

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Системное программирование

Приходько Станислав Витальевич

Поддержка программирования микроконтроллера STM32 в TRIK Studio

Выпускная квалификационная работа

Научный руководитель:
к.т.н., доцент кафедры системного программирования Литвинов Ю.В.

Научный консультант:
ст. преп. кафедры системного программирования Кириленко Я.А.

Рецензент:
к.ф.-м.н., доцент Рядчиков И.В.

Санкт-Петербург
2017

SAINT PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems

Software Engineering

Stanislav Prikhodko

Support of STM32 microcontroller in TRIK Studio

Graduation Project

Scientific supervisor:
C.Sc., Associate Professor Yurii Litvinov

Scientific consultant:
Senior Lecturer Iakov Kirilenko

Reviewer:
C.Sc., Associate Professor Igor Ryadchikov

Saint Petersburg
2017

Оглавление

Введение	4
1. Постановка задачи	6
2. Обзор	7
2.1. STM32	7
2.1.1. Семейство микроконтроллеров STM32	7
2.1.2. Плата микроконтроллера STM32F429I-Discovery .	8
2.2. Средства программирования микроконтроллера STM32 .	10
2.2.1. Цели и критерии обзора	10
2.2.2. IAR Embedded Workbench for ARM	10
2.2.3. Keil MDK	11
2.2.4. Eclipse ARM plugin	12
2.2.5. Atollic TrueStudio	13
2.2.6. Coocox	13
2.2.7. Выводы	14
2.3. TRIK Studio	15
2.3.1. Общая архитектура	15
2.3.2. Генераторы	16
3. Архитектура системы	18
4. Плагин для микроконтроллера STM32 в TRIK Studio	21
5. Генератор кода на языке C++	22
6. Интеграция компилятора и загрузчика с TRIK Studio	23
7. Апробация	25
Заключение	30
Список литературы	31

Введение

Во многих сферах деятельности для выполнения разнообразных задач внедряются роботы, которые способны заменить человеческий труд машинным. В связи с этим масштабнее становится развитие робототехнической отрасли, одним из направлений деятельности которой является создание программных средств, которые дадут возможность пользователям удобно и быстро программировать различных роботов.

Одним из средств, способных упростить и ускорить процесс создания программ, является визуальное программирование [24]. Визуальное программирование — это способ задавать программу в терминах графических объектов вместо привычного для многих программистов применения абстракции текстового программирования. Данный способ позволяет упростить представление объектов, с которыми приходится оперировать программисту при разработке программы, что может сэкономить время, затрачиваемое на создание программного продукта, и упростить процесс обучения программированию, тем самым расширив аудиторию разработчиков программного обеспечения.

На кафедре системного программирования Санкт-Петербургского государственного университета в течение нескольких лет ведется разработка платформы QReal [27]. QReal представляет собой metaCASE-инструментарий, то есть среду для создания новых визуальных языков и интегрированных сред программирования для них. Визуальное программирование можно использовать в качестве средства управления роботами [22], в связи с чем на основе среды QReal [15] была создана технология визуального программирования роботов TRIK Studio [25], которая позволяет создавать графические программы для различных роботов и исполнять их.

Для управления разнообразными электронными устройствами, в том числе роботами, используются определенные микросхемы — микроконтроллеры. Микроконтроллер представляет собой вычислительное устройство небольших размеров, которое содержит в себе одно или несколько процессорных ядер, память, программируемые устройства

ввода/вывода и прочие периферийные устройства.

С недавнего времени в робототехнической отрасли возрос интерес к относительно молодой серии микроконтроллеров STM32. Широкое распространение данная линейка получила благодаря своим преимуществам по сравнению с аналогами: высокой производительности, относительно низкой стоимости и работе в условиях низкого электропотребления.

В связи с этим с целью расширения возможностей среды TRIK Studio была поставлена задача осуществить поддержку программирования микроконтроллера STM32 в TRIK Studio.

1. Постановка задачи

Цель данной работы заключается в том, чтобы получить возможность создавать программы для микроконтроллера STM32 в TRIK Studio.

Для достижения поставленной цели нужно:

- разработать архитектуру системы программирования микроконтроллера STM32 в среде TRIK Studio;
- реализовать подключаемый модуль для микроконтроллера STM32 в TRIK Studio;
- реализовать генератор кода из программы на языке визуальных диаграмм в программу на языке C++;
- разработать подсистему интеграции компилятора и загрузчика программы с TRIK Studio;
- провести апробацию средства программирования на плате микроконтроллера STM32F4-Discovery.

2. Обзор

2.1. STM32

2.1.1. Семейство микроконтроллеров STM32

STM32 — это семейство 32-битных микроконтроллеров от компании STMicroelectronics [21]. Микроконтроллеры STM32 сгруппированы в серии, основанные на ядрах 32-битных ARM процессоров серии Cortex-M. На уровне ядра процессора работа с микроконтроллером происходит в основном с использованием языков C, C++, также используется язык ассемблера.

ARM [1] — это архитектура процессоров, в основе которой лежит концепция RISC (Reduced Instruction Set Computer). Данная стратегия организации вычислительной системы предполагает использование упрощенного набора инструкций, то есть каждая отдельная операция процессора выполняет фиксированный небольшой объем работы, например чтение данных из памяти в регистр, запись данных из регистра в память, операции над данными, содержащимися в регистрах (арифметические, логические и т.д.) и прочие. Как правило инструкции RISC-процессоров имеют фиксированную длину и простой формат кодирования, что упрощает выборку и декодирование команды. Упрощение набора команд позволяет увеличить производительность системы.

Cortex-M представляет собой серию 32-битных ядер ARM процессоров, которые в основном используются для работы в микроконтроллерах. В серию входят ядра Cortex-M0, Cortex-M0+, Cortex-M1, Cortex-M3, Cortex-M4(F), Cortex-M7(F), Cortex-M23, Cortex-M33(F). Приставка «F» указывает на наличие модуля операций с плавающей точкой (FPU — Floating Point Unit), однако в некоторых комплектациях его может и не быть.

Ядра Cortex-M0, Cortex M0+ и Cortex-M23 отличаются небольшим размером и относительно низкой ценой. Ядра Cortex-M3, Cortex-M4 и Cortex-M33 поддерживают оптимальный баланс между производи-

ностью и низкими энергозатратами. Ядро Cortex-M7 предназначено для обеспечения функционирования систем, требующих высокой производительности.

Ядра ARM Cortex-M4, Cortex-M7 и Cortex-M33 поддерживают обработку цифровых сигналов и операции с плавающей точкой, что обеспечивает высокую производительность и энергоэффективность вычислений на данных процессорах.

2.1.2. Плата микроконтроллера STM32F429I-Discovery

На рисунке 1 представлено изображение платы микроконтроллера STM32F429I-Discovery [20], которая использовалась в рамках данной работы для апробации. Плата содержит следующие компоненты:

- микроконтроллер STM32F429ZIT6 с ядром Cortex-M4F с 2 Мб флэш-памяти, 256 Кб оперативной памяти (RAM) в корпусе LQFP144;
- программатор ST-LINK/V2;
- питание платы через USB или от внешнего источника 3V или 5V;
- жидкокристаллический дисплей 2.4”;
- 64 Мб SRAM;
- датчик движения ST-MEMS L3GD20 и цифровой выход трехосевого гироскопа;
- шесть светодиодов:
 - LD1 (красный/зеленый) для подключения USB;
 - LD2 (красный) для питания 3.3 V;
 - два пользовательских светодиода: LD3 (зеленый) и LD4 (красный);

- два светодиода для USB OTG¹: LD5 (зеленый) и LD6 (красный).
- две кнопки: пользовательская и кнопка перезапуска;
- USB OTG с разъемом micro-AB.

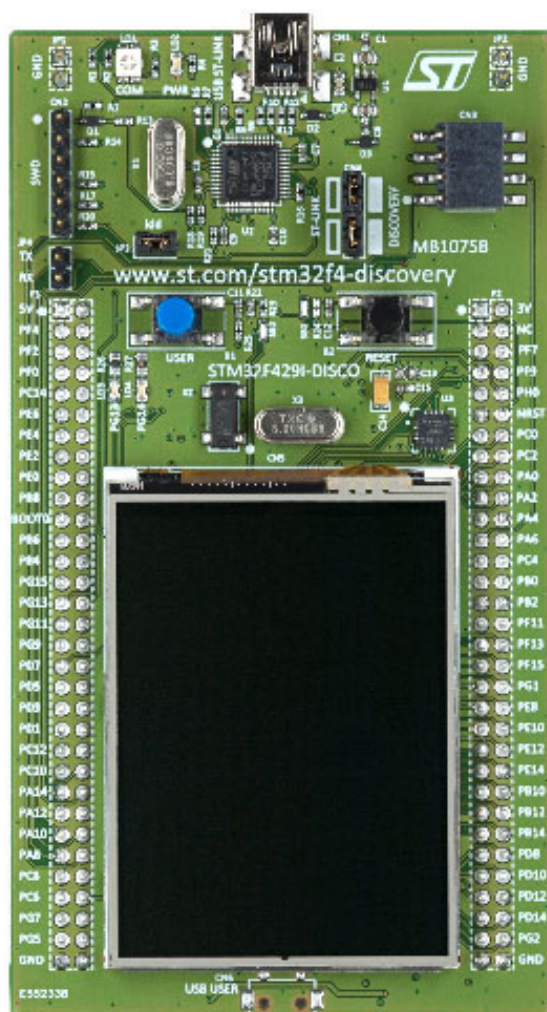


Рис. 1: Плата микроконтроллера STM32F429I-Discovery. Изображение взято из источника [20].

¹USB OTG — это расширение стандарта USB 2.0 для установки связи между периферийными устройствами напрямую без устройств-посредников.

2.2. Средства программирования микроконтроллера STM32

2.2.1. Цели и критерии обзора

Данный обзор посвящен сравнительному анализу существующих средств, предоставляющих возможность создавать программы для микроконтроллеров STM32, с целью рассмотреть уже достигнутые результаты в области программирования плат серии STM32, изучить способы организации таких инструментариев и обосновать целесообразность разработки средств поддержки STM32 в TRIK Studio.

В качестве критериев для сравнения были выбраны: во-первых, свободное распространение продукта, т.к. в рамках исследовательской деятельности желательно работать с инструментами, предполагающими открытое и свободное распространение программного обеспечения; во-вторых, кроссплатформенность средства, т.к. для универсального использования необходимо иметь инструментарий, который будет доступен на всех основных операционных системах, таких как Windows, Linux, MacOS; и, в-третьих, поддержка визуальных языков программирования для создания программ, т.к. есть желание применить методологию визуального программирования к созданию программ для микроконтроллера STM32.

2.2.2. IAR Embedded Workbench for ARM

IAR Embedded Workbench for ARM [11] представляет собой многофункциональную среду разработки приложений для целого ряда микроконтроллеров под управлением процессоров ARM. Разработка программы может вестись пользователем на языках C, C++ и языке ассемблера.

Среда включает в себя следующие инструменты: компилятор C/C++, библиотеки C/C++, транслятор ассемблера, линковщик, текстовый редактор, отладчик в кодах C, C++ и ассемблера, менеджер проектов.

Данная программа работает только на устройствах под управлением операционной системы Windows и является коммерческим продуктом.

2.2.3. Keil MDK

Keil MDK (Microcontroller Development Kit) [12] — инструментарий для разработки программного обеспечения для широкого спектра микроконтроллеров на основе процессоров ARM Cortex-M. MDK включает среду разработки μ Vision, C/C++ компилятор для архитектуры ARM, отладчик и прочие компоненты, необходимые для разработки, которые можно увидеть на рисунке 2.

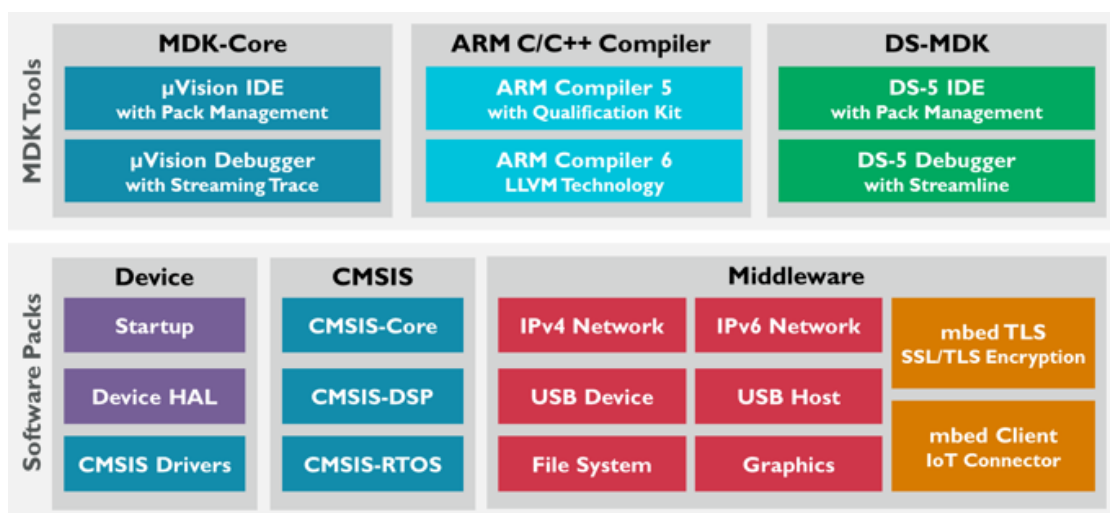


Рис. 2: Компоненты Keil Microcontroller Development Kit. Изображение взято из источника [13].

Важно отметить, что среду разработки Keil μ Vision можно использовать только на устройствах под управлением операционной системы Windows. В дополнение к этому, средство программирования является коммерческим продуктом.

2.2.4. Eclipse ARM plugin

Eclipse [6] — это кроссплатформенная интегрированная среда разработки, которая позволяет разрабатывать приложения на многих языках программирования, таких как Java, C, C++, JavaScript, Python и прочих. Такое многообразие поддерживаемых возможностей достигается благодаря системе подключаемых модулей (плагинов), которые позволяют настроить работу среды в зависимости от своих нужд.

Плагин GNU ARM Eclipse [8] позволяет в среде программирования Eclipse осуществлять разработку и отладку программ для ARM процессоров на языках C, C++. Плагин, как и среда Eclipse, находится в открытом доступе и предоставляет право на свободное распространение и использование. Изображение окна программы Eclipse с примером проекта для микроконтроллера STM32F4 можно увидеть на рисунке 3.

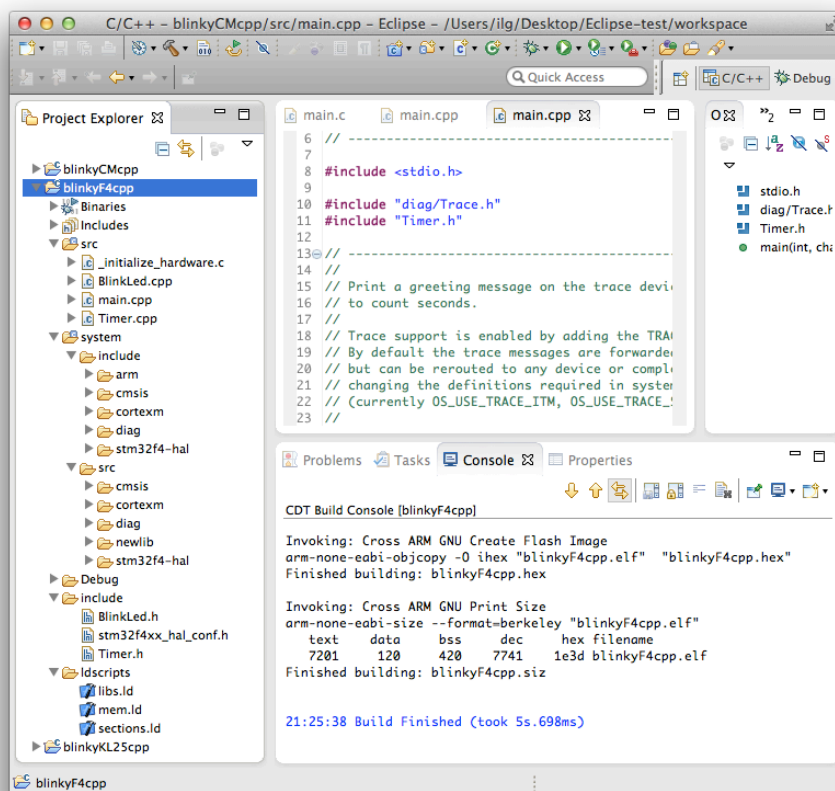


Рис. 3: Окно программы Eclipse с примером проекта для микроконтроллера STM32F4. Изображение взято из источника [7].

2.2.5. Atollic TrueStudio

Atollic TrueStudio [2] представляет собой кроссплатформенную интегрированную среду разработки программного обеспечения для процессоров ARM. Средство разработки создано на основе платформы Eclipse, рассмотренной ранее. Инструментарий включает в себя редактор кода с поддержкой языков C, C++ и ассемблера, C/C++ компилятор, транслятор ассемблера, линковщик, текстовый редактор кода, отладчик, систему сборки исходного кода, менеджер проектов; поддерживает такие системы контроля версий, как Git, SVN, CVS.

Программа Atollic TrueStudio является коммерческой. Имеется бесплатная версия продукта, однако она обладает рядом ограничений, препятствующих полноценной разработке программного обеспечения.

2.2.6. CooCox

CooCox [4] — инструментарий для разработки программного обеспечения для процессоров ARM Cortex-M. В состав средства программирования входят: отладочный адаптер CoLinkEx, интегрированная среда разработки CoIDE, графический инструмент генерации кода CoSmart. Инструментарий CooCox предусматривает свободное распространение и использование.

CooCox CoIDE — среда разработки, основанная на инструменте Eclipse и компиляторе GCC (инструментарий GCC-ARM-Embedded).

Данное средство поддерживает только операционную систему Windows для разработки программного обеспечения.

2.2.7. Выводы

На основе проведенного обзора можно сделать вывод, что все рассмотренные среды программирования обладают рядом преимуществ и недостатков. Результаты сравнительного анализа можно увидеть в таблице на рисунке 4. При этом все вышеизложенные средства предполагают создание программ на текстовом языке программирования, и ни один из приведенных инструментариев не предоставляет возможность создавать программы на визуальном языке. Поэтому целесообразно осуществить поддержку программирования микроконтроллеров STM32 в TRIK Studio.

Технология \ Критерий сравнения	Критерий сравнения		
	Свободное распространение	Кроссплатформенность	Поддержка визуальных языков
IAR Embedded Workbench for ARM	—	—	—
Keil MDK	—	—	—
Eclipse ARM plugin	+	+	—
Atollic TrueStudio	—	+	—
CooCox	+	—	—

Рис. 4: Сравнительная таблица средств программирования микроконтроллера STM32.

2.3. TRIK Studio

2.3.1. Общая архитектура

TRIK Studio [5] — это среда программирования роботов, которая позволяет создавать графические программы для роботов Lego Mindstorms NXT, Lego Mindstorms EV3, ТРИК. Созданные программы можно исполнять прямо на компьютере, посылая команды роботу через Wi-Fi, Bluetooth или USB-интерфейс, можно исполнять на двумерной модели робота, которая симулирует поведение реального робота в процессе исполнения программы, а также можно генерировать по графическим диаграммам код на различных текстовых языках программирования и отправлять его для исполнения на роботе. В состав архитектуры TRIK Studio входят такие модули, как ядро технологии QReal, ядро системы TRIK Studio, наборы моделей роботов для поддерживаемых средой робототехнических конструкторов, интерпретаторы графических программ, генераторы исходного кода программ на текстовых языках программирования по графическим диаграммам, а также средства коммуникации с роботами и прочие компоненты. Общую архитектуру программы TRIK Studio можно увидеть на рисунке 5.

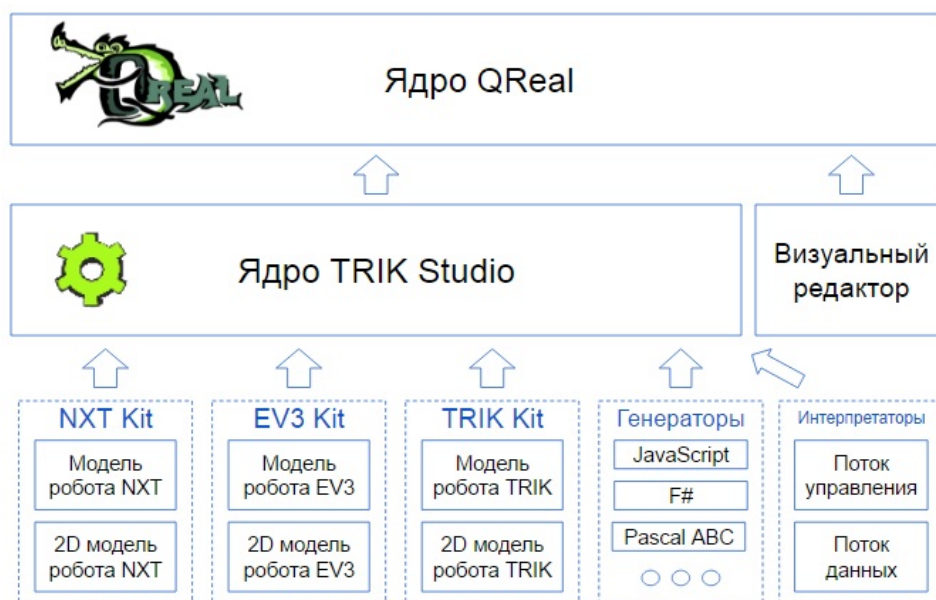


Рис. 5: Общая архитектура TRIK Studio.

2.3.2. Генераторы

Процесс генерации кода на текстовом языке программирования на основе графической программы в TRIK Studio происходит так: сперва по визуальной диаграмме, построенной пользователем, строится граф потока управления программы, который путем обхода проверяется на корректность. В случае синтаксической корректности программы строится семантическое дерево программы. Для каждого поддерживаемого текстового языка программирования задан набор шаблонов генерации, которые позволяют сконструировать код программы в процессе обхода семантического дерева. Архитектура подсистемы генерации кода в TRIK Studio представлена на рисунке 6.

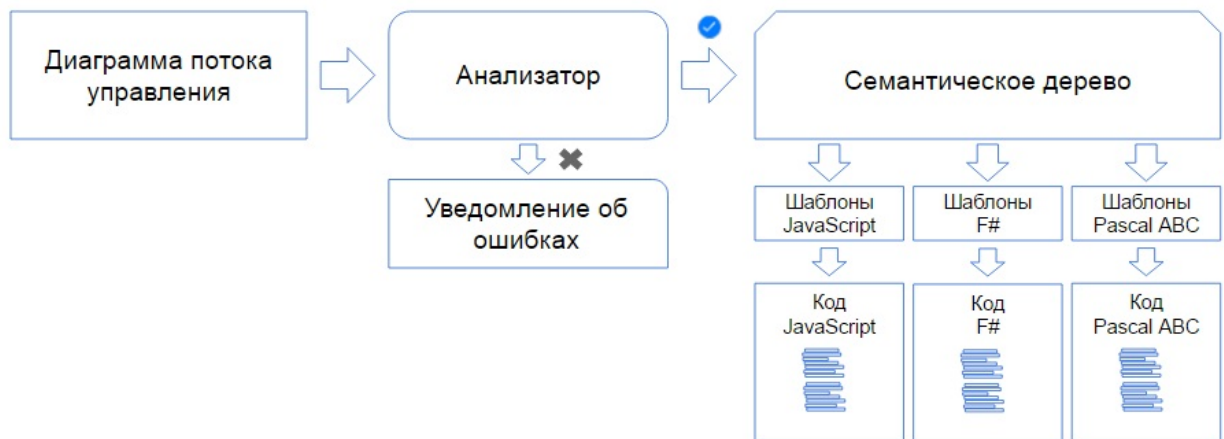


Рис. 6: Архитектура подсистемы генерации кода в TRIK Studio.

Все генераторы кода программы TRIK Studio оформлены в виде подключаемых модулей. Основным классом для всех генераторов является `RobotsGeneratorPluginBase`. От него наследуются все конкретные генераторы для каждой из поддерживаемых платформ роботов и для каждого поддерживаемого программой TRIK Studio языка программирования. Иерархию наследования и пример генераторов кода программы TRIK Studio можно увидеть на рисунке 7.

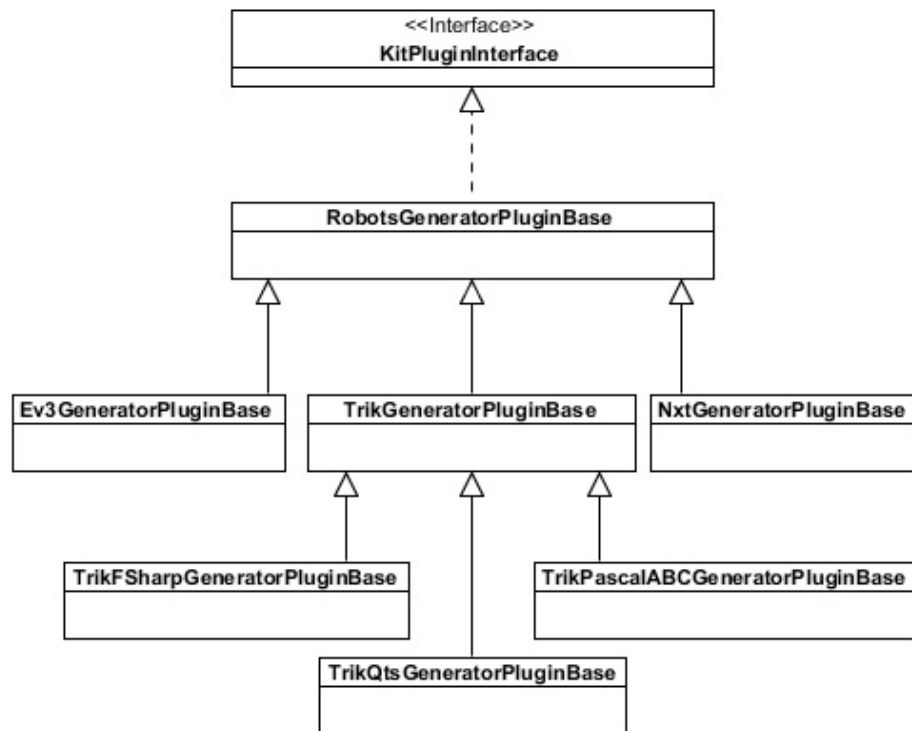


Рис. 7: Генераторы кода в TRIK Studio.

3. Архитектура системы

Разработанная в процессе данной работы система состоит из двух модулей. Первый модуль содержится внутри среды TRIK Studio и представляет собой подсистему получения на основе графической диаграммы, созданной пользователем, кода программы на языке C++. Генерация кода запускается при нажатии пользователем соответствующей кнопки в меню программы TRIK Studio. Схематичное изображение данного модуля можно увидеть на рисунке 8.

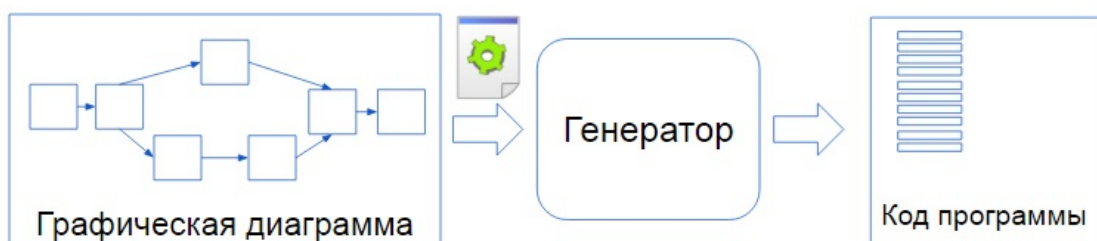


Рис. 8: Схема генерации кода на языке C++ по графической диаграмме.

Для поддержки новой модели микроконтроллеров была расширена архитектура среды программирования TRIK Studio, а именно была добавлена модель робота STM32 и был реализован генератор кода на языке C++. Общую архитектуру программы TRIK Studio с модулем программирования микроконтроллера STM32 можно увидеть на рисунке 9. Зеленым цветом отмечены новые компоненты, которые были добавлены в архитектуру в процессе описываемой работы.

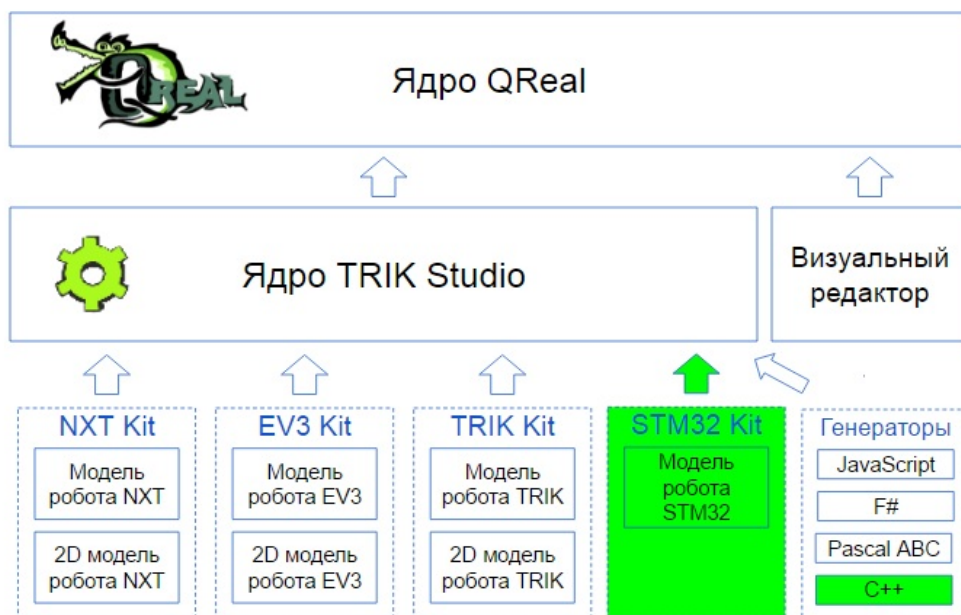


Рис. 9: Архитектура TRIK Studio с модулем программирования микроконтроллера STM32.

Для осуществления генерации кода на языке программирования C++ была расширена архитектура подсистемы генерации кода TRIK Studio, а именно были реализованы шаблоны блоков, которые используются в процессе конструирования кода на текстовом языке программирования. Архитектура подсистемы генерации кода в TRIK Studio с модулем генерации C++ кода представлена на рисунке 10. Зеленым цветом отмечены компоненты, добавленные в процессе выполнения данной работы.

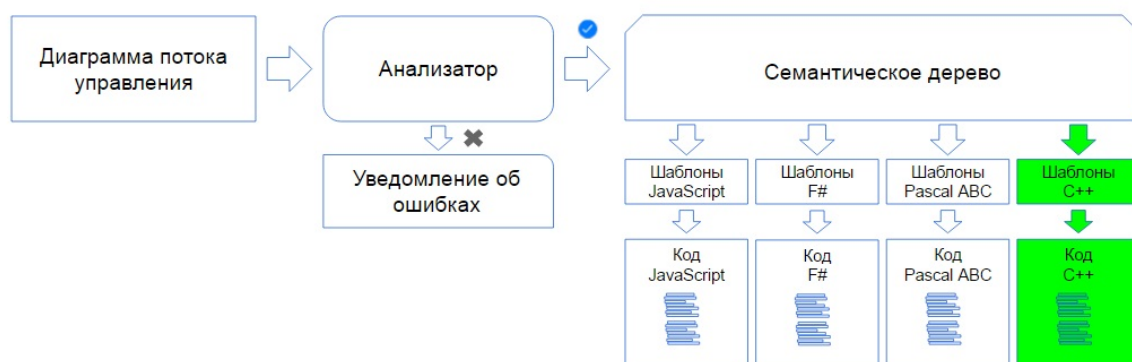


Рис. 10: Архитектура подсистемы генерации кода в TRIK Studio с модулем генерации C++ кода.

Иерархию классов генераторов кода по визуальным диаграммам в TRIK Studio можно увидеть на рисунке 11. Зеленым цветом закрашены блоки, добавленные в архитектуру среды программирования роботов TRIK Studio в процессе работы над системой.

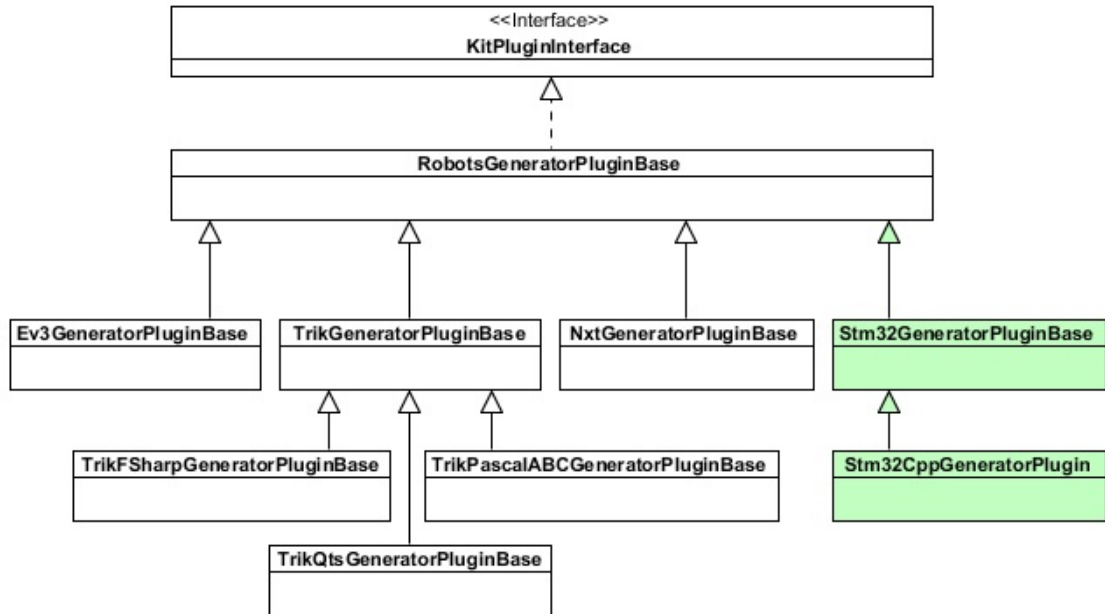


Рис. 11: Архитектура генераторов в TRIK Studio с генератором кода на языке C++.

Вторым модулем является подсистема, основной задачей которой является компиляция исходного кода программы на языке C++ вместе со всеми необходимыми библиотеками в двоичный код и загрузка полученного кода на плату микроконтроллера. Компиляция и загрузка программы начинается после нажатия соответствующей кнопки пользователем. Схему данной подсистемы можно увидеть на рисунке 12.

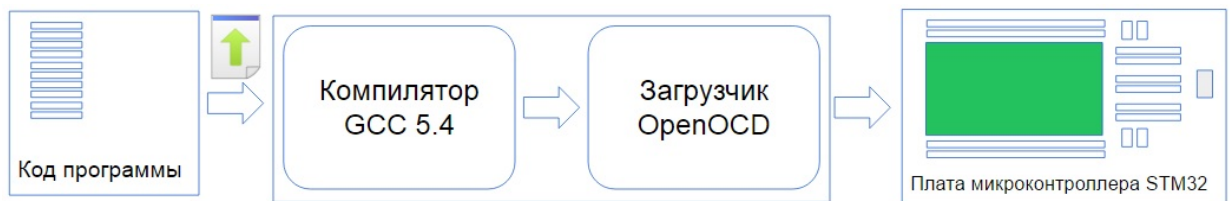


Рис. 12: Схема процесса компиляции и загрузки программы на плату микроконтроллера STM32.

4. Плагин для микроконтроллера STM32 в TRIK Studio

В процессе выполнения работы был создан новый подключаемый модуль, который позволяет создавать графические программы для микроконтроллера STM32 в TRIK Studio. Данный модуль отвечает за создание новой модели робота, именуемой «STM32», и добавление всех необходимых возможностей для работы, а именно внедрение режима генерации программного кода на языке C++.

Рассмотрим иерархию классов плагинной системы TRIK Studio, изображение которой можно увидеть на рисунке 11. Базовым интерфейсом для всех подключаемых модулей, будь то генератор или интерпретатор, является KitPluginInterface. Он объявляет самые общие для всех плагинов компоненты, такие как уникальный идентификатор, метод инициализации, имя для отображения пользователю и т.д. Реализовывает описанный выше интерфейс класс RobotsGeneratorPluginBase, который является базовым для всех генераторов. От него наследуются все подключаемые модули, отвечающие за генерацию кода для каждой из поддерживаемых моделей роботов, а именно Ev3GeneratorPluginBase, NxtGeneratorPluginBase, TrikGeneratorPluginBase и добавленный к ним в рамках данной работы Stm32GeneratorPluginBase. На этом уровне иерархии создается модель робота, с которой в дальнейшем происходят необходимые манипуляции. Для данного класса был реализован потомок Stm32CppGeneratorPlugin, ответственный за генерацию кода на языке C++ и запуск процесса компиляции и загрузки программы на плату микроконтроллера.

5. Генератор кода на языке C++

Для осуществления процесса генерации кода на ранее не поддерживаемой средой TRIK Studio языке программирования нужно, как говорилось ранее, сформировать структуру генерируемого кода путем задания шаблонов генерации. Для этого были объявлены и реализованы шаблоны генерации для таких блоков, как блок чтения значений аналогового сенсора — были реализованы блоки для инфракрасного сенсора, сенсора света, датчика расстояния, датчика касания; блок ожидания в миллисекундах, блок установки цвета фона на экране платы микроконтроллера, блок установки цвета светодиода на плате микроконтроллера (светодиоды LD3 и LD4), блок вывода переданного пользователем текста на экране микроконтроллера. Данные шаблоны генерируют вызовы соответствующих функций, которые объявлены в выделенном модуле Brick, который подключается к полученной в процессе генерации программе. Цель данного модуля заключается в предоставлении интерфейса между пользовательской программой на языке C++ и драйверами микроконтроллера, входящими в состав библиотеки STM32Cube. Данный модуль интегрирован в программу TRIK Studio вместе с необходимыми библиотеками для работы с микроконтроллером.

Полученный в результате генерации код программы отображается пользователю в отдельном окне, в котором он может вносить изменения в программный код перед отправкой программы на исполнение на плате микроконтроллера.

В качестве целевого языка программирования выбрана версия стандарта языка C++11, одной из основных возможностей которого является автоматический вывод типов, который удобно использован в процессе генерации. Ключевое слово «auto» позволяет уменьшить избыточность и громоздкость программного кода, однако ничто не мешает пользователю писать программу в привычном стиле в окне редактирования кода и исполнять созданный код на плате микроконтроллера.

6. Интеграция компилятора и загрузчика с TRIK Studio

Существует множество различных руководств и методических пособий по программированию микроконтроллеров STM32 [3], [18], [23]. Основные идеи, касающиеся компиляции и загрузки программы, были взяты из руководства по разработке программ для STM32 разработчика Энди Брауна.

Для запуска программы на языке C++ на исполнение необходимо сначала получить бинарный код программы, а потом полученный двоичный код отправить на исполнение на плату микроконтроллера. Для этого разработана подсистема интеграции компилятора и загрузчика программы с TRIK Studio.

Для осуществления работы подсистемы используется инструментальный GNU Tools ARM Embedded [9], в который входит компилятор GCC 5.4 (GNU Compiler Collection) [10] для архитектуры ARM, драйвер встроенного в плату микроконтроллера программатора ST-LINK/V2 [16], который работает по интерфейсу JTAG, и библиотека OpenOCD (Open On-Chip Debugger) [14] для загрузки прошивки на плату микроконтроллера.

JTAG (Joint Test Action Group) — это общепризнанный промышленный аппаратный интерфейс, который используется во многих микросхемах для прошивки, отладки и тестирования. Название произошло от группы разработчиков стандарта IEEE 1149.1-1990, в котором были описаны результаты работы над созданием данного интерфейса. Стандарт предусматривает выделение в микросхеме специального набора блоков, состояние работоспособности которых можно оценить независимо от микросхемы с помощью отдельных выводов. При этом есть возможность не только проверять работу блоков, но и осуществлять управление ими. Обычно используется четыре или пять выделенных сигнальных линий.

При запуске сформированной программы на языке C++ на исполнение происходит вызов спроектированного в рамках данной работы пакетного файла. При этом в зависимости от операционной системы, на которой запущена среда программирования TRIK Studio, будет запущен соответствующий своей операционной системе пакетный файл. Поддерживаются операционные системы Windows, Linux, MacOS X. Каждый пакетный файл содержит набор команд для подготовки окружения для запуска процесса компиляции, а именно запись в переменную окружения путей к инструментам, необходимым для осуществления компиляции и загрузки программы. Пользователю достаточно лишь указать в используемом пакетном файле пути до установленных им библиотек. Также пакетный файл содержит команду для запуска на исполнение файла Makefile.

За основу Makefile для подсистемы интеграции компилятора и загрузчика с TRIK Studio был взят файл, созданный Стефаном Вогелем — разработчиком из Германии, который разместил руководство по компиляции и прошивке платы STM32 для свободного использования [17]. При содействии студента кафедры системного программирования Санкт-Петербургского государственного университета Романа Белкова данный файл был оптимизирован для работы с платой микроконтроллера STM32F429I-Discovery. Также Роман выступил в качестве автора прототипа средства программирования микроконтроллера.

Makefile производит запуск компиляции на устройстве пользователя и загрузку программы на исполнение программатору платы ST-LINK/V2. Makefile обеспечивает корректную линковку со всеми необходимыми библиотеками и драйверами микроконтроллера STM32, которые интегрированы в среду программирования TRIK Studio.

7. Апробация

Была проведена апробация разработанного средства программирования микроконтроллера STM32 на плате микроконтроллера STM32F429I-Discovery.

На рисунке 13 представлен пользовательский интерфейс среды программирования роботов TRIK Studio.

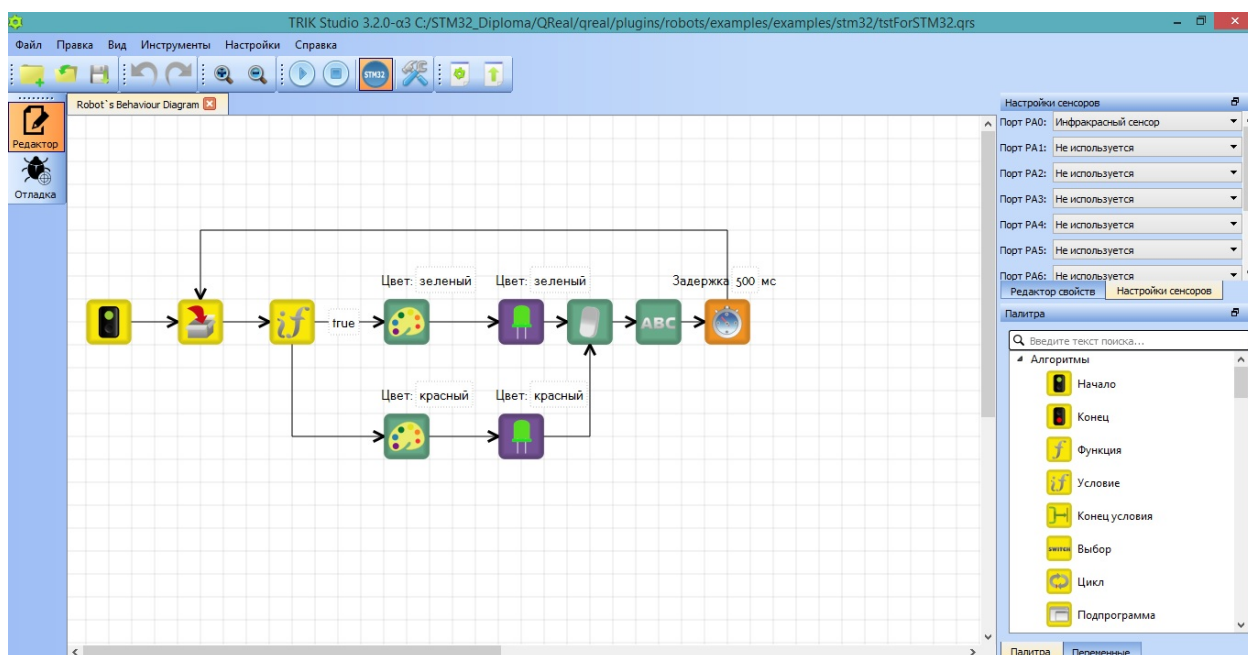


Рис. 13: Пользовательский интерфейс среды TRIK Studio.

Апробация проводилась на примере графической программы, представленной на рисунке 14.

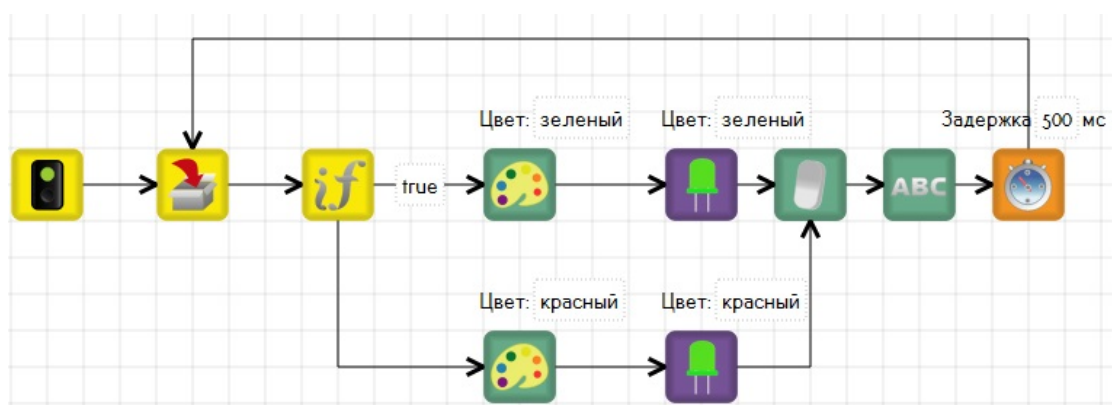
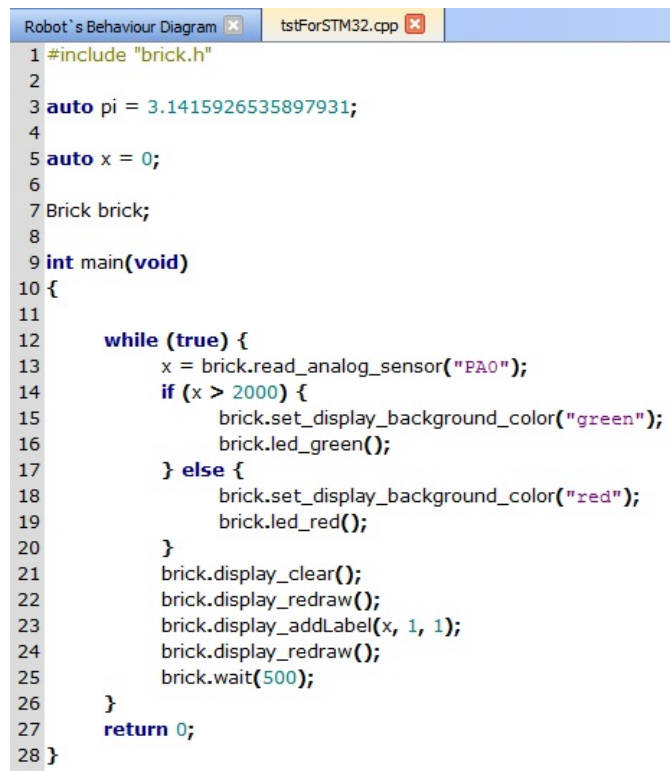


Рис. 14: Тестовая программа на визуальном языке.

Данная программа в бесконечном цикле каждые полсекунды считывает значение инфракрасного сенсора, подключенного к порту PA0, и в зависимости от значения либо закрашивает фон дисплея платы микроконтроллера зеленым цветом и выставляет значение светодиода в зеленый цвет (светодиод LD3), либо закрашивает экран красным цветом и включает красный цвет светодиода платы (светодиод LD4). Затем экран очищается, и на дисплее выводится значение, которое показывает датчик. Таким образом происходит определение присутствия объекта перед датчиком.

Для проведения апробации к плате микроконтроллера STM32F429I-Discovery был подключен инфракрасный аналоговый датчик, значения которого и измерялись в процессе проведенного эксперимента. Микроконтроллер последовательно подключался к устройствам под управлением операционных систем Windows 8.1, Ubuntu 16.04, MacOS X. На каждом устройстве были установлены все необходимые для проведения апробации инструментарию, а именно были установлены библиотеки GNU Tools ARM Embedded и OpenOCD, а также были прописаны пути до установленных библиотек в соответствующий операционной системе пакетный файл. Для устройства под управлением операционной системы Windows также были установлены драйвер для программатора ST-LINK и инструментарий Cygwin для исполнения команд среды UNIX в системе Windows. На каждом из устройств была запущена программа TRIK Studio с подключенным плагином для программирования микроконтроллера STM32. В среде программирования роботов был открыт проект с графической диаграммой программы, представленной и описанной выше. По данной диаграмме сгенерировался код программы на языке C++, который можно увидеть на рисунке 15. Код открывается в отдельном окне, в котором пользователь может вносить изменения, что было проверено во время апробации.



```

1 #include "brick.h"
2
3 auto pi = 3.1415926535897931;
4
5 auto x = 0;
6
7 Brick brick;
8
9 int main(void)
10 {
11
12     while (true) {
13         x = brick.read_analog_sensor("PA0");
14         if (x > 2000) {
15             brick.set_display_background_color("green");
16             brick.led_green();
17         } else {
18             brick.set_display_background_color("red");
19             brick.led_red();
20         }
21         brick.display_clear();
22         brick.display_redraw();
23         brick.display_addLabel(x, 1, 1);
24         brick.display_redraw();
25         brick.wait(500);
26     }
27     return 0;
28 }

```

Рис. 15: Тестовая программа на языке C++.

Затем программа была запущена на исполнение. Через некоторое время (порядка 15 секунд) на микроконтроллере замигала лампочка светодиода LD1, что свидетельствовало о приеме двоичных данных, т.к. данный светодиод является индикатором передачи данных через интерфейс USB. В окне программы TRIK Studio «Ошибки» появилась информация о предупреждениях, а также отладочная информация, которая собралась за время компиляции и загрузки программы. Состояние отладочной платы микроконтроллера можно увидеть на рисунке 16.

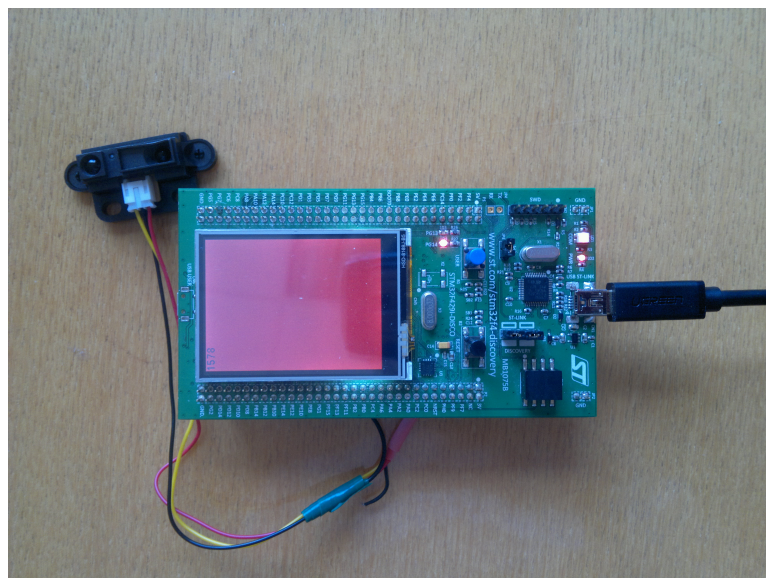


Рис. 16: Состояние микроконтроллера при отсутствии объекта перед инфракрасным датчиком.

В случае приближения некоторого объекта на близкое расстояние к инфракрасному датчику происходит переход отладочной платы в другое состояние, которое можно увидеть на рисунке 17. Дисплей окрашивается в зеленый цвет, и загорается зеленый светодиод платы.

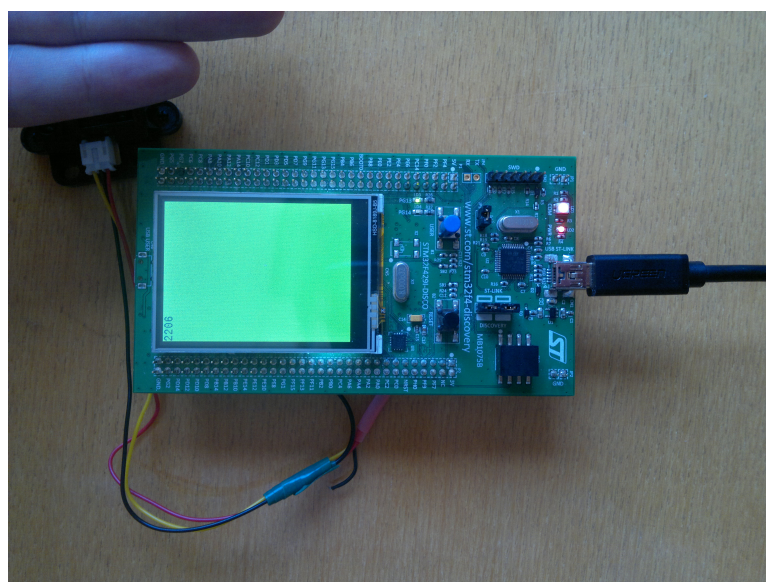


Рис. 17: Распознавание объекта перед инфракрасным датчиком.

Если объект удалить от датчика, то плата вернется в свое первоначальное состояние. Это можно увидеть на рисунке 18.

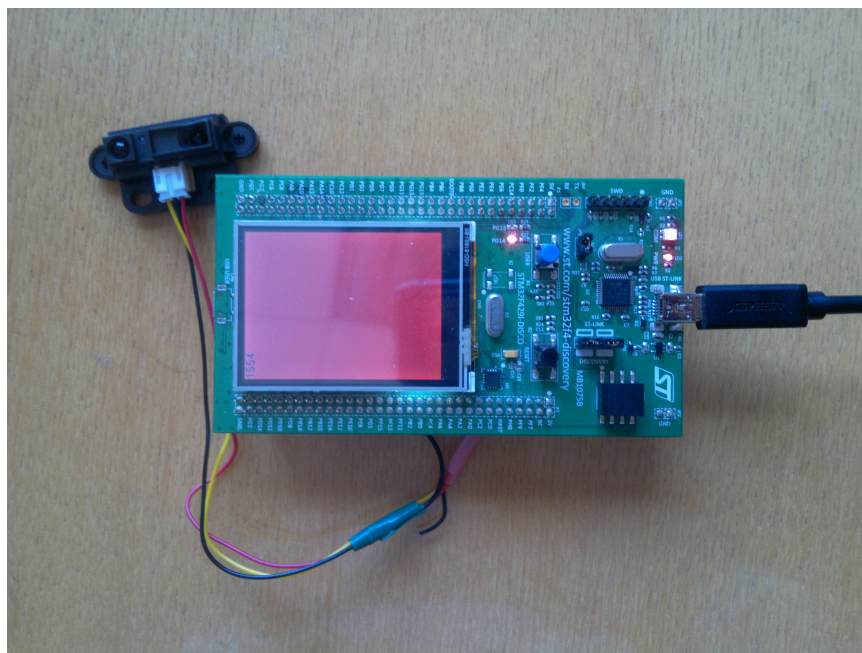


Рис. 18: Состояние микроконтроллера при удалении объекта от инфракрасного датчика.

При отключении платы микроконтроллера от компьютера питание выключается. При этом при последующем включении микроконтроллера будет исполняться та программа, которая была загружена на плату. Каждая последующая загрузка прошивки перезаписывает рабочую программу микроконтроллера.

Заключение

В результате работы были получены следующие результаты:

- разработана архитектура системы программирования микроконтроллера STM32 в среде TRIK Studio;
- реализован подключаемый модуль для микроконтроллера STM32 в TRIK Studio;
- реализован генератор кода из программы на языке визуальных диаграмм в программу на языке C++;
- разработана подсистема интеграции компилятора и загрузчика программы с TRIK Studio;
- проведена апробация средства программирования на плате микроконтроллера STM32F4-Discovery;
- по теме выпускной работы был сделан доклад на конференции студентов, аспирантов и молодых ученых «Современные технологии в теории и практике программирования» в Политехническом университете и были опубликованы материалы в сборнике конференции [26].

Результат работы доступен по ссылке [19].

Список литературы

- [1] ARM. — URL: <http://www.arm.com/> (дата обращения: 14.05.2017).
- [2] Atollic. TrueSTUDIO. — URL: <https://atollic.com/truestudio/> (дата обращения: 14.05.2017).
- [3] C++ STM32 Development Environment. — URL: <http://andybrown.me.uk/2015/03/22/stm32dev-windows/> (дата обращения: 14.05.2017).
- [4] Coocox. — URL: <http://www.coocox.org/index.html> (дата обращения: 14.05.2017).
- [5] Dmitry Mordvinov, Yurii Litvinov, Timofey Bryksin. TRIK Studio: Technical Introduction. — Proceedings of the FRUCT'20, 2017, ISSN 2305-7254, ISBN 978-952-68653-0-0. pp 296-308.
- [6] Eclipse. — URL: <http://www.eclipse.org/> (дата обращения: 14.05.2017).
- [7] Eclipse GNU ARM. STM32Fxx templates. — URL: <http://gnuarmecclipse.github.io/templates/stm32f/> (дата обращения: 14.05.2017).
- [8] GNU ARM Eclipse. — URL: <http://gnuarmecclipse.github.io/> (дата обращения: 14.05.2017).
- [9] GNU ARM Embedded Toolchain. — URL: <https://developer.arm.com/open-source/gnu-toolchain/gnu-rm> (дата обращения: 14.05.2017).
- [10] GNU Compiler Collection. — URL: <https://gcc.gnu.org/> (дата обращения: 14.05.2017).
- [11] IAR Embedded Workbench. — URL: <https://www.iar.com/iar-embedded-workbench> (дата обращения: 14.05.2017).

- [12] Keil Embedded Development Tools. — URL: <http://www.keil.com/> (дата обращения: 14.05.2017).
- [13] Keil Microcontroller Development Kit. — URL: <http://www2.keil.com/mdk5/> (дата обращения: 14.05.2017).
- [14] OpenOCD. — URL: <http://openocd.org/> (дата обращения: 14.05.2017).
- [15] QReal. — URL: <https://github.com/qreal/qreal> (дата обращения: 14.05.2017).
- [16] ST-LINK USB driver for Windows. — URL: http://www.st.com/content/st_com/en/products/embedded-software/development-tool-software/stsw-link009.html (дата обращения: 14.05.2017).
- [17] STM32 Cube-GCC. — URL: <https://github.com/stv0g/stm32cube-gcc> (дата обращения: 14.05.2017).
- [18] STM32 Firmware and Debugging Manual. — URL: <https://github.com/mrsin/k-lab/wiki/STM32-прошивка-с-помощью-OpenOCD-и-отладка-в-Eclipse> (дата обращения: 14.05.2017).
- [19] STM32 plugin for TRIK Studio. — URL: <https://github.com/PrihodkoStanislav/qreal/tree/STM32> (дата обращения: 14.05.2017).
- [20] STM32F429I-Discovery. — URL: <http://www.st.com/en/evaluation-tools/32f429idiscovery.html> (дата обращения: 14.05.2017).
- [21] STMicroelectronics. — URL: http://www.st.com/content/st_com/en.html (дата обращения: 14.05.2017).

- [22] Брыксин Т.А., Литвинов Ю.В. Среда визуального программирования роботов QReal:Robots. — Материалы международной конференции «Информационные технологии в образовании и науке». Самара. 2011. С. 332-334.
- [23] Бугаев В.И., Мусиенко М.П., Крайнык Я.М. Лабораторный практикум для изучения микроконтроллеров архитектуры ARM Cortex-M4 на базе отладочного модуля STM32F4 Discovery. — Москва-Николаев: МФТИ-ЧГУ, 2013.
- [24] Кознов Д.В. Основы визуального моделирования. — БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий, 2008.
- [25] Литвинов Ю.В., Кириленко Я.А. TRIK Studio: среда обучения программированию с применением роботов. — V Всероссийская конференция «Современное технологическое обучение: от компьютера к роботу» (сборник тезисов), СПб., ЗАО «Полиграфическое предприятие № 3», 2015, С. 5-7.
- [26] Приходько С.В., Литвинов Ю.В. Поддержка программирования микроконтроллера STM32 в TRIK Studio. — Материалы научно-практической конференции студентов, аспирантов и молодых ученых «Современные технологии в теории и практике программирования». СПб.: Изд-во Политехн. ун-та, 2017. С. 71-72.
- [27] Терехов А.Н., Брыксин Т.А., Литвинов Ю.В. и др. Архитектура среды визуального моделирования QReal. — Системное программирование. Вып. 4. СПб.: Изд-во СПбГУ, 2009.