Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 4, No. 3, 2023

QT and Simulink Mixed Call Implementation

Xingxing Chen

College of Electrical Engineering, Southwest Minzu University, Chengdu 610041, China

Abstract: In order to give full play to the advantages of Qt and Simulink, this paper puts forward a mixed programming method of Qt and Simulink, through the mixed programming technology of VC++ and Simulink, the code generated by Simulink model simulation is called by Qt, and it can be displayed vividly by ui interface. The mixed call of the two can greatly reduce the difficulty of manually writing the algorithm, thus improving the efficiency of the project to a great extent.

Keywords: Qt; Simulink; Mixed Call.

1. Introduction

Qt is an efficient and cross-platform application solution. Qt supports Microsoft operating system, Mac OS and Linux operating system and supports most commercial Unix operating system and Linux embedded operating system [1]. Since its birth as a cross-platform graphical user interface toolkit, Qt has expanded to almost all areas of programming today, including portable devices. Because of the wide adaptability and good portability of Qt, as long as the code has been written once, it can be reused in other different operating system platforms by compiling it again.

Qt is a fully functional, high-performance, multi-platform client / server attached C++ graphical user interface application framework [2]. Qt contains a rich and extensible class library, a powerful GUI layout and form constructor, a set of tools to remove obstacles to international workflows, and a completely customized, reassigned help document or document browser, but it provides limited mathematical functions and is inadequate in dealing with complex mathematical calculations.

Simulink has the advantages of wide adaptability, clear structure and flow, fine simulation, close to reality, high efficiency, flexibility and so on. Based on the above advantages, Simulink has been widely used in complex simulation and design of control theory and digital signal processing. At the same time, there are a large number of third-party software and hardware that can be used or required to be applied to Simulink.

Simulink can be modeled with continuous sampling time, discrete sampling time or two mixed sampling times [3]. It also supports multi-rate systems, that is, different parts of the system have different sampling rates. In order to create a dynamic system model, Simulink provides a graphical user interface to create a block diagram of the model, which can be completed by clicking and dragging the mouse. It provides a faster and more straightforward way, and users can immediately see the simulation results of the system.

Simulink is a multi-domain simulation and model-based design tool for dynamic systems and embedded systems. Simulink provides an interactive graphical environment and a customizable module library for design, simulation, execution and testing of a variety of time-varying systems, including communication, control, signal processing, video processing and image processing systems.

Other products built on top of Simulink extend Simulink's multidomain modeling capabilities and provide tools for design, execution, verification, and validation tasks. Simulink is closely integrated with MATLAB and has direct access to a large number of MATLAB tools for algorithm development, simulation analysis and visualization, batch script creation, modeling environment customization, and definition of signal parameters and test data.

2. The Raising of the Question

The background of the research to realize the mixed call of Qt and Simulink is that the algorithm programming is complex and astringent, which leads to the scarcity of algorithm engineers in the market. The algorithm generation function in Simulink can solve this problem very well, and then use QT language programming design to show the function of the simulation. Therefore, the interface implementation of mixed programming between Qt and Simulink is of great significance to the whole project.

The common practice of mixed programming in multiple languages is to first select a host language according to the needs of the application. Most applications will require to choose a high-level language or application language as the main language to express the top-level structure of the program [1]. Under the Windows system, the project software chooses Qt as the main language program to implement some modules or processes in Simulink.

3. QT and Simulink Mixed Call Implementation

3.1. Simulink Works

1.First, we need to build an atomic subsystem

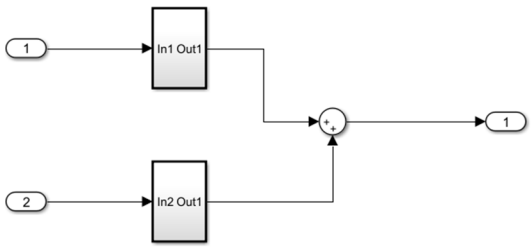

Fig 1. Build an atomic subsystem

2.Select one of the functional modules, right-click, and select Create Subsystem. The operations for the remaining

function modules are the same as above.

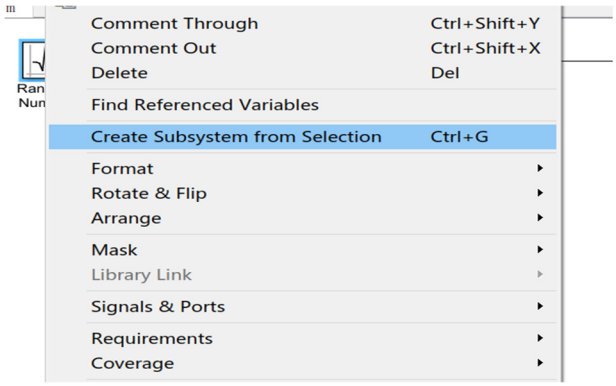

Fig 2. Select one of the functional modules

3.After the creation is completed, we will adjust the subsystem module to the appropriate size, straightening all signal lines, which seems to have no impact on the function, but for complex projects, more modules, more complex functions, this operation will be a very meaningful thing. Next select the Subsystem you are creating, right-click, and select BlockParameters (Subsystem) to create a subsystem.

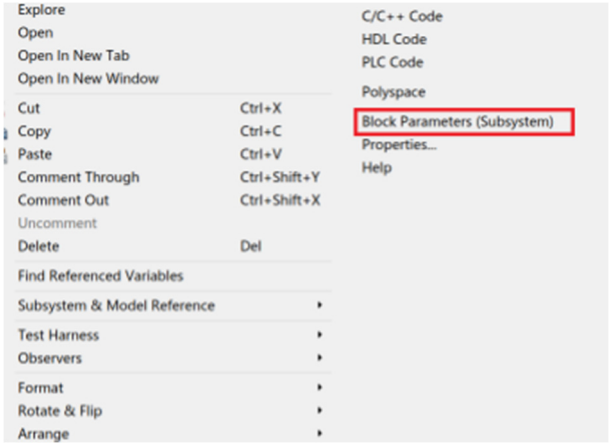

Fig 3. Create a subsystem

4.In the dialog box that is displayed, set the following parameters. Functionpackaging must be set. If it is in the default state, the generated code will not change. Functionname can be set or not set. If Functionname is not set, the system automatically assigns a name, which is basically model name_subsystem. in actual programming, the name of the function is not arbitrary, and the function name should be as good as possible to represent the function of the function.

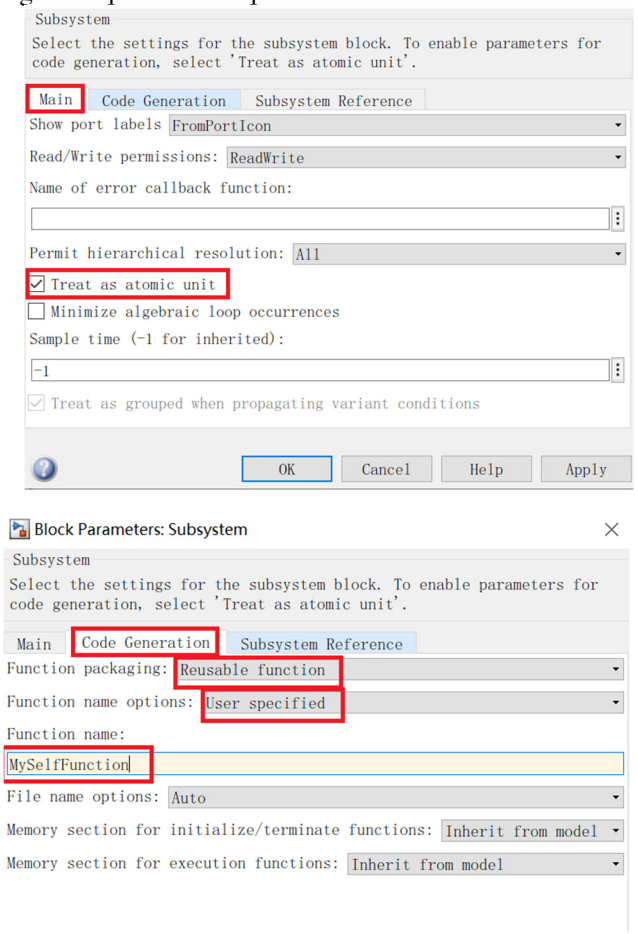

Fig 4. Set the parameters

5.Generate c++ code to change Type under Sloveroptions to Fixed-step.

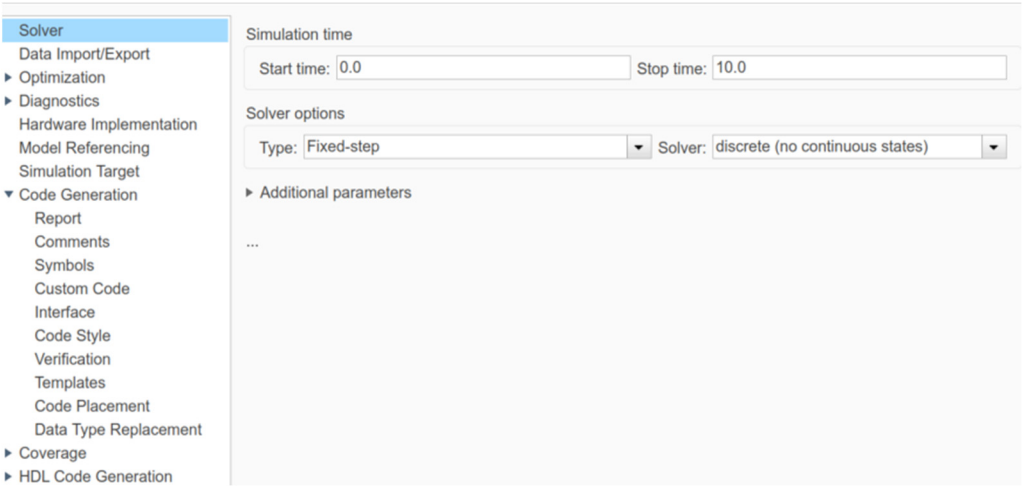

Fig 5. Generate c++ code to change Type under Sloveroptions to Fixed-step

6. In Code Generation, change the System target file under Target selection to ert. tlc, set the language to C++, and check Generate code only under the Build process.

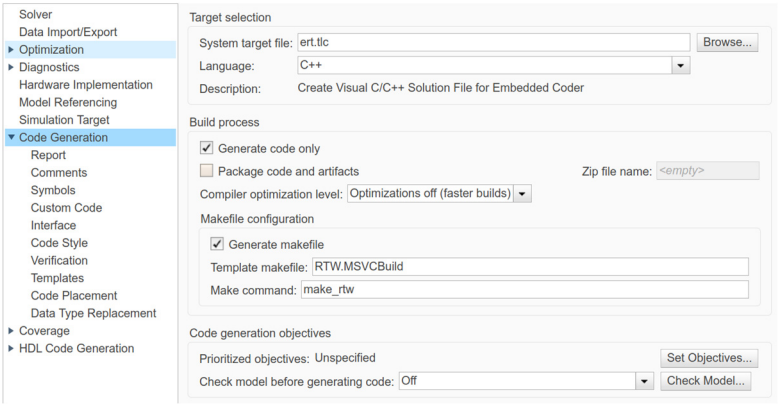

Fig 6. In Code Generation, change the System target file under Target selection to ert. tlc, set the language to C++, and check Generate code only under the Build process

7. Select Create code generation report and Open report automatically in the report

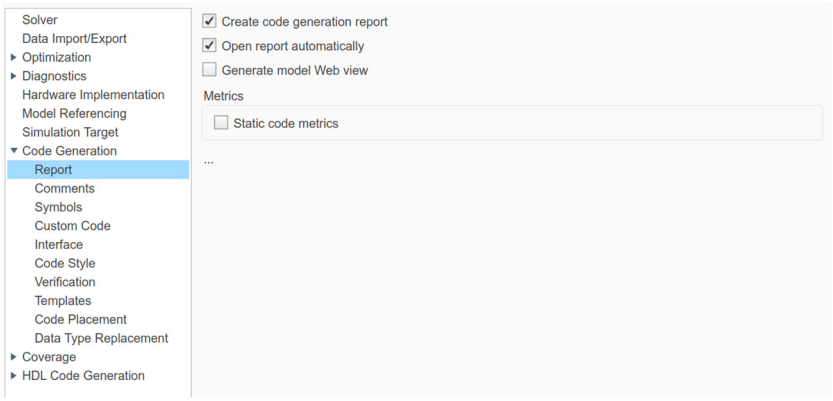

Fig 7. Select Create code generation report and Open report automatically in the report

8.Change the Filepackagingformat to compact in CodePlacement.

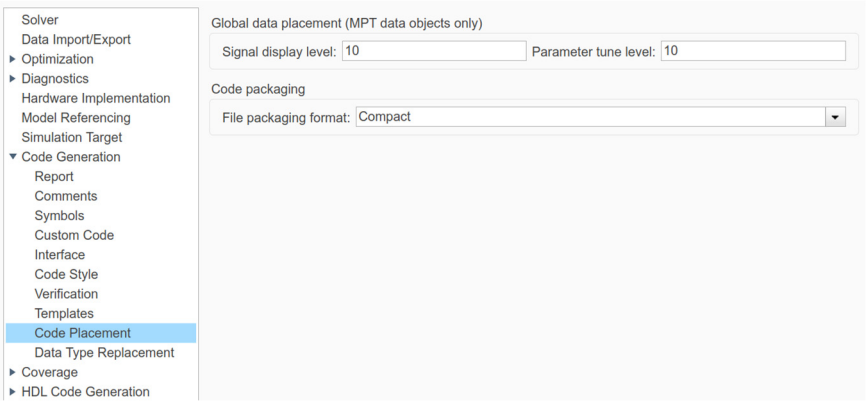

Fig 8. Change the Filepackagingformat to compact in CodePlacement

9.Click Bilud Model or Ctrl + B to generate the code.

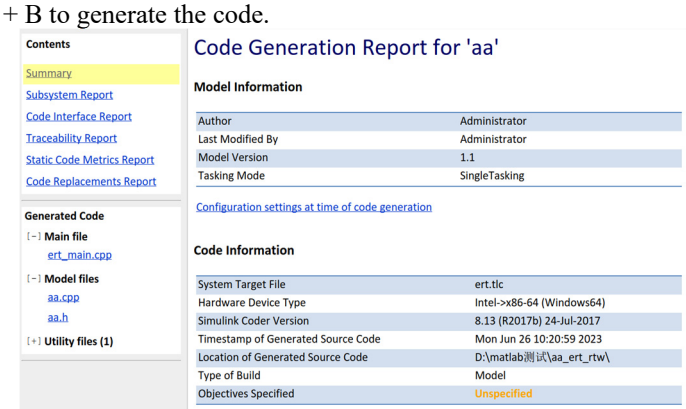

Fig 9. Click Bilud Model or Ctrl + B to generate the code

3.2. Qt Works

1.Create a project in Qt that imports the code generated in simulink

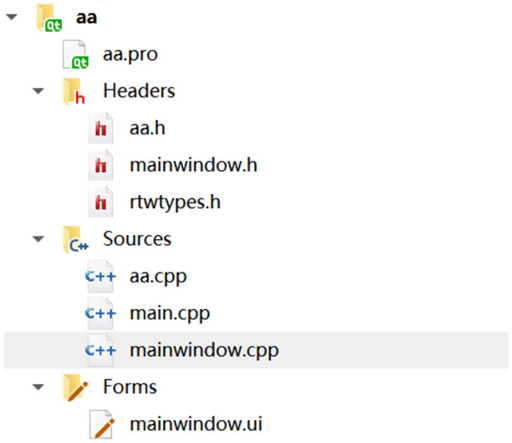

Fig 10. Create a project in Qt that imports the code generated in simulink

2.Write the code in mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include"untitled.h"
#include"rtwtypes.h"

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_pushButton_clicked()
{
   ptest->untitled_U.In1=ui->IN1->text().toInt();
   ptest->untitled_U.In2=ui->IN2->text().toInt();
   ptest->step();
   if(ptest->untitled_Y.Out1>20){
       ui->IN3->setText(QString::number(ptest->untitled_Y.Out1));
   }
   else {
      ui->IN3->setText("wrong");
   }
}
```

Fig 11. Write the code in mainwindow.cpp

3.Layout in mainwindow.ui

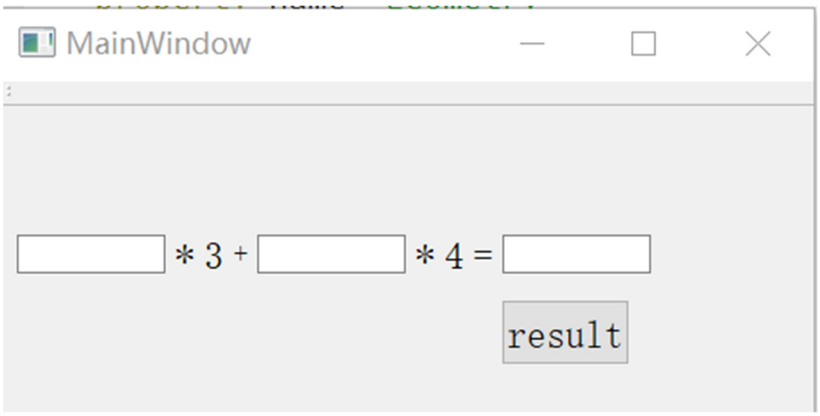

Fig 12. Layout in mainwindow. ui

4. When the calculation result is greater than 20, the normal value is output; when the calculation result is less than 20, the output is wrong, and the result is displayed as follows:

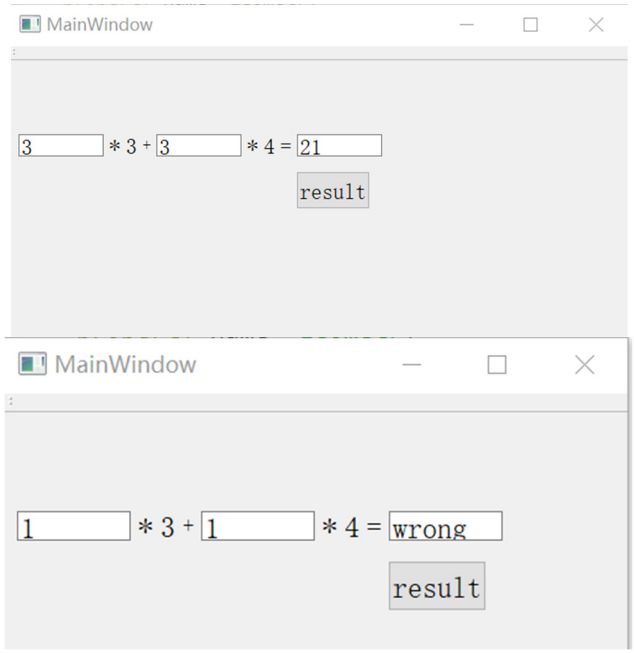

Fig 13. Final result

4. Conclusion

In this paper, the mixed programming method of Qt and Simulink is adopted, which greatly reduces the difficulty of writing algorithms in Qt engineering, and the function of simulation model can be vividly displayed through Qt Ui interface. Through the project practice, the independently callable c++ code generated by calling Simulink has been well applied in the project. Of course, the complexity of Simulink simulation models needs to be further improved to meet the drawing and algorithmic requirements of different projects, and to provide users with a powerful development platform. Due to the excellent cross-platform characteristics of Qt and the algorithm generation function of Simulink, it is believed that the hybrid programming of Qt and Simulink will have a broader application prospect in the future embedded system.

Acknowledgments

This work was financially supported by the Southwest Minzu University Graduate Innovative Research Project No. (YB2023515) fund.

References

[1] Cai Zhiming, Lu Chuanfu, Li Lixia, et al. Proficient in Qt4 programming [M]. Beijing: electronic Industry Press 2008.

[2] Huo Yafei. QtCreator Quick start (2nd Edition) [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 2014.

[3] Liu Wei. Proficient in mixed programming of MATLAB and Chammer + (2nd edition) [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 2008.