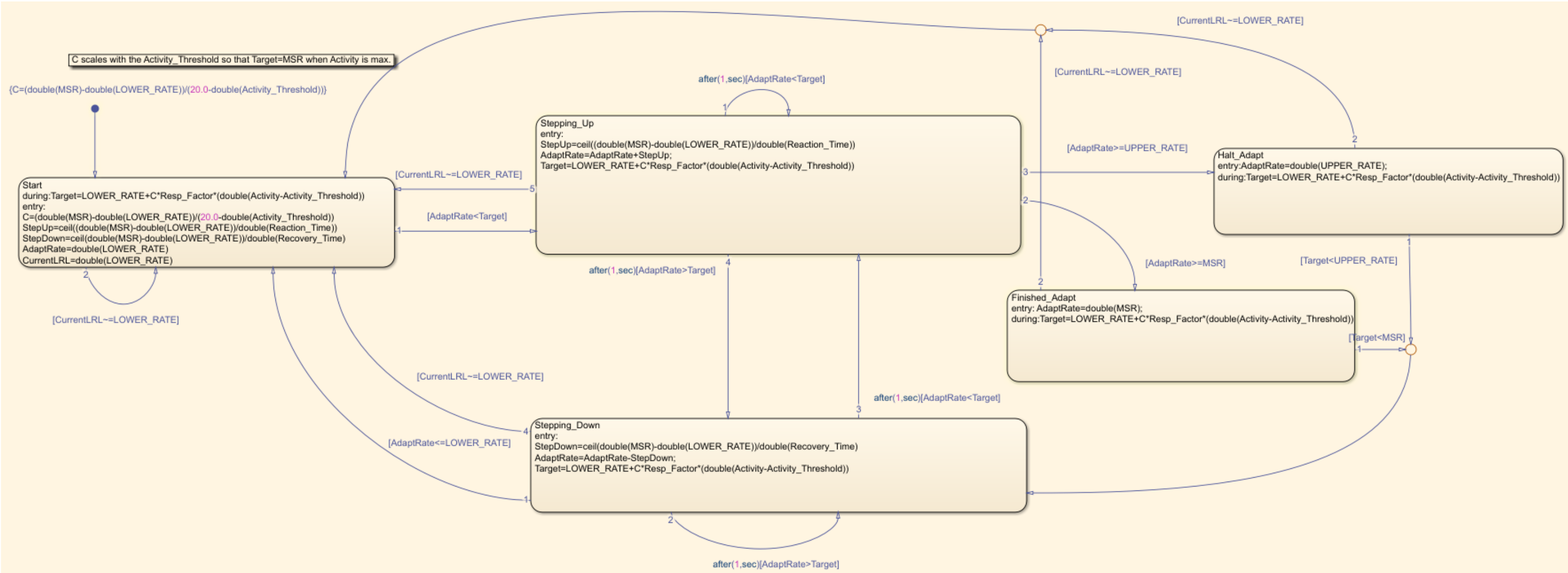


Figure 2.1.2.2.5 Rate Adaptive Stateflow Logic

Table 2.1.2.2.1 States Table for Rate Adaptive Stateflow

State	State Variables	Action	Transition Condition	Destination State
Start	AdaptRate Target C StepUp StepDown CurrentLRL	1.Sets adaptive rate to LOWER_RATE. 2.Continuously calculates Target based on current Activity. 3.C is the constant that scales with Activity Threshold 4. StepUp and StepDown represent the increments or decrements. 5. CurrentLRL is a state variable to check when the LRL input is changed	1. AdaptRate < Target 2. CurrentLRL~=LRL	1.Stepping_Up 2. Start
Stepping _Up	AdaptRate Target	Increases the AdaptRate by the value stored in StepUp every second until the AdaptRate satisfies one of the transition conditions.	1.AdaptRate < Target 2.AdaptRate > Target 3.AdaptRate >=MSR 4.AdaptRate>= UpperRate 5. CurrentLRL~=LRL	1.Stepping_Up 2.Stepping_Down 3.Finished_Adapt 4.Halt_Adapt 5. Start
Stepping _ Down	StepDown AdaptRate Target	Recalculates StepDown and decrements AdaptRate by it until a transition condition is satisfied	1.AdaptRate < Target 2.AdaptRate > Target	1.Stepping_Up 2.Stepping_Down

			3.AdaptRate <= LRL 4. CurrentLRL~=LRL	3.Start 4. Start
Finished _Adapt	AdaptRate = Target =	Keeps calculating Target until one of the transition conditions are satisfied	1.Target < MSR 2. CurrentLRL~=LRL	1.Stepping_Down 2. Start
Halt_ Adapt	AdaptRate Target	Keeps calculating Target until one of the transition conditions are satisfied	1.Target < UpperRate 2. CurrentLRL~=LRL	1.Stepping_Down 2. Start

Table 2.1.2.2.2 State Variables, Definitions, Data Types

State Variable (*purpose*)	Definition	Data Type
Target (Stores the Target Heart Rate)	*LOWER_RATE+C*Resp_Factor*(double(Activity-Activity_Threshold))*	double
AdaptRate (Stores current adapted rate)	*double(LOWER_RATE)*	double
StepUp (Stores value to increment AdaptRate)	*ceil((double(MSR)-double(LOWER_RATE))/double(Reaction_Time))*	double
StepDown (Store value to decrement AdaptRate)	*ceil(double(MSR)-double(LOWER_RATE))/double(Recovery_Time)*	double
C (is a constant that changes according to the Activity_Threshold. It changes to adapt so that Target=MSR when Activity is maximum, which was found by shaking the boards as hard as possible)	*(double(MSR)-double(LOWER_RATE))/(20.0-double(Activity_Threshold))*	double
CurrentLRL(stores value of current LOWER_RATE)	*double(LOWER_RATE)*	double

2.1.2.3 VOO/VOOR

Seen below is the stateflow for VOO along with a table explaining the FSM:

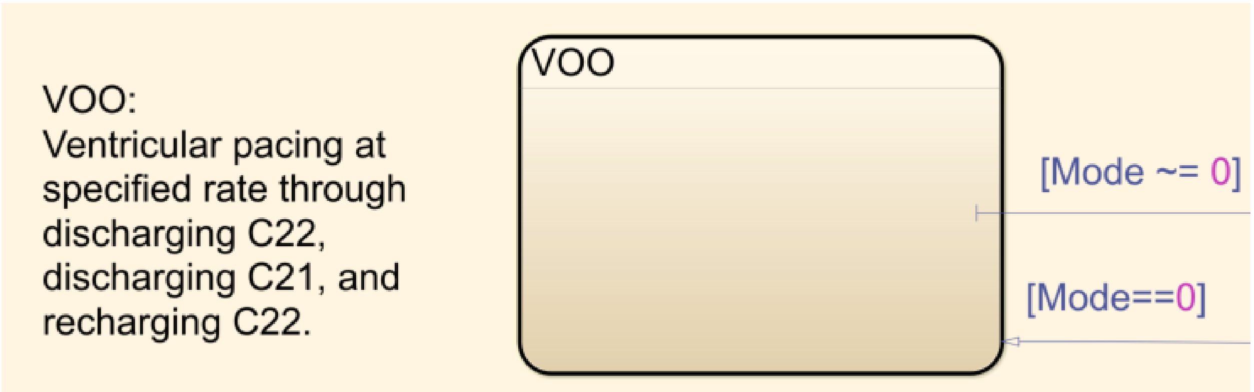


Figure 2.1.2.3.1 VOO State

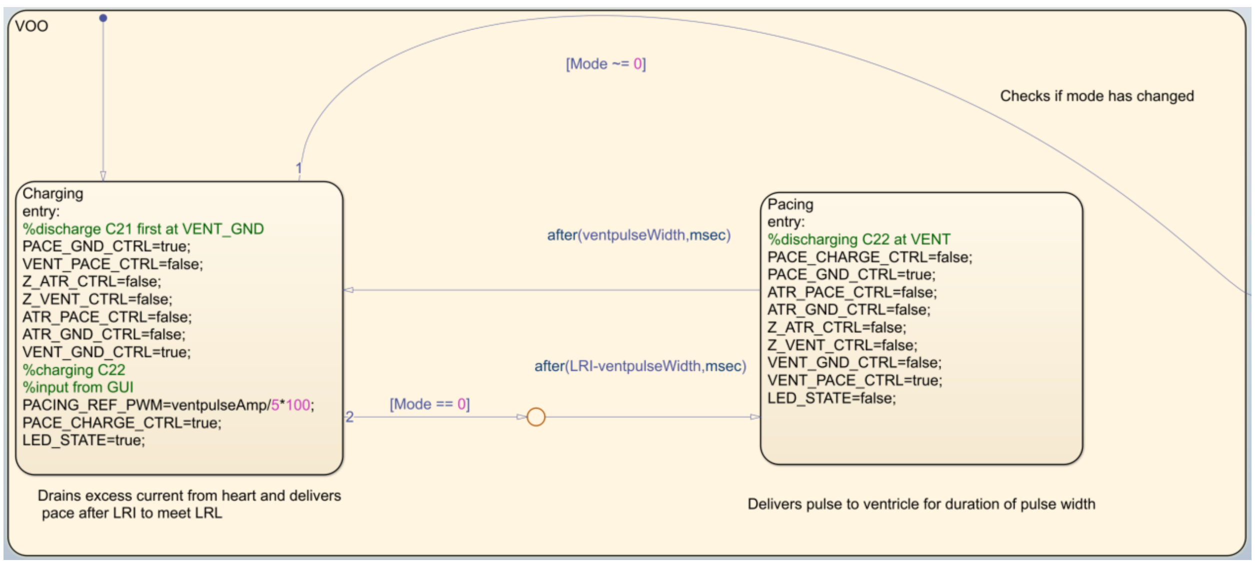


Figure 2.1.2.3.2 VOO Logic

Table 2.1.2.3.1: VOO FSM

State	Description	State Variables	Entry	Exit	Next State
Charging	Begins by discharging capacitor C21 into ventricle and initializing PACING_REF_PWM	PACE_GND_CTRL, VENT_PACE_CTRL, Z_ATR_CTRL, Z_VENT_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, VENT_GND_CTRL, PACING_REF_PWM, PACE_CHARGE_CTR, LED_STATE	Default After ventricle pulse is completed over a time period of ventpulseWidth	Mode == 0 (Ensures once cycle is completed before state change) Followed up by waiting LRI from the last pulses rising edge	Pacing Initializer

				Mode != 0	
Pacing	Discharges capacitor C22 into ventricle over ventpulseWidth	PACE_CHARGE_CTRL, PACE_GND_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, Z_ATR_CTRL, Z_VENT_CTRL, VENT_GND_CTRL, VENT_PACE_CTRL, LED_STATE	After LRI from the previous pulses rising edge	After delivering the pulse over ventpulseWidth	Charging

Seen below is an explanation of VOOR. The logic for VOOR is identical to VOO with the variable LRI being swapped for Adapted_Interval, which is the time period calculated for activity in the Rate Adaptive subsystem. Refer to 2.1.2.2 Rate Adaptive for an explanation on Adapted_Interval:

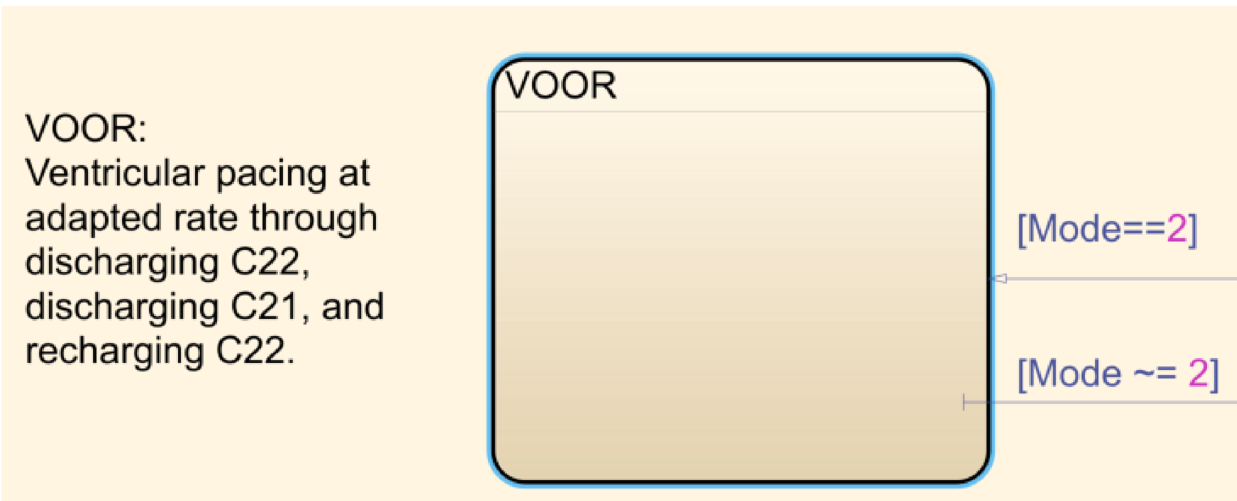


Figure 2.1.2.3.3 VOOR State

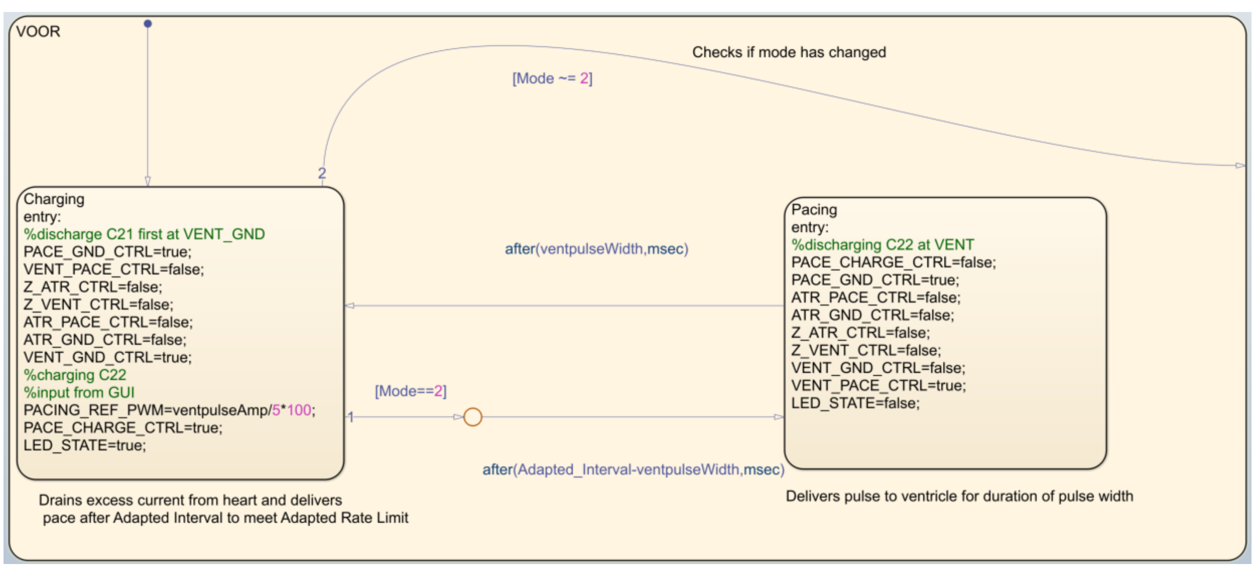


Figure 2.1.2.3 VOOR Logic

2.1.2.4 AOO/AOOR

Seen below is the state flow for AOO:

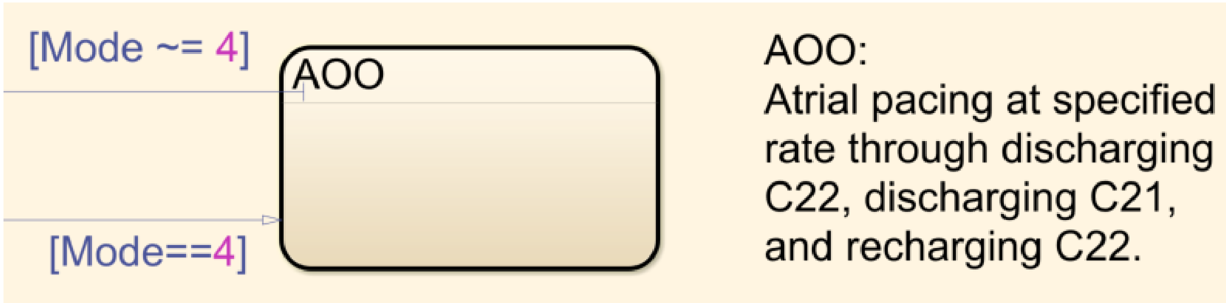


Figure 2.1.2.4.1 AOO State

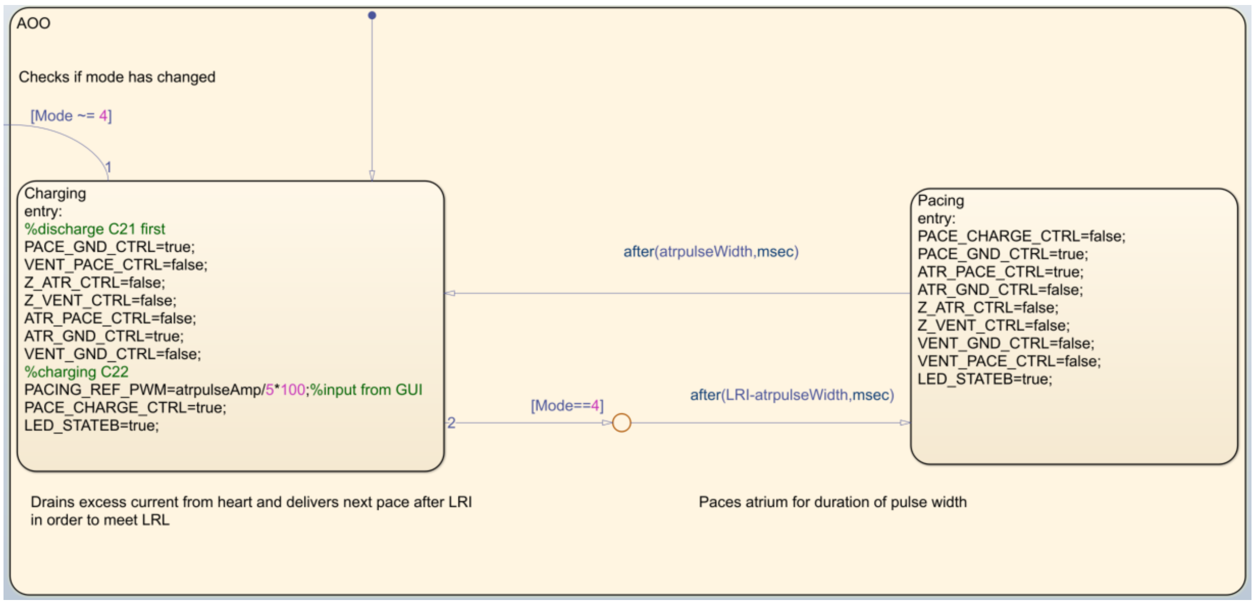


Figure 2.1.2.4.2 AOO Logic

Table 2.1.2.3.1: AOO FSM

State	Description	State Variables	Entry	Exit	Next State
Charging	Begins by discharging capacitor C21 into atrium and initializing PACING_REF_PWM	PACE_GND_CTRL, VENT_PACE_CTRL, Z_ATR_CTRL, Z_VENT_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, VENT_GND_CTRL, PACING_REF_PWM, PACE_CHARGE_CTR, LED_STATEB	Default After atrial pulse is completed over a time period of atrpulseWidth	Mode == 4 (Ensures once cycle is completed before state change) Followed up by waiting LRI from the last pulses rising edge Mode != 4	Pacing Initializer
Pacing	Discharges capacitor C22 into atrium over	PACE_CHARGE_CTRL, PACE_GND_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL,	After LRI from the previous pulses rising edge	After delivering the pulse over atrpulseWidth	Charging

	atrpulseWidth	Z_ATR_CTRL, Z_VENT_CTRL, VENT_GND_CTRL, VENT_PACE_CTRL, LED_STATEB			

Seen below is an explanation of AOOR. The logic for AOOR is identical to AOO with the variable LRI being swapped for Adapted_Interval, which is the time period calculated for activity in the Rate Adaptive subsystem. Refer to 2.1.2.2 Rate Adaptive for an explanation on Adapted_Interval:

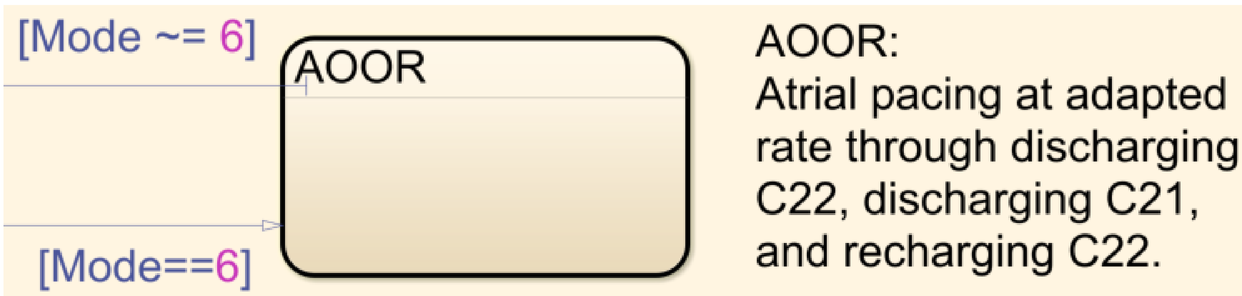


Figure 2.1.2.4.3 AOOR State

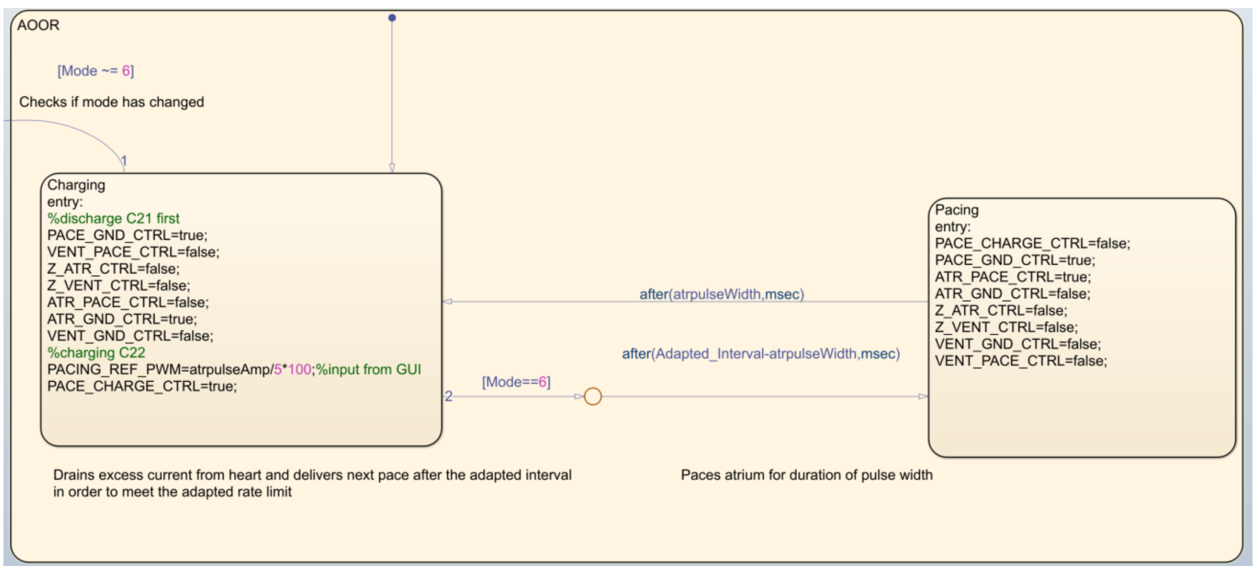


Figure 2.1.2.4.4 AOO Logic

2.1.2.5 VVI/VVIR

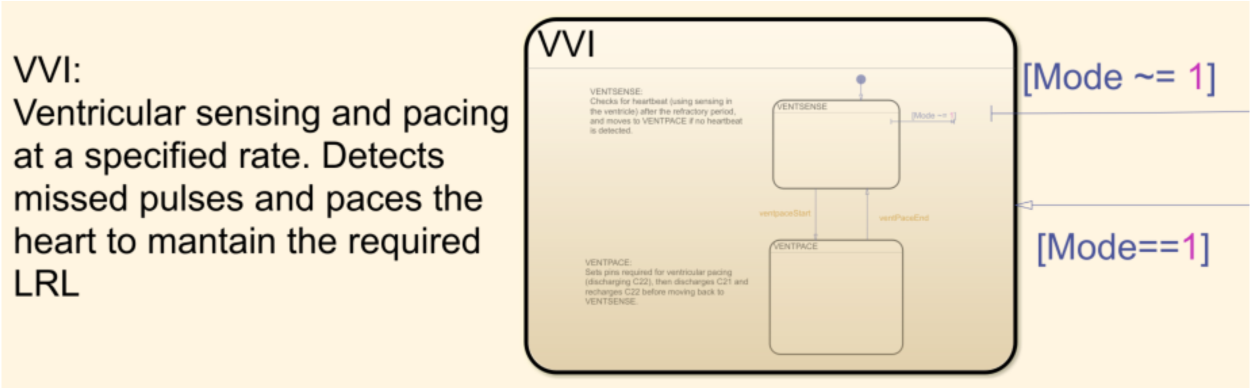


Figure 2.1.2.5.1 VVI State

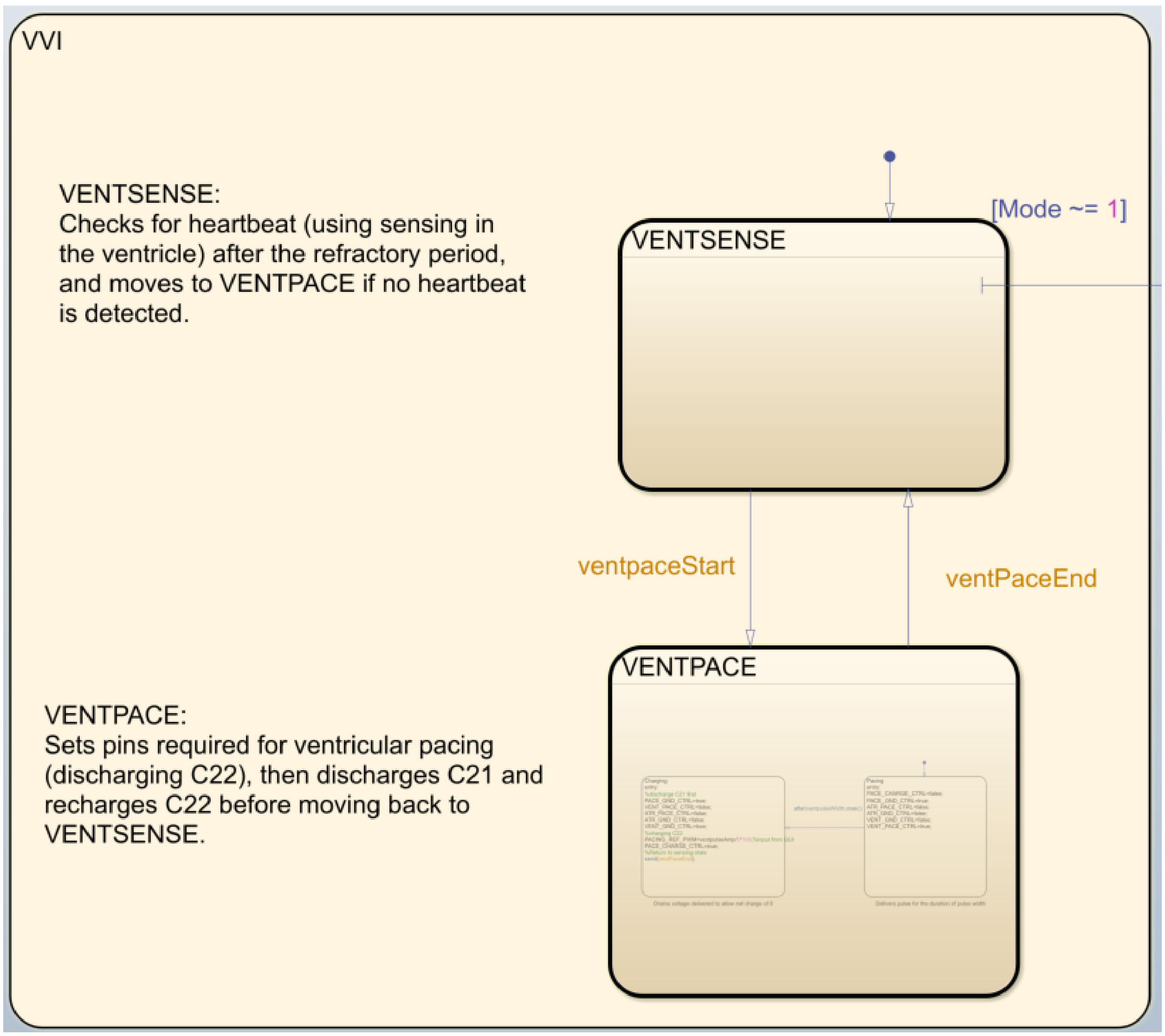


Figure 2.1.2.5.2 VVI Stateflow Logic

The sensing and pacing is split up into individual states. This allows for a more modular design. If any logic changes are necessary for sensing, they can be made without affecting the pacing. Furthermore, it ensures that the pace timing such as pulse width is completely independent of sensing meaning that the pulse is always delivered as intended improving patient safety.

The purpose of the orange transition labels is to improve readability and organization.

Table 2.1.2.5.1 Event Labels

Event	Purpose
ventpaceStart	A shortcut to the pacing state
ventPaceEnd	A shortcut back to the sensing state after pace is completed

Ventricular Sensing

Seen below is the state flow logic for ventricular sensing.

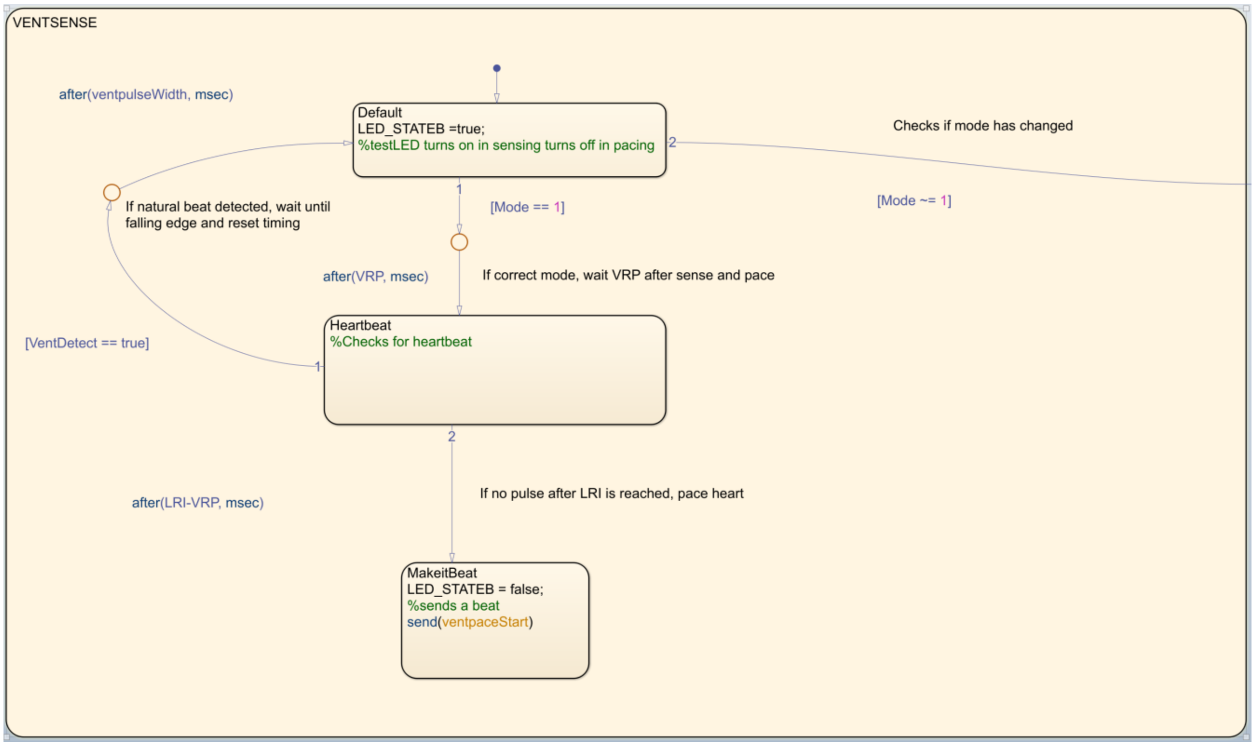


Figure 2.1.5.2.3 Ventricular Sensing

Table 2.1.5.2.2 Ventricular Sensing FSM

State	Description	State Variables	Entry	Exit	Next State
Default	Exists to continuously check mode change. Also used as neutral state while in VRP	LED_STATEB	Default Natural pace is detected before LRI since last event	Mode == 1 (Ensures at least once cycle is completed) Followed by VRP time period to ensure no actions in refractory period Mode != 1	Heartbeat Initializer

Heartbeat	Checks for natural pulse	N/A	After VRP	VentDetect == true (Followed by after ventpulseWidth to start VRP from falling edge of pulse) After LRI from last pulse (subtract VRP since already waited before entry)	Default MakeitBeat
MakeitBeat	Transitions to pacing state	LED_STATEB	After LRI-VRP or LRI from last pulses falling edge	Always exits to pacing state using “send(ventpaceStart)”	Pacing (In VENTPACE)

Ventricular Pacing

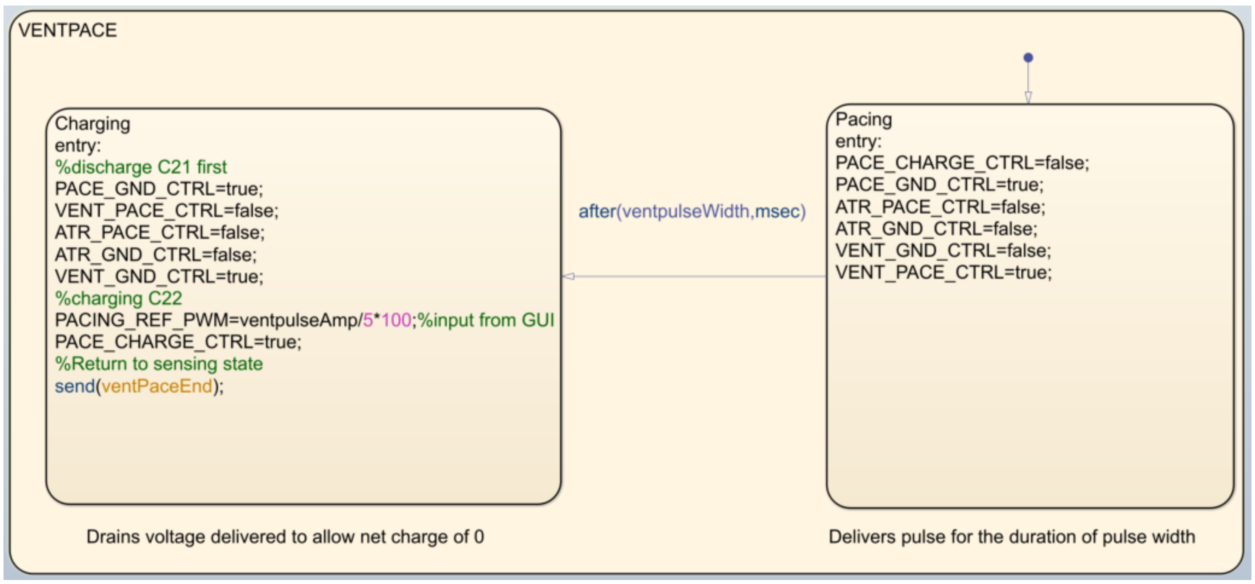


Figure 2.1.2.5.4 Ventricular Pacing

Table 2.1.5.2.3 Ventricular Pacing

State	Description	State Variables	Entry	Exit	Next State
Pacing	Discharges capacitor C22 into ventricle over ventpulseWidth	PACE_CHARGE_CTRL, PACE_GND_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, VENT_GND_CTRL, VENT_PACE_CTRL	Default	After delivering the pulse over ventpulseWidth	Charging
Charging	Begins by discharging capacitor C21 into ventricle and initializing PACING_REF_PWM	PACE_GND_CTRL, VENT_PACE_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, VENT_GND_CTRL, PACING_REF_PWM, PACE_CHARGE_CTR	After ventpulseWidth	Always exits by returning to sensing state via “send(ventPaceEnd)”	Default (In VENTSENSE)

Seen below is an explanation of VVIR. The logic for VVIR is identical to VVI with the variable LRI being swapped for Adapted_Interval, which is the time period calculated for activity in the Rate Adaptive subsystem. Refer to 2.1.2.2 Rate Adaptive for an explanation on Adapted_Interval:

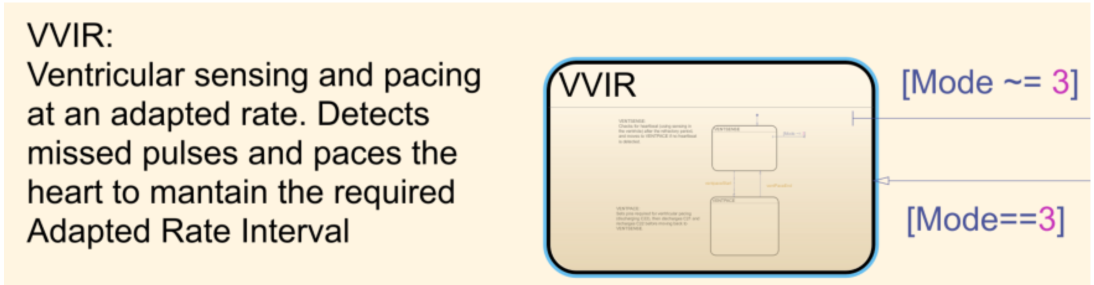


Figure 2.1.2.5.5 VVIR State

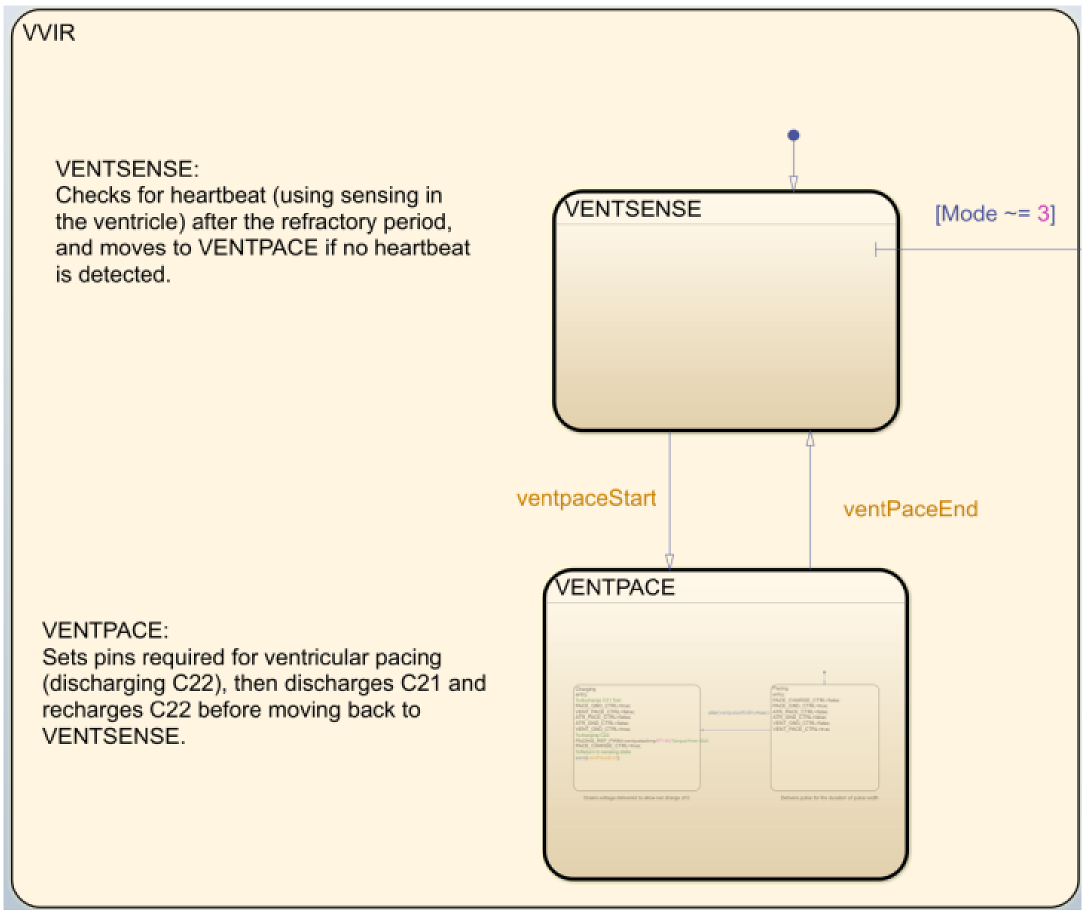


Figure 2.1.2.5.6 VVIR Stateflow Logic

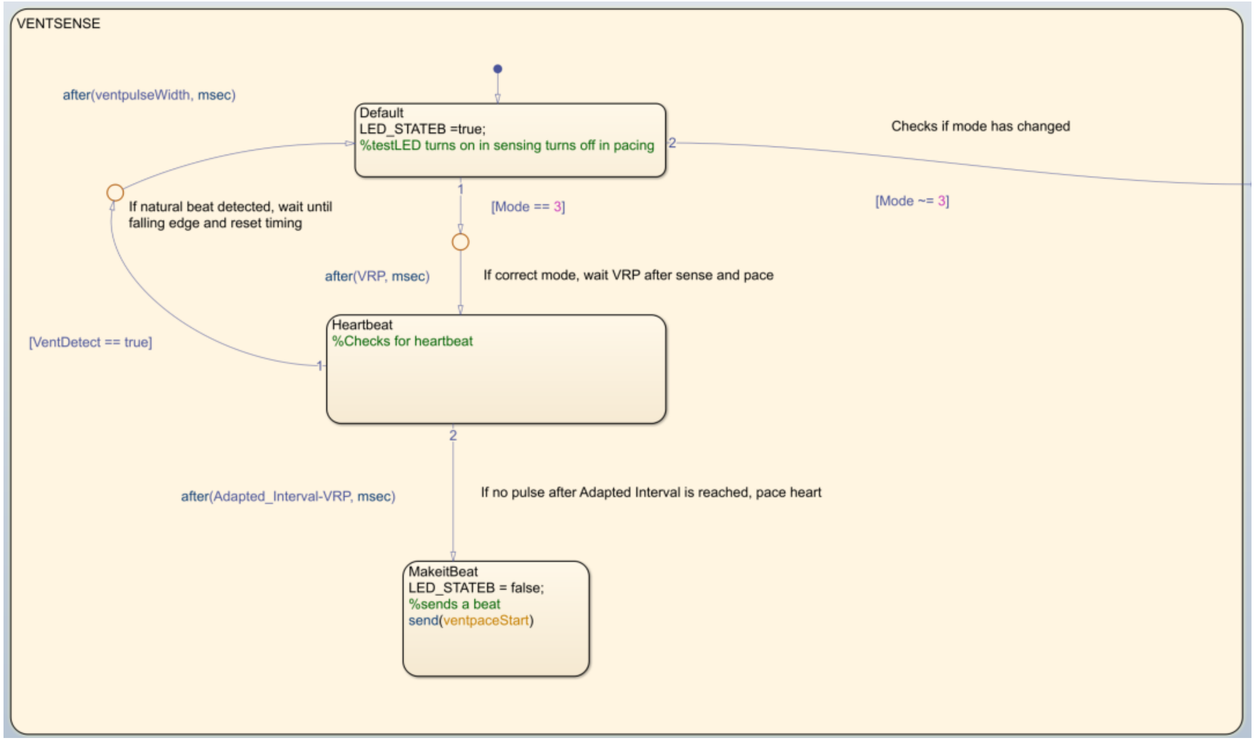


Figure 2.1.2.5.7 Ventricular Sensing (Rate Adaptive)

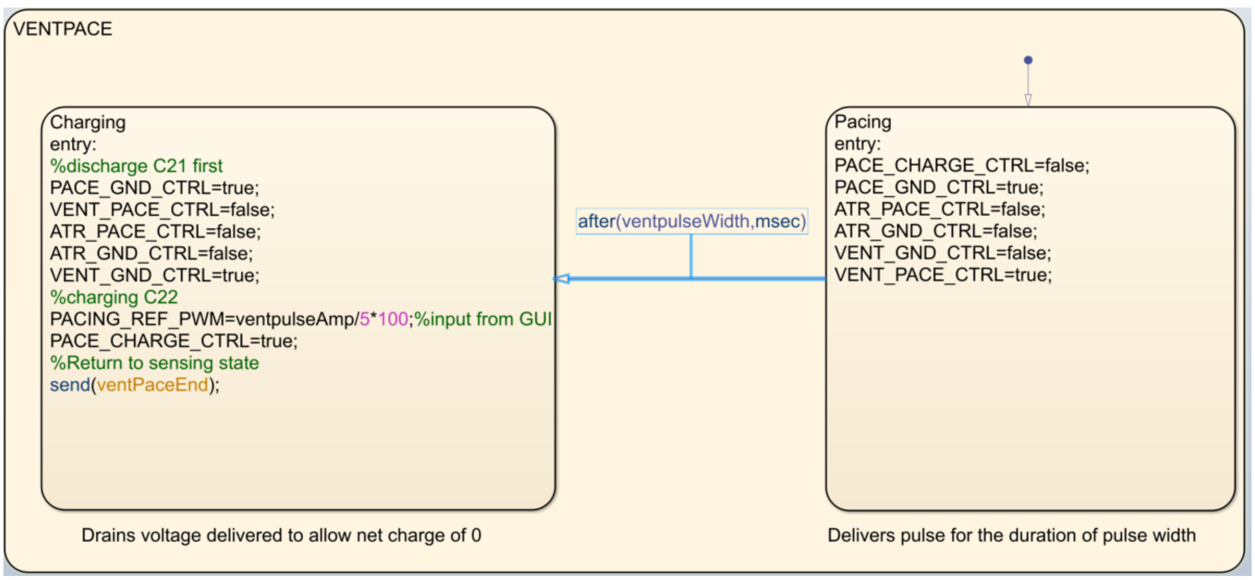


Figure 2.1.2.5.8 Ventricular Pacing (Rate Adaptive)

2.1.2.6 AAI/AAIR

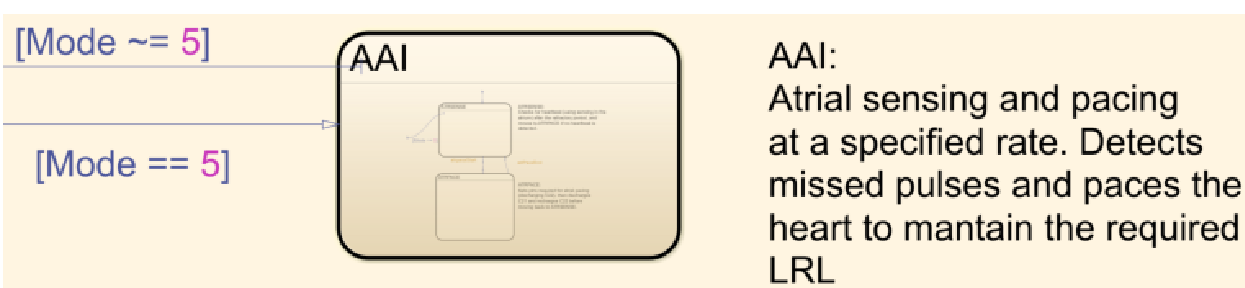


Figure 2.1.2.6.1 AAI State

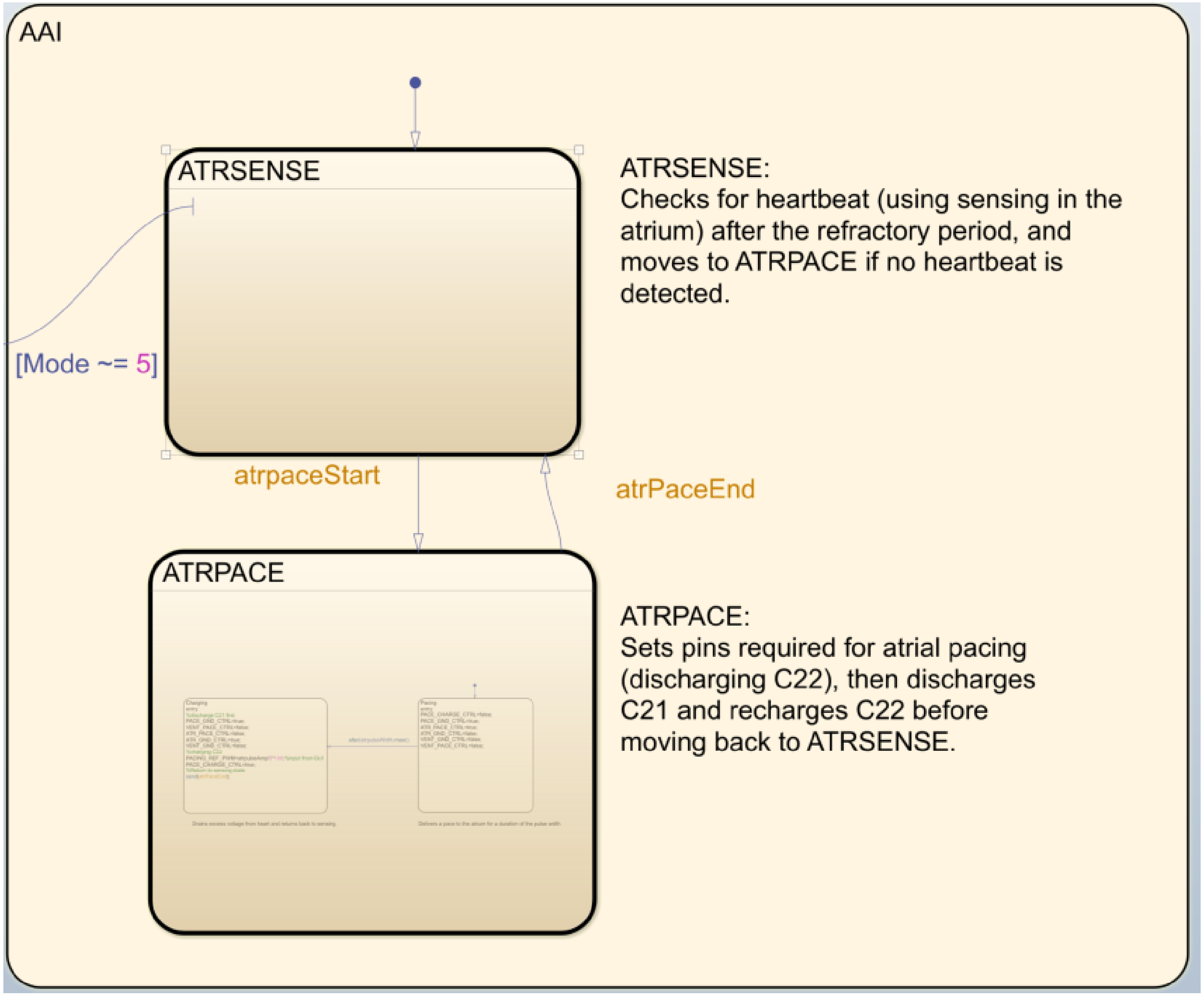


Figure 2.1.2.6.2 AAI Sateflow Logic

The sensing and pacing is split up into individual states. This allows for a more modular design. If any logic changes are necessary for sensing, they can be made without affecting the pacing. Furthermore, it ensures that the pace timing such as pulse width is completely independent of sensing meaning that the pulse is always delivered as intended improving patient safety.

The purpose of the orange transition labels is to improve readability and organization.

Table 2.1.2.6.1 Event Labels

Event	Purpose
atrpaceStart	A shortcut to the pacing state
atrPaceEnd	A shortcut back to the sensing state after pace is completed

Atrial Sensing

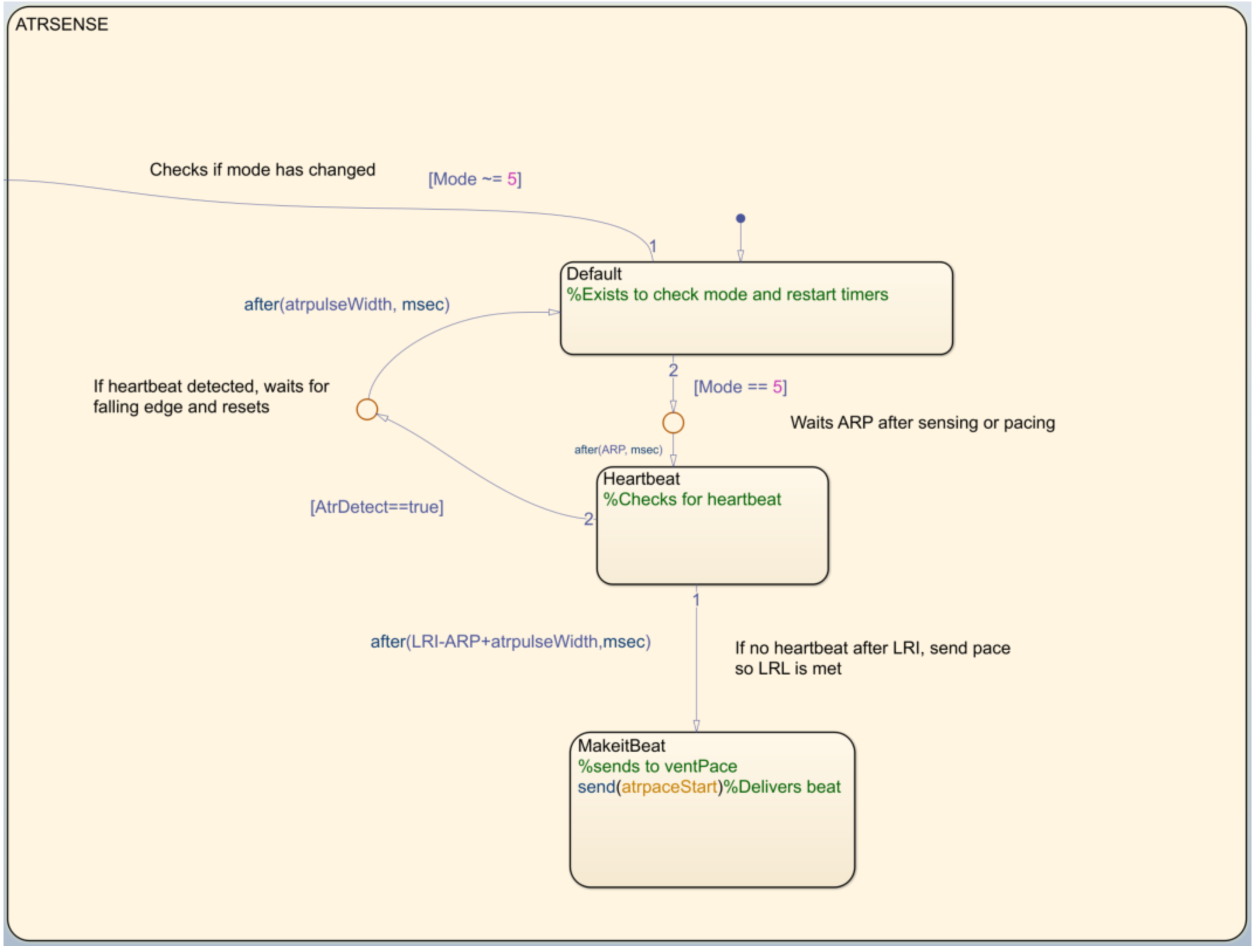


Figure 2.1.2.6.3 Atrial Sensing

Table 2.1.6.2.2 Atrial Sensing FSM

State	Description	State Variables	Entry	Exit	Next State
Default	Exists to continuously check mode change. Also used as neutral state while in ARP	N/A	Default Natural pace is detected before LRI since last event	Mode == 5 (Ensures at least once cycle is completed) Followed by ARPtime period to ensure no actions in refractory period Mode != 5	Heartbeat Initializer
Heartbeat	Checks for natural pulse	N/A	After ARP	AtrDetect == true (Followed by after ventpulseWidth to start ARP from falling edge of pulse)	Default MakeitBeat

				After LRI from last pulse (subtract ARP since already waited before entry) (Added atrpulseWidth due to timing issues of being off by a pulse width)	
MakeitBeat	Transitions to pacing state	N/A	After LRI-ARP+ atrpulseWidth or LRI from last pulses falling edge	Always exits to pacing state using “send(atrpaceStart)”	Pacing (In ARTPACE)

Atrial Pacing

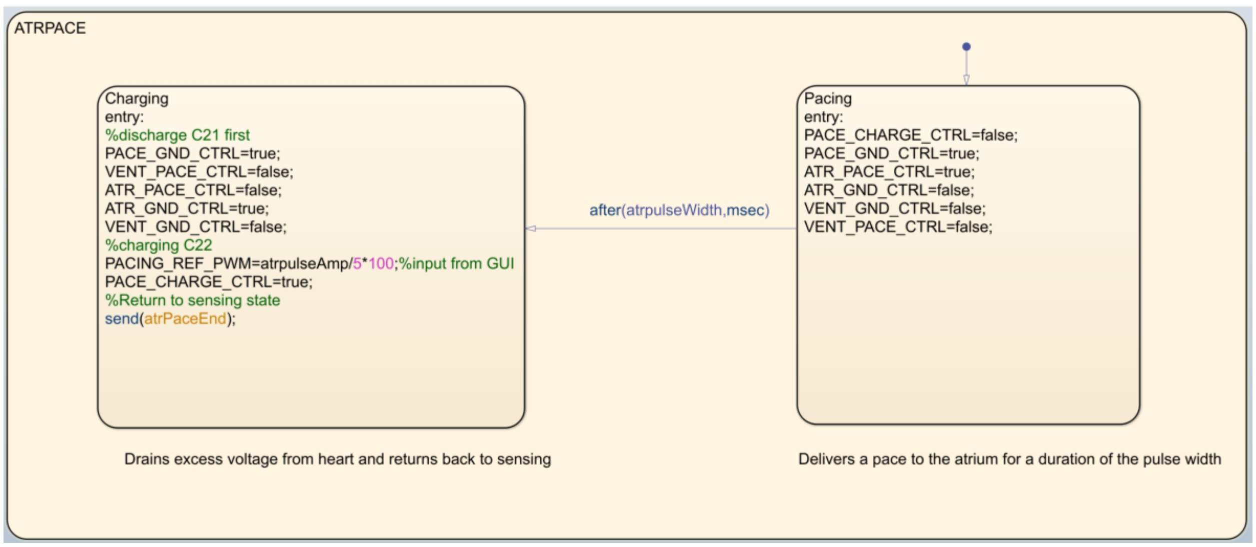


Figure 2.1.2.6.4 Atrial Pacing

Table 2.1.6.2.3 Atrial Pacing

State	Description	State Variables	Entry	Exit	Next State
Pacing	Discharges capacitor C22 into atrium over atrpulseWidth	PACE_CHARGE_CTRL, PACE_GND_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, VENT_GND_CTRL, VENT_PACE_CTRL	Default	After delivering the pulse over atrpulseWidth	Charging
Charging	Begins by discharging capacitor C21 into atrium and initializing PACING_REF_PWM	PACE_GND_CTRL, VENT_PACE_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, VENT_GND_CTRL, PACING_REF_PWM, PACE_CHARGE_CTR	After atrpulseWidth	Always exits by returning to sensing state via “send(atrPaceEnd)”	Default (In ATRSENSE)

Seen below is an explanation of AAIR. The logic for AAIR is identical to AAI with the variable LRI being swapped for Adapted_Interval, which is the time period calculated for activity in the Rate Adaptive subsystem. Refer to 2.1.2.2 Rate Adaptive for an explanation on Adapted_Interval:

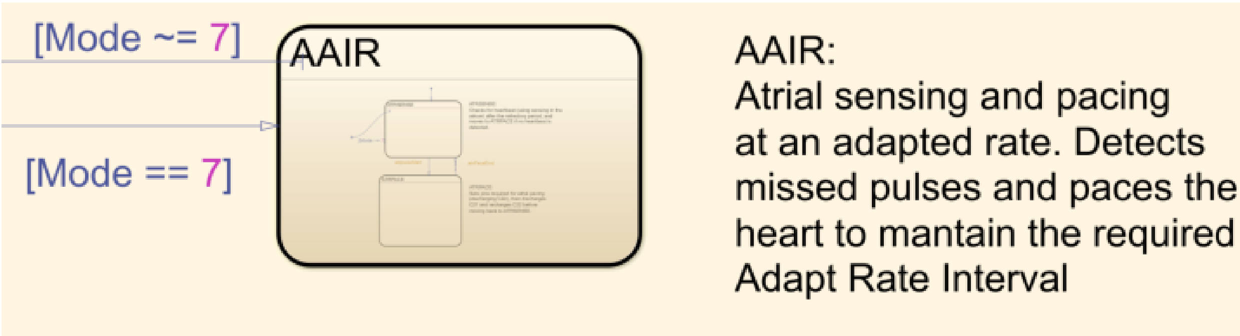


Figure 2.1.2.6.5 AAIR State

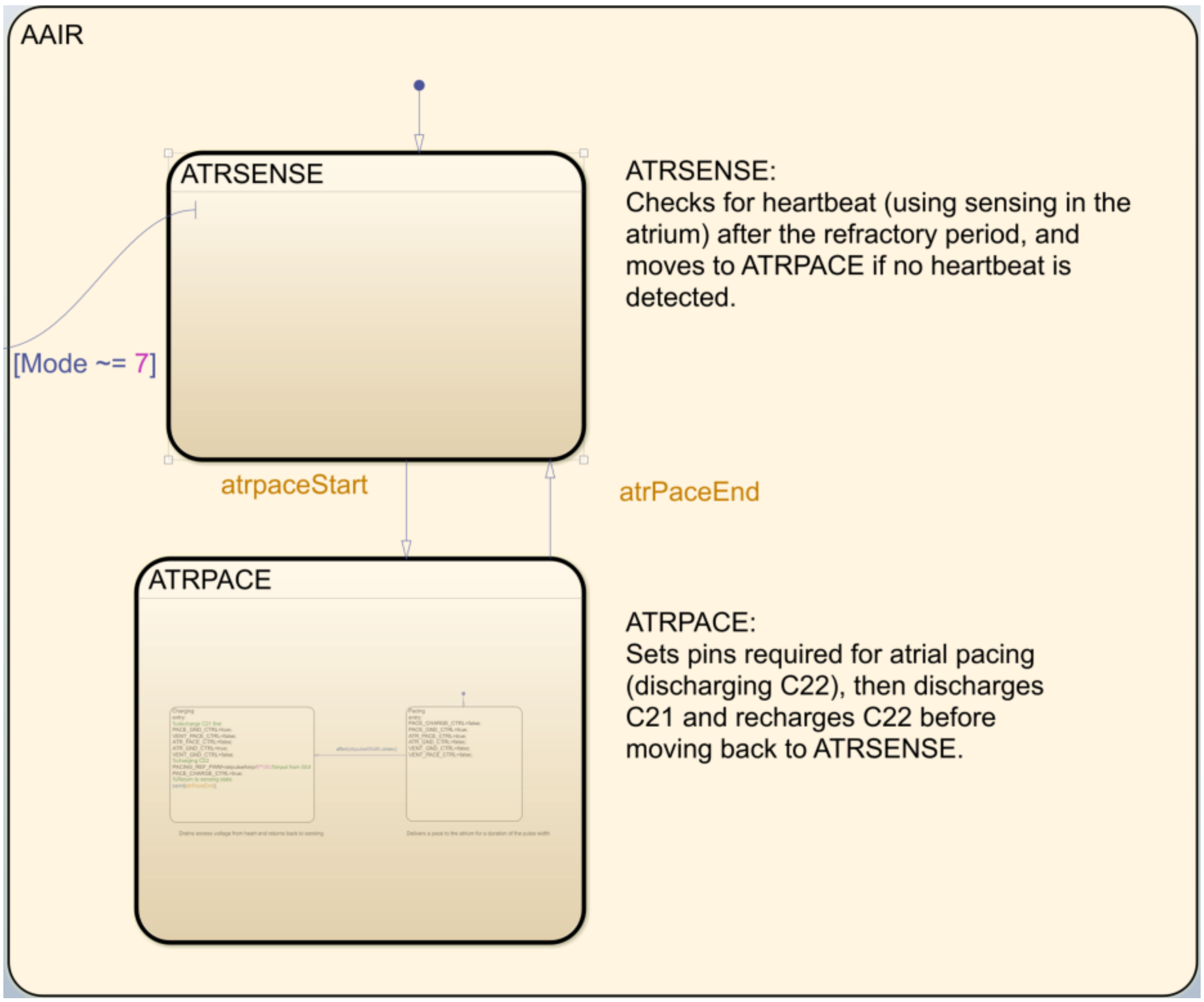


Figure 2.1.2.6.6 AAIR Stateflow Logic

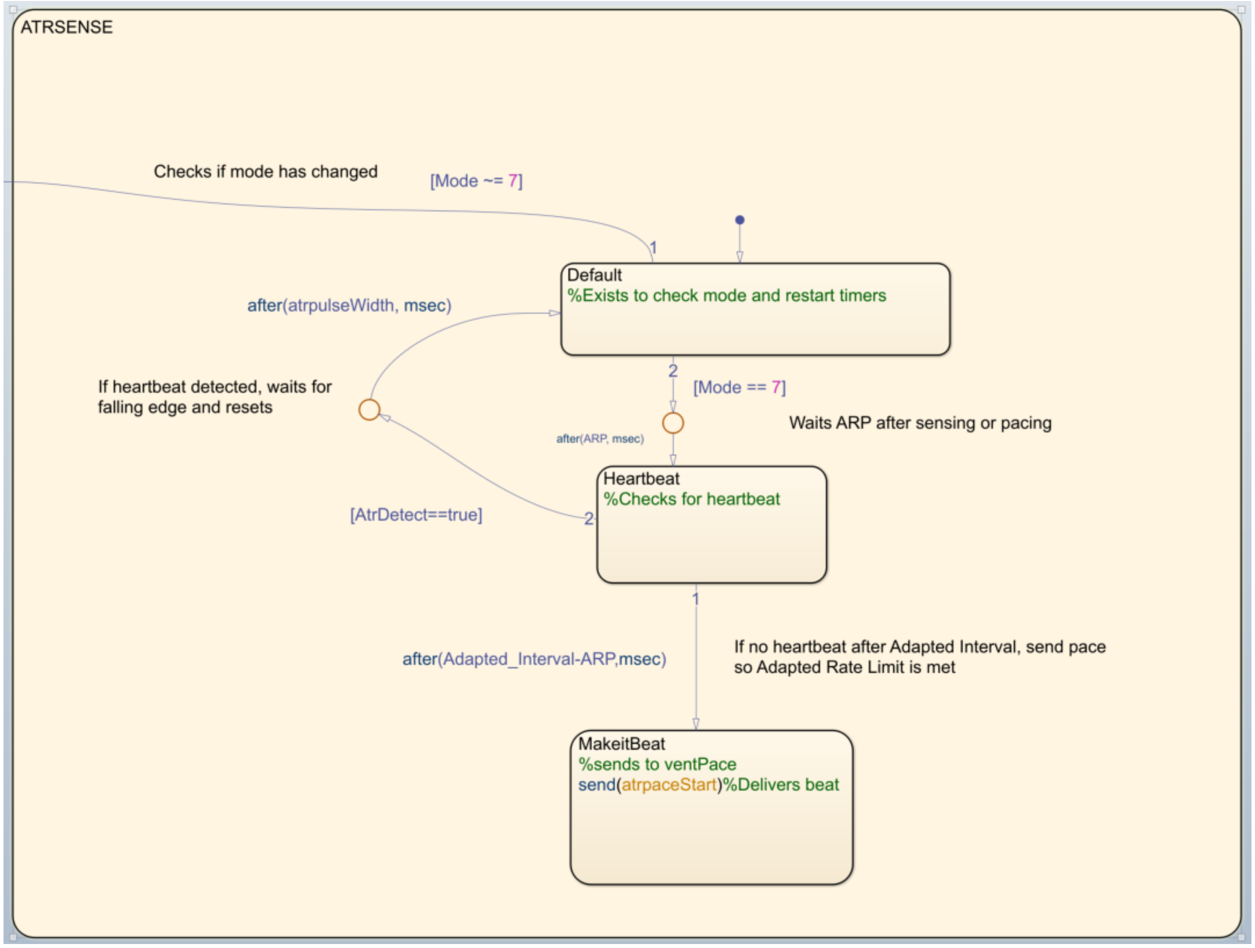


Figure 2.1.2.6.7 Atrial Sensing (Rate Adaptive)

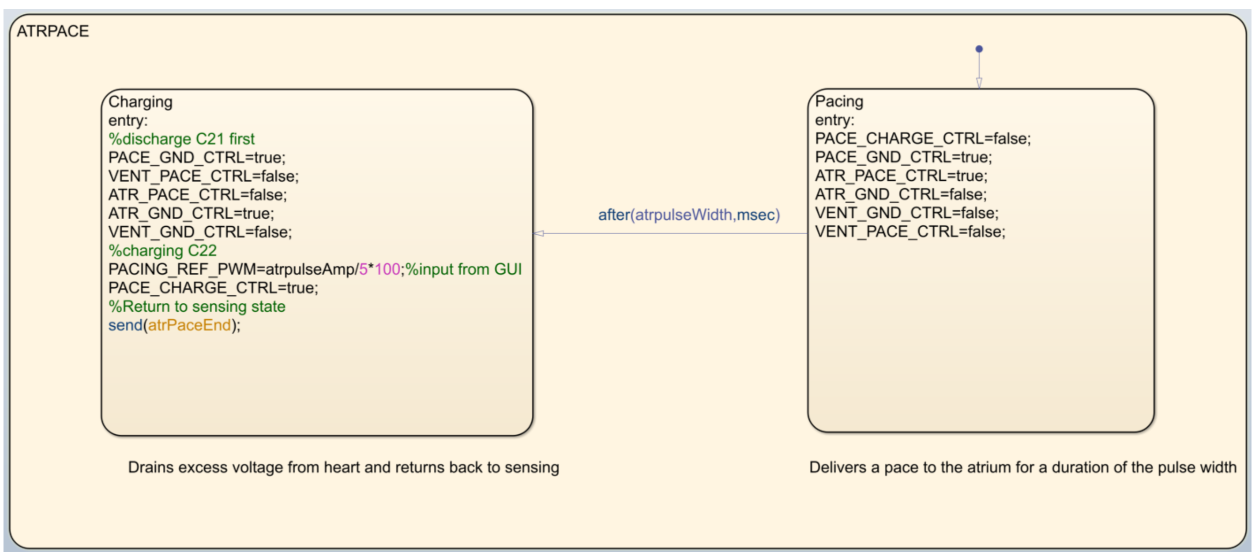


Figure 2.1.2.6.8 Atrial Pacing (Rate Adaptive)

2.1.2.7 DOO/DOOR

Seen below are the states for DOO along with a table explaining the FSM:

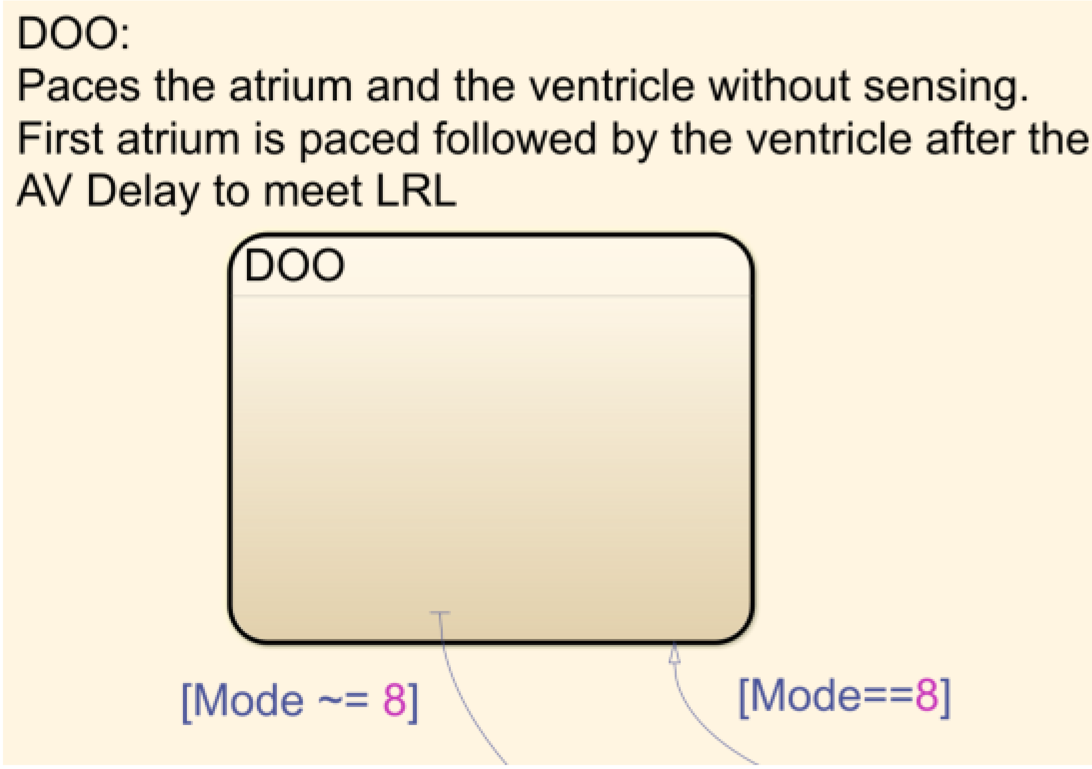


Figure 2.1.2.7.1 DOO State

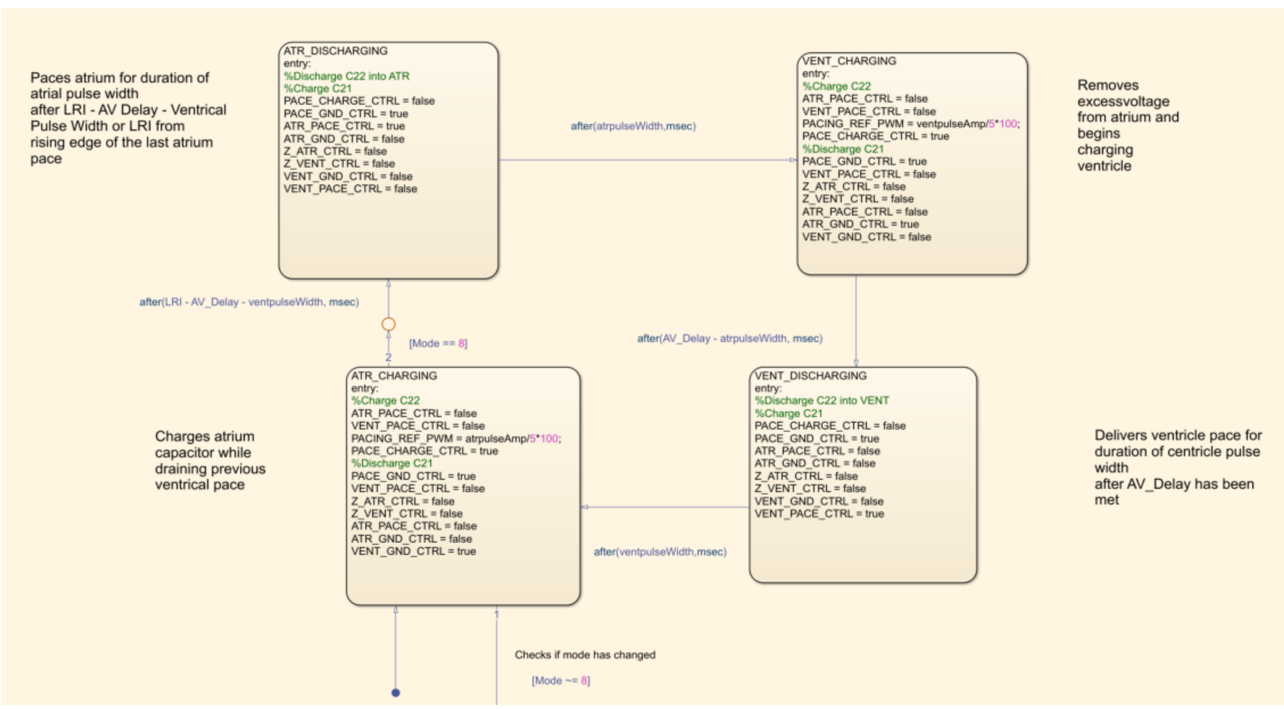


Figure 2.1.2.7.2 DOO Logic

Table 2.1.2.3.1: AOO FSM

State	Description	State Variables	Entry	Exit	Next State
ATR_CHARGING	Begins by discharging capacitor C21 into atrium and initializing PACING_REF_PWM	PACE_GND_CTRL, VENT_PACE_CTRL, Z_ATR_CTRL, Z_VENT_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, VENT_GND_CTRL, PACING_REF_PWM, PACE_CHARGE_CTR	Default After atrial pulse is completed over a time period of atrpulseWidth	Mode == 8 (Ensures once cycle is completed before state change) Followed up by waiting LRI from the last atrial pulses rising edge Mode != 8	ATR_DISCHARGING Initializer
ATR_DISCHARGING	Discharges capacitor C22 into atrium over atrpulseWidth	PACE_CHARGE_CTRL, PACE_GND_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, Z_ATR_CTRL, Z_VENT_CTRL, VENT_GND_CTRL, VENT_PACE_CTRL	After LRI from the previous atrial pulses rising edge	After delivering the pulse over atrpulseWidth	VENT_CHARGING
VENT_CHARGING	Begins by discharging capacitor C21 into atrium and initializing PACING_REF_PWM to charge C22	PACE_GND_CTRL, VENT_PACE_CTRL, Z_ATR_CTRL, Z_VENT_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, VENT_GND_CTRL, PACING_REF_PWM, PACE_CHARGE_CTR	After completing atrial pulse over atrpulseWidth	After AV_Delay period has passed (Subtract atrpulseWidth to count delay from riding edge)	VENT_DISHCARGING
VENT_DISCHARGING	Discharges capacitor C22 into ventricle over ventpulseWidth	PACE_CHARGE_CTRL, PACE_GND_CTRL, ATR_PACE_CTRL, ATR_GND_CTRL, Z_ATR_CTRL, Z_VENT_CTRL, VENT_GND_CTRL, VENT_PACE_CTRL	After AV_Delay from last atrial event	After delivering pulse over ventpulseWidth	ATR_CHARGING

To ensure correct logic, one cycle should be completed over LRI. By summing each after statement: (LRI-AV_Delay-ventpulseWidth) + (atrpulseWidth) + (AV_Delay - atrpulseWidth) + (ventpulseWidth) = LRI. This proves that the overall cycle is correctly timed.

Seen below are the diagrams of DOOR. The logic for DOOR is identical to DOO with the variable LRI being swapped for Adapted_Interval, which is the time period calculated for activity in the Rate Adaptive subsystem. Refer to 2.1.2.2 Rate Adaptive for an explanation on Adapted_Interval:

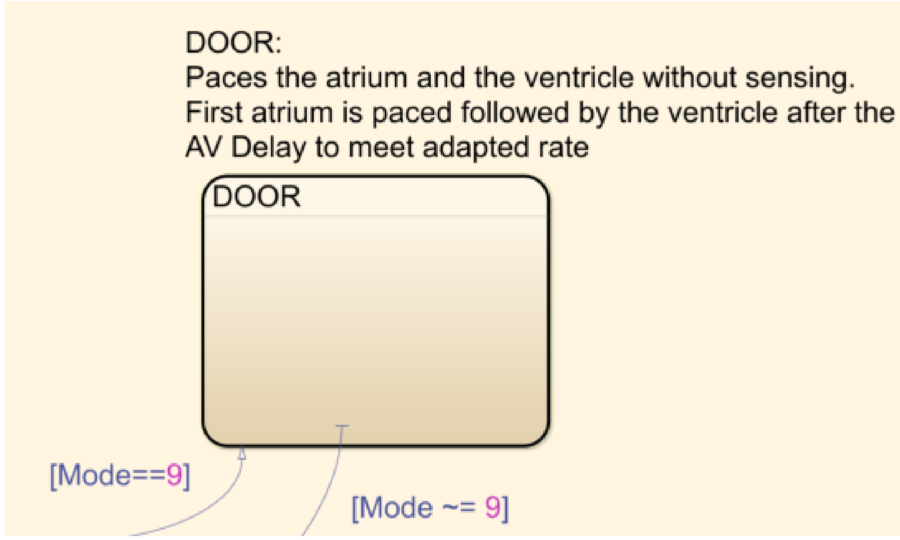


Figure 2.1.2.7.3 DOOR State

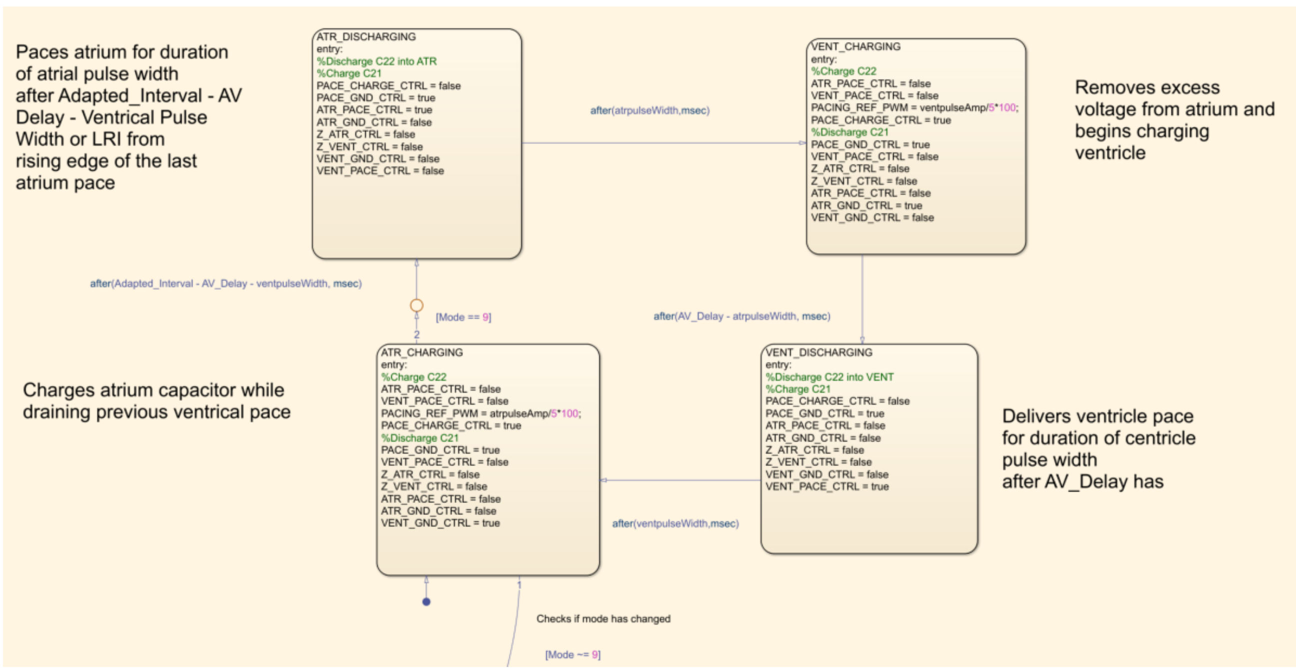


Figure 2.1.2.7.4 DOOR Logic

2.1.3 Hardware Outputs

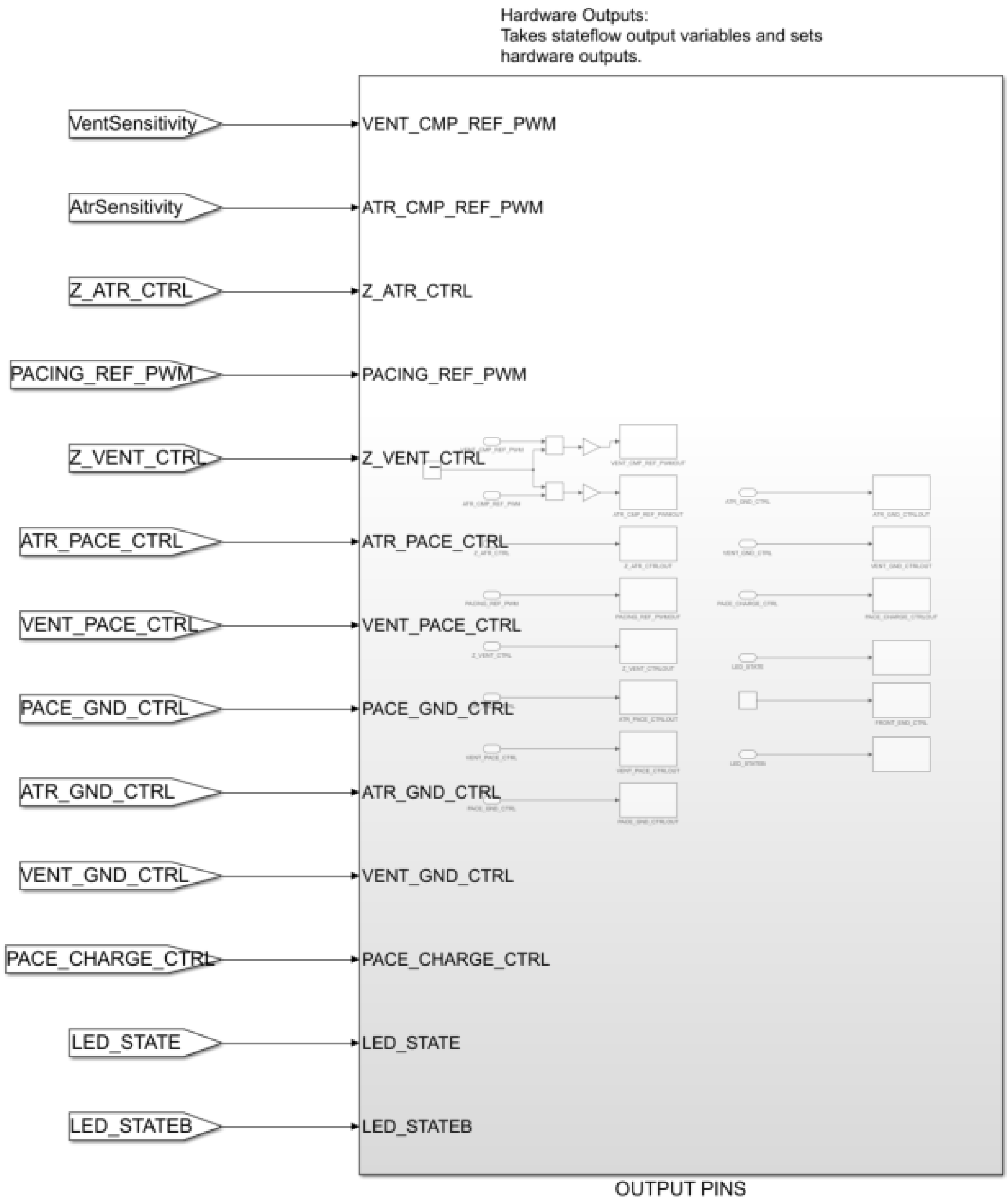


Figure 2.1.3.1 Hardware Outputs Subsystem

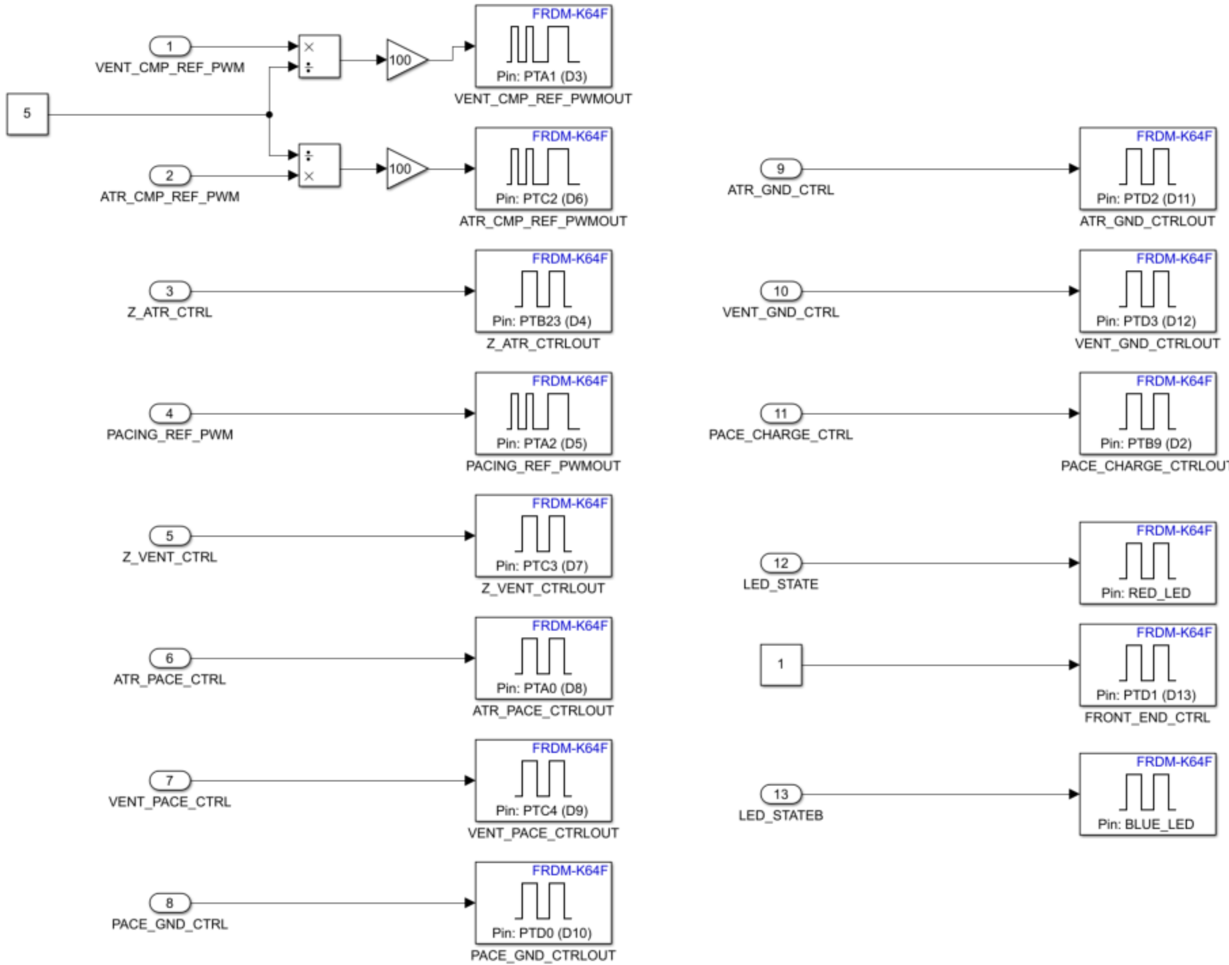


Figure 2.1.3.2 Hardware Output Pins

Table 2.1.3.1 Output Pins Parameters [1]

Parameter	Description	Data Type
VENT_CMP_REF_PWM	Pin sustains constant voltage for comparison for ventricle events. Divided by 5V and multiplied by 100 to get a duty cycle value	Double
ATR_CMP_REF_PWM	Pin sustains constant voltage for comparison for atrial events. Divided by 5V and multiplied by 100 to get a duty cycle value	Double
Z_ATR_CTRL	Connects impedance circuit to ring electrode of atrium	Boolean
PACING_REF_PWM	Used to charge C22 of the pacing circuit to the reference voltage. Calculated in Duty Cycles	Double
Z_VENT_CTRL	Connects impedance circuit to the ring	Boolean

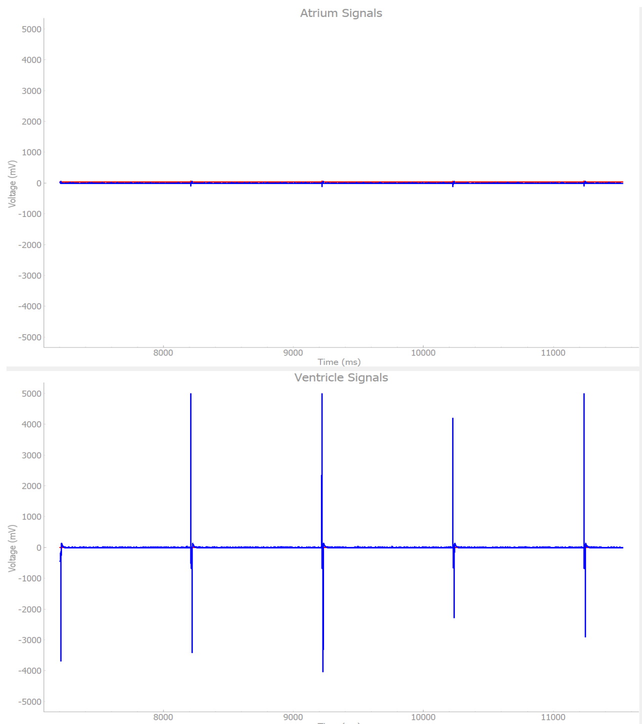
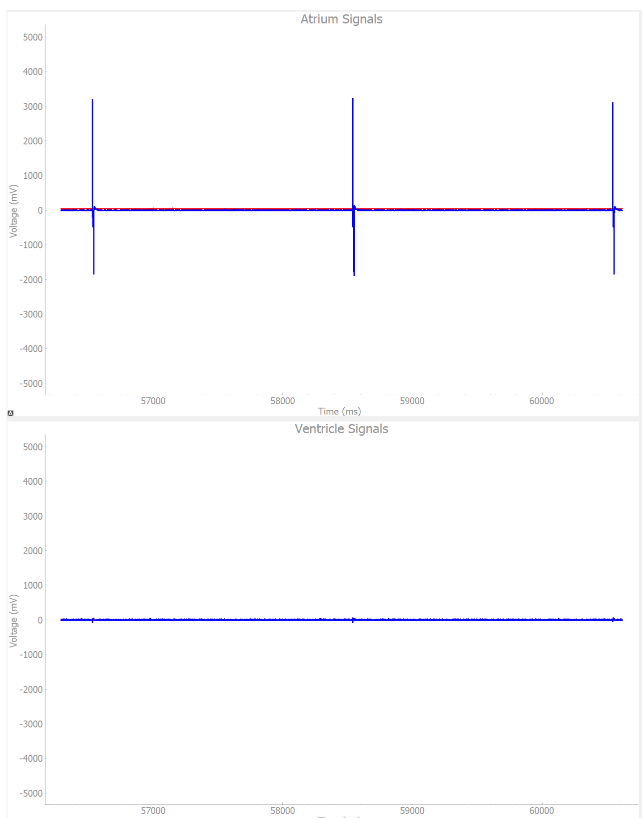
	electrode of the ventricle	
ATR_PACE_CTRL	Discharges primary capacitor into atrium	Boolean
VENT_PACE_CTRL	Discharges primary capacitor into atrium	Boolean
PACE_GND_CTRL	Allows current to flow from ring of both chambers	Boolean
ATR_GND_CTRL	Allows pulse to be discharged from atrium to prevent charge build up	Boolean
VENT_GND_CTRL	Allows pulse to be discharged from the ventricle to prevent charge build up	Boolean
PACE_CHARGE_CTRL	Start and stop charging of primary capacitor	Boolean
LED_STATE	Red LED for testing	Boolean
FRONT_END_CTRL	Used to activate sensing circuit	Constant (High/1)
LED_STATEB	Blue LED for testing	Boolean

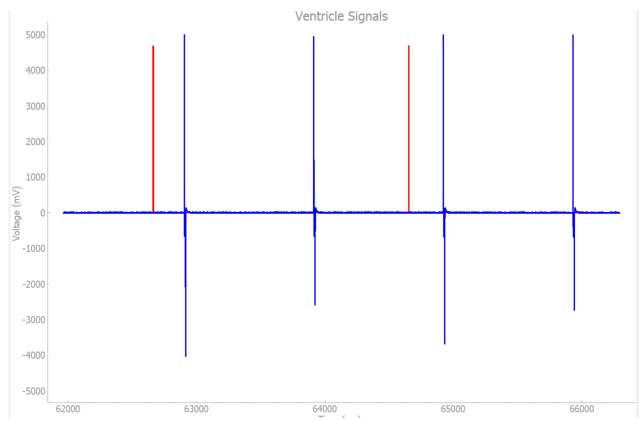
Part 3: Testing

VOO and AOO Testing

The testing for VOO and AOO can be combined since they have similar logic. Three test were completed, one with no natural pulse for each VOO and AOO to ensure the correct chamber is paced, then one with a natural pace in the ventricle for VOO to see if it is paced regardless of inputs.

Table 3.1: VOO and AOO Test Results

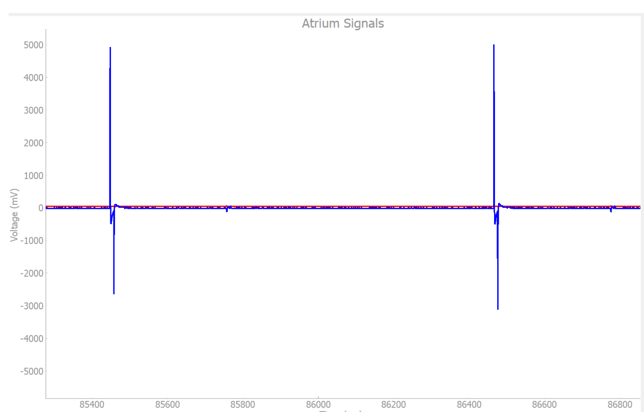
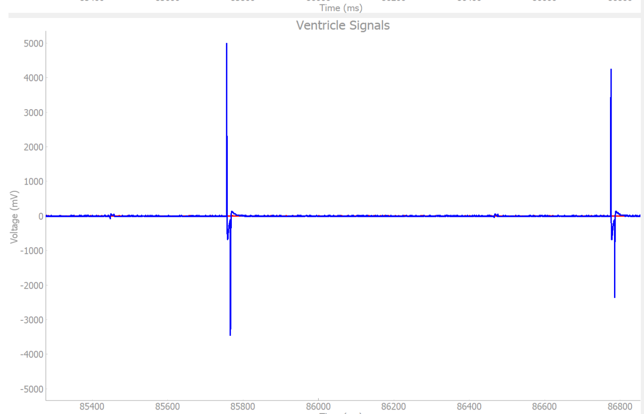
Settings	Expected Output	Actual Output	Result
Mode: VOO PPM: 60 Vent Pulse Amp: 5.0V Vent Pulse Width: 1 ms Heart: No Input	The ventricle is paced at 60 ppm		Pass
Mode: AOO PPM: 30 Atr Pulse Amp: 3.5V Atr Pulse Width: 1 ms Heart: No Input	The atrium is paced at 30 ppm		Pass

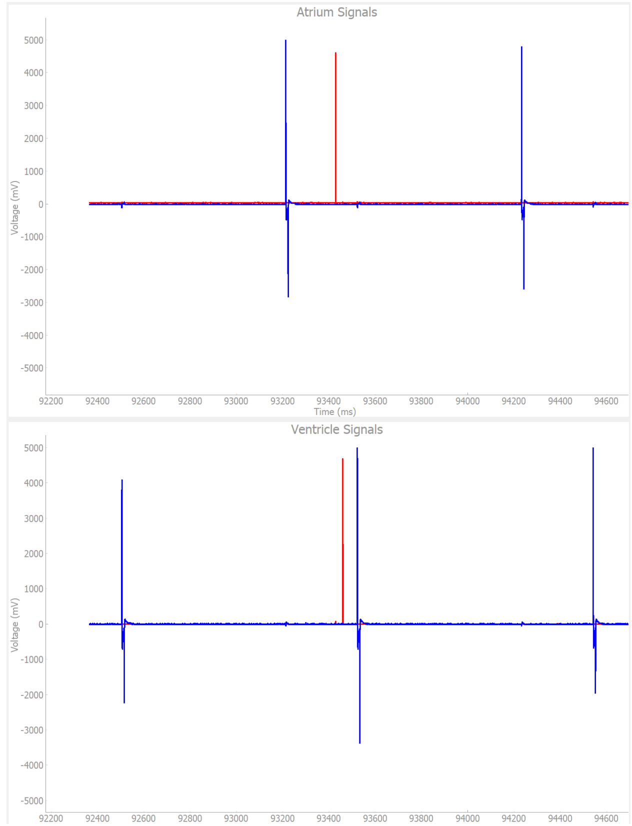
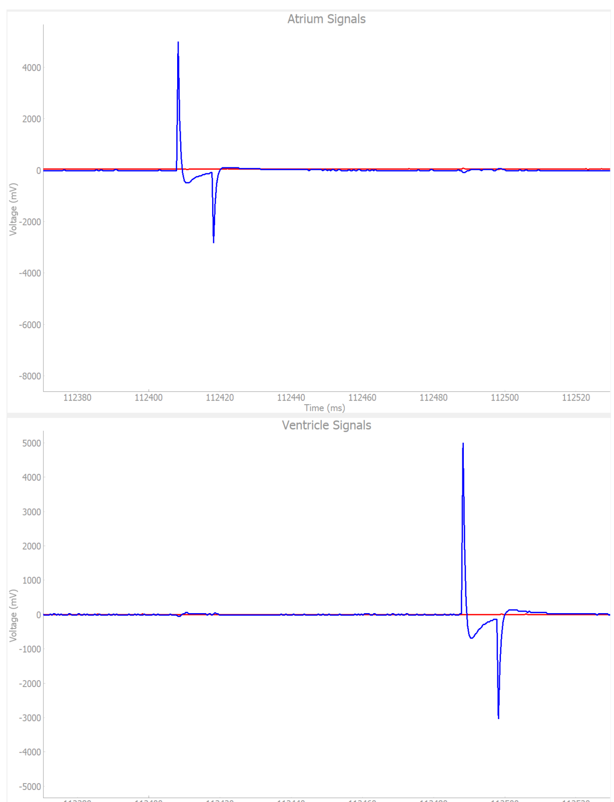
Mode: VOO PPM: 60 Vent Pulse Amp: 5.0V Vent Pulse Width: 1 ms Heart: 30 ppm	The ventricle is paced at 60 ppm		Pass

DOO Testing

To test DOO, three tests were attempted. One test to ensure the heart is paced at the correct rate, one to test the change in AV Delay, and one to ensure the mode does not react to natural pulses.

Table 3.2: DOO Test Results

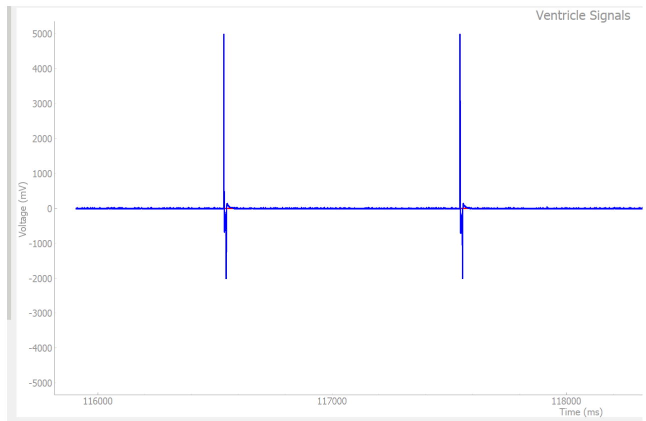
Settings	Expected Output	Actual Output	Result
Mode: DOO PPM: 60 ppm Pulse Amp: 5.0 V Pulse Width: 1 ms AV Delay: 300 ms Heart: No Input	Both chambers are paced at 60 ppm with an AV Delay		Pass

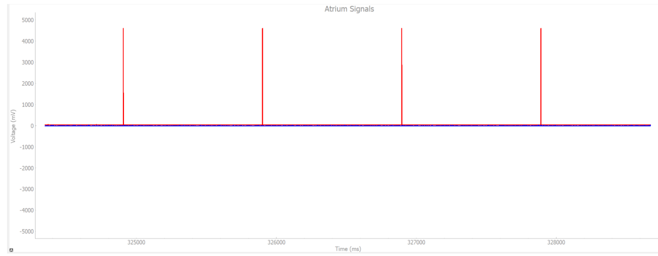
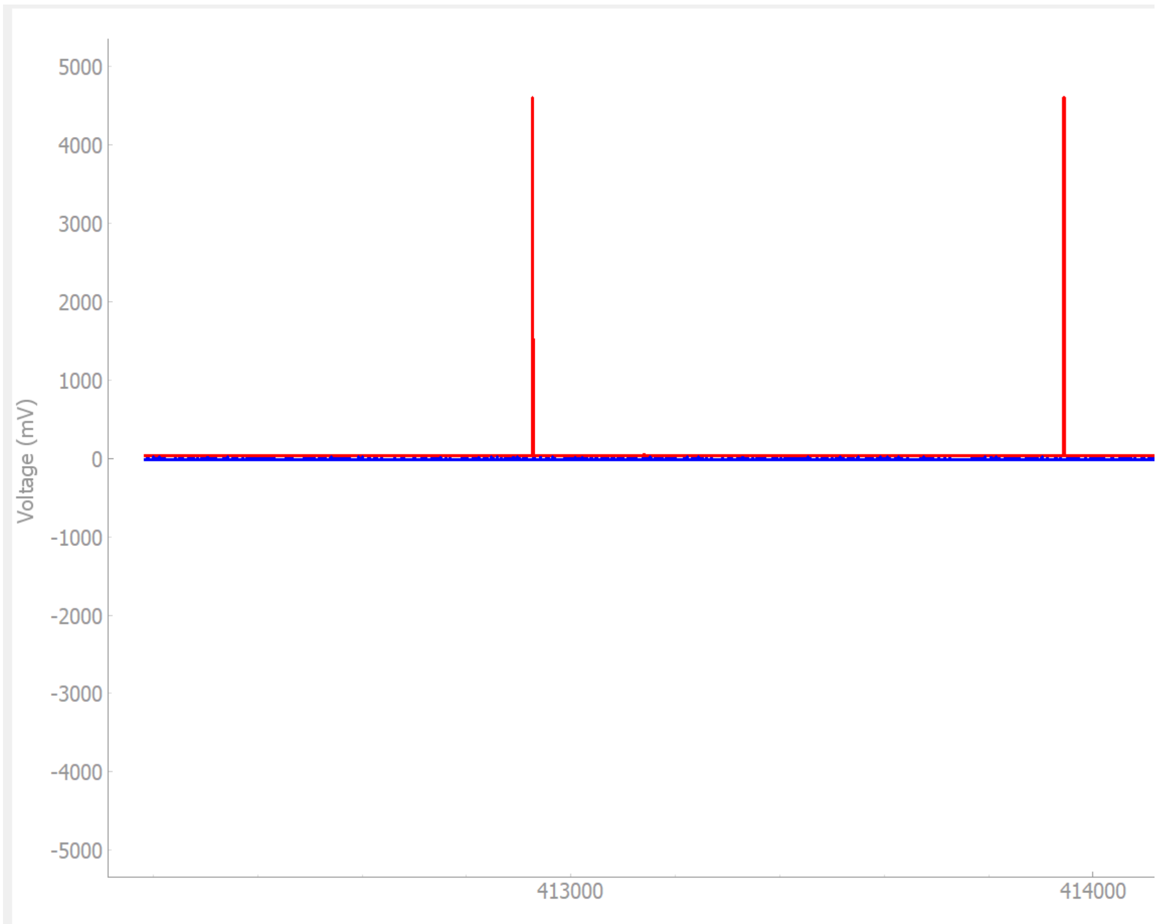
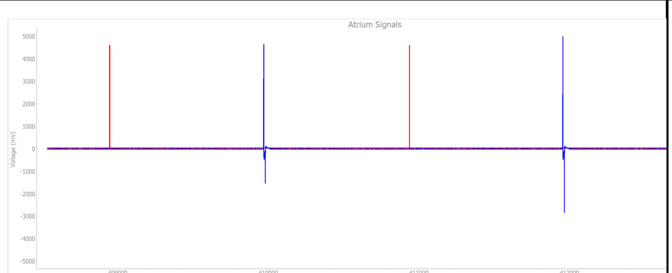
<table>
<tr>
<td>Mode: DOO
PPM: 60 ppm
Pulse Amp: 5.0 V
Pulse Width: 1 ms
AV Delay: 300 ms
Heart: 30 ppm</td>
<td>Both chambers are paced at 60 ppm with an AV Delay</td>
<td>
</td>
<td>Pass</td>
</tr>
<tr>
<td>Mode: DOO
PPM: 60 ppm
Pulse Amp: 5.0 V
Pulse Width: 1 ms
AV Delay: 70 ms
Heart: No Input</td>
<td>Both chambers are paced at 60 ppm with an AV Delay of 70 ms</td>
<td>
</td>
<td>Pass</td>
</tr>
</table>

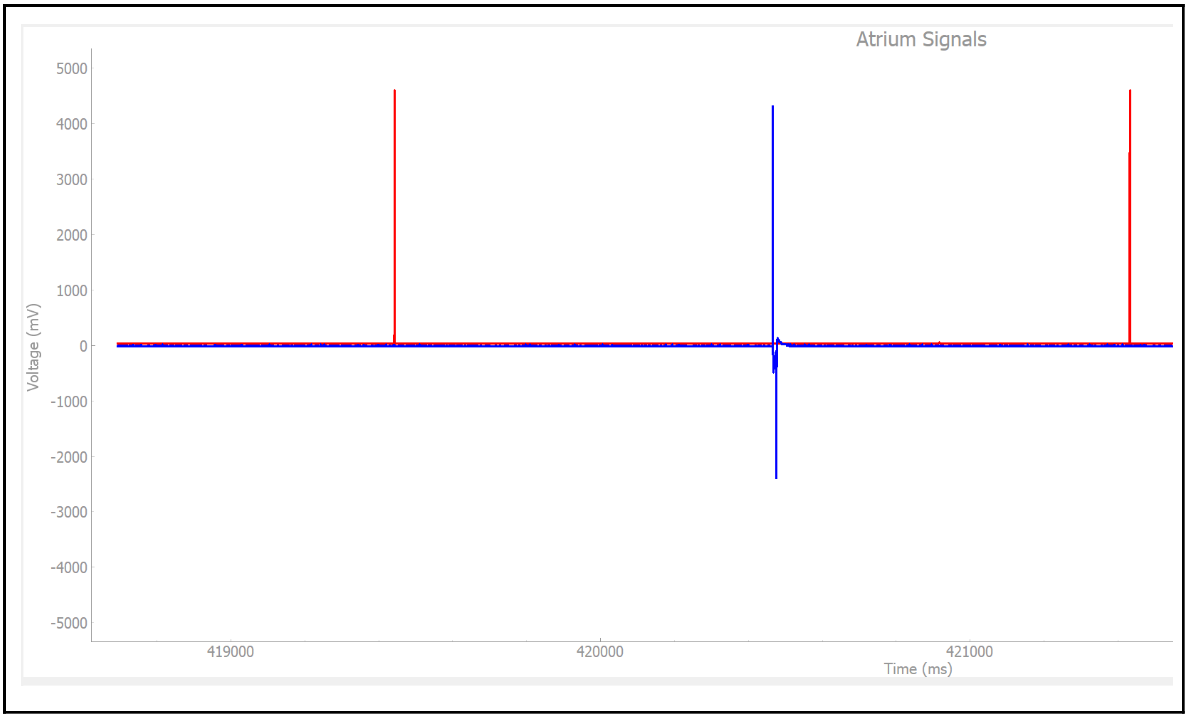
VVI and AAI Testing

VVI and AAI are fairly similar hence both can be tested together. There will be 3 tests done. One when the heart does not pace at all, one where the heart paces at the LRI, and one where the heart paces at half the LRI. This will test when Heart >= LRI, Heart == 0, and Heart < LRI.

Tabel 3.2.3 VVI and AAI Test Results

Settings	Expected Output	Actual Output	Result
Mode: VVI PPM: 60 Pulse Amp: 5.0V Pulse Width: 1 ms Heart: No Input	Ventricle is paced at 60 ppm		Pass

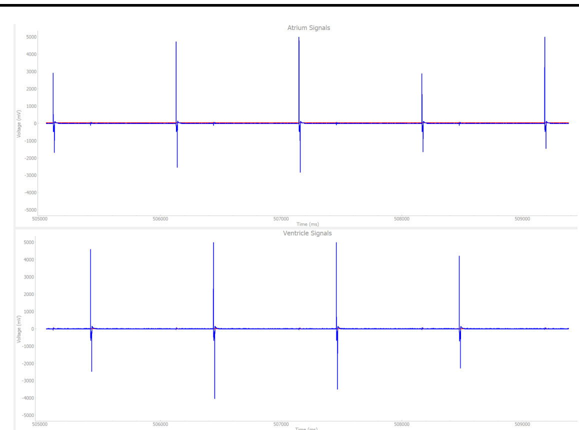
<table>
<tr>
<td>Mode: AAI
PPM: 60
Pulse Amp: 5.0V
Pulse Width: 1 ms
Heart: 60 ppm</td>
<td>Atrium is not paced at all</td>
<td>
</td>
<td>Pass</td>
</tr>
<tr>
<td colspan="4">
</td>
</tr>
<tr>
<td>Mode: AAI
PPM: 60
Pulse Amp: 5.0V
Pulse Width: 1 ms
Heart: 60 ppm</td>
<td>Atrium is paced every other natural pace</td>
<td>
</td>
<td>Pass</td>
</tr>
</table>



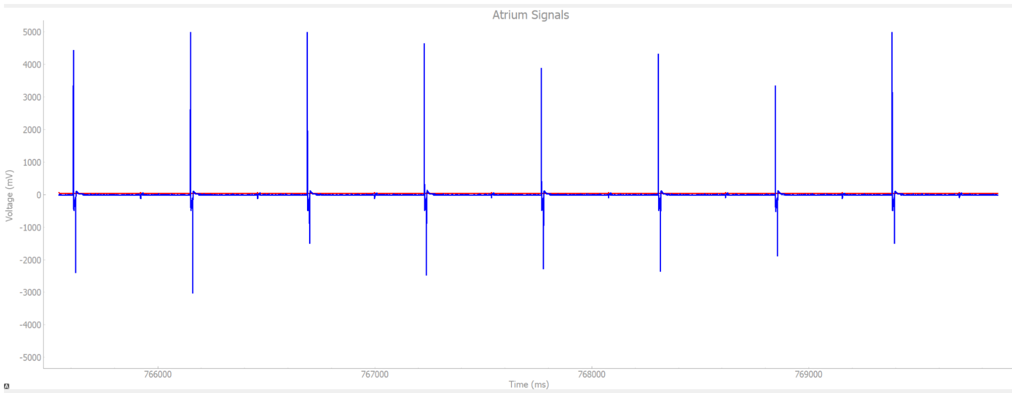
VOOR, AOOR, and DOOR Testing

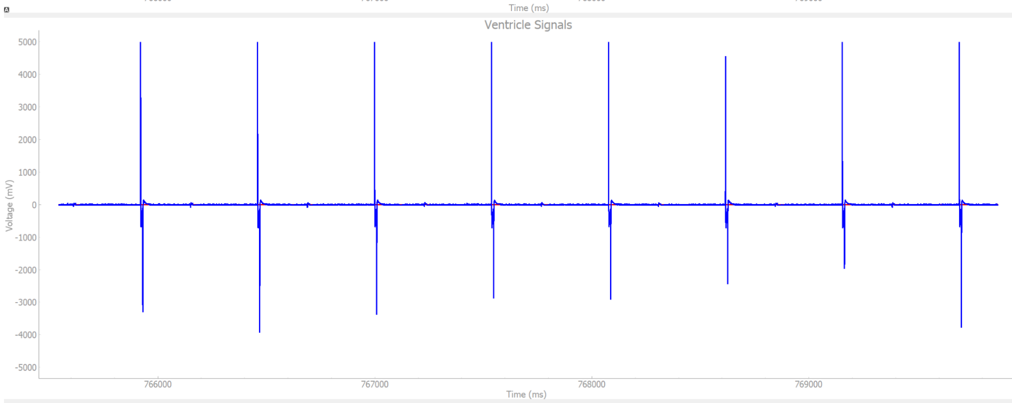
Three test cases will be completed using non sensing rate adaptive modes. The first test will begin at a steady state and then the board will be shaken for a few minutes and then set down for a 2 minute recovery time, the second test is the same but with a natural heart beat to ensure it does not impact rate adaptation, and one where URL is much smaller than MSR. This will test reaction time, recovery time, URL taking priority overMSR, and reaction to natural pulses.

Table 3.4 VOOR, AOOR, DOOR Test Results

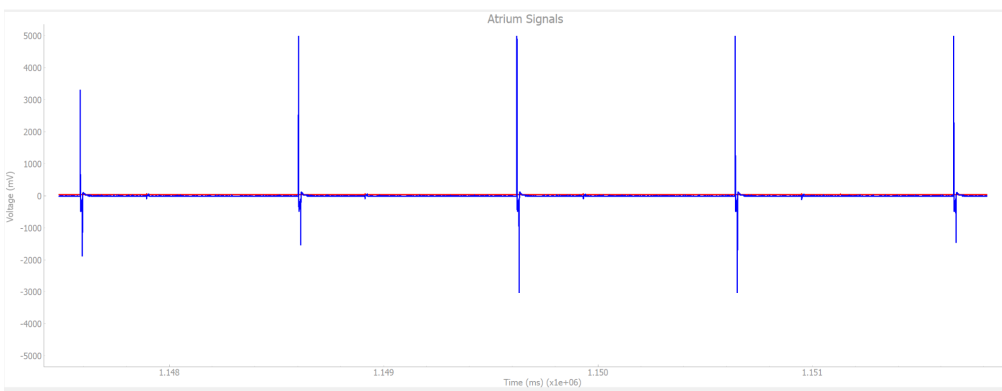
Settings	Expected Output	Actual Output	Result
Mode: DOOR LRL: 60 URL: 140ppm MSR: 120 ppm AV Delay: 300 ms Response Factor: 16 Threshold: V-Low Reaction TIme: 30 s Recovery Time: 2 min Heart: No Input	Heart Paces at 60 ppm, increases to 120 ppm during shaking, recovered to 60 ppm after 2 mins		Pass

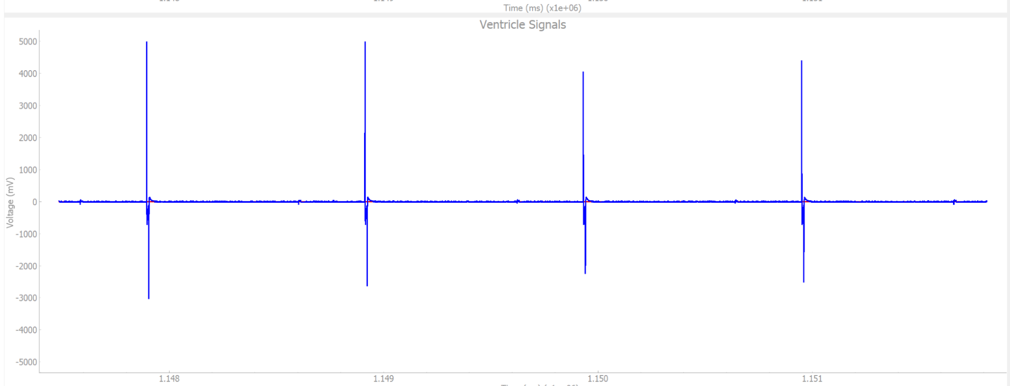
With Activity:

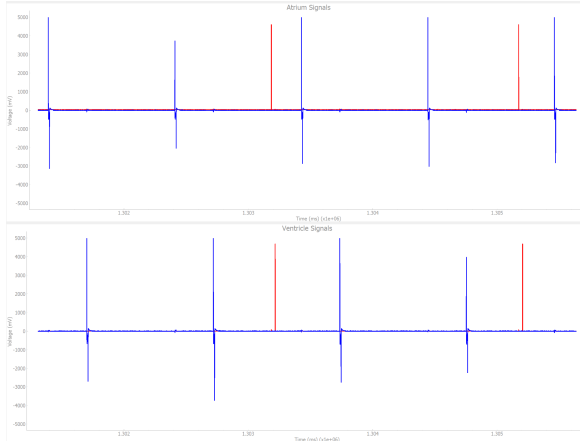
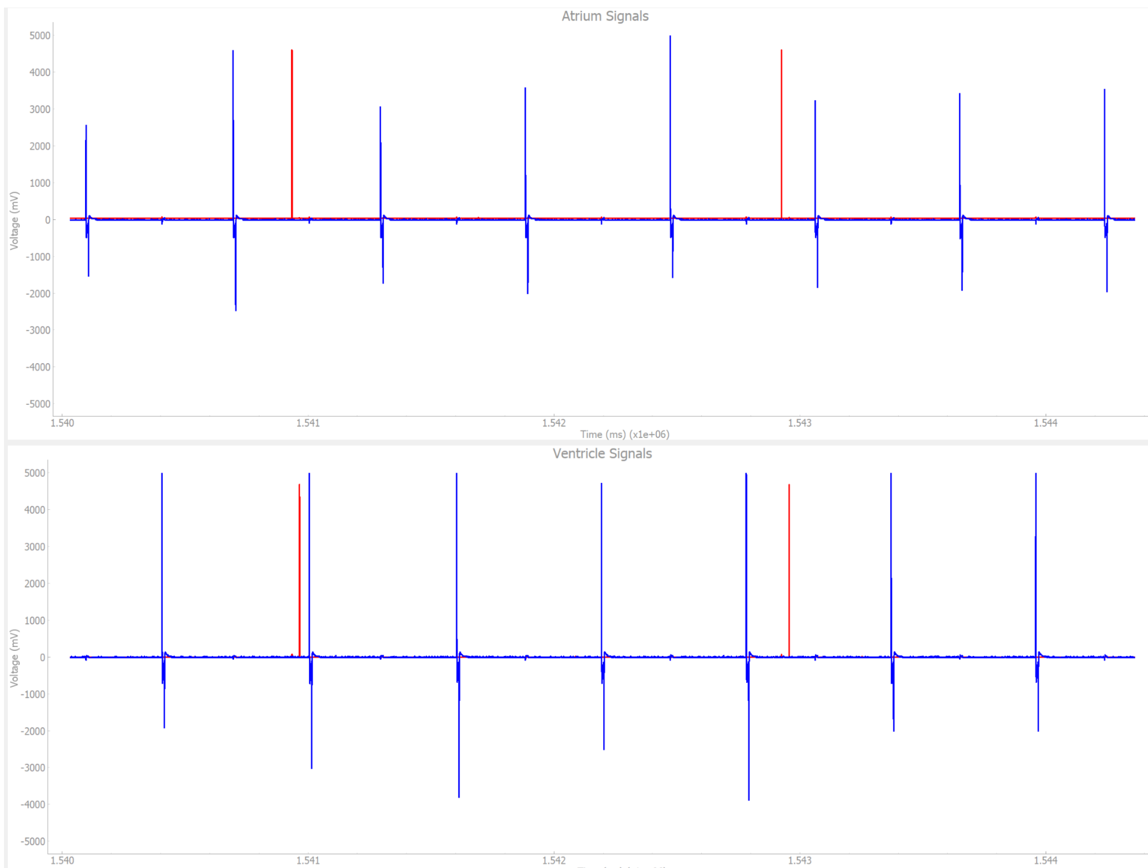




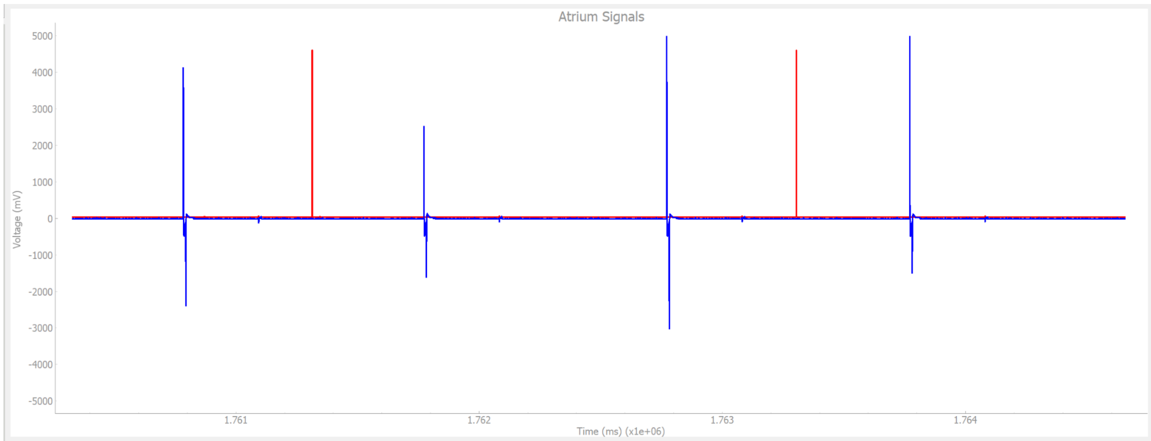
After Recovery Time:

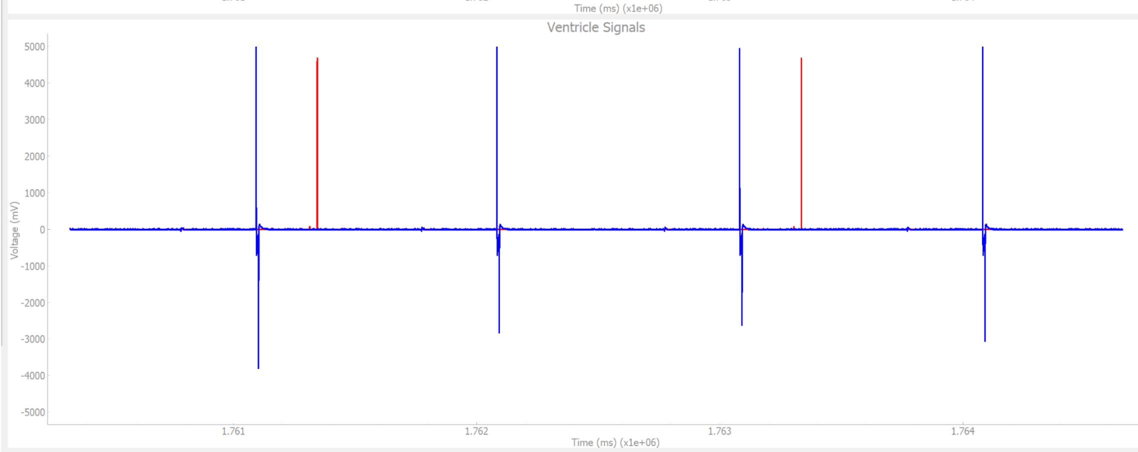


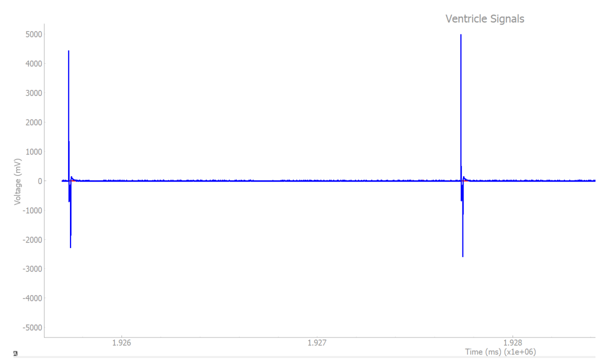


<table>
<tr>
<td>Mode: DOOR
LRL: 60
URL: 140ppm
MSR: 120 ppm
AV Delay: 300 ms
Response Factor: 16
Threshold: V-Low
Reaction TIme: 30 s
Recovery Time: 2 min
Heart: 30 ppm</td>
<td>Heart Paces at 60 ppm, increases to 120 ppm during shaking, recovered to 60 ppm after 2 mins</td>
<td></td>
<td>Pass</td>
</tr>
<tr>
<td colspan="4">With Activity:

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

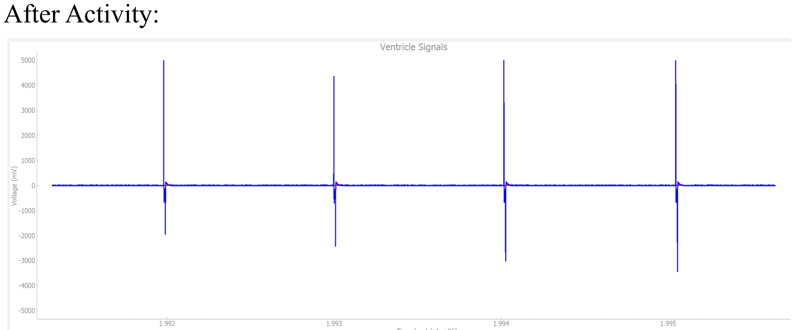
After Recovery Time:





Mode: VOOR LRL: 30 URL: 60 ppm MSR: 120 ppm Response Factor: 16 Threshold: V-Low Reaction TIme: 30 s Recovery Time: 2 min Heart: No Input	Heart Paces at 30 ppm, increases to 60 ppm during shaking and stops at URL		Pass

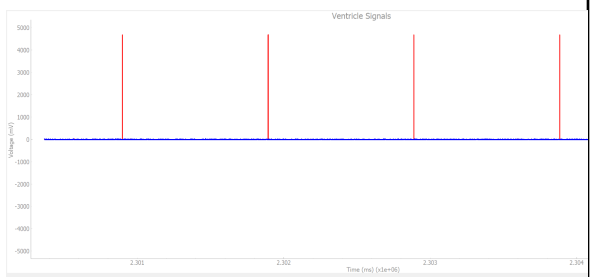
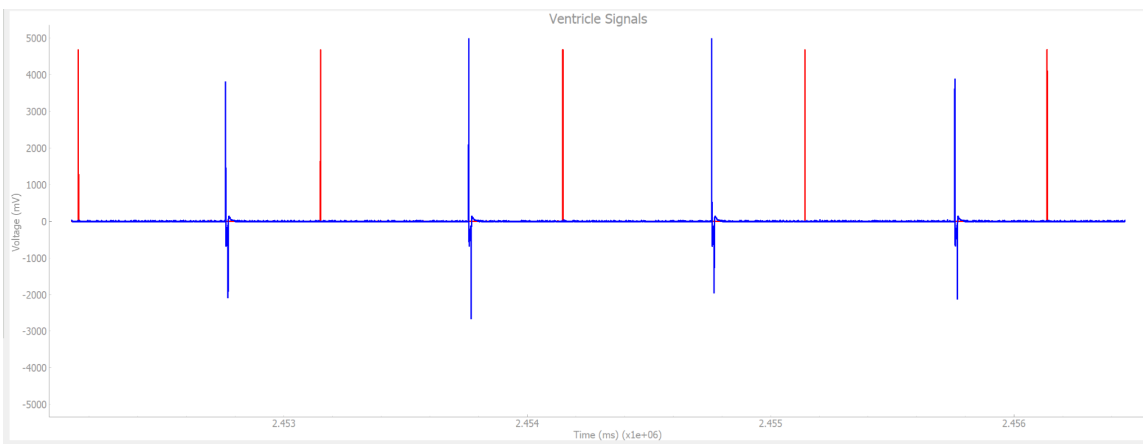
After Activity:



VVIR and AAIR Testing

There will be one test done. The pacemaker will start in a state where LRL = Natural Pace, therefore not inhibiting at all. The pacemaker will be shaken to show that the Adapted Rate increases which leads to inhibiting.

Table 3.5 VVIR and AAIR Test Results

<table>
<tr><th>Settings</th><th>Expected Output</th><th>Actual Output</th><th>Result</th></tr>
<tr><td>Mode: VVIR
LRL: 60
URL: 140 ppm
MSR: 120 ppm
Response Factor: 16
Threshold: V-Low
Reaction TIme: 30 s
Recovery Time: 2 min
Heart: 60 ppm</td><td>Heart Paces at 60 ppm with no inhibited paces, the pacemaker is shaken and inhibition begins</td><td></td><td>Pass</td></tr>
<tr><td colspan="4">After Activity:

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

Part 4: References

[1] G. Meyer, “Pacemaker Microcontroller Shield.” 02-Nov-2020.

[2] “PACEMAKER System Specification.” Boston Scientific, Boston, 03-Jan-2007.