SFWRENG 3K04

Part 2 Documentation - DCM Design

Group #12
Matt Mione, Riley Mione, Derek Paylor, Johnathan Spinelli, Laura Yang

Table of Contents

1.1 Module: dcm_main.py

1.1.1 Purpose of Module

This module is designed to be the main component of the program. Akin to an executable, this module is used to “launch” the program. With minimal code, this module interacts with the other components of the program.

1.1.2 Black Box Behaviour of Module

Upon launching the program, a small Welcome screen is presented to the user. There are two buttons, one says “Login” and the other says “Create User”. By clicking the Login button, the screen is changed to a screen with a message saying “Enter your credentials”, two entry boxes, a “Submit” button and a “Return to start page” button. Clicking the Return button takes the user back to the Welcome screen. If the user clicks the Submit button with no credentials or invalid credentials typed into the entry boxes, a pop-up window informs the user that they entered invalid credentials. If the user enters valid credentials, a new window appears (see next paragraph). If the user had clicked on the Create User button from the main menu, they would be taken to a similar looking screen as the Login screen, but this one says “Create a new user” at the top. Clicking the Return button takes the user back to the Welcome screen. If the user clicks the Submit button without entering a username, a pop-up telling the user to input a username. If the user clicks the Submit button with an existing username in the entry box, a pop-up tells the user to login with that existing account, or enter a different username. If the user clicks Submit without a password entered, a pop-up tells the user to enter a password. If the user clicks the Submit button with a unique username and a valid password, a new window appears (see next paragraph) and a new entry in the Database.JSON file appears.

The new window contains 11 tabs at the top, listing the different modes for the pacemaker to be programmed in. Each tab contains a varying number of number sliders with different labels beside each. At the bottom of the screen, the user’s username is displayed. There is a Submit button, as well as three buttons that say “Atrium Egram”, “Ventricle Egram” and “Dual Egram”. When the Atrium Egram button is clicked, a line graph with a blue line pops up. When the Ventricle Egram button is clicked, a line graph with a red line pops up. When the Dual Egram button is clicked, a line graph containing both the red line and blue line appears. The user is able to interact with the sliders, which changes the number above the slider bar. Upon clicking the submit button, a pop-up tells the user that the Pacemaker values were updated successfully. If the “Lower Rate Limit” value is greater than the “Upper Rate Limit” value, the pop-up also says that the Lower Rate Limit was fixed to 50ppm. Another pop-up tells the user that the database was successfully updated with their credentials. The file called “SerialComm.JSON” is updated with new values. The username of the current user is written in the SerialComm file, with the rest of the programmable parameters after it. If the parameter name appeared on the tab that the user clicked submit from, that parameter has the same value in the SerialComm file, but even if that parameter did not appear on that tab, the parameter still appears in the file with a value. If the pacemaker is connected to the computer upon clicking the submit button, the pacemaker will behave according to the programmed parameters.

1.1.3 Secret of Module

The secret of *dcm_main.py* is that it utilizes the other two modules of the program, *Interface.py* and *MiscFunctions.py*, to do the brunt of the work for it. The workflow is split up for neatness and separation of concerns for the program. *Interface* handles the GUI portion and most of the visual aspects of the DCM. It calls upon *MiscFunctions* for many of its processes, such as dumping and encrypting data, updating the program's user database, and the pacemaker parameters for each user, as well as for handling the serial communication between the microcontroller and the DCM host machine. Essentially, *dcm_main.py* is the main executable file, so-to-speak for the entire application.

1.2 Module: Interface.py

1.2.1 Purpose of Module

The interface module is designed to handle everything to do with user interaction. This module contains classes and functions that manage the menus and navigation through the program.

1.2.2 Global Variables in Module

This module contains three global variables.

UserID: this global variable contains the current user's username, for use in several places in the module.

AEgram: this variable was one of two to be used for Egram processing, but ended up not being used, due to the requirements change that Egram will not be mandatory.

VEgram: the other of two variables to be used for Egram processing, but not currently in use for reasons stated above.

1.2.3 Private Functions in Module

def _switch_frame(frame_class):

Takes a Tkinter frame object and switches to that part of the interface.

def _create_user(username, password, frame_class)

This function takes the desired username and password input into the two text boxes in the GUI and does some operations. It checks it against the existing database, which must be limited to 10 users, to see if it exists, and if not, will input it into the database. Also takes a frame_class object.

def _login_test(username, password, frame_class)

Function takes text-variable version of username and password, as well as frame class.

def _egramSwitch(value)

Egram has not been implemented fully yet.

def _animate(i)

Egram has not been implemented fully yet.

1.2.4 Internal Behaviour of Each Function

def _switch_frame(frame_class):

This is a function contained in the DCM class. It takes the current Tkinter frame, and then replaces it with the inputted frame_class object. It does this through using the tkinter frame.destroy() function and then setting the window's frame equal to the desired frame.

def _create_user(username, password, frame_class)

This function encrypts the given username and password using the encrypt() function. After encryption, the encrypted values are compared against the existing database of users. In order for user creation to be successful, the database must have less than 10 existing users, the username must not already exist, and at least one character has to be typed for both username and password. If these conditions are met, the database is updated with the user's encrypted credentials, and the current frame is switched. If one or more of the conditions are not met, a message box pops up with the first condition that was not met.

def _login_test(username, password, frame_class)

Function encrypts the given username and password combination. If the encrypted username and password do not match the database, the combination is rejected and the user is informed of the condition that was not met. If the username does not exist, the user is also informed.

def _egramSwitch(value)

This function controls which set of ECG data would be displayed for the user. Depending on the value that is passed to the function, the global variables AEgram and VEgram are altered to define which values should be displayed on the plot. Calls animation function.

def _animate(i)

Reads values from text files and plots values on the plot. Depending on the values of AEgram and VEgram, it selects which text file to read from.

1.2.5 Module's Secret

To prevent the user from inputting invalid pacemaker parameters, the interface uses sliders to obtain values. The sliders are able to limit the input values between a maximum and minimum acceptable value with appropriate increments. To reduce the number of functions in the module, the same function is used to collect values from each parameter tab. When a programmable parameter isn't used in a specific mode, the program passes that parameter's "nominal value", which is a value within its functioning range.

1.3 Module: MiscFunctions.py

1.3.1 Purpose of Module

This module controls the data interactions and other miscellaneous functions. This involves the databases where the data is stored and communication of data between devices. It contains many public functions used by other modules as well.

1.3.2 Global Variables in Module

database: dictionary used to store all username and password combinations of existing users.

pacemaker_values: dictionary used to store pacemaker parameters for each user.

board: a PySerial object used to communicate over USB with the microcontroller.

1.3.3 Constants

SHIFT: constant value used for encryption

baud_rate: constant value used for serial communication

UPLOAD_LOCATION: constant string variable that is declared globally, denotes where encrypted user database will be dumped to on local machine.

DUMP_LOCATION: constant string variable that is declared globally, denotes where the pacemaker's inputted values will be dumped in the local machine.

```
''' DATABASE '''
# Checks if the json file exists
if os.path.exists(DUMP_LOCATION):
    # if it exists, load in that database as the current database. Now it has memory
    with open(DUMP_LOCATION) as f:
        database = json.load(f)
```

SHIFT: constant integer used in encryption and decryption processes.

1.3.4 Public Functions Provided by Module

def update_info(mode, low, up, AAmp, VAmp, APW, VPW, ASense, VSense, ARP, VRP, MaxSense, PVARP, FAVD, ReTime, RecTime, RespFact, AThresh, user)

Neatly updates dictionary with pacemaker parameters as per requirements in documentation and communicates the parameters to the pacemaker.

class IO (*args, **kwargs)

Simple class to do with file i/o and encryption/decryption. Consists of three class methods.

def decrypt(some_phrase)

This function takes the given phrase and returns the decrypted message.

def dump(path, data_dict)

This function writes the current state of the user database to a json file. Having a local copy is very important, since this copy can be written to memory and accessed whether or not the DCM python script is running or not, and when the script reboots it remembers who has been registered.

def encrypt(some_phrase)

This function takes the given phrase and returns the encrypted message.

def communicate_parameters(mode, low, up, AAmp, VAmp, APW, VPW, ASense, VSense, ARP, VRP, MaxSense, FAVD, ReTime, RecTime, RespFact, AThresh):

Takes input values and communicates them to the board.

1.3.5 Private Functions in Module

def _to_bytes(mode, low, up, Aamp, Vamp, Apw, Vpw, Asense, Vsense, ARP, VRP, MSR, FAVD, RE, REC, RES, AT):

Takes input variables and packs them into a byte array, including parity byte at the end.

1.3.6 Internal Behaviour of Each Function

def update_info(mode, low, up, AAmp, VAmp, APW, VPW, ASense, VSense, ARP, VRP, MaxSense, PVARP, FAVD, ReTime, RecTime, RespFact, AThresh, user)

This function checks the parameters *low* and *up*, the lower and upper pacing thresholds respectively, and makes sure that *low* is not greater than *up*, and if it is, *low* is set equal to 50 to circumvent that from happening. If the *mode* involves a maximum sensor rate, the function checks that *low* is less than the maximum sensor rate. If it isn't, the function returns and displays a pop-up that informs the user of this issue. If there is no maximum sensor rate issue, the program writes the pacemaker values for that username to the JSON file *SerialComm.json*. The function uses the *communicate_parameters* function to send the parameters to the pacemaker.

class IO (*args, **kwargs)

def decrypt(some_phrase)

This function uses the constant SHIFT to negatively shift the ASCII value of each letter, which effectively decrypts the phrase.

def dump(path, data_dict)

The contents in data_dict are written into the path and formats it in a way that is easy to read, both for the user and the program.

def encrypt(some_phrase)

This function uses the constant SHIFT to positively shift the ASCII value of each letter, which effectively encrypts the phrase.

def communicate_parameters(mode, low, up, AAmp, VAmp, APW, VPW, ASense, VSense, ARP, VRP, MaxSense, FAVD, ReTime, RecTime, RespFact, AThresh):

Receives input parameters and converts them to a format that can be used to communicate to the board. Sends the parameters to the board and confirms that the board accepted the values. Compares parameters on the board with parameters from DCM to make sure there are no inconsistencies.

def _to_bytes(mode, low, up, Aamp, Vamp, Apw, Vpw, Asense, Vsense, ARP, VRP, MSR, FAVD, RE, REC, RES, AT):

Converts each programmable parameter into the data type accepted by the pacemaker and forms them into a byte array with a parity byte at the end.

1.3.7 Module's Secret

This module uses a very basic encryption method to keep user credentials safe. This is used to make sure that if someone were to read the database file, they wouldn't know every user's credentials. By simply changing the ASCII value of each character, the credentials are encrypted and decrypted.

To communicate with the pacemaker, the DCM uses serial communication. The PySerial object, *board*, needs to be configured for the host machine. The *port* parameter needs to be changed to the USB port that the microcontroller is plugged into. Serial communication allows values to be transmitted very quickly to the pacemaker. To ensure that the values were being passed in the correct order and were passed correctly, the pacemaker sends the received values back to the DCM to confirm.

2. Requirements Likely to Change

The initial requirements for the DCM specified a functional ECG component. The requirements have changed and no longer require an ECG. Development for displaying the ECG was started and included, but it is not functional. For future models of the DCM, the ECG would likely be implemented and functioning, or removed altogether. The initial requirements also required two-way communication between the board and the DCM, but communication from the board to the DCM is no longer required. Two-way communication is already implemented, but not to the point where it would work for the ECG. Future models will have communication implemented for more advanced communication between the board and the DCM.

3. Design Decisions Likely to Change

The program is split into five different files - 3 python files and 2 JSON files. The python files are separated based on functionality, but they could be separated further based on their content. The program functions properly with the current configuration, but to make it easier to adjust or implement new functions, the files can be made more specific.

The GUI consists of multiple windows with varying sizes. The largest window takes up the majority of the screen and would not be ideal for a device with a small screen size. A future implementation would likely have everything be displayed on a single window. For accessibility purposes, the window could have a scrollbar that allows users to navigate through the pages with ease.

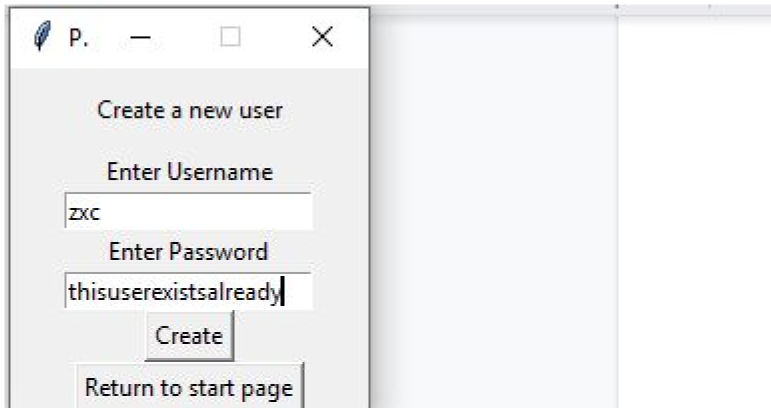
Modularity was a large factor in the design of the program. Separating the program into modules made development easy, as team members were able to work on their own module without any conflict. Modularity allowed for separation of concerns. Components that deal with communicating with the pacemaker and data files are encapsulated within one module, while other functions dealing with the interface are encapsulated in another. This created high cohesion and low coupling in the program.

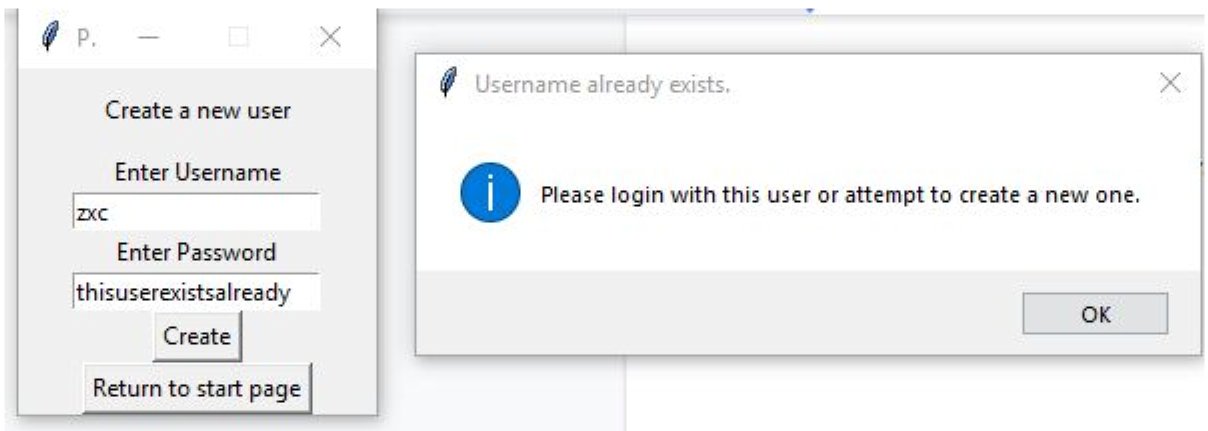
In the case where an external element is missing (ie. file doesn't exist, file name change), the program allows for an easy correction. File names and databases are declared and visible near the top of each module. Data access or delivery is broken down into small functions, that make changing the code easy. For example, the ECG functionality is not fully implemented, but since the process is composed of smaller processes, the code can be altered easily to change how the values are stored and accessed.

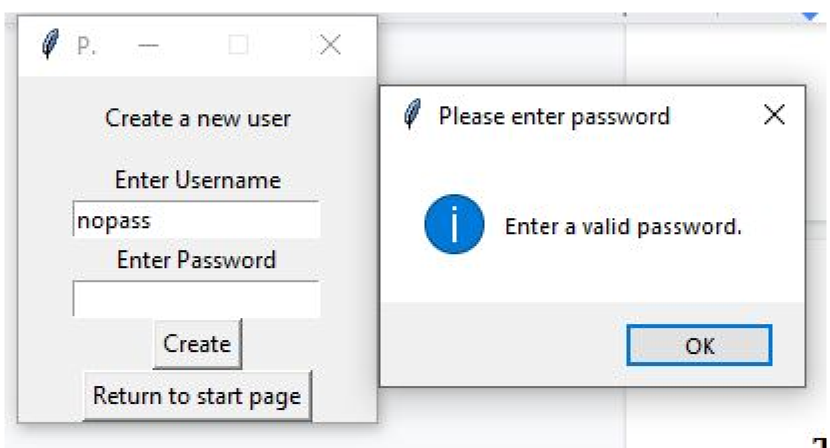
4. Testing DCM

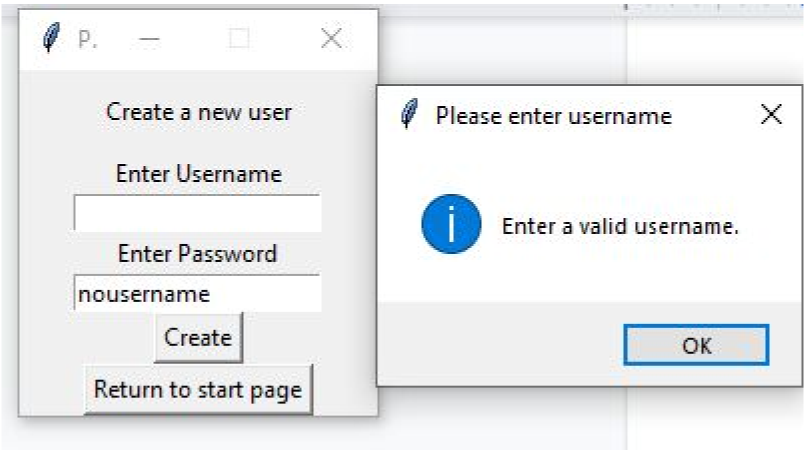
4.1 User Creation Procedure:

A few errors can occur in the user creation procedure. These being if the database is full, if the username exists, and if there is no password or username entered and submit is clicked.



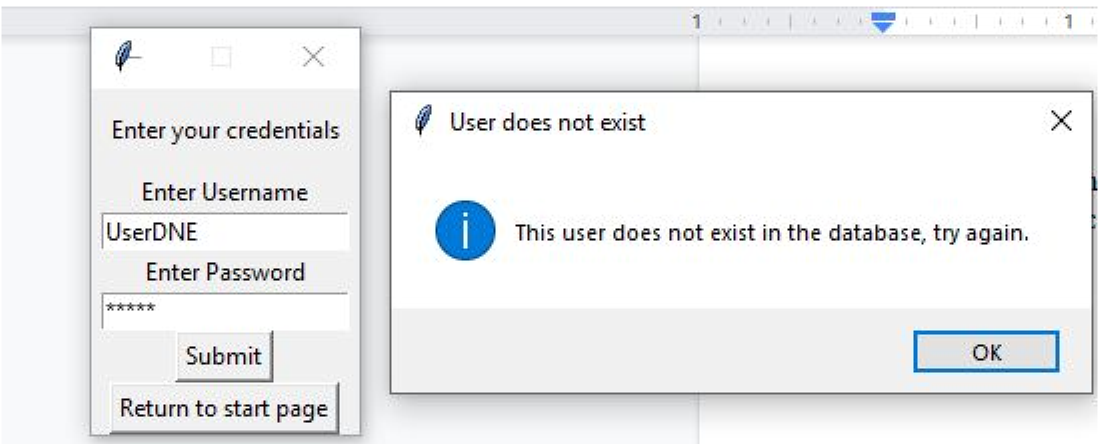




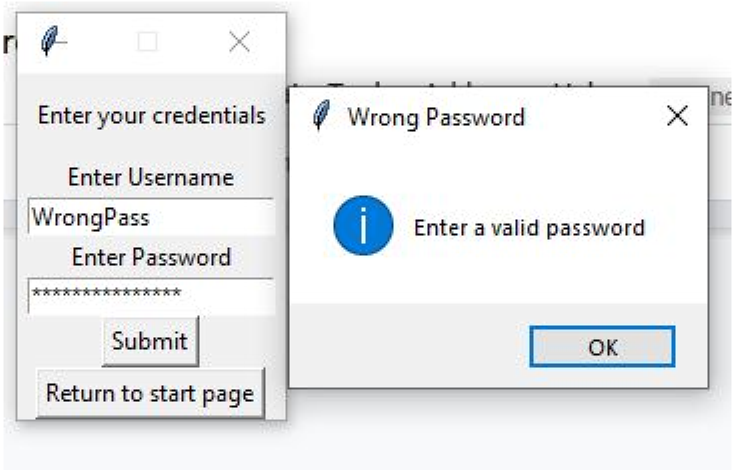


4.2 Login Procedure:

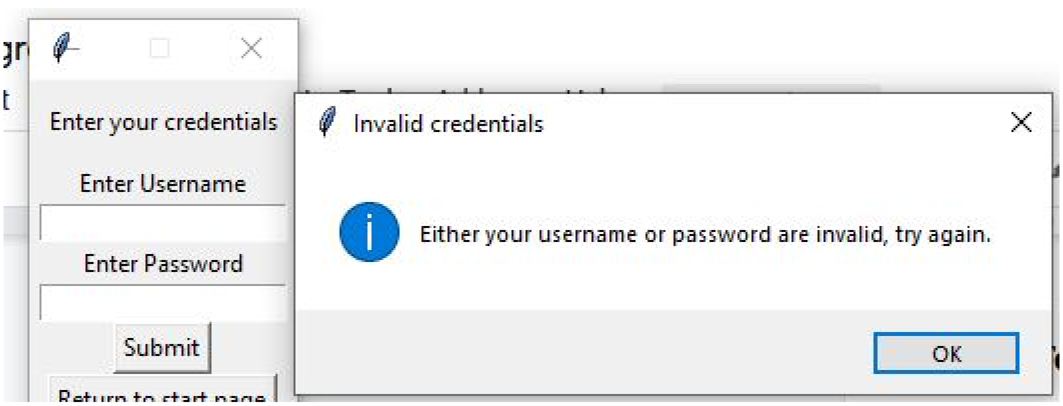
Here we entered a username that didn't exist, expecting it to show a User Does Not Exist window, and it had the expected behaviour.



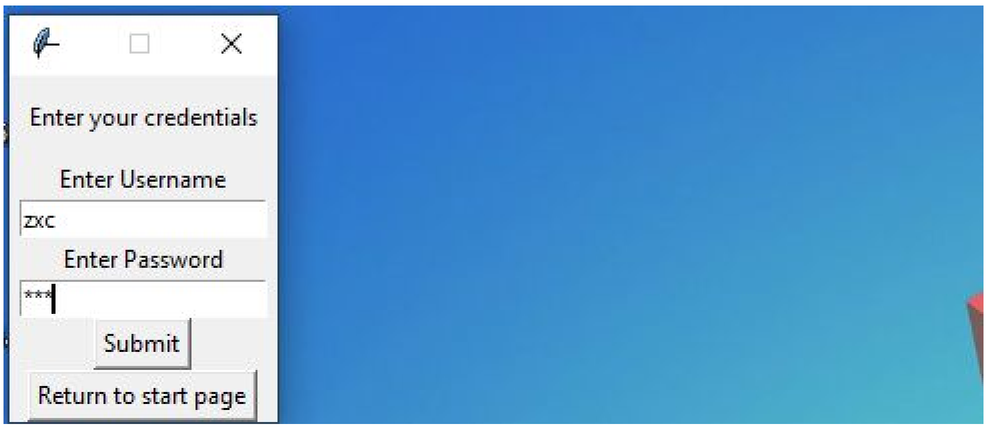
Here we entered the incorrect password, expecting it to show that the password was incorrect. It behaved as expected.



This case involves having either the username or password be blank. The program performed as expected and sent out the corresponding window.



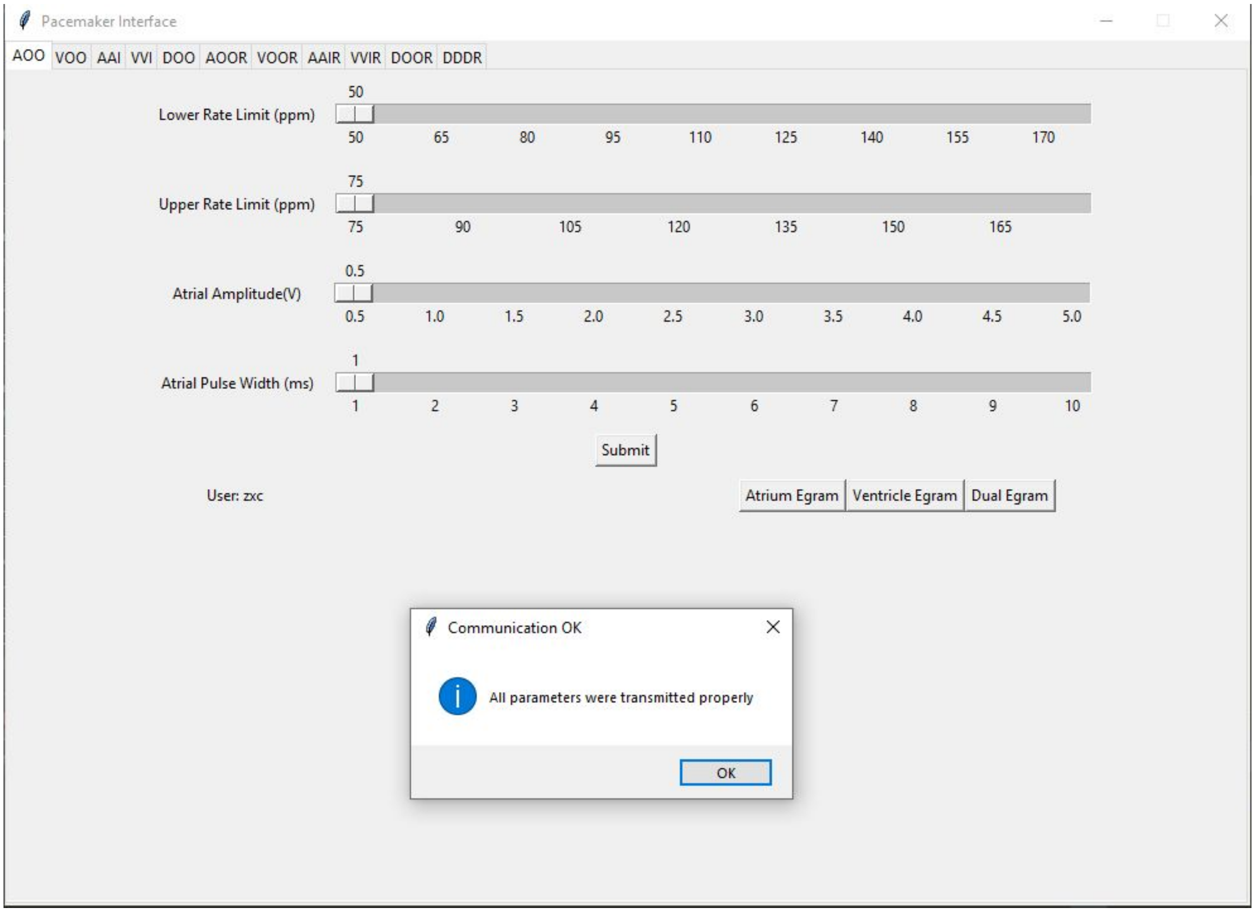
Here we are testing the case that the user exists and that the password is correct. The login should go smoothly.

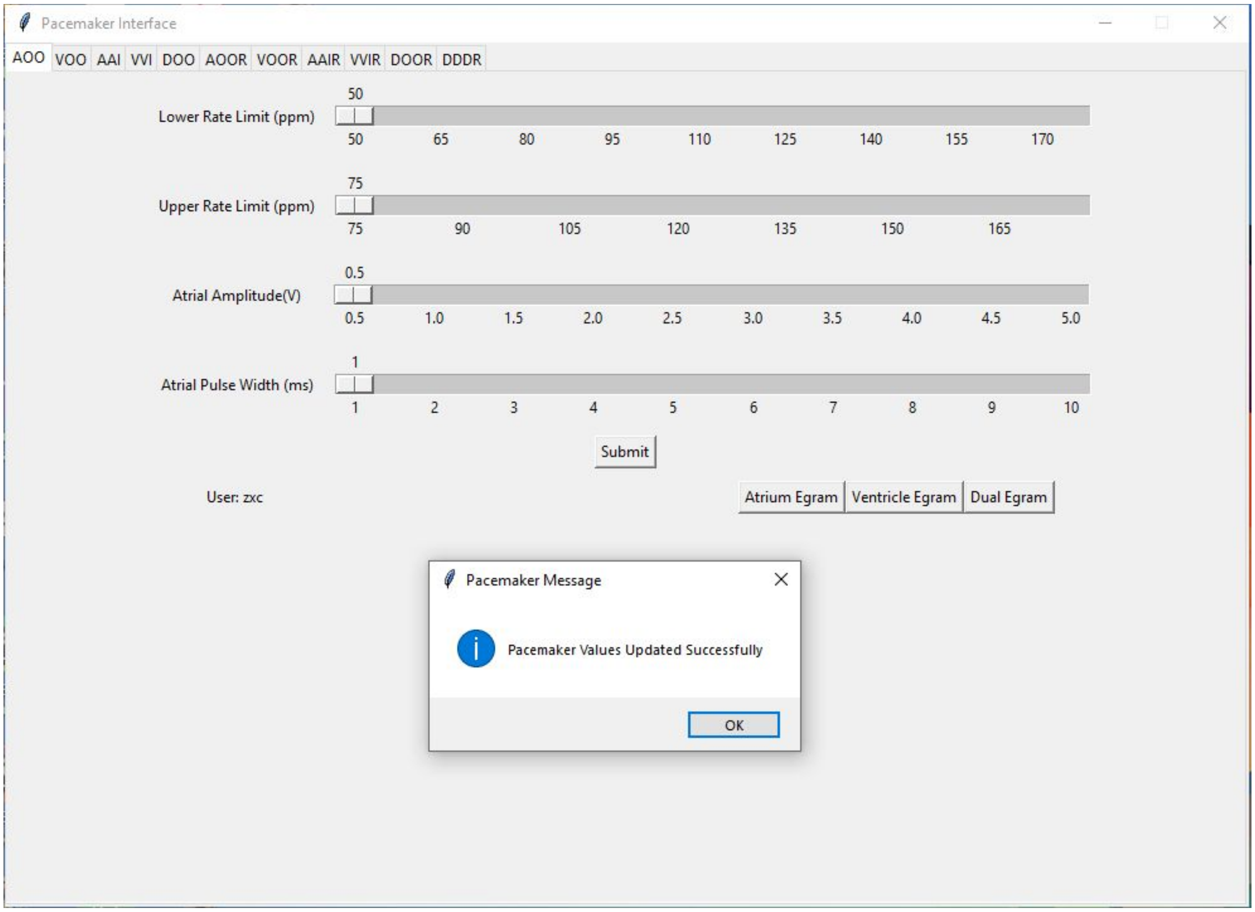


4.3 Program Interface:

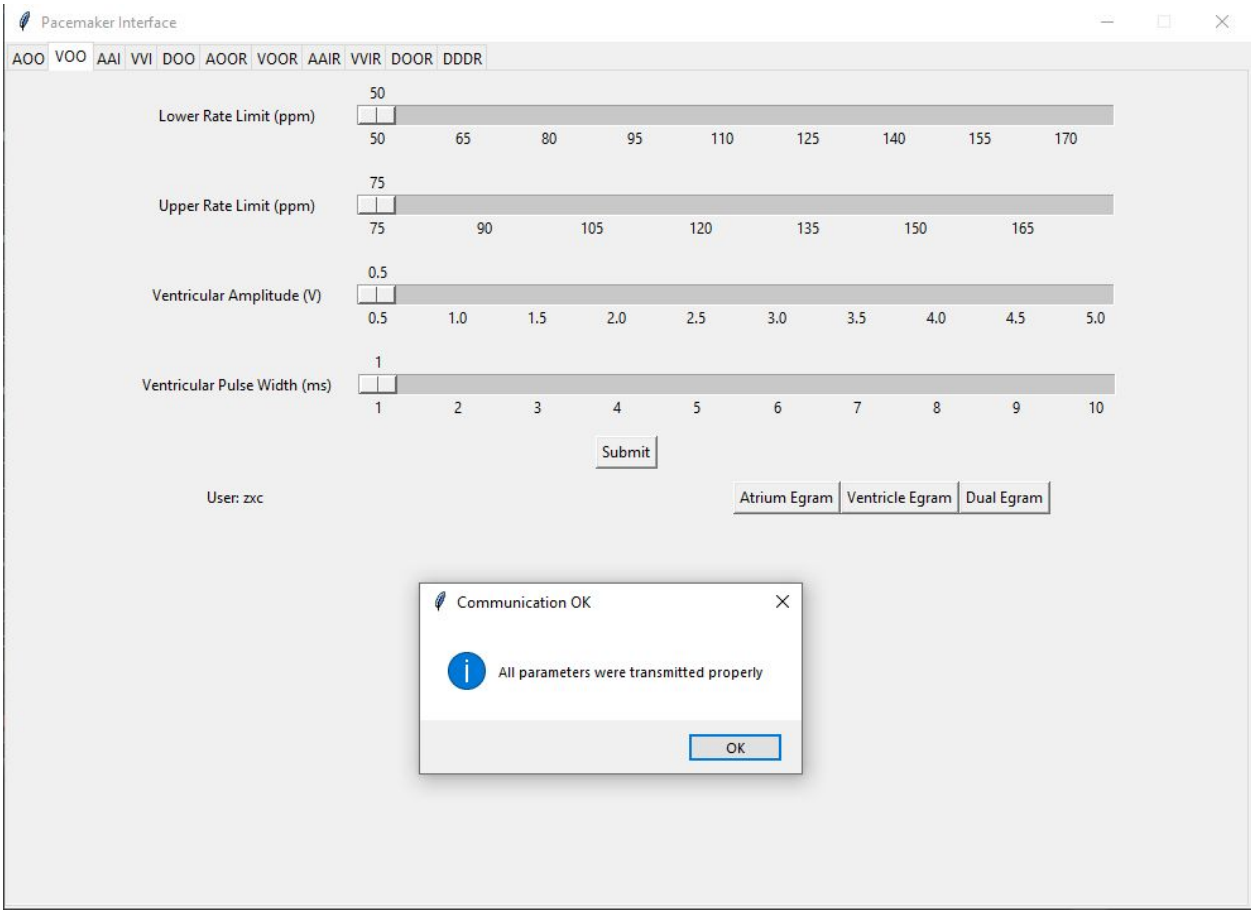
Proper Saves and Transmissions
We will need to test the submitting of parameters for each of the modes. Here we submit good values for all of them. The values transmit and save properly, as expected of them.
AOO

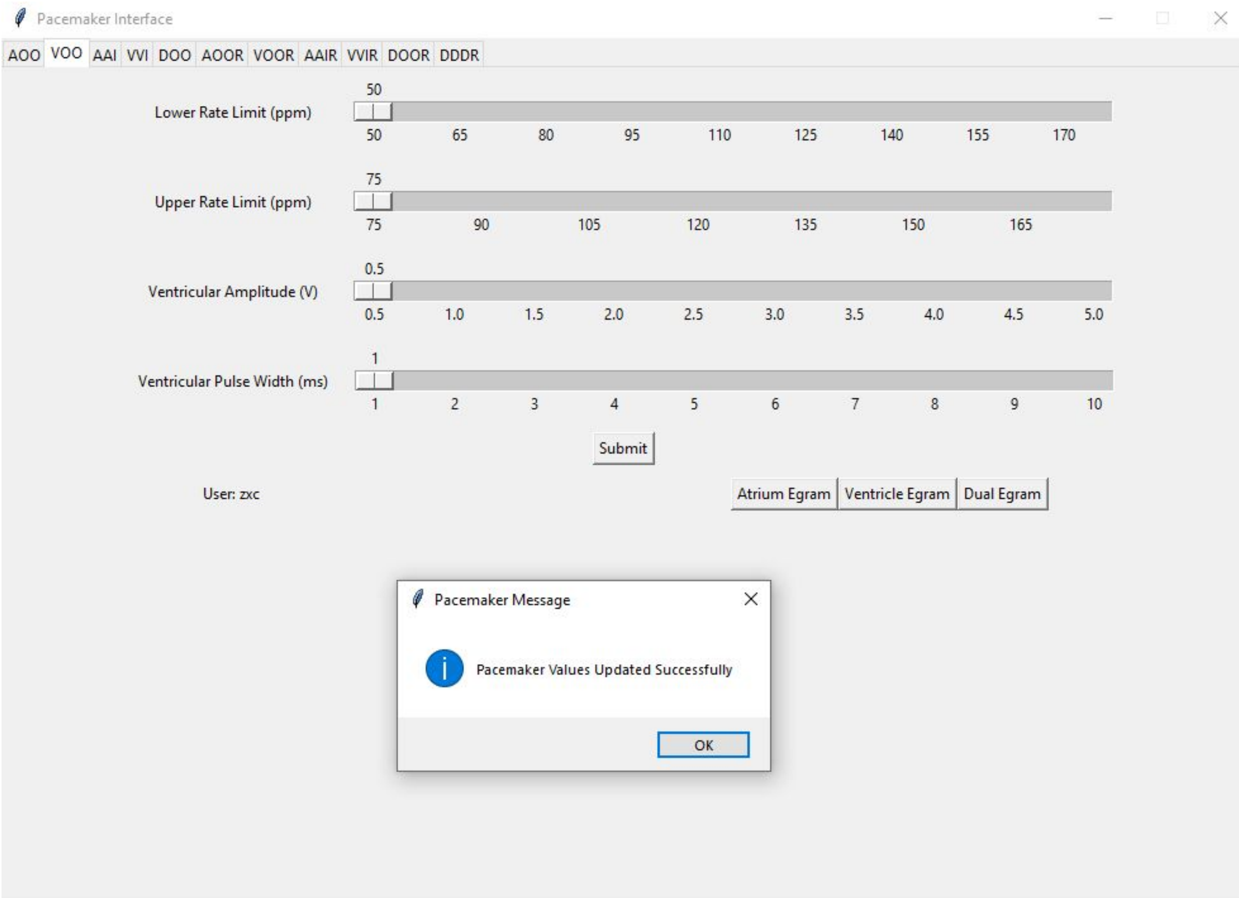

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Atrial Amplitude(V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram


Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Atrial Amplitude(V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Communication OK
All parameters were transmitted properly
OK

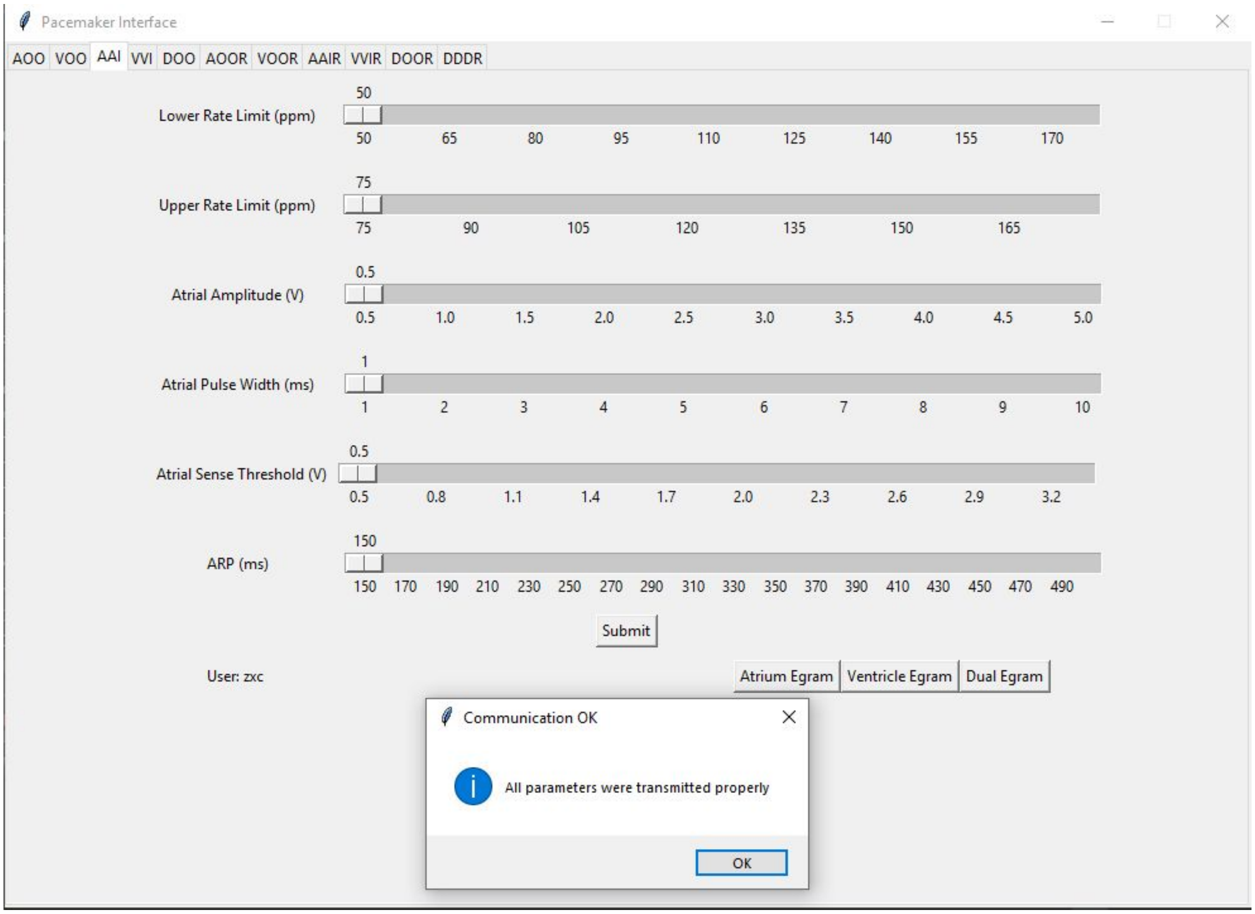

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Atrial Amplitude(V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Pacemaker Message
Pacemaker Values Updated Successfully
OK

VOO

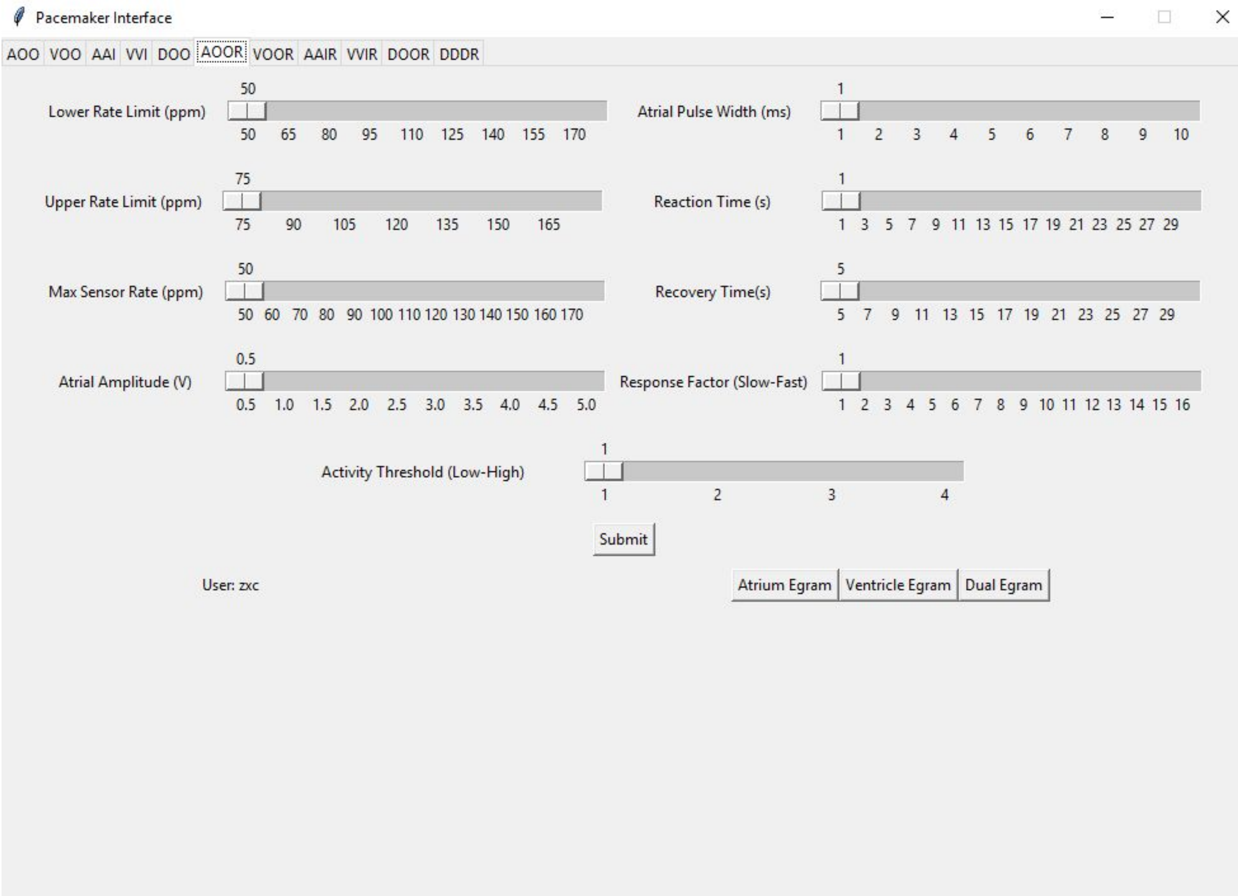

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Ventricular Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Ventricular Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Communication OK
All parameters were transmitted properly
OK

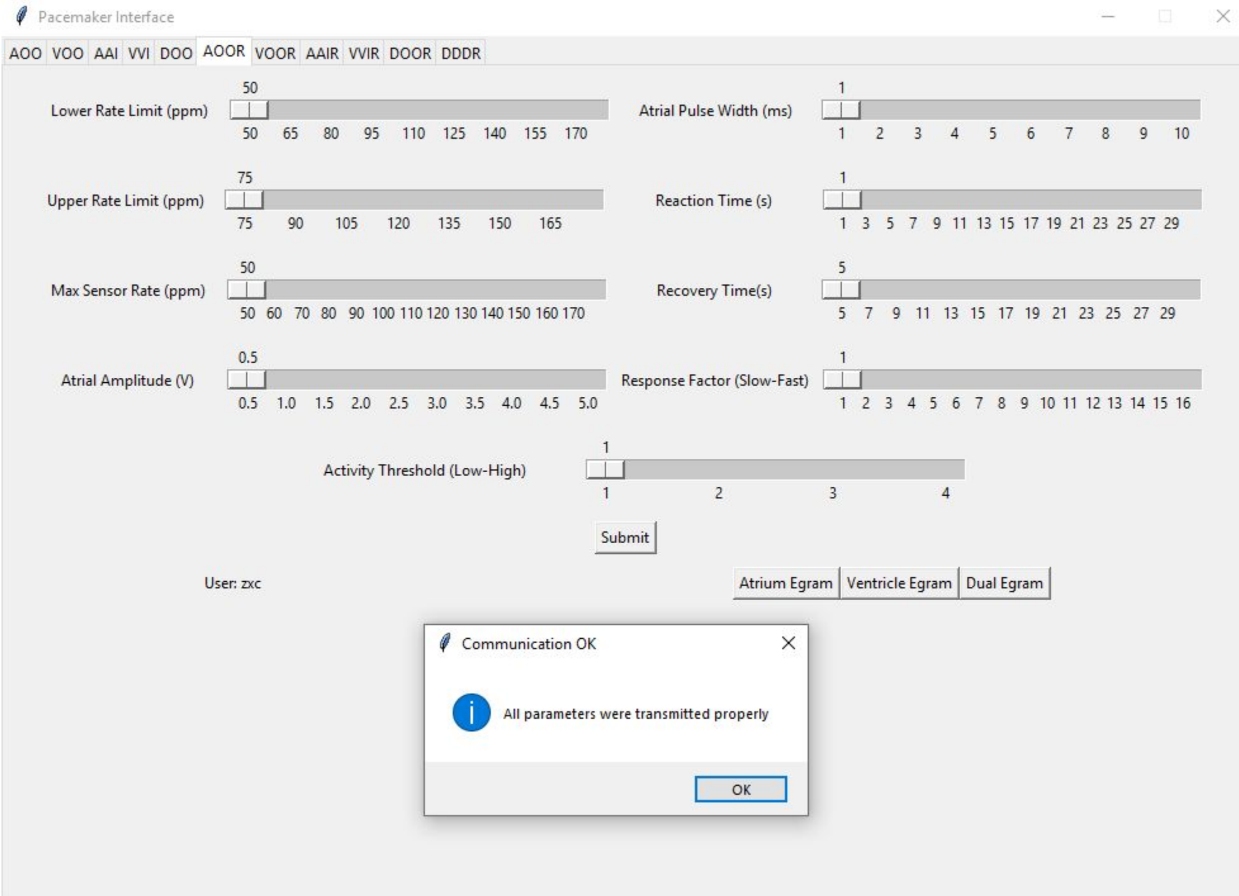

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Ventricular Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Ventricular Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Pacemaker Message
Pacemaker Values Updated Successfully
OK

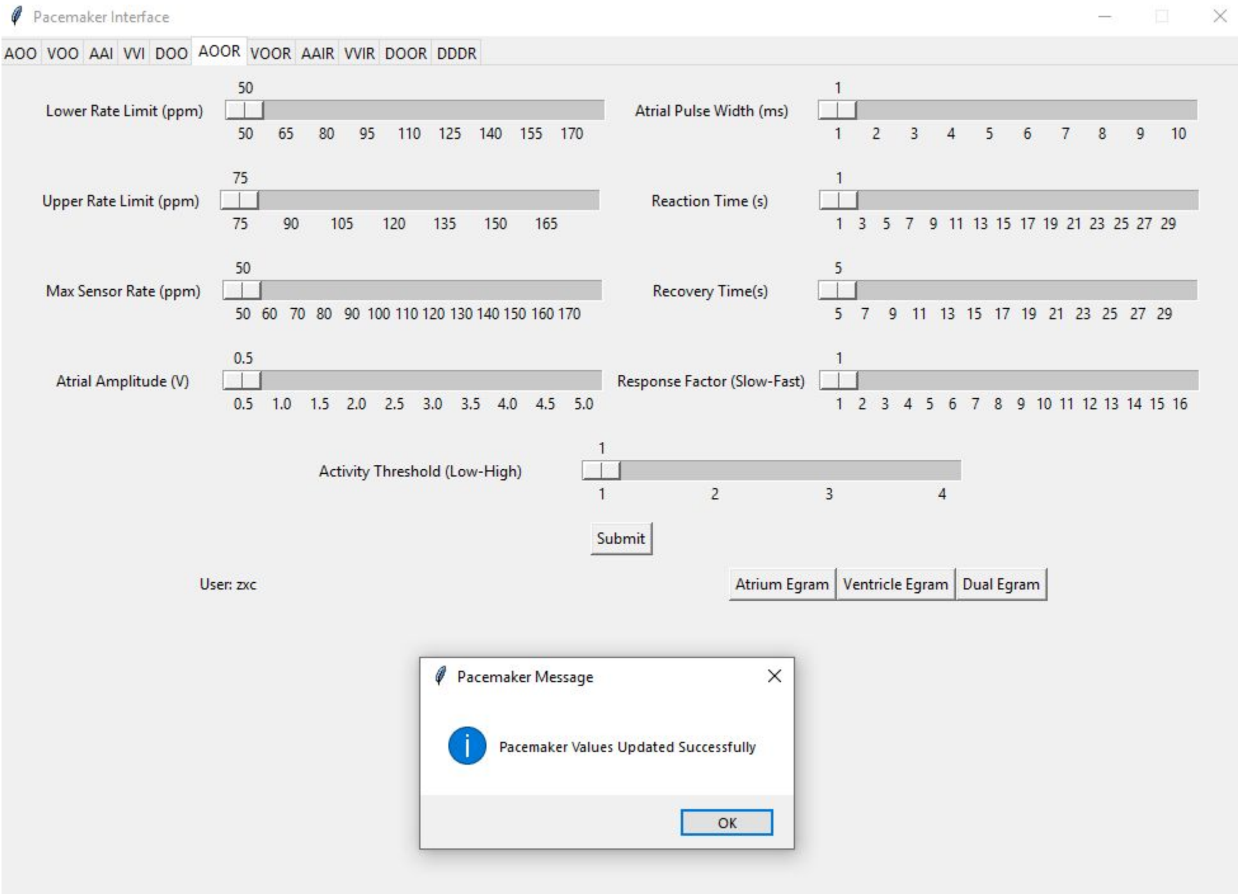
AAI


Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Atrial Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Atrial Sense Threshold (V)
0.5
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
ARP (ms)
150
150 170 190 210 230 250 270 290 310 330 350 370 390 410 430 450 470 490
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Communication OK
All parameters were transmitted properly
OK

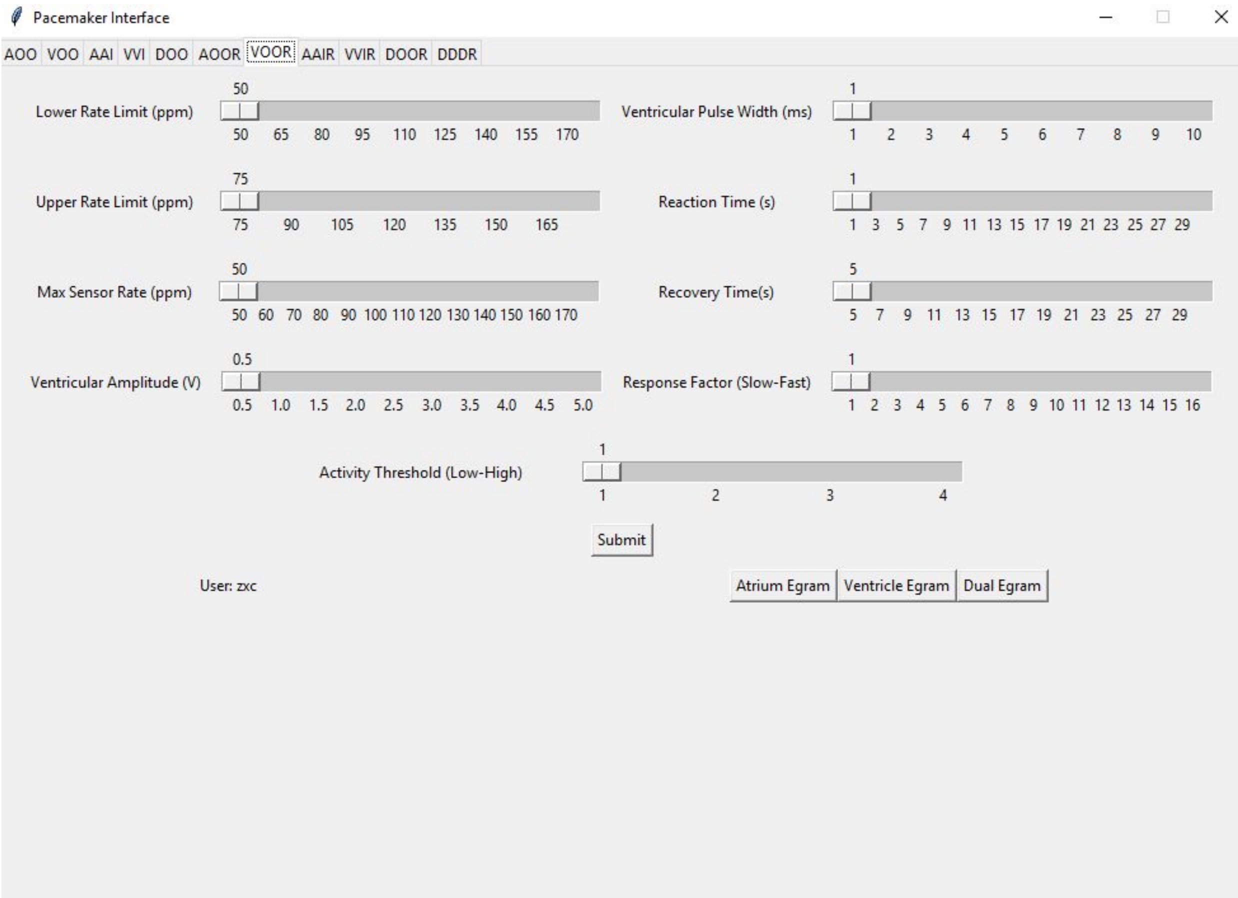
AOOR

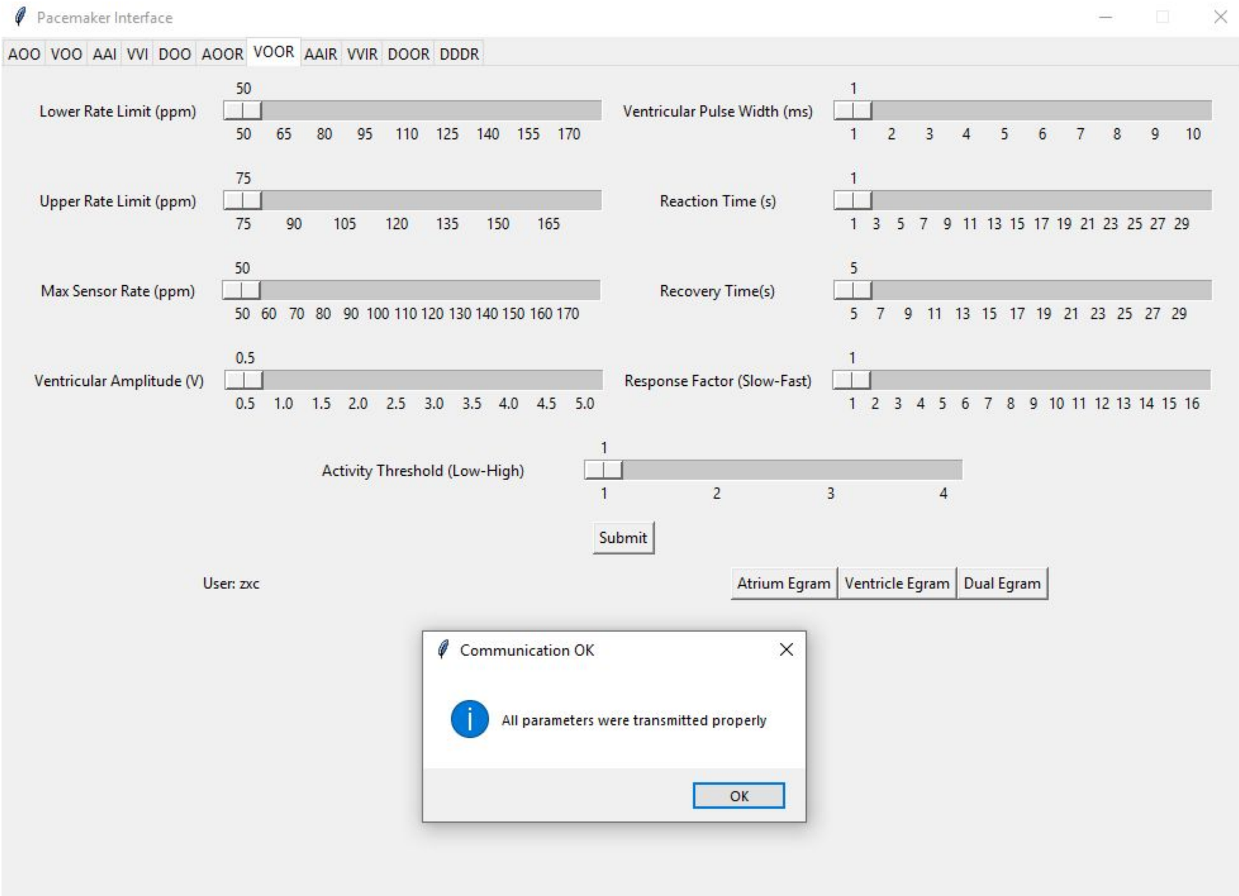

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
Upper Rate Limit (ppm)
Max Sensor Rate (ppm)
Atrial Amplitude (V)
Atrial Pulse Width (ms)
Reaction Time (s)
Recovery Time(s)
Response Factor (Slow-Fast)
Activity Threshold (Low-High)
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram

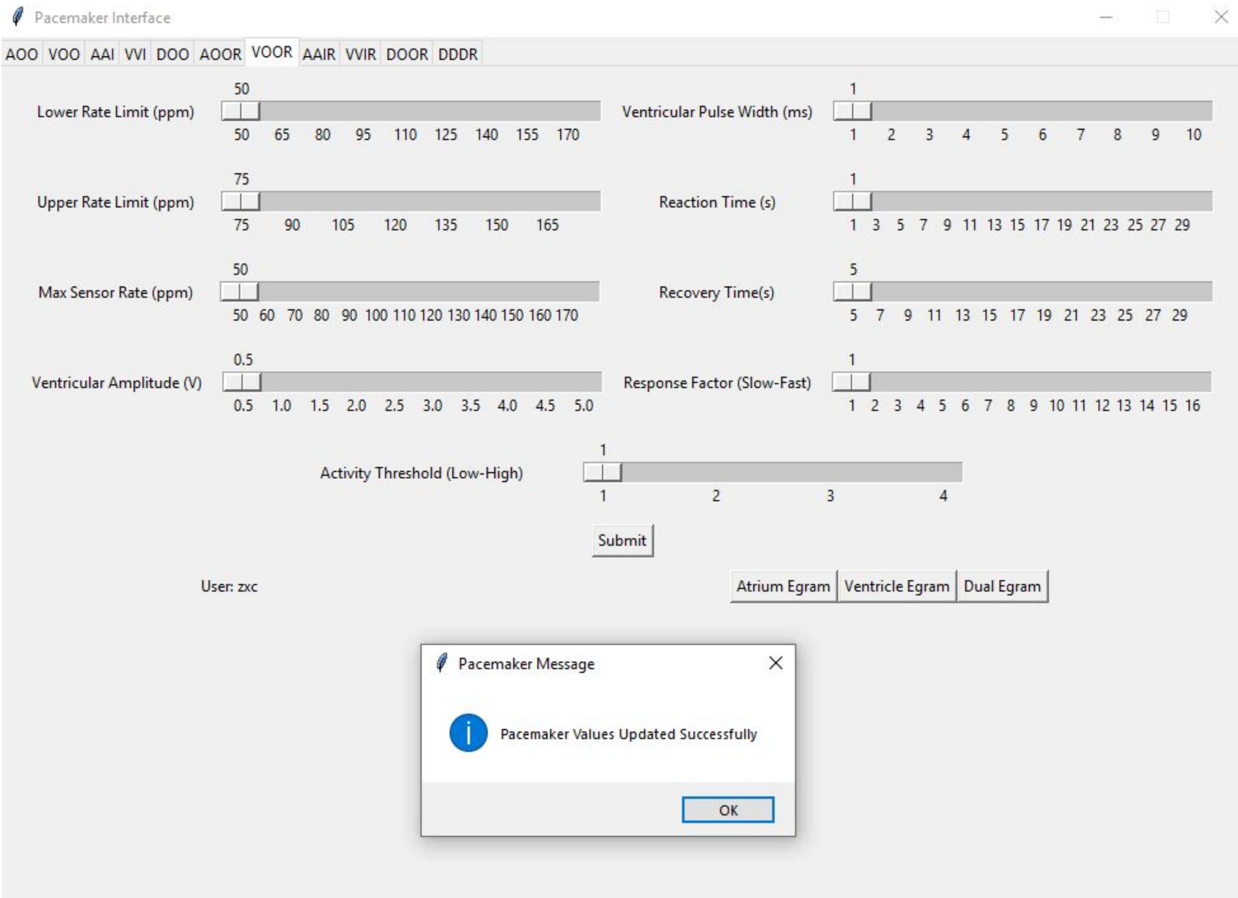

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Reaction Time (s)
1
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Atrial Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Communication OK
All parameters were transmitted properly
OK


Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
Atrial Pulse Width (ms)
Upper Rate Limit (ppm)
Reaction Time (s)
Max Sensor Rate (ppm)
Recovery Time(s)
Atrial Amplitude (V)
Response Factor (Slow-Fast)
Activity Threshold (Low-High)
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Pacemaker Message
Pacemaker Values Updated Successfully
OK

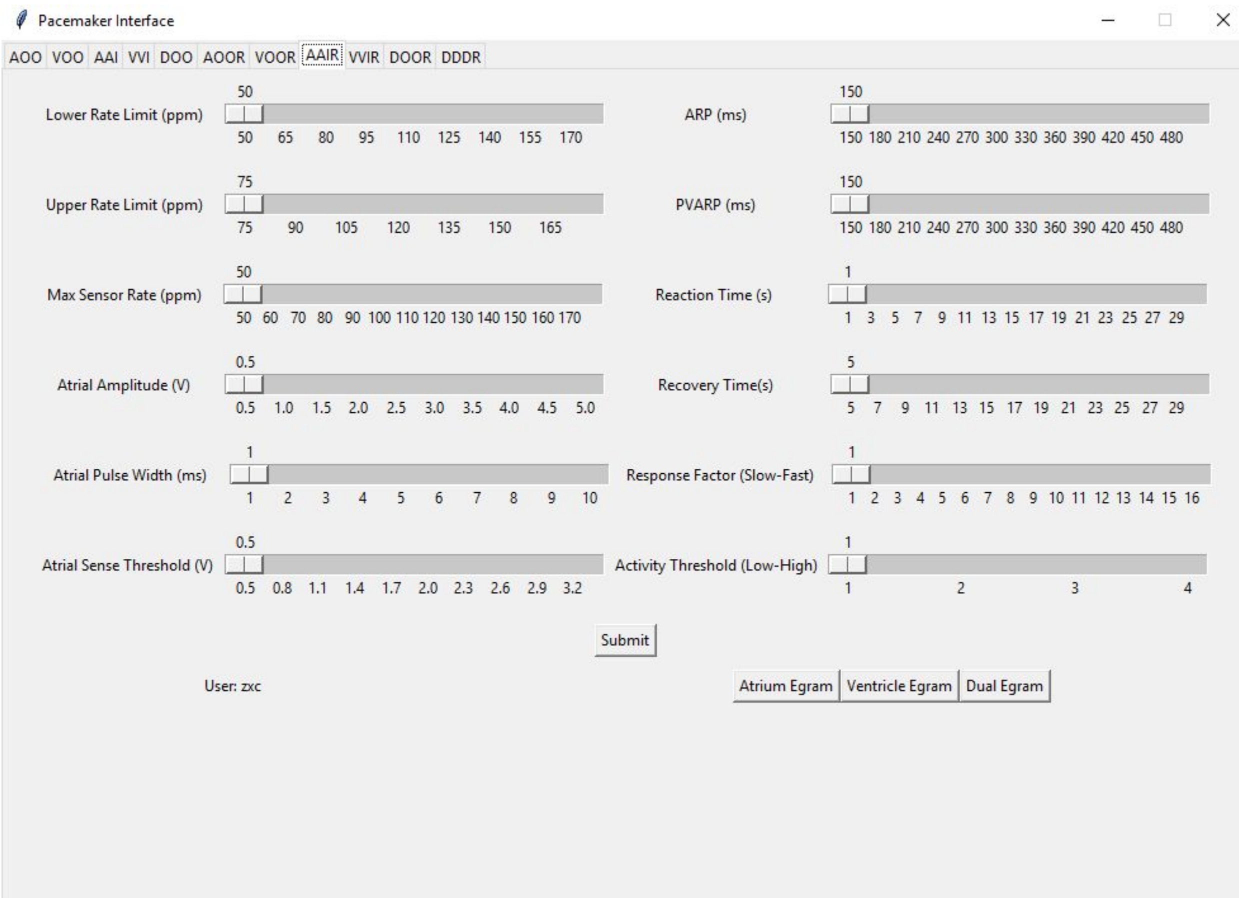
VOOR

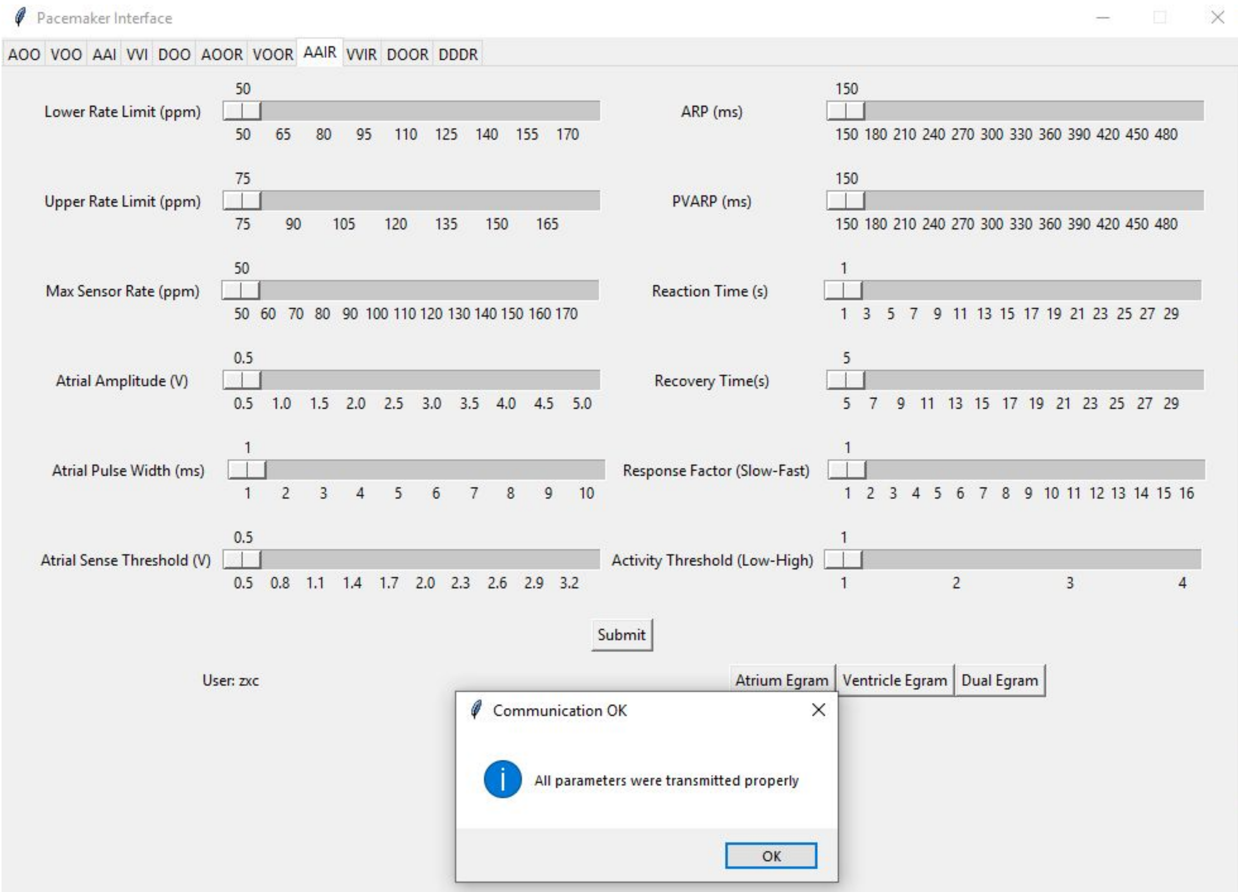

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Ventricular Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Reaction Time (s)
1
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Ventricular Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram

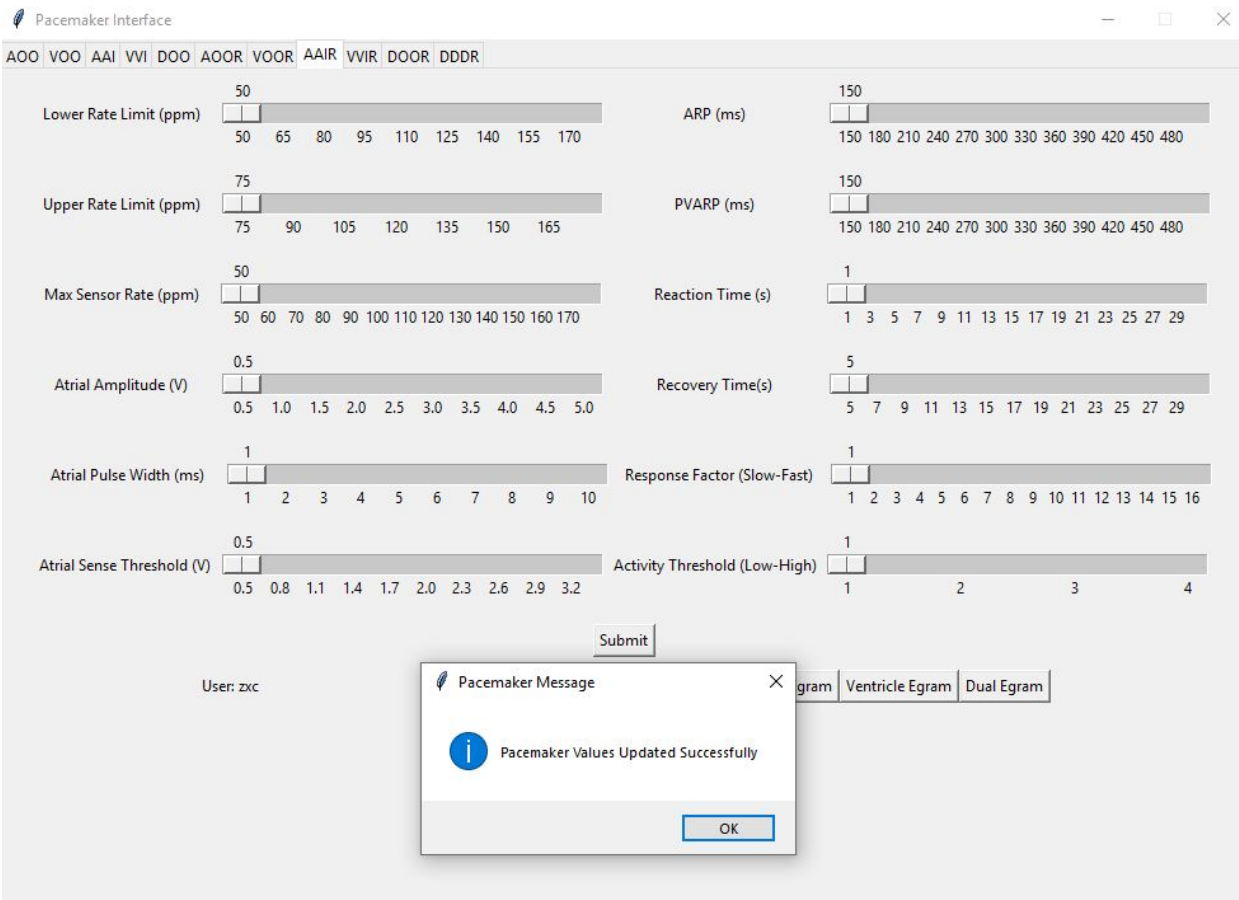

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
Upper Rate Limit (ppm)
Max Sensor Rate (ppm)
Ventricular Amplitude (V)
Ventricular Pulse Width (ms)
Reaction Time (s)
Recovery Time(s)
Response Factor (Slow-Fast)
Activity Threshold (Low-High)
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Communication OK
All parameters were transmitted properly
OK


Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Ventricular Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Reaction Time (s)
1
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Ventricular Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Pacemaker Message
Pacemaker Values Updated Successfully
OK

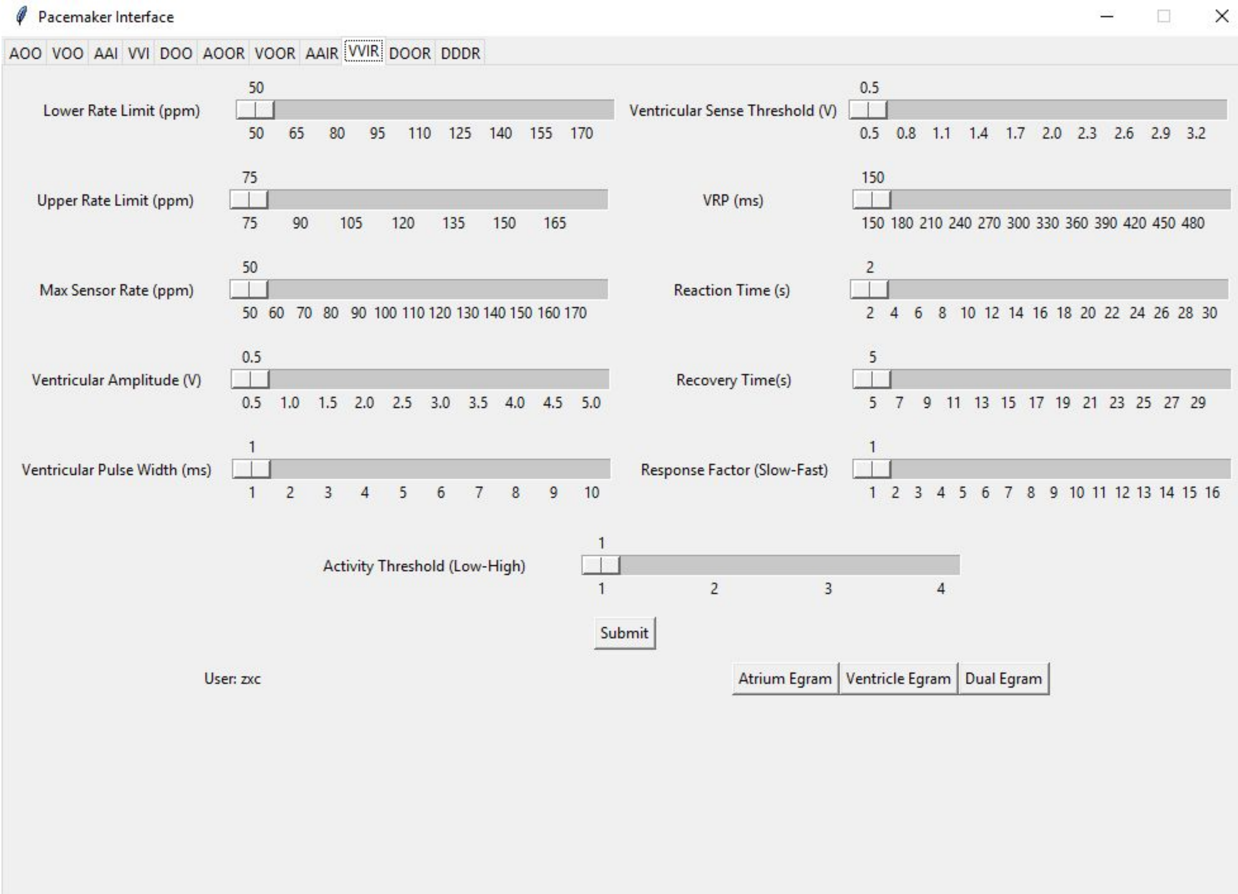
AAIR

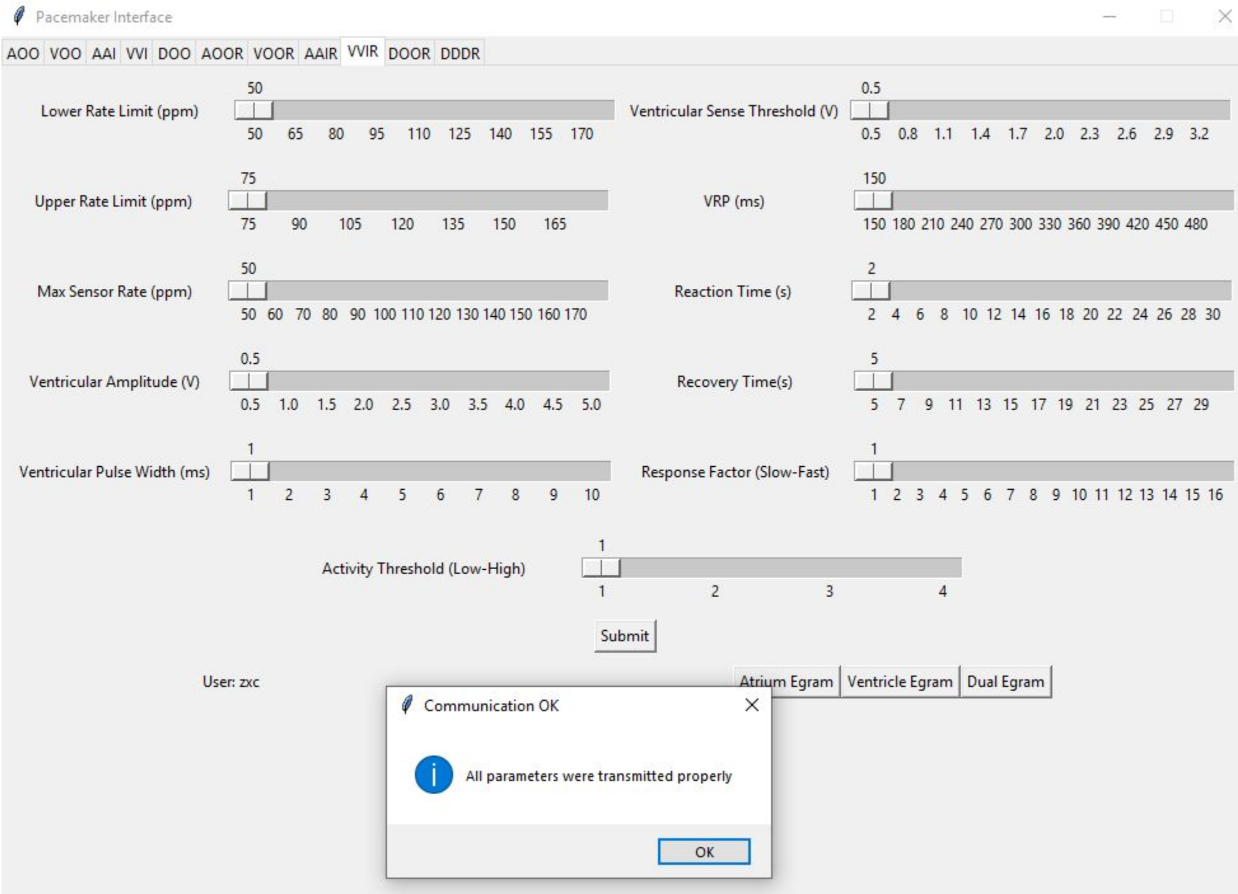

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
ARP (ms)
150
150 180 210 240 270 300 330 360 390 420 450 480
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
PVARP (ms)
150
150 180 210 240 270 300 330 360 390 420 450 480
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Reaction Time (s)
1
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Atrial Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Atrial Sense Threshold (V)
0.5
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram

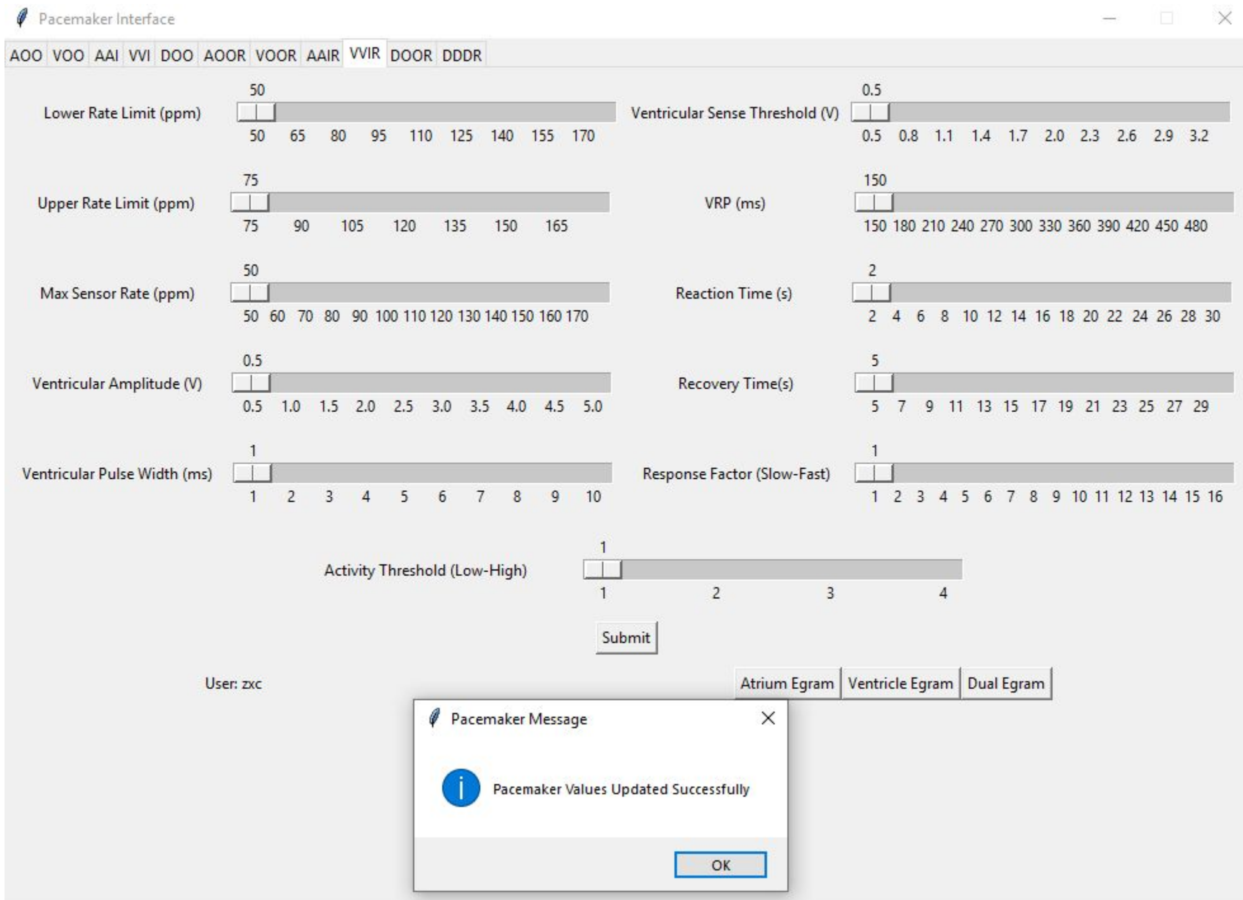

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
ARP (ms)
150
150 180 210 240 270 300 330 360 390 420 450 480
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
PVARP (ms)
150
150 180 210 240 270 300 330 360 390 420 450 480
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Reaction Time (s)
1
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Atrial Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Atrial Sense Threshold (V)
0.5
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Communication OK
All parameters were transmitted properly
OK


Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Atrial Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Atrial Sense Threshold (V)
0.5
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
ARP (ms)
150
150 180 210 240 270 300 330 360 390 420 450 480
PVARP (ms)
150
150 180 210 240 270 300 330 360 390 420 450 480
Reaction Time (s)
1
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
gram
Ventricle Egram
Dual Egram
Pacemaker Message
Pacemaker Values Updated Successfully
OK

VVIR

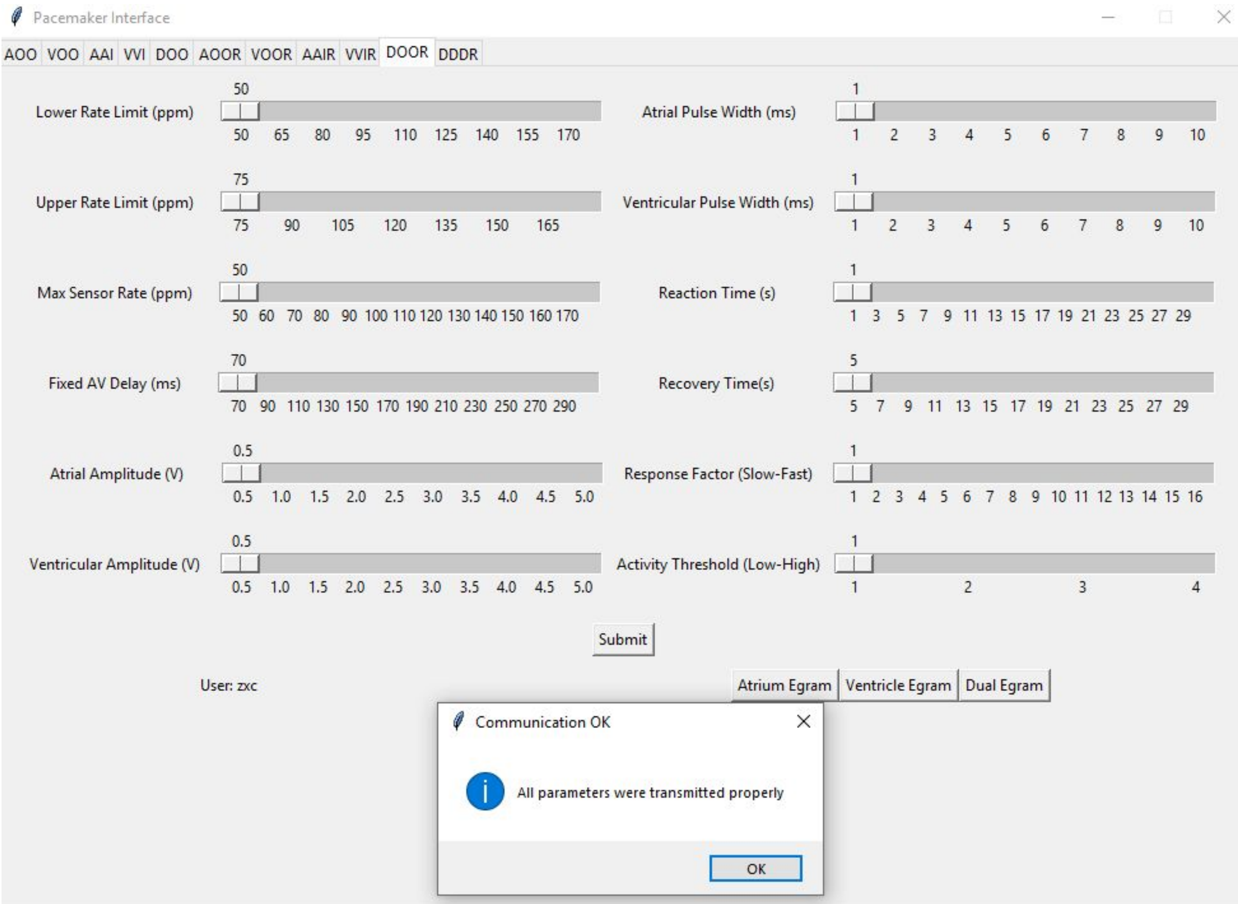

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50 65 80 95 110 125 140 155 170
Ventricular Sense Threshold (V)
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
Upper Rate Limit (ppm)
75 90 105 120 135 150 165
VRP (ms)
150 180 210 240 270 300 330 360 390 420 450 480
Max Sensor Rate (ppm)
50 60 70 80 90 100 110 120 130 140 150 160 170
Reaction Time (s)
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
Ventricular Amplitude (V)
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Recovery Time(s)
5 7 9 11 13 15 17 19 21 23 25 27 29
Ventricular Pulse Width (ms)
1 2 3 4 5 6 7 8 9 10
Response Factor (Slow-Fast)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Activity Threshold (Low-High)
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram

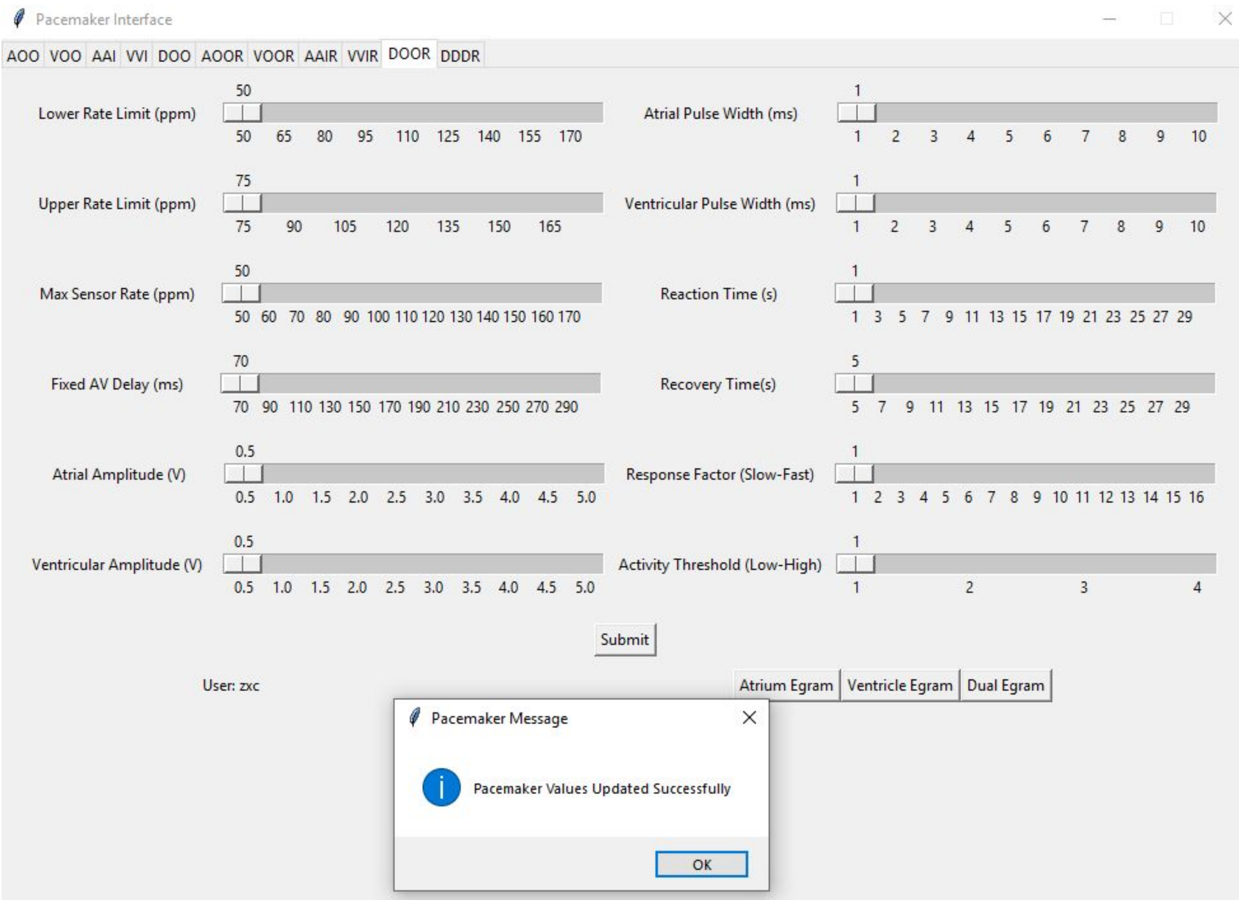

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Ventricular Sense Threshold (V)
0.5
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
VRP (ms)
150
150 180 210 240 270 300 330 360 390 420 450 480
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Reaction Time (s)
2
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
Ventricular Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Ventricular Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Communication OK
All parameters were transmitted properly
OK


Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Ventricular Sense Threshold (V)
0.5
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
VRP (ms)
150
150 180 210 240 270 300 330 360 390 420 450 480
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Reaction Time (s)
2
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
Ventricular Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Ventricular Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Pacemaker Message
Pacemaker Values Updated Successfully
OK

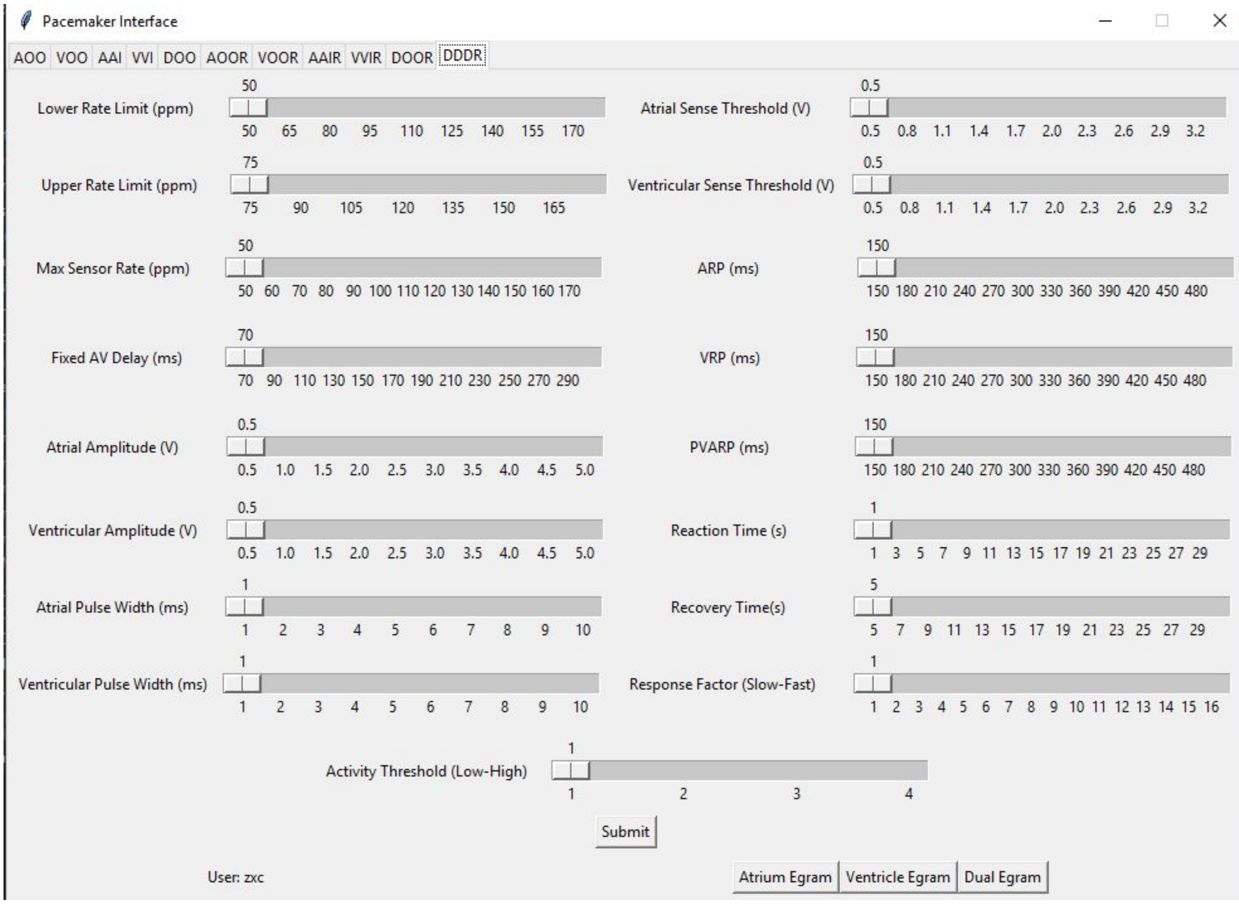
DOOR

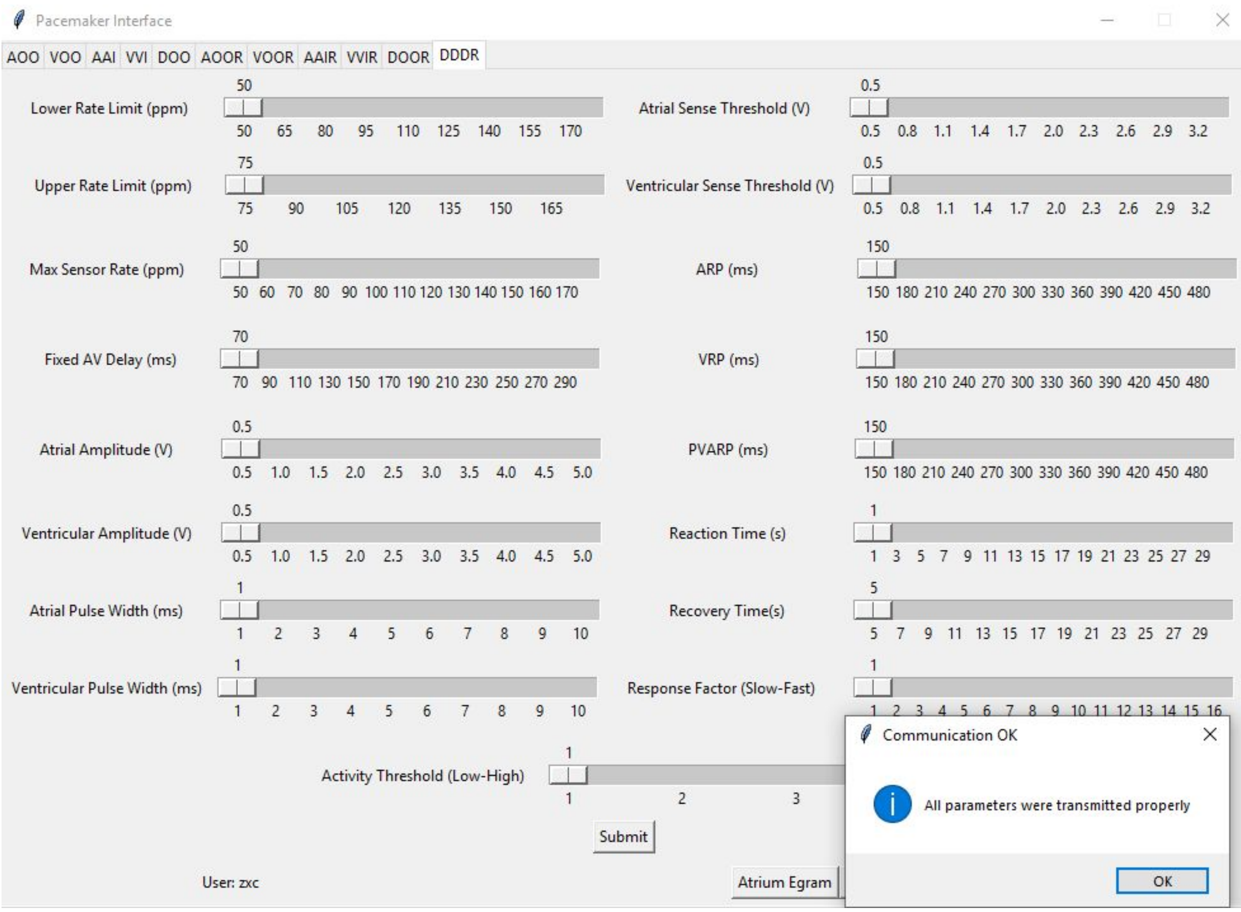

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50
50 65 80 95 110 125 140 155 170
Atrial Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Ventricular Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Max Sensor Rate (ppm)
50
50 60 70 80 90 100 110 120 130 140 150 160 170
Reaction Time (s)
1
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Fixed AV Delay (ms)
70
70 90 110 130 150 170 190 210 230 250 270 290
Recovery Time(s)
5
5 7 9 11 13 15 17 19 21 23 25 27 29
Atrial Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Response Factor (Slow-Fast)
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Ventricular Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Activity Threshold (Low-High)
1
1 2 3 4
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram

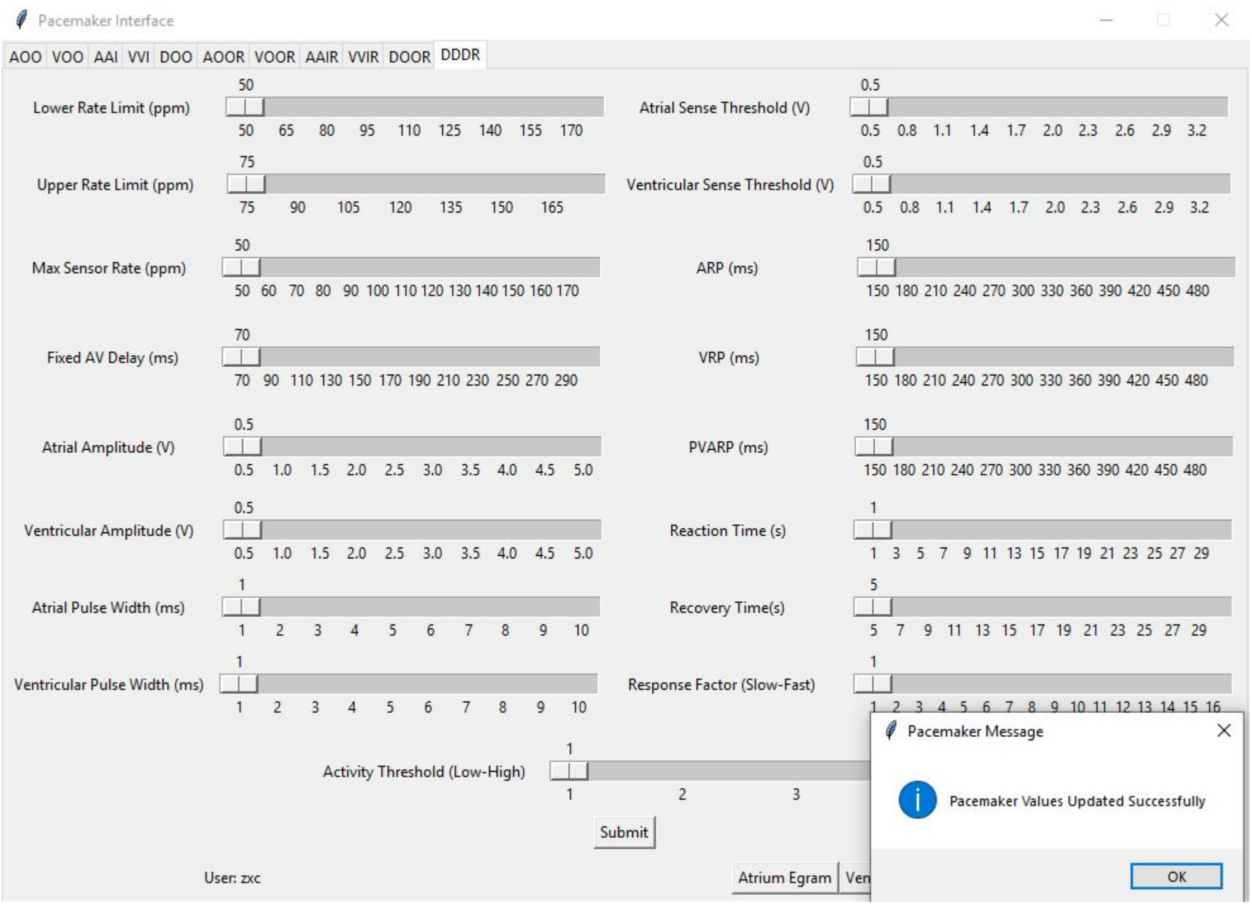

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
Upper Rate Limit (ppm)
Max Sensor Rate (ppm)
Fixed AV Delay (ms)
Atrial Amplitude (V)
Ventricular Amplitude (V)
Atrial Pulse Width (ms)
Ventricular Pulse Width (ms)
Reaction Time (s)
Recovery Time(s)
Response Factor (Slow-Fast)
Activity Threshold (Low-High)
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Communication OK
All parameters were transmitted properly
OK


Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
Upper Rate Limit (ppm)
Max Sensor Rate (ppm)
Fixed AV Delay (ms)
Atrial Amplitude (V)
Ventricular Amplitude (V)
Atrial Pulse Width (ms)
Ventricular Pulse Width (ms)
Reaction Time (s)
Recovery Time(s)
Response Factor (Slow-Fast)
Activity Threshold (Low-High)
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Pacemaker Message
Pacemaker Values Updated Successfully
OK

DDDR


Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
Upper Rate Limit (ppm)
Max Sensor Rate (ppm)
Fixed AV Delay (ms)
Atrial Amplitude (V)
Ventricular Amplitude (V)
Atrial Pulse Width (ms)
Ventricular Pulse Width (ms)
Atrial Sense Threshold (V)
Ventricular Sense Threshold (V)
ARP (ms)
VRP (ms)
PVARP (ms)
Reaction Time (s)
Recovery Time(s)
Response Factor (Slow-Fast)
Activity Threshold (Low-High)
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram

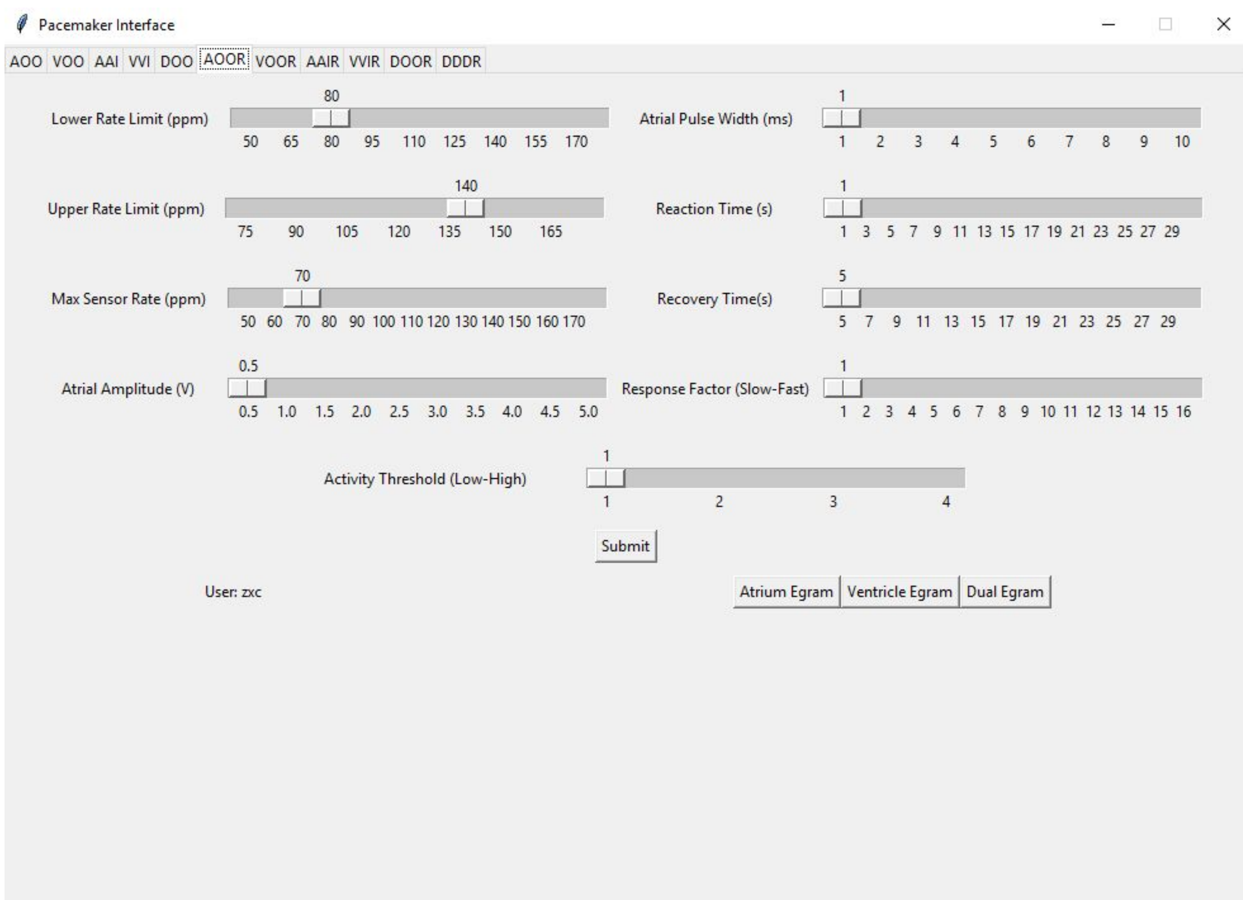

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75 90 105 120 135 150 165
Max Sensor Rate (ppm)
50 60 70 80 90 100 110 120 130 140 150 160 170
Fixed AV Delay (ms)
70 90 110 130 150 170 190 210 230 250 270 290
Atrial Amplitude (V)
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Ventricular Amplitude (V)
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Atrial Pulse Width (ms)
1 2 3 4 5 6 7 8 9 10
Ventricular Pulse Width (ms)
1 2 3 4 5 6 7 8 9 10
Atrial Sense Threshold (V)
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
Ventricular Sense Threshold (V)
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
ARP (ms)
150 180 210 240 270 300 330 360 390 420 450 480
VRP (ms)
150 180 210 240 270 300 330 360 390 420 450 480
PVARP (ms)
150 180 210 240 270 300 330 360 390 420 450 480
Reaction Time (s)
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Recovery Time(s)
5 7 9 11 13 15 17 19 21 23 25 27 29
Response Factor (Slow-Fast)
Activity Threshold (Low-High)
1 2 3
Submit
User: zxc
Atrium Egram
Communication OK
All parameters were transmitted properly
OK

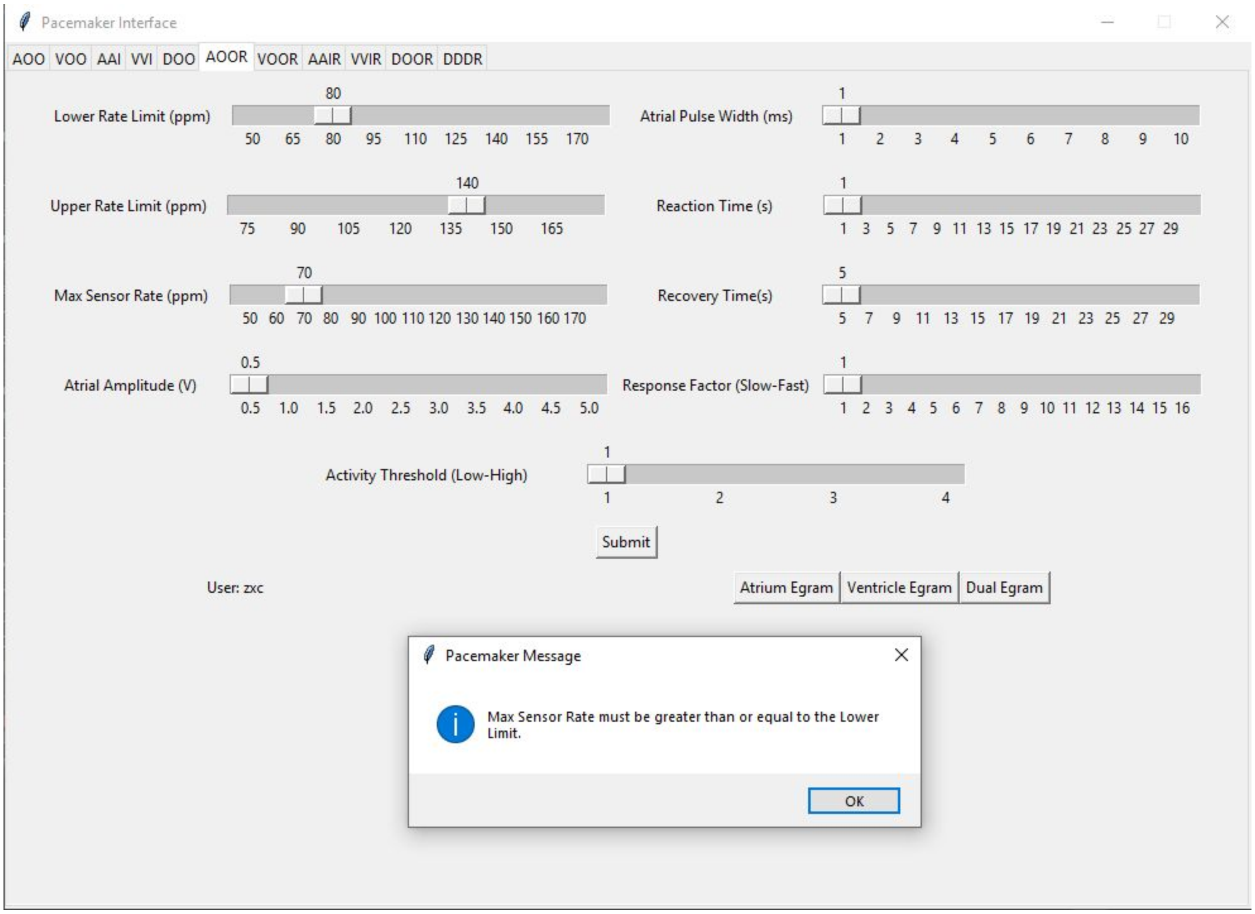

Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75 90 105 120 135 150 165
Max Sensor Rate (ppm)
50 60 70 80 90 100 110 120 130 140 150 160 170
Fixed AV Delay (ms)
70 90 110 130 150 170 190 210 230 250 270 290
Atrial Amplitude (V)
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Ventricular Amplitude (V)
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Atrial Pulse Width (ms)
1 2 3 4 5 6 7 8 9 10
Ventricular Pulse Width (ms)
1 2 3 4 5 6 7 8 9 10
Atrial Sense Threshold (V)
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
Ventricular Sense Threshold (V)
0.5 0.8 1.1 1.4 1.7 2.0 2.3 2.6 2.9 3.2
ARP (ms)
150 180 210 240 270 300 330 360 390 420 450 480
VRP (ms)
150 180 210 240 270 300 330 360 390 420 450 480
PVARP (ms)
150 180 210 240 270 300 330 360 390 420 450 480
Reaction Time (s)
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Recovery Time(s)
5 7 9 11 13 15 17 19 21 23 25 27 29
Response Factor (Slow-Fast)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Activity Threshold (Low-High)
1 2 3
Submit
User: zxc
Atrium Egram
Pacemaker Message
Pacemaker Values Updated Successfully
OK

4.4 Rate-Adaptive Modes:

The rate adaptive modes add some new parameters to the DCM interface. As a result, there are more possible test cases.

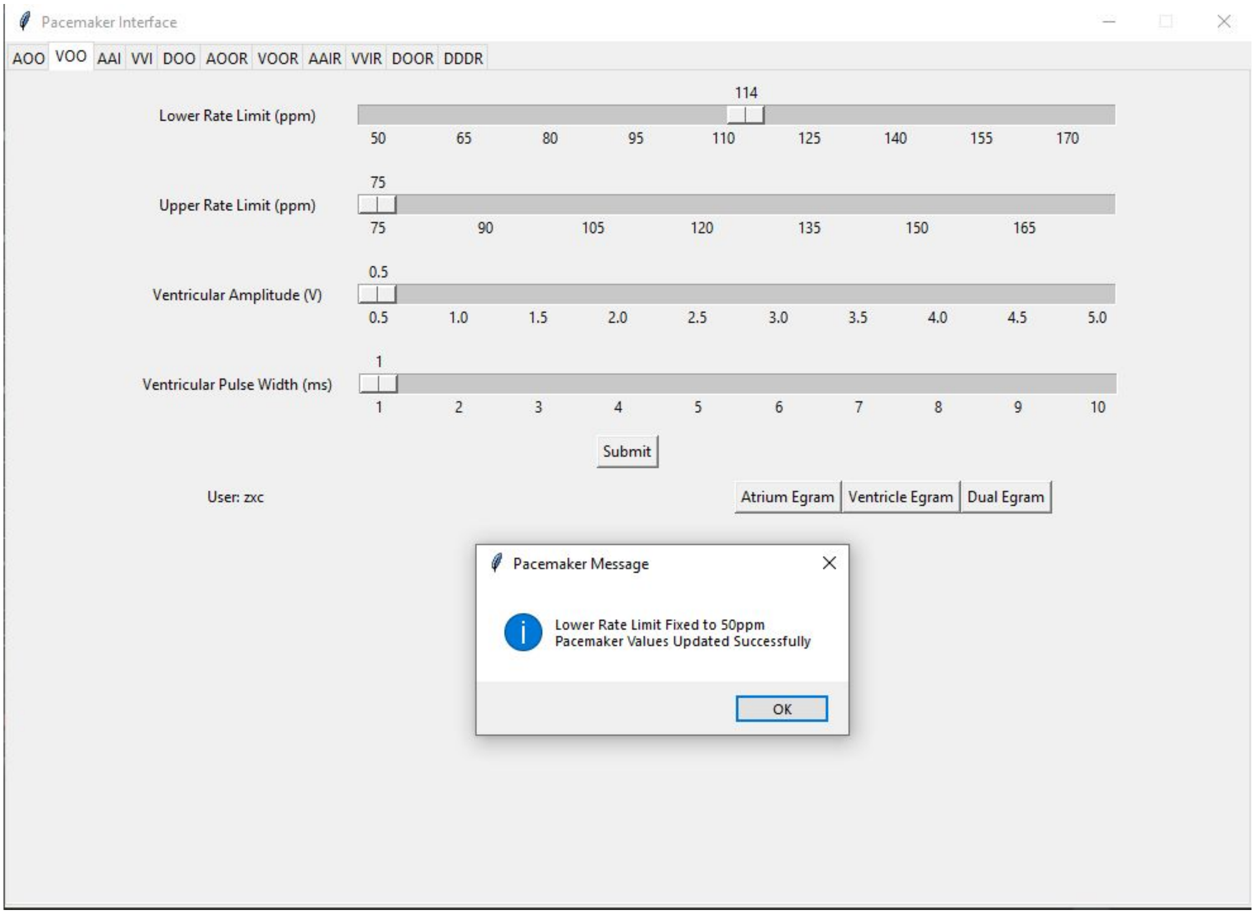
For any rate adaptive modes, the Max Sensor Rate should be greater than or equal to the Lower Rate Limit. In the case that it is not, a popup window that notifies the user and then prompts them to retry. In the below test case, we have such a situation, and when the values are submitted it performs as expected.





4.5 Improper Transmission/Other Issues

Some other issues can arise for each of the modes. For instance, if the Lower Rate Limit slider is above the Upper Rate Limit, there is a special case in the code to catch this. It sets the Lower Rate Limit equal to 50 ppm. Here is this case in action:



Pacemaker Interface
AOO VOO AAI VVI DOO AOOR VOOR AAIR VVIR DOOR DDDR
Lower Rate Limit (ppm)
114
50 65 80 95 110 125 140 155 170
Upper Rate Limit (ppm)
75
75 90 105 120 135 150 165
Ventricular Amplitude (V)
0.5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Ventricular Pulse Width (ms)
1
1 2 3 4 5 6 7 8 9 10
Submit
User: zxc
Atrium Egram
Ventricle Egram
Dual Egram
Pacemaker Message
Lower Rate Limit Fixed to 50ppm
Pacemaker Values Updated Successfully
OK