SFWRENG 3K04: Software Development

Assignment 2

Simulink

Group 25

December 1st, 2021

2. Simulink

2. Simulink

2.1 Simulink Overview

The purpose of the simulink implementation is to simulate the functionalities of a pacemaker. The pacemaker design consists of 10 working states in total, which are VOO, AOO, VVI, VOO, VOOR, AOOR, VVIR, AAIR, DOO, and DOOR. VOO, AOO and DOO are states that will apply direct pacing, VVI and AAI are states that need to sense the heart activities before determining whether to pace or not. VOOR, AOOR, VVIR, AAIR are modes that require adaptive rate pacing, which means the pacing rate depends on the activity level sensed by the accelerometer on the pacemaker.

2.2 Parameters

2.2.1 Programmable Input Parameters Regarding Chart (Inputs)

Name	Initial Values	Units	Data Type	Functionality
Mode	1		uint8	Select from mode 1 - 10
LRL	60	ppm	uint8	Default pacing pulse rate
MSR	160	ppm	uint8	Maximum pacing rate allowed for adaptive pacing modes
ARP	250	msec	uint16	Atrial Refractory Period
VRP	250	msec	uint16	Ventricle Refractory Period
ATR_AMP	3.3	Volt	single	Desired pacing PWM for atrial, the value will be convert to duty cycle later
VENT_AMP	3.3	Volt	single	Desired pacing PWM for ventricle, the value will be convert to duty cycle later
ATR_Width	10	msec	uint8	Pacing width for atrial pacing

VENT_Width	10	msec	uint8	Pacing width for ventricular pacing
ATR_Sensitivity	3	Volt	single	Threshold voltage for sensing a natural atrial beat.
VENT_Sensitivity	3	Volt	single	Threshold voltage for sensing a natural ventricular beat.
Threshold	1.1		single	Threshold value for adaptive pacing mode. It decides when the LRL should increase. Range 1 - 6.9
RF	16		uint8	Response factor, which decides the desired pacing rate.
Reaction_Time	10	sec	uint8	The time takes to increase from LRL to the MSR.
Recovery_Time	30	sec	uint8	The time takes to decrease from MSR to LRL.
AV_delay	150	msec	uint16	AV delays for dual chamber pacing modes
URL	160	ppm	uint8	Upper rate limit for pacing

2.3 Design Decision

2.3.1 Overview

The simulink design consists of 6 parts in general. The input chart, the input subsystem, the main pacemaker, the data processing MatLab function group, the output subsystem and the set_parameter() function block. By dividing the design into smaller parts, the dataflow relationship is more clear. The input data is being set in the input chart, then it flows into two directions, one to the set_parameter() function block where the data is sent back to the DCM, the other is to the input subsystem where some data pre-processing occurs. From the input subsystem, while part of the data flows into the MatLab function group to perform some calculations, the other part together with the calculation results from the function group flow into the main pacemaker chart. Finally, the data flows to the output subsystem.

2.3.2 AOO and VOO Design

In these two modes, the design will continue pacing the atrial/ventricle. It is achieved by switching between the Charging_Discharging state and the Pacing state. The Charging_Discharging state integrates the functionalities to both charge/discharge the capacitor. The Pacing state simply paces the atrial/ventricle.

2.3.3 AAI and VVI Design
Firstly, the design will first charge C22 / discharge the blocking capacitor. Then, a state named Sensing_VENT/ATR is inserted to determine if the ATR/VENT_CMP_DETECT detects a valid heartbeat. If the heart activity is sensed, it will loop back to the initial Charging_Discharging state to prepare for the next detection, otherwise, it will proceed to the pacing state to pace the Atrial / Ventricle. After that, it will loop back to the initial Charging_Discharging state.

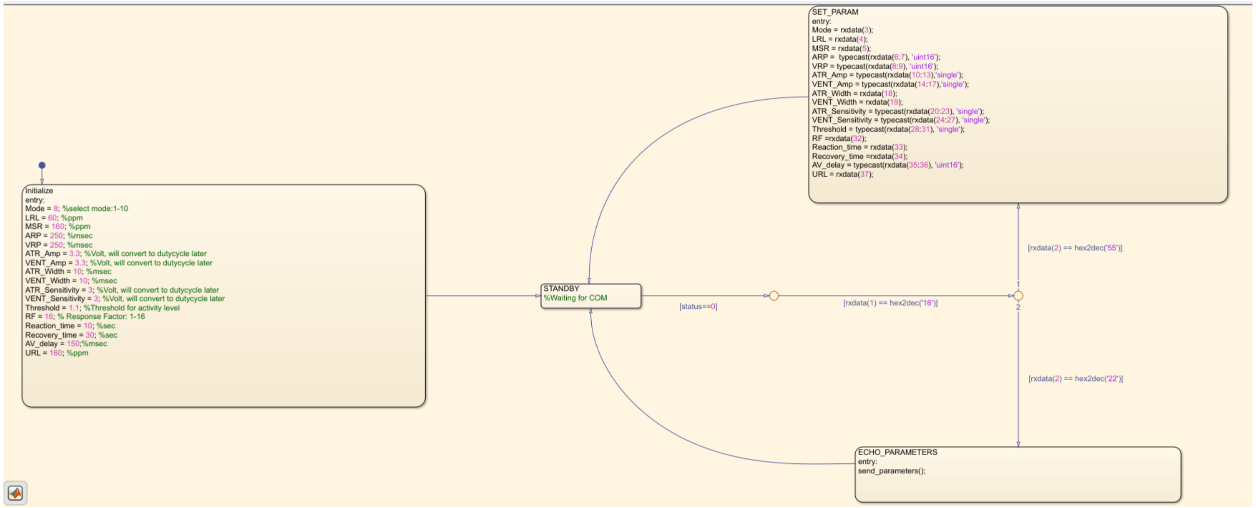
2.3.4 DOO Design
The DOO mode consists of 5 states in total. The states are, Charging_Discharging_A, Pacing_Atrial, ChargingV_DischargingA, Pacing_Ventricle, ChargingA_DischargingV. The first Charging_Discharging_A state is a lead in state where it will only be executed one time after entry, then the code will loop among the other 4 states in the order that is mentioned above. It will allow the pacing to occur in the desired sequence with proper AV delay.

2.3.5 AOOR, VOOR, AAIR, VVIR, DOOR Design
The adaptive rate modes share the same design logic as their corresponding normal modes mentioned above. The only difference is that the pacing period in the rate adaptive mode depends on the calculated rate rather than the LRL.

2.3.6 Subsystem Usage
The main purpose of the subsystem is to implement the hardware hiding, the input/output pin assignments are all done in the subsystem. Also, the subsystem also makes the design look more clean when some basic data processing is required, such as data type conversion, all the data are calculated based on double type.

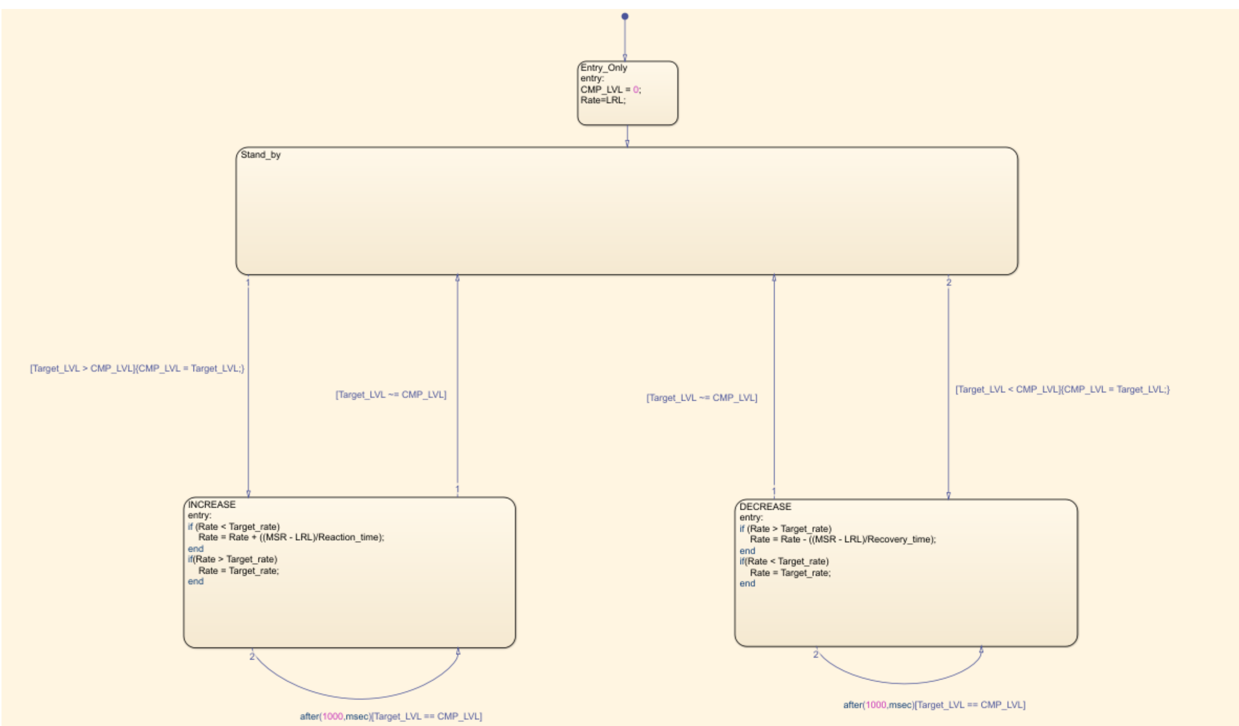
2.4 State Diagram

2.4.1 Input Chart



The Initialize state is used to set the initial value for all the parameters. And it will enter the standby state to determine whether to enter SET_PARAM state or the ECHO_PARAMETERS state. SET_PARAM state is used to receive the serial input and assign the data to the corresponding variables. ECHO_PARAMETERS is used to transfer all the data back to the DCM side by calling the function send_parameters(). (the function will be introduced in the '2.5 Function blocks' section)

2.4.2 Adaptive Rate Adjusting Chart



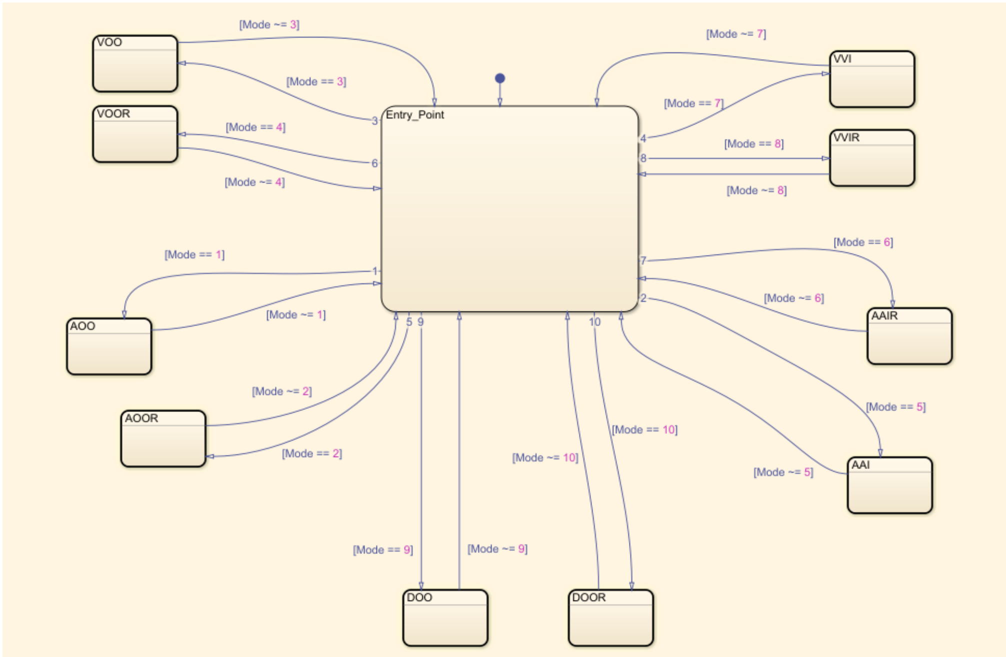
This state takes LRL, MSR, Reaction time, Recovery time, Target_rate and Target_LVL as input, variable 'Target_LVL' is the activity level determined by the previous MatLab function block, variable ‘Target_rate’ is the target rate of the adaptive mode which is also determined by the previous MatLab function block. These function blocks will be introduced in the ‘2.5 Function Blocks’ section later in this report. After processing, the state will return an output represented by variable ‘Rate’, which is the pacing rate used in the adaptive pacing mode. This rate is a dynamic variable, it will keep changing as the accelerometer data change.

The CMP_LVL is an internal parameter, it represents the current activity level of the pacemaker, its value will keep updating as the activity level changes. After its initial value is set, it represents the current activity level and is used to compare with the incoming activity level to decide if the pacing rate needs to increase or decrease, then the CMP_LVL is updated with that value and wait for the next comparison.

The chart starts with the entry state Entry_only, where Rate and CMP_LVL are set to default value (CMP_LVL = 0, Rate = LRL). Then it enters a Stand_by state, the code will compare the CMP_LVL with the Target_LVL. If (CMP_LVL < Target_LVL), it will enter the INCREASE state. If (CMP_LVL > Target_LVL), it will enter the DECREASE state. If (CMP_LVL == Target_LVL), it will wait at the Stand_by state. In INCREASE state, the value of Rate will increment with the formula every 1 sec: Rate = Rate + ((MSR - LRL)/Reaction_time), where ‘(MSR - LRL)/Reaction_time’ is the increment step, the increment will stop when the Rate reach Target_rate. As long as CMP_LVL is not equal to Target_LVL, the state will transfer back to Stand_by state for further decision. In DECREASE state, the logic is similar, the only difference is the formula, Rate = Rate - ((MSR - LRL)/Recovery_time), the Rate will decrease until it reaches the Target_rate. As long as CMP_LVL is not equal to Target_LVL, the state will transfer back to Stand_by state for further decision.

2.4.3 Main Pacemaker Chart

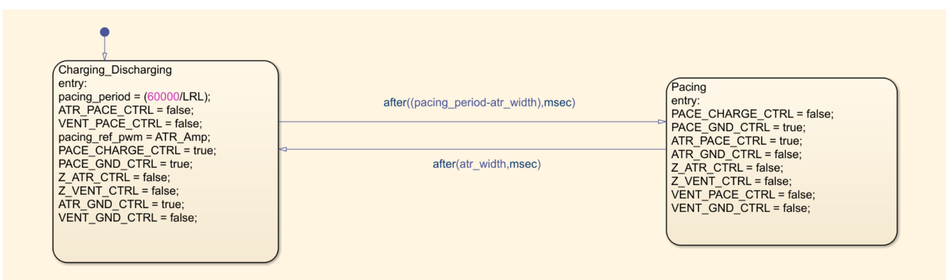
(1). Overview



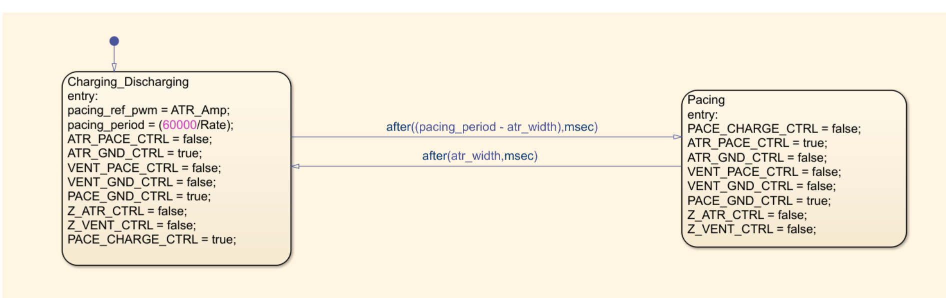
The entire process starts from the Entry_Point, where the desired pacing mode is determined by the value of input mode. There are 10 modes that could be selected.

(2). AOO and VOO. AOOR and VOOR

AOO mode



AOOR mode



AOO and VOO modes share similar logic. On the LHS, the Charging_Discharging state integrates the function of charging for the next pace and discharge after pacing occurs. On the RHS, the Pacing state executes the pacing after the capacitor has been charged. In these two modes, the pacing period are both determined by LRL.

AOOR and VOOR have exactly the same design diagrams as their OO modes. The only difference is the pacing period, the pacing period of the R mode is determined by the calculated Adaptive rate (represented by variable: Rate) rather than LRL.

(3). AAI and VVI. AAIR and VVIR

AAI mode

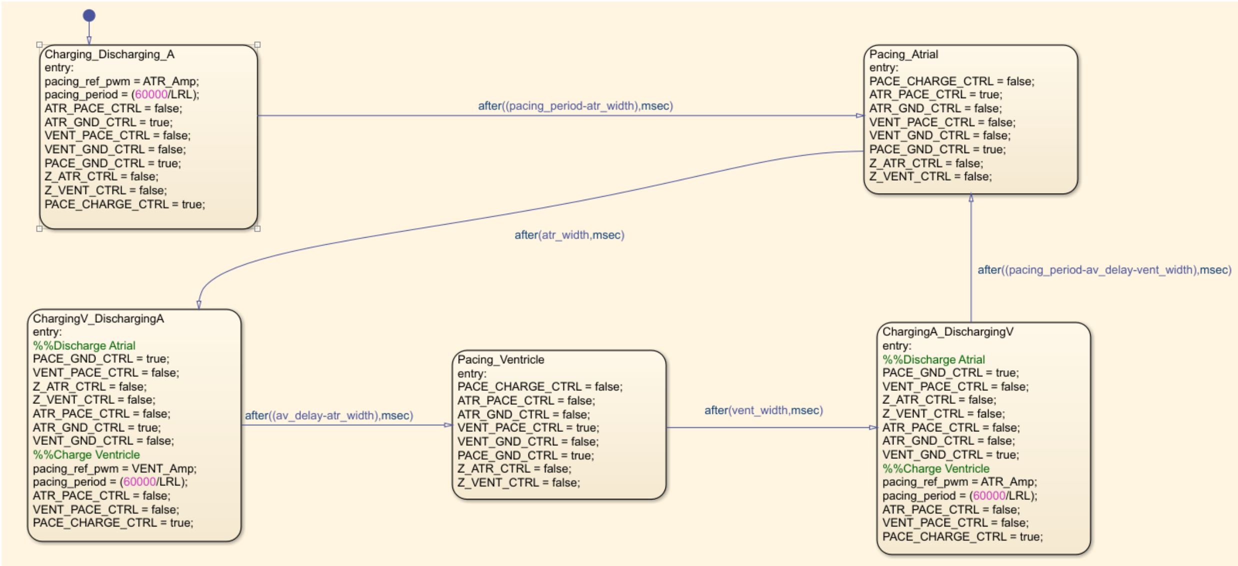
AAIR mode

AAI and VVI modes share similar logic. On the LHS, the Charging_Discharging state integrates the function of charging for the next pace and discharge after pacing occurs. The Sensing_ATR/VENT state creates the pacing inhibited time interval after natural beats have been sensed (It is when ATR/VENT_CMP_DETECT are true). On the RHS, the Pacing state executes the pacing after the capacitor has been charged and the inhibit time period has passed. In these two modes, the pacing period are both determined by LRL.

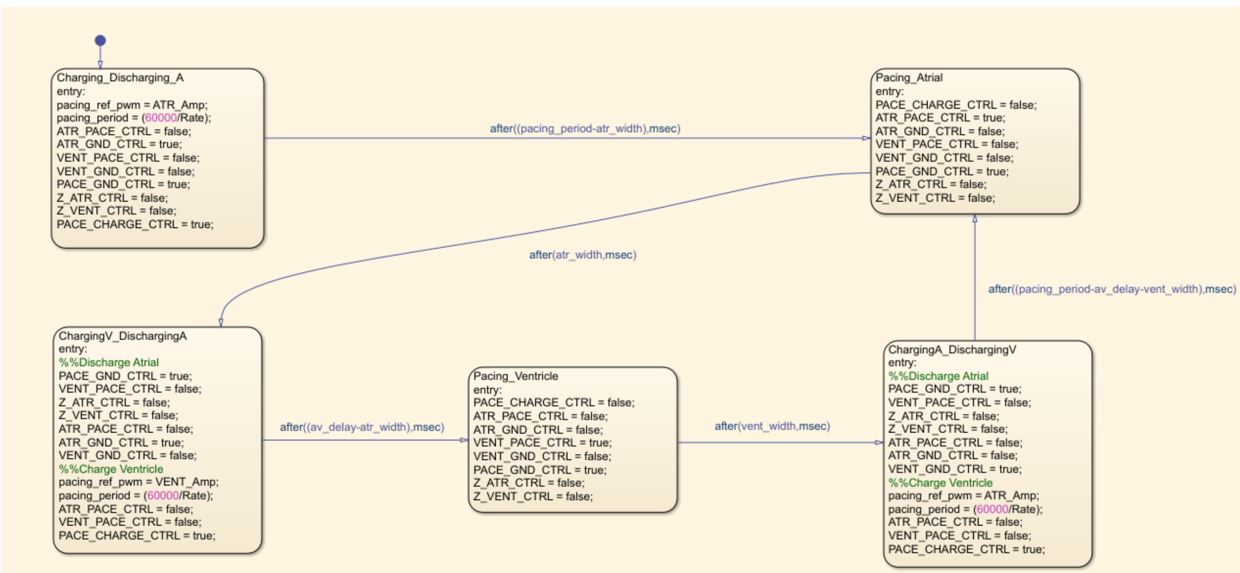
AAIR and VVIR have exactly the same design diagrams as their 'I' modes. The only difference is the pacing period, the pacing period of the R mode is determined by the calculated Adaptive rate (represented by variable: Rate) rather than LRL.

(4) DOO and DOOR

DOO mode



DOOR mode

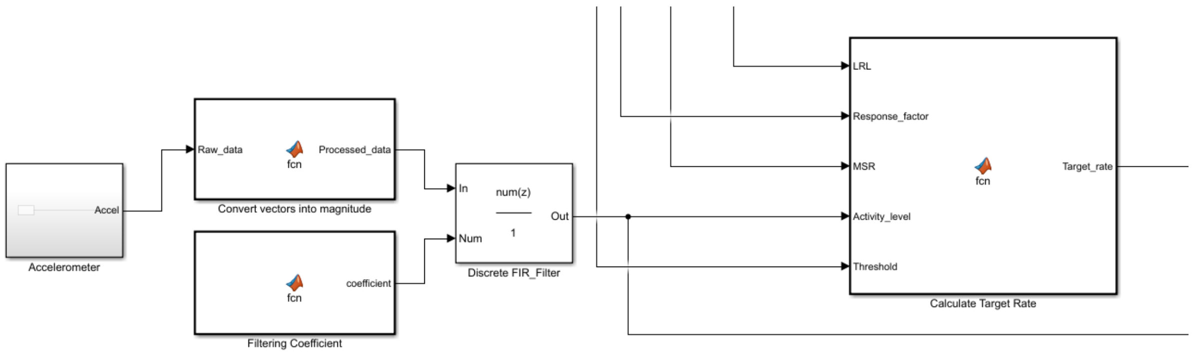


The DOO mode consists of 5 states in total. However, the entry state Charging_Dischaging_A is a boost up state which will only be executed one time after entering this mode, the code will loop among the other 4 states. The Charging_Dischaging_A state charges the capacitor for Atrial pacing, after(pacing_period - atr_width) msec, the Pacing_Atrial state executes the atrial pacing, then after (atr_width) msec, ChargingV_DischargingA state discharges for atrial and recharges for the next ventricular pacing. After(av_delay - atr_width) msec, Pacing_Ventricle state paces the ventricle, then after(vent_width) msec, the ChargingA_DischargingV state discharges for the ventricle and recharges the capacitor for next atrial pacing, after(pacing_period - av_delay - vent_width) msec, the Pacing_Atrial state paces the atrial again. For the rest of the time, the code will loop among the above 4 states. For this DOO mode, all the pacing period depends on the LRL.

The DOOR mode has the same design logic as the DOO mode, the only difference is the pacing period depends on the calculated adaptive pacing rate (represented by variable ‘Rate’).

2.5 Function Blocks

2.5.1 Overview of the Accelerometer Data Processing Function Group



This function group takes the raw data from the accelerometer and outputs the calculated activity level and the corresponding target rate.

2.5.2 Function Block: Convert vectors into magnitude

```
%This function calculate the magnitude of the input-3dimentional-vector input
function Processed_data = fcn(Raw_data)

        Processed_data = sqrt((Raw_data(1))^2+(Raw_data(2))^2+(Raw_data(3))^2);
```

This function calculates the magnitude of the accelerometer data, since the data represents 3 values from the x, y, z axis.

2.5.3 Function Block: Filtering Coefficient

```
%This function creates the coefficient for the signal filter.
%It generates a array of all ones, it takes each 200 samples and calculate
%the average.
function coefficient = fcn()
coefficient = (1/200)*ones(1,200);

```

This function creates the coefficient for the signal filter. It generates an array of all ones. The idea is to take each 200 samples and calculate the average.

2.5.4 Function Block: Calculate Target Rate

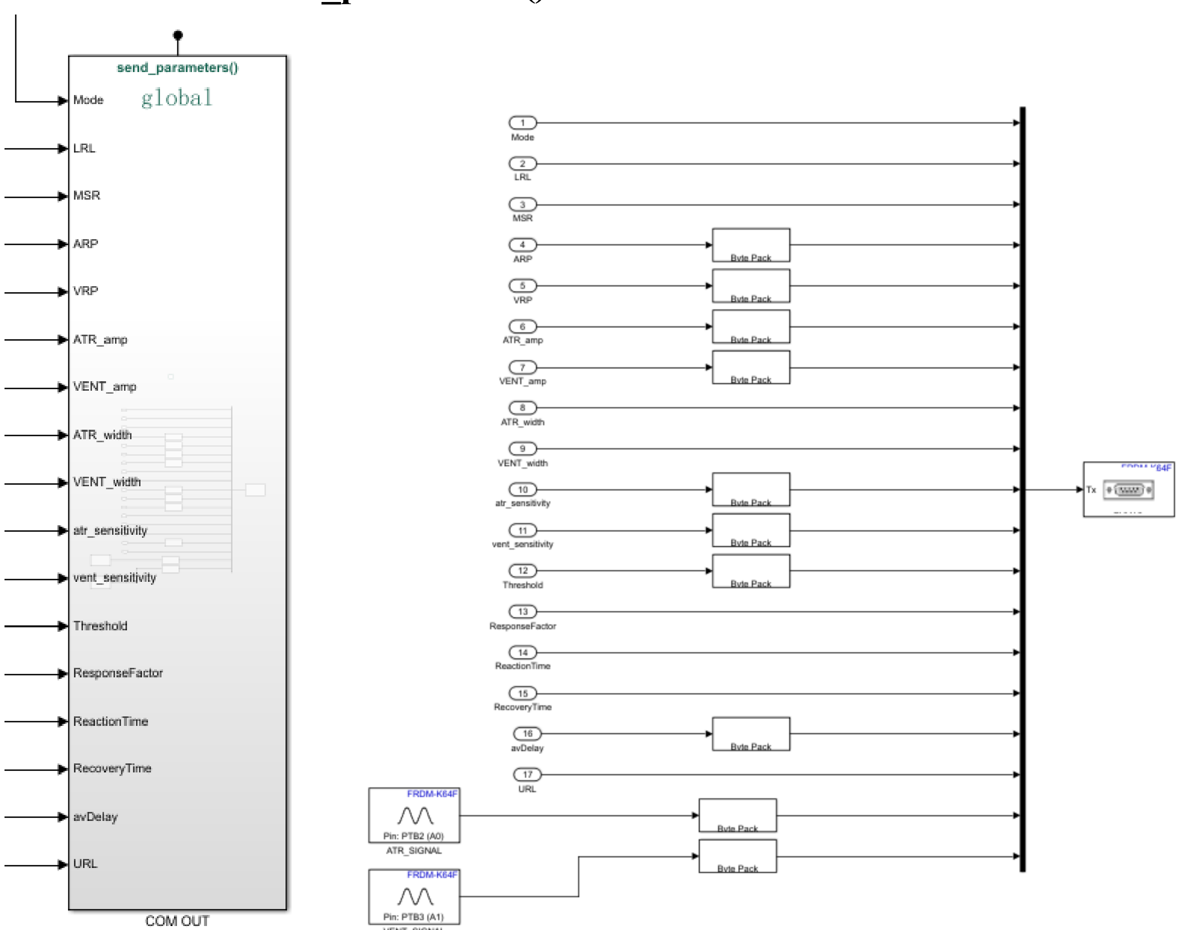
```
%This function calculates the target pacing rate corresponding to the activity
%level.
function Target_rate = fcn(LRL, Response_factor, MSR, Activity_level, Threshold)

      if(Activity_level > Threshold)

         x1 = LRL + Activity_level*Response_factor;
         if(x1 > MSR)
             Target_rate = MSR;
         else
             Target_rate = x1;
         end
      else
          Target_rate = LRL;
      end
```

The Activity_level is the output value from the FIR filter. The function decides the target rate by adding the multiplication result of the activity level and the response factor to the base LRL. When the calculated value exceeds MSR, then it is clipped to the MSR value.
The discussed calculation will only execute if the activity level exceeds the input activity level, otherwise, the target rate will equal to the LRL.

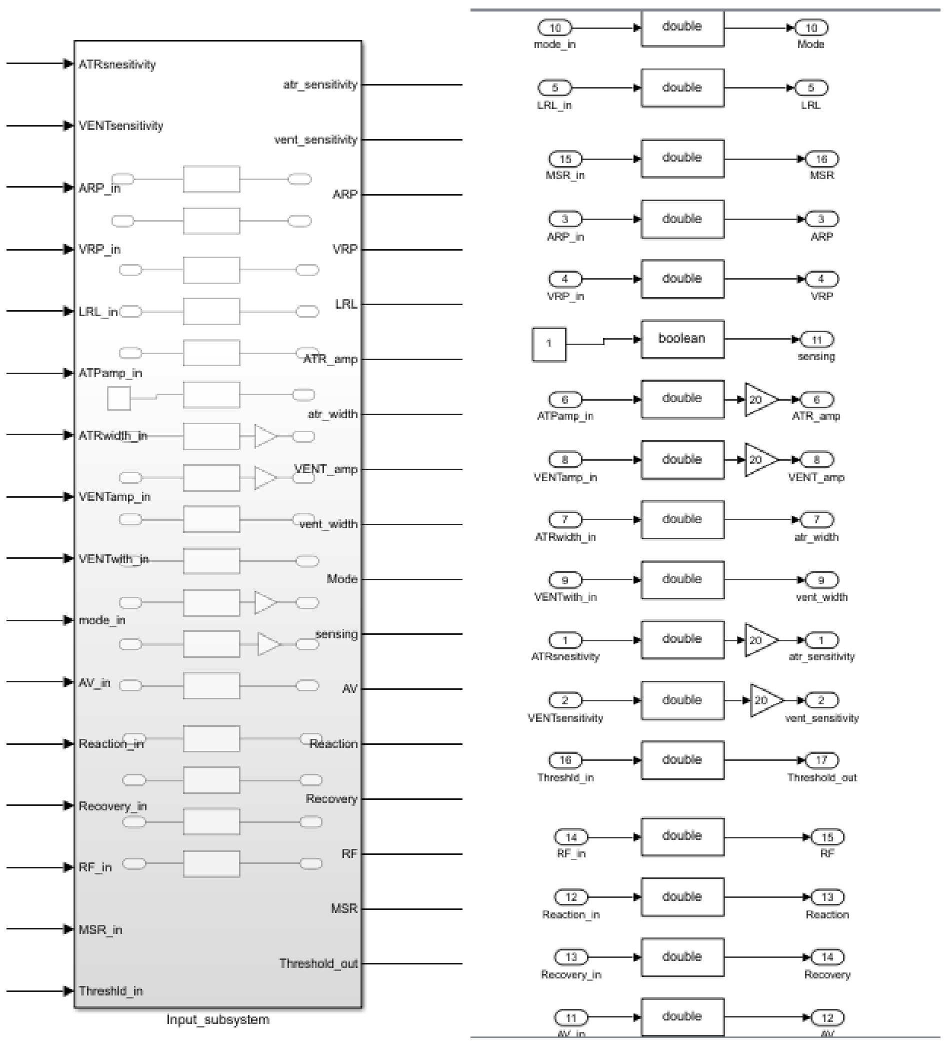
2.5.5 Function: send_parameters()



When this function is called in the Input state chart, it will send all the parameters back to the DCM.

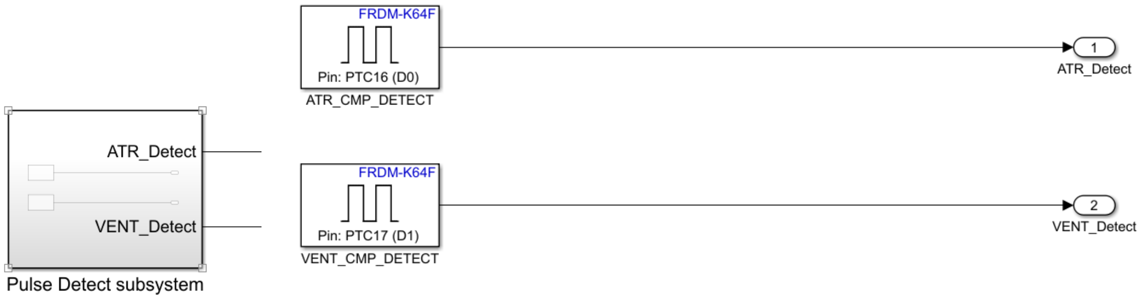
2.6 Hardware Hiding

2.6.1 Input_subsystem



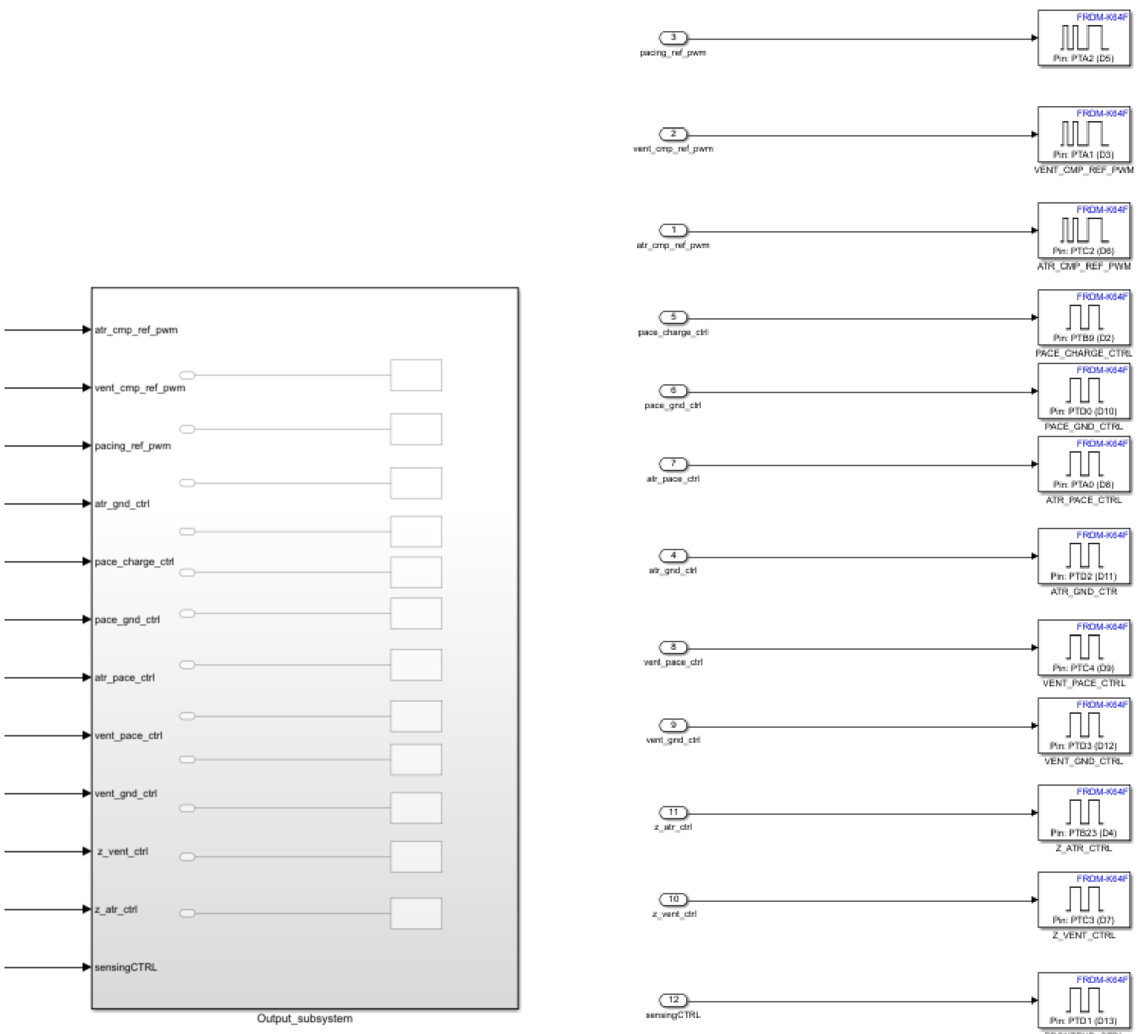
This subsystem isolates the input chart from the implementation. In this intermediate subsystem, the input parameters can be adjusted according to the actual implementation needs. In this pacemaker implantation, all the calculations are done in type Double, and the input voltage is converted to duty cycle here.

2.6.2 Pulse Detect subsystem



This subsystem contains the input pins for ATR_CMP_DETECT and VENT_CMP_DETECT.

2.6.3 Output_subsystem



This output subsystem groups all the output pins.

2.7 Testing

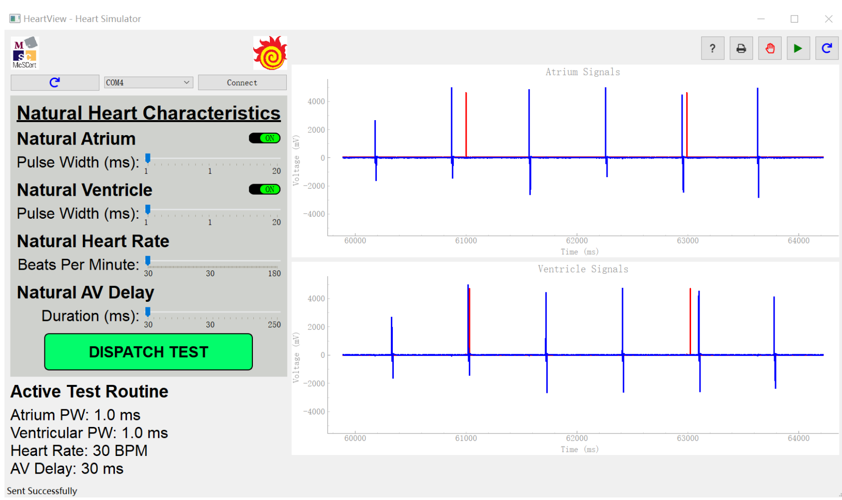
Test Case	Modes select	Input Parameters	Expected output	Actual Output	Pass / Fail
1. Shake the pacemaker from still condition	DOOR, AOOR, VOOR,	LRL = 60; MSR = 160 ARP = 250 VRP = 250 ATR_Amp = 3.3 VENT_Amp = 3.3 ATR_Width = 10 VENT_Width = 10 ATR_Sensitivity = 3 VENT_Sensitivity = 3 Threshold = 1.1 RF = 16 Reaction_time = 10 Recovery_time = 30 AV_delay = 150 URL = 160	Pacing rate increase until it reaches MSR	Pacing rate increase until it reaches MSR	Pass
2. Rest the pacemaker after shaking	DOOR, AOOR, VOOR	LRL = 60; MSR = 160 ARP = 250 VRP = 250 ATR_Amp = 3.3 VENT_Amp = 3.3 ATR_Width = 10 VENT_Width = 10 ATR_Sensitiv	Pacing rate decrease until it reaches LRL	Pacing rate decrease until it reaches LRL	Pass

		ity = 3 VENT_Sensitivity = 3 Threshold = 1.1 RF = 16 Reaction_time = 10 Recovery_time = 30 AV_delay = 150 URL = 160			
3. Dual chamber pacing modes react correctly to the AV delay	DOO/DOOR	LRL = 30; MSR = 175 ARP = 250 VRP = 250 ATR_Amp = 3.3 VENT_Amp = 3.3 ATR_Width = 10 VENT_Width = 10 ATR_Sensitivity = 3 VENT_Sensitivity = 3 Threshold = 1.1 RF = 16 Reaction_time = 10 Recovery_time = 30 AV_delay = 200 URL = 175	There is correct AV delay interval between atrial and ventricular pacing	There is correct AV delay interval between atrial and ventricular pacing	Pass
4. Shake the pacemaker from still condition	AAIR, VVIR	LRL = 60; MSR = 170 ARP = 240 VRP = 250 ATR_Amp =	Pacing rate increases while still maintain the required	Pacing rate increases while still maintain the required	Pass

		3.3 VENT_Amp = 3.3 ATR_Width = 10 VENT_Width = 10 ATR_Sensitivity = 3 VENT_Sensitivity = 3 Threshold = 1.1 RF = 12 Reaction_time = 15 Recovery_time = 40 AV_delay = 150 URL = 170	inhibition time interval after natural beat is sensed	inhibition time interval after natural beat is sensed	
5. Rest the pacemaker after shaking	AAIR, VVIR	LRL = 60; MSR = 165 ARP = 250 VRP = 250 ATR_Amp = 3.3 VENT_Amp = 3.3 ATR_Width = 10 VENT_Width = 10 ATR_Sensitivity = 3 VENT_Sensitivity = 3 Threshold = 1.1 RF = 16 Reaction_time = 16 Recovery_time = 60 AV_delay =	Pacing rate decreases while still maintain the required inhibition time interval after natural beat is sensed	Pacing rate decreases while still maintain the required inhibition time interval after natural beat is sensed	Pass

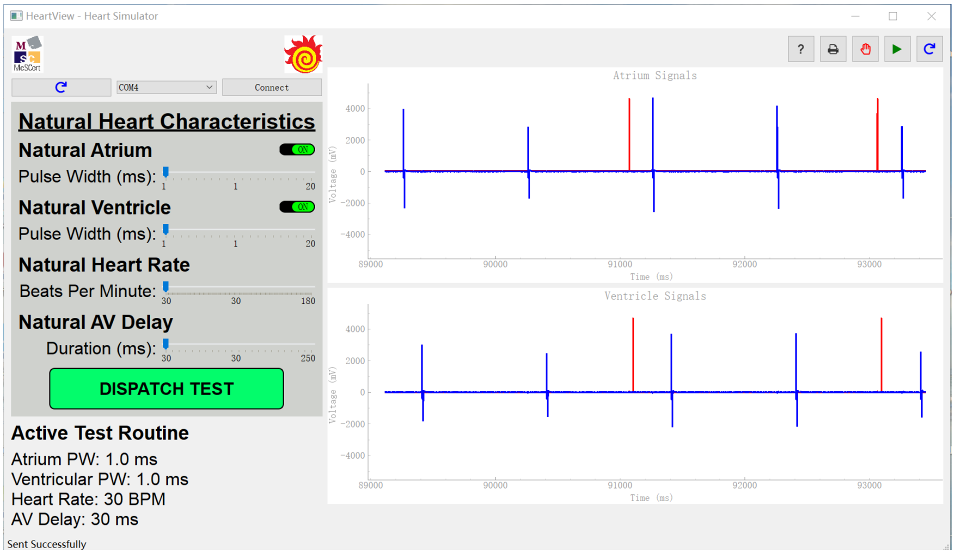
		150 URL = 165			

2.7.1 Test case 1



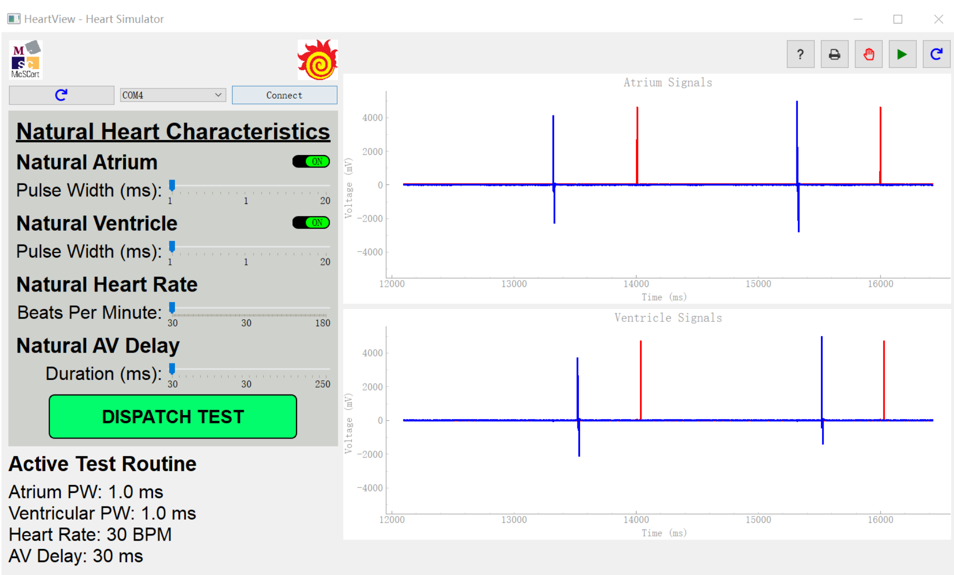
The example pictures were taken under DOOR mode. The first picture shows that the pacemaker is resting on the table, the second one shows that when the pacemaker is being shakened. It is obvious that the pacing rate is getting faster when the pacemaker is on the move.

2.7.2 Test case 2



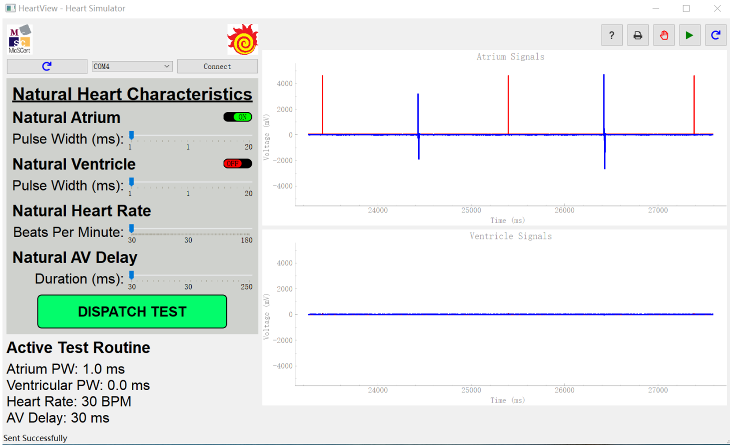
The example pictures were taken under DOOR mode. Following test case 1, when the pacemaker is resting on the table after going through activity, it is obvious that the pacing rate is slower.

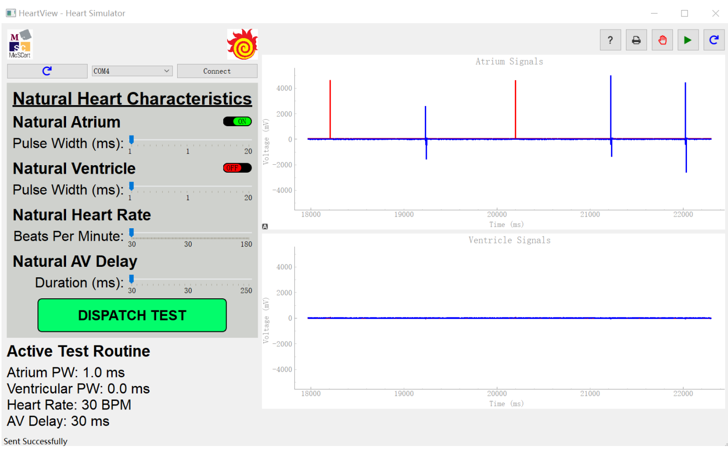
2.7.3 Test case 3

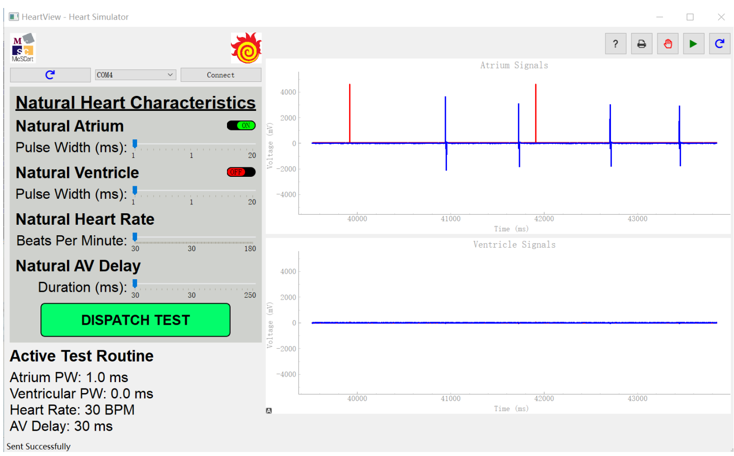


The example picture was taken under DOO mode. The AV delay is set to be 200 msec, it is obvious that the order of pacing and time intervals between each pacing are correct, according to the input settings.

2.7.4 Test case 4

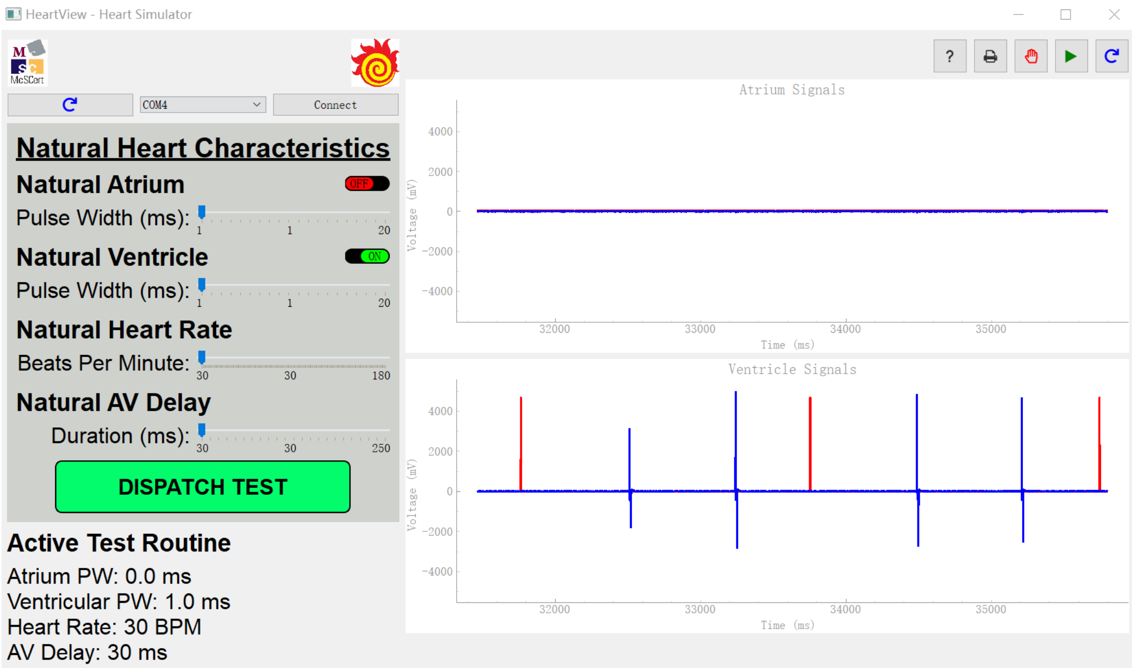


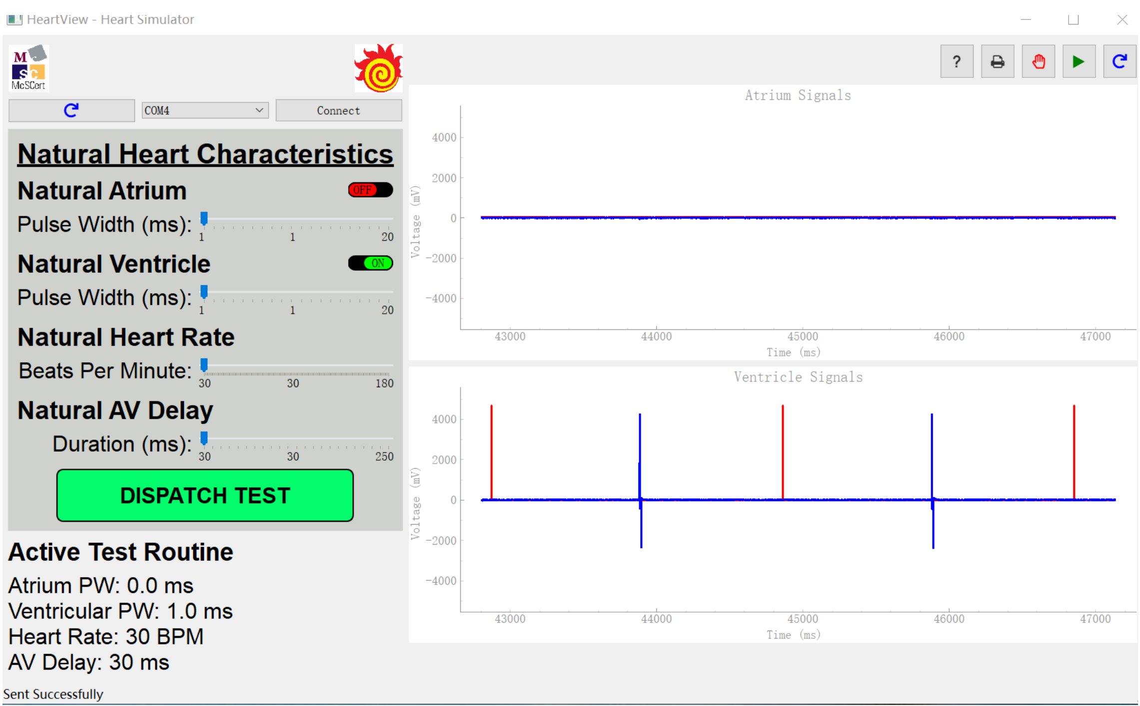




The example pictures were taken under AAIR mode. From the first two pictures, it can tell that when the pacemaker is resting on the table, it shows the same habitat as the normal sensing mode. The third picture shows that when the pacemaker is being shakened, the pacing rate increases while still maintaining the required inhibition time interval after natural beat is sensed.

2.7.5 Test case 5





The example pictures were taken under VVIR mode. The first picture is when the pacemaker is being shakened, the second one is when the pacemaker comes to a rest. It is obvious that the pacing rate is slower, and at the same time, it still maintains the required inhibition time interval after natural beat is sensed.

2.8 Pin Map

Input			
Name	Pin_Name	Connection	Functionality
ATR_CMP_DETECT	PTC16(D0)	ATR_Detect	Used To check if the signal voltage is higher than threshold voltage which will make it output ON to start the atrium sensing.
VENT_CMP_DETECT	PTC17(D1)	VENT_Detect	Same functionality as in ATR_CMP_DETECT but for the ventricle sensing signal
ATR_SIGNAL	PTB2 (A0)	FRDM-K64F Serial Transmit Block	Send the atrial signal to the DCM
VENT_SIGNAL	PTB3 (A1)	FRDM-K64F Serial Transmit Block	Send the ventricle signal to the DCM
Output			
Name	Pin_Name	Connection	Functionality
PACE_CHARGE_CTRL	PTB9(D2)	pace_charge_ctrl	To charge C22 when it is ON, and discharge C22 when it is off
VENT_CMP_REF_PWM	PTA1(D3)	vent_cmp_ref_pwm	To set a threshold voltage when the ventricle is going to be sensed

Z_ATR_CTRL	PTB23(D4)	z_atr_ctrl	Allows the impedance circuit to be connected to the ring electrode of the atrium
PACING_REF_PWM	PTA2(D5)	pacing_ref_pwm	Used to charge C22
ATR_CMP_REF_PWM	PTC2(D6)	atr_cmp_ref_pwm	Same functionality as in VENT CMP REF PWM but for the atrial action potential.
Z_VENT_CTRL	PTC3(D7)	z_vent_ctrl	Used identically to Z ATR CTRL but for the ventricle.
ATR_PACE_CTRL	PTA0(D8)	atr_pace_ctrl	Discharge C22 through the atrium.
VENT_PACE_CTRL	PTC4(D9)	vent_pace_ctrl	Same functionality as in ATR PACE CTRL but for the ventricle.
PACE_GND_CTRL	PTD0(D10)	pace_gnd_ctrl	Used to connect controls the switch directly following the tip
ATR_GND_CTRL	PTD2(D11)	atr_gnd_ctrl	Used to connect the ATR RING OUT to GND.
VENT_GND_CTRL	PTD3(D12)	vent_gnd_ctrl	Same functionality as in ATR_GND_CTRL but for the ventricle.

FRONTEND_CTRL	PTD1(D13)	sensingCTRL	Used to activate the sensing circuitry. The circuit outputs high when it is true.

2.9 Requirements Changes That are Likely

1. Implement more modes.
2. More threshold values could be assigned if there are more modes

3.0 Design Decisions That Are Likely To Change

1. If more threshold values are assigned, the MatLab function group may need to be modified, since different calculating algorithms will be required.
2. More states will be added to the main pacemaker chart, if more modes are implemented.