**SE 3K04 Assignment 2**

**Part 1: Pacemaker Design**

**Group 9**

Kevin Cheung

Forrest Herman

Atiqul Islam

Jingming Liu

**Sunday, November 29, 2020**

Table of Contents

[**Description**](#_heading=h.i4ll7di92q16) **4**

[**Parameters**](#_heading=h.30j0zll) **4**

[Monitored variables (monitored by Heartview)](#_heading=h.1fob9te) 4

[Programmable Parameters for Pacemaker](#_heading=h.3znysh7) 4

[Programmable Quantities Regarding Chart (PACING)](#_heading=h.2et92p0) 5

[Parameters Regrading Chart (COM\_IN)](#_heading=h.tyjcwt) 6

[Parameters Regarding Function (SEND\_Parameter)](#_heading=h.3dy6vkm) 7

[Parameters regrading Chart (Fn\_levelCheck)](#_heading=h.1t3h5sf) 8

[**Parameters regarding Chart (rateAdaptiveControl\_Chart)**](#_heading=h.4a9lnqa5gh0c) **8**

[Initialization values](#_heading=h.17dp8vu) 9

[Limitations to parameters](#_heading=h.k130r9dub914) 10

[**Simulink Model**](#_heading=h.d3l28v4h98kj) **11**

[States(Pacing)](#_heading=h.35nkun2) 11

[**State Diagram (Pacing)**](#_heading=h.v77isy6kkos4) **12**

[Description of Diagram](#_heading=h.44sinio) 12

[State Diagram (DOO)](#_heading=h.2jxsxqh) 14

[Description of Diagram (DOO)](#_heading=h.z337ya) 14

[States(DOO)](#_heading=h.3j2qqm3) 14

[State Diagram (COM\_IN)](#_heading=h.ptflf5yme9v1) 15

[Description of Diagram (COM\_IN)](#_heading=h.4i7ojhp) 15

[States(COM\_IN)](#_heading=h.2xcytpi) 16

[rateAdaptiveContorl\_Chart](#_heading=h.dlqgsbvfnr00) 16

[Description of Diagram (rateAdaptiveControl\_Chart)](#_heading=h.58pfn2ukr7wy) 16

[States (rateAdaptiveContorl\_Chart)](#_heading=h.3whwml4) 17

[Function (Fn\_levelCheck)](#_heading=h.r5j7udpbbpel) 18

[Function (send\_parameters)](#_heading=h.qsh70q) 18

[Transition Table (COM\_IN)](#_heading=h.3as4poj) 19

[Transition Table(rateAdaptiveControl\_Chart)](#_heading=h.t3r5qg7o6uyi) 20

[Transition Table(Pacing)](#_heading=h.49x2ik5) 20

[Data Smoothing](#_heading=h.rwm0pvjjv3y8) 22

[Modularity](#_heading=h.vb6i6gla0oa1) 22

[**Bonus**](#_heading=h.mc4q8he8klq) **23**

[Inhibition of Ventricular Pace](#_heading=h.5k5t9quadgkq) 23

[DDDR Mode](#_heading=h.6nbq2jkrsphd) 24

[Description of Diagram](#_heading=h.3adgmz4x8xbz) 24

[Design Decisions](#_heading=h.2p2csry) **26**

[**Testing (Assignment 2)**](#_heading=h.xjhguqmge1op) **28**

[Screenshots of Corresponding Test Cases](#_heading=h.23ckvvd) **28**

[Requirement Changes That Are Likely](#_heading=h.jdj3kfd01e89) **31**

[Design Decision Changes That Are Likely](#_heading=h.32hioqz) **31**

## 

## 

## Description

The stateflow, built using Simulink (a matlab-based graphical programming environment), creates a model of a reactive system in the form of a pacemaker. Several states were created in a finite state machine variant, using the control logic functionalities that are available in Simulink.

## Parameters

#### **Monitored variables (monitored by Heartview)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Input to Device** | **Output from Device** | **Min** | **Max** | **Unit** | **Description** |
| Atrial pace signal voltage | True |  | 0 | 3.3 | Volt | Measured natural heart atrial amplitude |
| Ventricle pace signal voltage | True |  | 0 | 3.3 | Volt | Measured natural heart Ventricle amplitude |
| Atrial pacing amplitude |  | True | 0 | 5 | Volt | Measured paced heart atrial amplitude |
| Ventricle pace  amplitude |  | True | 0 | 5 | Volt | Measured paced heart Ventricle amplitude |
| Natural heart beat per minute | True |  | 0 | 180 | Bpm | Measured beat per minute from the natural heart |
| Pacing beat per minute |  | True | 0 | 180 | Bpm | Measured beat per minute from pacemaker |
| Atrial natural pulseWidth | True |  | 0 | 20 | ms | Measured pulse width from the natural heart |

#### **Programmable Parameters for Pacemaker**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Min** | **Max** | **Unit** | **Description** |
| Lower Rate Limit | 40 | 175 | bpm | Lower rate limit, minimum beats per minute |
| Upper Rate Limit | 60 | 175 | bpm | Upper rate limit bpm |
| MSR | 50 | 175 | bpm | Maximum rate of pacing can set to for Rate adaptive mode |
| ARP | 0 | 600 | msec | Atrial Refractory Period, time in msec after an atrial pace where atrium sensing is disabled |
| VRP | 0 | 600 | msec | Ventricular Refractory Period, time in msec after a ventricular pace where ventricular sensing is disabled |
| Atrial Amplitude | 0 | 5 | volt | The desired amplitude of atrial pacing. Converted to a duty cycle percentage for Pins D3,D5,D6 inputs. |
| Atrial Pulse Width | 0 | 10 | msec | Pulse width of atrial pacing |
| Ventricular Amplitude | 0 | 5 | volt | The desired amplitude of ventricular pacing. Converted to a duty cycle percentage for Pins D3,D5,D6 inputs. |
| Ventricular Pulse Width | 0 | 10 | msec | Pulse width of ventricular pacing |
| Fixed AV Delay | 70 | 300 | msec | Fixed value delay for DOO and DOOR mode. The delay between atrial and ventricular pace in both modes. |
| Activity Threshold | Low | High |  | Three levels including low, med and high for comparing the activity data from the accelerometer. |
| Response Factor | 1 | 16 |  | The step size factor for increment and decrement of pacing rate, when rate adaptive modes are selected |
| Recovery Time | 2 | 16 | min | The total time it takes for rate decrease from MSR to LRL in rate adaptive mode. |
| Reaction Time | 10 | 50 | sec | The total time it takes for rate increase from LRL to MSR in rate adaptive mode. |

#### **Programmable Quantities Regarding Chart (PACING)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Input to Stateflow** | **Output from Stateflow** | **Min** | **Max** | **Initial Value** | **Unit** | **DataType** | **Description** |
| ATRpulseWidth | Yes |  | 0 | 10 | 10 | MSEC | Single | Pace width of atrial pacing |
| VENTpulseWidth | Yes |  | 0 | 10 | 10 | MSEC | Single | Pace width of ventricular pacing |
| sensing | True |  | 0 | 1 | 1 |  | boolean | control the value of sensingCTRL |
| PACE\_CHARGE\_CTRL |  | True | 0 | 1 | Control by states |  | boolean | Output to pin D2 in FRDM-L64F to start and stop charging of C22 capacitor |
| PACE\_GND\_CTRL |  | True | 0 | 1 | Control by states |  | boolean | Output to pin D10 in FRDM-L64F for current flow in atrium and ventricle |
| ATR\_PACE\_CTRL |  | True | 0 | 1 | Control by states |  | boolean | Output to pin D8 in FRDM-L64F for current flow to atrium when pacing |
| ATR\_GND\_CTRL |  | True | 0 | 1 | Control by states |  | boolean | Output to pin D11 in FRDM-L64F for discharging capacitor c21 when pacing atrium |
| VENT\_GND\_CTRL |  | True | 0 | 1 | Control by states |  | boolean | Output to pin D12 in FRDM-L64F for discharging capacitor c21 when pacing ventricle |
| Z\_ATR\_CTRL |  | True | 0 | 1 | Control by states |  | boolean | Output to pin D4 in FRDM-L64F for analyzing atrial electrode impedance |
| Z\_VENT\_CTRL |  | True | 0 | 1 | Control by states |  | boolean | Output to pin D7 in FRDM-L64F for analyzing ventricular electrode impedance |
| sensing\_CTRL |  | True | 0 | 1 | Control by states |  | boolean | Output to pin D13 in FRDM-L64F for Frontend\_CTRL specified for activate sensing circuitry |
| msecPerPace |  |  | 0 | 1500 | 60000/PPM\_RATE | MSEC | double | Convert msecPerPace to a sec unit |
| atr\_detect | True |  | 0 | 1 | Control by states |  | boolean | Connected to pin D0 in FRDM-L64F for ATR\_CMP\_DETECT |
| vent\_detect | True |  | 0 | 1 | Control by states |  | boolean | Connected to pin D1 in FRDM-L64F for VENT\_CMP\_DETECT |
| paceLocation | True |  | 0 | 1 | Control by states |  | Uint8 | Indicate mode for atrium or ventricle. 0 atrium, 1 ventricle, 2 for Dual mode |
| PPM\_RATE | True |  | 40 | 175 | Control by states | PPM | single | Pacing rate per minute that can vary as needed in R modes. |
| AV\_delay | True |  | 70 | 300 | 50 | PPM | single | Fixed AV\_ delay for D pacing mode |
| ARP | True |  | 0 | 600 | 250 | MSEC | single | Atrial Refractory Period, time in msec after an atrial pace where atrium sensing is disabled |
| VRP | True |  | 0 | 600 | 250 | MSEC | single | Ventricular Refractory Period, time in msec after a ventricular pace where ventricular sensing is disabled |

#### **Parameters Regarding Chart (COM\_IN)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Input to Stateflow** | **Output from Stateflow** | **Min** | **Max** | **Initial Value** | **Unit** | **DataType** | **Description** |
| VentSensitivity |  | TRUE |  |  | 50 | mV | single | how large a pulse needs to be in order to be sensed (in sensing modes) to inhibit a ventricular pacing |
| AtrSensitivity |  | TRUE |  |  | 50 | mV | Single | how large a pulse needs to be in order to be sensed (in sensing modes) to inhibit atrial pacing |
| ATRpulseWidth |  | True | 0 | 10 | 10 | msec | Single | Pace width of atrial pacing |
| VENTpulseWidth |  | True | 0 | 10 | 10 | msec | Single | Pace width of ventricular pacing |
| ATRpulseAmp |  | True | 0 | 5 | 50 | V | Single | The desired amplitude of atrial pacing. Converted to a duty cycle percentage for Pins D3,D5,D6 inputs. |
| VENTpulseAmp |  | True | 0 | 5 | 50 | V | Single | The desired amplitude of ventricular pacing. Converted to a duty cycle percentage for Pins D3,D5,D6 inputs. |
| sensingTrue |  | True | 0 | 1 | 1 |  | boolean | control the value of sensingCTRL |
| paceLocation |  | True | 0 | 1 | Control by states |  | Uint8 | Indicate mode for atrium or ventricle. 0 atrium, 1 ventricle, 2 for Dual mode |
| LRL |  | True | 40 | 175 | 60 | PPM | Uint8 | Lower rate limit, minimum beats per minute |
| AV\_delay |  | True | 70 | 300 |  | msec | Uint16 | Fixed AV\_ delay for D pacing mode |
| URL |  | True | 60 | 150 | Not set | PPM | Uint8 | Upper rate limit |
| ARP |  | True | 0 | 600 | 250 | msec | Single | Atrial Refractory Period, time in msec after an atrial pace where atrium sensing is disabled |
| VRP |  | True | 0 | 600 | 250 | msec | Single | Ventricular Refractory Period, time in msec after a ventricular pace where ventricular sensing is disabled |
| RX\_data | True |  | N/A | N/A | N/A | N/A | Uint8 | Serial transmission data through UART port from FRDM-K64F |
| Status | True |  | 0 | 1 | N/A | N/A | Uint8 | Status regarding UART port from FRDM-K64F |
| rateAdaptiveTrue |  | True |  | 0 | 1 | N/A | Uint8 | When equal to 1 switch pacing modes with R mode, When equal to 0, switch to pacing modes without R mode. |
| responseFactor |  | True |  | 1 | 16 | N/A | Single | The step size factor for increment and decrement of pacing rate, when rate adaptive modes are selected |
| reactionTime |  | True |  | 10 | 50 | sec | Single | The total time needed for rate to increase to MSR in rate adaptive modes. |
| recoveryTime |  | True |  | 2 | 16 | min | Single | The total time needed for rate to decrease to LRL in rate adaptive modes. |
| acc\_threshold\_LOW |  | True | -4 | 4 | N/A | g | single | Send the real time user input for low threshold to the board in rate adaptive modes. |
| acc\_threshold\_MED |  | True | -4 | 4 | N/A | g | single | Send the real time user input for the medium threshold to the board in rate adaptive modes. |
| acc\_threshold\_HIGH |  | True | -4 | 4 | N/A | g | single | Send the real time user input for the high threshold to the board in rate adaptive modes. |

#### **Parameters Regarding Function (SEND\_Parameter)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Input** | **Output** | **Min** | **Max** | **Initial Value** | **Unit** | **DataType** | **Description** |
| ATR\_SIGNAL | True |  | 0 | 5 | Control by pacing stateflow | V | double | Send the data of real time Atrium pacing to the DCM |
| VENT\_SIGNAL | True |  | 0 | 5 | Control by pacing stateflow | V | double | Send the data of real time Ventricular pacing to the DCM |
| SERIAL Number | True |  | N/A | N/A | 92020 | N/A | double | Send the data of the pacemaker serial number to DCM |
| Accelerometer | True |  | N/A | N/A |  |  | double | Send the data of the real time serial number to DCM |

#### **Parameters regarding Chart (Fn\_levelCheck)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Input** | **Output** | **Min** | **Max** | **Initial Value** | **Unit** | **DataType** | **Description** |
| Threshold\_low | True |  | -4 | 4 | N/A | g | single | The output from chart COM\_IN where set the low threshold is set by DCM |
| Threshold\_med | True |  | -4 | 4 | N/A | g | single | The output from chart COM\_IN where set the med threshold is set by DCM |
| Threshold\_high | True |  | -4 | 4 | N/A | g | single | The output from chart COM\_IN where set the high threshold is set by DCM |
| targetLevel |  | True | 0 | 3 | 0 |  | single | Output a target level for reference in  rateAdaptiveControl\_Chart |

#### **Parameters regarding Chart (rateAdaptiveControl\_Chart)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Input to Stateflow** | **Output from Stateflow** | **Min** | **Max** | **Initial Value** | **Unit** | **DataType** | **Description** |
| LRL | True |  | 40 | 175 | 60 | PPM | single | Lower rate limit, minimum beats per minute |
| MSR | True |  | 50 | 175 |  | PPM | single | Maximum rate of pacing can set to for Rate adaptive mode |
| rateAdaptiveTrue | True |  | **0** | **1** | **0** |  | boolean | If R modes are selected, equal to true, vice versa. Used to enable rate adaptive on A,V,D pacing modes |
| RF | True |  | **1** | **16** | **1** |  | single | Factor setting for increment and decrement in rate adaptive modes. |
| reactionTime | True |  | 10 | 50 | 10 | sec | single | The total time needed for rate to increase to MSR in rate adaptive modes. |
| recoveryTime | True |  | 2 | 16 | 2 | min | single | The total time needed for rate to decrease to LRL in rate adaptive modes. |
| targetLevel | True |  | **0** | **3** | **0** |  | single | The targetlevel send from Fn\_levelCheck function used to compare with the previous level |
| previous level |  |  | **0** | **3** | **0** |  | single | compare with the targetlevel as state transition conditions |
| RATE |  | True | 40 | 175 | 60 | PPM | single | The rate output to the Pacing states where control the pacing rate in rate adaptive mode. |
| LED\_red |  | True | 0 | 1 |  |  | boolean | Used to test and indicate states within chart |
| LED\_green |  | True | 0 | 1 |  |  | boolean | Used to test and indicate states within chart |
| LED\_blue |  | True | 0 | 1 |  |  | boolean | Used to test and indicate states within chart |

#### **Initialization values**

|  |  |
| --- | --- |
| Variable Name | Initialized Value |
| ATRpulseWidth | 10 ms |
| VENTpulseWidth | 10 ms |
| ATRpulseAmp | 50 %duty cycle |
| VENTpulseAmp | 50 %duty cycle |
| LRL | 60 beats/min |
| MSR | 120 beats/min |
| msecPerPace | 60,000/LRL ms |
| ARP | 250 ms |
| VRP | 250 ms |
| sensing | 0 (boolean) |
| paceLocation | 0 (boolean) |
| PACING\_REF\_PWM  (Frequency / Initial duty cycle) | 2000 Hz /50 |
| VENT\_CMP\_REF\_PWM  (Frequency / Initial duty cycle) | 2000 Hz / 50 |
| ATR\_CMP\_REF\_PWM  (Frequency / Initial duty cycle) | 2000 Hz/ 50 |
| AtrSensitivity | 50 %duty cycle |
| VentSensitivity | 50 %duty cycle |
| acc\_threshold\_LOW | 1.1 g |
| acc\_threshold\_MED | 1.5g |
| acc\_threshold\_HIGH | 2g |
| responseFactor | 1 |
| reactionTime | 10 sec |
| recoveryTime | 30 min |

#### 

#### **Limitations to parameters**

|  |  |  |
| --- | --- | --- |
| Variable Name | Min/Max Value | Reason |
| pulseWidth | 0msec / 20 msec | Pulse with should not be too long so that the time between pacing and discharge will be too long which create a temporary non zero net current that is harmful to the patient |
| dutyCycle/dutyCycle1  /dutyCycle2 | 0% / 100 % | All duty cycle of PWM is specified from 0% to 100% |
| LRL | 40 bpm / 175 bpm | Minimum heartbeat should be low enough to not overwork the heart, but should be enough to keep a high enough blood pressure. |
| URL | 60 bpm / 150 bpm | Heart should be allowed to beat faster than average during exercise for example. |
| VRP/ARP | 0 msec / 500 msec | 0 if not enabled, 500 seems to be the maximum that would be required. Since the LRL will typically be around 60 bpm , meaning the longest typical time between paces would be around 1000-1500 msec, a 500 msec refractory period seems the longest that could be desirable. |
| MSR | 50~175 ppm | TheMSR rate will have at least equal or higher value than LRL to make sure the reaction time in rateAdaptive mode is appropriate. The rate should not be higher than 175 ppm that may casing pacing rate to fast for the patient |
| AV\_delay | 70~300ms | The delay will be desired to stay in this region because the delay should be too short that causes interference from atrium pacing and ventricular pace. But not too far apart for an ideal dual mode pacing. |
| Reaction Time | 10~50sec | The increment time from LRL to MSR should not be too little to cause a sudden change in pacing rate but not too slow for the rate to keep up with the activity data from the accelerometer. |
| Recovery Time | 2~16miin | The decrement time from pacing rate to LRL should not be too little to cause a sudden change in pacing rate but not too slow for the rate to keep up with the activity data from the accelerometer. |

## 

## 

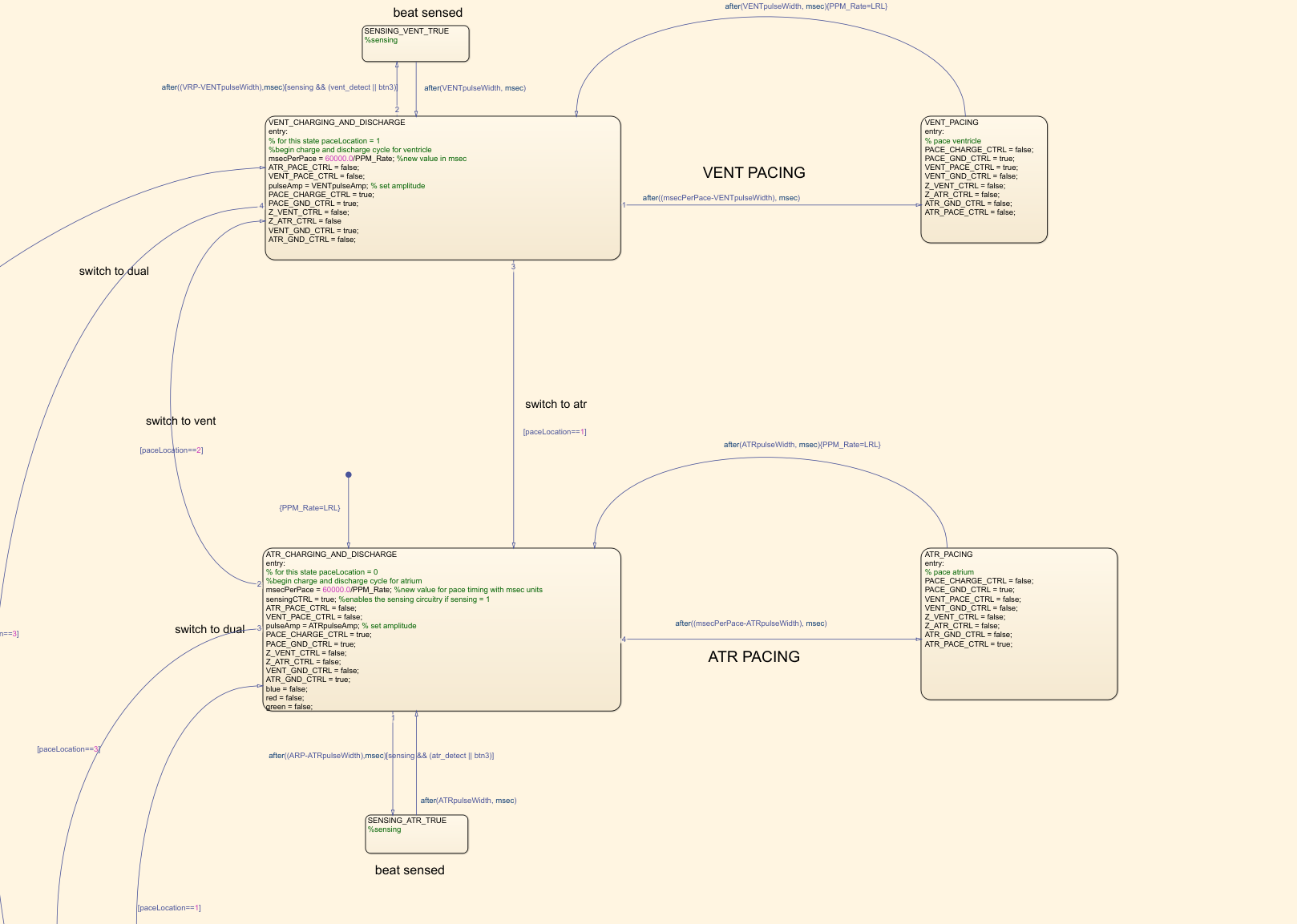
## Simulink Model

### States(Pacing)

|  |  |
| --- | --- |
| State name | State description |
| VENT\_CHARGING\_AND\_DISCHARGE | This state is responsible for prepare ventricular pacing by charging capacitor C22 and then discharging capacitor C21 after the ventricle been paced (charge flow back through heart in order to result a net zero current ) |
| VENT\_PACING | This state is responsible for discharging capacitor C22 and letting the current flow through the ventricle to pace the heart. |
| ATR\_CHARGING\_AND\_DISCHARGE | This state is responsible for prepare ventricular pacing by charging capacitor C22 and then discharging capacitor C21 after the atrium been paced(charge flow back through heart in order to result a net zero current ) |
| ATR\_PACING | This state is responsible for discharging capacitor C22 and letting the current flow through the atrium to pace the heart. |
| SENSING\_VENT\_TRUE | This state is responsible for activate the sensing circuit for VVI pacing mode |
| SENSING\_ATR\_TRUE | This state is responsible for activate the sensing circuit for AAI pacing mode |

### 

### State Diagram (Pacing)



#### **Description of Diagram**

There are a total 6 states in the diagram. The upper part of the diagram shows ventricular pacing and sensing logic. The lower part of the diagram is for atrial pacing and sensing.

For AOO and VOO pacing mode, the parameter called *sensing* will be set to false. Thus, the state only goes through VENT\_CHARGING\_AND\_DISCHARGE, VENT\_PACING for VOO and ATR\_CHARGING\_AND\_DISCHARGE, ATR\_PACING for AOO .

**AOO Pacing Sequence:**

For AOO, starting with the charging capacitor C22, PACE\_CHARGE\_CTRL is true for charging, PACE\_GND\_CTRL is also true but does not take effect until it goes to the next stage for pacing. ATR\_GND\_CTRL is also true for preparing discharging capacitor C21 after the pacing stage. After a period of time from parameter *msecPerPace* (60000/PPM\_Rate) the stage goes to ATR\_PACING. ATR\_GND\_CTRL is false and ATR\_PACING\_CTRL is set to true to let the C22 discharge the current to the heart . While the PACE\_CHARGE\_CTRL is set to false to prevent connecting voltage directly to the patient's heart. After a period of time in msec specified by ATR*pulseWidth* parameter, the stage will go back to the ATR\_CHARGING\_AND\_DISCHARGE stage. ATR\_PACE\_CTRL will be set to false and ATR\_GND\_CTRL is true for discharging the capacitor C21 to create a net zero current in the heart.

For both states, Z\_VENT\_GND\_CTRL, Z\_ATR\_GND\_CTRL, VENT\_PACE\_CTRL, and VENT\_GND\_CTRL are set to false.

**VOO Pacing Sequence:**

For VOO, starting with the charging capacitor C22, PACE\_CHARGE\_CTRL is true for charging, PACE\_GND\_CTRL is also true but does not take effect until it goes to the next stage for pacing. VENT\_GND\_CTRL is also true for preparing discharging capacitor C21 after the pacing stage. After a period of time from parameter *msecPerPace* (60000/ PPM\_Rate), the stage goes to VENT\_PAING. VENT\_GND\_CTRL is false and VENT\_Paing\_CTRL is set to true to let the C22 discharge the current to the heart . While the PACE\_CHARGE\_CTRL is set to false to prevent connecting voltage directly to the patient's heart. After a period of time in msec specified by VENT*pulseWidth* parameter. The stage will go back to VENT\_CHARGING\_AND\_DISCHARGE stage.VENT\_PACE\_CTRL will be set to false and VENT\_GND\_CTRL is true for discharging the capacitor C21 to create a net zero current in the heart.

For both states, Z\_VENT\_GND\_CTRL, Z\_ATR\_GND\_CTRL, ATR\_PACE\_CTRL, and ATR\_GND\_CTRL are set to false.

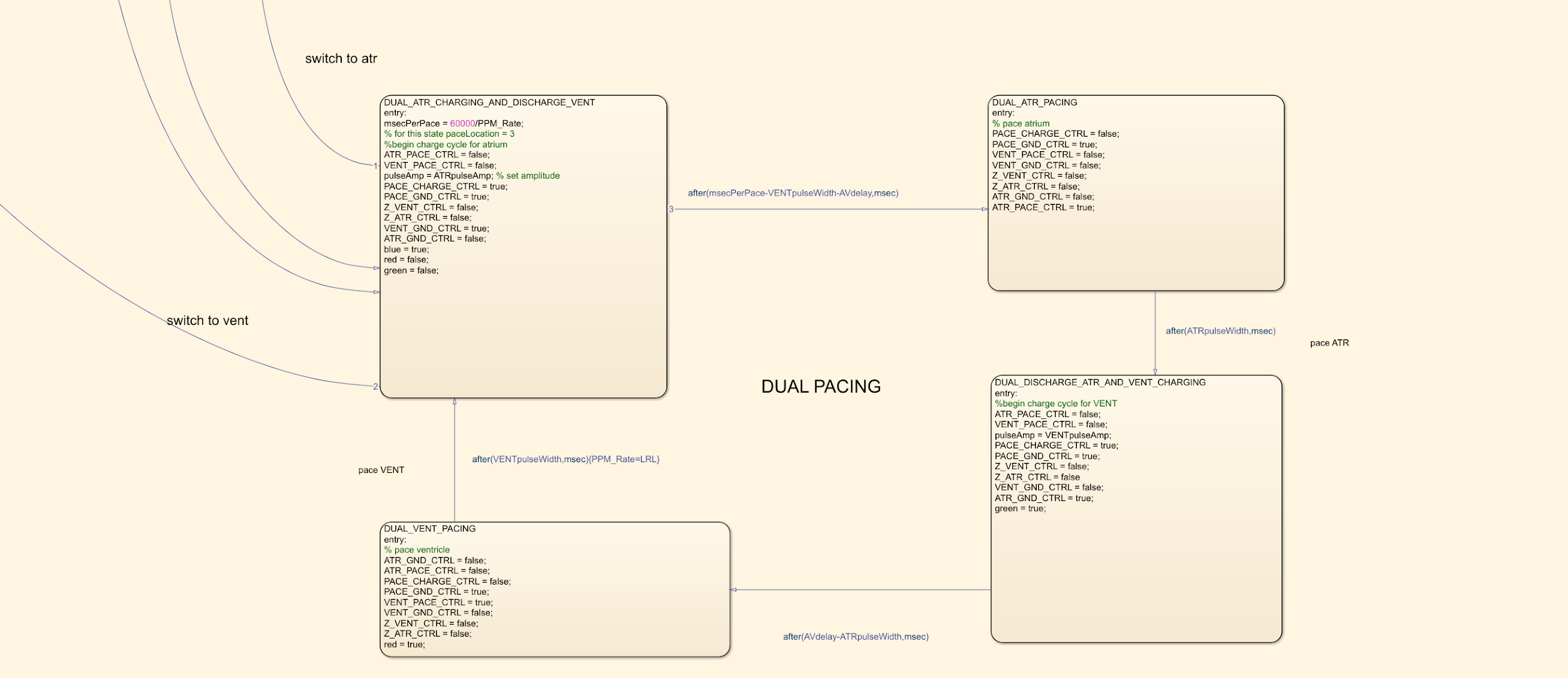
**AAI and VVI Pacing and Sensing Sequences:**

For AAI and VVI pacing mode, input parameter *sensing* will be set to true by the DCM and thus *sensingCTRL* will also be true (the output to pin D13 FRONTEND\_CTRL). As such the program is switched from either AOO or VOO to instead AAI or VVI pacing mode. Then the program will perform sensing logic while in the CHARGING\_AND\_DISCHARGE state. After ARP or VRP minus *pulseWidth(ATR,VENT)* amount of time (this accounts for the *pulseWidth* amount of time that already elapsed since the last pace while moving from the PACING state to the CHARGING\_AND\_DISCHARGE state), the diagram checks whether atr\_detect or vent\_detect (which is connected from ATR\_CMP\_DETECT or VENT\_CMP\_DETECT) has detected a pulse. It will then go to SENSING\_ATR\_TRUE if so. Otherwise it will continue with the next pace as usual once msecPerPace has elapsed.

For AAI, if atr\_detect is high, this means ATR\_CMP\_DETECT from FRDM-L64F is high, which means a natural atrial heartbeat has been detected. In this case, (if ARP has elapsed) control will move from ATR\_CHARGING\_AND\_DISCHARGING state to SENSING\_ATR\_TRUE. It will wait in this state for the duration of pulseWidth before returning to ATR\_CHARGING\_AND\_DISCHARGING to implement the same process for pacing described in AOO pacemode.

For VVI, If vent\_detect is high means VENT\_CMP\_DETECT from FRDM-L64F is high, which means a natural ventricular heartbeat has been detected. In this case, (if VRP has elapsed) control will move from VENT\_CHARGING\_AND\_DISCHARGING state to SENSING\_VENT\_TRUE. It will wait in this state for the duration of *pulseWidth* before returning to VENT\_CHARGING\_AND\_DISCHARGING to implement the same process for pacing described in VOO pacemode.

### State Diagram (DOO)



#### **Description of Diagram (DOO)**

DOO mode is implemented by four states in total. Which includes DUAL ATR CHARGING AND DISCHARGE VENT, DUAL ATR PACING, DUAL\_DISCHARGE ATR AND VENT CHARGING

and DUAL\_VENT\_PACING. First state is responsible to charge the capacitor C22 for atrium pacing . Then after (msecPerPace-VENTpulseWidth-Avdelay) ms , the capacitor is going to pacing the atrium and then going to third state, the third state is going to discharged the atrium pacing from the pacemaker and prepare C22 for next pacing in ventricle. In the fourth state, the pacemaker is going to pace the ventricle after a period of (AVdelay-ATRpulseWidth). Finally, the state will return to the last state where a discharge of ventricle happens.

In this case the time for discharging Ventricle pacing is msecPerPace(LRL)-VENTpulseWidth-AVdelay,

Then the time for discharging Atrium is AVdelay-ATRpulseWidth.

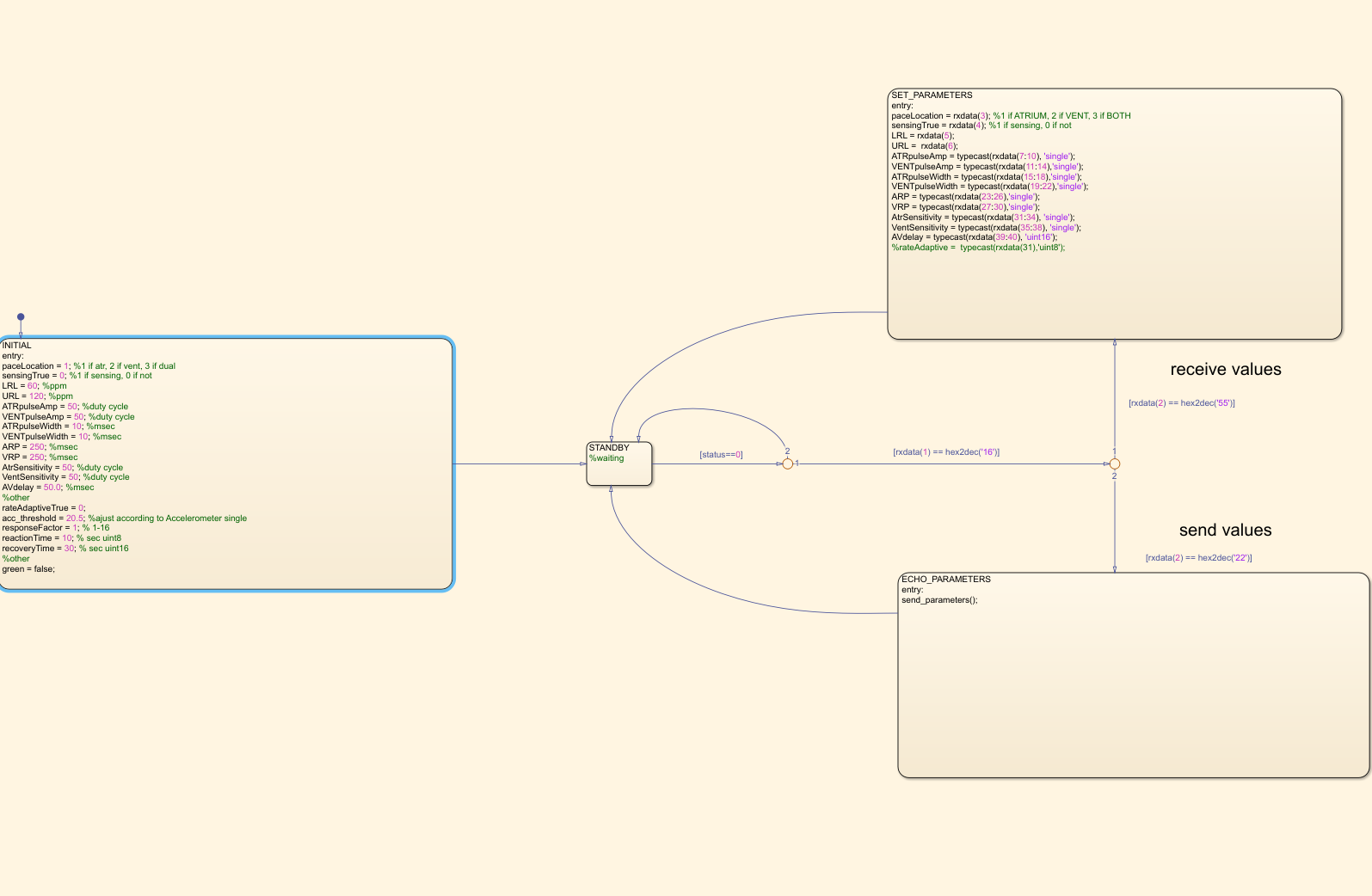
Note, both charging and discharging for Atrium and Ventricle follow the same steps as above for AOO, VOO mode, except the charging and discharging is interchanged with each other as described.

#### 

#### **States(DOO)**

|  |  |
| --- | --- |
| State name | State description |
| DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT | This state is responsible for charging the C22 capacitor for pacing the atrium and discharging C21 capacitor for discharging the ventricle. |
| DUAL\_ATR\_PACING | This state is responsible for pacing the atrium |
| DUAL\_DISCHARGE\_ATR\_AND\_VENT\_CHARGING | This state is responsible for charging the C22 capacitor for pacing the ventricle. and discharging C21 capacitor for discharging the atrium. |
| DUAL\_VENT\_PACING | This state is responsible for pacing the ventricle |

### State Diagram (COM\_IN)



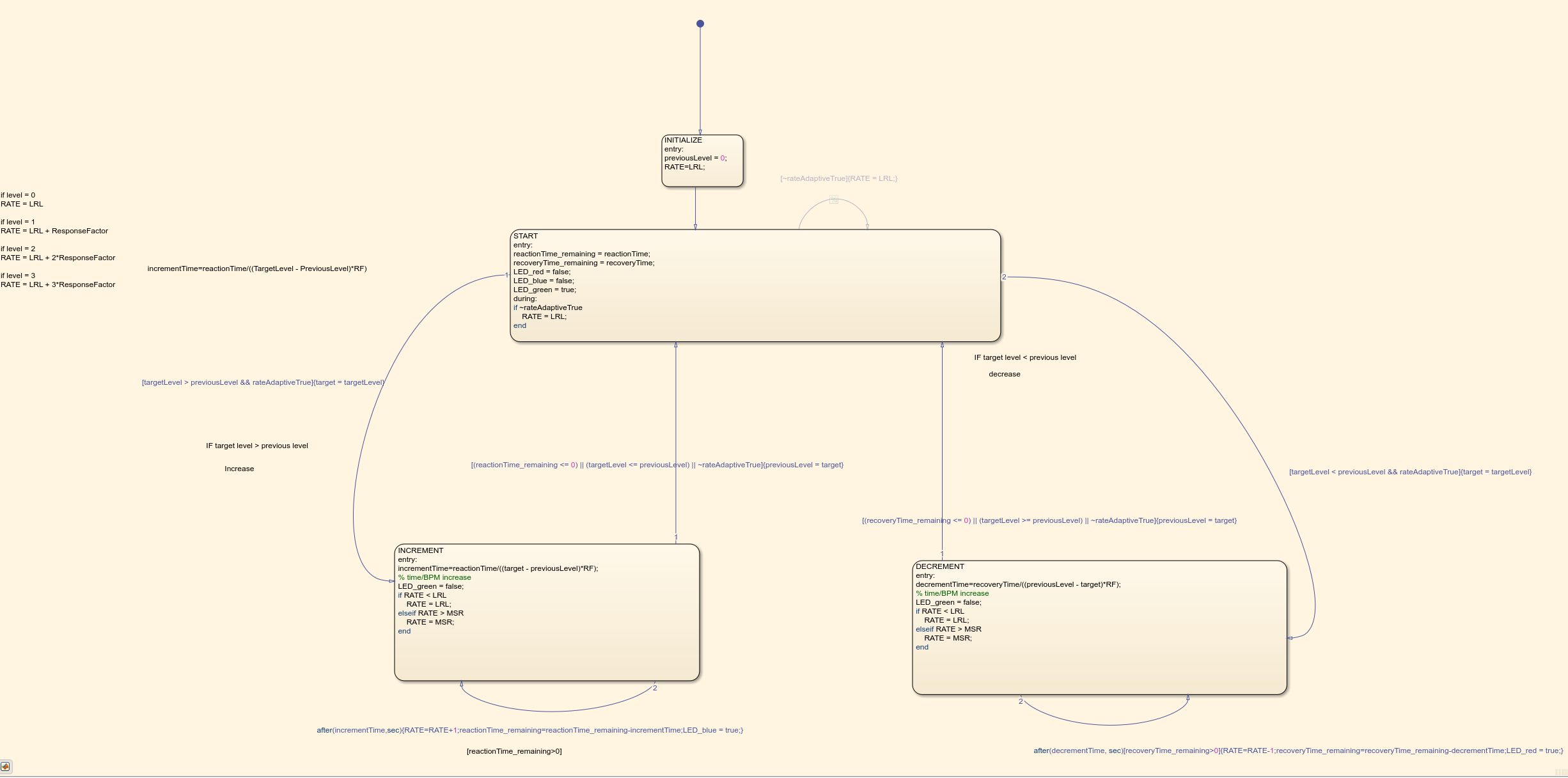
#### **Description of Diagram (COM\_IN)**

There are four states for the serial communication stages, which are INITIAL that gives the initialized value to the parameters; SET\_PARAMETERS that receiving data from the DCM; and ECHO PARAMETERS that calls the send\_parameter() function to send the values from Simulink to DCM. To display useful information. First the state will go through the INITIAL state. After each value is initialized, the state will go to the standby State where it will wait until the status changes to zero. If rxdata(1) received from the UART port is equal to 16, the state will move forward to decide whether to receive values or send values for serial communication. When the rxdata(2) is found to be 55, the state will go to SET\_PARAMETERS to receive data. If rxdata(2) is equal to 22, the state will go to ECHO PARAMETERS to send data to DCM.

#### **States(COM\_IN)**

|  |  |
| --- | --- |
| State name | State description |
| INITIAL | Initialize all programmable parameters for pacemaker |
| SET\_PARAMETERS | Receive the programmable parameter form DCM |
| ECHO\_PARAMETERS | Call sendFunction to send the parameters to DCM to display |
| STANDBY | Wait for the status change to proceed |

### rateAdaptiveContorl\_Chart



#### **Description of Diagram (rateAdaptiveControl\_Chart)**

There are four states for designing rate adaptive mode. They are INITIALIZE; START; INCREMENT; DECREMENT. From the start, the previousLevel will be set to 0 and RATE will be set to Lower Rate Limit (LRL). Then the state will go to START and initialize reactionTime\_remaining equal to reactionTime which is a programmable parameter from the DCM and initialize recoveryTime\_remaining equal to recoveryTime which is a programmable parameter from the DCM. If rateAdaptive is disabled, the rate will equal to LRL. The LED will show green for START state, which means no change in RATE is needed.

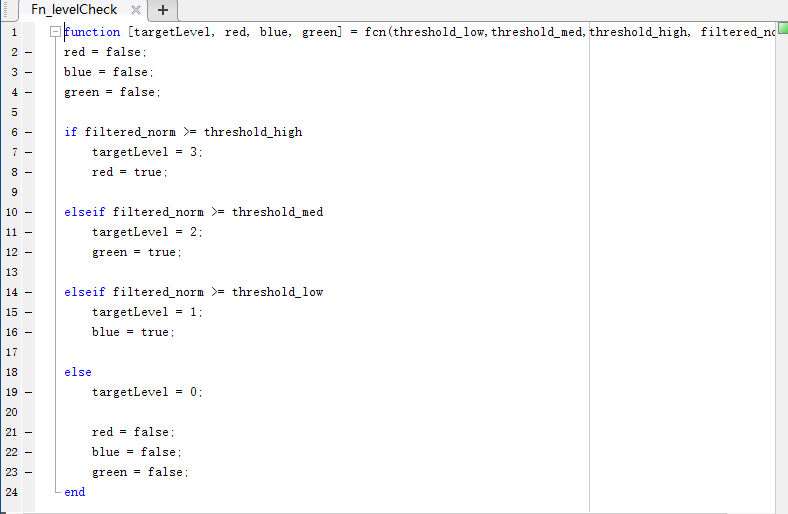
Start state will decide to go to the INCREMENT state if targetLevel is greater than the previousLevel while the rateAdaptive is enabled. During the INCREMENT state, the step time it takes the rate to increase to MSR will be incrementTime. incrementTime is determined by both reactionTime and the response factor. The exact formula is as follows: reactionTime/((target - previousLevel)\*RF). This state will keep looping itself with a time period of incrementTime between each loop. After each loop, incrementTime will be subtracted from reactionTime\_remaining. Then it will keep in the INCREMENT state until either the reactionTime remaining is equal or less than zero or the targetLevel changes to be less than or equal to the previousLevel. When there is no reactionTime\_remaining left or the targetLevel is less than the previousLevel, it will set the previousLevel to the targetLevel and return to START state. During INCREMENT looping condition the LED\_blue will be enabled.

Start state will decide to go to the DECREMENT state if targetLevel is less than the previousLevel while the rateAdaptive is enabled. During the INCREMENT state, the step time takes the rate to decrease to LRL will be decrementTime that from recoveryTime/(( previousLevel -target)\*RF). This state will keep looping itself with a time period of decrementTime between each loop. The recoveryTime\_remaining will minus the decrement time. Then it will keep in the DECREMENT state until the recoveryTime remaining is equal or less than zero. When there is no recoveryTime\_remaining left and the targetLevel is greater than the previousLevel, it will return to START state. During DECREMENT looping condition the LED\_red will be turned on.

#### **States (rateAdaptiveContorl\_Chart)**

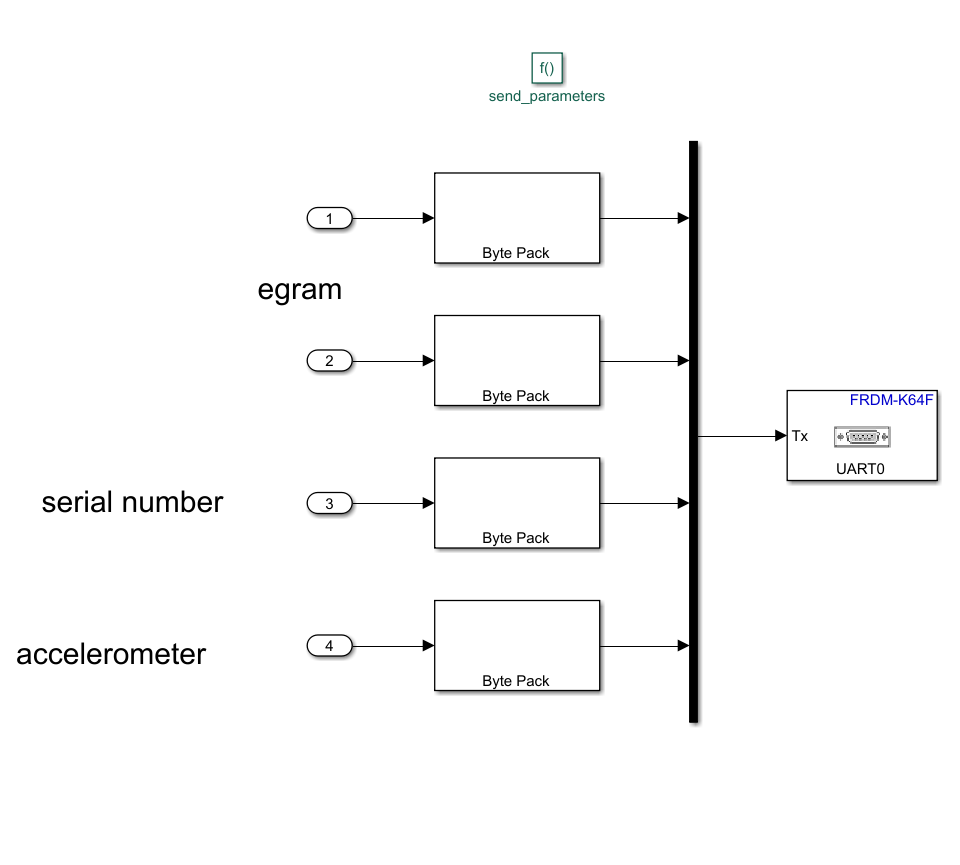
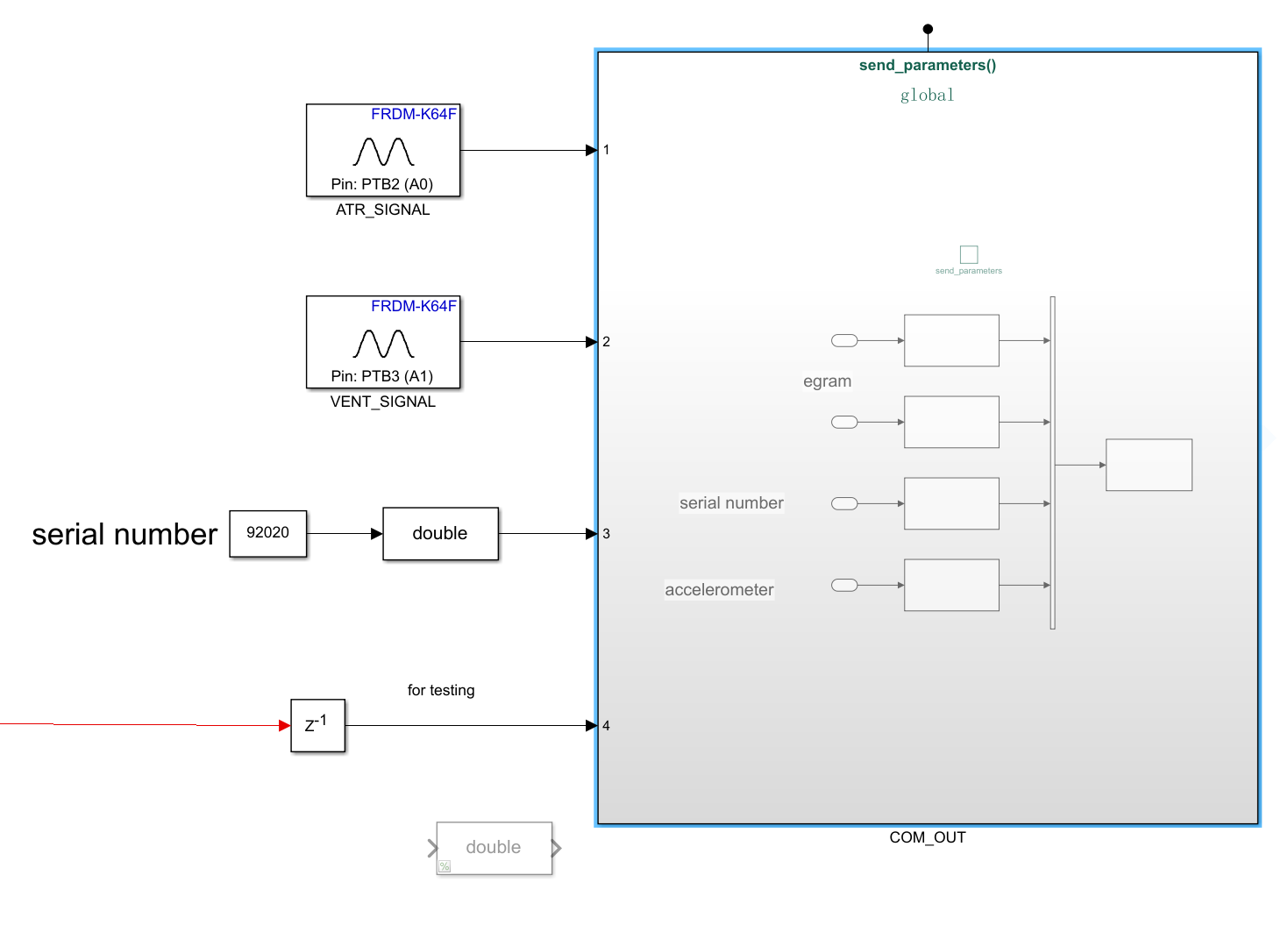
|  |  |
| --- | --- |
| State name | State description |
| INITIALIZE | Initialize the previousLevel to zero and RATE to lower rate limit. |
| START | Reset the reactionTime\_reaminging and recoverTime\_reminging and wait for the next activity data. |
| INCREMENT | Increase the pacing rate regarding the activity data and programmable parameters |
| DECREMENT | Decrease the pacing rate regarding the activity data and programmable parameters |

### Function (Fn\_levelCheck)



This function is used to determine the targetLevel used in the rateAdaptiveControl\_Chart . The targetLevel is set to 0, 1, 2, 3, respectively regarding the different programmable parameters sent by the DCM. After the filtered acceleration input feeds into this function, it will be used to compare with the user input of threshold (med, low, high, or zero) to set the targetLevel accordingly.

### Function (send\_parameters)

This function is used to send the real time data from the FRDM-K64F to the DCM in order to display. There are 3 main data parameters being sent to DCM, which are serial number of the pacemaker and both atrium pacing signal and ventricular pacing signal for creating the egram. The last parameter was added purely for testing purposes. The send\_parameters() function converts the double inputs to uint8 bytes (using Byte Pack in Simulink) for serial transmit.

### Transition Table (COM\_IN)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Source State** | **Condition** | **Entry Actions** | **Condition Actions** | **Exit Actions** | **Transition**  **Actions** | **Destination State** | **During**  **Actions** |
| **INITIAL** | **NC** | **paceLocation = 1;**  **sensingTrue = 0;**  **LRL = 60;**  **MSR = 120;**  **ATRpulseAmp = 50;**  **VENTpulseAmp = 50;**  **ATRpulseWidth = 10;**  **VENTpulseWidth = 10;**  **ARP = 250; %msec**  **VRP = 250; %msec**  **AtrSensitivity = 50;**  **VentSensitivity = 50;**  **AVdelay = 50.0;**  **rateAdaptiveTrue = 0;**  **acc\_threshold\_LOW = 1.1; %ajust according to Accelerometer single**  **acc\_threshold\_MED = 1.5;**  **acc\_threshold\_HIGH = 2;**  **responseFactor = 1;**  **reactionTime = 10;**  **recoveryTime = 30;** | **NC** | **NC** | **NC** | **STANDBY** | **NC** |
| **STANDBY** | **[status==0]** | **NC** | **NC** | **NC** | **NC** | **STANDBY** | **NC** |
| **[status==0] &&**  **[rxdata(1) == hex2dec('16')]&&**  **rxdata(2) == hex2dec('55')]** | **NC** | **NC** | **NC** | **NC** | **SET\_PARAMETERS** | **NC** |
| **[status==0] &&**  **[rxdata(1) == hex2dec('16')]&&**  **rxdata(2) == hex2dec('22')]** | **NC** | **NC** | **NC** | **NC** | **ECHO\_PARAMETERS** | **NC** |
| **SET\_PARAMETERS** | **NC** | **paceLocation = rxdata(3);**  **sensingTrue = rxdata(4);**  **LRL = rxdata(5);**  **MSR = rxdata(6);**  **ATRpulseAmp = typecast(rxdata(7:10), 'single');**  **VENTpulseAmp = typecast(rxdata(11:14),'single');**  **ATRpulseWidth = typecast(rxdata(15:18),'single');**  **VENTpulseWidth = typecast(rxdata(19:22),'single');**  **ARP = typecast(rxdata(23:26),'single');**  **VRP = typecast(rxdata(27:30),'single');**  **AtrSensitivity = typecast(rxdata(31:34), 'single');**  **VentSensitivity = typecast(rxdata(35:38), 'single');**  **AVdelay = typecast(rxdata(39:40), 'uint16');**  **rateAdaptiveTrue = typecast(rxdata(41),'uint8');**  **responseFactor = typecast(rxdata(42),'uint8');**  **acc\_threshold\_LOW = typecast(rxdata(43:46), 'single');**  **acc\_threshold\_MED = typecast(rxdata(47:50), 'single');**  **acc\_threshold\_HIGH = typecast(rxdata(51:54), 'single');**  **reactionTime = typecast(rxdata(55),'uint8');**  **recoveryTime = typecast(rxdata(56:57), 'uint16');** | **NC** | **NC** | **NC** | **STANDBY** | **NC** |
| **ECHO\_PARAMETERS** | **NC** | **send\_parameters();** | **NC** | **NC** | **NC** | **STANDBY** | **NC** |

### 

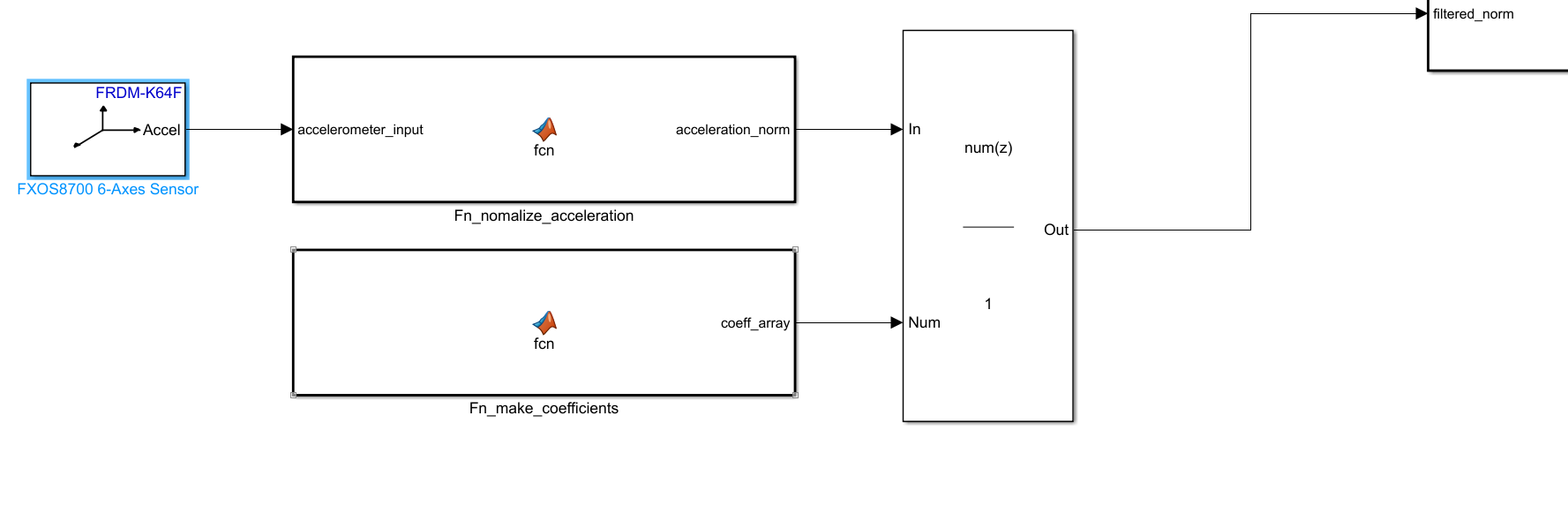
### Transition Table(rateAdaptiveControl\_Chart)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Source State** | **Condition** | **Entry Actions** | **Condition Actions** | **Exit Actions** | **Transition**  **Actions** | **Destination State** | **During**  **Actions** |
| **INITIALIZE** | **NC** | **previousLevel = 0;**  **RATE=LRL;** | **NC** | **NC** | **NC** | **START** | **NC** |
| **START** | **[targetLevel < previousLevel && rateAdaptiveTrue]** | **reactionTime\_remaining = reactionTime;**  **recoveryTime\_remaining = recoveryTime;**  **LED\_red = false;**  **LED\_blue = false;**  **LED\_green = true;** | **{target = targetLevel}** | **NC** | **NC** | **DECREMENT** | **if ~rateAdaptiveTrue**  **RATE = LRL;**  **end** |
| **targetLevel > previousLevel && rateAdaptiveTrue]** | **Same as above** | **{target = targetLevel}** | **NC** | **NC** | **INCREMENT** | **Same as above** |
| **INCREMENT** | **after(incrementTime,sec)** | **incrementTime=reactionTime/((target - previousLevel)\*RF);**  **LED\_green = false;**  **if RATE < LRL**  **RATE = LRL;**  **elseif RATE > MSR**  **RATE = MSR;**  **end** | **RATE=RATE+1;reactionTime\_remaining=reactionTime\_remaining-incrementTime;LED\_blue = true** | **NC** | **NC** | **START** | **NC** |
| **[(reactionTime\_remaining <= 0) || (targetLevel <= previousLevel) || ~rateAdaptiveTrue]** | **Same as Above** | **RATE=RATE+1;reactionTime\_remaining=reactionTime\_remaining-incrementTime;LED\_blue = true;** | **NC** | **NC** | **INCREMENT** | **NC** |
| **DECREMENT** | **(recoveryTime\_remaining <= 0) || (targetLevel >= previousLevel) || ~rateAdaptiveTrue]** | **decrementTime=recoveryTime/((previousLevel - target)\*RF);**  **LED\_green = false;**  **if RATE < LRL**  **RATE = LRL;**  **elseif RATE > MSR**  **RATE = MSR;**  **end** | **{previousLevel = target}** | **NC** | **NC** | **START** | **NC** |
| **after(decrementTime, sec)[recoveryTime\_remaining>0]** | **Same as Above** | **RATE=RATE-1;recoveryTime\_remaining=recoveryTime\_remaining-decrementTime;LED\_red = true** | **NC** | **NC** | **DECREMENT** | **NC** |

### Transition Table(Pacing)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Source State** | **Condition** | **Entry Actions** | **Condition Actions** | **Exit Actions** | **Transition**  **Actions** | **Destination State** | **During**  **Actions** |
| VENT\_CHARGING\_AND\_DISCHARGE | [paceLocation==1] | msecPerPace = 60000/LRL  sensingCTRL = sensing;  ATR\_PACE\_CTRL = false;  VENT\_PACE\_CTRL = false;  PACE\_CHARGE\_CTRL = true;  PACE\_GND\_CTRL = true;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false  VENT\_GND\_CTRL = true;  ATR\_GND\_CTRL = false; | NC | NC | NC | ATR\_CHARGING\_AND\_DISCHARGE | NC |
| after((msecPerPace-pulseWidth), msec) | Same as above | PACE\_CHARGE\_CTRL = false | NC | NC | VENT\_PACING | NC |
| after((VRP-pulseWidth),msec) | Same as above | NC | NC | NC | SENSING\_VENT\_TRUE | NC |
| paceLocation = 3 | Same as above | NC | NC | NC | State = DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT | NC |
| VENT\_PACING | after(pulseWidth, msec) | PACE\_GND\_CTRL = true;  VENT\_PACE\_CTRL = true;  VENT\_GND\_CTRL = false;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  ATR\_GND\_CTRL = false;  ATR\_PACE\_CTRL = false;  sensingCTRL = sensing; | NC | NC | NC | VENT\_CHARGING\_AND\_DISCHARGE | NC |
| ATR\_CHARGING\_AND\_DISCHARGE | after((msecPerPace), msec) | msecPerPace = 60000/LRL;  sensingCTRL = sensing;  ATR\_PACE\_CTRL = false;  VENT\_PACE\_CTRL = false;  PACE\_CHARGE\_CTRL = true;  PACE\_GND\_CTRL = true;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  VENT\_GND\_CTRL = false;  ATR\_GND\_CTRL = true; | PACE\_CHARGE\_CTRL = false; | NC | NC | ATR\_PACING | NC |
| after((ARP-pulseWidth),msec)  [sensingCTRL && atr\_detect] | Same as above | NC | NC | NC | SENSING\_ATR\_TRUE | NC |
| paceLocation = 2 | Same as above | NC | NC | NC | VENT\_CHARGING\_AND\_DISCHARGE | NC |
| paceLocation = 1 | Same as above | NC | NC | NC | ATR\_CHARGING\_AND\_DISCHARGE | NC |
| ATR\_PACING | after(pulseWidth, msec) | PACE\_GND\_CTRL = true;  VENT\_PACE\_CTRL = false;  VENT\_GND\_CTRL = false;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  ATR\_GND\_CTRL = false;  ATR\_PACE\_CTRL = true;  sensingCTRL = sensing; | NC | NC | NC | ATR\_CHARGING\_AND\_DISCHARGE | NC |
| SENSING\_VENT\_TRUE | after(pulseWidth, msec) | sensingCTRL = true; | NC | NC | NC |  | NC |
| SENSING\_ATR\_TRUE | after(pulseWidth, msec) | sensingCTRL = true; | NC | NC | NC |  | NC |
| DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT | after(msecPerPace-VENTpulseWidth-AVdelay,msec) | msecPerPace = 60000/PPM\_Rate;  ATR\_PACE\_CTRL = false;  VENT\_PACE\_CTRL = false;  pulseAmp = ATRpulseAmp; % set amplitude  PACE\_CHARGE\_CTRL = true;  PACE\_GND\_CTRL = true;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  VENT\_GND\_CTRL = true;  ATR\_GND\_CTRL = false;  blue = true;  red = false;  green = false; | NC | NC | NC | DUAL\_ATR\_PACING | NC |
| DUAL\_ATR\_PACING | after(ATRpulseWidth,msec) | PACE\_CHARGE\_CTRL = false;  PACE\_GND\_CTRL = true;  VENT\_PACE\_CTRL = false;  VENT\_GND\_CTRL = false;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  ATR\_GND\_CTRL = false;  ATR\_PACE\_CTRL = true; | NC | NC | NC | DUAL\_DISCHARGE\_ATR\_AND\_VENT\_CHARGING | NC |
| DUAL\_DISCHARGE\_ATR\_AND\_VENT\_CHARGING | after(AVdelay-ATRpulseWidth,msec) | ATR\_PACE\_CTRL = false;  VENT\_PACE\_CTRL = false;  pulseAmp = VENTpulseAmp;  PACE\_CHARGE\_CTRL = true;  PACE\_GND\_CTRL = true;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false  VENT\_GND\_CTRL = false;  ATR\_GND\_CTRL = true;  green = true; | NC | NC | NC | DUAL\_VENT\_PACING | NC |
| DUAL\_VENT\_PACING | after(VENTpulseWidth,msec) | ATR\_GND\_CTRL = false;  ATR\_PACE\_CTRL = false;  PACE\_CHARGE\_CTRL = false;  PACE\_GND\_CTRL = true;  VENT\_PACE\_CTRL = true;  VENT\_GND\_CTRL = false;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  red = true; | NC | NC | NC | DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT | NC |

### Data Smoothing



In order to properly analyze the accelerometer data for rate adaptive modes, the volatile data needed smoothing. First, the acceleration vector was normalized using

*sqrt((accelerometer\_input(1))^2+(accelerometer\_input(2))^2+(accelerometer\_input(3))^2);*

That value is then sent into a FIR Filter block in order for a moving average filter to be applied. We chose 500 coefficients (each with a value of 1/500) after much testing as it provided the best results.

### Modularity

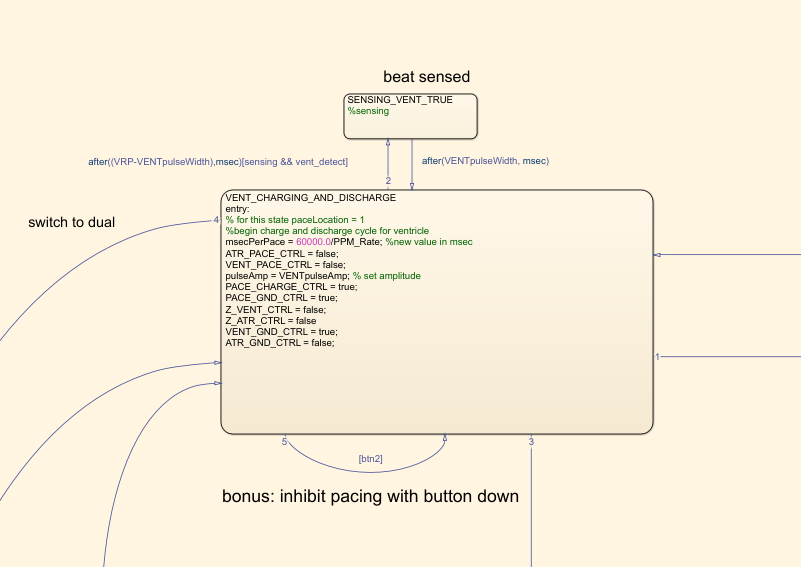
The stateflow to determine whether the Rate should increase, decrease or remain constant were separated from the main sensing stateflow. This was intentional and helped to increase modularity of the design. If new thresholds are added, it will be easy to implement them and no change is needed in the main PACING stateflow diagram. Furthermore, if more coefficients are required for the filter, the make\_coefficients function will make that a seamless process with only 1 necessary change.

## 

## Bonus

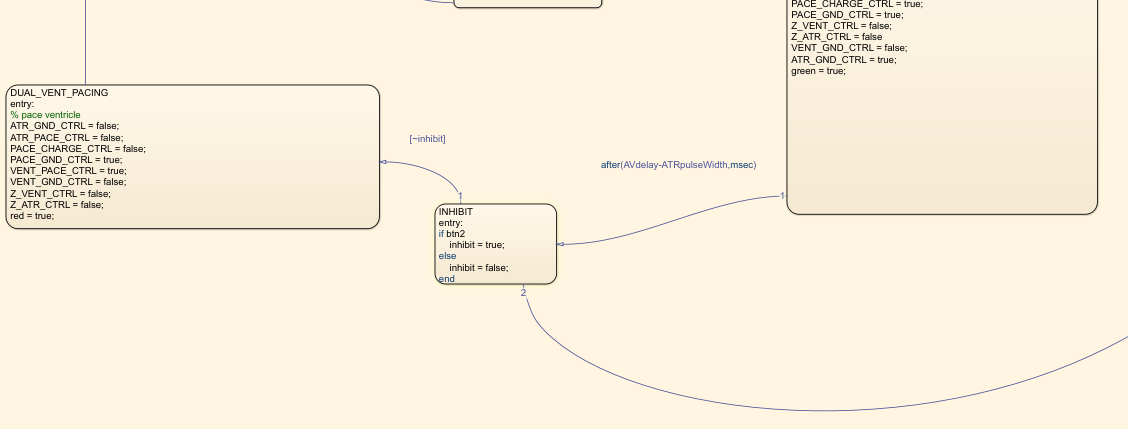
### Inhibition of Ventricular Pace

**VOO, VVI modes:**

****

While in VOO or VVI, push button 2 on the board will inhibit ventricular paces.

**DOO, DDDR modes:**

While in Dual pacing modes, holding push button 2 on the board will inhibit ventricular paces while leaving atrial paces unaffected. Releasing the button will allow a ventricular pace to occur following the next atrial pulse, once the proper AV delay has elapsed.

### DDDR Mode

### 

#### **Description of Diagram**

In order to implement DDDR mode, 2 new states were added to the Dual Pacing stateflow. These states function as atrial inhibition and ventricular inhibition. While DDDR mode is enabled, they will check for an atrium or ventricle sense (ATR or VENT\_detect == True), and only pace at the required time if nothing was detected. If a sense is detected in the atrium, there will be no atrium pulse, and the state will move to prepare for a ventricular pulse. Once more it will verify there is no pulse sensed before otherwise continuing to pace the ventricle. This mode adheres to the ARP and VRP requirements.

|  |  |
| --- | --- |
| **State name** | **State description** |
| DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT | This state is responsible for charging the C22 capacitor for pacing the atrium and discharging C21 capacitor for discharging the ventricle. |
| DUAL\_ATR\_PACING | This state is responsible for pacing the atrium |
| DUAL\_DISCHARGE\_ATR\_AND\_VENT\_CHARGING | This state is responsible for charging the C22 capacitor for pacing the ventricle. and discharging C21 capacitor for discharging the atrium. |
| DUAL\_VENT\_PACING | This state is responsible for pacing the ventricle |
| DUAL SENSING ATR TRUE | This state becomes active once an atrium sense is detected. It then proceeds to skip pacing of the atrium for that round. |
| DUAL SENSING VENT TRUE | This state becomes active once a ventricular sense is detected. It then proceeds to skip pacing of the ventricle for that round. |
| INHIBIT | This state was added for the push button bonus. Each round this state check if the button is pressed:   * If so: it will return and pace the atrium instead of ventricle. * If not, pace ventricle as normal. |

Transition\_Table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Source State** | **Event** | **Condition** | **Entry Actions** | **Condition Actions** | **Exit Actions** | **Transition**  **Actions** | **Destination State** | **During**  **Actions** |
| DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT | True | after(msecPerPace-VENTpulseWidth-AVdelay,msec) | msecPerPace = 60000/PPM\_Rate;  ATR\_PACE\_CTRL = false;  VENT\_PACE\_CTRL = false;  pulseAmp = ATRpulseAmp; % set amplitude  PACE\_CHARGE\_CTRL = true;  PACE\_GND\_CTRL = true;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  VENT\_GND\_CTRL = true;  ATR\_GND\_CTRL = false;  blue = true;  red = false;  green = false; | NC | NC | NC | DUAL\_ATR\_PACING | NC |
| True | after(ARP-VENTpulseWidth-(AVdelay-ATRpulseWidth),msec | Same as above | [sensing && atr\_detect] | NC | NC | DUAL\_SENSING\_ATR\_TRUE | NC |
| DUAL\_ATR\_PACING | True | after(ATRpulseWidth,msec) | PACE\_CHARGE\_CTRL = false;  PACE\_GND\_CTRL = true;  VENT\_PACE\_CTRL = false;  VENT\_GND\_CTRL = false;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  ATR\_GND\_CTRL = false;  ATR\_PACE\_CTRL = true; | NC | NC | NC | DUAL\_DISCHARGE\_ATR\_AND\_VENT\_CHARGING | NC |
| DUAL\_DISCHARGE\_ATR\_AND\_VENT\_CHARGING | True | after(AVdelay-ATRpulseWidth,msec) | ATR\_PACE\_CTRL = false;  VENT\_PACE\_CTRL = false;  pulseAmp = VENTpulseAmp;  PACE\_CHARGE\_CTRL = true;  PACE\_GND\_CTRL = true;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false  VENT\_GND\_CTRL = false;  ATR\_GND\_CTRL = true;  green = true; | NC | NC | NC | INHIBIT | NC |
| DUAL\_VENT\_PACING | True | after(VENTpulseWidth,msec) | ATR\_GND\_CTRL = false;  ATR\_PACE\_CTRL = false;  PACE\_CHARGE\_CTRL = false;  PACE\_GND\_CTRL = true;  VENT\_PACE\_CTRL = true;  VENT\_GND\_CTRL = false;  Z\_VENT\_CTRL = false;  Z\_ATR\_CTRL = false;  red = true; | NC | NC | NC | DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT | NC |
| INHIBIT | True | after(msecPerPace-AVdelay,msec) | if btn2  inhibit = true;  else  inhibit = false;  end | NC | NC | NC | DUAL\_ATR\_PACING | NC |
| True | [~inhibit] | Same as above | NC | NC | NC | DUAL\_VENT\_PACING | NC |
| DUAL\_SENSING\_VENT\_TRUE | True | after(VENTpulseWidth,msec) | NC | NC | NC | NC | DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT | NC |
| DUAL\_SENSING\_ATR\_TRUE | True | after(ATRpulseWidth,msec) | NC | NC | NC | NC | DUAL\_SENSING\_ATR\_TRUE | NC |

## Design Decisions

***AOO,AAI,VOO,VVI:***

The stateflow of all the pacing modes were created in one chart. Six different states were created called

ATR\_CHARGING\_AND\_DISCHARGE, ATR\_PACING, SENSING\_ATR\_TRUE, VENT\_CHARGING\_AND\_DISCHARGE, VENT\_PACING and SENSING\_VENT\_TRUE. Moving through these states allows us to change to and access the AOO, VOO, AAI and VVI modes. This was decided because it made cycling through the different states easy to manage.

Mode AOO uses the ATR\_CHARGING\_AND\_DISCHARGE and ATR\_PACING states to charge the capacitor, pace the Atrium accordingly and then discharge the capacitor. The circuit can be broken down to charging capacitor C22, disconnect power and pace the heart, and discharge capacitor C21 to GND for zero net current to the heart. ATR\_CHARGING\_AND\_DISCHARGE state is both responsible for charging the capacitor C22 before pacing and discharging capacitor C21 after pace. Then the state ATR\_Pacing is only designed for allowing current flow through the atrium to generate a successful pace.

Mode VOO uses the VENT\_CHARGING\_AND\_DISCHARGE and VENT\_PACING states to charge the capacitor, pace the Ventricle accordingly and then discharge the capacitor. The circuit can be broken down to charging capacitor C22, disconnect power and pace the heart, and discharge capacitor C21 to GND for zero net current to the heart. VENT\_CHARGING\_AND\_DISCHARGE state is both responsible for charging the capacitor C22 before pacing and discharging capacitor C21 after pace. Then the state VENT\_Pacing is only designed for allowing current flow through the atrium to generate a successful pace.

The Mode VVI uses the state SENSING\_VENT\_TRUE to finish the sensing of the ventricle

heartbeat and the mode AAI uses the state SENSING\_ATR\_TRUE to finish the sensing of the atrium heartbeat. The time duration of each sensing action is designed according to the Ventricular Refractory Period or Atrial Refractory Period. These describe the period of time following a pace or natural heartbeat where the sensing circuitry will not be active. This prevents any false senses due to the heart depolarization.

***DOO***

Mode DOO, is designed by four states. The purpose of four states is pacing the atrium and ventricle with a fixed AV delay while followed by the consistent pacing rate. To achieve this goal, atrium pacing and ventricle discharge has to be in one state. Then ventricular pacing and atrium discharge is designed in the same state as well. Therefore the four states are going to loop through with atrium charging and ventricle discharging, atrium pacing, ventricle charging and atrium discharging, ventricular pacing. With this sequence, the charging and discharging of shared two capacitors will not interfere with each other.

The timing between the four states was determined by first taking msecPerPace as the total time between atrial - atrial paces (also equal to time between ventricle - ventricle paces).

In this case, the time between DUAL\_ATR\_CHARGING\_AND\_DISCHARGE\_VENT and

DUAL\_ATR\_PACING is msecPerPace value minus the pace width of ventricle and minus the fixed AV\_delay. This assures that the total remains fixed regardless of the values set to AV\_delay or pulse width.

The time between DUAL\_DISCHARGE\_ATR\_AND\_VENT\_CHARGING and DUAL\_VENT\_PACING will be fixed\_AV\_delay minus the pace width of the atrium.

***AOOR, AAIR, VOOR, VVIR, DOOR:***

In the rate adaptive mode, the threshold values regarding the required different level should be received by the serial communication port(UART). Then it will input into a MATLAB function where threshold values will be compared with the smoothing accelerometer input to set a target level where the rate will change accordingly. The target level will be input into a chart where it will compare to the previous level to determine the action of rate is increment or decrement. The difference between levels also indicates how many times the response factor the rate needs to change by. The programmable parameters regarding the rate adaptive mode will be used in this chart as well. During the increment state, the reaction time remaining is used to determine how long is left for the increment to stop. During that period the rate is going to increment until a maximum of MSR with a difference between target level and previous level times the response factor to achieve the action where the rate increases with the change of activity data and threshold with a step size increment in specific period (reaction time). The decrement state and recovery time works in the same way, with a minimum rate of LRL. Then the increased or decreased rate will be output to the pacing module where it acts as the pacing rate per minute for AOOR, AAIR,VOOR,VVIR,DOOR,DDDR modes when rateadptiveTrue is enabled.

***Switch between modes:***

In order to switch between modes, there are three quantities being used. PaceLocation switches the mode regarding atrium, ventricle and dual, where PaceLoaction is 1,2,3 respectively. Then the variable sensing determines whether it is an I mode. For example, if sensing is true then AOO becomes AAI, and if VOOR, becomes VVIR. RateAdaptiveTrue indicates whether the mode is a R mode. For example, if RateAdaptiveTrue is true then DOO becomes DOOR. With these 3 parameters given from DCM, the stateflows can easily determine exactly which mode is being implemented.

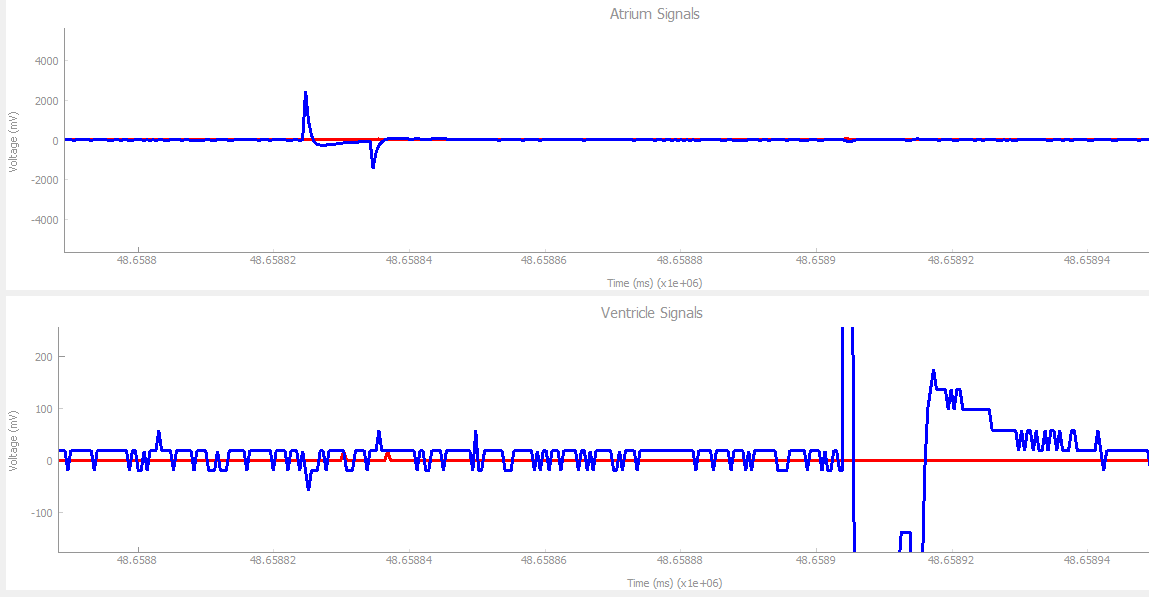
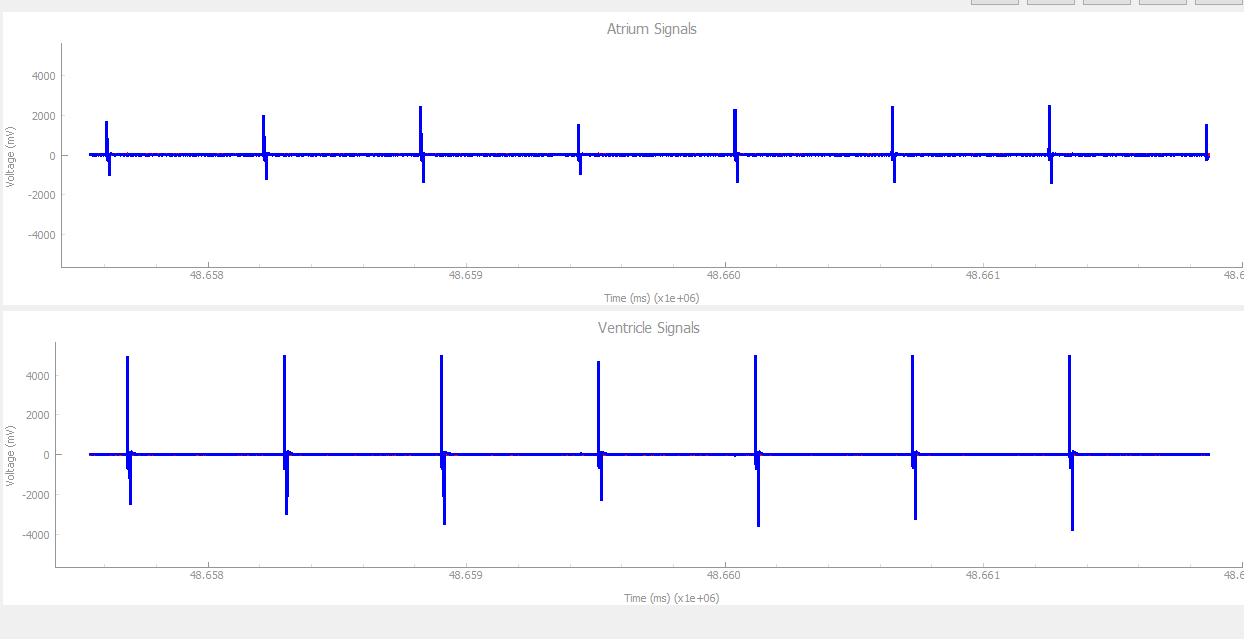
## 

## Testing (Assignment 2)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Mode Selected** | **Result (pass/fail)** | **Additional Description** |
| 1.Pacing both atrium and ventricle with proper AV delay | DOO,DOOR | pass |  |
| 2.Pace location can change through different mode | All modes | pass |  |
| 3.Rate can be increased to MSR when shaking the board | AOOR,AAIR,VOOR,VVIR,  DOOR, DDDR | pass |  |
| 4.Rate can be decreased to LRL when stop shaking the board | AOOR,AAIR,VOOR,VVIR,  DOOR, DDDR | pass |  |
| 5.Rate can be at rest level while no activity is implemented | AOOR,AAIR,VOOR,VVIR,  DOOR, DDDR | pass |  |
| 6.DDDR mode with proper AV delay | DDDR | pass/fail | AV does not not vary with change in rate. (See future changes) |
| 7.Inhibit ventricular pacing while holding down btn2 | All modes with ventricular pacing | pass |  |

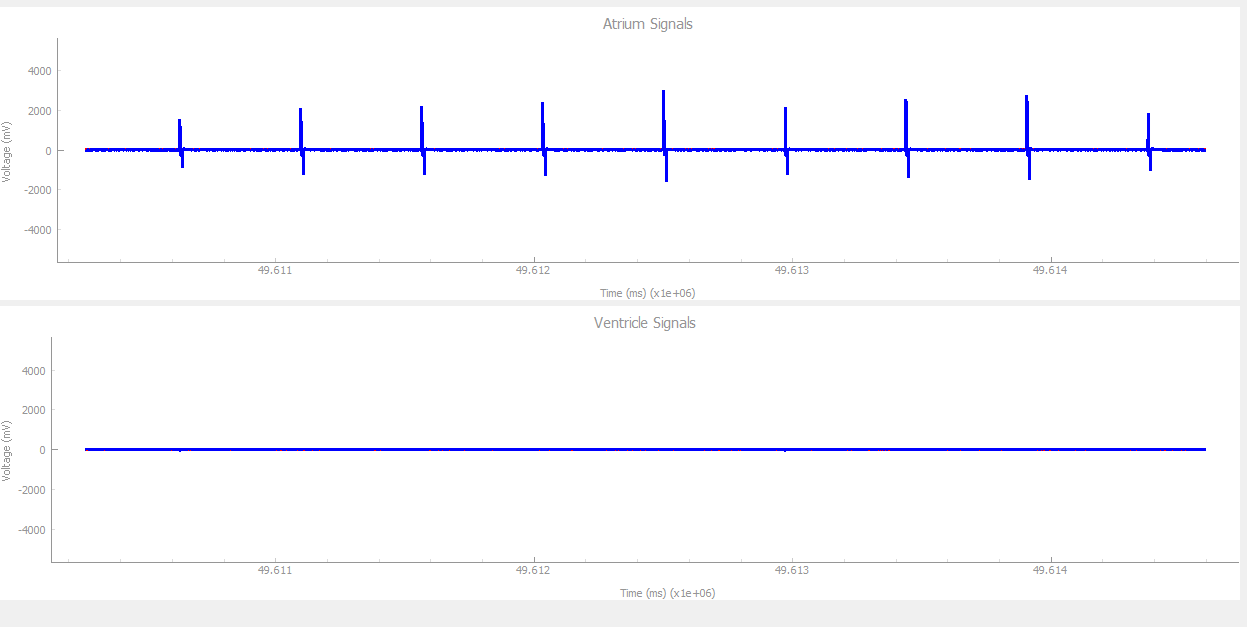
## Screenshots of Corresponding Test Cases

1. DOO with proper AV delay (set to 80 ms)



The delay between atrium and ventricle pacing is 80 ms as expected.

1. Pacing works for all modes.

1. With LRL set to 100, and MSR set to 130, shaking the pacemaker at max speed for an extended period of time does not bring pacing up any higher than 130. 
2. Similarly, the Rate never lowers past the set LRL for any R mode.
3. With no activity, the board remains at LRL.
4. DDDR mode follows correct AV delay for both sensed and tracked ventricular pacing.



1. Holding btn2 on board inhibits all ventricular paces. Upon release, pacing resumes. This occurs in VVIR, VVI, VOO, DOO and DDDR modes.



## 

## Requirement Changes That Are Likely

* More threshold levels would be great in order to design for more pacing levels in Rate Adaptive modes.
* Adaptive AV delay in DDDR mode to ensure pacing rate increases as expected for increased physical activity.
* Add additional modes including DDD, DDI, VDDD, VDDR, etc.
* Improve egram output to also include graphs of paces and inhibition clearly
* Enable hysteresis pacing.
* Implementing Atrial Tachycardia Response (ATR)

## Design Decision Changes That Are Likely

* By creating different levels of threshold in the COM\_IN states to get the user input from DCM and implement it into the Fn\_levelCheck function, so that more levels can be detected for ideal rate change in the future.
* For design hysteresis pacing, the Hysteresis Rate Limit should be also added as a parameter regarding the rate.
* More states in the simulink should be designed to add regarding the Atrial Tachycardia Detection, ATR Duration and ATR Fallback