Rapid Development of Embedded Software Based on Matlab

LIU Yuliang ZHANG Ying XU Guo ZHANG Yong
College of electric information and automation, Tianjin University of Science and Technology
Tianjin 300222, China

Abstract

In order to get high efficiency of code design and convenience of system maintenance, this paper applies the most advanced method, unified modeling language (UML) and model-driven architecture (MDA), on the development of the embedded system based on the MSC1212. Firstly, using Matlab/Stateflow toolbox to model and simulate the complex event driven system in the embedded system introduced in this paper. Then using Stateflow Coder translated the state chart into high efficiency C code. By making some simple replacement to the blank C function with the low-layer function of the target system based MSC1212, the code rapid generation of the embedded software is finished. It has been proved by experiment that, the automatically code generation introduced in this paper works well in the embedded target system. It has good performance and high code efficiency than those code generated by hand.

1. Introduction

Following the raising of complexity of embedded system, the development and maintenance of embedded system have become a very important work. It put forward more requirements to the software engineer. In the development process of embedded software, modern designing idea and advanced development method should be applied. For application software of real-time embedded system, real-time and rapidity of the software development and maintenance is required.

At present, Unified Modeling Language (UML) and Model-Driven Architecture (MDA) have been widely used in the development of embedded system software. Many international well-know companies have designed their own designing toolbox already: UML 2.0 tool Tau Developer designed by Telelogic, the System C Modeler based on UML designed by Mirabilis, the Platform Creation Suite (PCS) aiming at Linux operator system designed by Metrowerks, Simulink/stateflow toolbox designed by Mathworks. These toolboxes all support modeling and simulating in embedded system based on state chart.

This work was supported by Talent Development Foundation of Tianjin University of Science and Technology, Tianjin, China (No.20070439)

But these design tools mentioned above are mostly basing on the primary embedded system with 32-bit control core. The 8-bit chip microcomputer, especially those single chip products of non-famous company, are not supported by toolboxes mentioned above well. This leads to that the unenlightened design methods are still used in the design of low class embedded system based on single chip. The conventional designing flow chart can neither expressing the configuration of embedded software exactly, nor analyzing the logic behavior of embedded software effectually. If the engineer finds error in the program during the software debugging, he must search for errors in hundreds of source code. This delays the coding speed, increases the cost of the project, and prevents us from setting up effective project document. All of this makes the software maintenance mire down in mud.

In this paper we try to applied advanced software design method based on MDA to the exploit of the target embedded system based on MSC1212. Using the Simulink/stateflow toolbox of Matlab, this paper realized high-efficiency coding and convenient system maintenance. Comparing with the code generated by hand of coding efficiency and performing effect was also made. Although the method mentioned in this paper can't achieve automatic coding completely, it has significance in the development of software in embedded system based on single chip.

2. Development Method

2.1 Modeling and Simulating

System simulating toolbox (Simulink) and state flow toolbox (Stateflow) are commonly used in Matlab. One system simulating model may have one or several charts. Usually, one system simulating module has one chart. Every chart has own data, inner input data and output data. Every chart may have one or several active event to enable this chart, and make it run.

In order to simulate the logic action of embedded system, we must set up the simulating model for the

978-0-7695-3288-2/08 $25.00 © 2008 IEEE
DOI 10.1109/ICESS.Symposia.2008.47

software to be simulated first using several state charts in the stateflow toolbox. Fig.1 shows the software model of the embedded system based on MSC1212.

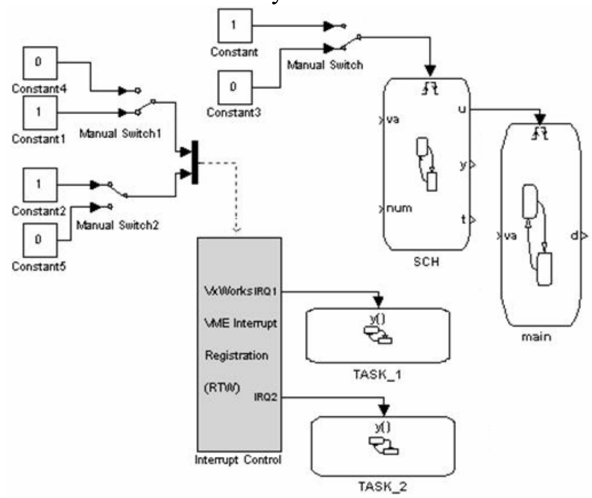

Figure 1. Software model of the embedded system based on MSC1212

The state chart can describe the action process of serial events well. But for the parallel events, the state chart use ‘pseudo’ parallel mechanism to simulate. For embedded system with single chip processor, whose requirement for the performance of real-time is not strict, this mechanism is a feasible scheme.

In the simulation model of embedded system shown in Fig1, SCH chart is an “event” called module, is used to find the event satisfying the condition, active the main chart, and make it run once. Sometimes the main model may respond to several active events. Main chart is one of subsystem of SCH chart to call the function. SCH chart and main chart is both the main program of the embedded system. The realize method of the system interrupt function is connecting “interrupt control” module-INTERRUPT CONTROL with several function calling subsystems TASK_x. Fig1 only shows two modules to give a demonstration. Especially, the function calling subsystem connected with ‘interrupt control’ module is used as interrupt function in the simulation.

After establishing the software simulation model of embedded system using Simulink/Stateflow, the syntax must be checked first, and then debug it by Stateflow Debugger. Here the pattern of “interrupt control” module-Mode should be “the simulation mode-Simulation”, so that the system can be simulated in real-time.

In the process of simulation, we can enter into interrupt service routine by using exterior event to enable “interrupt module” at any time. The method of entering call module-SCH is using dummy event to active it. When event symbol appears, the main chart will be active and the program will run once. Software model of embedded system is independent with platform and object language. After system simulation, the object language of the code and the compilation platform may be chosen.

2.2 Code Generation

After simulation of system software, ANSI-C is chosen as the language of the output code. All of state charts will be translated to ANSI-C code by Stateflow Coder. Generally 2+X *.c files will be generated, where X is the number of calling function subsystems connecting with “interrupt control” module. In order to complete the final embedded software, all of *.c files must be assembled by hand. The program frame to be assembled by hand can be shown by bogus code below.

```
//****** universal frame of program ******
// variable-definition
unsigned char  sfevent;
boolean   event_A，event_B;
…… (Ignored)
// definition constant
CALL_EVENT        (255);
…… (Ignored)
MAIN
{
initialize_MCU()           // law-layer function, compiled by hand
initialize_RAM()           // law-layer function, compiled by hand
sfevent＝CALL_EVENT
// law-layer function，assigned by hand
Loop
sleep（）                  //function resting order
SCH（）                    //call the code of SCH chart
goto loop
}

//****** function frame of SCH chart ******
SCH（）
{
IF（event_A）THEN main（）;
IF（event_B）THEN main（）;
…… (Ignored)
}

//****** frame of interrupt service routine ******
```

Interrupt TASK_x
{
TASK_x（）；
}
… (Ignored)

As for the code generation of embedded system, the reason of not using Real-Time-Workshop (RTW) is that compared with the program generated and assembled by RTW, program generated and assembled by hand has a clear frame and compact structure. This has been proved by test. But research also indicates that the similar interface has been provided by RTW. Embedded the software simulation model for certain single chip target system and low-layer driven function corresponding to special single chip into RTW, the rate of code automatic generation is almost 100%.

As for the program frame to embed the low-layer driver in, blank function can be generated by automatic code generation firstly. Then replace the corresponding empty function with the code of low-layer driving function. The program generated and assembled by hand has more advantages in embedding assembly routine of certain single chip processor into C code.

Keil C is chosen as the software compiler system of target embedded system. Generated project is compiled by Keil C, then complied result is downloaded and carried out in the target embedded system. The running state and the efficiency of code is observed and compared with the code generated by hand.

3. Examples for Design

Using the method mentioned above, target embedded system based on MSC1212 is set up. The function of the system is listed below: After powering on the system, the welcome interface (disp_welcome ()) is displayed, 4 seconds later, the main menu interface (disp_interface ()) is entered. Choose different buttons, different function will be carried out. If button A is chosen, measure state (func ()) is selected. The function will return unit any button is chosen. If button B is chosen, enter save state. The function will return after save. If button C is chosen, enter replay state, The function will return after replay. If button D is chosen, enter set state. Then the beginning wavelength, the ending wavelength and scan time interval will be configured. In the set state, choose button A can set this three values in turn. Choose button B increase 1, choose button C decrease 1, until the set is complete. Choose button D will return set state. Choose button E, return the main menu interface. According to the function mentioned above, the logic of the main chart can be shown in Fig.2. The state charts of scheduler modular-SCH chart and interrupt routine-TASK_x are showed by Fig. 1.

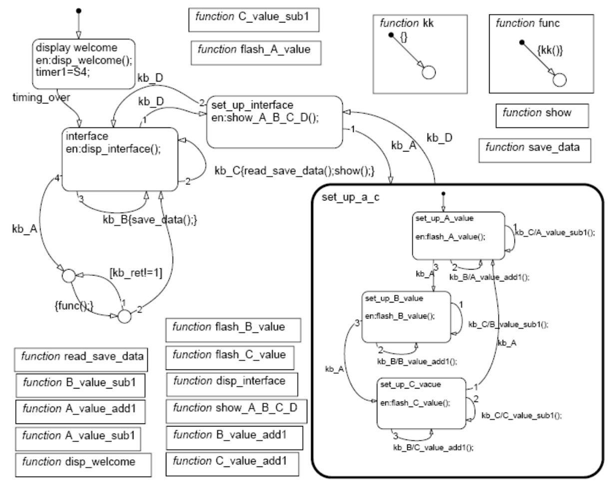

Figure 2. The logic of the main chart of the target embedded system

The system model set up is simulated in the Simulink toolbox. According to the system hint, design parameter will be edited till the result of simulation fulfilled the designing requirement.

Translate all of the state charts by stateflow coder, and several *.c source files are obtained. Last, the final program of embedded system is generated by hand assembling. below several points must be paid more attention:

(1) Interrupt function should be assembly routine and code by hand.
(2) The attribute of chart function in the main chart should be edited after compiling. In order to list generated C code function separately, the default value should be set to function in stead of auto.
(3) In order to use C language variable in the source program of assembled routine, the structural variable of C code generated by Stateflow toolbox should be changed as several independent variables, and defined separately.

As the source code generated by Stateflow Coder is standard C code, it is applicable for every single chip microprocessor which providing C language complier. In the practical application, the source routine can be transplanted in every single chip microprocessor, if only the low-layer driven code according to the company and type of the single chip is changed. Then the problem of the code transplanting between different platforms is resolved at a certain extent.

Compile the C code of the software assembled by hand in the Keil C. After the compiling finished, download it to the single chip microprocessor of target embedded system and run it. All of the requirements of the system function is reached. In the future research, our goal is to adding software simulation model to target system of given single chip by consumer interface in the RTW and embedding low-layer driven function. So it can realize the automatic generation of routine code completely.

Compare with the development period of several days or even tens of days of code generated by hand, the period of code generated by method mentioned in this paper makes the development and maintenance period of the system software short a lot.

4. The Conclusions

This paper has researched on the method of code rapid generation based on the Simulink toolbox and Stateflow toolbox. The conclusion of this paper is:

1 It has been proved that: Compare with code generated by hand, the code generated by method in this paper has more compact frame, more readability, higher efficiency, and easier to maintenance.

2 Making use of advanced software development method based on the MDA to exploit the low-class embedded system, the speed of development is accelerated, the cost of time is cut, the labor force of software engineer is decrease and this method has significance in the application of embedded system.

References

[1] The Unified Modeling Language™. http://www.uml.org/, July 25, 2006.

[2] Model Driven Architecture, http://www.omg.org/mda/. July 25, 2006.

[3] LIU Ran; CHEN Ying; ZHAO Xiao lin, code generation of a visual CASE platform based on UML，Journal of Beijing Institute of Technology，2002, vol22:196-200。(In Chinese).

[4] Bran Selic, Using UML for Modeling Complex Real-Time Systems, Lecture Notes in Computer Science, 1998,V1474, p250-272.

[5] Basso, F.P., Oliveira, T.C., Becker, L.B., Using the FOMDA approach to support object-oriented real-time systems development, proceeding of ISORC2006, Gyeongju, Korea, p:1-8.

[6] Khan, M.U., Geihs, K., Gutbrodt, F., Gohner, P., Trauter, R., Model-Driven Development of Real-Time Systems with UML 2.0 and C,MBD/MOMPES 2006. P:33-42.

[7] Shourong Lu, Halang, W.A., Lichen Zhang,A component-based UML profile to model embedded real-time systems designed by the MDA approach, Proceedings. 11th IEEE International Conference on 2005 P:563-566.

[8] Bastian Florentz, Martin Mutz, Avoiding Unpredicted Behaviour of Large Scale Embedded Systems by Design and Application of Modelling Rules, Model, Design and Validation, 2004. Proceedings, Bretagne,France. Nov. 2004 Page(s):59 – 66.

[9] ZOU Hui; CHEN Wan-chun; YIN Xing-liang, Simulation of a Cruise Missile with Stateflow, Acta Simulata Systematica Sinica, 2004.8, vol.16, p1854-1856.