Configure UI Code Documentation

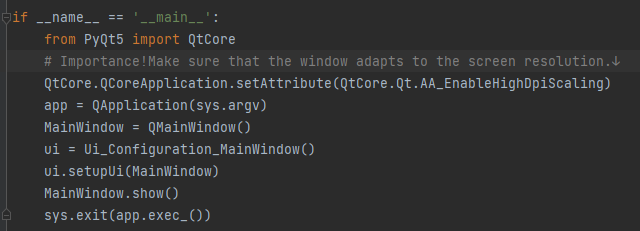
Author: Li Yuxuan

This code document is divided into two parts, user interaction and implementation method

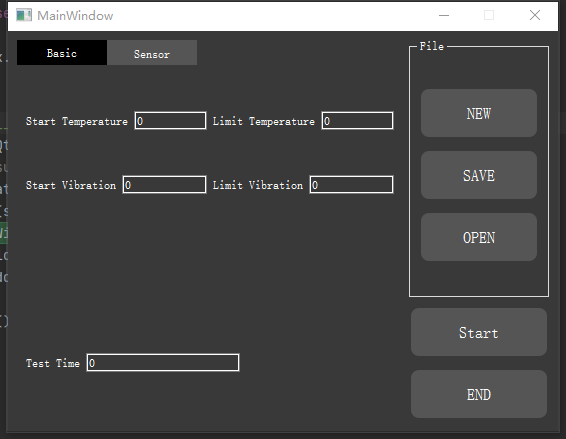
## User interaction

The UI provides a window for the user to enter the simulation configuration and control the simulation process.

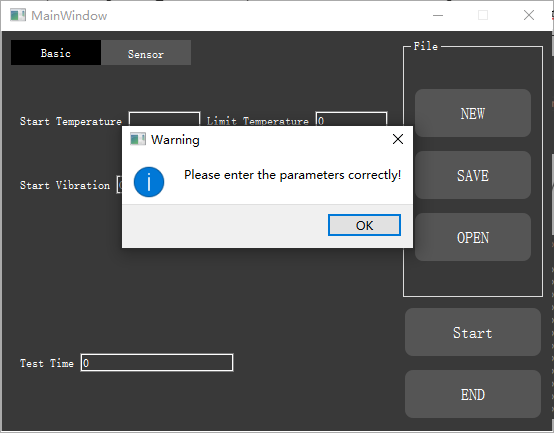
Users can use this interface to run through the following methods:



After the user runs the code, the following window will be displayed:



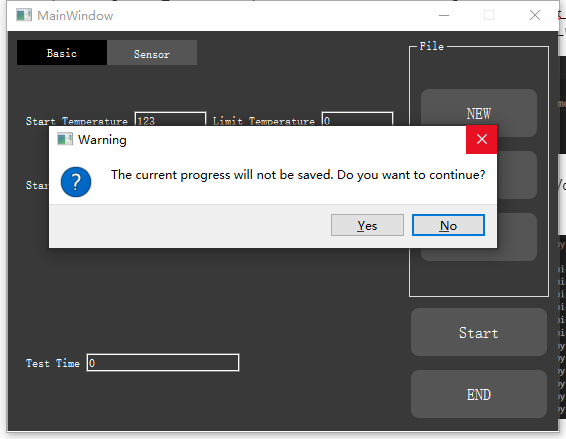
In this interface, users can enter the parameters they want to import DES in the input box. But the input box cannot be empty, otherwise it will be judged as illegal input and will not continue to the next step.



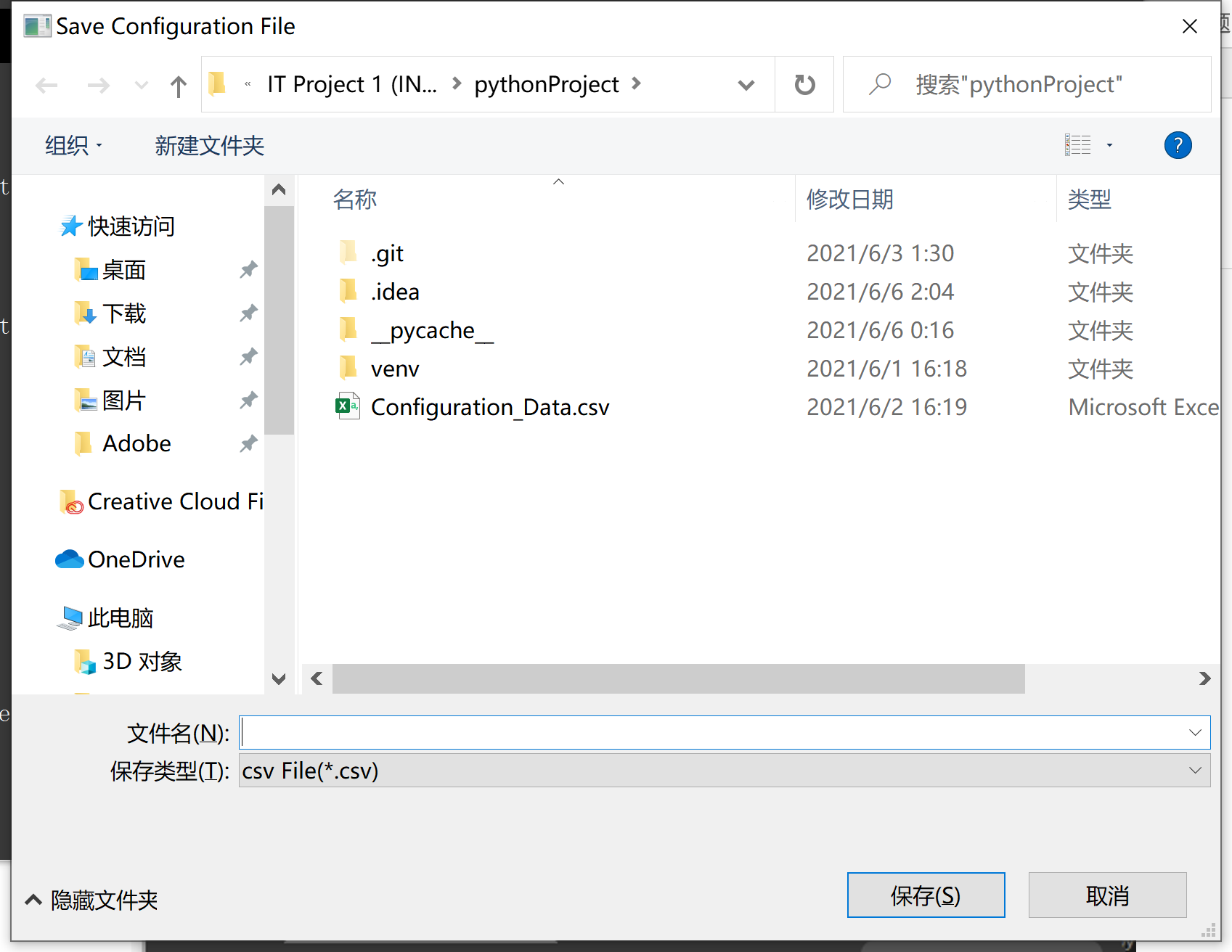
For configuration parameters, users can perform three operations on the file:

1. **Create a new DES**

**New DES New DES input means to reset all interface parameters. There will be an information box for inquiry. Users should carefully select "Yes" to avoid unsaved processes.**



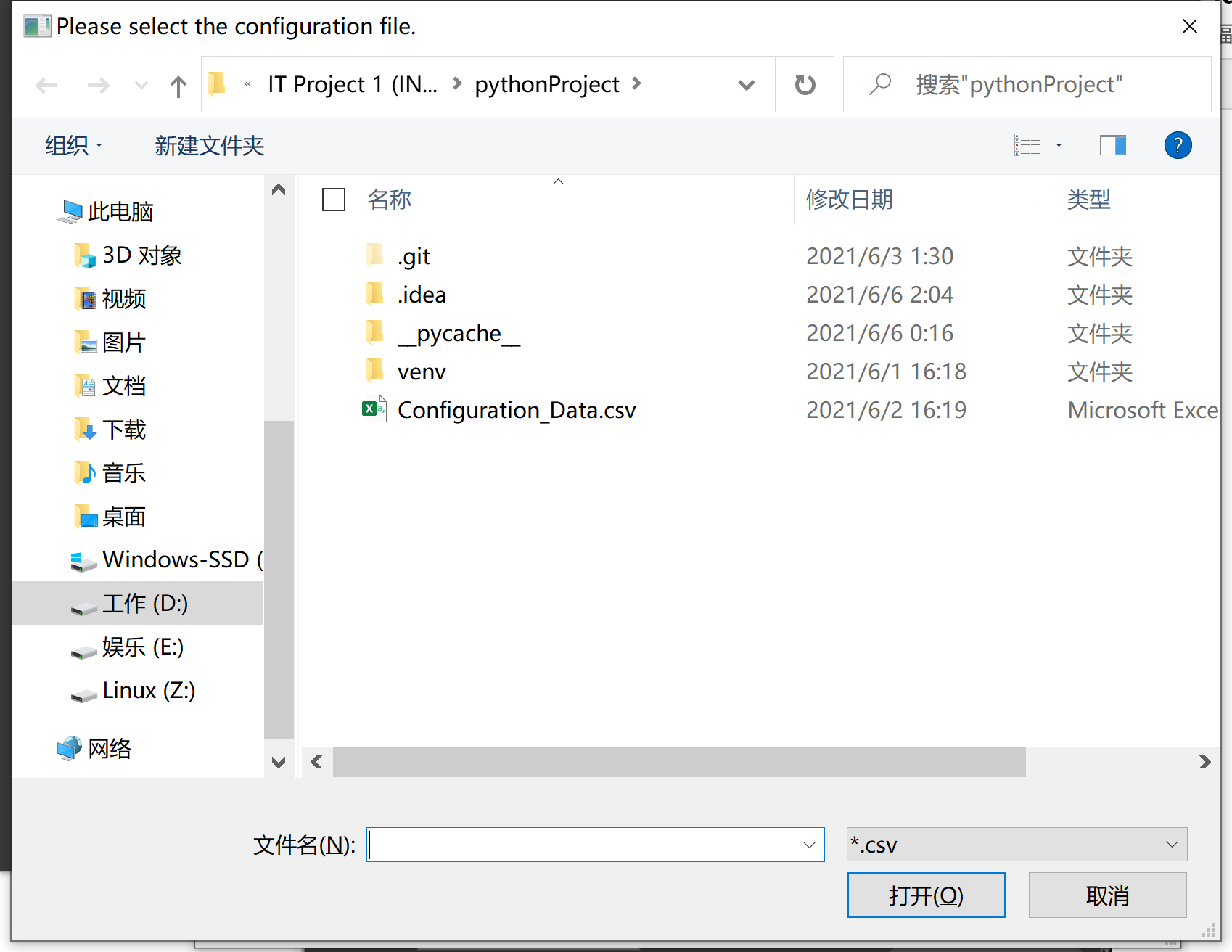
1. **Save current configuration parameters**



Click Save to open a window for saving operation. This operation will generate a configuration parameter csv in a specific format(csv).

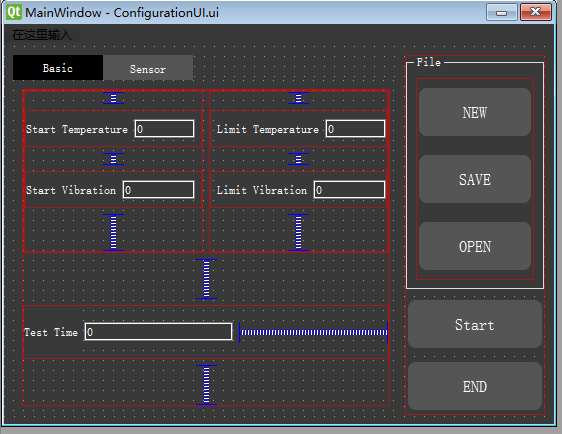
1. **Open the configuration file**

After clicking "Open", the user can select the saved configuration file through this window.



\*Note that the configuration file should be imported in a fixed format, otherwise it cannot be imported.

## Code implementation method

The front end of this interface is based on QTdesigner for layout and settings.

In this window, the layout mainly involves the following controls:

**Layout**

Through the built-in layoutStretch parameter of horizontal layout and vertical layout, the designer can assign the proportion of the control in the layout, and the proportion can be adaptive with the change of the window size.

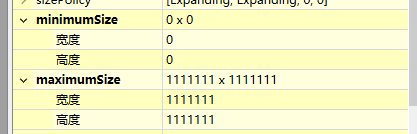
**Groupbox, Tagwidget**

The containers provide docked classification groups for the control.

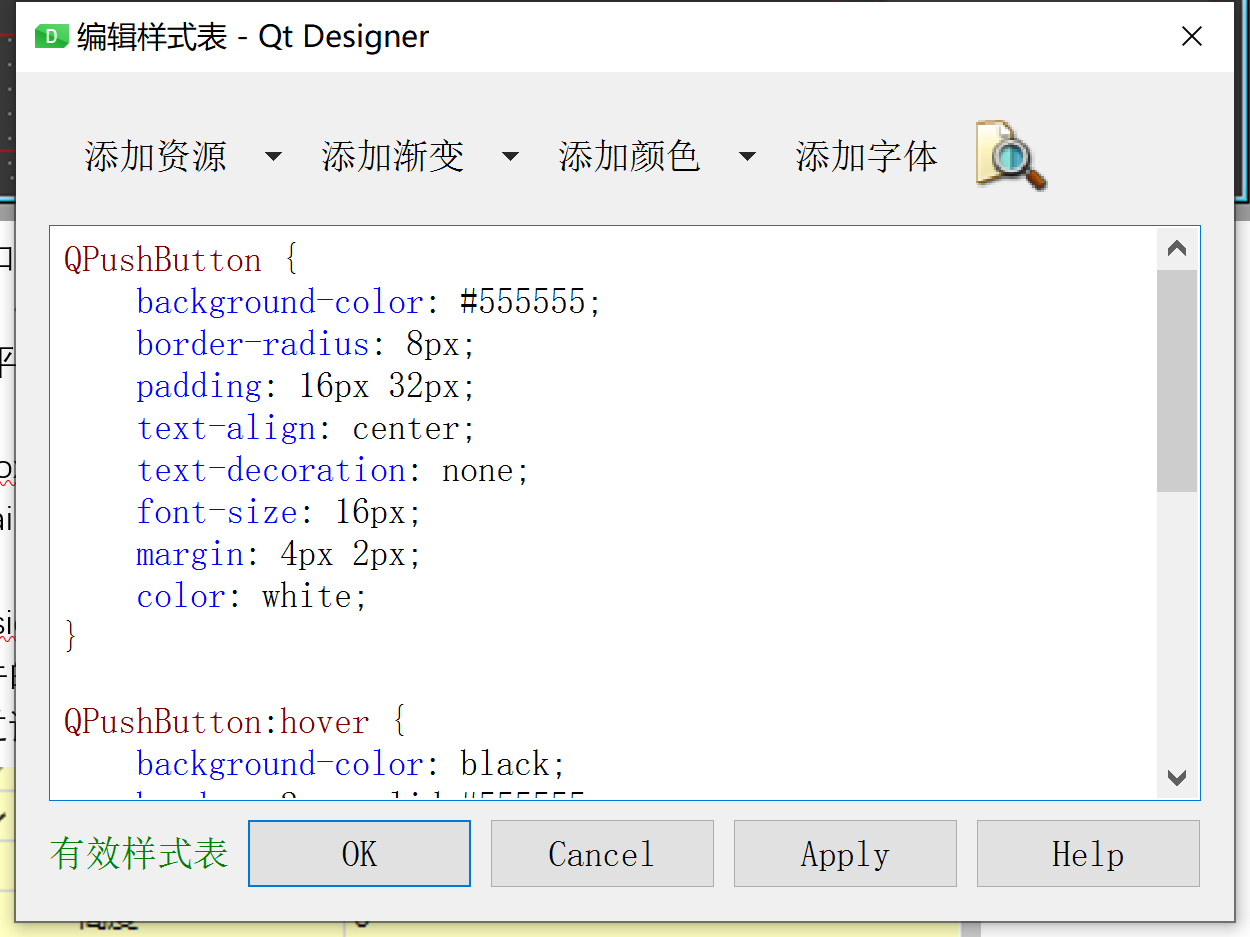
The designer can also preset the following parameters in qtdesigner:

**1. The maximum and minimum size of the control**

Through this parameter, the designer can set the maximum and minimum size of the window or each control separately.



**2. Style sheet**



Through the style sheet, the designer can change the style of different groups of controls. Including: background color, font, size. The designer can also set the change of the floating style of the button.

The interface designed in qtdesigner will be saved in .ui format, which is C++ style. The designer can use the external tool PYUIC to convert the ui file into a .py file.

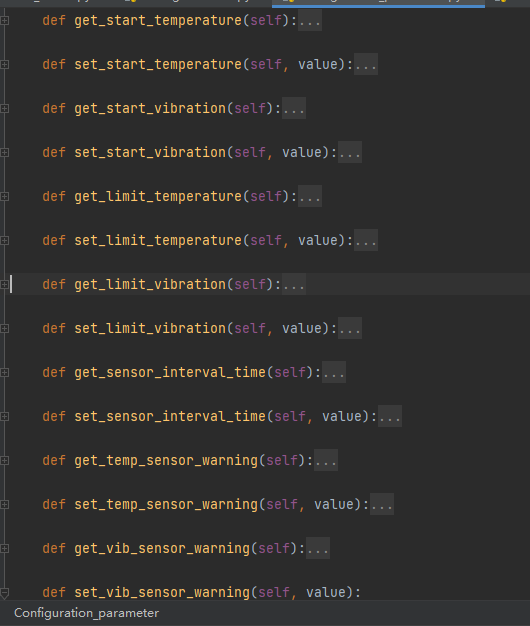
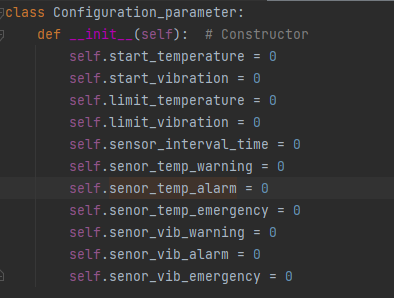
The backend method will be implemented in the py file

**Backend code**

Back-end functions are implemented using code. The file is divided into the window file ConfigurationUI.py and the parameter class Configuration\_paramer.py.

## Parameter class

The parameter class is mainly used to store the configuration parameters of the current interface, so that other modules or classes can call and input.

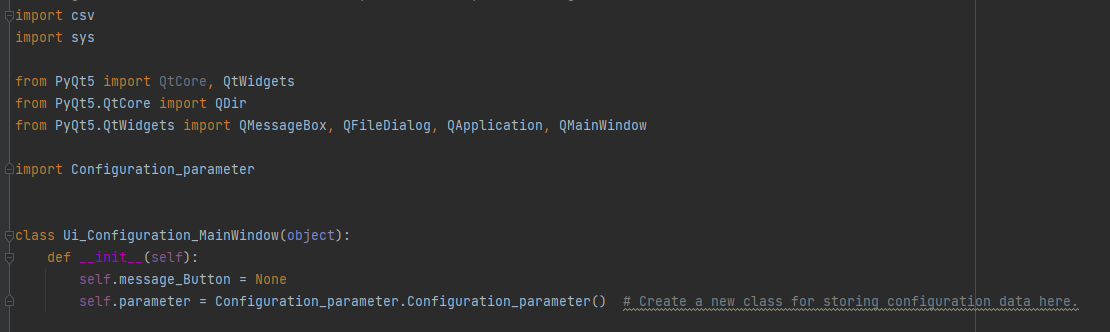


The methods in the class are the class constructor and the setter and getter of each parameter.

## Window class

The window class is converted from the qtdesigner file .ui.

Import the packages and parameter classes used. Initialize the parameter class and the button of the message box in the constructor. The reason is that the controls used in the message box are not defined in qtdesigner, so they need to be defined in the constructor.



Use the button method clicked to trigger the connect corresponding method for the button.

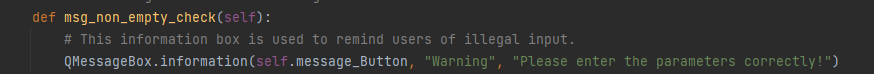


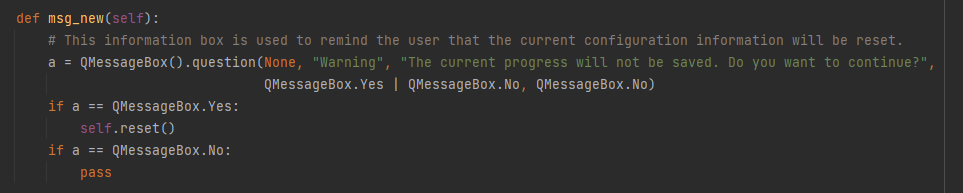
Limit the input type of the input box to avoid entering illegal types (such as: string)



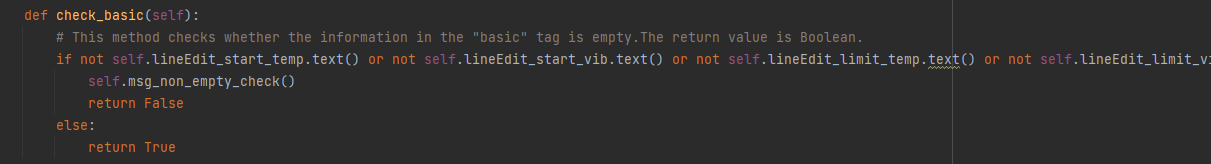
Information box method:

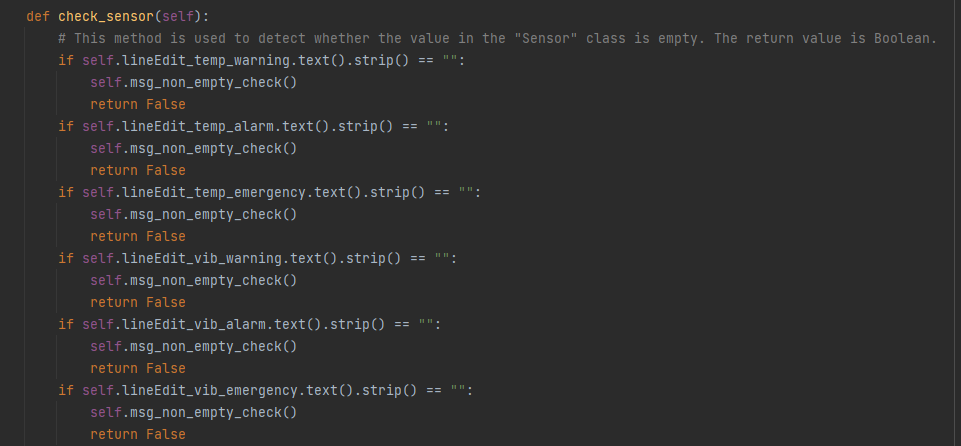
The method creates a new information box. It contains buttons for triggering methods and reminder boxes for untriggered methods. Boxes that need a trigger method need to define a trigger button.



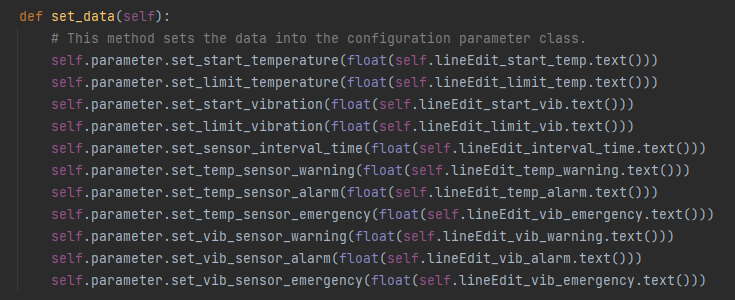


Check the input box to avoid illegal input values due to empty values in the input box.

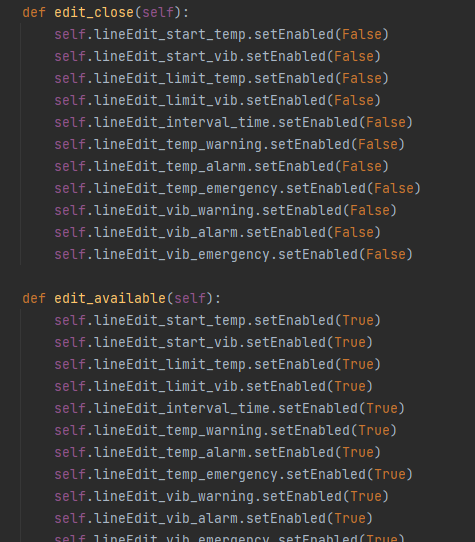




This method converts the characters in the input box to float type and saves it in the parameter class.



The input permission of the input box can be set through the setEnabled method. By closing the input box, avoid misoperations during the simulation progress that will cause the simulation configuration to change.



These five methods are all triggered by clicking the button.

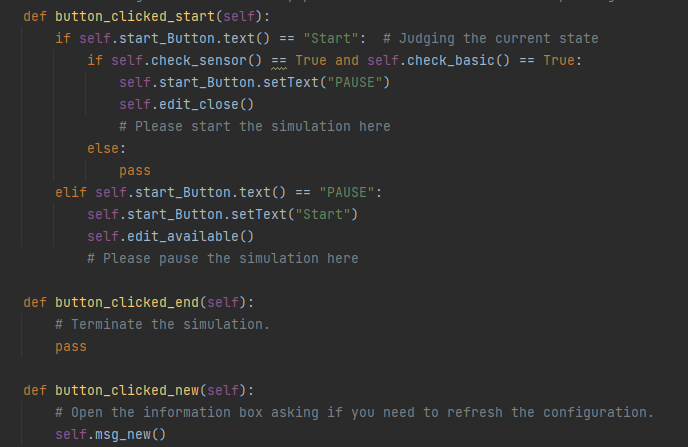
**Start button method:**

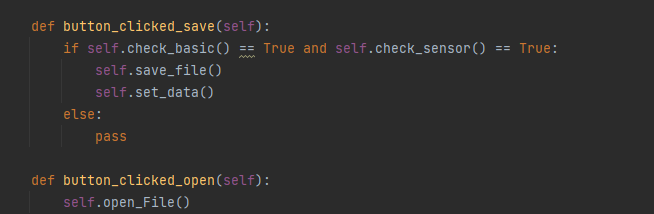
This method will judge the current state based on the button text. The input box will be checked in the Start state. Under the premise that the values in the input box are all legal values, the input permission of the input box will be closed. The button text will change to "pause".

**Pause:**

Since the initial state of the window is "start" when the window is constructed, there is no need to check the state of the input box in this state. After clicking, the input permission of the input box will be opened. The button text will change to "start"

**New, save and open will call methods for saving,** opening files and resetting configuration parameters. The purpose of this design structure is to separate public methods so that methods can be called from other methods.





**Open file method:**

First get the file path through getOpenFileNames in QFileDialog. This method returns the path of the file selected by the user. The parameter "\*.csv" restricts the user to only select csv files.

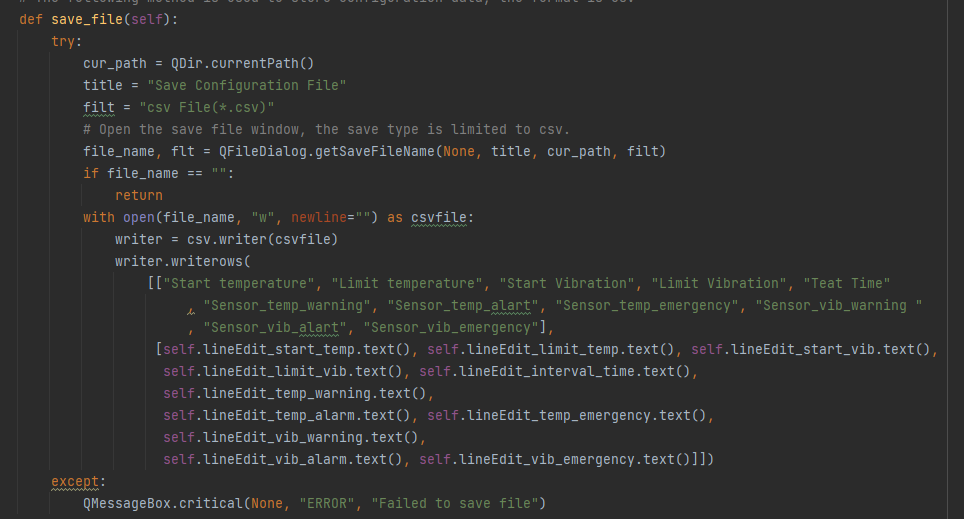
Open the path file through the open method and import it into csv.reader.

Read the second line of configuration parameters, import them into the input boxes, and import them into the parameter class.



**Save csv configuration method:**

This method uses getSaveFileName to specify the save path and uses the open method to open the file. Import into the csv.writer method. And write to csv according to the format



The user runs this method to create a new app window and display the window.

The following statement ensures that the window adapts to the resolution of the current display screen and avoids distortion of the control ratio due to different display screens.

QtCore.QCoreApplication.setAttribute(QtCore.Qt.AA\_EnableHighDpiScaling)

