Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 2, No. 1, 2022

The Implementation of Mixed Call Between QT and Simulink

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. The Implementation of Mixed Call Between QT and Simulink

3.1. Simulink Works

1. Build a simple simulation model, double-click Gain to set its parameter to K1, double-click Gain1 to set its parameter to K2, and double-click Switch to set its parameter to K3.

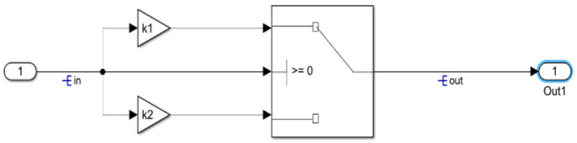

Figure 1. Simulink model

2. Establish a data dictionary, open ModelExplorer- > File- > New- > DataDictionary on the upper toolbar of the model to create a data dictionary. Its name should be the same as the simulink model name, and it should be saved in the same path in the Sminulink file. Then select DesignData under the data dictionary and click Add above to add data objects to the data dictionary.

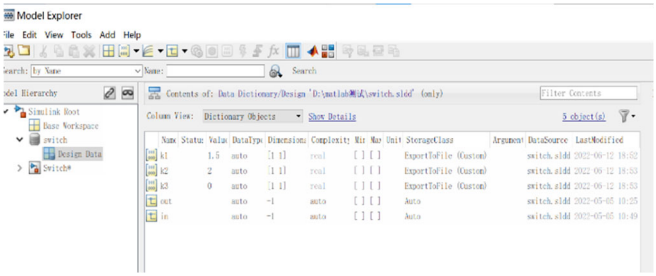

Figure 2. Data Dictionary

Set the Value of K1 to 1.5, the Value of K2 to 2, and the Value of K3 to 0. Then change the StorageClass of K1 and ExportToFile,Customattributes to the name of HeaderFile under the file named p.hscape DefinitionFile as P.C. StorageClass for in and out defaults to Auto.

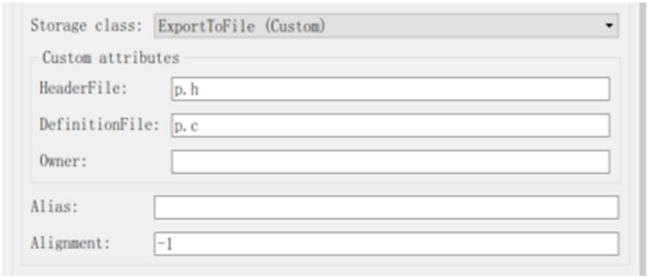

Figure 3. The storage class setting

3. Associated data dictionary

Open Simulink Model-> ModelProperties-> LinktoDataDictionary-> DataDictionary-> Browse, and then select the newly created data dictionary to associate.

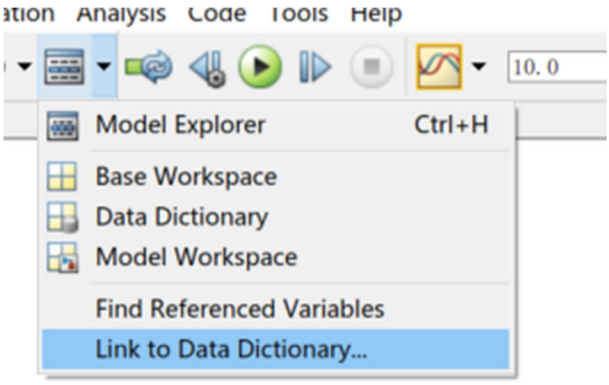

Figure 4. Data Dictionary Association Model

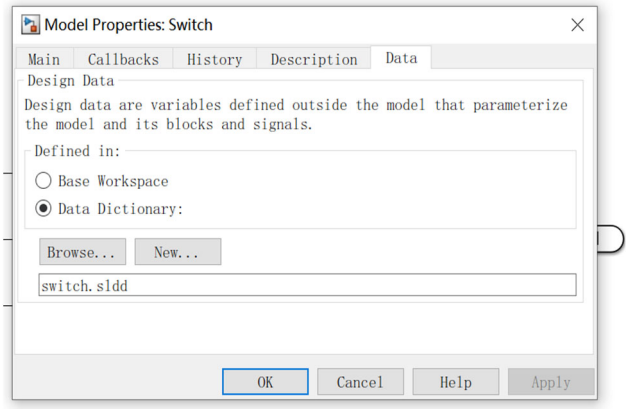

Figure 5. Data Dictionary Association Model

4. Generate C++ code, Change Type under Sloveroptions to Fixed-step,Slover to discrete (nocontinuousstates) in Solver

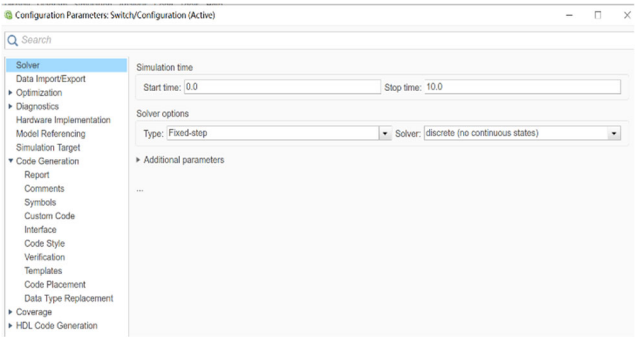

Figure 6. Solver settings

In CodeGeneration, change the Ststemtargetfile under Targetselection to ert.tlc, set Language to clocked, and check the Generatecodeonly under Buildprocess.

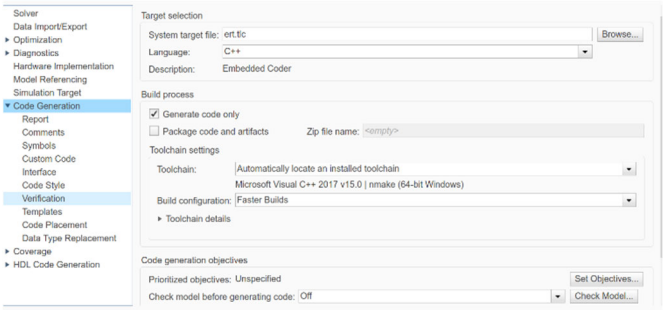

Figure 7. Code generation setup

Check Create code generation report and Open report auto matically in the report.

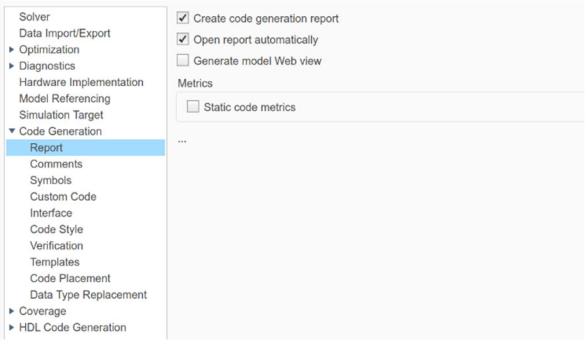

Figure 8. Report setup

Change Filepackagingformat to compact. exe In codeplacement.

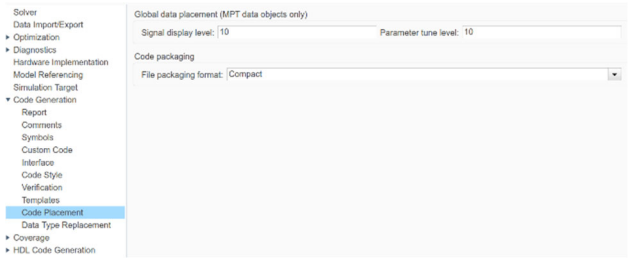

Figure 9. Code placement settings

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

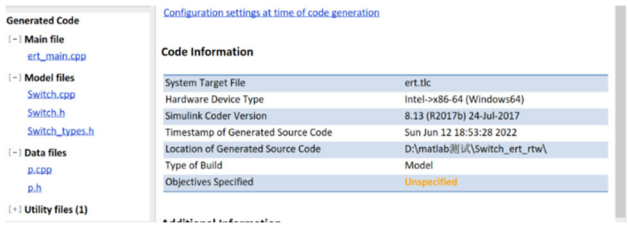

Figure 10. Generates the code

The generated algorithm is as follows.

```
void untitledModelClass::step()
{
  // Switch: '<Root>/Switch' incorporates:
  //   Inport: '<Root>/In1'

  if (Switch_U.in >= k3) {
    // Outport: '<Root>/Out1' incorporates:
    //   Gain: '<Root>/Gain1'

    Switch_Y.Out1 = k1 * Switch_U.in;
  } else {
    // Outport: '<Root>/Out1' incorporates:
    //   Gain: '<Root>/Gain'

    Switch_Y.Out1 = k2 * Switch_U.in;
  }
```

Figure 11. Algorithm function

Enter the data dictionary into the Excel file (for some projects, there is a need for customers to fill in data)

	A	B	C	D	E	F	G	H	I
	Name	Status	Value	DataType	Dimensions	Complexity	Min	Max	Unit
	in		0	auto	-1	auto	[]	[]	
	k1		1.5	auto	[1 1]	real	[]	[]	
	k2		2	auto	[1 1]	real	[]	[]	
	k3		0	auto	[1 1]	real	[]	[]	
	out		0	auto	-1	auto	[]	[]	

Figure 12. Excel table

3.2. Qt Works

1.Set up a project in Qt

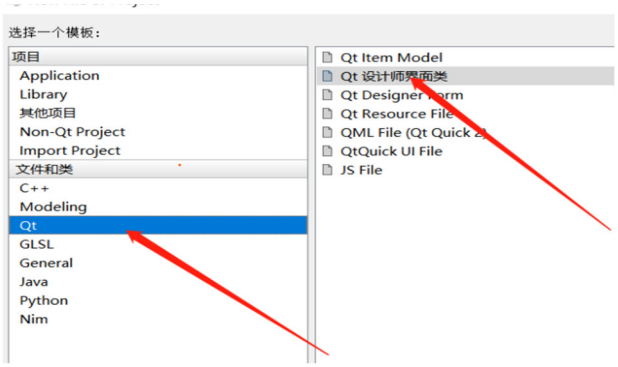

Figure 13. Establishes the QT project

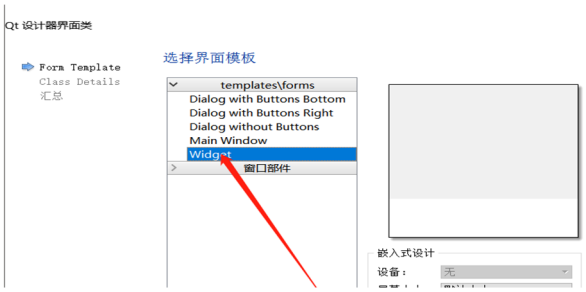

Figure 14. Selection part

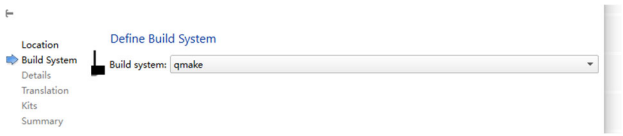

Figure 15. Select build system

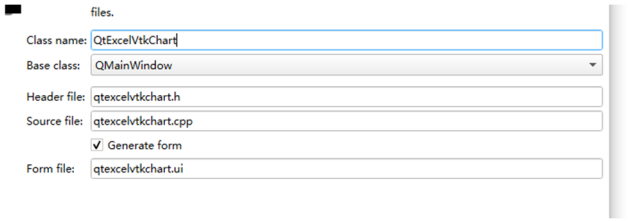

Figure 16. Sets the name

2. Copy the p.cpp p.h Switch.h Switch.cp Switch_types.h file generated by Simulink to the Qt project file, and then add these files to the project directory in Qt.

3. Then assign the values of several linedit controls to K 1, k 2, K 3, in, out.

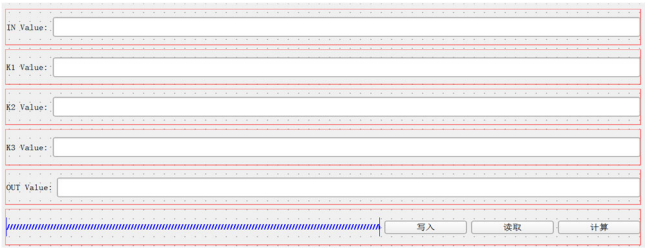

Figure 17. UI design

4. Result

4.1. Write Function

You can synchronize the written data to the Excel table

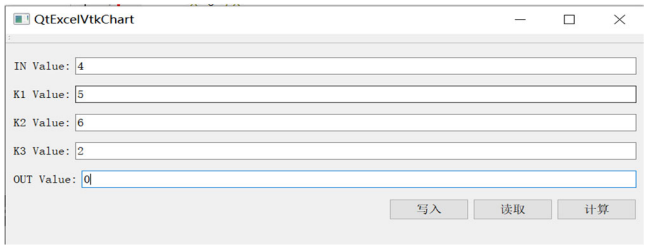

Figure 18. Sets the parameters

Name	Status	Value	DataType	Dimensions	Complexity	Min	Max
in		4	auto	-1	auto	[]	[]
k1		5	auto	[1 1]	real	[]	[]
k2		6	auto	[1 1]	real	[]	[]
k3		2	auto	[1 1]	real	[]	[]
out		0	auto	-1	auto	[]	[]

Figure 19. Saves the parameters

4.2. Read Function

You can read the data in Excel directly

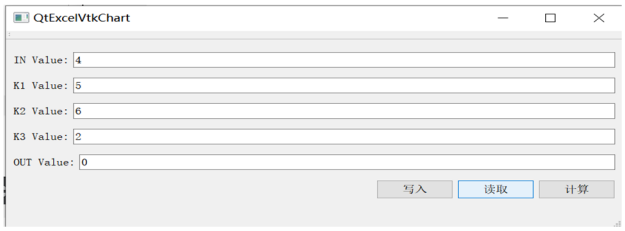

Figure 20. Read function

4.3. Computing Function

By using the algorithm generated by Simulink, the results can be calculated directly. Where the value of the IN,K1,K2,K3 control is adjustable.

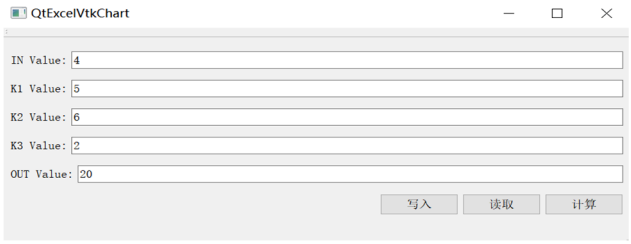

Figure 21. Computational function

5. Conclusion

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

Acknowledgment

This work was financially supported by Xingxing chen 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.

[3] 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.