MECHTRON 3K04: Software Development Assignment 2 – Pacemaker Design

L02

Group 17

Name:

Zichuan You – youz7 – 400358506

Congling Tang – tangc61 – 400363198

Yueyue Sun – sun9 – 400294173

Xinyi Zhou – zhoux171– 400370690

Charlie O’Brien - obriec20 - 400305005

Academic Integrity Statement

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University. [Charlie O'Brien, 400305005]

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University. [Xinyi Zhou, 400370690]

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University. [Yueyue Sun, 400294173]

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University. [Congling Tang, 400363198]

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University. [Zichuan You, 400358506]

Contents

Overview:

The primary aim of this project is to build a dual-chamber pacemaker, the user will be able to choose the mode and customize the input variable to the pacemaker. The user will use DCM (Device Control Monitor) to input their data and selected mode, DCM will then check the correctness of the inputs by preset limits in code. If the inputs are in the correct range, then the data would be passed to the board, which will generate machine language and compile to the Simulink Stateflow. The DCM should also be able to receive the data transferred from the board and sketch real-time atrium and ventricle plots. In our project, the pacing result can be visualized in Heartview.

Before we go through our Simulink System design, here is a quick summary of what we are required to accomplish:

Table 1: Design Requirements

Requirement	Description
Algorithm for pacing	Implement reliable pacing control stateflow, including heartbeat detection and pacing signal transmission with the provided parameters.
Data sensing and processing	Acquire the sensing signals from the hardware and make corresponding pacing decisions.
Monitoring and variable management	The sensing and pacing results should be monitored, and the pacing parameters can be controlled by the user.
Program operation stability	The Simulink model must be constant, to ensure the stability of pacing signal.
Hardware hiding	Abstract hardware details to simplify operations and improve security and stability. Allow developers and users focus on developing and using the pacemaker without worrying about the impact of

	hardware configuration on the pacemaker system.

In the project, the DCM needs to transmit data to Simulink, so we assign the mode and variables in the serial communication subsystem in order to receive data from DCM to Simulink. Based on the requirements, we construct four modes in our system: **AOO, VOO, AAI, VVI, AOOR, VOOR, AAIR, VVIR**.

As shown below in Figure 1, our system contains three subsystems, which are **Inputs**, **StateFlow** and **Hardware Hidden**. Inside the **Inputs** subsystem, first we determine preset the mode and pacing parameters (which are set as constants). The values were then passed into the **StateFlow** block, inside we perform pacing operations by controlling a series of variables that charge and discharge the capacitors in the pacemaker. All the variables through StateFlow would then be delivered to the **Hardware Hidden** block to control the board through configured pins.

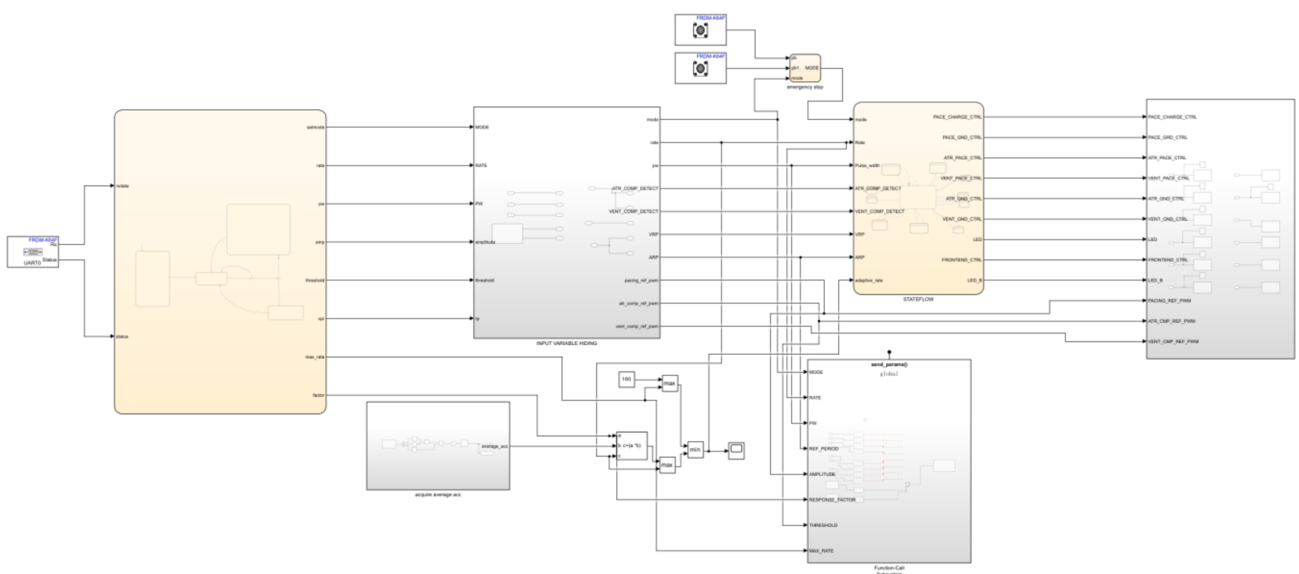

Figure 1: Simulink Model

We design the whole system such that users can only modify the inputs and observe the pacing result from Heartview. The use of three well-defined subsystems not only contributes to a clean and organized layout, but also reserves space for future improvements and integration with DCM. The internal operation is encapsulated in the **StateFlow** block, and the pins are mapped in the **Hardware hidden** block. Therefore, users can only interface the system by entering the inputs through DCM side and send them through UART by serial communication. This design effectively abstracts the hardware implementation details from the software, allowing the software to interact with the hardware without knowing the specific implementation of the hardware.

Inputs

Add Serial and connections.

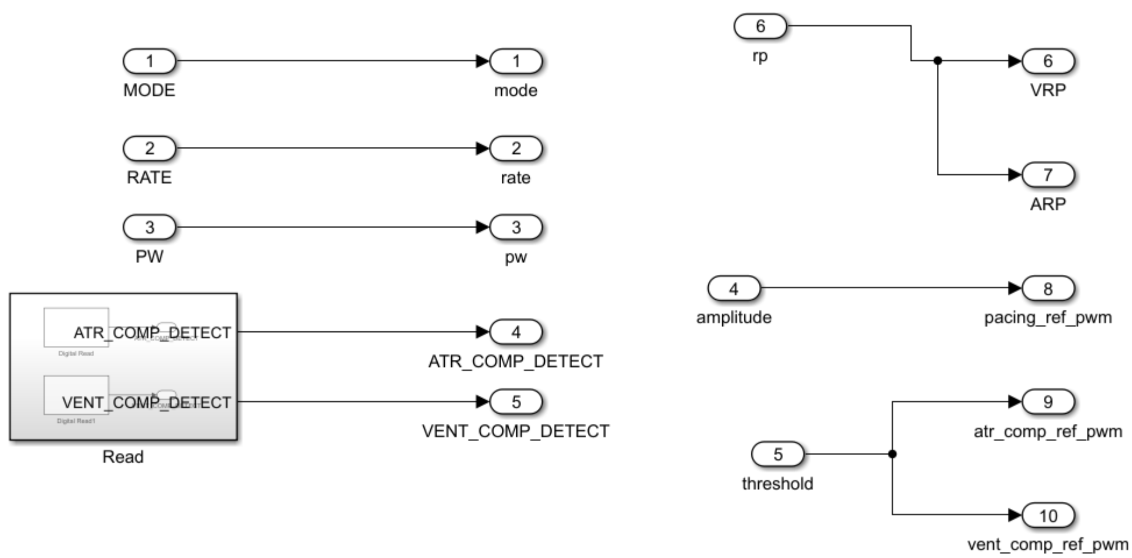

Figure 2: Inputs Subsystem

1. **Mode**

   We define a variable called **mode** to control the pacing mode, in the final project, we are getting this variable from the DCM. Due the pacemaker should receive the mode input from DCM, we switch between different pacing mode by manually input the variable **mode** in DCM, and the DCM will send the number of modes to the pacemaker as shown below:

Table 2: Mode

mode	Pacing mode
0	AOO
1	AAI
2	VOO
3	VVI
4	AOOR
5	VOOR
6	AAIR
7	VVIR

2. Rate

In our system, the **Rate** variable places an important role for all four modes. In AOO and VOO, it directly represents the pacing rate, controlling the frequency of delivering stimuli. In AAI and VVI, **Rate** is one of the main parameters to determine whether further pacing is required.

3. Pulse_width (ms)

This variable is defined as the width of the pacing pulse.

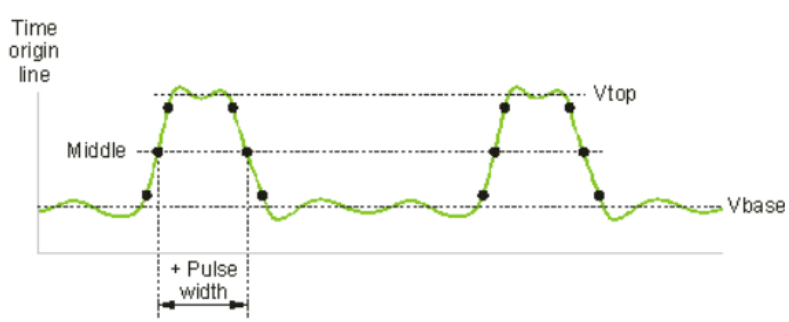

Figure 3: Pulse Width Definition

4. ATR_COMP_DETECT / VENT_COMP_DETECT

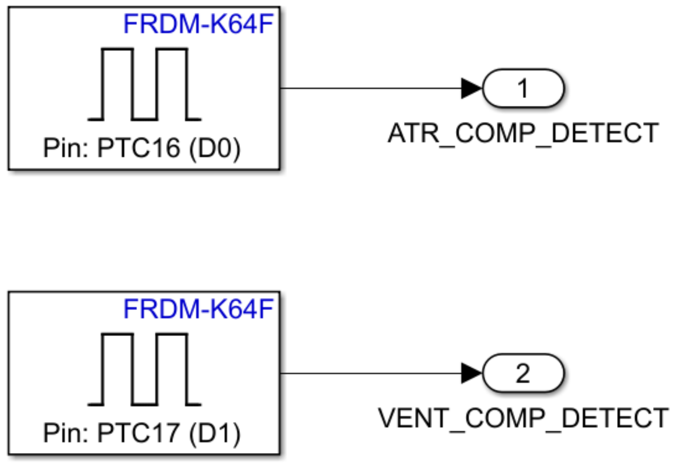

Figure 4: Get ATR/VENT_COMP_DETECT from the board as inputs.

These two variables are used to detect the pulse in atrium/ventricle in AAI and VVI mode. They are two inputs we collected from the pacemaker through the heart board.

5. ATR_REF_PERIOD / VENT_REF_PERIOD

These two variables control the refractory period in AAI and VVI mode. In refractory period, any sensed signal is ignored to avoid false sensing.

6. PACING_REF_PWM

The PWM value for charging the primary capacitor.

7. ATR_CMP_REF_PWM / VENT_CMP_REF_PWM

These variables are PWM inputs, they establish the threshold voltage of the sensing signal. If the sensed voltage is larger than the threshold voltage, the signal is considered as a pulse, which would return True to the corresponding ATR_CMP_DETECT / VENT_CMP_DETECT as explained above, vice versa.

Table 3: Programmable Parameters

Variable	Value (Limit)	Increment
Lower Rate Limit	30-50 ppm 50-90 ppm 90-175 ppm	5 ppm 1 ppm 5 ppm
Upper Rate Limit	50-175 ppm	5 ppm
Pulse_width(ms)	0.05 0.1-1.9	 0.1
ATR_COMP_DETECT / VENT_COMP_DETECT	LO/HI	NA
ATR_REF_PERIOD / VENT_REF_PERIOD	150-500 ms	10 ms
PACING_REF_PWM	0-100	1
PVARP	150-500ms	10ms
ATR_CMP_REF_PWM / ATR_CMP_REF_PWM	0-100	1
Activity threshold	V-Low, Low, Med-Low, Med, Med-High, High, V- High	NA
Response Factor	1-16	1
Reaction Time	10-50sec	10sec
Recovery Time	2-16 min	1min

StateFlow

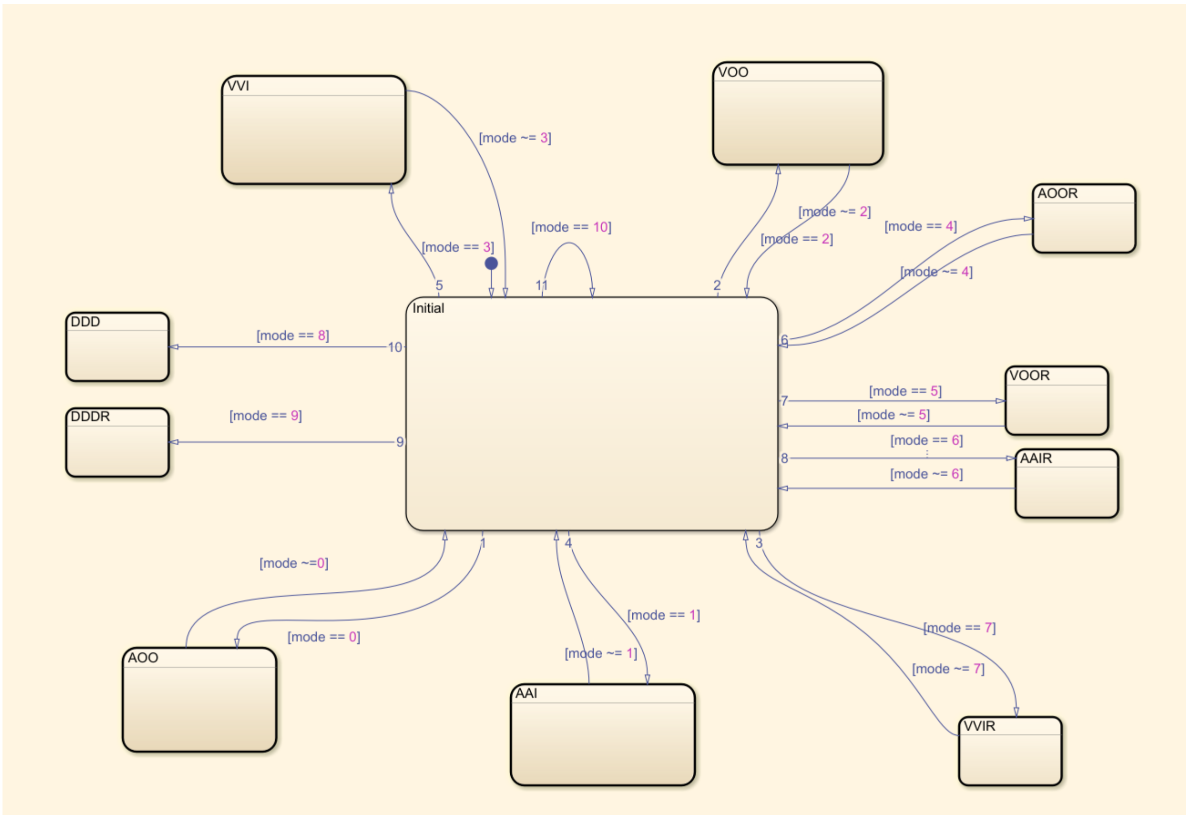

Figure 5: StateFlow

The program will stay at the Initial state until receives a different mode. In the State flow diagram, we use sub chart block to implement different modes. The controlled variables are changed in the sub chart to execute the functions in each mode.

Table 4: Bradycardia Operating Modes

	I	II	III	IV (optional)
Category	Chambers Paced	Chambers Sensed	Response To Sensing	Rate Modulation
Letters	O–None A–Atrium V–Ventricle D–Dual	O–None A–Atrium V–Ventricle D–Dual	O–None T–Triggered I–Inhibited D–Tracked	R–Rate Modulation

AOO

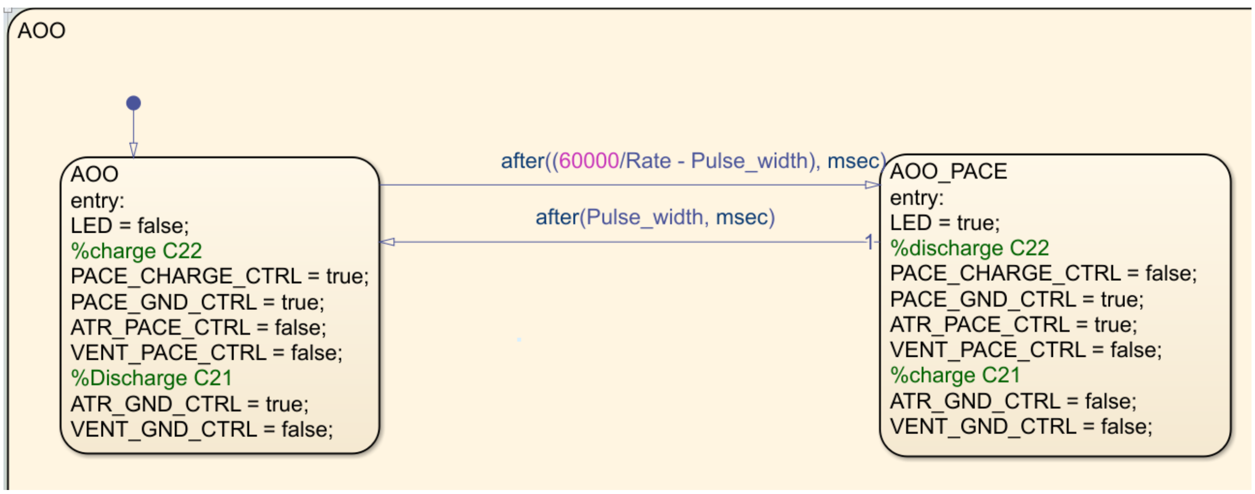

Figure 6: AOO StateFlow

Based on Table 3, AOO represents Atrium No Response to Sensing mode. In this mode, the pacemaker is pacing at a fixed rate and there is no sensing required.

First entering mode AOO, the capacitor C22 is charged and C21 is discharged to prepare for the pacing. After waiting for a period that equal to the time difference between the previous pulse's falling edge to the next rising edge, the system would enter the **AOO_PACE** state, which discharge C22 and charge C21, gives the pace to the heart, the pacing operation last for one pulse width then stop pacing and return to the **AOO** state and prepare for the next pacing.

Pacing circuit

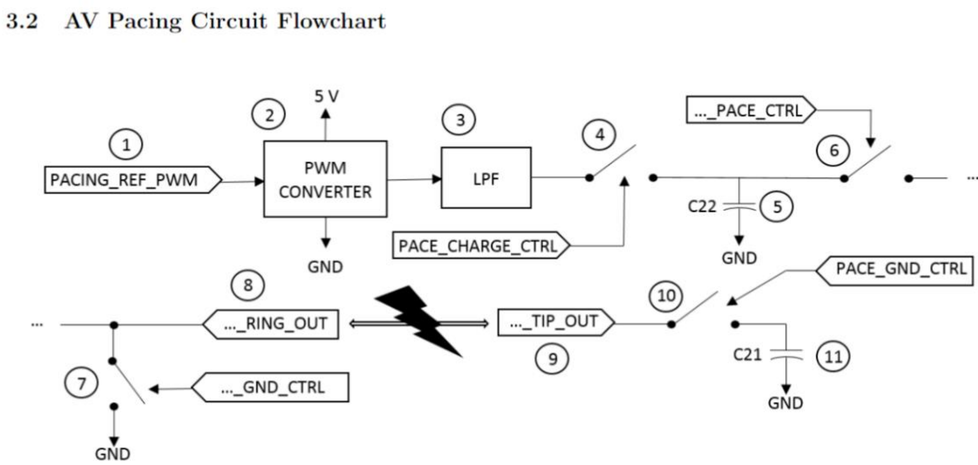

Figure 7: Pacing Flowchart Overview

Our design is based on this circuit provided in the document, which shows a complete process of pacing.

VOO

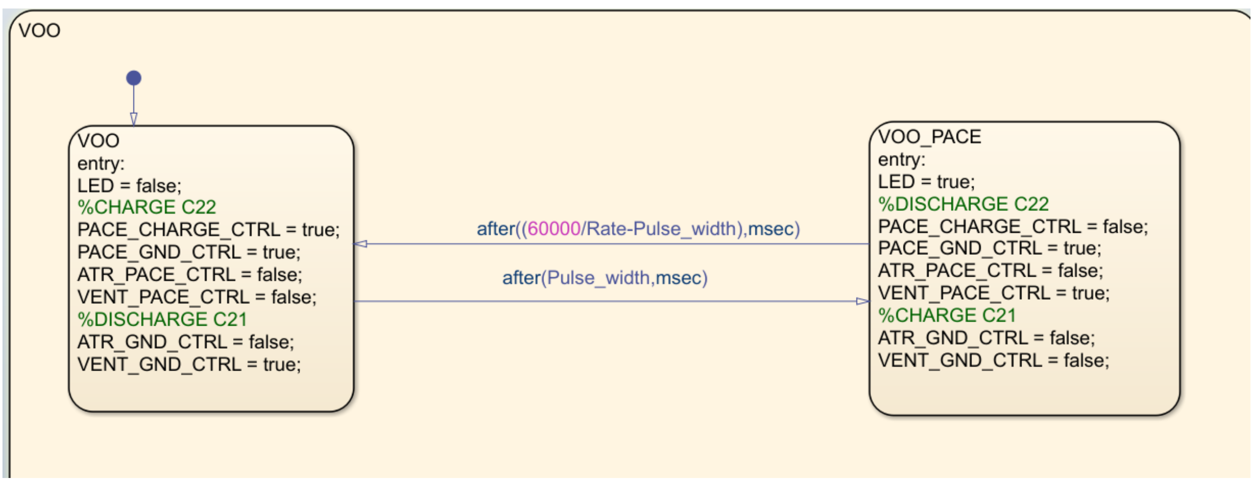

Figure 8: VOO StateFlow

In VOO mode, the functionality is the same in AOO mode, but the pacing signal is applied on the ventricle.

AAI

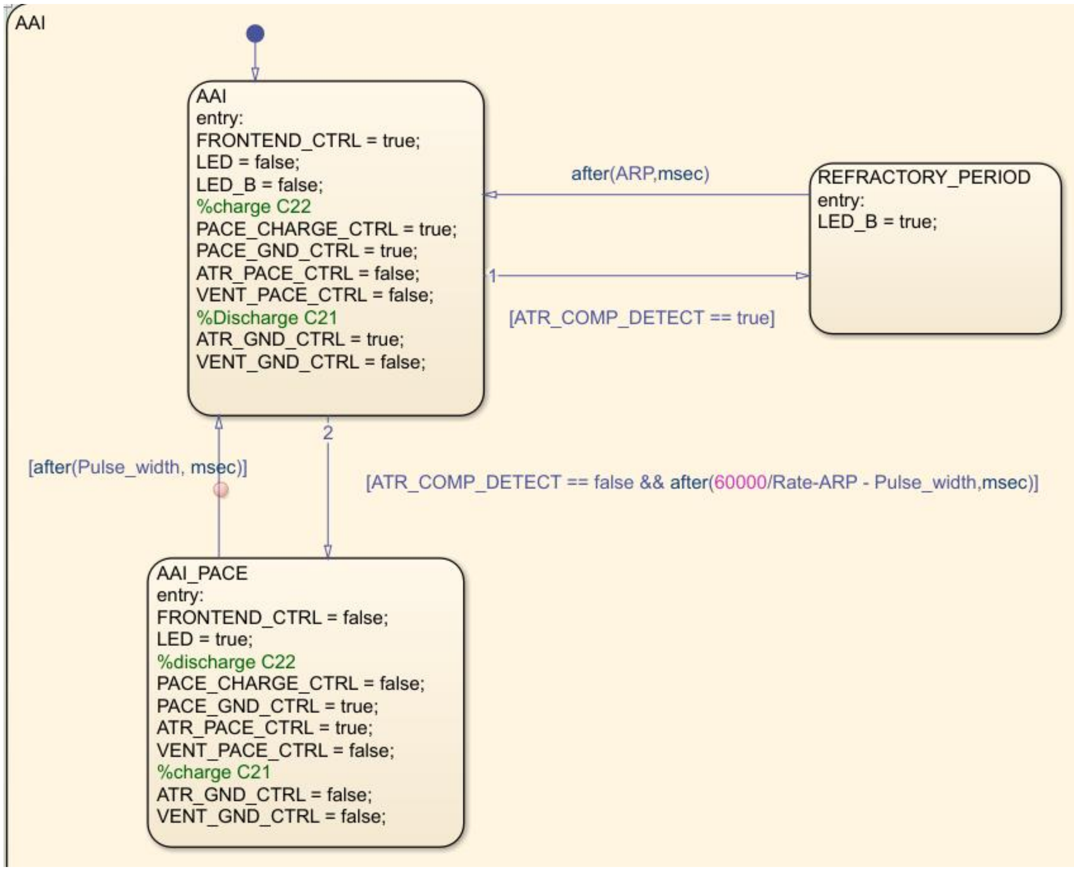

Figure 9: AAI StateFlow

AAI mode is Atrium Inhibited Response to Sensing according to Table 4 above. In this mode, when a signal is sensed in the atrium in the preset range, the pacemaker will stop pacing, and if the signal is not sensed in atrium, the pacemaker will deliver pace to the atrium depending on how slow the actual rate is.

When the pacemaker is operating in AAI mode, it will first enter AAI state. In this state, the pacemaker charges C22, and discharge C21 to prepare for the pace, also, the pacemaker constantly sensing from the atrium, if the signal is detected, it will enter **REFRACTORY_PERIOD** state and after an atrium refractory period, we return to the initial state. The refractory period would prevent the possibility of tetany. This process will be repeated until the heartbeat is not detected when the next heartbeat should be happening (meets two conditions at the same time), then it will enter the **AAI_PACE** state. In this state, we charge C21 and discharge C22, to pace the atrium. After pacing to the atrium, it returns to the initial state and starts the cycle again.

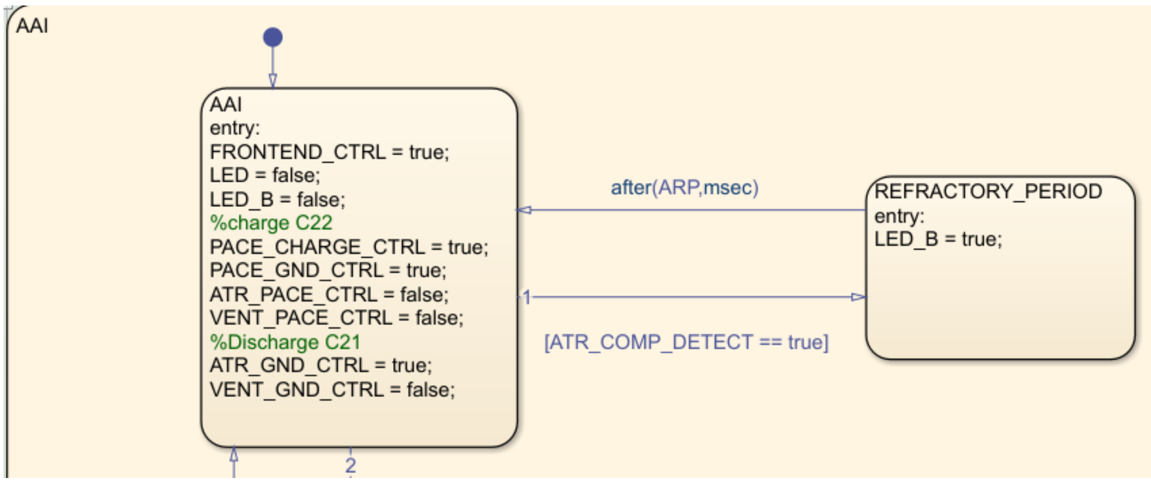

Figure 10: AAI StateFlow for sensing

Sensing the state of AAI mode, we set FRONTEND_CTRL to be true to enable sense the atrium event. The signal sensed is compared to the threshold voltage define by the CMP_REF_PWM in the LPF (Low-Pass Filter). If the sensed voltage is larger than threshold voltage, the CMP_DETECT will output High. Then the program enters the **REFRACTORY_PERIOD** state.

Sensing circuit

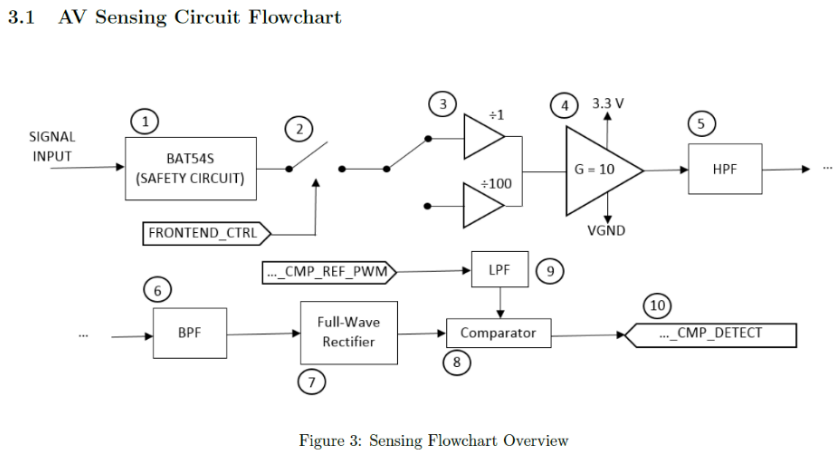

Figure 11: Sensing circuit

We used the circuit above to help design our sensing system in both AAI and VVI modes. The only switch in this circuit is FRONTENT_CTRL, which is an important variable that controls sensing on or off.

VVI

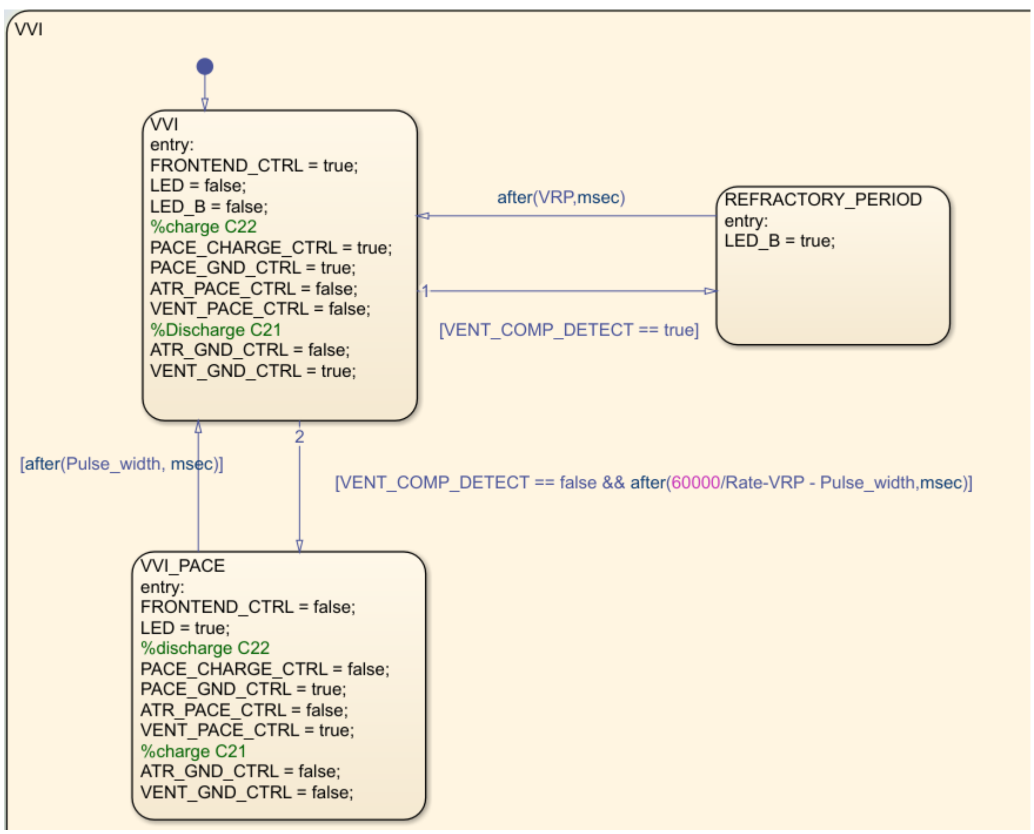

Figure 12: VVI StateFlow

In VVI mode, the functionality is the same in AAI mode, but the pacemaker is sensing the signal from ventricle and the pacing signal is output to the ventricle.

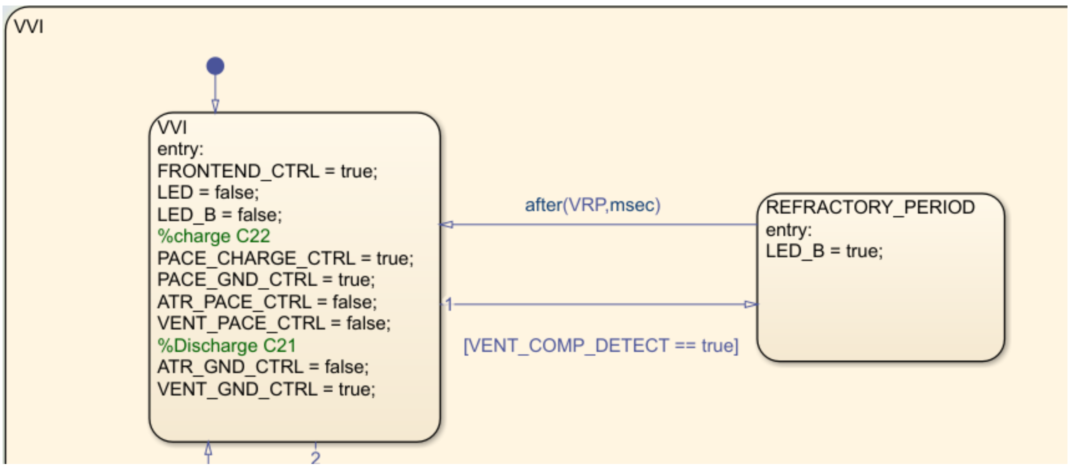

Figure 13: VVI StateFlow for sensing

The above figure is the sensing StateFlow of the VVI mode. In state VVI, we enable variable FRONTEND_CTRL to activate the sensing circuitry. When the pulse is detected, the program goes into the **REFRACTORY_PERIOD** state, and after a predefined ventricle refractory period, it will go back to the VVI state and start the new sensing cycle.

AOOR

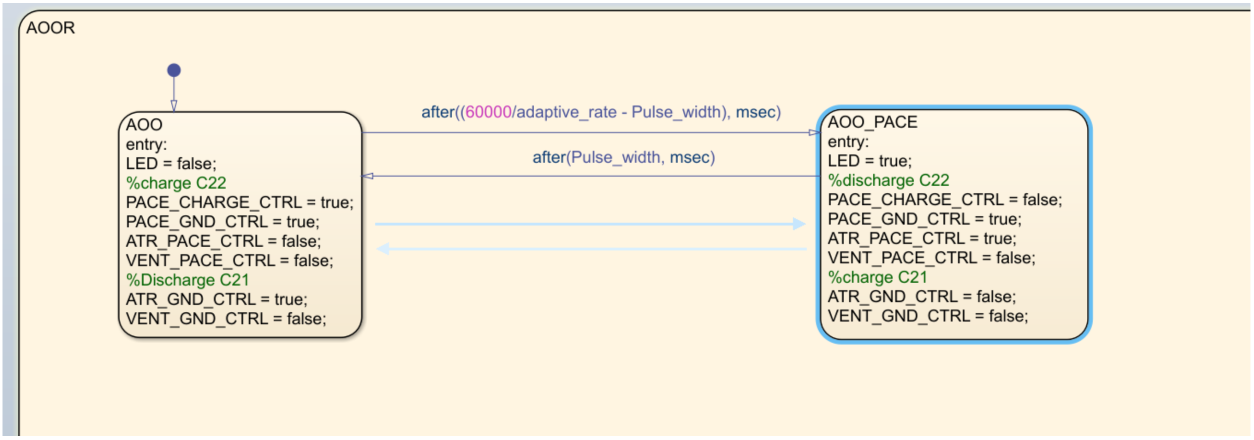

Figure 14: AOOR StateFlow

The mode AOOR is similar to the mode AOO, except AOOR is using adaptive rate as the pacing rate. The adaptive rate is acquired by the 'acquire average acc' subsystem and calculated by the formula from the "Rate_Adaptive_Pacing" file:

$$responce\ factor * average\ acceleration + rate$$

VOOR

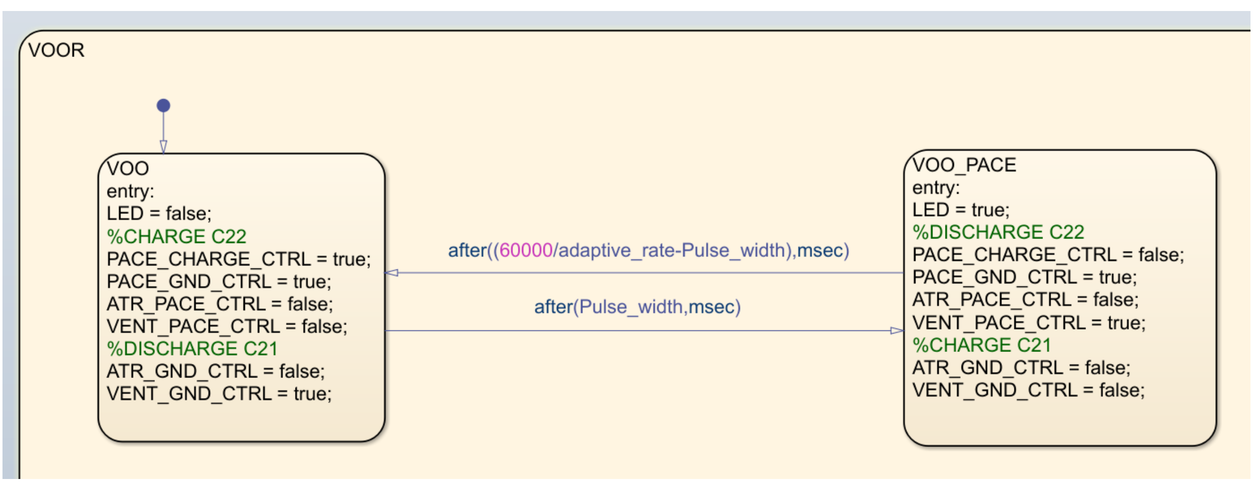

Figure 15: VOOR StateFlow

VOOR is like VOO but using the adaptive rate and pace to the ventricle.

AAIR

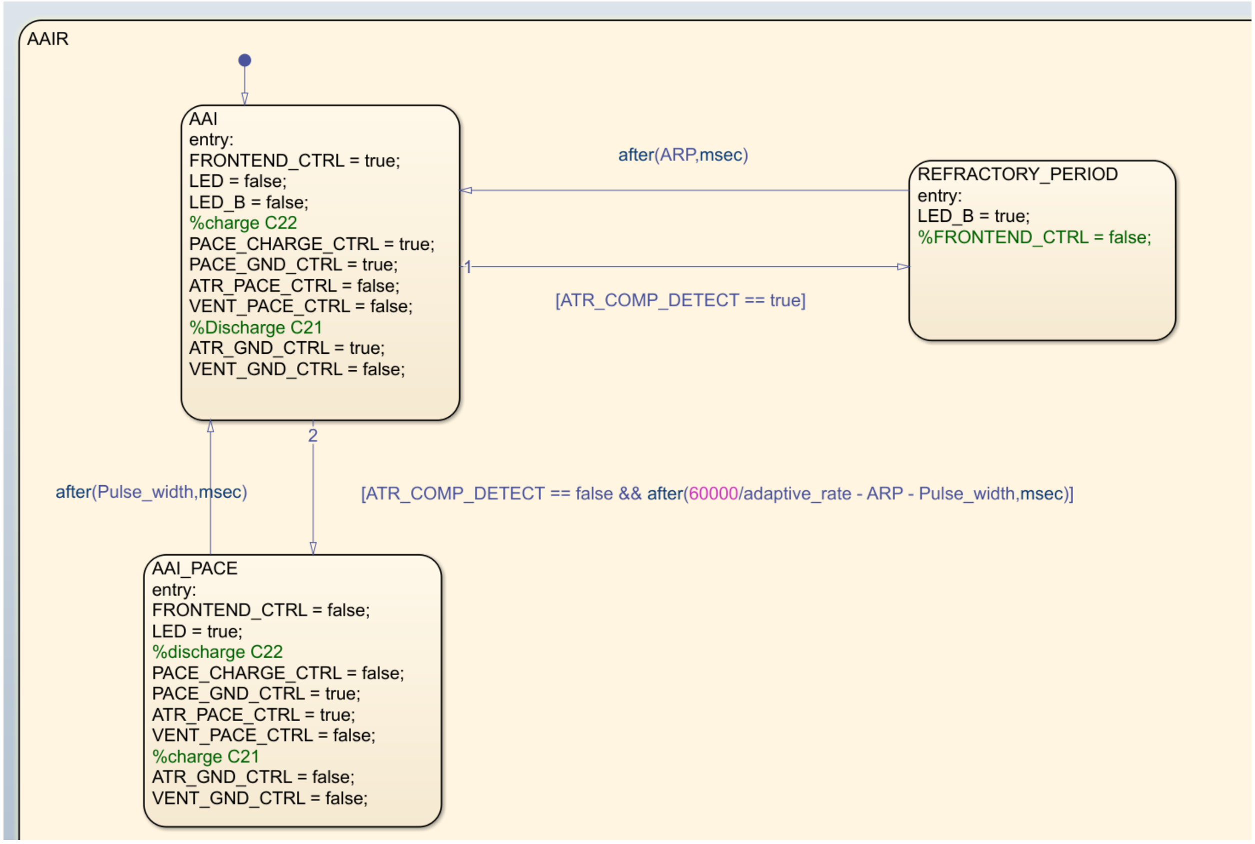

Figure 16: AAIR StateFlow

AAIR use active rate to replace the rate in AAI, the pacemaker first detects the atrium signal at the specific rate, if the heartbeat is detected, the pacemaker will not pace to the heart is the person is in low activity and the active rate is below the current heart rate. If the acceleration increased, and the active rate exceeded the current heart rate, the pacemaker will start pacing at the active rate. If the heartbeat is not detected the pacemaker will immediately start to pace to the atrium at active rate.

VVIR

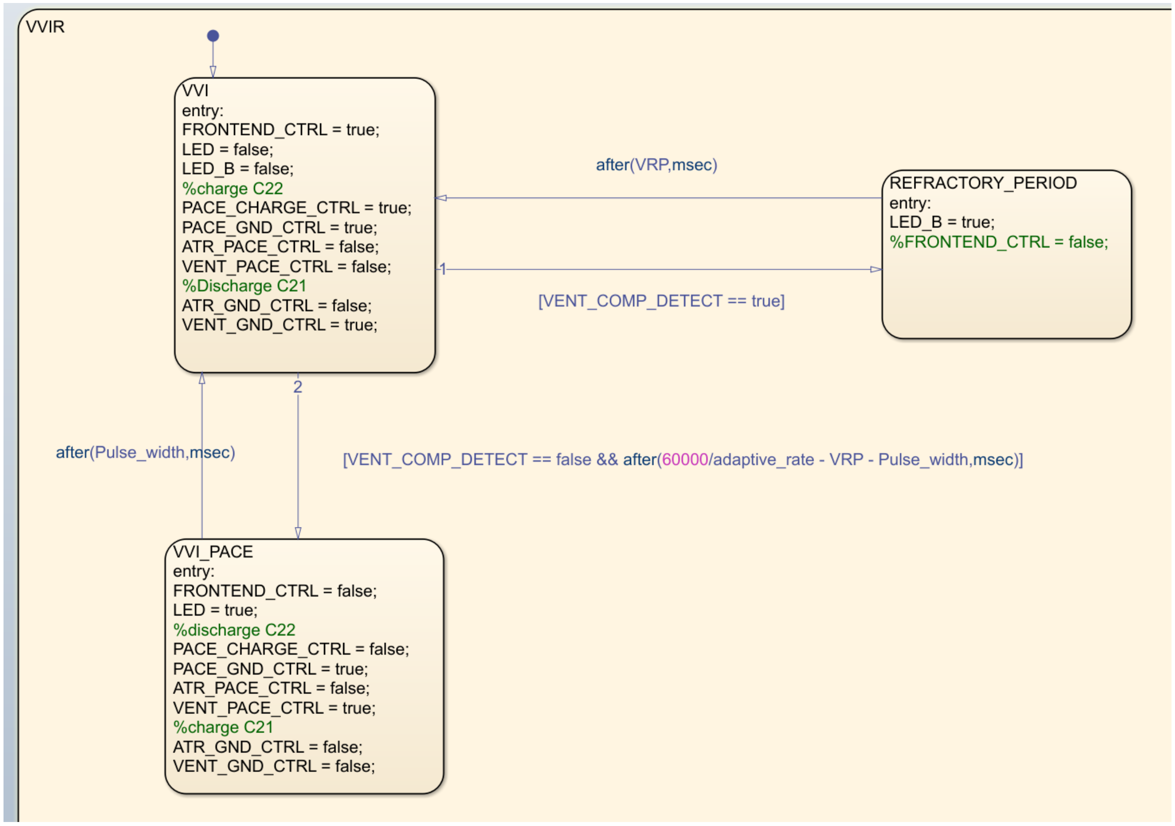

Figure 17: VVIR StateFlow

Mode VVIR has the same function as AAIR, instead of pacing to the atrium, mode VVIR will send the pace to ventricle.

Adaptive rate

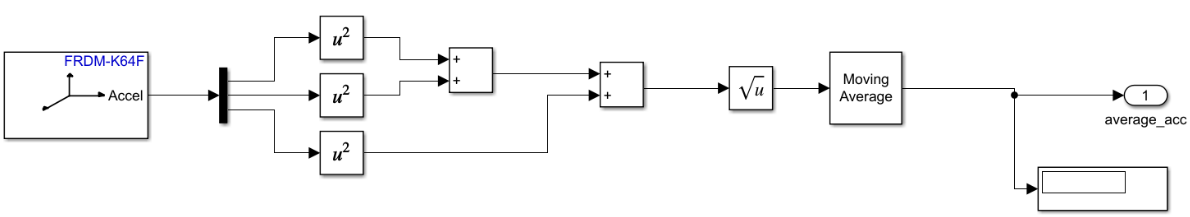

Figure 18: Acceleration Calculate Block

The adaptive rate subsystem is used to generate adaptive. It takes the reading of on-board accelerometer and applies the Euclidean norm to compute the magnitude of the acceleration. By using the '**moving average'** function, we can get the average acceleration of the pacemaker. And using the multiply-add function to compute the adaptive rate. The result of adaptive rate then compares with the lower rate limit and upper rate limit as shown below in Figure X. Then we use the adaptive rate in all the rate adaptive modes.

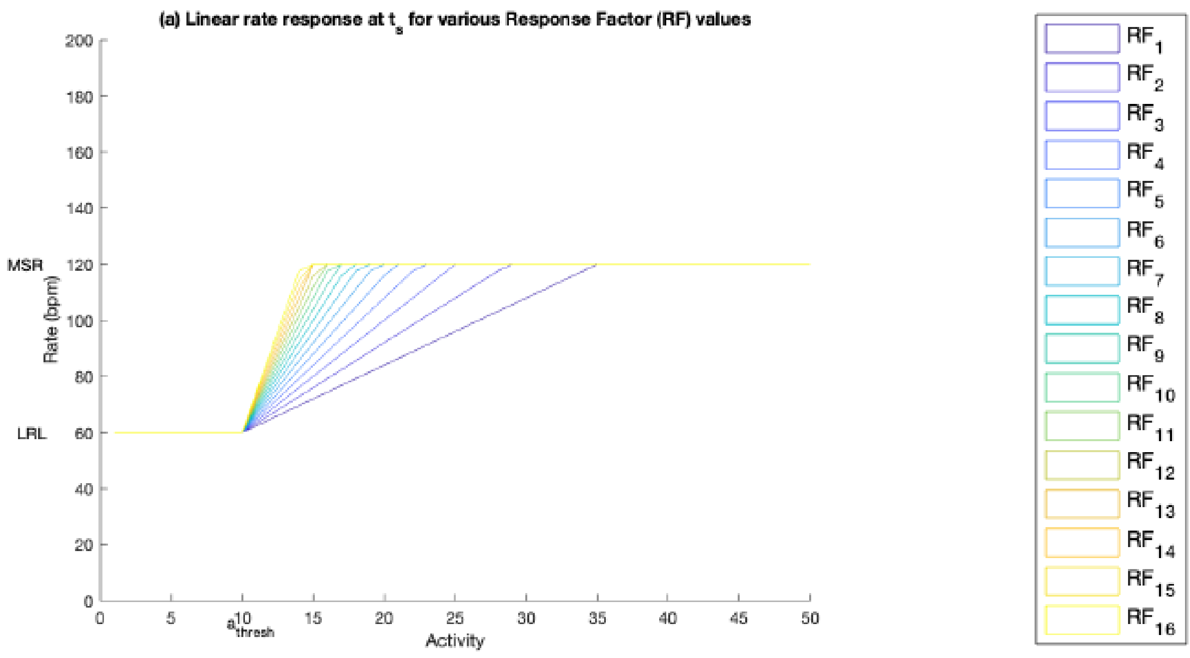

Figure 19: Rate to Activity level graph with various Response Factor

System Safety

1.Emergency stop button

In the pacemaker system, we have multiple designs to enhance the system security to prevent potential failure. The first design is the emergency terminate function using the push button on board.

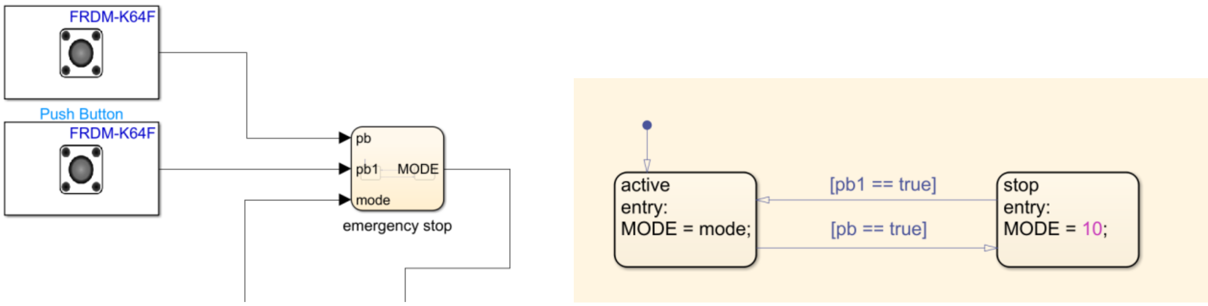

Figure 20: Emergency stop button.

This function uses two on board push buttons, the user can use the push button ‘pb’ to make the pacemaker enter mode10, which the pacemaker will shut off immediately. Also, when the user wants to resume the pacing, the user can press the other push button ‘pb1’ to change the parameter ‘mode’ back to the mode before the pace stops. The pacemaker will continue pacing using the current parameters.

2. pacing rate limit

Although the DCM will limit the input pacing rate, we hope to design the system to be more secure. Therefore, again we compare the input pacing rate parameter to the maximum rate of 180ppm using the max function in Simulink to make sure the pacing rate will not cause system errors and thus dangerous situations.

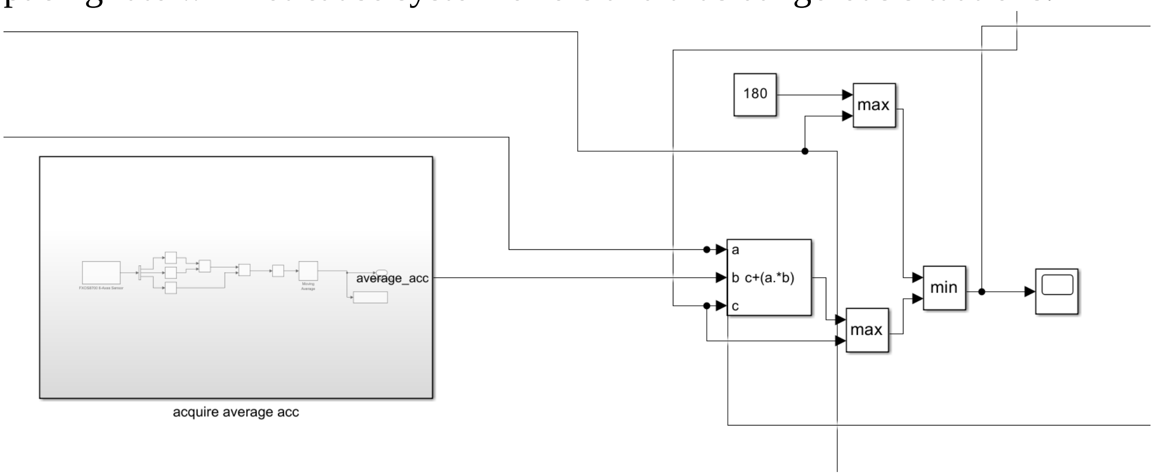

Figure 21: Threshold for adaptive Rate

With these two functions, our system would be safe in case things unexpected happened.

Assurance Case

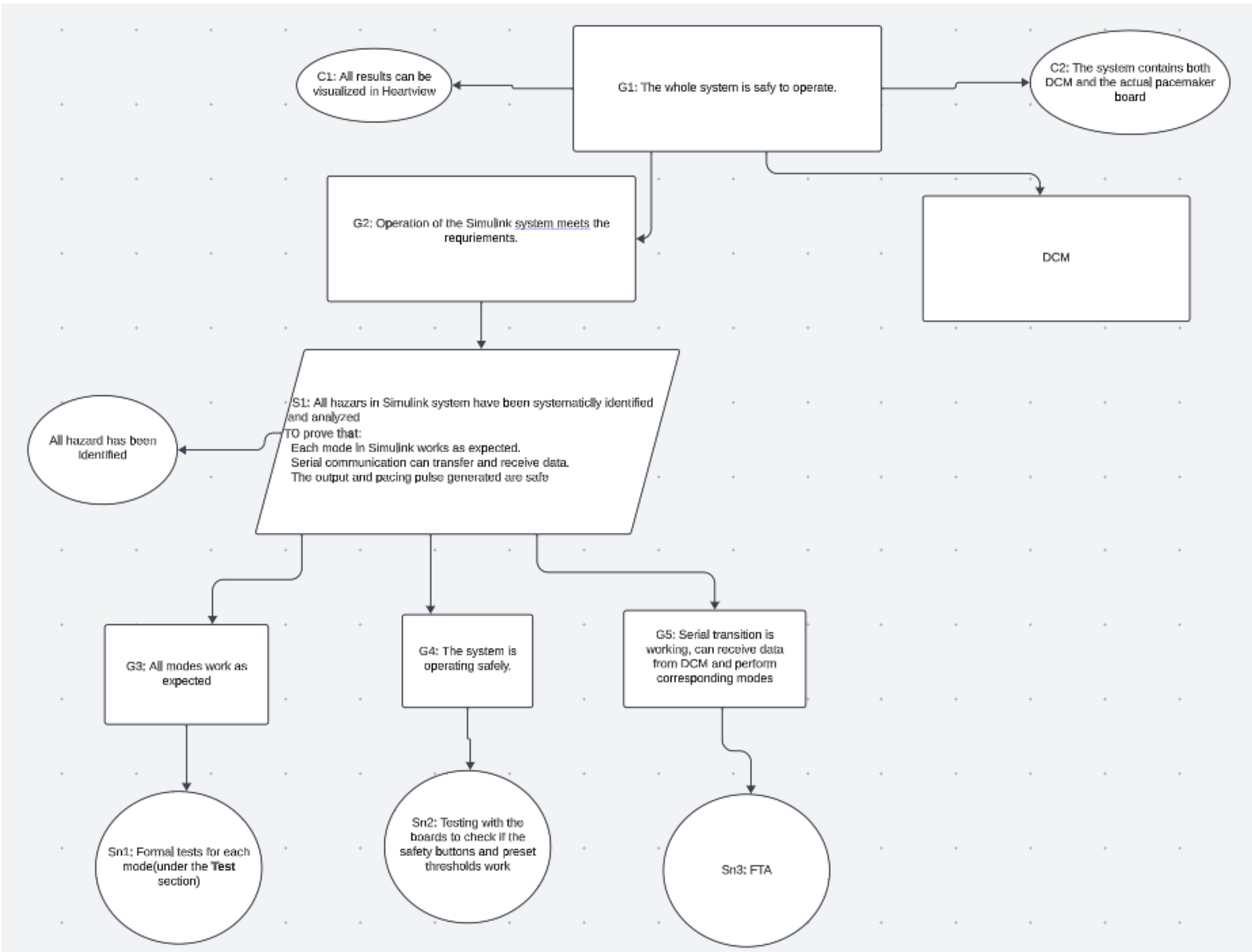

Figure 22: Assurance Case analysis

FTA (Fault Tree Analysis)

The FTA shown below is used to analyze the potential hazard during Serial Communication.

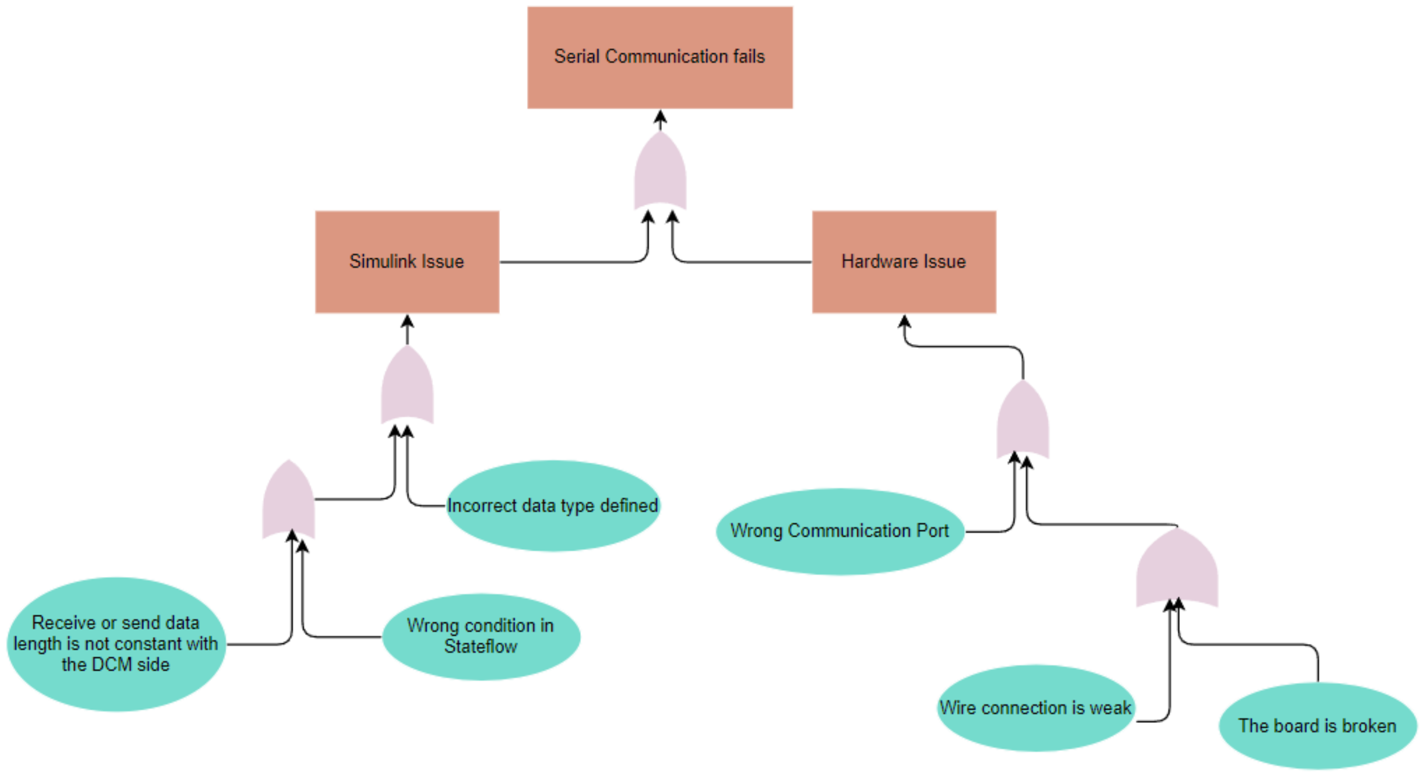

Figure 23: FTA analysis for serial communication

Serial Communication

Serial Input

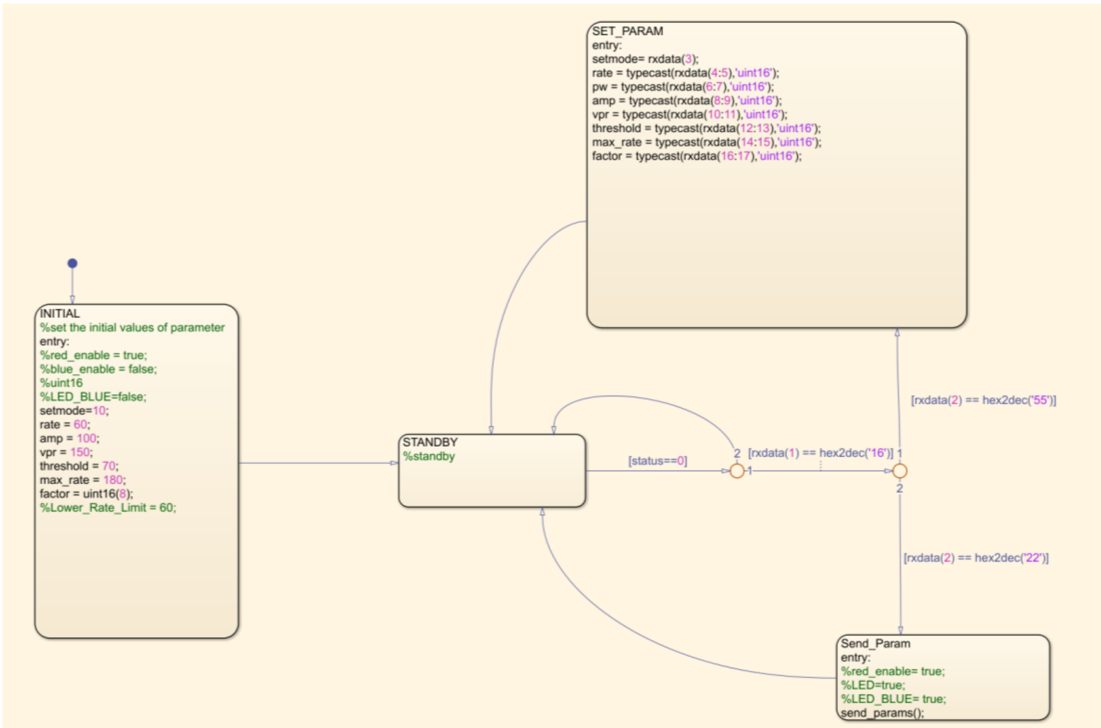

Figure 24: Serial Subsystem

The implementation of seral communication with DCM of the pacemaker is based on UART protocol. In the serial input state flow, the initial state initializes all the parameters and transfers to the standby state automatically. When the input header from DCM is 16 in hexadecimal, the serial transmission begins, if the third and fourth byte of the input header is 55 in hexadecimal, the state flow enters the SET_PARAM mode. In this mode, the pacemaker receives data from the DCM, and set up the pacemaker using the parameters input by the user. If the third and fourth byte of the4 input is 22 in hexadecimal, the pacemaker enters the SEND_PARAM mode, it will send the measurement signal of the heart to the DCM and allow user to visualize the pacing result.

Input parameters

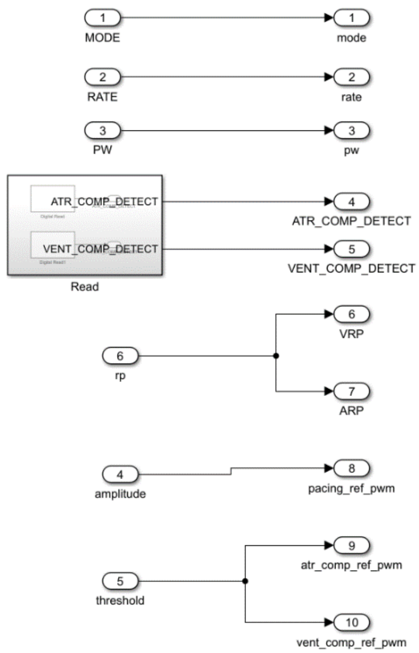

Figure 25: Input parameters

The programable parameters can be set using the DCM and be transmitted to the pacemaker by serial communication. Due to pacemaker will not pace the atrium and the vertical at the same time, the input of the magnitude and the refractory period for atrium and ventricle are integrated together to simplify the structure. The subsystem takes the input parameters from serial input, and then outputs the parameters into different modes.

Output parameters

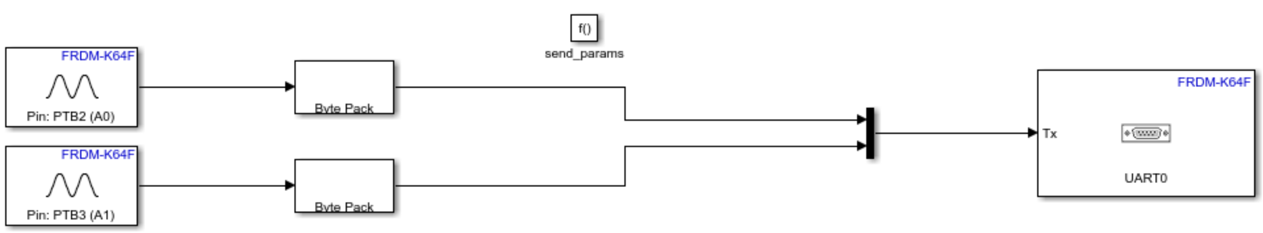

Figure 26: Serial output for generating egrams.

The figure above shows the output signal to the DCM, port A0 and A1 will transmit the atrium pacing signal and ventricular pacing signal to the DCM to be able to visualize by the user. The subsystem uses a trigger to activate. When the serial input header is ‘1622’, the pacemaker enters SEND_PARAM mode, the function “send_param” will be called and send the pacing signal to the DCM.

Hardware Design

Hardware Hidden

Hardware Hidden is a subsystem that is used for configuring the corresponding variables from the StateFlow to the NXP FRDM-K64F board based on the provided documentation. It hides hardware from the software and maps the correct pin to the model. It’s such an important subsystem that any accidental changes of the configurate we made wouldn’t cause any impact to the code. We figured out that using a subsystem also creates a clean layout.

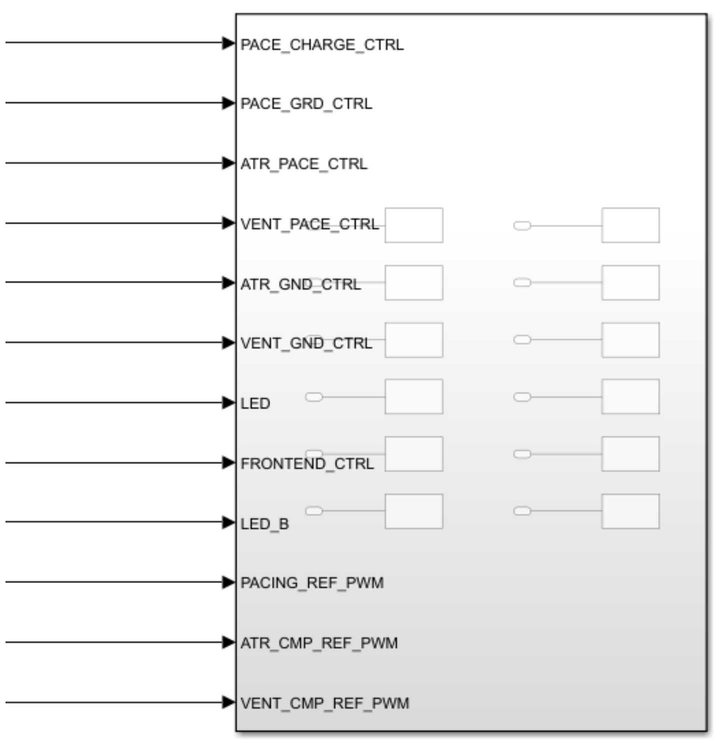

Figure 27: Hardware Hidden

Pin setting

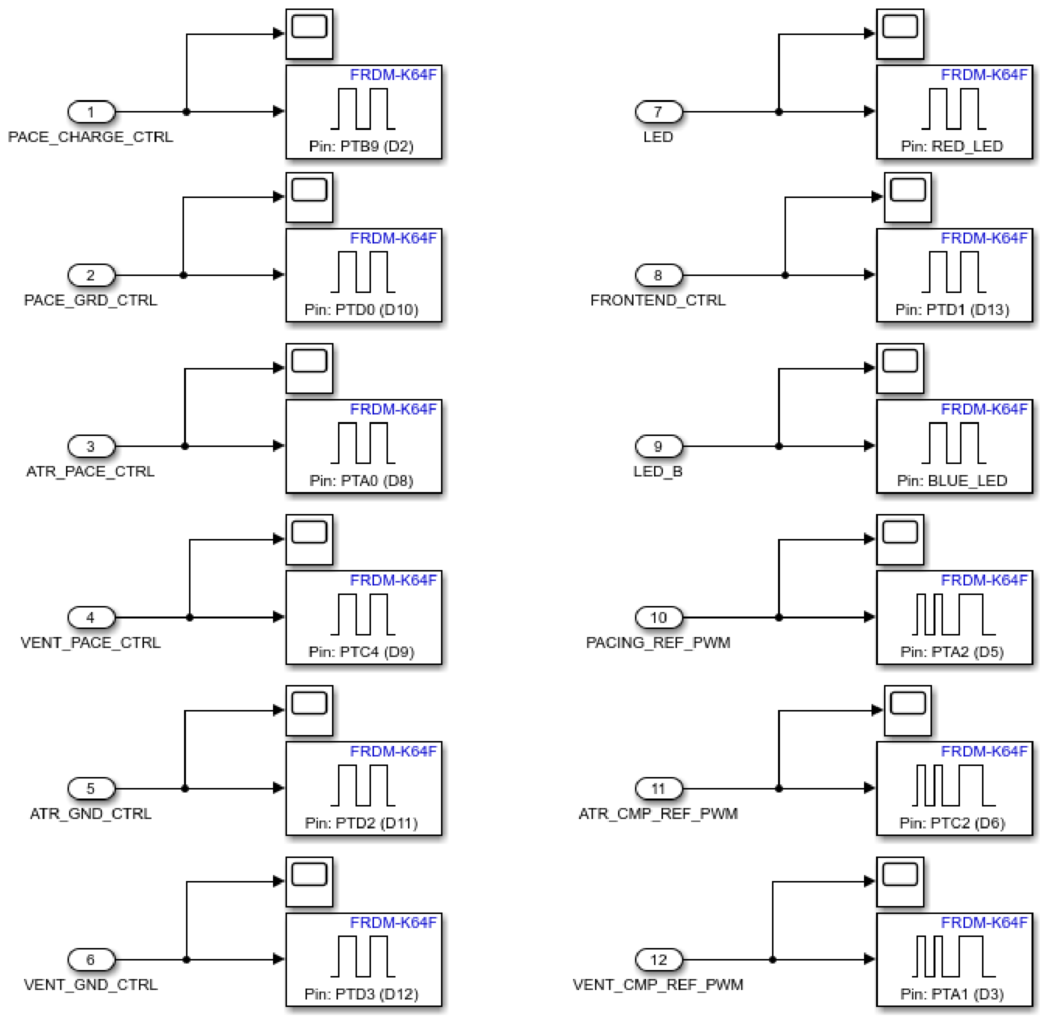

Figure 28: Pin configuration inside Hardware Hidden

We add scope to each pin to test under each mode, whether the pin value matches with the requirement.

Table 5: Pin Assignment

Name	Pin	Explanation	Application
PACE_CHARGE_CTRL	D2	Low while VCD is high, vice versa	All modes
VENT_CMP_REF_PWM	D3	Receive the threshold we set in the input subsystem and use it to detect if there is a signal from ventricle.	VVI
PACING_REF_PWM	D5	PWM value as percent value inside our **Inputs** subsystem, used to charge the primary capacitor (C22) of the pacing circuit.	All modes
ATR_CMP_REF_PWM	D6	Same function as VENT_CMP_REF_PWM but for atrium	AAI
ATR_PACE_CTRL	D8	Determine if pacing is required, discharging the primary capacitor (C22) through Atrium	All modes
VENT_PACE_CTRL	D9	Same function as ATR_PACE_CTRL	All modes
PACE_GND_CTRL	D10	Controls the switch directly following the tip, and it will prevent the current flow directly into human's heart.	All modes
ATR_GND_CTRL	D11	Discharge current to protect heart	AAI, AOO
VENT_GND_CTRL	D12	Same as ATR_GND_CTRL	VVI, VOO
FRONTEND_CTRL	D13	Enable sensing when it's high, vice versa	AAI, VVI
LED	RED_LED	Debug, indicate current state	All modes
LED_B	BLUE_LED	Debug, indicate current state	All modes

Test

AOO

1. PACE_CHARGE_CTRL

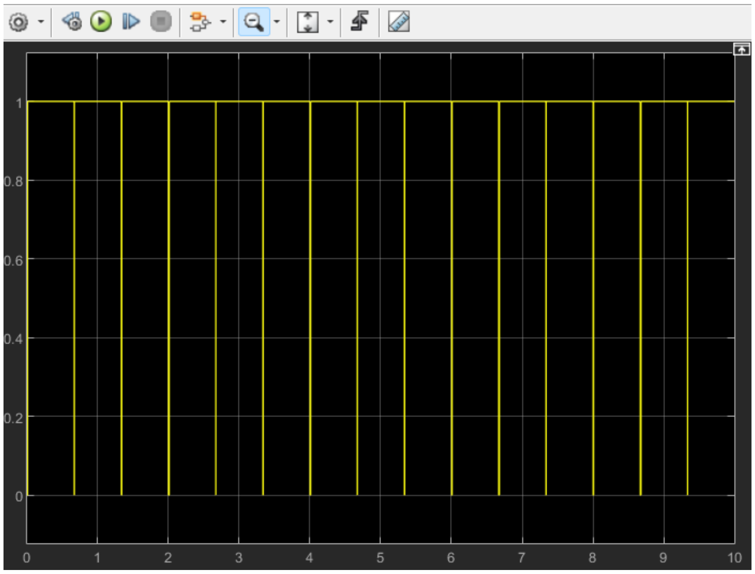

Figure 29: PACE_CHARGE_CTRL test result

This variable controls capacitor C22 (PACE_CHARGE_CTRL = true charges C22) The frequency of the pace charge control is 90Hz which is the rate we set for pacing. The pulse width is 1.5ms in setting. The figure above shows the test result of PACE_CHARGE_CTRL.

2. PACE_GND_CTRL

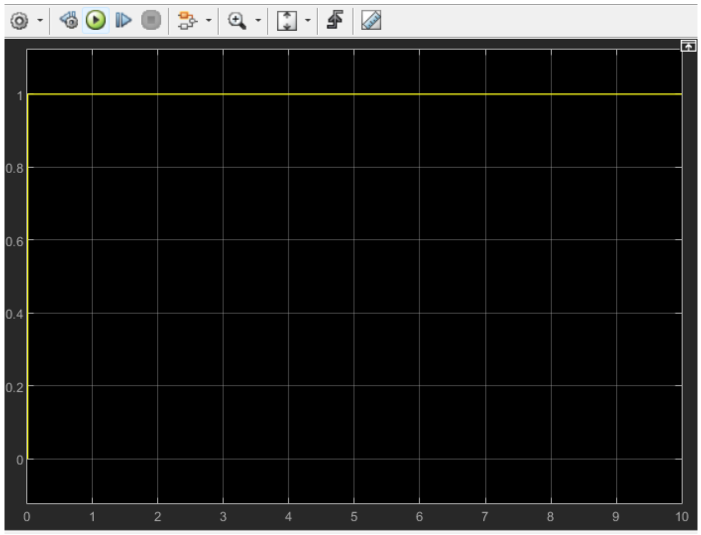

Figure 30: PACE_CHARGE_CTRL test result

The PACE_GND_CTRL test result in figure 17 show that this variable is always set to HI because it controls the switch directly following the tip, and it will prevent the current flow directly into human's heart.

3. ATR_PACE_CTRL

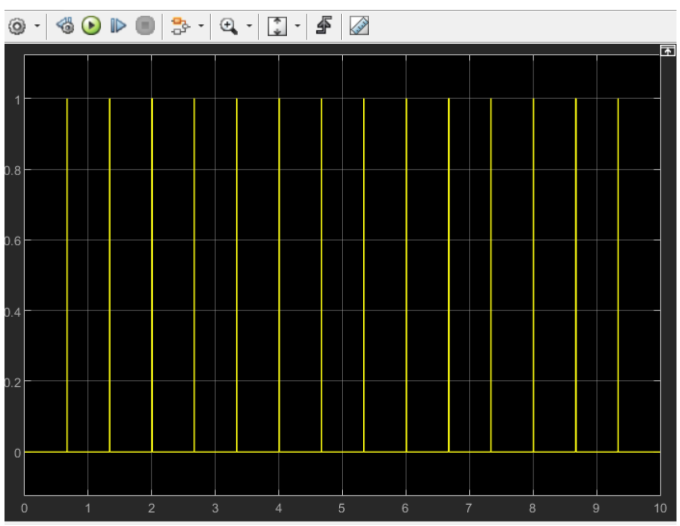

Figure 31: ATR_PACE_CTRL test result

This variable is the pacing signal we want to generate on the atrium, the signal has frequency of 90Hz, and pulse width equal to 1.5 ms.

4. ATR_GND_CTRL

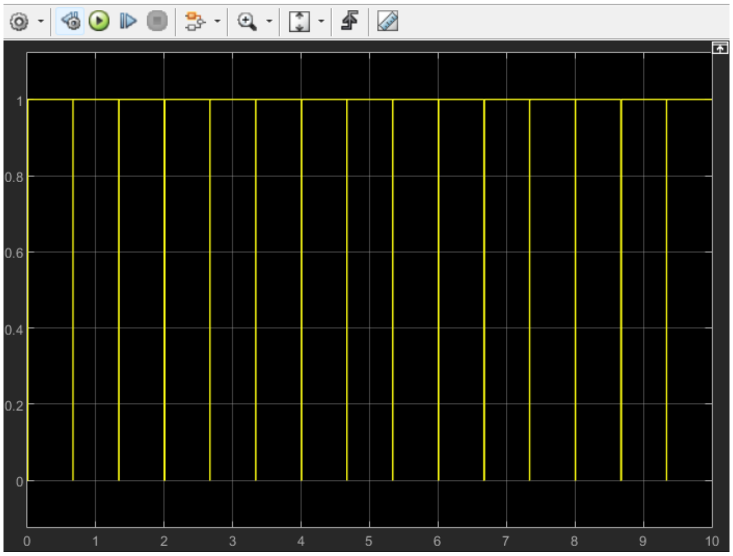

Figure 32: ATR_GND_CTRL test result

ATR_GND_CTRL is the opposite of ATR_PACE_CTRL, it is used to discharge C22.

5. VENT_PACE_CTRL / VENT_GND_CTRL

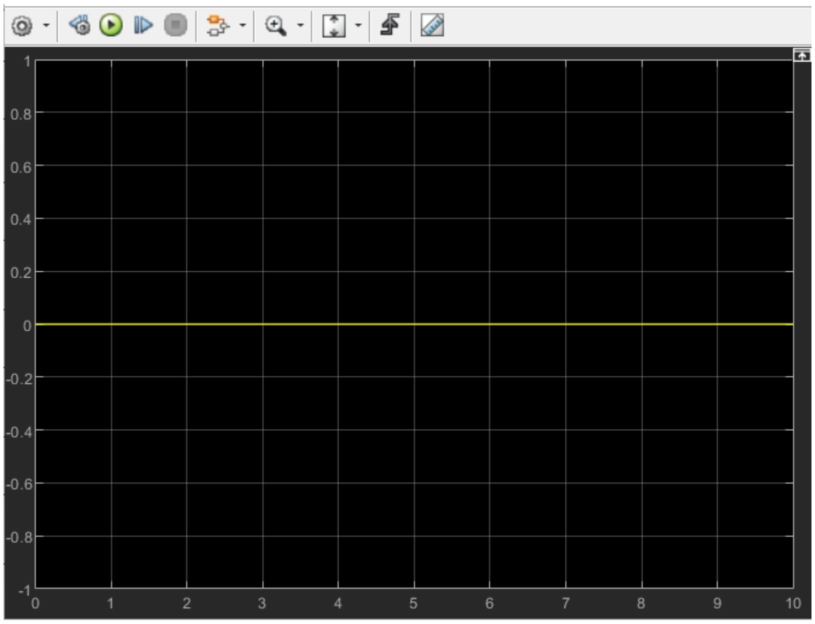

Figure 33: VENT_PACE_CTRL and VENT_GND_CTRL test result

In AOO mode, we are only pacing to the atrium, therefore the variable for pacing ventricle will be set to low.

AOO HeartView Result

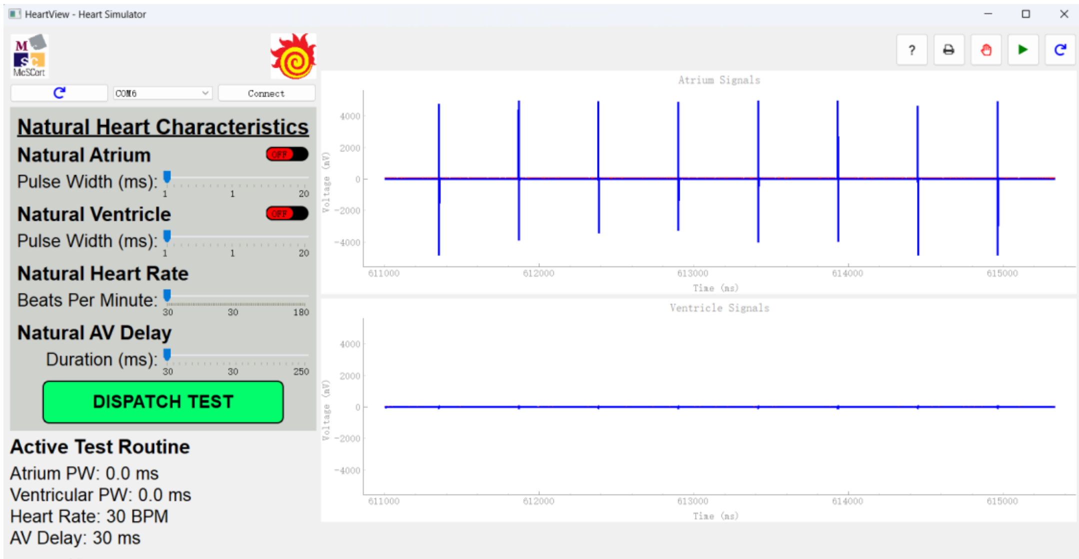

Figure 34: AOO Heartview Result

In AOO mode, the pacemaker is pacing in a fixed frequency. In figure 21, we can observe that the pacemaker is pacing at 90Hz.

AAI

1. FRONTEND_CTRL

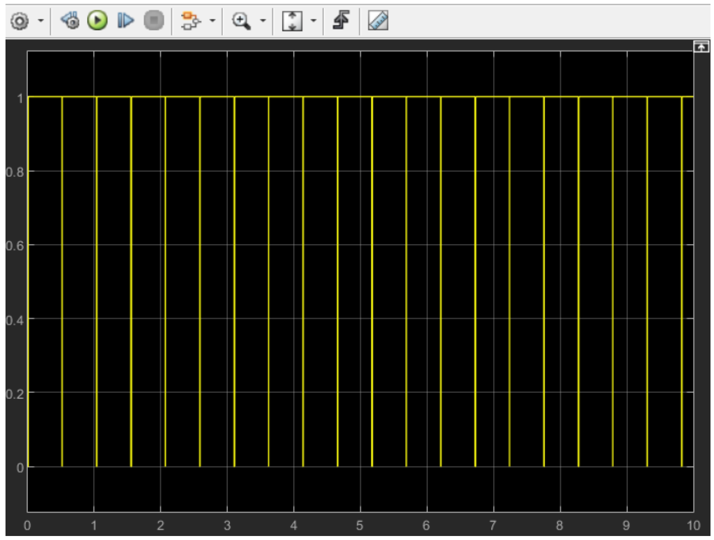

Figure 35: FRONTEND_CTRL

FRONTEND_CTRL is used to activate the sensing circuitry (FRONTEND_CTRL = true will active the sensing circuitry and vice versa). In figure 22, FRONTEND_CTRL will be turned off when no heart event is sensed, and we need to pace the heart.

2. PACE_GND_CTRL

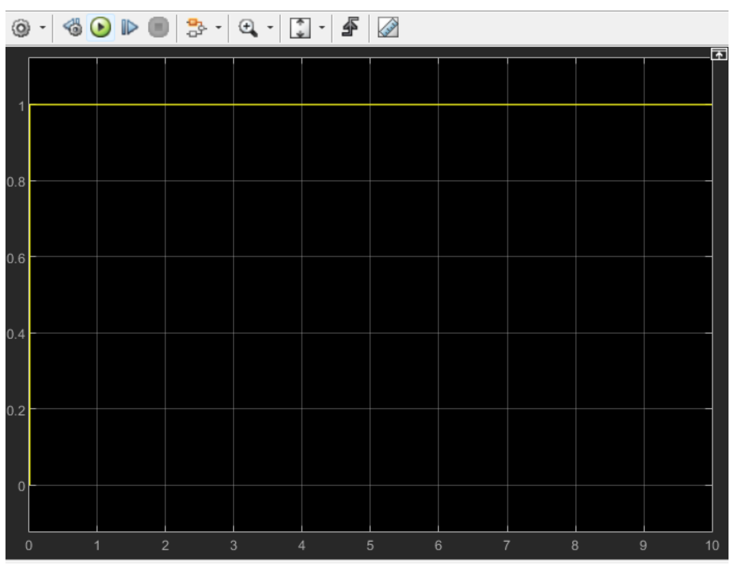

Figure 36: PACE_CHARGE_CTRL test result

The PACE_GND_CTRL test result in figure 23 show that this variable is always set to HI because it controls the switch directly following the tip, and it will prevent the current flow directly into human's heart.

3. PACE_CHARGE_CTRL

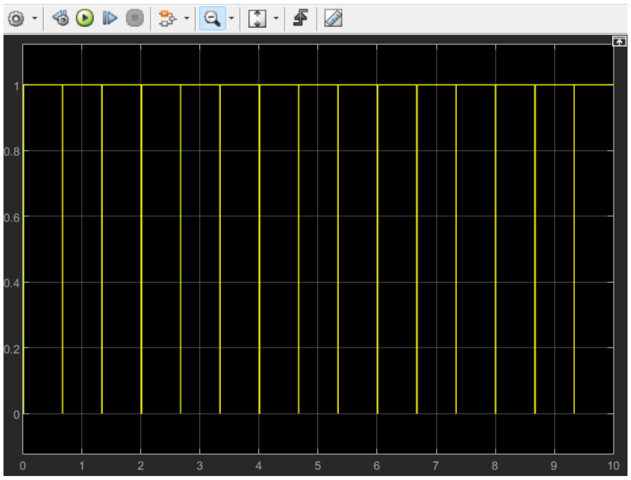

Figure 37: PACE_CHARGE_CTRL test result

This variable controls capacitor C22 (PACE_CHARGE_CTRL = true charges C22) The frequency of the pace charge control is 90Hz which is the rate we set for pacing. The pulse width is 1.5ms in setting. The figure above shows the test result of PACE_CHARGE_CTRL.

4. ATR_PACE_CTRL

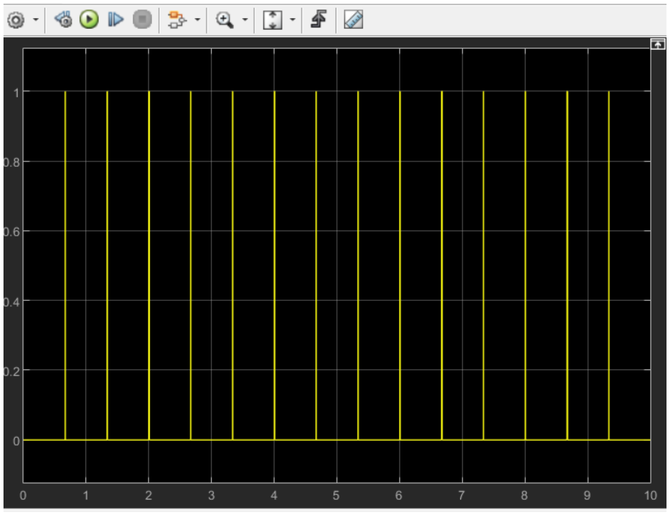

Figure 38: ATR_PACE_CTRL test result

This variable is the pacing signal we want to generate on the atrium, the signal has frequency of 90Hz, and pulse width equal to 1.5 ms.

5. ATR_GND_CTRL

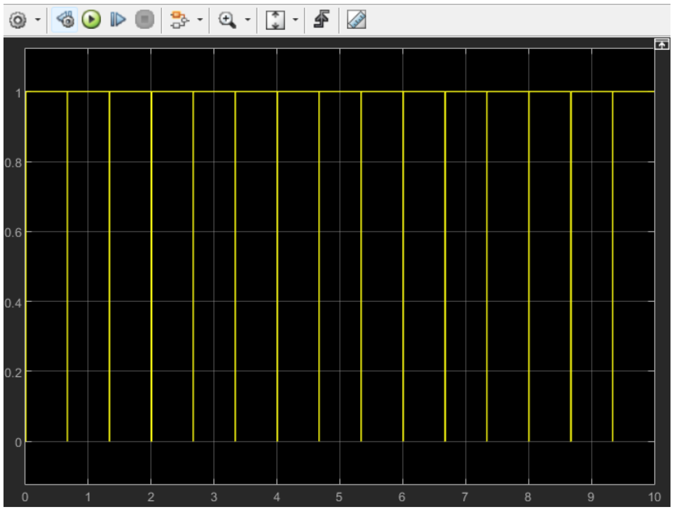

Figure 39: ATR_GND_CTRL test result

ATR_GND_CTRL is the opposite of ATR_PACE_CTRL, it is used to discharge C22.

6. VENT_PACE_CTRL / VENT_GND_CTRL

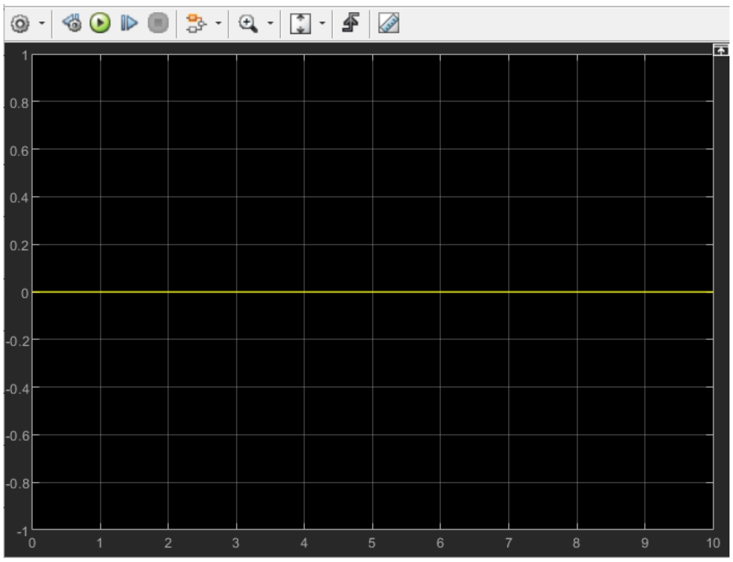

Figure 40: VENT_PACE_CTRL and VENT_GND_CTRL test result

In AAI mode, we are only pacing to the atrium, therefore the variable for pacing ventricle will be set to low.

AAI Heartview Result

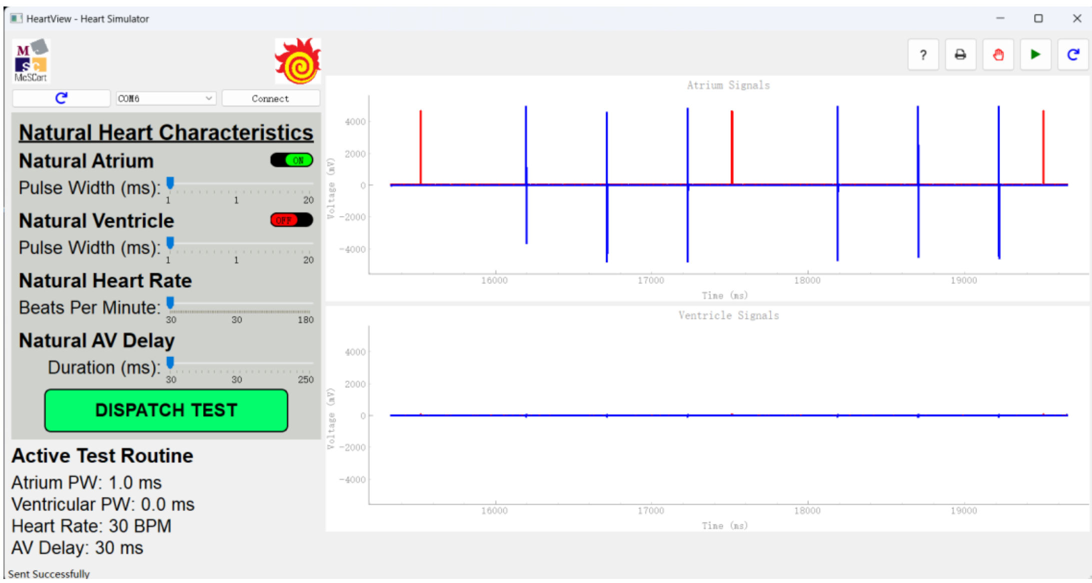

Figure 41: AAI Heartview Result when heart is beating slower than rate.

In AAI mode when the heart is beating slower than the rate we defined, the pacemaker will pace to the atrium. In figure 28 the heart is beating at 30Hz, so the pacing frequency is 90Hz. (There are 3 paces between every two heartbeat)

Figure 42: AAI Heartview Result when heart is beating faster than rate.

In AAI mode, when heartbeat is faster than rate, the pacemaker will not pace to the atrium. As the result shows in figure 29, there are no pacing signals in the atrium when it is beating at 100ppm.

VOO

1. PACE_CHARGE_CTRL

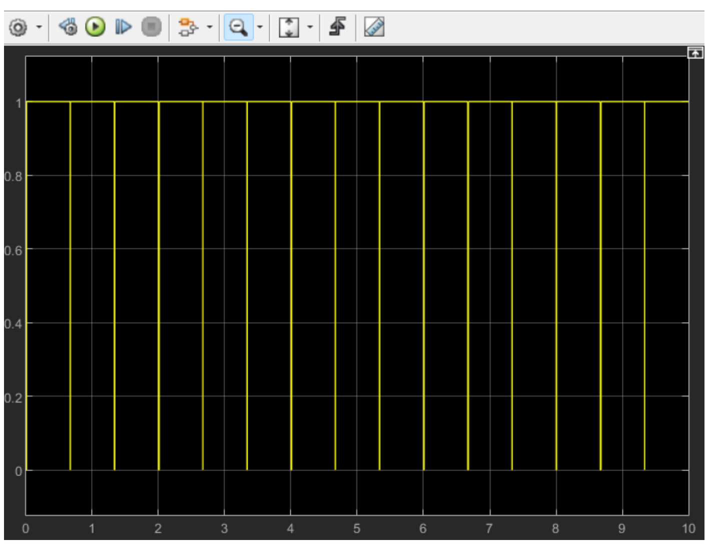

Figure 43: PACE_CHARGE_CTRL test result

This variable controls capacitor C22 (PACE_CHARGE_CTRL = true charges C22) The frequency of the pace charge control is 90Hz which is the rate we set for pacing. The pulse width is 1.5ms in setting. The figure above shows the test result of PACE_CHARGE_CTRL.

2. PACE_GND_CTRL

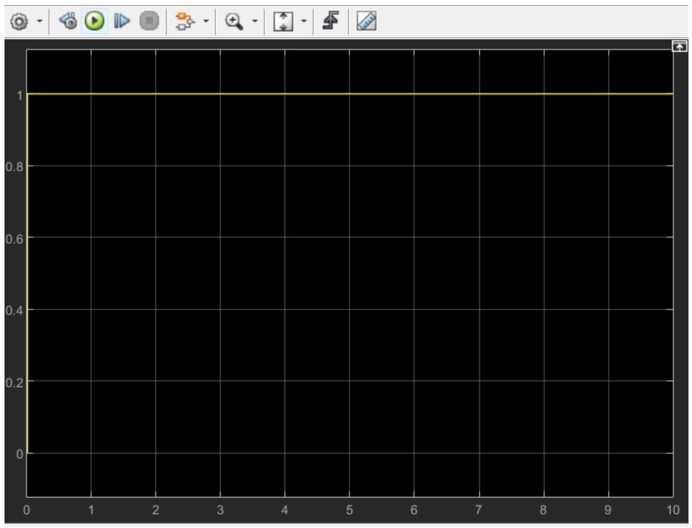

Figure 44: PACE_CHARGE_CTRL test result

The PACE_GND_CTRL test result in figure 31 show that this variable is always set to HI because it controls the switch directly following the tip, and it will prevent the current flow directly into human's heart.

3. VENT_PACE_CTRL

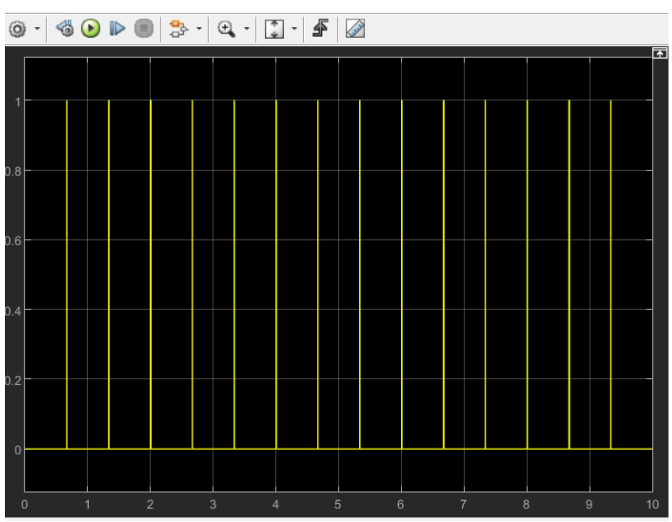

Figure 45: VENT_PACE_CTRL test result

This variable is the pacing signal we want to generate on the ventricle, the signal has frequency of 90Hz, and pulse width equal to 1.5 ms.

4. VENT_GND_CTRL

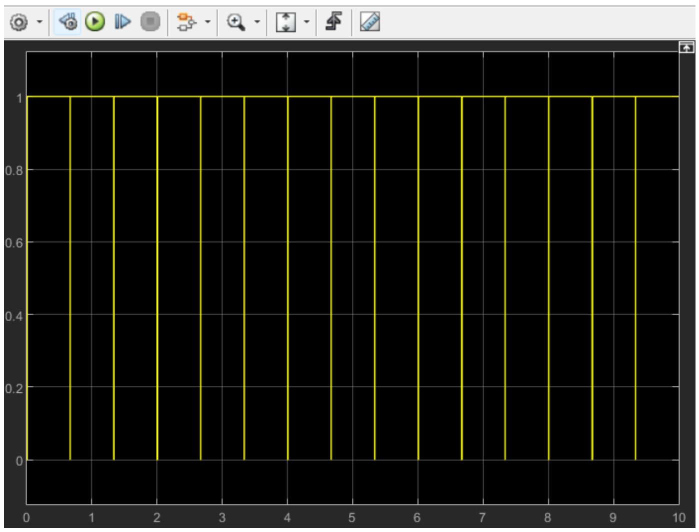

Figure 46: VENT_PACE_CTRL test result

VENT_GND_CTRL is the opposite of VENT_PACE_CTRL, it is used to discharge C22.

5. ATR_PACE_CTRL / ATR_GND_CTRL

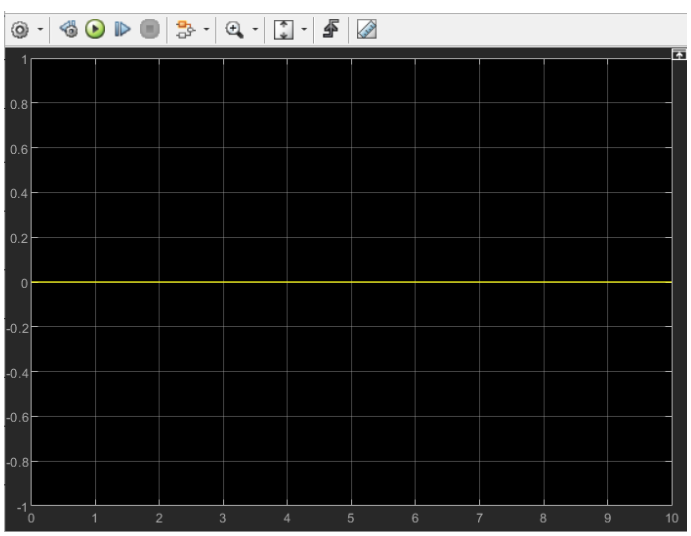

Figure 47: VENT_PACE_CTRL and VENT_GND_CTRL test result

In VOO mode, we are only pacing to the ventricle, therefore the variable for pacing atrium will be set to low.

VOO Heartview Result

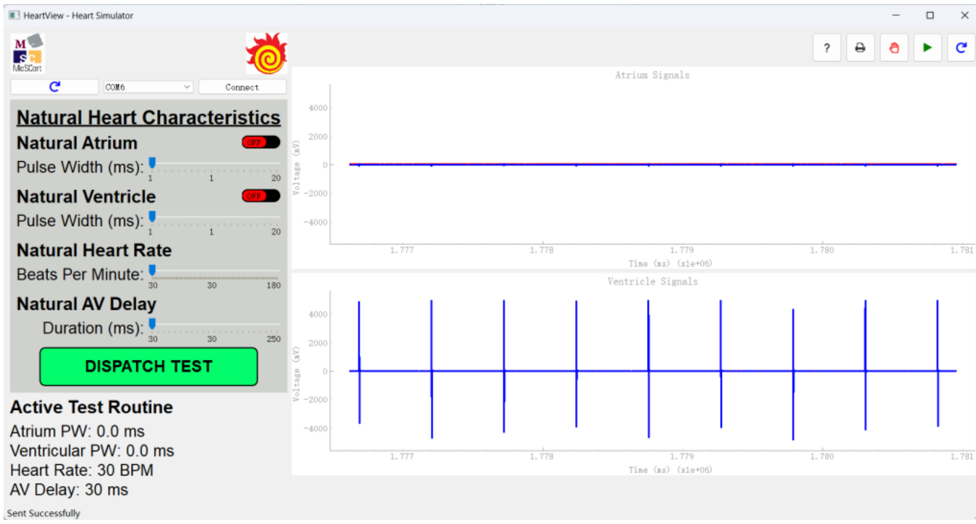

Figure 48: VOO Heartview Result

We can observe in the figure, pacemaker is pacing to ventricle at a fixed frequency of 90Hz in VOO mode.

VVI

1. FRONTEND_CTRL

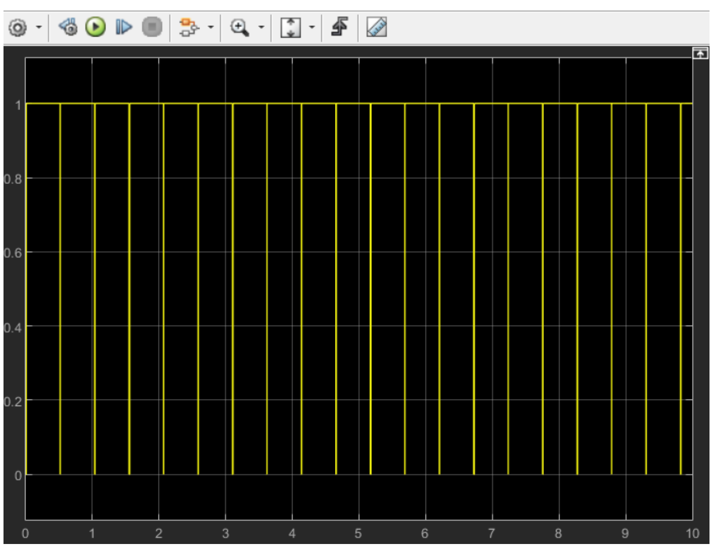

Figure 49: FRONTEND_CTRL

FRONTEND_CTRL is used to activate the sensing circuitry (FRONTEND_CTRL = true will activate the sensing circuitry and vice versa). In figure 36 the frontend control will be turned off when no heart event is sensed, and we need to pace the heart.

2. PACE_CHARGE_CTRL

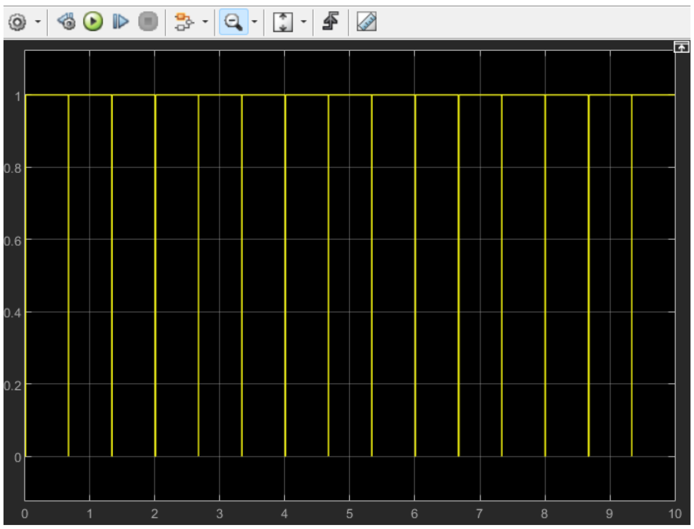

Figure 50: PACE_CHARGE_CTRL test result

This variable controls capacitor C22 (PACE_CHARGE_CTRL = true charges C22) The frequency of the pace charge control is 90Hz which is the rate we set for pacing. The pulse width is 1.5ms in setting. The figure above shows the test result of PACE_CHARGE_CTRL.

3. PACE_GND_CTRL

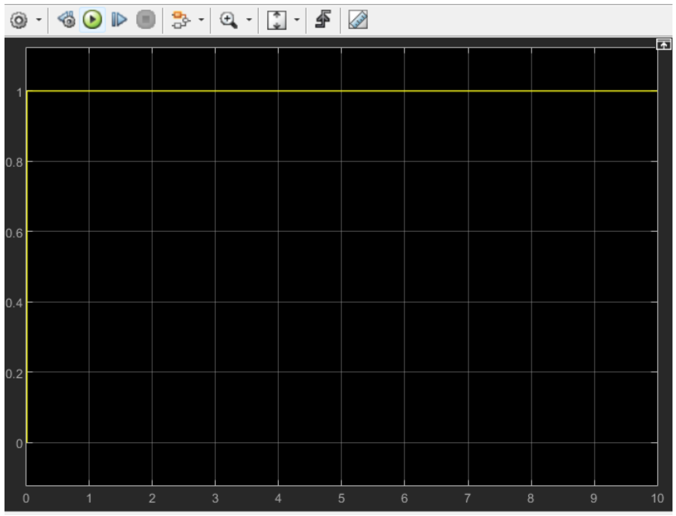

Figure 51: PACE_CHARGE_CTRL test result

The PACE_GND_CTRL test result in figure 38 show that this variable is always set to HI because it controls the switch directly following the tip, and it will prevent the current flow directly into human's heart.

4. VENT_PACE_CTRL

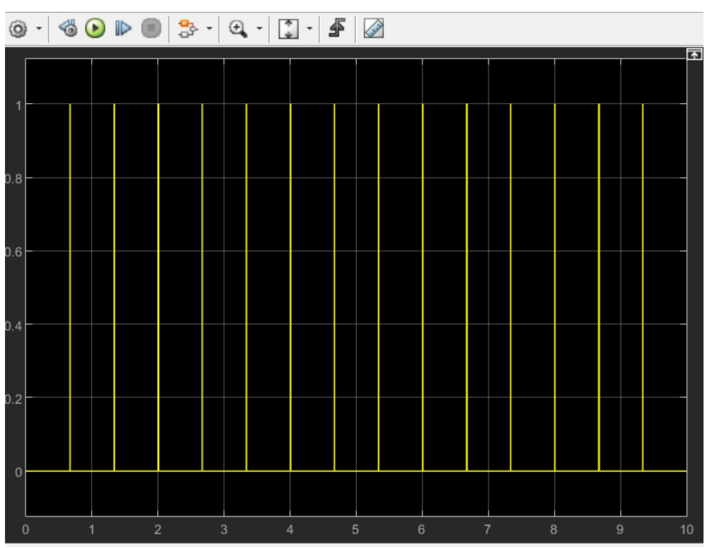

Figure 52: ATR_PACE_CTRL test result

This variable is the pacing signal we want to generate on the ventricle, the signal has frequency of 90Hz, and pulse width equal to 1.5 ms.

5. VENT_GND_CTRL

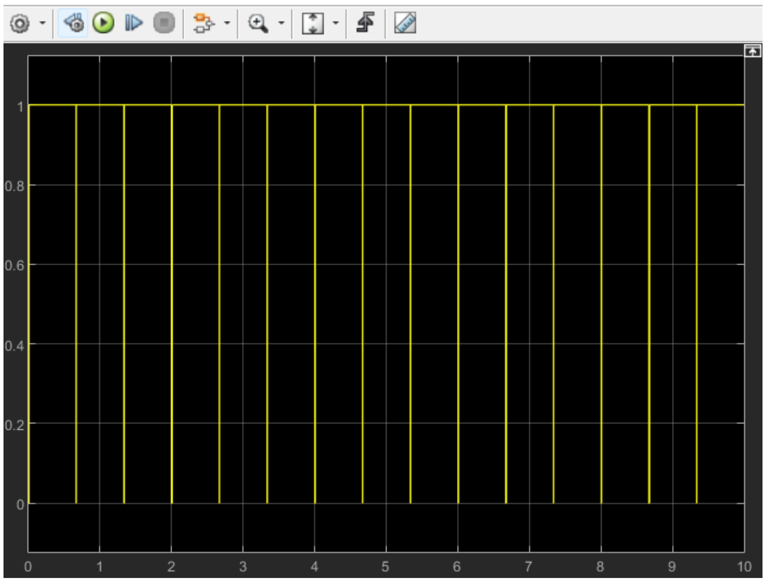

Figure 53: VENT_GND_CTRL test result

VENT_GND_CTRL is the opposite of VENT_PACE_CTRL, it is used to discharge C22.

6. ATR_PACE_CTRL / ATR_GND_CTRL

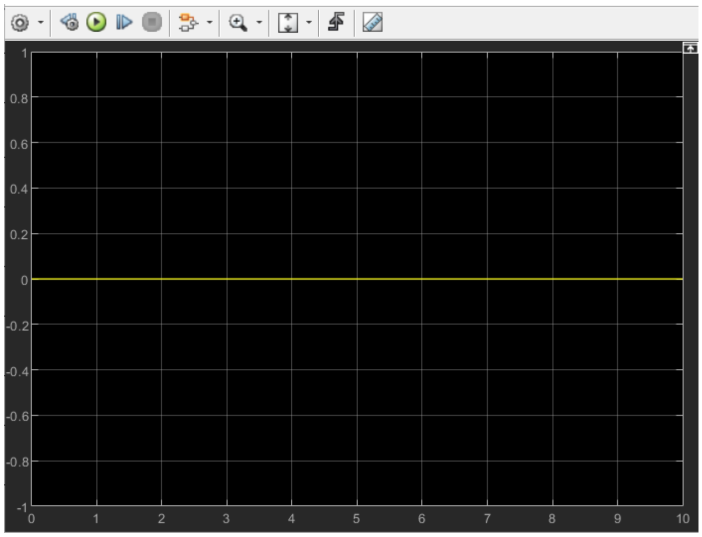

Figure 54: ATR_PACE_CTRL and ATR_GND_CTRL test result

In VVI mode, we are only pacing to the ventricle, therefore the variable for pacing atrium will be set to low.

VVI Heartview Result

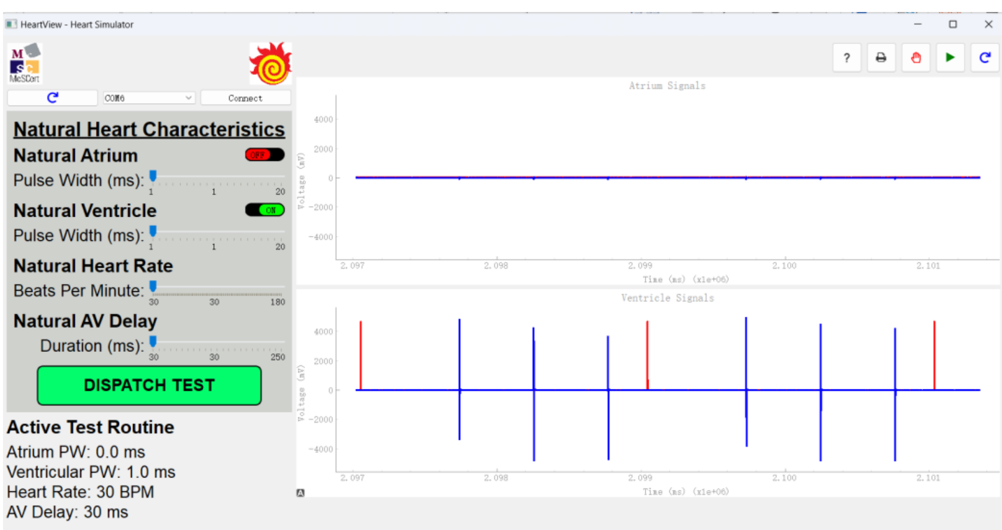

Figure 55: VVI Heartview Result when heart is beating slower than rate.

In figure 42, the heart is beating slower than the predefined rate, which requires the pacemaker to pace the heart at the fixed rate. The heart rate is 30ppm and the rate of pacing is 90ppm. The result is correct since between two heartbeats, there are 3 pacing.

Figure 56: VVI Heartview Result when heart is beating faster than rate.

The above figure shows that the heart is beating faster than the rate we defined. The pacemaker does not need to do any pacing operation to the heart. Therefore, there are no pacing pulse shows in the ventricle signal.

To test the new adding modes and the serial functions. We are going to set testing scenario for each new mode through DCM and verify the result on Heartview to see if the performance reaches the requirements. To simulate different motion states, there is an accelerator on the pacemaker board as mentioned above. We are going to shake the board to speed up the device.

AOOR

Under this mode, the pacemaker should keep pacing atrium constantly. While there is an acceleration increased, the pacing pulse should appear more frequently, and when the acceleration stops, the pulse should return to normal as preset.

After connecting the device, the first step is to fill in all required parameters in DCM and submit the data to the pacemaker as shown below, this is used to test the Serial communication.

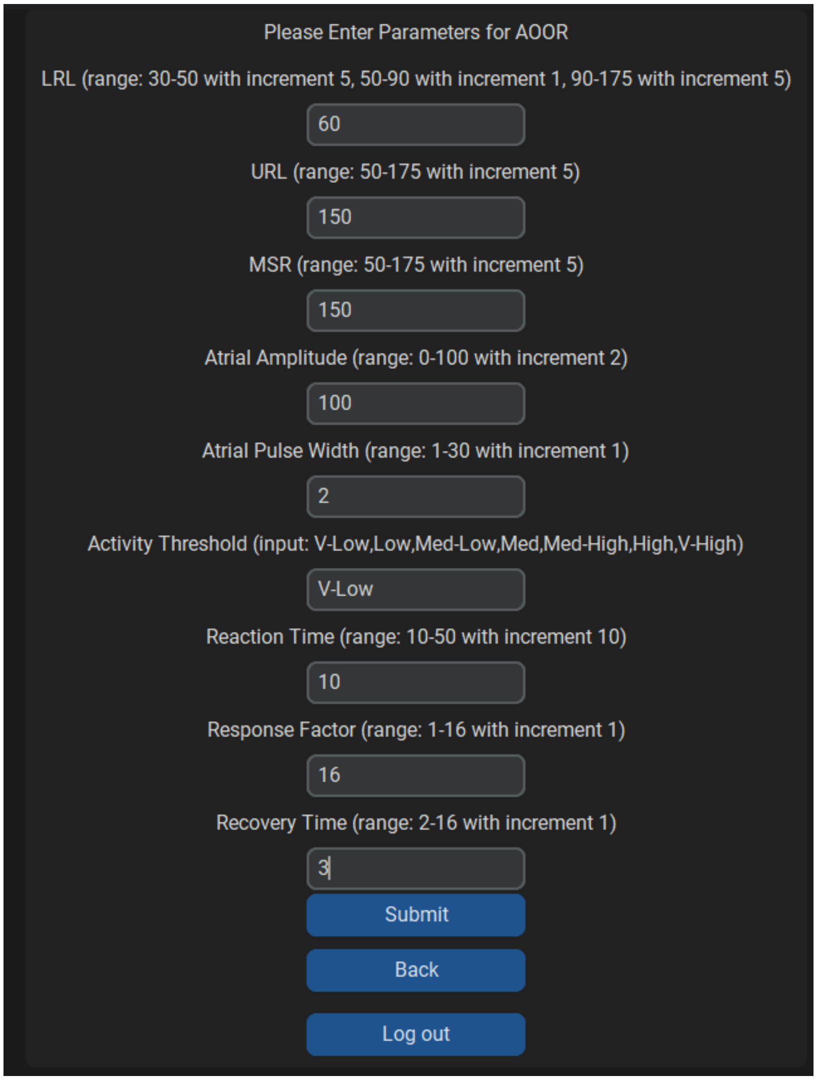

Figure 57: DCM input data

It turns out that the pacemaker is working as expected. From figure X to figure X, in Heartview below, it starts to pacing the atrium, same as the values inputs from DCM, and when we shake the pacemaker, the frequency of pacing pulse increases. The pulse goes back to normal after stopping shaking, which represents the motion level return to normal.

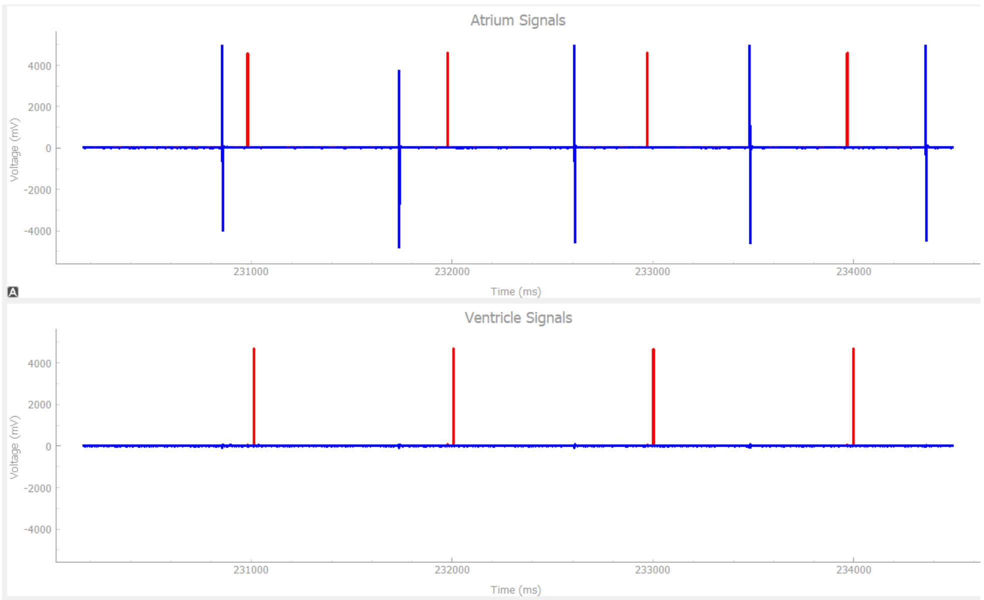

Figure 58: AOOR pacing

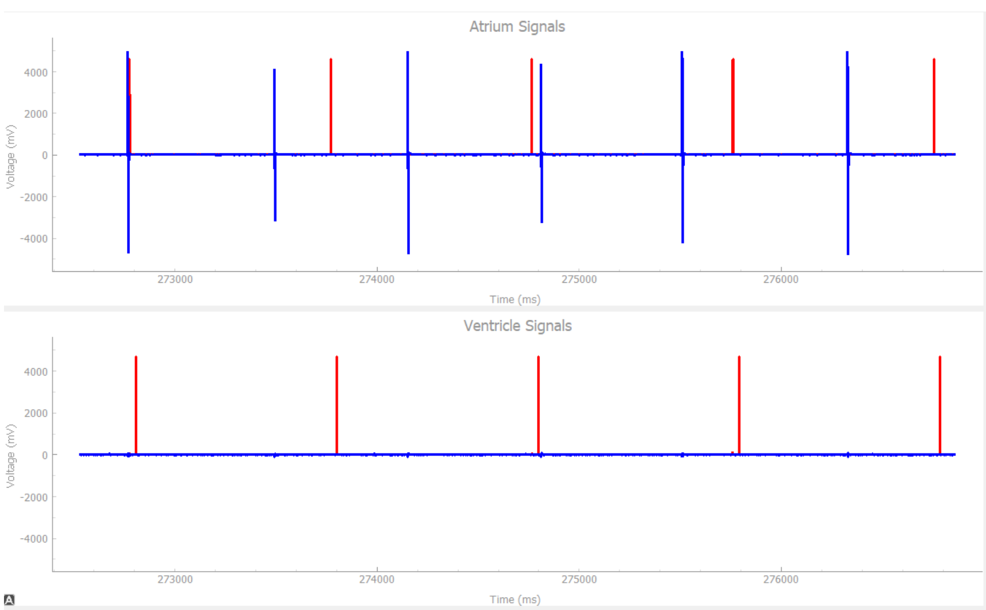

Figure 59: AOOR pacing when motion level is increased.

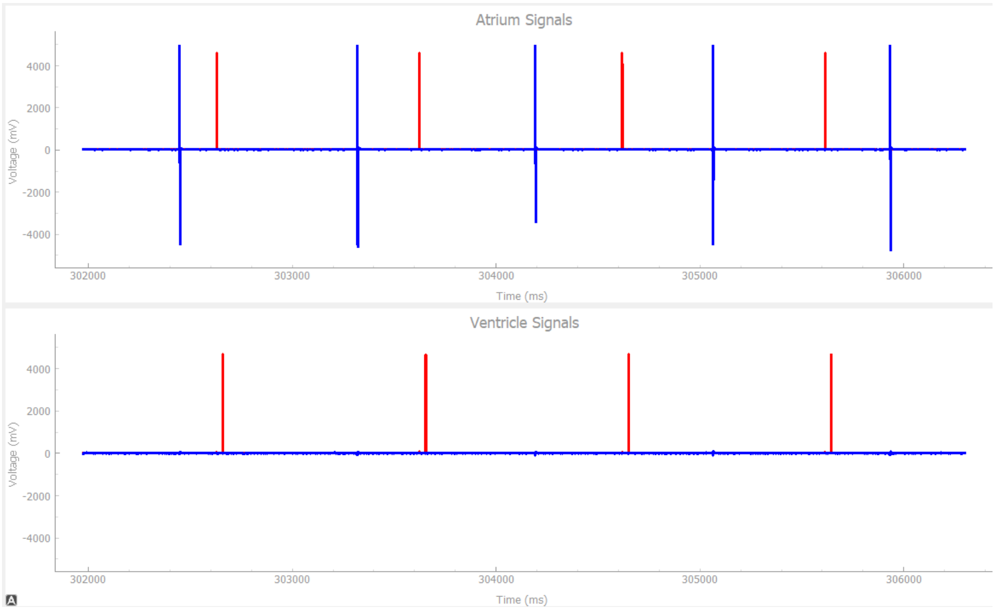

Figure 60: AOOR pacing back to steady state.

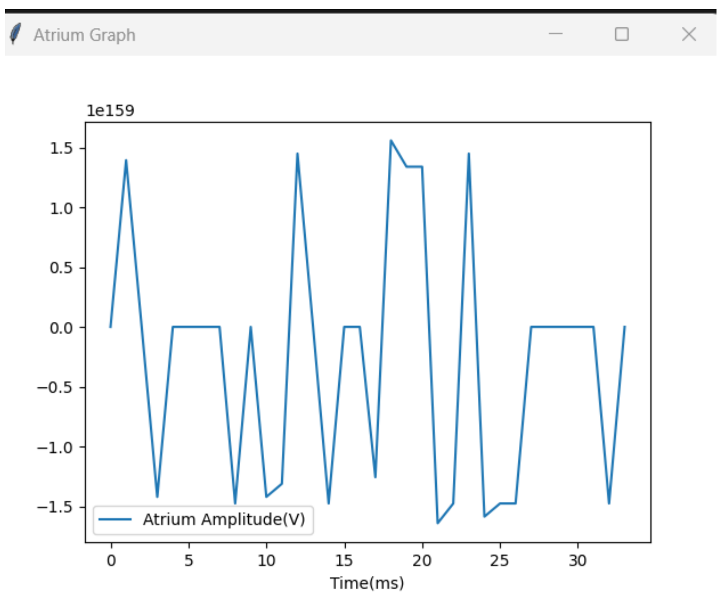

Figure 61: AOOR atrium plot

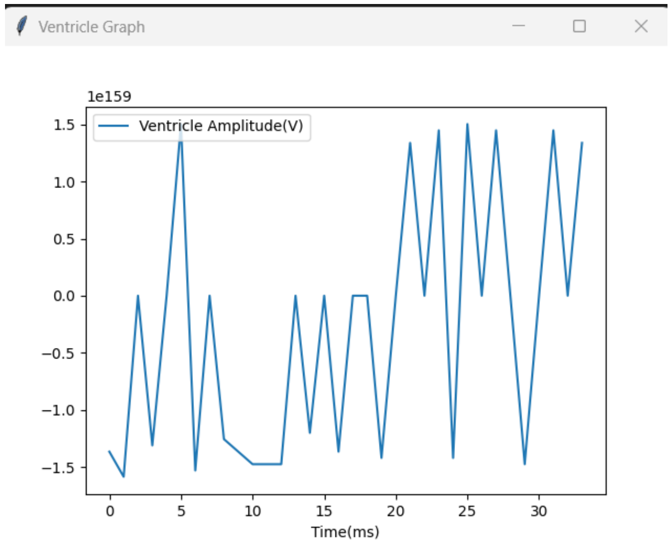

Figure 62: AOOR ventricle plot

Based on the above two figures, it also shows that our pacemaker successfully send both ATR_signal and VENT_signal to the DCM.

VOOR

The test of VOOR would be similar to AOOR. For safety considerations, we would like to repeat the test for each mode to ensure that the functions work as expected.

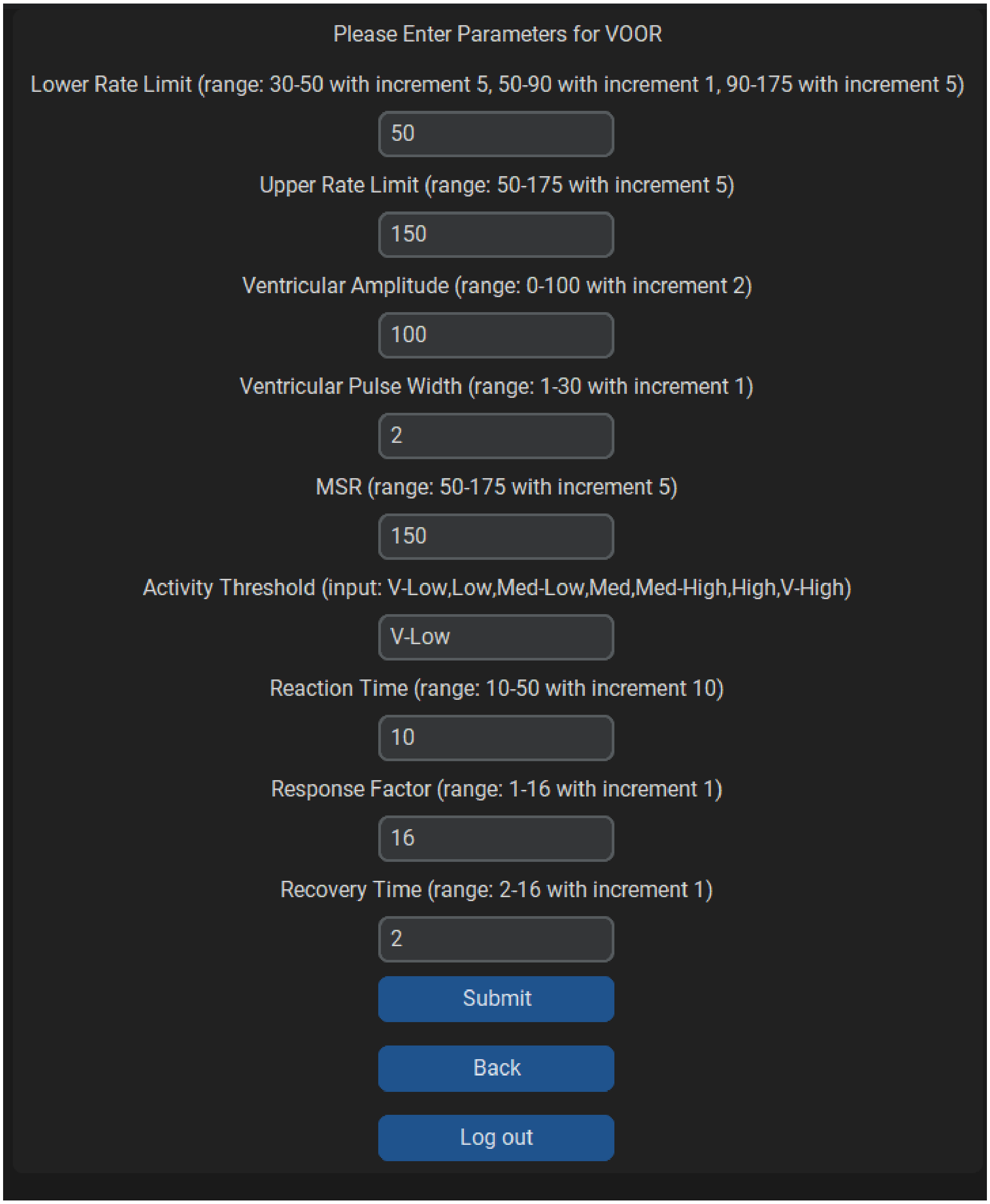

Figure 63: VOOR DCM input

To test the flexibility of our system, we change the rate while input the data, and as shown below, the system works as expected while it is pacing the ventricle and when the motion increases, the frequency increases and then back to normal.

Figure 64: VOOR pacing

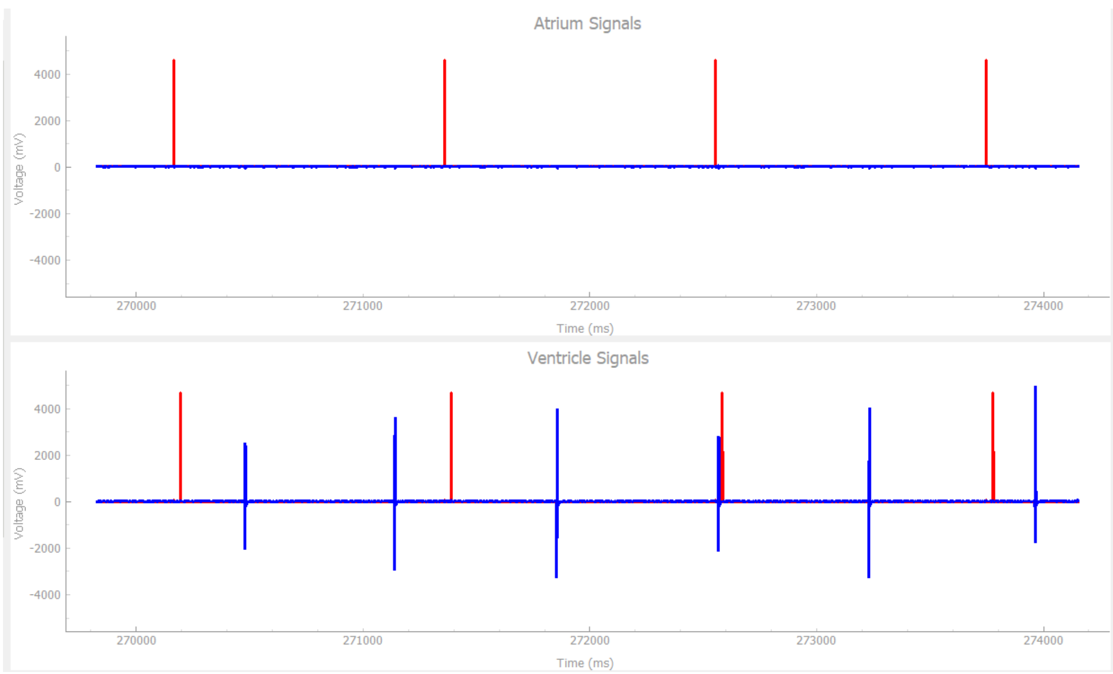

Figure 65: VOOR pacing when motion level is increased.

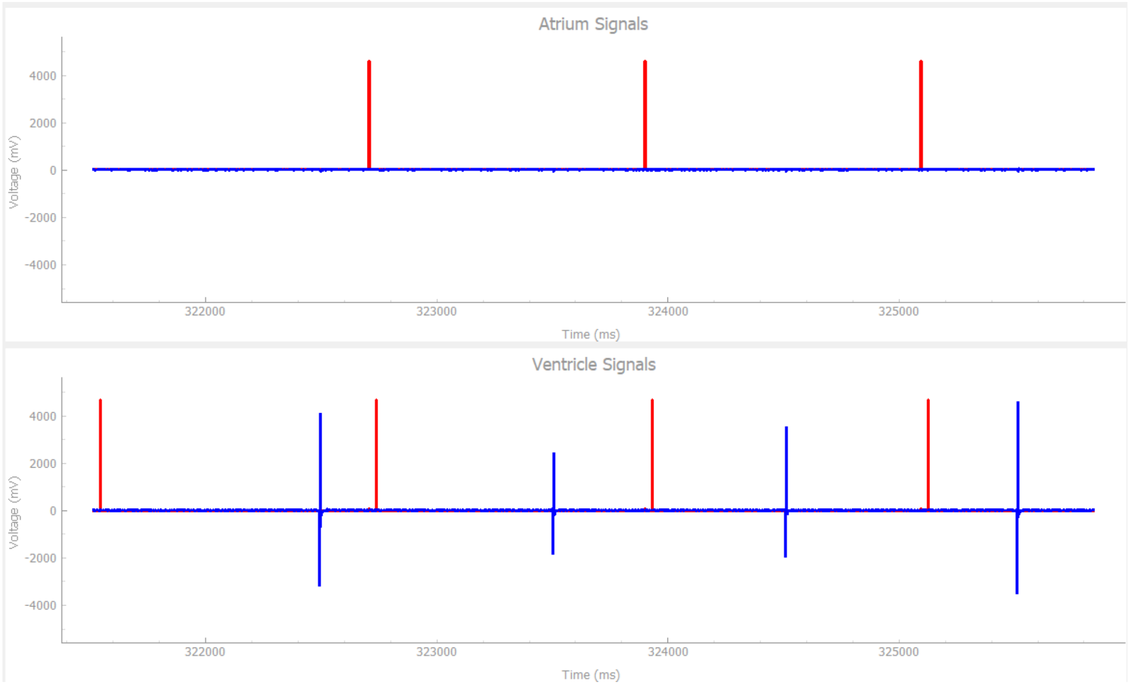

Figure 66: VOOR pacing return to normal

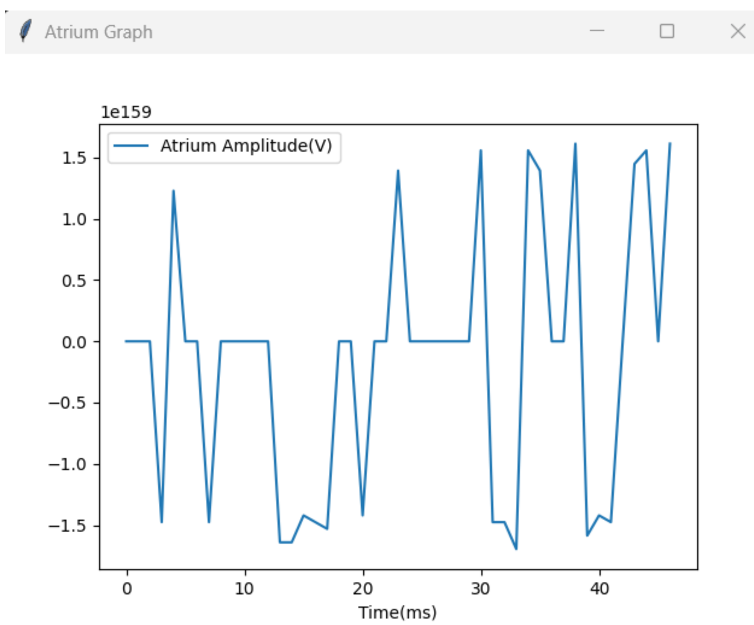

Figure 67: VOOR Atrium graph

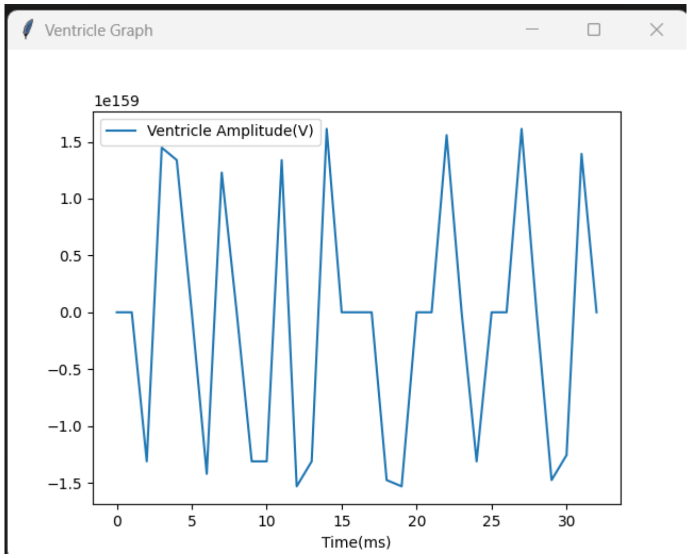

Figure 68: VOOR Ventricle graph

We can also receive the egrams for VOOR model on the DCM side.

AAIR

As explained, we need to set a threshold for pacing just like AAI mode. Besides the inputs as usual, I decrease the size amplitude for testing purpose.

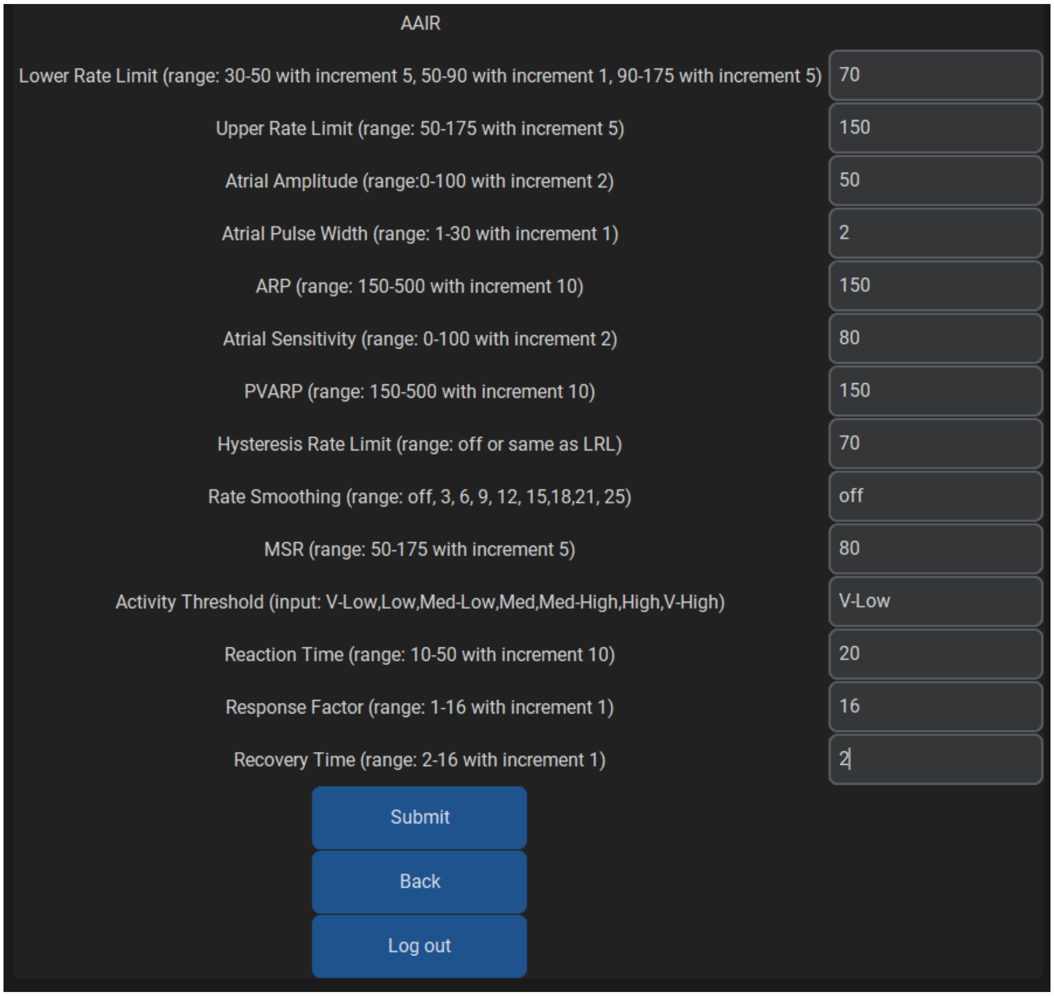

Figure 69: AAIR DCM input

Figure 70: AAIR pacing

As shown above, the amplitude decreases corresponding to the input from DCM. It's pacing since the current heart rate is lower than the preset threshold.

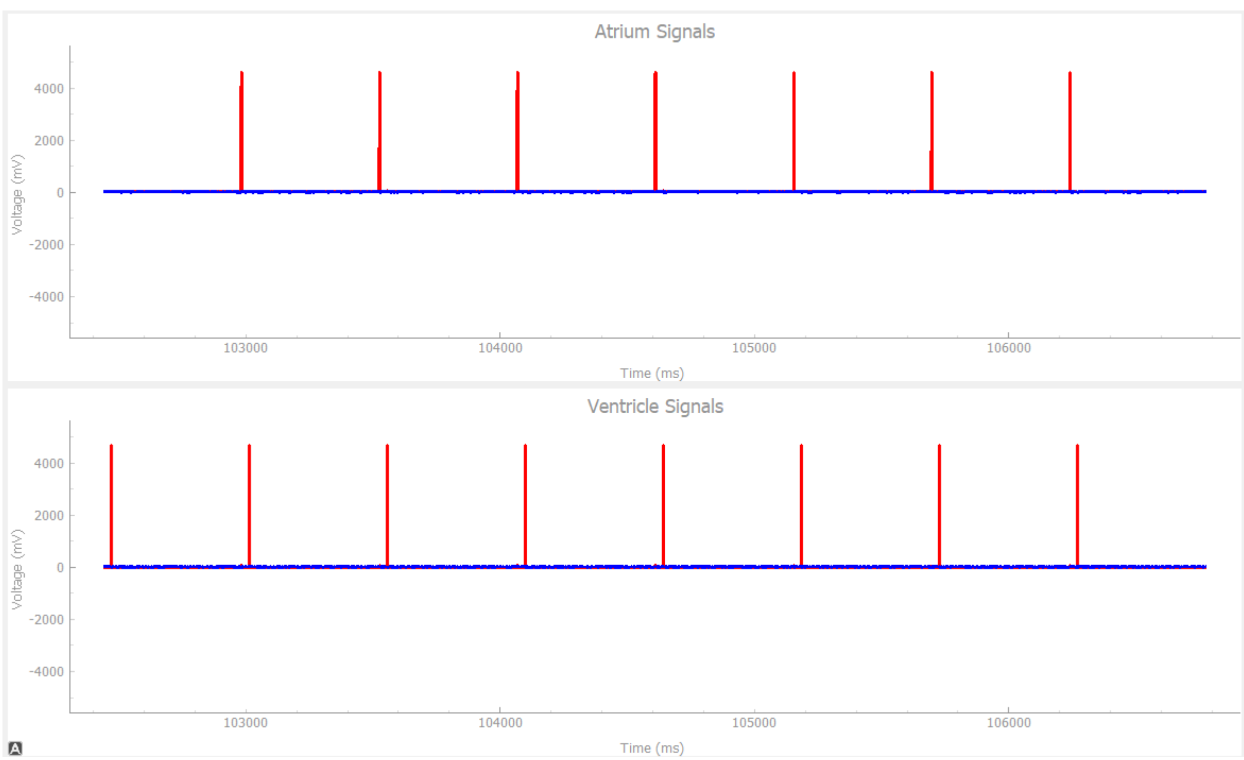

Figure 71: AAIR rate above threshold

Adjust the rate to be higher than the preset threshold, there would be no pacing which is the same performance as AAI mode. Based on Figure X below, at this time the motion level increases, there would be pacing again since the threshold is increased corresponding to the activity level. The output is the same as expected and AAIR pass the test.

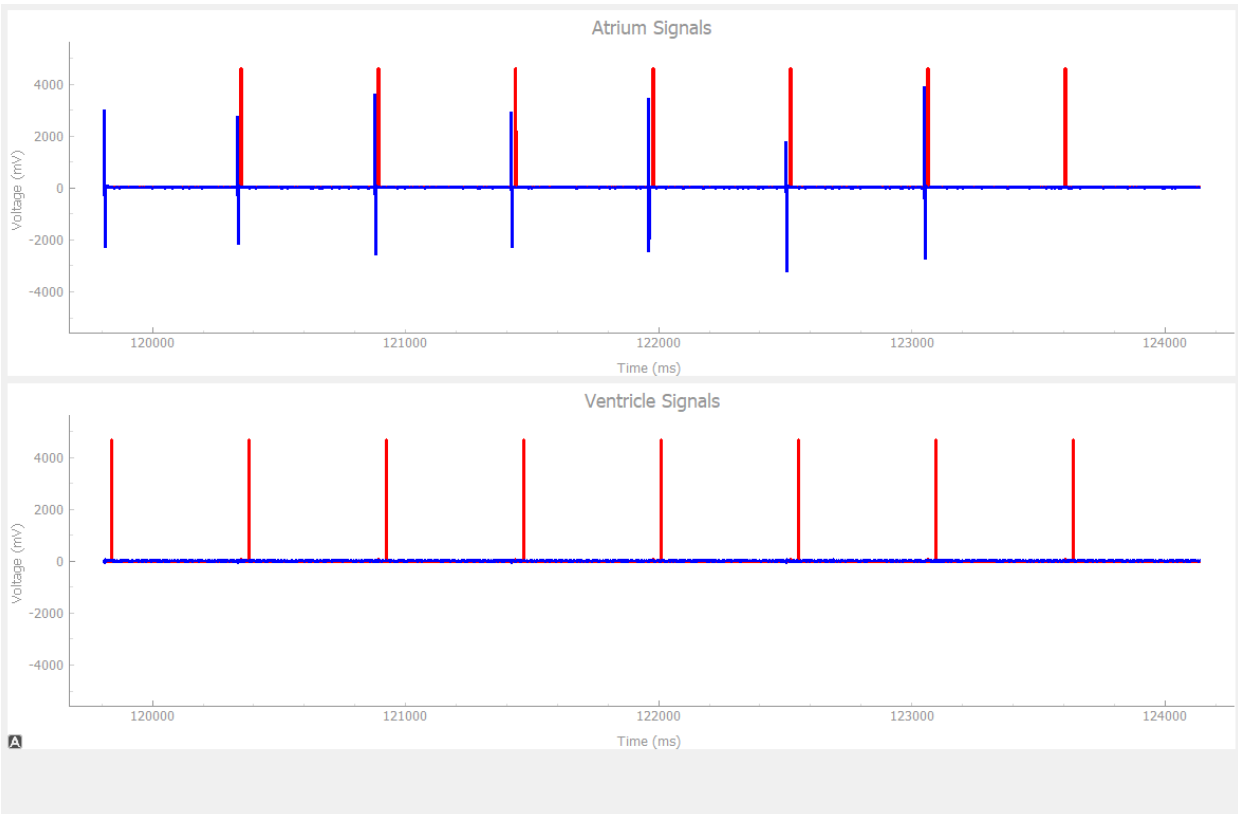

Figure 72: AAIR rate above threshold and motion level increases

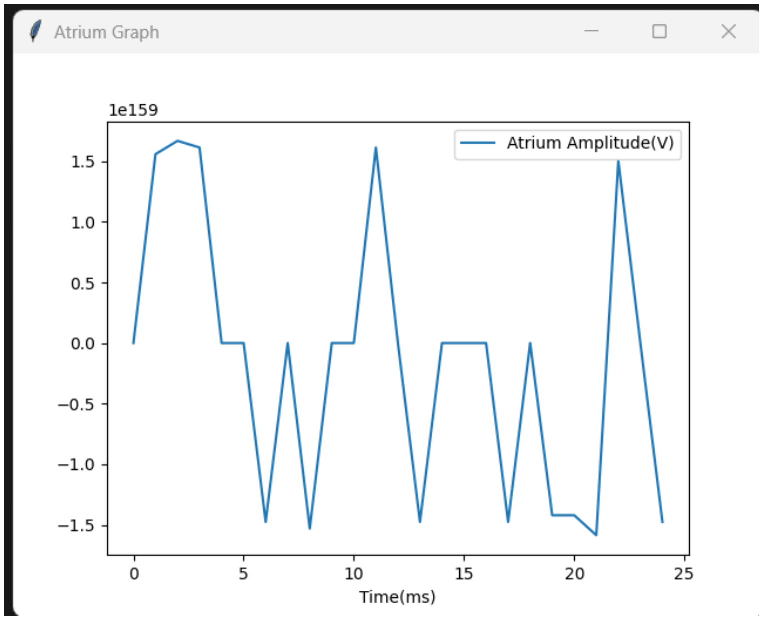

Figure 73: AAIR Atrium graph

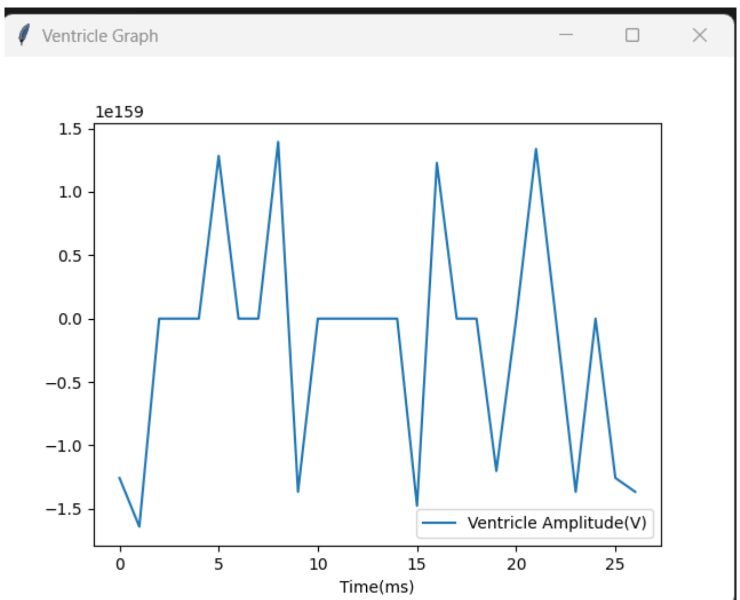

Figure 74: AAIR Ventricle graph

We can receive and check the egrams for AAIR as usual.

VVIR

Like AAIR, in order to test VVIR mode, first we need to fill in the data on the DCM side and transfer them to the board. As shown in Figure X, the pulse width is increased.

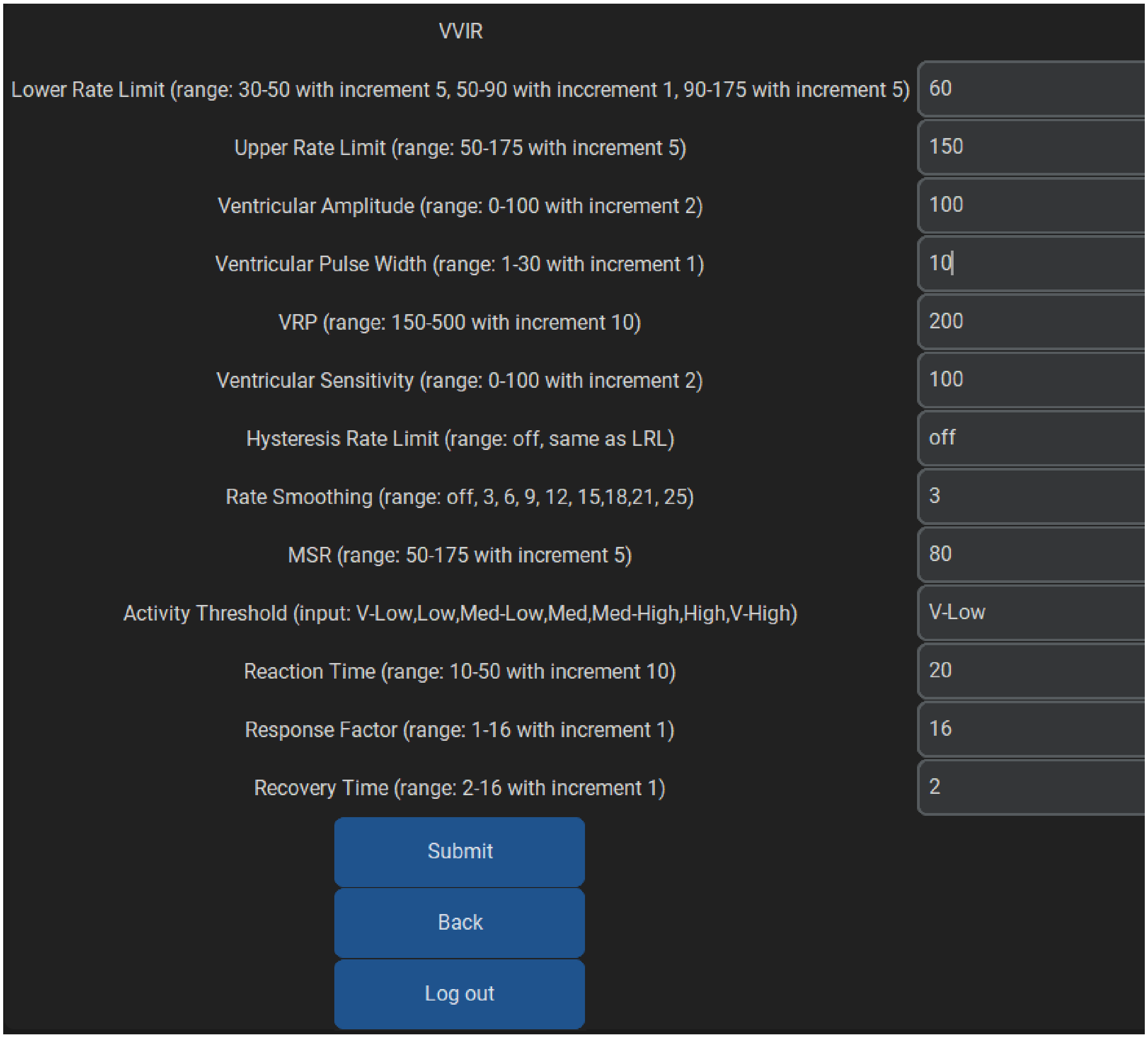

Figure 75: VVIR DCM inputs

Figure 76: VVIR pulse pacing

When the rate is set below the threshold, the pacing appears I the ventricle diagram as expected. And when the rate is above the threshold, the pacing disappears until the activity level increases as shown below.

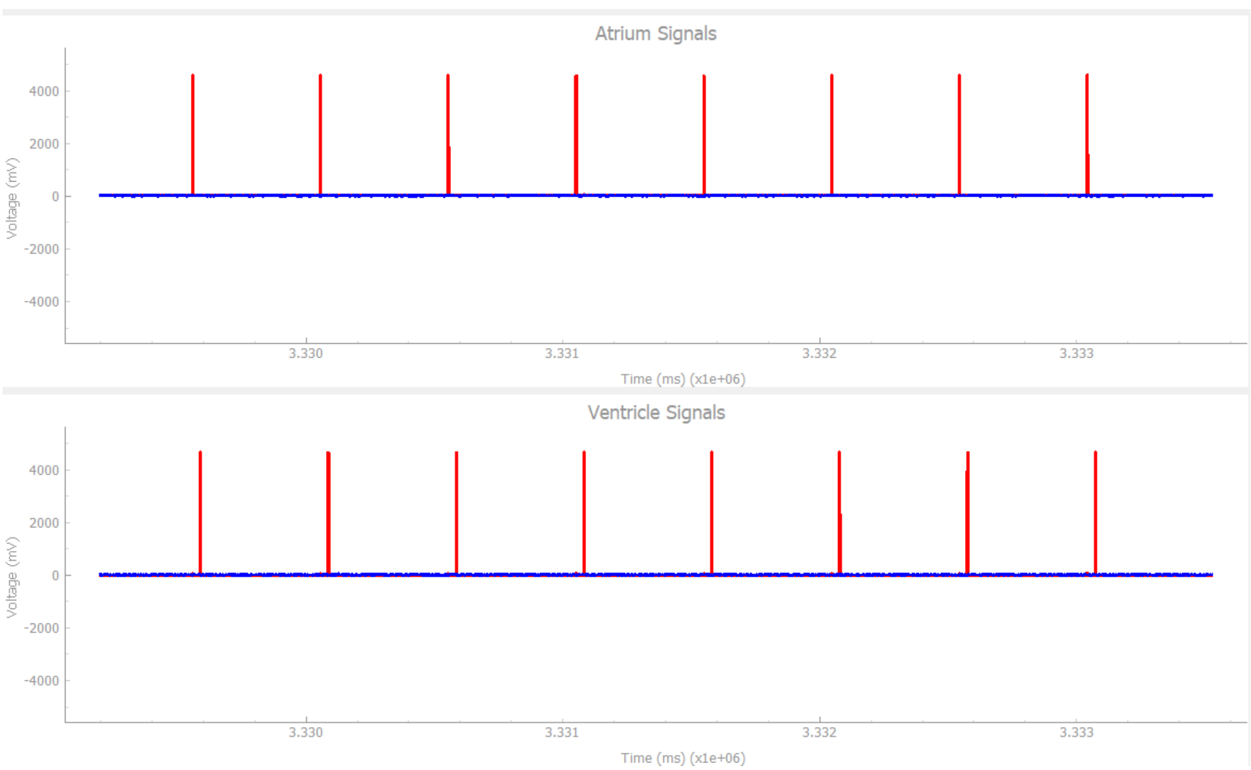

Figure 77: VVIR when the rate is above threshold

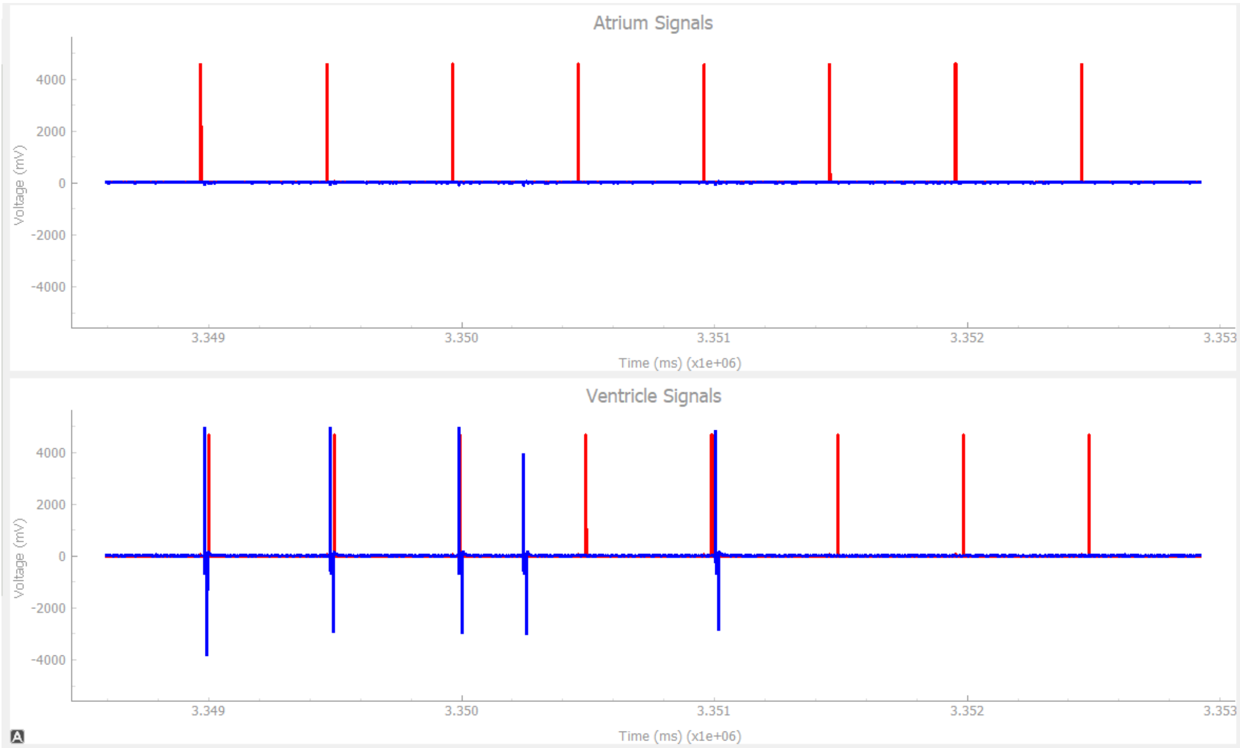

Figure 78: VVIR activity level increases

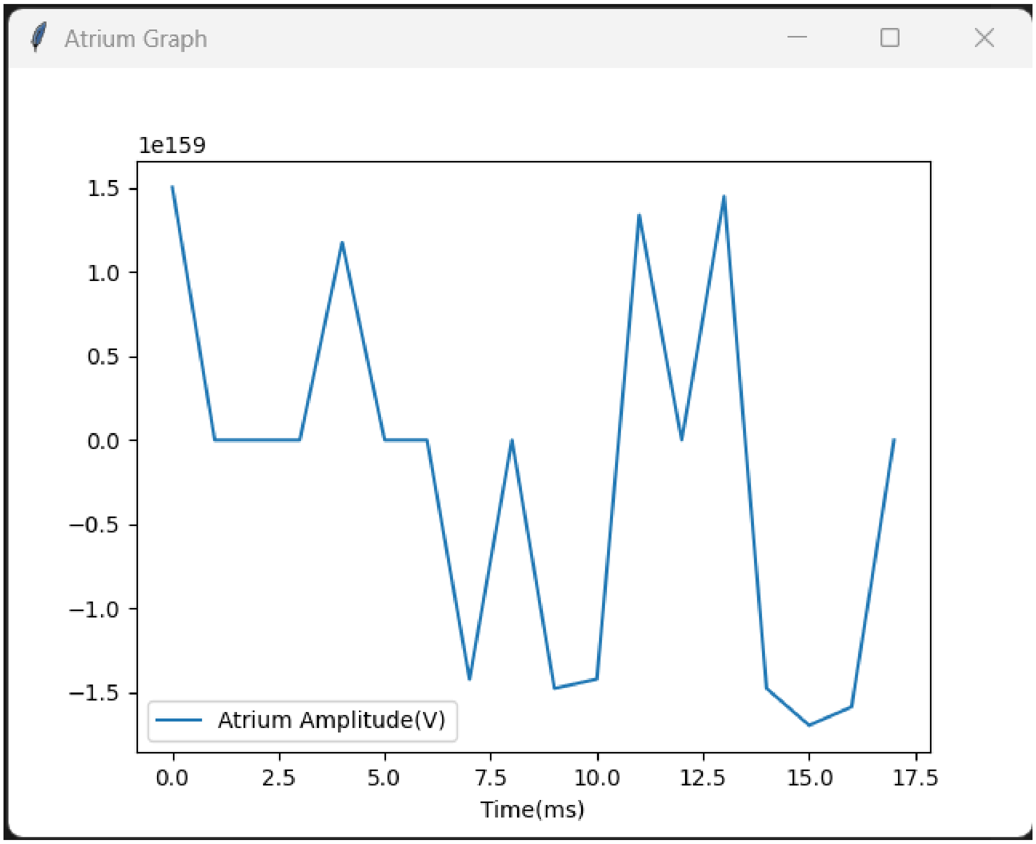

Figure 79: VVIR Atrium Plot

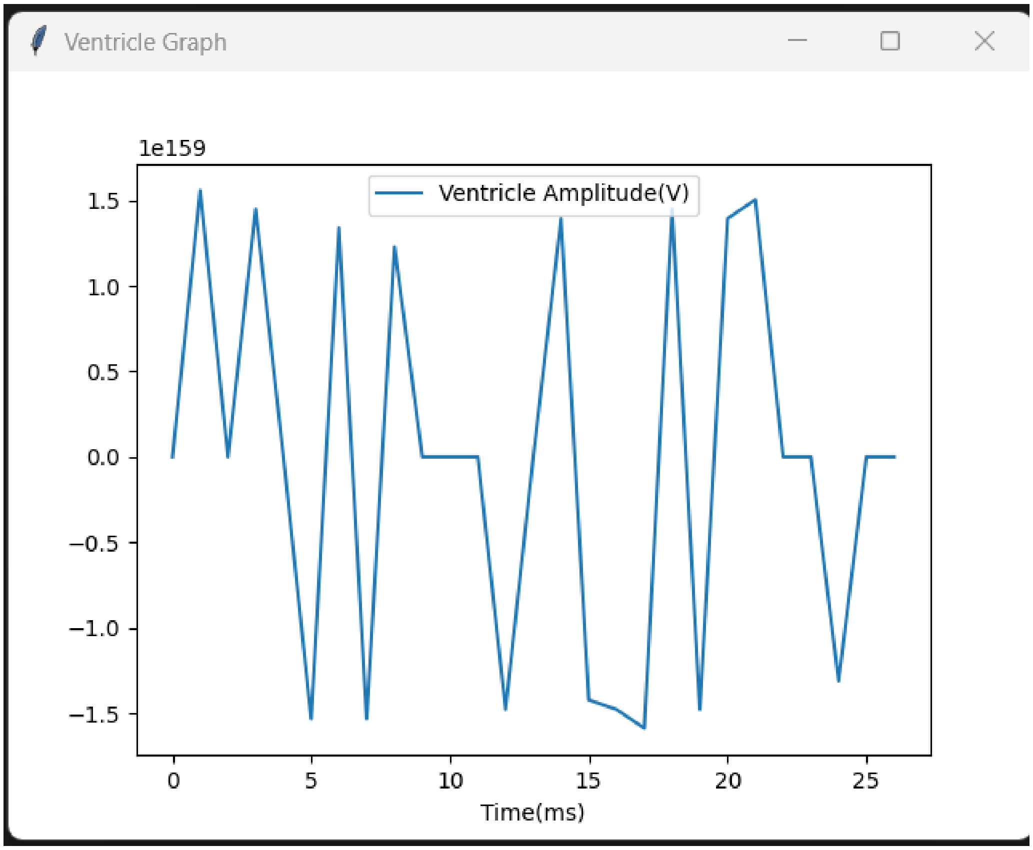

Figure 80: VVIR Ventricle Plot

Like other modes, VVIR passes the egram test.

Improvements and likely changes

We have successfully implemented eight modes that are able to simulate the normal usage of pacemaker, there is still space left for us to improve. We shall try to implement the bonus modes DDD and DDDR since these are two important modes that pacemaker works for both atrium and ventricle. Besides continuing to develop functionality, we should also improve the testing method. Since we are using Jlink to compile, we can do some research to see if we can apply Jscope to debug and see if the parameters are working properly before connecting with the DCM side. We also need to shrink the data length while doing serial communication to save space and make the whole process more efficient.

References

[1] "+ Pulse Width." Accessed: Oct. 12, 2023. [Online]. Available: https://rfmw.em.keysight.com/DigitalPhotonics/flexdca/UG/Content/Topics/Oscilloscope-Mode/Time-Measurements/pos_pulse_width.htm

[2] "PACEMAKER System Specification," 2007.

[3] G. Meyer, "Functionality and Circuitry Explanation Pacemaker Microcontroller Shield," 2020.

[4] M. Kehinde, "What is a Pacemaker? The Cardiac Conduction System and the Artificial Pacemaker," 2020.

[5] M. Kehinde, "Accelerometer Rate Adaptive Pacing Elaboration of Requirements," 2019.