McMaster University

SE 3K04 Fall 2019

Instructor: Dr. Alan Wassyng

Assignment 2 Documentation Part 2

Date issued: December 5th, 2019

Group 29

An, Zifeng	400114057
Fang,Yuxing	400130676
Yuan, Quan	400148577
Xu,Ziyi(Kyrie)	400121488
Zheng, Yiming	400144302

Table of Contents

Changes of Requirement

The current modes in the DCM are definitely not enough for a real pacemaker. Therefore, AAT, VVT, DOOR, DDIR, and DDDR will be added.

Future changes:

1) The unit mark will be added in the name of each parameter in order not to confuse the users. For instance, for the parameter recovery time, the programmable values should be in the range of 2 to 16 minutes. However, the unit mark is not be given, users can not determine whether the unit is minute or not. They may regard seconds as the standard units. For example, when a user input 120 seconds which meets the requirements of the programmable values, the system would still send an error message because the number 120 is out of the limitations. Therefore, it is necessary to add a unit mark in the end of each parameter
2) In most cases, users might not remember all the requested programmable values without specific instructions. In the current DCM, users will only get the specific instructions showing the specific requirements after the users input wrong data. This is unfriendly for users to input digits at the first time because they have no idea about the data they should input. Therefore, a new method of scrollbar instead of bar will be used in the future DCM. The scrollbar has an ability to show all of the numbers that meet the requirements and users only need to click on their desired digit instead of inputting digit.
3) In the future, classes will be used in the DCM, since it will enhance the effect of information hiding. By default, Python regards all of the functions as public. In spite of using modularity programming and doing encapsulation, the information safety still needs to be improved. By using classes, many functions will be turned into private and this will improve the safety of the information. Meanwhile, many global variables can be eliminated by passing function parameters. This will make the code easier to follow and make the result more predictable.[2]
4) In the future, an additional function will be added, which could delete the from the user list. At this time, existing users who don’t want to keep their accounts can delete their account, so that the user will not take up the 10 user spaces and allows other new users have chance to sign in.

Changes of Design Decisions

1. In the new DCM, modularity has been applied by dividing different functions into different Python files. There are several advantages for using modules in different files:
    - Using modules allows programmers to read quicker. In the last DCM, there is only one Python file (main module) which includes all of the functions and windows inside. This caused the main module looked lengthy and jumbled and it was hard for programmers to read. Instead, in the current DCM, all Bradycardia Therapy modes are extracted to be formed as corresponding modules. This helps the main module shorter and clearer for reading.
    - Using modules allows programmers to detect errors easier. For the mode modules of the current DCM, mode modules are independent to each other and they can be called only by the main module, which shows the DCM has high cohesion and low coupling. Furthermore, each mode module can make its own window without influence others. In this case, if programmers want to test and revise details in a specific Bradycardia Therapy mode, the programmers only need to change in each corresponding mode module. This helps programmers do the test and make the revisement quicker and easier.
    - Using modules allows programmers to reuse the code. In the current DCM, there is a module called functions module, which defines all of the functions for each parameter's error message. The module is used in every Bradycardia Therapy mode module, which save redundant codes to be written.

2. The new version DCM will pop a warning window when the registered user tries to register again. This improvement prevents the re-registration error happens which would cause the number of maximum allowed users to decreases.
3. In the function interface, the new version DCM includes a title bar at the top of the window which helps the user memory which mode they are currently in.

Compared to the original version of DCM which does not have the title bar, this makes the new DCM more user-friendly.

4. In the function interface, users can only fire the entered parameters onto the actual pacemaker when the entered parameters are complete and correct. In the original DCM, the user can enter the parameters incompletely and fire these values. The new version makes sure all the parameters are entered before firing. This fulfills the security requirements of the DCM.
5. In the function interface, if the user inputs and saves a parameter which is out of the ideal range, a warning window will pop up. The expected range of parameters will show on that window. Users do not need to go to the specification of the pacemaker to check the expected parameter range. In the previous version DCM, there will be a warning window but with no ideal range. This makes the new DCM more user-friendly.

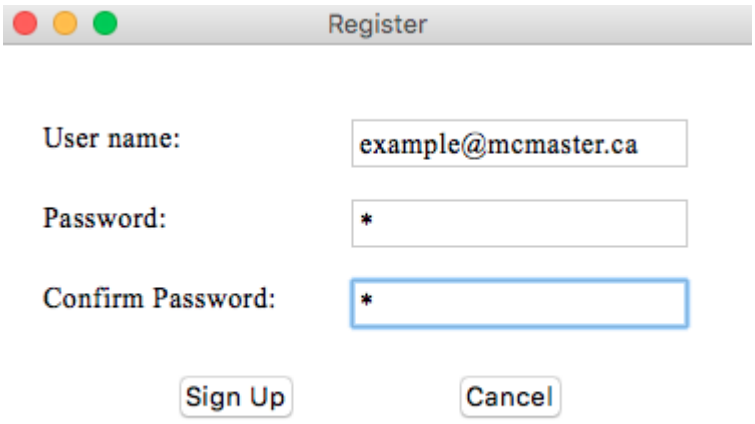
Description of the Purpose of Module

In Python, each file is regarded as a module. There are many modules in DCM.

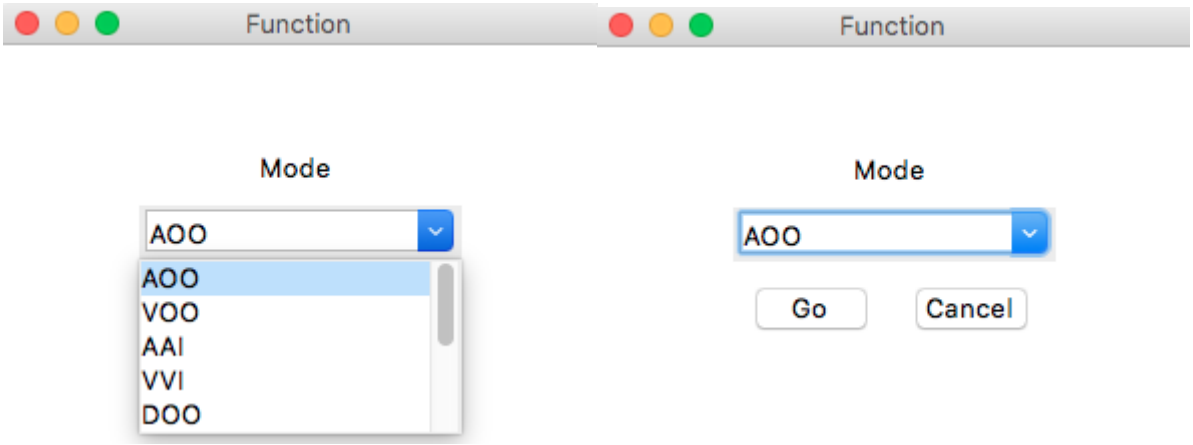
Main Module

The Main Module contains 3 interfaces, which are Login interface, Register interface, and Function interface.

In the Login interface, first-time users are allowed to sign up for an account by clicking the “sign up” button. If the user is already registered, a warning window will appear when this user tries to re-register. This prevents the re-registration error happens which occurred in the original version of DCM.



If the user is successfully logged in by pressing the “login” button, the window below will show. In this window, users are allowed to select different pacemaker modes. If the cancel button is pressed, the user will log out automatically. After the expected mode is selected and the “go” button is pressed, the current window will be closed and the function interface will appear.



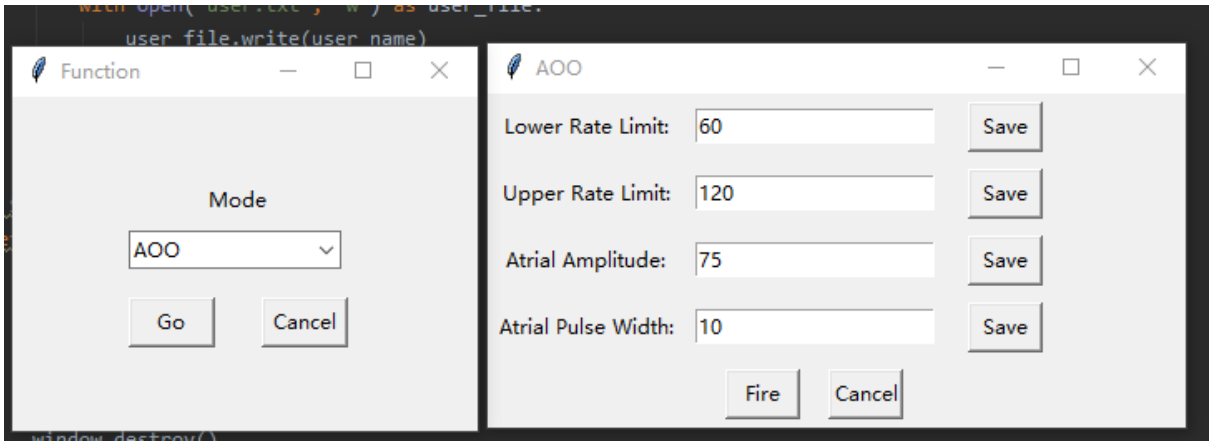
In the Function interface, a scroll bar is added and the user can choose any mode from the scroll bar. After choosing the mode, the user can click on the Go button to reach the corresponding module interface. Besides, there is a Cancel button that allows the user to go back to the Register interface. The above title bar will mention users of which mode they are currently in. Users need to save all the parameters successfully first before they fire these values of parameters onto the actual pacemaker. This makes the DCM robust and secure because the user only can fire complete and correct parameters onto the pacemaker. If users save parameter which is out of the ideal range, a warning window will pop up which includes the expected ideal range of value of parameters.

Serial Communication Module: Provide serial communication between DCM and board. Programming parameters can be sent to the board through this module. Connected to all of the Fire buttons in AOO, VOO, AAI, VVI, DOO, DDD, AOOR, VOOR, AAIR, and VVIR modules.

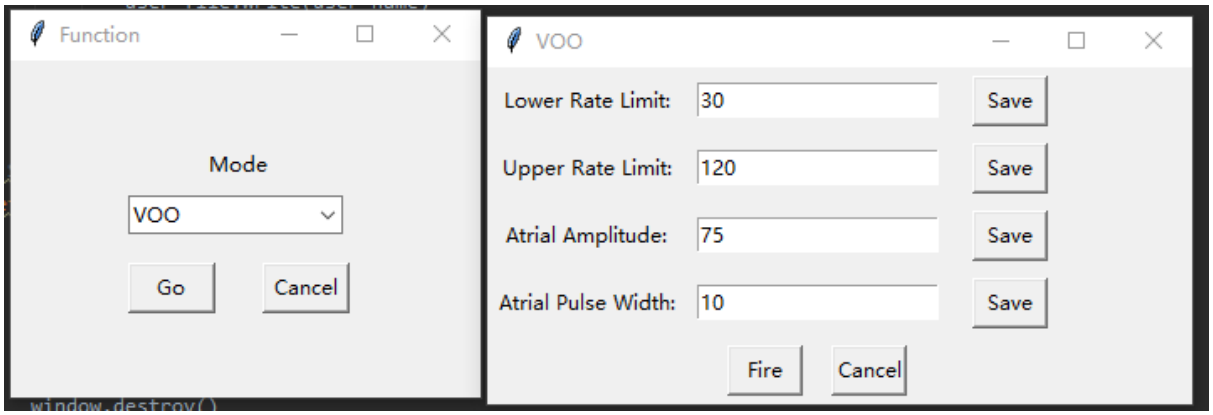
Functions Module: All of the warning and information message boxes are stored here.

The 10 Mode Modules in DCM are listed below. Each module will generate a window for each corresponding mode and all of the modules have similar functions. The Mode Modules allow user to input value for each listed parameter by keyboard and save them individually. The “Fire” button is connected with the writePara function in Serial Communication Module. The “Cancel” button will destroy the mode window and regenerate the mode choosing window.

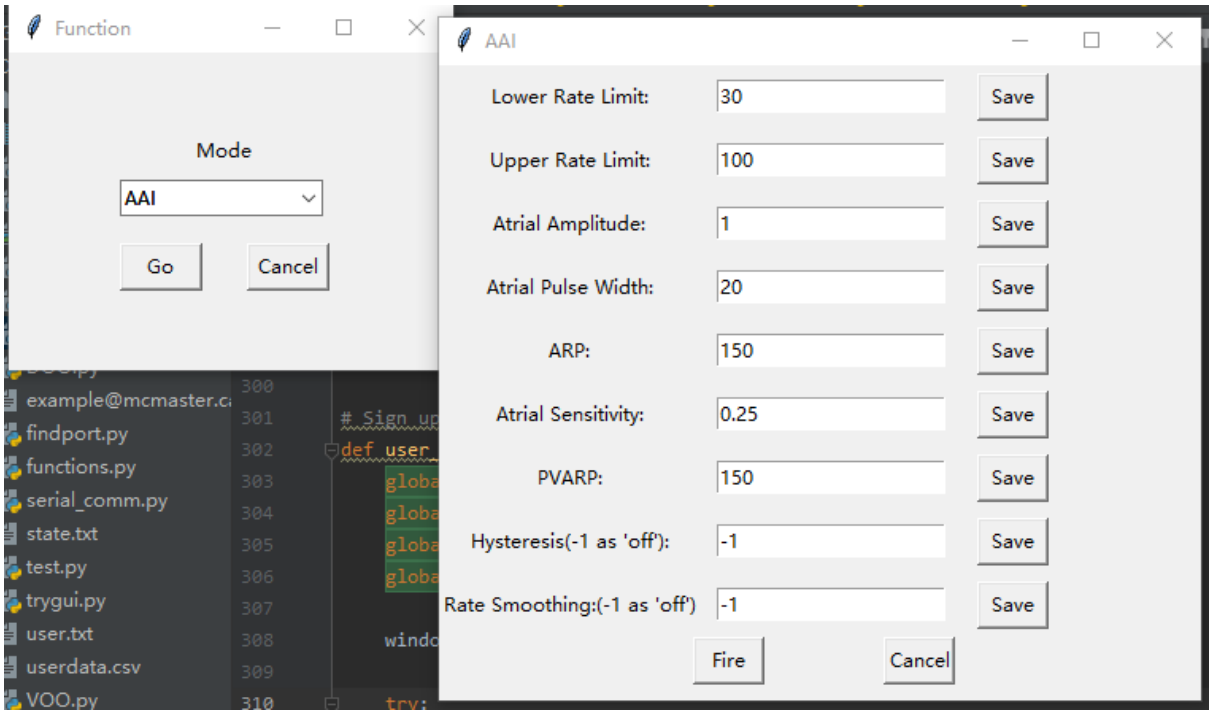
AOO Module



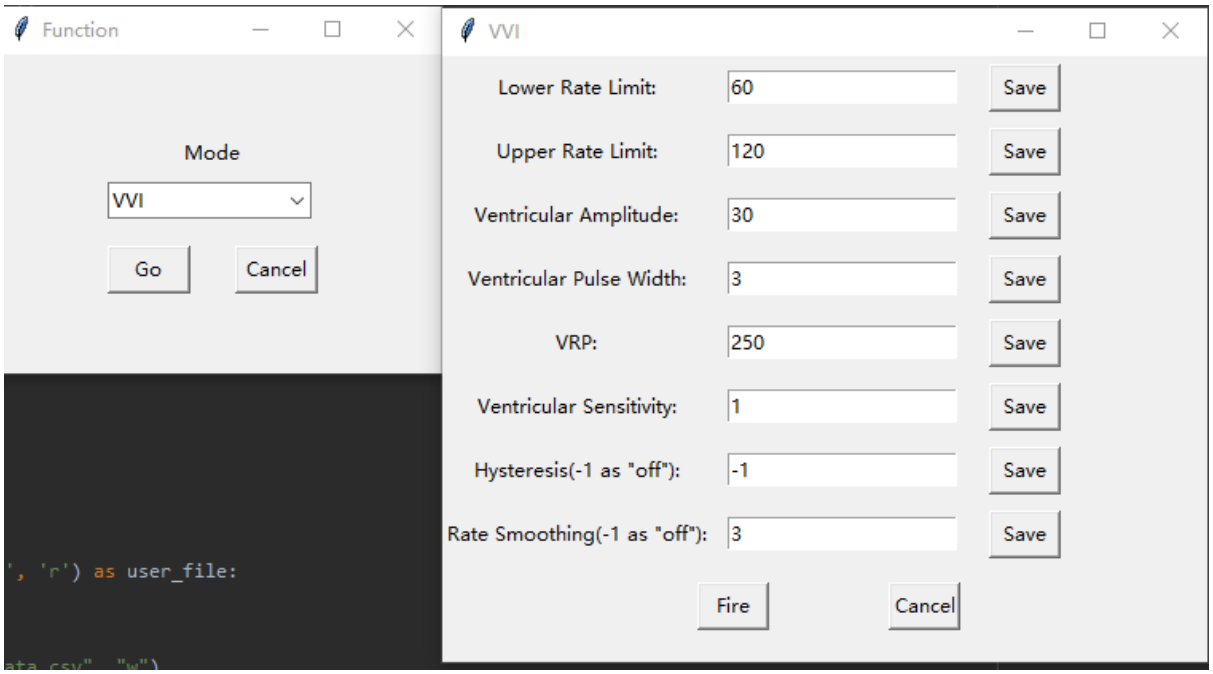
VOO Module



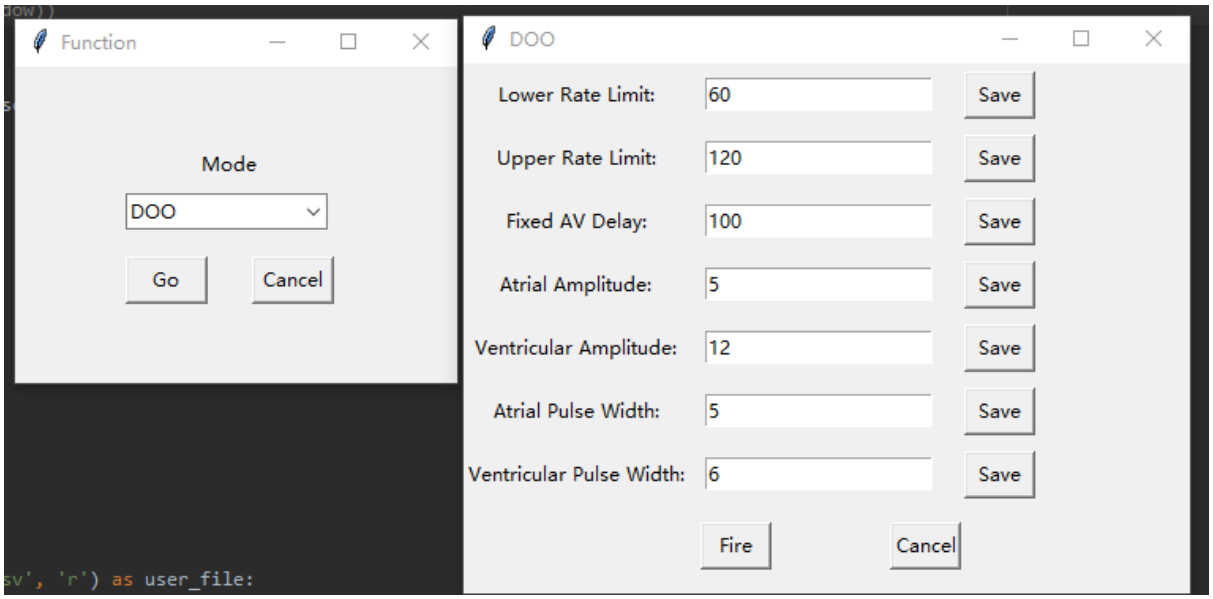
AAI Module



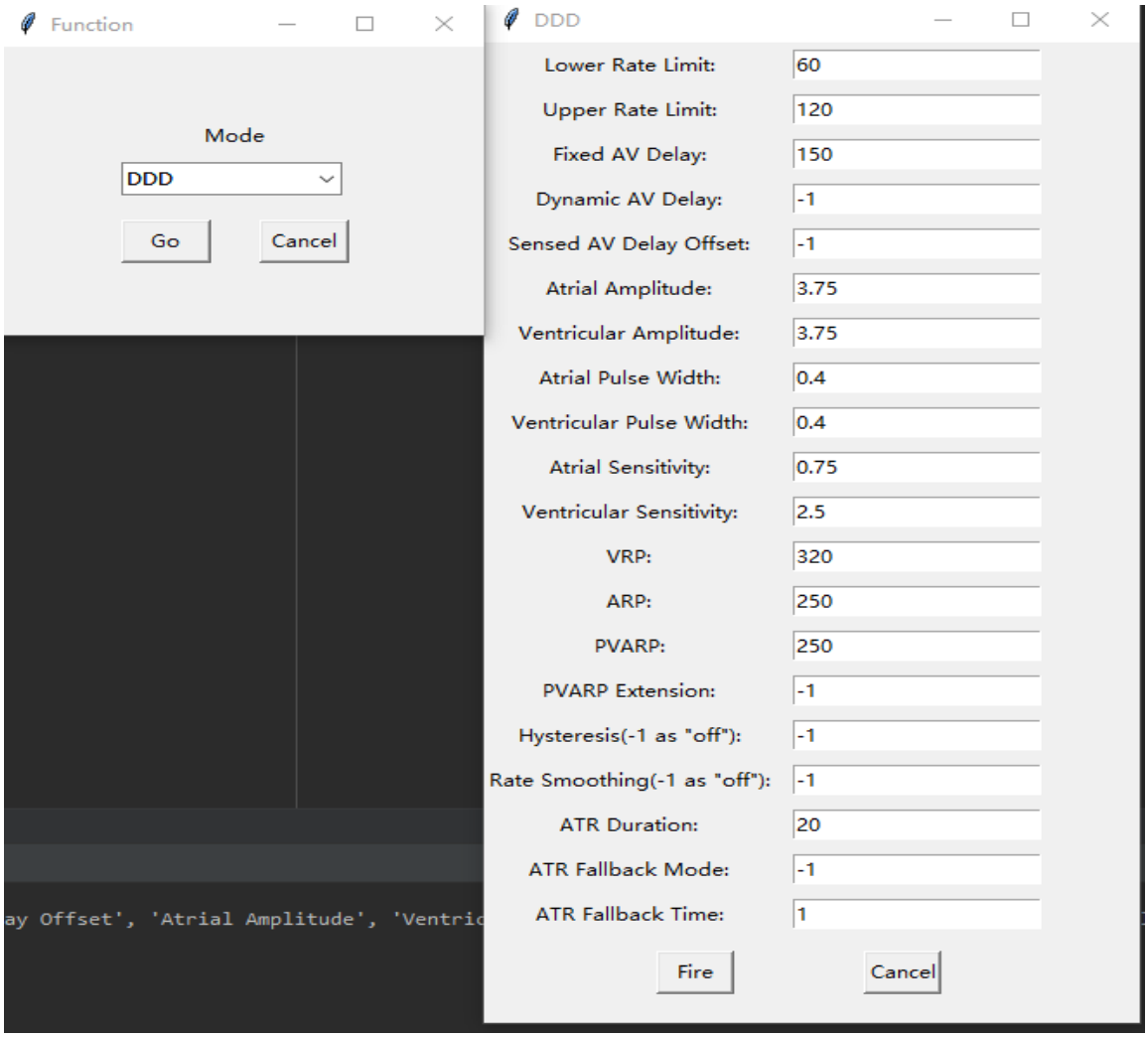
VVI Module



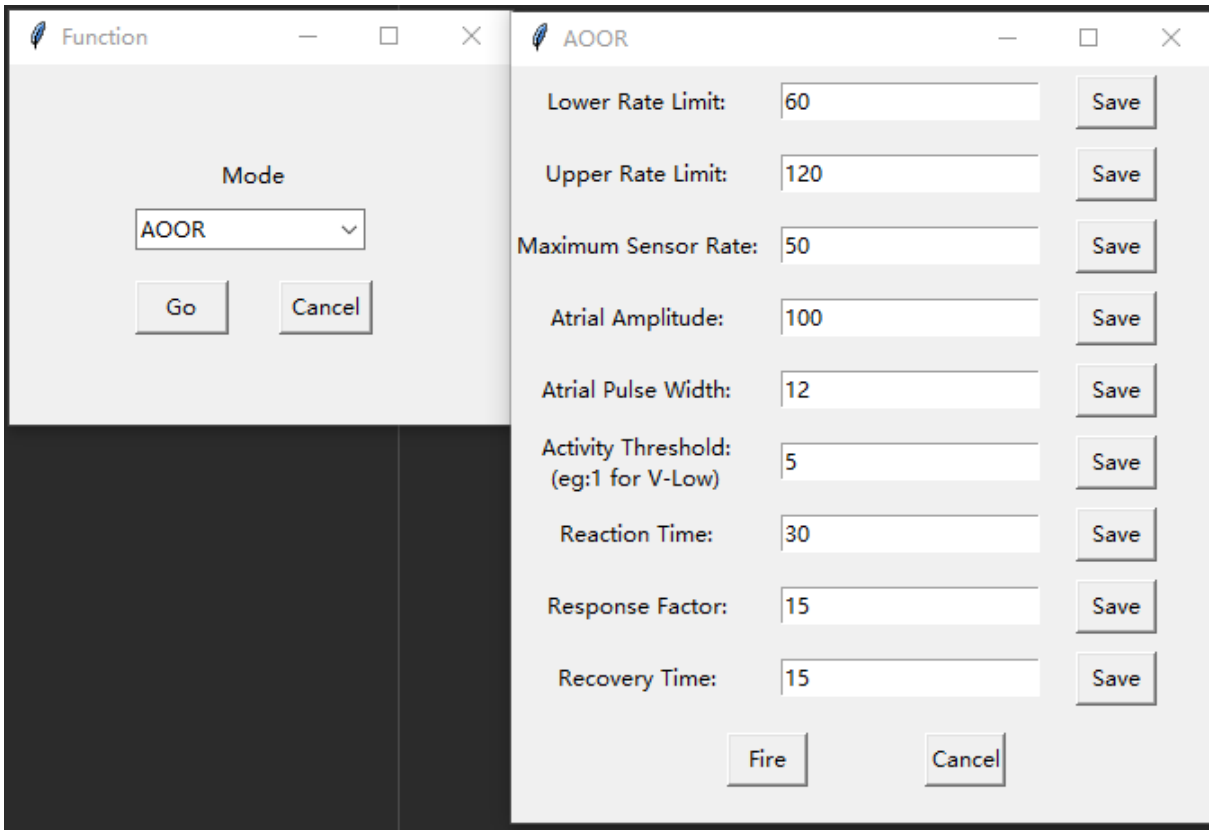
DOO Module



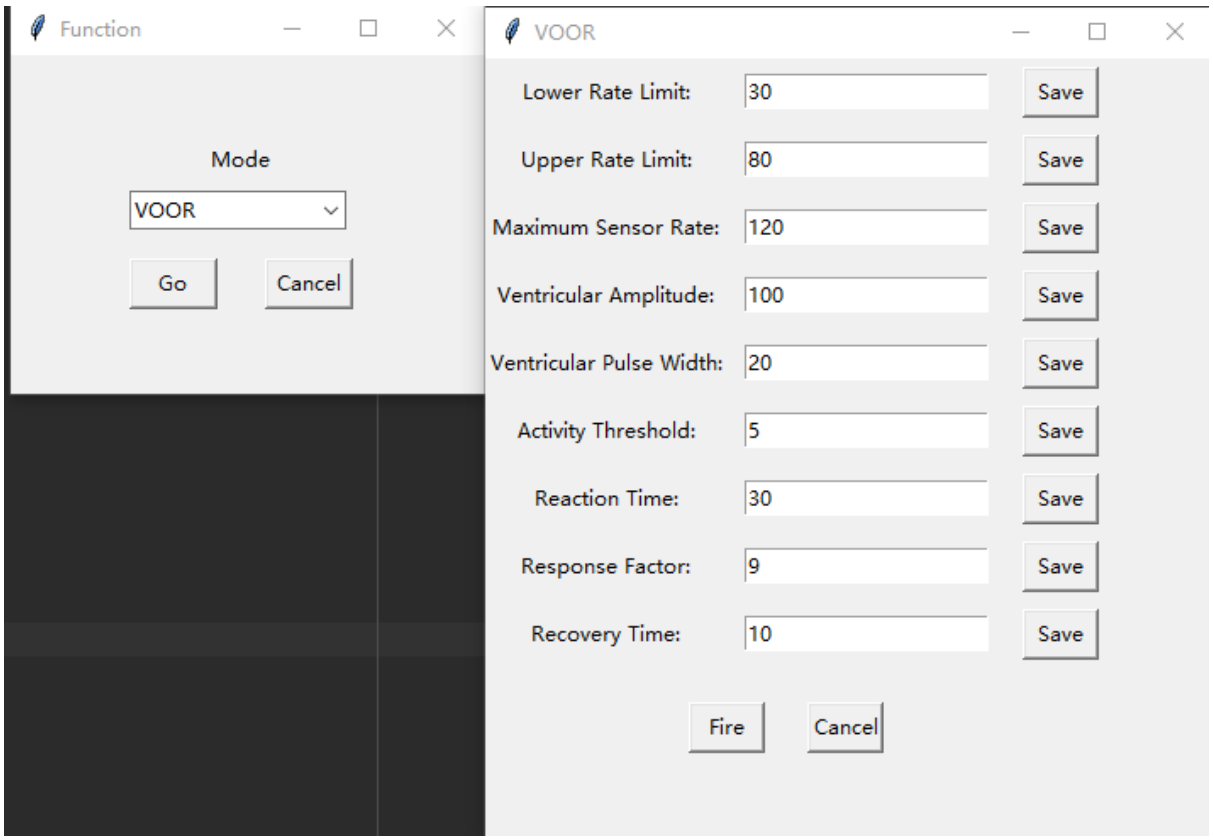
DDD Module



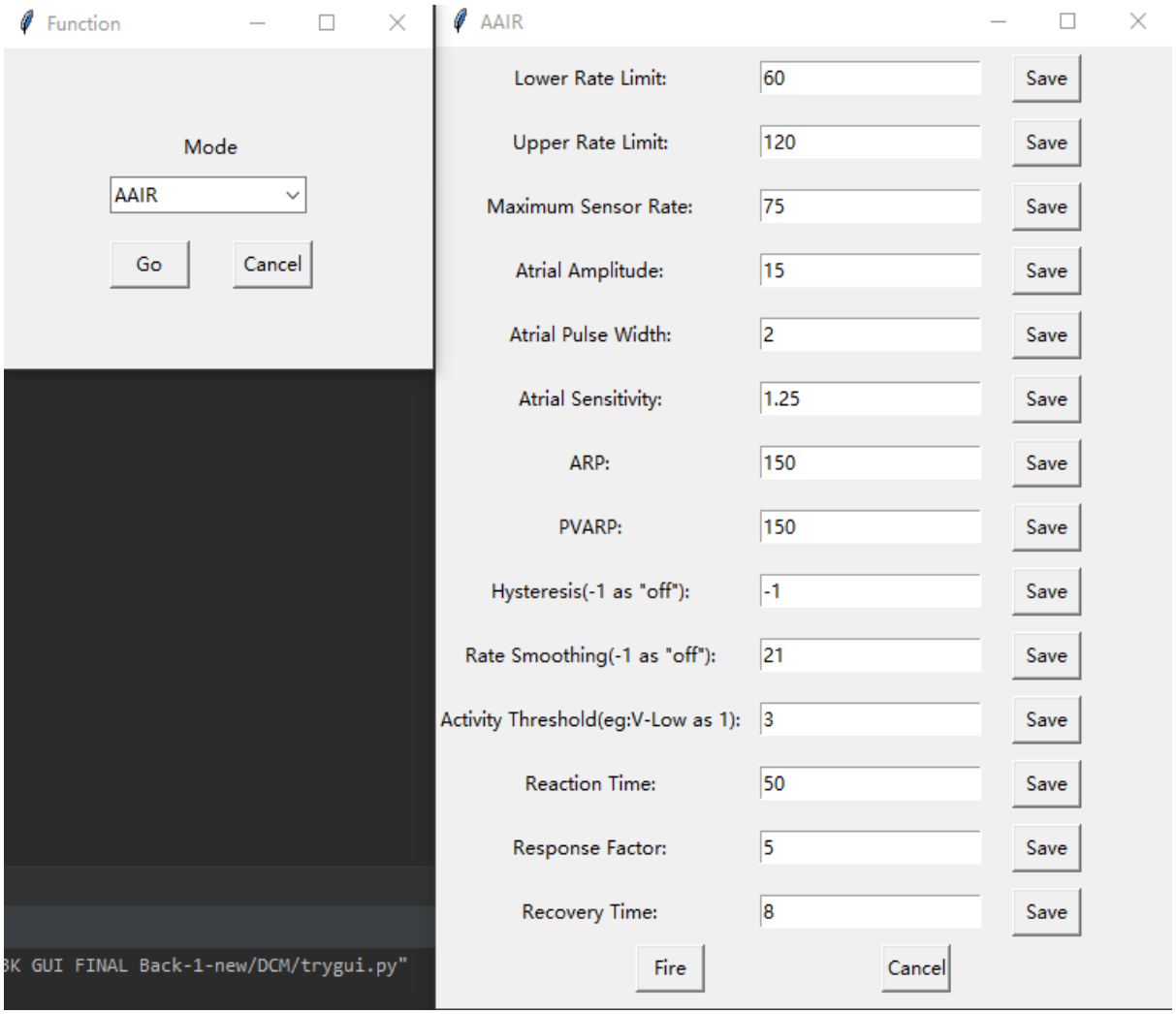
AOOR Module



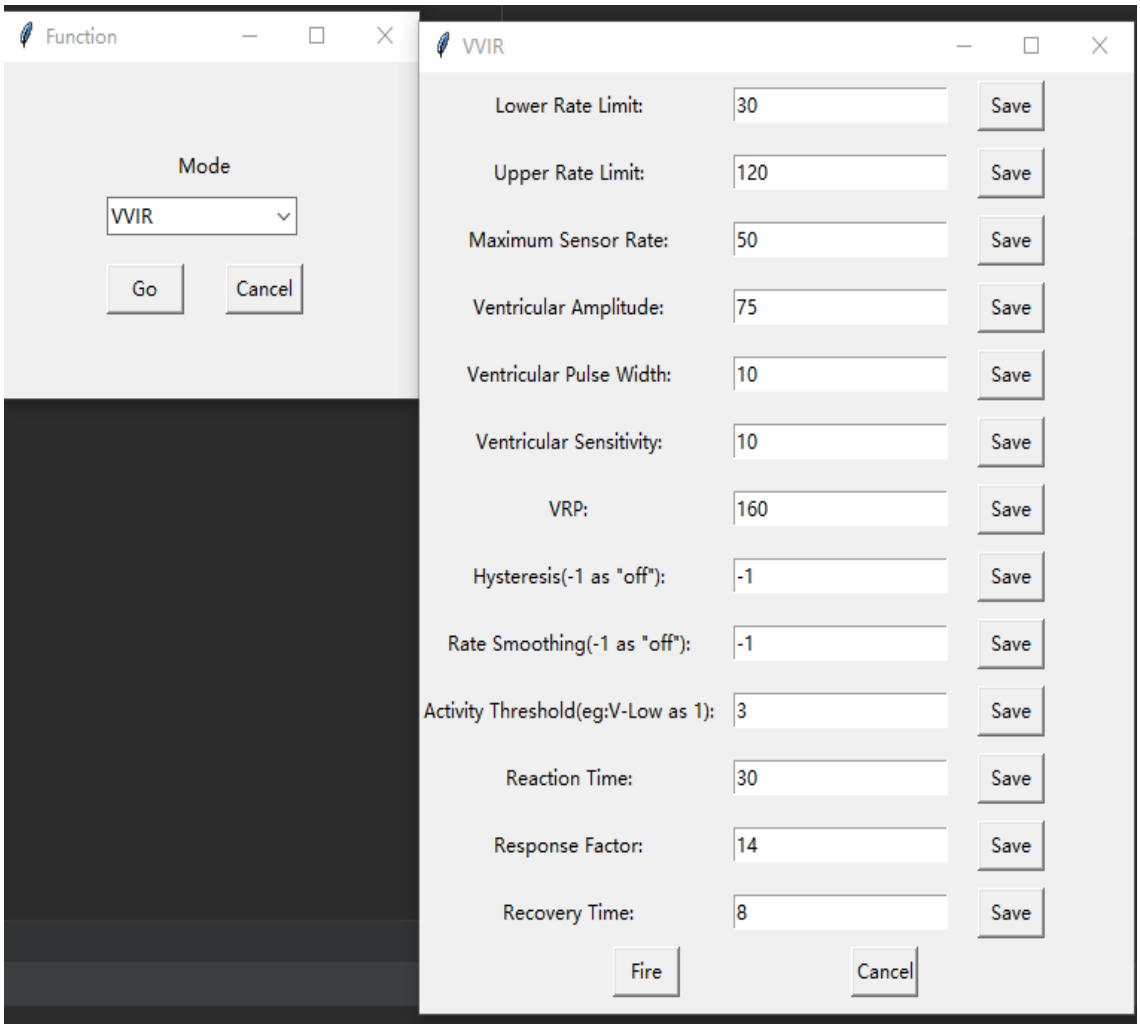
VOOR Module



AAIR Module


Function
Mode
AAIR
Go
Cancel
8K GUI FINAL Back-1-new/DCM/trygui.py"
AAIR
Lower Rate Limit:
60
Save
Upper Rate Limit:
120
Save
Maximum Sensor Rate:
75
Save
Atrial Amplitude:
15
Save
Atrial Pulse Width:
2
Save
Atrial Sensitivity:
1.25
Save
ARP:
150
Save
PVARP:
150
Save
Hysteresis(-1 as "off"):
-1
Save
Rate Smoothing(-1 as "off"):
21
Save
Activity Threshold(eg:V-Low as 1):
3
Save
Reaction Time:
50
Save
Response Factor:
5
Save
Recovery Time:
8
Save
Fire
Cancel

VVIR Module



“Secret” of the Module

1. Main Module: When the 10th user is registered successfully, there will be a message box popped up with “Wow, you are the last person who registered!”. At this time, a global variable called state will be changed from 1 to 0, so that the 11th user will not be allowed to register.
2. n/a

List of Public Functions in the Module and Its Parameters

The public functions, which can be used by the user, are the following: Users can sign up and log in their personal accounts up to 10 in the main window. After the account is logged in, the user is able to select 10 different pacing modes by using scroll bar. After the user chooses one of the modes, the user is able to view the current saved parameters and modify them.

There is only one public function that uses parameters since there are lots of global variables used.

Serial Communication Module:

writePara(mode=0, Lower_Rate=60, MSR=120, AV_Delay=150, ATR_Amplitude=75, VENT_Amplitude=75, ATR_Width=10, VENT_Width=10, VENT_Refractory=320, ATR_Refractory=250, Activity_Threshold=2, Reaction_Time=5, Response_Factor=10, Recovery_Time=5): There are 14 parameters with their default values in writePara function

Functions Module:

successFire

success

Lower_Rate_Limit_Error

Upper_Rate_Limit_Error

Ventricular_Amplitude_Error

Atrial_Amplitude_Error

Ventricular_Pulse_Width_Error

Atrial_Pulse_Width_Error

ARP_Error

VRP_Error

Atrial_Sensitivity_Error

Ventricular_Sensitivity_Error

Hysteresis_Error

Rate_Smoothing_Error

PVARP_Error

FixDelay_Error

Reaction_Time_Error

Response_Factor_Error

Recovery_Time_Error

Activity_Threshold_Error

Maximum_Sensor_Rate_Error

Serial Communication Module:

writePara

AOO Module:

AOO_window

show_AOO_LRL

show_AOO_URL

show_AOO_AA

show_AOO_APW

AOOR Module:

AOOR_window

show_AOOR_LRL

show_AOOR_URL

show_AOOR_MSR

show_AOOR_AA

show_AOOR_APW

show_AOOR_AT

show_AOOR_ReactionT

show_AOOR_RF

show_AOOR_RecoveryT

AAI Module:

AAI_window

show_AAI_LRL

show_AAI_URL

show_AAI_AA

show_AAI_APW

show_AAI_ARP

show_AAI_AS

show_AAI_PVARP

show_AAI_Hysteresis

show_AAI_RS

AAIR Module:

AAIR_window

show_AAIR_LRL

show_AAIR_URL

show_AAIR_AA

show_AAIR_APW

show_AAIR_ARP

show_AAIR_AS

show_AAIR_PVARP

show_AAIR_Hysteresis

show_AAIR_RS

show_AOOR_AT

show_AOOR_ReactionT

show_AOOR_RF

show_AOOR_RecoveryT

VOO Module:

VOO_window

show_VOO_LRL

show_VOO_URL

show_VOO_VA

show_VOO_VPW

VOOR Module:

VOOR_window

show_VOOR_LRL

show_VOOR_URL

show_VOOR_VA

show_VOOR_VPW

show_VOOR_MSR

show_VOOR_AA

show_VOOR_APW

show_VOOR_AT

show_VOOR_ReactionT

show_VOOR_RF

show_VOOR_RecoveryT

VVI Module:

VVI_window

show_VVI_LRL

show_VVI_URL

show_VVI_VA

show_VVI_VPW

show_VVI_VRP

show_VVI_VS

show_VVI_Hysteresis

show_VVI_RS

VVIR Module:

VVIR_window

show_VVIR_LRL

show_VVIR_URL

show_VVIR_VA

show_VVIR_VPW

show_VVIR_MSR

show_VVIR_VRP

show_VVIR_VS

show_VVIR_Hysteresis

show_VVIR_RS

show_VVIR_AT

show_VVIR_ReactionT

show_VVIR_RF

show_VVIR_RecoveryT

DOO Module:

DOO_window

show_DOO_LRL

show_DOO_URL

show_DOO_FixDelay

show_DOO_AA

show_DOO_VA

show_DOO_APW

show_DOO_VPW

DDD Module:

DDD_window

show_DDD

Description of Black-Box Behavior of Each Function

1. Main Window:

- Show the picture as the upper background and the color pink as the background color.
- Generate the main window which allows the user to type in the user name,
  password and also the “Sign Up” button and the “Login” button which allows the user to click on.
  i. While clicking on the “Sign Up” button, the main window will be shut down and the “Sign up” window will be popped up.
  ii. While clicking on the “Login” button, the details will be shown in Login below.

2. Sign Up Window:

○ Generate a Sign-Up window which allows the user to type in a username,

password, confirmed password and click on the “Sign Up” button or “Cancel” button.

i. While clicking on the “Cancel” button, the Sign Up window will be shut down and the Main window will appear again.
ii. After inputting username, password, confirm password, clicking on

the “Sign Up” button, there will be several situations:

1. If the user signed up before, then an error box will be popped up which states the user has been registered.
2. If the confirmed password doesn’t match the password, then an error box will be popped up which states the confirmed password should stay the same with the password.
3. If the username and password are blank, then a message box will be popped up to show that the username or password cannot be blank.
4. If all the inputs are correct, then a message box will be popped up showing that the user signs up successfully and the user information will be stored in a pickle file.

3. Log In: Generate a message box that states whether the user can successfully log in. After inputting the username and password, clicking on the “Login” button, there will be several situations:

- If the username did not be registered before, a message box will be popped up to ask whether jumping to the Sign Up window to sign up for an account or not.
- If the password is wrong, then an error box will be popped up which states the password is wrong.
- If the username and password are both correct, a welcome information message will be popped up and then jump to Mode-Choosing Window.

4. Mode-Choosing Window: A scroll bar is added for choosing different modes. After choosing the mode and clicking on the Go button, the corresponding mode window will show up.

5. Serial Communication: When this function is called, input parameters will be sent from DCM to board through serial communication. Besides, the users can send the parameters without saving them. If the input values are not correct for its

corresponding parameter, a warning and information message box will be popped up to show the correct range of the parameter.

Description of Global Variables and Its Data Structure

Main Module:

global window: <class 'tkinter.Tk'>
global username: <class 'tkinter.StringVar'>
global userPswd: <class 'tkinter.StringVar'>
global username_input: <class 'tkinter.Entry'>
global userpswd_input: <class 'tkinter.Entry'>
global func_window: <class 'tkinter.Tk'>
global roll: <class 'tkinter.ttk.Combobox'>
global state: <class 'str'>
global user_name: <class 'tkinter.StringVar'>
global condition: <class 'bool'>
global column_name: <class 'list'>
global register_window: <class 'tkinter.Tk'>
global counter: <class 'int'>

Serial Communication Module: No global variable.

Functions Module: No global variable.

Mode Modules

All the global variables in Mode Modules are created for passing the value to Serial Communication Module. Their names are:

global e1: integer
global e3: integer
global e4: integer
global e5: integer

List of Private Function in the Module

The internal functions, which are hidden from the users, are the following: When the users sign up, usernames and passwords would be saved into a CSV file. The program would check the number of accounts that have been signed up, the sign up is not allowed if the number of accounts exceeds 10 in the current DCM. When the user logs in successfully for the first time, a csv file named with its username will be created for storing the parameters for this user. Therefore, different users would have their parameters saved in their private files.

Main Module:

user_log_in
user_sign_up
sign_to_Pacemaker

Description of the Internal Behaviour of Each Public and Private Function in Details

Main Module:

main_window: The main_window function generates the main window of our project consisting of a pink background, a “Sign Up” button, a “Login” button, a block for inputting the username and a block for inputting a password. The “Sign up” button is linked to the user_sign_up function(which will be shown below) and the “Login” button is linked to the user_login function.

function_window: The function_window function generates a function window consisting of a scroll bar of all 10 modes, a “Go” button, and a “Cancel” button. The scroll bar can be used to choose modes. The “Go” button is linked with the modeChoose function. The “Cancel” button is linked with the mode window dest

modeChoose: This function is for mode choice and it’s connected with the “Go” button. According to the scroll bar data, this function will pick up the corresponding mode window and pop it up.

user_login: The user_login function generates a conditional statement. If the program does not detect the username that the user put, the “Login” button will be linked to the user_sign_up function and at the same time create a blank pickle file for recording all users’ usernames and passwords. If the program detects that both username and password are mismatched, an error message will be popped up. If the program detects that both username and password are matched each other, the “Login” button will be linked to the function window page with a welcome information message.

user_sign_up: The user_sign_up function generates a window that allows the user to register an account. If it is the first user signing up, a blank file will be created for recording data from users’ information.

sign_to_Pacemaker: This function makes sure users input the correct username and user password. If the

register_window_close: The register_window_close function is able to destroy the register window and regenerate the main window at the same time. This function is linked with the “Cancel” button in the register window.

Mode Modules:

There are 10 Mode Modules in total. In the AOO Module, 4 parameters can be input and saved individually. If the input value is not correct, a warning will show up and an information window will pop up to show the correct range of the input. If the input is empty, a warning will pop up to show the input shouldn’t be empty. After clicking on the Fire button, 3 required parameter inputs can be sent to the board by serial communication. To compare with AOO Module, VOO Module is pretty similar and all buttons have similar functions. It’s just like all the atrial parameters are changed into ventricular parameters. Except for the number of parameters and parameter types, there are only very few differences between these Mode Modules.

Functions Module:

successFire: Pop up message box to inform users that the value has been successfully sent to board by serial communication.

success: Pop up message box to inform users that the value has been successfully saved into the corresponding csv file.

All the functions listed below are for popping up message boxes to inform users that there are something wrong with the corresponding input values and the correct ranges will be shown on the message box.

Lower_Rate_Limit_Error
Upper_Rate_Limit_Error
Ventricular_Amplitude_Error
Atrial_Amplitude_Error
Ventricular_Pulse_Width_Error
Atrial_Pulse_Width_Error
ARP_Error
VRP_Error
Atrial_Sensitivity_Error
Ventricular_Sensitivity_Error
Hysteresis_Error
Rate_Smoothing_Error
PVARP_Error
FixedDelay_Error
Reaction_Time_Error
Response_Factor_Error
Recovery_Time_Error
Activity_Threshold_Error
Maximum_Sensor_Rate_Error

Serial Communication Module:

writePara: This function is for opening serial port between the board and the DCM. Besides, it can send all the required parameter inputs from DCM to the board

by serial communication. This function is linked with every “Fire” button in each Mode Module.

Reference

[1]"Python Release Python 3.7.0", *Python.org*, 2019. [Online]. Available: https://www.python.org/downloads/release/python-370/. [Accessed: 05- Dec- 2019].

[2] A. Forloney, M. Flaschen and I. Sheershoff, "The advantage / disadvantage between global variables and function parameters in PHP?", *Stack Overflow*, 2019. [Online]. Available: https://stackoverflow.com/questions/2216340/the-advantage-disadvantage-between-global-variables-and-function-parameters-in. [Accessed: 05- Dec- 2019].