

Модельно-ориентированное проектирование для генерации кода алгоритма управления двигателем на базе микроконтроллера STM32

П.С. Зубков, Н.А. Кулаев, В.Ю. Артамонов, Н.В. Дубов

Мордовский государственный университет им. Н.П. Огарёва, Саранск

Аннотация: В данной работе рассмотрены этапы генерации С-кода алгоритма управления двигателем с помощью инструментов Embedded Coder и STM32 Embedded Target для Matlab/Simulink. На примере выбранного микроконтроллера STM будут инициализированы его периферийные устройства и подключены к графической модели алгоритма. Помимо этого представлено описание всех блоков и их функции полученной модели. На последнем этапе генерируется код на языке С, готовый для загрузки его в микроконтроллер.

Ключевые слова: Модельно-ориентированное проектирование, Matlab/Simulink, STM32, алгоритм управления, STM32 Embedded Target, Embedded Coder.

При модельно-ориентированном подходе разработчики проходят несколько этапов графического проектирования алгоритма: создание алгоритма организованного как конечный автомат, классификация компонентов алгоритма управления как входов/выходов, локальных и глобальных переменных, тестирование модели в контуре (MIL) алгоритма и моделируемой установки (электропривода). При успешном моделировании установки, логика управления может быть успешно сгенерирована в С-код для интеграции в микроконтроллер [1].

Генерация кода – это способность программного обеспечения автоматически создавать кодовое представление алгоритма. Инструментом, используемым в этом случае, является Embedded Coder для Matlab/Simulink [2]. С учетом модели или подсистемы Simulink, Embedded Coder создаёт несколько файлов, объединённые в один проект с использованием функции C/C++ Code и генерируются 3 файла: 1. rtwtypes.h (содержит определенные аппаратные типы данных); 2. Subsystem.h (в нём находятся определённые типы данных для модели и объявления функции); 3. Subsystem.c (содержит

исходный код модели). Функции вызываются из исходного кода проекта, используя периферийные устройства микроконтроллера в качестве входов/выходов [3].

Предполагается, что генерация кода алгоритма управления и подключение к периферийным устройствам MCU являются отдельными процессами. Тем не менее, STMicroelectronics предоставляет полезный инструмент для выполнения всех частей генерации кода на одной платформе – STM32 Embedded Target для Matlab/Simulink [4]. Этот инструмент предоставляет набор блоков Simulink в Library Browser, который представляет периферийные устройства микроконтроллеров STM32, чьи порты могут быть графически подключены к алгоритму управления в модели Simulink. STM32 Embedded Target связана с Embedded Coder, поэтому весь проект C (алгоритм управления, инициализация периферийных устройств и их подключение) генерируется автоматически и готов к компиляции без необходимости добавления рукописного кода [5-7].

После завершения установки STM32 Embedded Target создается пустая модель Simulink со следующей конфигурацией:

```
> Configuration Parameters
    > Code Generation
        > System target file
            > stm32.tcl
```

Периферийные блоки находятся под меткой Target Support Package – адаптер STM32 Adapter в библиотеке Browser Library. В модель переносится блок STM32_Config, который вызывает файл ioc (приложение STM32CubeMX создает файлы с расширением ioc) для включения блоков Simulink, соответствующих настроенным периферийным устройствам для выбранного микроконтроллера STM32F303RETx Nucleo [8].

Теперь опишем модель Simulink для генерации кода.

В соответствии с проектом выполняются две процедуры обработки прерываний (ISR): ISR таймера для единичной ступенчатой функции алгоритма управления и внешнее прерывание для канала Z энкодера. Блоки Timers и GPIO_Exti заносятся в модель. Сигналы прерывания этих периферийных устройств обновляются на их выходах. Каждый из этих сигналов запускает ISR внутри блока Function-Call (подсистема, запускаемая внешним сигналом) (см. рисунок 1).

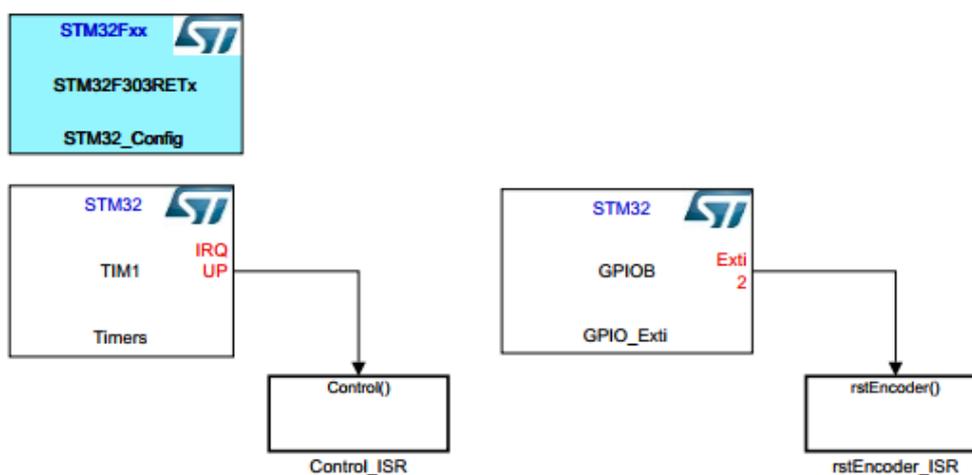


Рисунок 1 – Конфигурация STM32 и обработчик прерываний

Процедура ISR, запускаемая каналом Z, сбрасывает счет таймера TIM4. На рисунке 2 установлен вход Reset_CNT блока Timer (то есть TIM4).

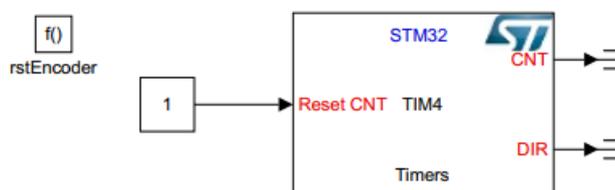


Рисунок 2 – rstEncoder_ISR

ISR, запускаемый TIM1, содержит блок Model Reference, который вызывает алгоритм управления IHz_model.slx (см. рисунок 3).

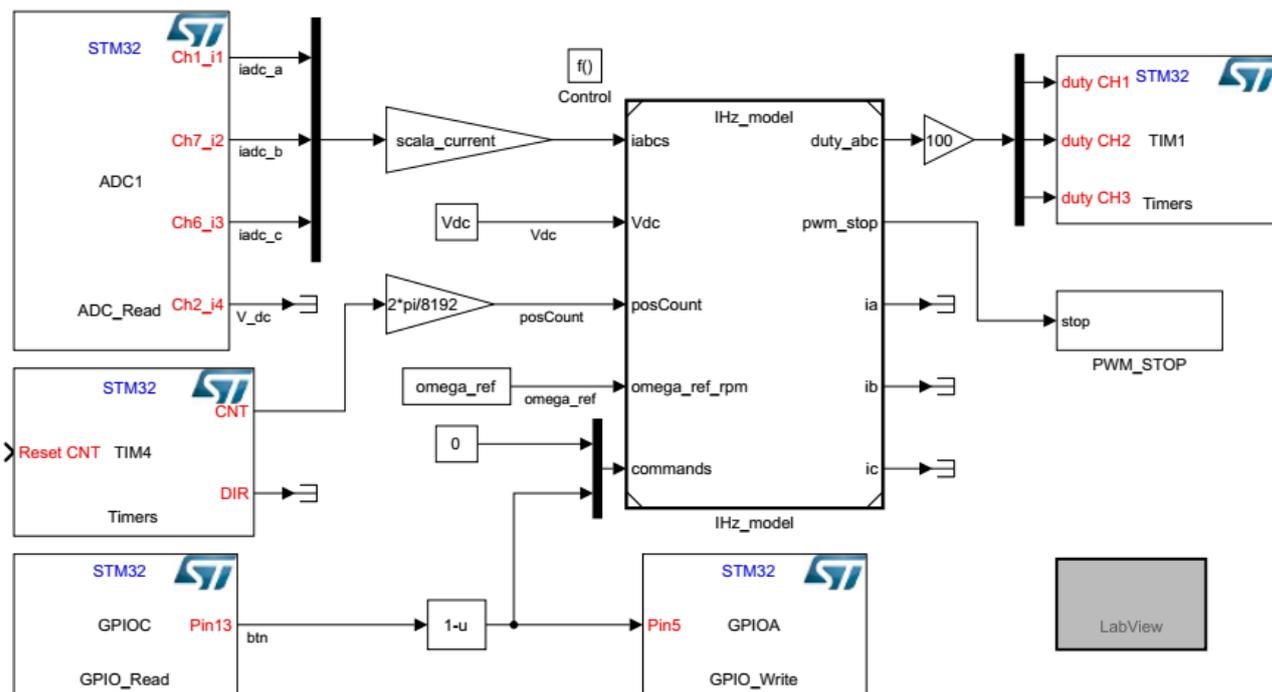


Рисунок 3 – Control_ISR для IHz_model

Этот ISR также содержит следующие периферийные блоки:

- ADC1: вызывается из блока ADC_Read. Он считывает введенные каналы из соответствующих физических входов MCU и дает 12-битное значение. Каналы, соответствующие текущим измерениям, умножаются на коэффициент масштабирования, поэтому они могут рассматриваться как фактические значения токов через порт iabcs алгоритма управления.

- TIM4: вызывается из блока Timers. Он предоставляет подсчитанные импульсы, передаваемые энкодером. Значение счетчика масштабируется, чтобы получить значение угла ротора в радианах.

- GPIOC: вызывается блоком GPIO_Read. Он читает двоичное значение кнопки для выполнения команды Go.

- GPIOA: вызывается блоком GPIO_Write. Включает или выключает светодиод в зависимости от состояния команды Go. Светодиод включен только при нажатии кнопки, в ином случае выключен и необходим лишь для отладки.

- TIM1: вызывается из блока Timers. Он получает значения коэффициента заполнения от 0 до 100, поэтому сигнал ШИМ генерируется в соответствующих физических портах MCU.

- PWM STOP: в зависимости от значения его входа каналы ШИМ включаются/отключаются с помощью блоков операторов if-else. Включение/отключение каналов осуществляется в блоке REGISTER [9, 10].

На заключительном этапе для генерации кода используют Build Model в Simulink. Когда проект будет создан, будет предложено открыть проект C в среде IDE, но поскольку нет необходимости проверки можно сразу выбрать опцию Compile, а затем уже загружать скомпилированные файлы в MCU по протоколу ST-LINK. Тем самым, завершая последний этап генерации алгоритма управления двигателем в C-коде для микроконтроллера.

Литература

1. Model-Based Design // MathWorks URL: mathworks.com/help/simulink/gs/model-based-design.html.
2. Embedded Coder: Generate C and C++ code optimized for embedded systems // MathWorks URL: mathworks.com/products/embedded-coder.html.
3. Legacy Code Integration // MathWorks URL: mathworks.com/help/simulink/legacy-code-integration.html.
4. STM32-MAT/TARGET Hands on // STMicroelectronics URL: st.com/en/development-tools/stm32-mat-target.html
5. Шайкин А. С., Шайкина Е. В. Применение комплекса полунатурного моделирования в процессе проектирования информационно-измерительных и управляющих систем // Инженерный вестник Дона. 2014. №1. URL: ivdon.ru/ru/magazine/archive/n1y2014/2248
6. Moreira O. Rapid Control Prototyping Using an STM32 Microcontroller: Bachelor's degree Clausthal-Zellerfeld, Germany, 2015. 85 p.

7. Кулаев Н.А., Зубков П.С. Математическое описание управления синхронным двигателем с постоянными магнитами для аппаратной реализации на ПЛИС // Инженерный вестник Дона. 2019. №6. URL: ivdon.ru/ru/magazine/archive/n6y2019/6039

8. STM32F398xE advanced ARM-based MCUs // STMicroelectronics URL: st.com/content/st_com/en/search.html#q=STM32F398xE%20-t=resources-page=1

9. Nicolescu Gabriela, Mosterman Pieter J. Model-Based Design for Embedded Systems. CRC Press, 2009. 766 p.

10. STM32Cube initialization code generator // STMicroelectronics URL: st.com/en/development-tools/stm32cubemx.html.

References

1. Model-Based Design // MathWorks URL: mathworks.com/help/simulink/gs/model-based-design.html.

2. Embedded Coder: Generate C and C code optimized for embedded systems // MathWorks URL: mathworks.com/products/embedded-coder.html.

3. Legacy Code Integration // MathWorks URL: mathworks.com/help/simulink/legacy-code-integration.html.

4. STM32-MAT/TARGET Hands on // STMicroelectronics URL: st.com/en/development-tools/stm32-mat-target.html.

5. Shaykin A. S., Shaykina E. V. Inzenernyj vestnik Dona. 2014. №1. URL: ivdon.ru/ru/magazine/archive/n1y2014/2248

6. Moreira O. Rapid Control Prototyping Using an STM32 Microcontroller: Bachelor's degree Clausthal-Zellerfeld, Germany, 2015. 85 p.

7. Kulayev N.A., Zubkov P.S. PLIS / Inzenernyj vestnik Dona. 2019. №6. URL: ivdon.ru/ru/magazine/archive/n6y2019/6039

8. STM32F398xE advanced ARM-based MCUs. STMicroelectronics URL: st.com/content/st_com/en/search.html#q=STM32F398xE -t=resources-page=1.



9. Nicolescu Gabriela, Mosterman Pieter J. Model-Based Design for Embedded Systems. CRC Press, 2009. 766 p.

10. STM32Cube initialization code generator. STMicroelectronics URL: st.com/en/development-tools/stm32cubemx.html.