

## Моделирование периферийных таймеров МК STM32 в MATLAB с возможностью отладки

Современные микроконтроллеры (МК) широко используются в различных областях, от автоматизации процессов до встраиваемых систем. Однако, тестирование и отладка программного обеспечения для МК требуют наличия физического оборудования, что может быть дорогостоящим и неудобным. Эмуляция МК в программной среде, такой как MATLAB, может значительно упростить процесс разработки и отладки, а также позволить тестировать системы в виртуальной среде без необходимости физического устройства.

Уже существуют несколько готовых решений.

QEMU (Quick Emulator) – это универсальный эмулятор, который поддерживает множество архитектур, включая ARM, на которой основаны микроконтроллеры STM32. Он используется для эмуляции не только микроконтроллеров, но и целых компьютерных систем, что делает его очень гибким инструментом. Тем не менее QEMU не моделирует всю периферию. А также он не имеет встроенной поддержки MATLAB и довольно сложен в настройке. Что делает процесс отладки и моделирования еще сложнее.

Proteus – система автоматизированного проектирования. Позволяет моделировать микроконтроллеры с визуализацией схем. Поддерживает интеграцию с различными микроконтроллерами. Но не имеет средств отладки исходного кода, т.к. для эмуляции необходим уже скомпилированный файл. Proteus также не моделирует всю периферию.

Как видно существующие инструменты для эмуляции и моделирования МК не всегда обладают достаточной функциональностью для полноценного тестирования и отладки. Например, некоторые эмуляторы могут не обеспечивать моделирование необходимых периферийных устройств. Или же они не интегрируются с популярными средами разработки, такими как MATLAB. Это создает трудности для разработчиков, которым необходимо тестировать приложения в виртуальной среде.

В статье предлагается решение в виде эмулятора для МК, интегрированного в MATLAB. Предложенный эмулятор выполняет программу МК в MATLAB, а также симулирует периферию МК. Основное отличие от существующих решений заключается в возможности отладки кода на языке C, а также пользовательская реализация периферии. Это позволяет делать упрощенные эмуляции периферии, если это не влияет на приложение МК.

Целью исследований является разработка и демонстрация работы эмулятора микроконтроллера STM32F407 и его периферийных таймеров в MATLAB.

Для проведения исследований использованы следующие методы и инструменты:

- MATLAB/Simulink для создания и интеграции эмулятора.

- для отладки программы МК в Simulink и компилятор Microsoft Visual C++ (MSVC) для компиляции программы МК для MATLAB.

- стандартные библиотеки и драйверы для STM32, адаптированные для MATLAB.

Программы на большинство МК пишутся на языке программирования C. В MATLAB существует блок S-Function, который представляет собой функцию, написанную на языке C/C++ или MATLAB [1]. Поэтому было решено использовать именно S-Function для портирования программы в MATLAB. Компиляция кода на языке C/C++ происходит с помощью компилятора MSVC. Данный компилятор является сторонним, но он имеет возможность интеграции в MATLAB.

Основная сложность моделирования программы МК STM32 в MATLAB заключается в различиях принципа работы программ на МК и в среде симуляции MATLAB. В общем случае, программа на микроконтроллере представляет собой бесконечный цикл. Каждую итерацию бесконечного цикла, программа МК считывает порты ввода-вывода, обрабатывает данные, и формирует выходной сигнал на портах ввода-вывода.

S-Function в MATLAB представляет собой функцию, которая вызывается на каждом шаг симуляции. Эта функция принимает входные данные, и выполняет некоторые расчеты. В это время MATLAB ожидает завершения функция, и только после этого формируются выходные данные и симуляция переходит на следующий шаг. Поэтому, если S-Function будет включать бесконечный цикл – симуляция просто зависнет.

Также, микроконтроллеры имеют доступ к периферийным устройствам, которые реализованы аппаратно. Их программы могут взаимодействовать с такими устройствами, как таймеры, АЦП, или UART через регистры периферии и прерывания [2]. В MATLAB программы не имеют никакой периферии. Поэтому, например, если программа в MATLAB должна ожидать выставление флага переполнения у таймера, то симуляция также зависнет, т.к. этот флаг аппаратно не выставится.

В связи с этим, для симуляции МК в MATLAB необходимо:

- контролировать ход программы МК, чтобы в определенный момент завершить её выполнение и перейти на следующий шаг симуляции;

- реализовать эмулятор периферии, который будет эмулировать периферию и её регистры, чтобы программа МК могла полноценно работать.

Для контроля ходом программы МК, было решено поместить её в отдельный поток выполнения. Этот поток возобновляется в начале каждого шага симуляции и выполняется в течение определённого времени, после чего приостанавливается, позволяя S-Function завершиться. Такой подход позволил создать модель, где программа микроконтроллера может

выполняться вместе с основным процессом симуляции, не зависая при ожидании данных от периферийных устройств, которые в реальном МК управляются аппаратно.

Эмуляция периферийных устройств - второй важный аспект, который требует детальной проработки. Поскольку микроконтроллеры взаимодействуют с аппаратными компонентами через регистры и прерывания, возникла необходимость имитировать их работу в MATLAB.

Во-первых, все регистры в МК привязаны к определенным адресам, которые недоступны программе в MATLAB. Поэтому надо выделить отдельную область оперативной памяти в программе, чтобы там хранить регистры периферии.

Во-вторых, отсутствие аппаратной периферии вынуждает писать модели для используемой периферии вручную. Можно максимально подробно прописать логику периферии, чтобы она полностью соответствовала реальной периферии МК. Или же можно написать упрощенную или абстрактную модель, которая не будет имитировать всех процессов в периферии, но её будет достаточно для симуляции программы.

В данной статье для примера будет рассматриваться программа управления двигателями. В ней активно используются два периферийных устройства: таймеры и интерфейс UART.

Таймеры, которые используются для формирования ШИМ, должны быть эмулированы максимально точно. Это связано с тем, что они играют ключевую роль в управлении двигателем и точность их работы существенно влияет на результат. Поэтому для их эмуляции была разработана модель, в которой прописана вся основная логика таймера: режимы счета, каналы захвата/сравнения, вывод каналов на порты ввода/вывода. Такой подход позволяет быстро менять настройки ШИМ и по симуляции смотреть, как это будет влиять на устройство, без необходимости переписывать эмулятор из-за смены параметров таймера.

Интерфейс UART используется для приема параметров ШИМ по протоколу MODBUS. В симуляции работа с этим интерфейсом была упрощена до считывания входов S-Function и записи напрямую в регистры Modbus. Это позволяет программе функционировать корректно и без необходимости моделировать всю сложную логику интерфейса и протокола. Это упрощение было сделано, чтобы снизить нагрузку на эмулятор, так как управление двигателем не требует точного воспроизведения работы UART.

На рис. 1 приведена модель в Simulink для демонстрации работы эмулятора.

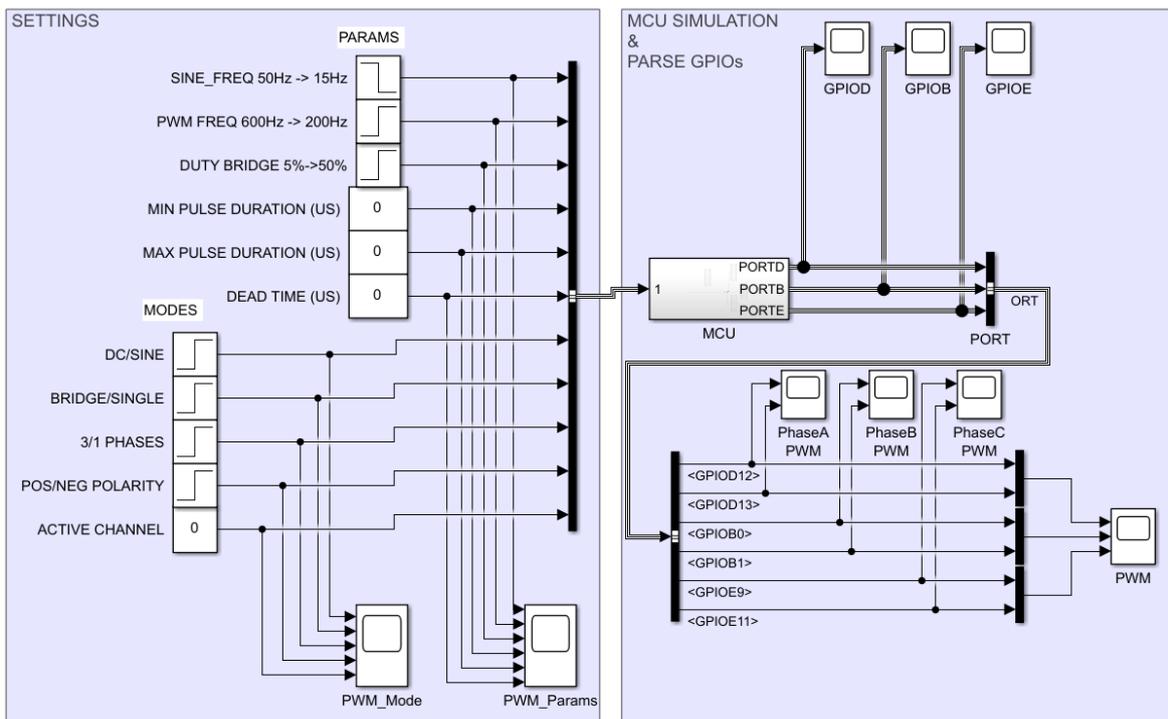


Рисунок 1 – Модель Simulink для демонстрации эмулятора МК

На рис. 2 приведен результат моделирования программы МК для управления двигателями. Программа формирует ШИМ в соответствии с разными заданными параметрами, которые передаются в модель МК. Более подробно симуляция ШИМ рассмотрена на рис. 3-6.

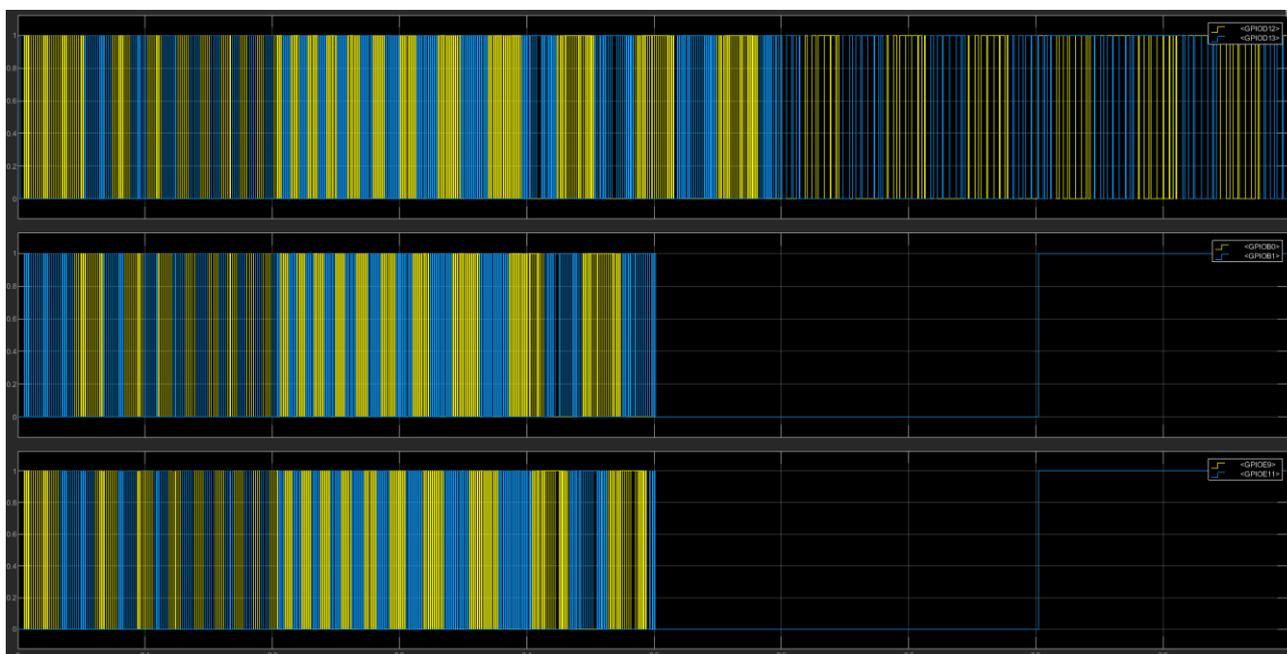


Рисунок 2 – Демонстрация симуляции МК в MATLAB

На рис. 3 приведен фрагмент симуляции трехфазного ШИМ с постоянным заполнением 5%, частотой ШИМ – 600 Гц, частота огибающей – постепенно растет от 0 Гц до 50 Гц.

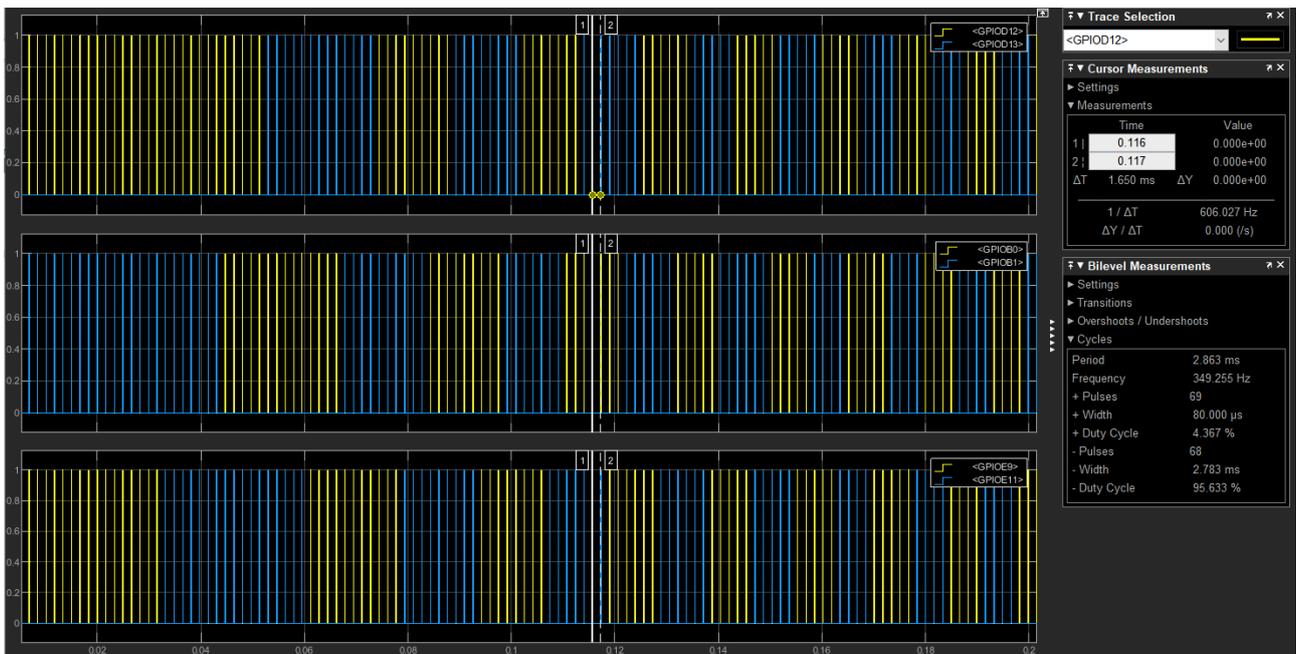


Рисунок 3 – Демонстрация симуляции трехфазного постоянного ШИМ с нарастанием частоты

На рис. 4 приведен фрагмент симуляции трехфазного ШИМ с постоянным заполнением 5%, частотой ШИМ – 600 Гц, частота огибающей – постепенно убывает от 50 Гц до 15 Гц.

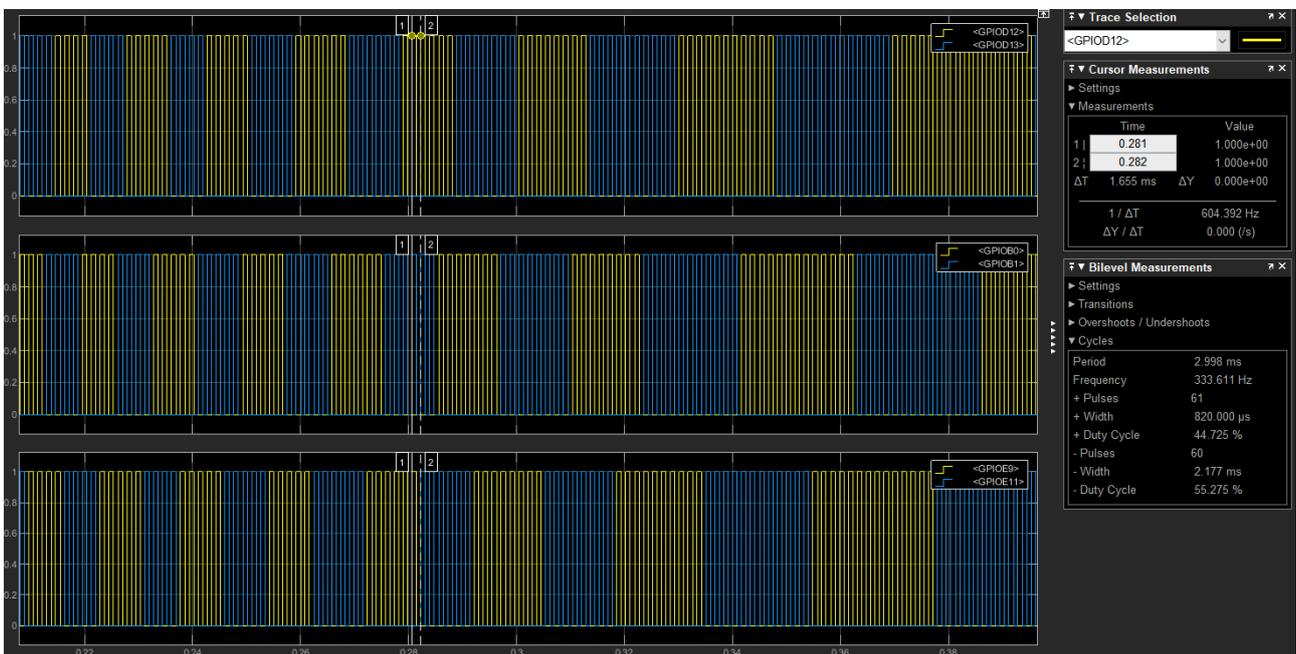


Рисунок 4 – Демонстрация симуляции трехфазного постоянного ШИМ с убыванием частоты

На рис. 5 продемонстрированы режимы генерации ШИМ с переменной скважностью по закону синуса, где положительная/отрицательная полуволны выведены на разные каналы.

Сначала формируется трехфазный синусный ШИМ, потом однофазный. Частота ШИМ – 600 Гц, частота огибающей – 15 Гц.

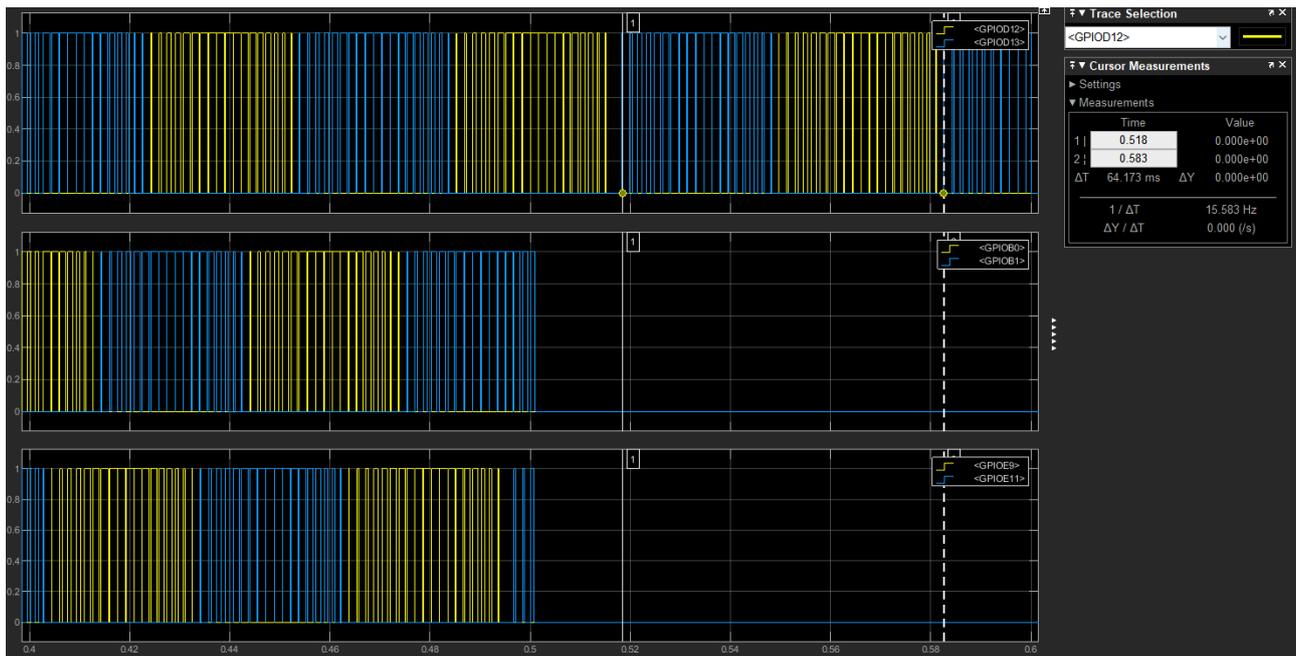


Рисунок 5 – Демонстрация симуляции трехфазного и однофазного синусного ШИМ

На рис. 6 продемонстрированы режимы генерации синусного ШИМ только на одной фазе. Сначала генерируется синусный ШИМ с положительной полярностью, а потом с полярность меняется на противоположную. Полярность остальных фаз также меняется, Частота ШИМ – 200 Гц, частота огибающей – 15 Гц

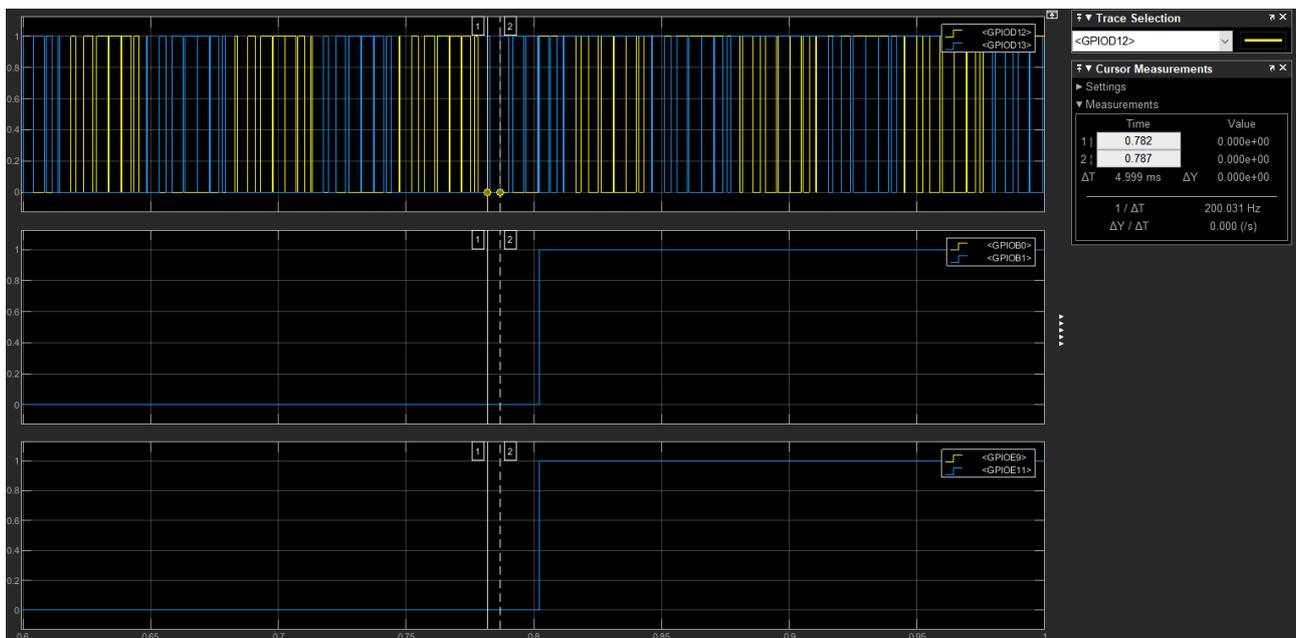


Рисунок 6 – Демонстрация смены полярности в симуляции однофазного синусного ШИМ

Компилятор MSVC позволяет скомпилировать код для отладки, следовательно появляется возможность отладки программы МК в MATLAB. Для этого необходимо открыть

папку с исходными файлами в среде разработки, например, Visual Studio. Далее перед запуском симуляции подключиться к процессу MATLAB. После этого при запуске симуляции появиться возможность отладки программы, т.е. возможность смотреть текущие значения всех доступных переменных и выполнять программу пошагово.

Разработка эмулятора микроконтроллеров для MATLAB продемонстрировала его эффективность в моделировании и тестировании программного обеспечения. Эмулятор позволяет проводить отладку и тестирование без необходимости физического устройства, что упрощает процесс разработки. Результаты моделирования совпадают с экспериментальными:

- ШИМ корректной симулируется с заданной частотой 600 или 200 Гц (рис. 3, 4, 6);
- постоянная скважность ШИМ корректно симулируется в соответствии с заданной скважностью 5% или 50% (рис. 3, 4);
- переменная скважность ШИМ симулируется корректно (рис. 5, 6);
- частота вращения трехфазного ШИМ корректно симулируется в соответствии с заданной частотой 15 Гц (рис. 5).

Цель исследования была достигнута, был создан функциональный эмулятор периферийных таймеров МК STM32, который позволяет моделировать и тестировать программное обеспечение для МК в среде MATLAB. В дальнейшем планируется расширение функциональности эмулятора для поддержки большего количества моделей микроконтроллеров, периферийных устройств и более комплексных программ МК (например, содержащих FreeRTOS).