
Pacemaker Device Controller-Monitor (DCM)

10/30/2021 Revision 1

This software is written in Python 3.8.8 and uses multiple open source libraries.

1.0 Modules

Module gui2.py

This module contains GUI functions and global members that build the front end of the DCM user program. gui2 uses tkinter to design the layout and different windows of the DCM user program. Moreover it uses a theme wrapper that applies a different theme than the default tkinter colorway which enhances the user experience.

This module only executes one function which is called initial_screen() which instantiates the welcome screen. which calls other functions described in the table below to accomplish requirements for Assignment 1 as of 10/30/2021

Global Variable (State Variable)	Description
login_status	tkk.StringVar for a label which displays messages if there are problems with signing in.
messags_status	tkk.StringVar for a label which displays messages if there are problems with creating a new account.
num_users	integer which keeps track of how many users there are in the database
LRL	All possible values of Lower Rate Limit Parameter.
pulse_width_values	All possible values of pulse width values.
amp_values	All possible values of amplitude values.

Function	Level of Scope	Black-Box Behavior	Internal Behavior
def programmable()	Public	Takes no input, will return a 1000x700 window that has 4 different tabs (pacing mode). Each tab displays spinbox and label depending on the programmable parameter	This function gets called when a login is successful, this is determined when the password user types is equivalent to the password stored for the specific username. This function will also be called after a user successfully registers a new account. Creates a 1000x700 window with 4 different tabs (labeled as "AOO", "VOO", "AAI", "VVI") using the Frame widget. Each tab consists of different programmable parameters depending on the pacing mode selected from tab. Each programmable parameter can be incremented and decremented using the Spinbox widget with a certain increment, upper limit and lower limit depending on the programmable parameter. Pulse Width and Lower Rate Limit programmable parameters will call the list pulse_width_values and LRL_values respectively which contain the values of those specific parameters.
def set_communication(simulinkCode)	Public	Inputs a byte of code and returns if this code is recognizable or not.	Checks serial communication port at 'COMX' and reads a byte of code to recognizes unique pacemaker controllers.

Function	Level of Scope	Black-Box Behavior	Internal Behavior
def getdetails(username, password, retypedpassword)	Public	returns 1 if successful and 0 if registration fails.	gets called when the 'New Account' Button is pressed. Open file 'User_data.txt' to first check if username already exists within the file. If it does have that name the messags_status global variable updates to the corresponding message. If it does not exist it checks if the user limit num_users global variable is over 10 and updates messags_status accordingly. Which then finally writes to the file the user data if password and retypedpassword match. Closes file at the end.
def checkdetails()	Public	returns 1 if successfully logs in and 0 if wrong username or password.	gets called when the 'Log in' Button is pressed. Open file 'User_data.txt' to first check if username exists within the file. If it does not have that name the login_status global variable updates to the corresponding message. Then it checks if the password matches the username that exists and will update the login_status global variable to the corresponding message (whether the password is correct or not)

Function	Level of Scope	Black-Box Behavior	Internal Behavior
def register_new_user()	Public	Takes no input and Returns three new entries and a button to register. Able to transfer to Pacing Mode Window	This function is called when the user would like to create a new account and is called from the initial_screen() function's Button 'b_new_acc' this function creates three entries, username password retyped password and a 'Register' Button to create a new account. This window also contains multiple message labels for login and register status if any errors in logging in or registering arises.
def intial_screen()	Public	Takes no input, will return a 1000x700 window that has the widgets defined in the description. Calls other functions based on button input.	The only function that is called by this module and is a bridge to other functions/features for this module. It initially Instantiates two entries, username and password and a login button below these entries to access a user's programmable parameters. It also instantiates a 'New Account Button' which, if pressed, calls register_new_user() function.

UI Button	Called Function
b_register	getdetails(dirname.get(), dirpass.get(), dirname3.get())
b_login	checkdetails(entry1.get(), entry2.get())
b_new_acc	register_new_user()
b_comm	set_communication(b_comm)

Programable Parameter	Object
Lower Rate Limit	ttk.Spinbox
Upper Rate Limit	ttk.Spinbox
Atrial Amplitude	ttk.Spinbox
Ventricular Amplitude	ttk.Spinbox
Atrial Pulse Width	ttk.Spinbox
Ventricular Pulse Width	ttk.Spinbox
ARP	ttk.Spinbox
VRP	ttk.Spinbox

- Module Secret: LRL array, pulse_width_values and amp_values which stores the possible allowed values for the entry to accept.
- Module Secret: We applied a theme we found on Github [here](#) which writes over the default tkinter theme.

2.0 Testing

Welcome Window (Welcome Page)

Specific Trial	Test Case	Expected outcome	Experimental Outcome
'New Account' Button	1. Clicking 'New Account' Button	Three new entries labelled username password and retyped password appear.	Three new entries labelled username password and retyped password appear.
'Register' Button	<p>1. Input a username that already exists. Password does not matter.</p> <p>2. Input a username that does not exist with a matching password and retypedpassword. (syrrhassan11: 12345)</p> <p>3. Input a username that does not exist with a NOT matching password and retypedpassword. (syrrhassan11: 12345)</p> <p>4. Input a username that contains a space with a matching password and retypedpassword. (syrrhassan: 123456)</p> <p>5. Input a username that does not exist with a matching password and retypedpassword that have a space. (syrrhassan11: 123 45)</p>	<p>1. message_status global variable writes to label that 'Username already exists'</p> <p>2. Writes to file the new account and enters Main Page (programmable parameters page) : syrrhassan11 1234</p> <p>3. message_status global variable writes to label that 'Password and retyped password do not match'</p> <p>4. FAIL. Writes to file the new account and enters Main Page (programmable parameters page) : syrrhassan 123456</p> <p>5. FAIL. Writes to file the new account and enters Main Page (programmable parameters page) : syrrhassan11: 123 45</p>	<p>1. message_status global variable writes to label that 'Username already exists'</p> <p>2. Writes to file the new account and enters Main Page (programmable parameters page) : syrrhassan11 1234</p> <p>3. message_status global variable writes to label that 'Password and retyped password do not match'</p> <p>4. FAIL. Writes to file the new account and enters Main Page (programmable parameters page) : syrrhassan 123456</p> <p>5. FAIL. Writes to file the new account and enters Main Page (programmable parameters page) : syrrhassan11: 123 45</p>
'Sign In' Button	<p>1. Input a valid username. Password is not correct.</p> <p>2. Input a valid username. Password is correct.</p> <p>3. Input an invalid username</p>	<p>1. login_status global variable outputs 'Password is not correct'</p> <p>2. Enters Main Page (Programmable Parameters)</p> <p>3. login_status global variable outputs 'Username is not correct'</p>	<p>1. login_status global variable outputs 'Password is not correct'</p> <p>2. Enters Main Page (Programmable Parameters)</p> <p>3. login_status global variable outputs 'Username is not correct'</p>

Pacing Parameters Window:

Test	Test Case	Expected Outcome	Test Outcome
Open selected pacing mode tab	Click on "AOO" tab	Display Label and Spinbox for "Lower Rate Limit", "Upper Rate Limit", "Atrial Amplitude", "Atrial Pulse Width"	Display Label and Spinbox for "Lower Rate Limit", "Upper Rate Limit", "Atrial Amplitude", "Atrial Pulse Width"
Open selected pacing mode tab	Click on "VOO" tab	Display Label and Spinbox for "Lower Rate Limit", "Upper Rate Limit", "Ventricular Amplitude", "Ventricular Pulse Width"	Display Label and Spinbox for "Lower Rate Limit", "Upper Rate Limit", "Ventricular Amplitude", "Ventricular Pulse Width"
Open selected pacing mode tab	Click on "AAI" tab	Display Label and Spinbox for "Lower Rate Limit", "Upper Rate Limit", "Atrial Amplitude", "Atrial Pulse Width", "Atrial Refractory Period"	Display Label and Spinbox for "Lower Rate Limit", "Upper Rate Limit", "Atrial Amplitude", "Atrial Pulse Width", "Atrial Refractory Period"
Open selected pacing mode tab	Click on "VVI" tab	Display Label and Spinbox for "Lower Rate Limit", "Upper Rate Limit", "Ventricular Amplitude", "Ventricular Pulse Width", "Ventricular Refractory Period"	Display Label and Spinbox for "Lower Rate Limit", "Upper Rate Limit", "Ventricular Amplitude", "Ventricular Pulse Width", "Ventricular Refractory Period"
Change Programable Parameters Values	Click on up arrow to increment/decrement "Lower Rate Limit"	Increments/decrements by 5 when in 30-50 ppm range or 90-175 ppm range. Increments/decrements by 1 in 50-90ppm range.	Increments/decrements by 5 when in 30-50 ppm range or 90-175 ppm range. Increments/decrements by 1 in 50-90ppm range.
Change Programable Parameters Values	Click on up arrow to increment/decrement "Upper Rate Limit"	Increments/decrements by 5 when in 50-175 ppm range	Increments/decrements by 5 when in 50-175 ppm range
Change Programable Parameters Values	Click on up arrow to increment/decrement "Atrial Pulse Width"	Increments/decrements by 0.1ms in 0.1 – 1.9ms	Increments/decrements by 0.1ms in 0.1 – 1.9ms
Change Programable Parameters Values	Click on up arrow to increment/decrement "Ventricular Pulse Width"	Increments/decrements by 0.1ms in 0.1 – 1.9ms range	Increments/decrements by 0.1ms in 0.1 – 1.9ms range
Change Programable Parameters Values	Click on up arrow to increment/decrement "Ventricular Amplitude"	Increments/decrements to the values 0 V, 1.25 V, 2.5 V, 3.75 V, 5 V	Increments/decrements to the values 0 V, 1.25 V, 2.5 V, 3.75 V, 5 V

Test	Test Case	Expected Outcome	Test Outcome
Change Programable Parameters Values	Click on up arrow to increment/decrement "Atrial Amplitude"	Increments/decrements to the values 0 V, 1.25 V, 2.5 V,3.75 V, 5 V	Increments/decrements to the values 0 V, 1.25 V, 2.5 V,3.75 V, 5 V

Change Programable Parameters Values	Click on up arrow to increment/decrement “ARP”	Increments/decrements by 10ms in 150 – 500ms	Increments/decrements by 10ms in 150 – 500ms
Change Programable Parameters Values	Click on up arrow to increment/decrement “VRP”	Increments/decrements by 10ms in 150 – 500ms	Increments/decrements by 10ms in 150 – 500ms
Test Programable Parameters Values Upper and Lower Boundaries	Click on up arrow to increment/decrement “Lower Rate Limit”	Increments until 175 ppm then wraps around to 30ppm. Decrements until 30 ppm then wraps to 175 ppm	Increments until 175 ppm then wraps around to 30ppm. Decrements until 30 ppm then wraps to 175 ppm
Test Programable Parameters Values Upper and Lower Boundaries	Click on up arrow to increment/decrement “Upper Rate Limit”	Increments until 175 ppm then wraps around to 50ppm. Decrements until 50 ppm then wraps to 175 ppm	Increments until 175 ppm then wraps around to 50ppm. Decrements until 50 ppm then wraps to 175 ppm
Test Programable Parameters Values Upper and Lower Boundaries	Click on up arrow to increment/decrement “Atrial Pulse Width”	Increments until 1.9ms then wraps around to 0.05ms. Decrements until 0.05ms then wraps to 1.9ms	Increments until 1.9ms then wraps around to 0.05ms. Decrements until 0.05ms then wraps to 1.9ms
Test Programable Parameters Values Upper and Lower Boundaries	Click on up arrow to increment/decrement “Ventricular Pulse Width”	Increments until 1.9ms then wraps around to 0.05ms. Decrements until 0.05ms then wraps to 1.9ms	Increments until 1.9ms then wraps around to 0.05ms. Decrements until 0.05ms then wraps to 1.9ms
Test Programable Parameters Values Upper and Lower Boundaries	Click on up arrow to increment/decrement “Ventricular Amplitude”	Increments until 5 V then wraps around to 0 V. Decrements until 0 V then wraps to 5 V	Increments until 5 V then wraps around to 0 V. Decrements until 0 V then wraps to 5 V
Test Programable Parameters Values Upper and Lower Boundaries	Click on up arrow to increment/decrement “Atrial Amplitude”	Increments until 5 V then wraps around to 0 V. Decrements until 0 V then wraps to 5 V	Increments until 5 V then wraps around to 0 V. Decrements until 0 V then wraps to 5 V
Test Programable Parameters Values Upper and Lower Boundaries	Click on up arrow to increment/decrement “ARP”	Increments until 500ms then wraps around to 150ms. Decrements until 150ms then wraps to 500ms	Increments until 500ms then wraps around to 150ms. Decrements until 150ms then wraps to 500ms

Change Programmable Parameters Values	Click on up arrow to increment/decrement "ARP"	Increments/decrements by 10ms in 150 – 500ms	Increments/decrements by 10ms in 150 – 500ms
Test Programmable Parameters Values Upper and Lower Boundaries	Click on up arrow to increment/decrement "VRP"	Increments until 500ms then wraps around to 150ms. Decrements until 150ms then wraps to 500ms	Increments until 500ms then wraps around to 150ms. Decrements until 150ms then wraps to 500ms

3.0 Requirements Changes that are Likely to Change

- Design a new Database which saves corresponding programmable parameters for each user registered in the system.
- Fix input of 'space' in username and password entries.
- Different Programmable parameters who are dependent on each other can change the corresponding parameter input.
- Support for more pacing modes for future revisions.
- Set up the communication between Pacemaker controller and DCM through the Simulink sending a byte of code which lets the DCM identify this unique pacemaker.

4.0 Design decisions that are likely to change:

- Create individual modules for serial communication database module for the programmable parameter for different users.
- Display exact time when user logs in.
- Display username in top corner when logged in.
- Design a drop-down menu for Lower Rate Limit, A or V Pulse Width, A or V Amplitude to allow user to select specific values and prevent user from selecting unavailable inputs for a specific programmable parameter.
- Logout Button and Quit Button.