PACEMAKER DCM
System Specifications

October 25, 2020

MECHTRON/SFWR ENG 3K04
Dr. Bokhari

-Alex Radikov - 400192066-
-Samuel de Haan - 400083906-
- Muhammad Ali - 400201794 -
- Justin Vandersar - 400136738 -

Contents

1 Introduction
2 System Requirements
2.1 List of Requirements
2.2 Requirements Likely to Change
3 Design Overview
3.1 DCM GUI Design
3.2 Design Decisions Likely to Change
4 Module Interface Specification
4.1 Table of Functions
4.2 Finite State Diagram
5 Module Internal Design
5.1 Global Constants
5.2 List Private Functions within Modules
5.3 Functionality of Private Functions
6 Testing
7 Future Additions

1 Introduction

The purpose of this documentation is to describe the functions and operational characteristics of the PACEMAKER Device Control Monitor (DCM). It will identify the functions and modules within the DCM and outline the interactions between them based on the defined requirements of the system.

2 Requirements

2.1 List of DCM Requirements

The purpose of the Device Control Monitor is to provide a user interface to facilitate the communication to the implanted PACEMAKER device.

The implementation of the DCM design shall be capable of carrying out the following requirements:

1. The user interface shall be capable of utilizing and managing windows for display of text and graphics.

2. The user interface shall be capable of processing user positioning and input buttons.

3. The user interface shall be capable of displaying all programmable parameters for review and modification.

4. The user interface shall be capable of visually indicating when the DCM and the device are communicating.

5. The user interface shall be capable of visually indicating when telemetry is lost due to the device being out of range.

6. The user interface shall be capable of visually indicating when telemetry is lost due to noise.

7. The user interface shall be capable of visually indicating when a different PACEMAKER device is approached than was previously interrogated.

2.2 Requirements Likely to Change

Throughout the development of the PACEMAKER DCM, some requirements are likely to change as development progresses. Throughout the interactions of the DCM development certain requirements will not be necessary. The list of requirements that are likely to be changed are:

1. PACEMAKER embedded Software

2. PACEMAKER hardware

3 Design Overview

3.1 DCM GUI Design

The goal of the DCM design is to achieve all of the requirements stated in section 2.1. Functionality was the priority over visual appeal. The design must be initiative and straight forward to provide ease of use.

Figure 1. Design of DCM Welcome Screen

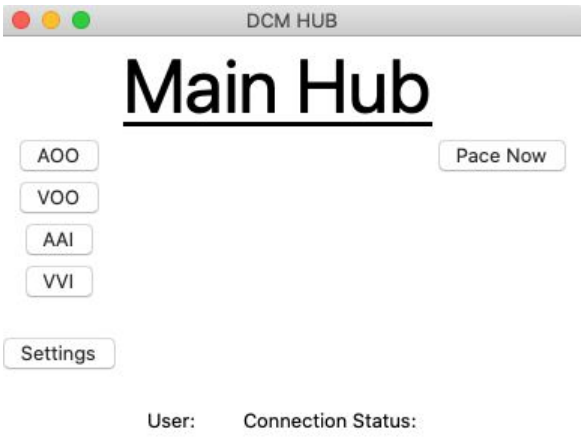


Figure 2. Design of the DCM Hub Screen

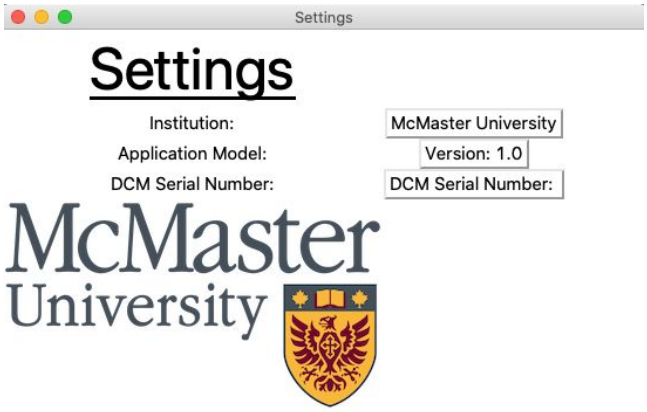


Figure 3. Design of the DCM Settings Screen

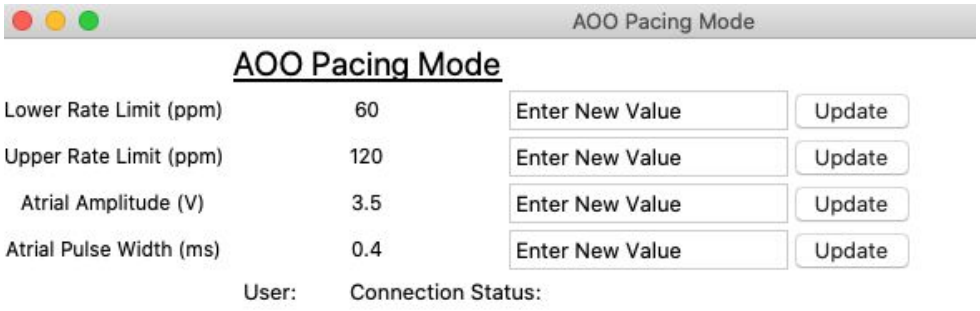


Figure 4. Design of the AOO Settings Screen

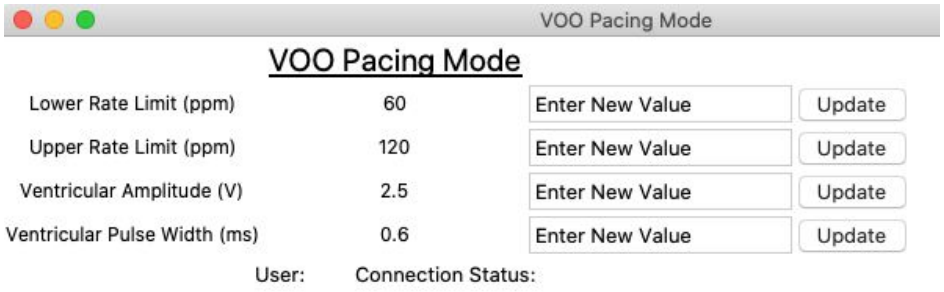


Figure 5. Design of the VOO Settings Screen

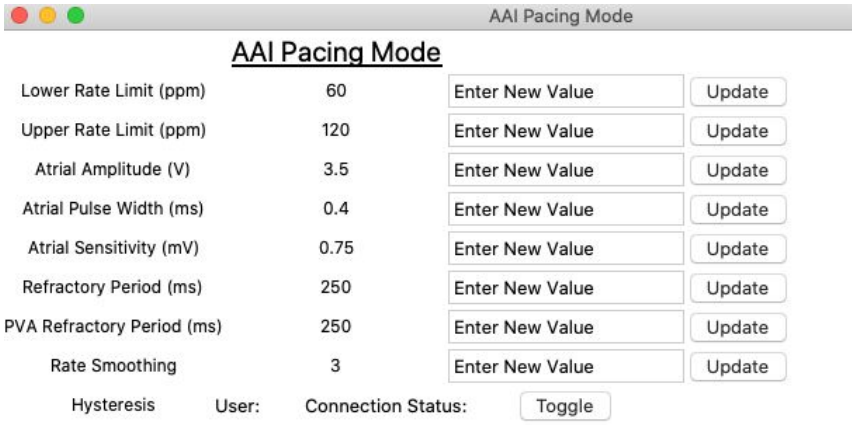


Figure 6. Design of the AAI Settings Screen

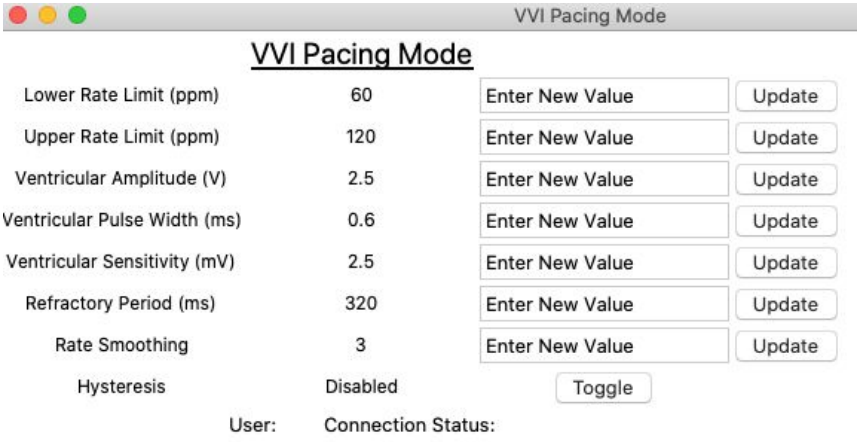


Figure 7. Design of the VVI Settings Screen

3.2 Design Decisions Likely to Change

Throughout the design process, some design decisions are likely to be changed.

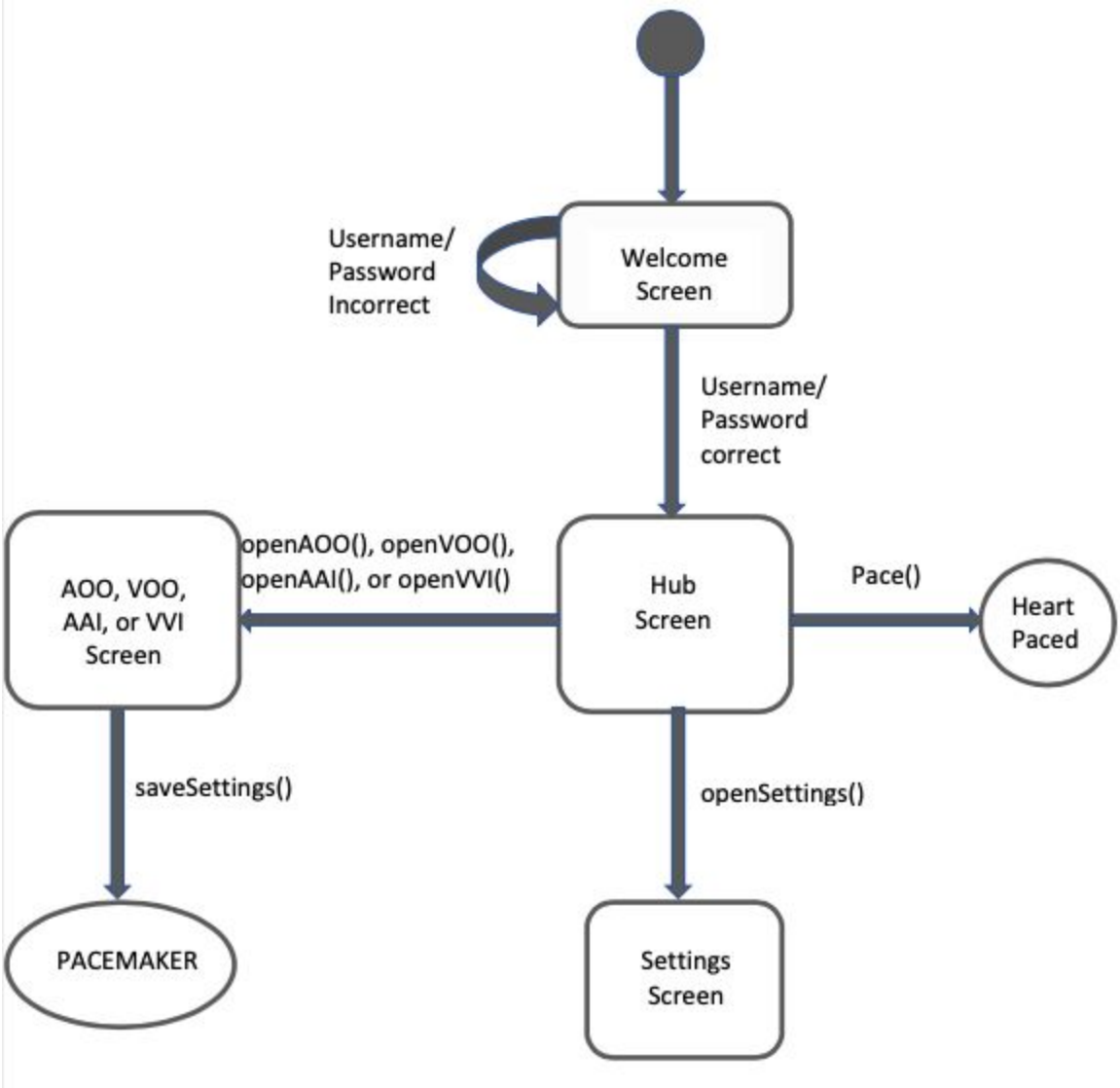
1. Addition of additional windows for more modes
2. Creation of real time Error pop up for connection lost
3. Addition of more variables as needed for additional modes

4 Module Interface Specification

4.1 Table of Modules

Name	Description
DCM_WelcomeScreen	Holds the Welcome screen for the DCM GUI
DCM_Hub	Holds the Hub screen for the DCM GUI
DCM_Setting	Holds the Settings screen for the DCM GUI
DCM_AOO, DCM_VOO, DCM_AAI, DCM_VVI	Holds the AOO, VOO, AAI, and VVI screens for the DCM GUI, respectfully

4.2 Finite State Diagram



5 Module Internal Design

5.1 Global Constants

Name	Type/Units	Value/Range/ Tol	Description
m_User	string	-	Holds the name of current user
m_Connection	boolean	True-False	Indications connection statues of PACEMAKER
p_pacingState	y_pacingState	True-False	Pacing State
p_pacingMode	y_pacingMode	AOO,VOO, AAI,VVI	Pacing Mode
p_hysteresis Hyst	boolean	True-False	True if hysteresis is to be used, False if not
Upper Rate Limit UpRL	ppm	120/ 50-175/ +/- 8	Maximum rate that ventricular events can be paced
Lower Rate Limit LowRL	ppm	60/ 30-175/ +/- 8	Minimum number of of generator pulses delivered per minute
Pulse Amplitude (Aor V) AtrAmp VentAmp	V	3.5/ 0.5-7.0 +/- 12%	Desired amplitude of ventricular pace
Pulse Width (A or V) AtrAmp VentAmp	mSec	0.4/ 0.05-1.9/ ±0.2	Desired width of a ventricular pace
Atrial Refractory Period ARP	mSec	250/ 150-500/ +/- 8	Duration of an Atrial refractory period
Ventricular Refractory Period VRP	mSec	320/ 150-500/ +/- 8	Duration of a ventricular refractory period
Post Ventricular Atrial Refractory Period PVARP	mSec	250/ 150-500/ +/- 8	Duration of an Atrial refractory period when using both Atrial and Ventricular pacing

Sensitivity (A or V) AtrSens VentSens	mV	A-0.75, V-2.5/ 0.25-1.0/ +/- 20%	Determines the sensitivity when using AAI or VVI modes respectively
Rate Smoothing RateSmo	unitless	3 (N/A)/ 3-21/ +/- 1%	Limits the pacing rate change

5.2 List Private Functions within Modules

Name	Parameters	References
checkLogin()	None	DCMWelcome()
createNewUser()	None	DCMWelcome()
openHub()	None	DCMWelcome()
openAOO()	None	DCMHub()
openVOO()	None	DCMHub()
openAAI()	None	DCMHub()
openVVI()	None	DCMHub()
openSetting()	None	DCMHub()
Pace()	None	DCMHub()
saveSettings()	None	AOOpage() VOOpage() AAIpage() VVIpage()
getModel()	None	DCMSetting()
getSerial()	None	DCMSetting()
changeLowRL()	None	AOOpage() VOOpage() AAIpage() VVIpage()

changeUpRL()	None	AOOpage() VOOpage() AAIpage() VVIpage()
changeAtrAmp()	None	AOOpage() AAIpage()
changeAtrPW()	None	AOOpage() AAIpage()
changeVentAmp( )	None	VOOpage() VVIpage()
changeVentPW()	None	VOOpage() VVIpage()
changeAtrSens()	None	AAIpage()
changeVentSens()	None	VVIpage()
changeARP()	None	AAIpage()
changeVRP()	None	VVIpage()
changePVARP()	None	AAIpage()
changeRateSmo()	None	AAIpage() VVIpage()
changeHyst()	None	AAIpage() VVIpage()

5.3 Functionality of Private Functions

DCM Welcome Screen Functions:

checkLogin()

- Gets login information from the entry fields provided. Checks to see if the given username and password are valid and contained in p_userDatabase. Opens DCMHub() if valid

createNewUser()

- Gathers new user information from entry fields provided. Checks to see if user already exist. Raises error if it does. Creates a new username and password pair in p_userDatabase if not.

openHub()

- Opens the Hub screen via function call.

DCM Hub Screen Functions

openAOO()

- Opens the AOO screen via function call.

openVOO()

- Opens the VOO screen via function call.

openAAI()

- Opens the AAI screen via function call.

openVVI()

- Opens the VVI screen via function call.

openSettings()

- Opens the settings screen via function call.

Pace()

- Paces the heart via sending instructions to the PACEMAKER.

DCM AOO, VOO, AAI, VVI Screen Functions

changeLowRL()

- Changes the value of Lower Rate Limit in LowRL.txt and displays value

changeUpRL()

- Changes the value of Upper Rate Limit in UpRL.txt and displays value

changeAtrAmp()

- Changes the value of Atrial Amplitude in AtrialAmp.txt and displays value

changeAtrPW()

- Changes the value of Atrial Pulse Width in AtrialPW.txt and displays value

changeVentAmp()

- Changes the value of Ventricular Amplitude in VentAmp.txt and displays value

changeVentPW()

- Changes the value of Ventricular Pulse Width in VentPW.txt and displays value

changeAtrSens()

- Changes the value of Atrial Sensitivity in AtrialSens.txt and displays value

changeVentSens()

- Changes the value of Ventricular Sensitivity in VentSens.txt and displays value

changeARP()

- Changes the value of Atrial Refractory Period in ARP.txt and displays value

changeVRP()

- Changes the value of Ventricular Refractory Period in VRP.txt and displays value

changePVARP()

- Changes the value of PVARP in PVARP.txt and displays value

changeRateSmo()

- Changes the value of Rate Smoothing in RateSmo.txt and displays value

changeHyst()

- Toggles Hysteresis between enabled and disabled and records setting in Hyst.txt

DCM Settings Screen Functions

getModel()

- Gets the current model/version number of the DCM from a software data text file. Returns a string of the version number.

getSerial()

- Gets the serial number of the PACEMAKER from the PACEMAKER. Returns a string of the serial number.

6 Testing

Testing of Welcome Screen

In order to test the create new user functionality of the DCM. We had to test that creating a new user would fail under two situations. The first situation is that a user already has the given name. To test, a user was created with the name “sam” and then the same user was attempted to be made again.

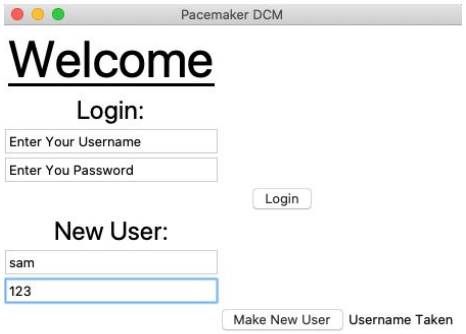


Figure 8. Username take test

This raised an error, and checking the file containing the usernames and passwords, a second occurrence of the user “sam” was absent. Therefore the test was passed.

In order to test the maximum number of users allowed, ten users were created and then an 11th was attempted to be made.

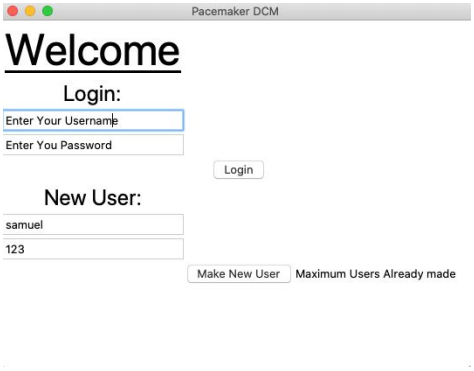


Figure 9. Over creating a user test

This raised an error and after checking the file containing the usernames and passwords, no occurrence of an 11th user was found. Therefore the test passed.

In order to test the login functionality of the DCM, two tests were conducted; one where a valid username and password were given and one where an invalid username and password were given.

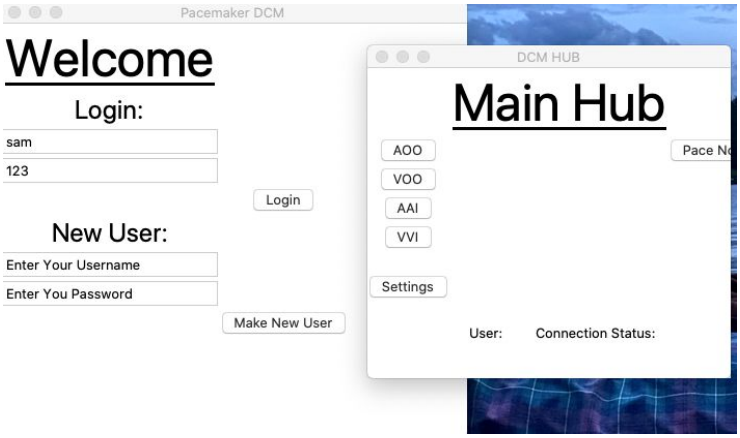


Figure 10. Test with valid username and password

Figure 11. Test with invalid username and password

Both of these tests passed as with a valid username and password the function openHub() was called and successfully executed. As with the invalid user test, it passed as an error was raised and the function openHub() was not called.

Testing AOO VOO AAI VVI Pages

In order to input new values for the variables needed in AOO, VOO, AAI, and VVI, the inputs must be in the accepted ranges outlined in section 5.1. All 4 modes follow the same logic and respond according to the variables required in each mode.

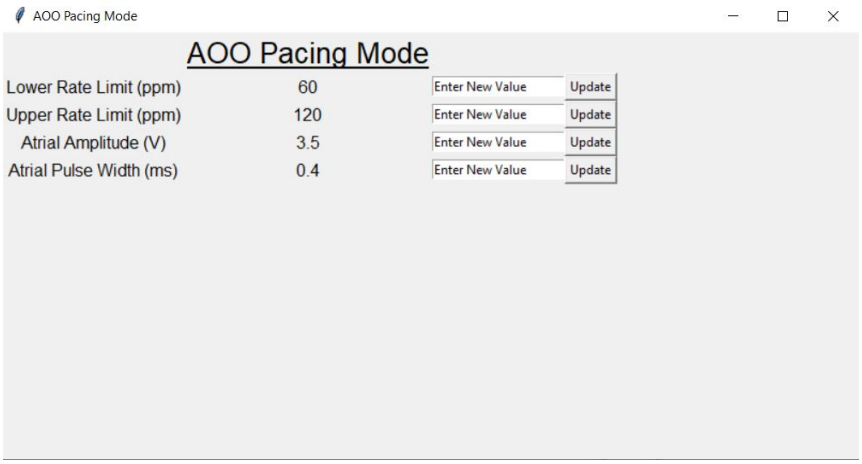


Figure 12. Default Screen before user input

Default values are set when launching the program for the first time, according to the table in 5.1

Figure 13. Alphabetical text test case

Alphabetical text is not accepted and informs the user of an error. The same is true with special characters or improperly formatted floating point values

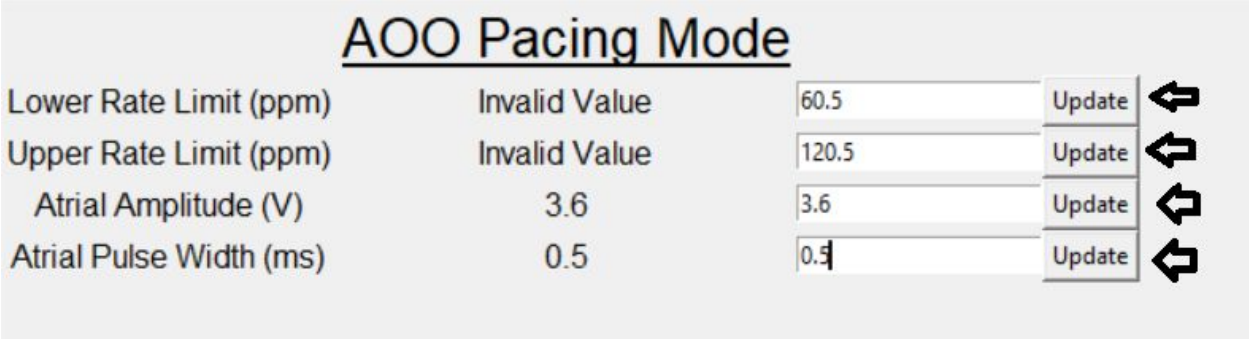


Figure 14. Floating point test case

Floating point numbers are not valid for variables that require integers, and inform the user of an error. The specific type of error is not specified.

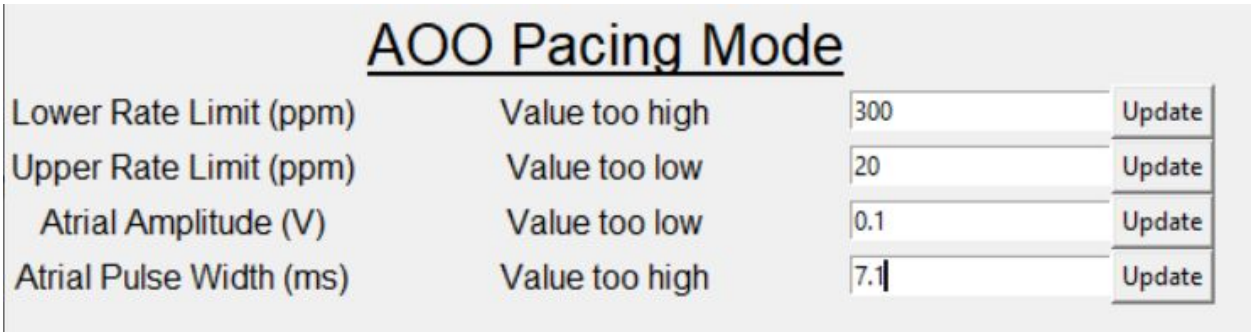


Figure 15. Range error test case

The program does not accept values above or below the required ranges for each variable. It informs the user if it is too high or too low.

Figure 16. Hysteresis Toggle test

The user can enable or disable the hysteresis mode using the toggle button. Hysteresis will be fully implemented later, but was prepared for in this version

7 Future Additions

Future Additions will have to be added to the DCM in order to complete all of the requirements. Such additions are:

1. Addition of DCM and PACEMAKER communication function
2. Addition of functional Pace function
3. Addition of further PACEMAKER modes
4. Addition of Functional Connection Statue and Error raised if lost
5. Additional warnings for incorrect variable entries
6. Hysteresis full implementation