

STM32VLDISCOVERY – средство для быстрой разработки опытных образцов цифровых систем управления

А.В. Ескин, В.А. Жмудь, В.Г. Трубин
ФГБОУ ВПО НГТУ, Новосибирск, Россия

Аннотация: Рассматриваются основные вопросы возникающие при знакомстве с микроконтроллерами STM32 на примере отладочной платы STM32VLDISCOVERY. Демонстрируется создание простейшего проекта программы мигающей светодиодом для бесплатной среды разработки (IDE) Eclipse⁴

Ключевые слова: Микроконтроллер, STM32, STM32DISCOVERY, Eclipse, ST-LINK, STM32 ST-LINK Utility.

ВВЕДЕНИЕ

Наряду с легендарным 8 битным открытым ядром MCS51, прародителем для которого стал микроконтроллер Intel 8051 (1980 год) и получившим широкое распространение, в сегменте 32 битных микроконтроллеров, подобной стала архитектура ARM. Начиная с момента появления первого работающего экземпляра (1985 год) и по сей день она продолжает набирать популярность и завоёвывать рынок встраиваемых устройств. Хотя родовой вотчиной для неё является рынок процессоров для мобильных устройств, где эта архитектура занимает лидирующие позиции (около 95 % рынка [1]), её высокая производительность и малые размеры на кристалле позволила создать микроконтроллеры на её основе.

На настоящий момент целый ряд известных производителей микроконтроллеров имеют в своём ассортименте целую линейку устройств с ядром этой марки. Совместимость системы команд между различными производителями предоставляет некоторые преимущества перед проприетарными решениями: независимость от конкретного производителя и его финансово — технологического состояния, а также свободная переносимость кода (не использующего периферийные модули). В дополнение к этому, высокая конкуренция предоставляет пользователю модели оптимальные по цене и качеству.

МИКРОКОНТРОЛЛЕРЫ STM32

Ситуация на рынке постоянно меняется, и на данный момент очень привлекательным выглядят предложения ARM микроконтроллеров от фирмы ST Microelectronics [2]. Компания предоставляет микроконтроллеры трёх версий ядер: ARM Cortex M0, ARM Cortex M3 и ARM Cortex M4 [3]. Каждое ядро представлено линейкой устройств, различающихся по типу корпуса и объёму памяти. Цены предлагаемые этим производителем позволяют конкурировать даже с 8 битными микроконтроллерами других поставщиков [4], при этом с характеристиками в которых, до этого момента, были сильны только 8 битные изделия.

Все микроконтроллеры с ядром ARM представлены под торговой маркой STM32 и делятся на шесть семейств: STM32F0 (Entry-Level MCUs), STM32L1 (Ultra-Low-Power MCUs), STM32F1 (Mainstream MCUs), STM32F2 (High - performance MCUs), STM32F3 (Mixed Signal MCUs) и STM32F4 (MCUs with DSP and FPU). STM32F0 оснащено ядром Cortex M0, STM32L1, STM32F1 и STM32F2 ядром Cortex M3, а STM32F3 и STM32F4 ядром Cortex M4. ST Microelectronics для продвижения на рынке своих устройств для каждого семейства выпускает отладочную плату Discovery для быстрого знакомства с ними и освоения.

Отличительной особенностью ядра ARM является встроенный модуль отладки, который в микроконтроллерах STM32 представлен двумя аппаратными интерфейсами: JTAG и SW (Single Wire). Для реализации связи с ядром по каналу SW компанией был разработан протокол ST-LINK. Платы Discovery содержат встроенный отладчик / программатор, работающий по протоколу ST-LINK и позволяющий программировать и отлаживать, как внутренний (смонтированный на плате) микроконтроллер, так и внешний, подключаемый при помощи расположенной на плате штыревой линейки. Подключение отладчика ST-LINK к персональному компьютеру осуществляется через USB интерфейс, также через него происходит питание всей платы.

По периметру плат Discovery расположены штыревые линейки с шагом 2,54 мм, каждый вывод которой соединён с одноименным выводом микроконтроллера, что позволяет вставлять эту

⁴ Работа выполнена по заданию Министерства образования и науки РФ, проект №7.599.2011, Темплан, НИР № 01201255056.

плату в свои устройства.

ST Microelectronics не разочаровала пользователей и в плане цены данных устройств. За небольшие деньги можно приобрести готовую плату с отладчиком в комплекте и с возможностью программировать и отлаживать другие устройства данной фирмы с подобным ядром.

На примере отладочной платы STM32VLDISCOVERY обсудим основные моменты касающиеся программирования микроконтроллера STM32F100RB из семейства STM32F1 с ядром ARM Cortex M3.

Внешний вид отладочной платы представлен на рисунке 1 [5].



Рис. 1 - Внешний вид отладочной платы STM32VLDISCOVERY

Данная плата помимо общих для Discovery элементов содержит: два кварцевых резонатора: на 8 МГц (является заменяемым) и часовой (32768 Гц), две кнопки: сброса и пользовательскую (помеченную как USER), два светодиода (LD3 и LD4) и перемычку JP1 для измерения тока потребления по цифровой цепи питания микроконтроллера. Питание платы может осуществляться как через разъем USB mini, так и через линии 5 вольт и 3,3 вольта выведенные на выводы штыревой линейки (5 вольт промаркирован «5V», 3,3 вольта - «3V3»).

Для программирования встроенного

STM32F100RB необходимо подсоединить плату к USB порту персонального компьютера и замкнуть двумя перемычками соседние контакты разъема CN3. На персональном компьютере нужно установить драйвер для ST-LINK, который можно загрузить с официального сайта производителя [6].

Теперь стоит сказать несколько слов о среде программирования, в которой будут писаться программы для микроконтроллера, и при помощи которой будет вестись её отладка.

СРЕДА РАЗРЕБОТКИ ECLIPSE

В настоящий момент для написания программ для открытой архитектуры ARM существует достаточное количество коммерческих сред: Keil for ARM, ARM Development Studio 5, (разработка компании ARM Ltd), Embedded Workbench for ARM (разработка компании IAR Systems), TrueSTUDIO for STM32 (разработка компании Atollic) и т. п. Все они достаточно удобные и качественные продукты, но имеют один недостаток — это то, что они платные.

Не так давно появился очень мощный и при том бесплатный продукт позволяющий заменить собой коммерческие среды разработок — это интегрированная среда разработки Eclipse.

По сути эта среда представляет собой конструктор построенный из отдельных частей (расширений среды Eclipse). При этом имеется возможность добавлять и удалять отдельные части (plugin) этой системы пользователем во время работы, настраивая систему под конкретную задачу. К счастью, эта возможность встроена в саму среду и затраты времени пользователя на это минимальные. Такое построение открывает большие возможности по изменению функциональности среды.

Eclipse относится к свободно распространяемому программному обеспечению и поддерживается сообществом Eclipse Foundation. С 2006 года фонд Eclipse координирует ежегодный общий релиз (Simultaneous Release), который происходит в июне [7]. На данный момент уже имеется 10 таких выпусков. С 30 июня 2006 года выпускам стали давать имена: «Callisto», «Europa», «Ganymede» и другие. Выпуск представляет собой набор последних продуктов, для различных задач и языков программирования. В свою очередь, внутри выпуска есть сборки, которые нацелены на решение конкретных задач. Так, например, для разработчиков на языке C/C++ существует набор (пакет — package) Eclipse IDE for C/C++ Developers, который присутствует во множестве выпусков, он также известен как Eclipse CDT (Eclipse C/C++ Development Tools). Практически готовые к использованию пакеты можно найти на официальном сайте [8].

Из-за того, что среда разделена на множество плагинов, каждый из которых является отдельной

программой, над которой работает отдельный коллектив программистов, улучшение среды в целом происходит достаточно быстро. Поэтому среда содержит в себе все самые новые функции по работе с кодом, которые, без сомнения, ускоряют создание программы.

Eclipse написана на языке *Java* и поэтому является кросс платформенным программным обеспечением. Из-за дополнительной языковой прослойки скорость её работы не очень высока на компьютерах средней мощности. Но тот факт, что она бесплатна и обладает неисчерпаемыми возможностями, практически полностью перевешивает эту медлительность.

Eclipse, по сути, это графическая оболочка, предоставляющая удобный интерфейс по работе с компилятором, программатором и отладчиком, но при этом не содержащая таковых. *Eclipse* самостоятельно не сможет откомпилировать программу, для этого в операционной среде должен быть установлен один из поддерживаемых наборов компиляции и отладки кода. К таким, например, для микроконтроллеров с ядром *ARM* относятся: *Sourcery CodeBench Lite Edition*, *GNUARM*, *WinARM* и *Yagarto*. Для интеграции с этими компиляторами в *Eclipse* используется специальный плагин - *GNU ARM Eclipse*, который добавляется штатным механизмом среды, о чём будет рассказано чуть позже.

Кроме этого *Eclipse* требует наличие установленной виртуальной машины *Java* (*Java Runtime Environment*). Наличие перечисленного ПО позволяет собрать минимальную *IDE* (*Integrated Development Environment*) способную компилировать программы для микроконтроллеров с ядром *ARM*.

СБОРКА ПОЛНОЦЕННОЙ СРЕДЫ РАЗРАБОТКИ

Процесс сборки можно начать с установки пакета компиляции. Выберем, для примера, *Sourcery CodeBench Lite Edition*. Его можно скачать с официального сайта предварительно пройдя регистрацию по форме [9]. При загрузке следуют верно указать Вашу операционную систему, а затем выбрать дистрибутив напротив поля *EABI* графы *Target OS*. Процесс установки достаточно прост, если следовать указаниям мастера.

Перед тем как установить *Eclipse* в Вашей операционной системе должна быть установлена виртуальная машины *Java*, если она ещё не установлена, то установите её с официального сайта [10]. Установка достаточно проста, стоит лишь следовать инструкциям мастера.

Настала очередь *Eclipse*. Её не нужно устанавливать, она просто распаковывается из *zip* архива в Вашу любимую папку. Стоит остановиться на выборе подходящей версии релиза. Здесь нужно сперва зайти на официальную страничку плагина *GNU ARM*

Eclipse [11] и узнать поддерживаемые им релизы *Eclipse*, а затем уже следуя этим указаниям загрузить *zip* архив подходящего релиза. Так, например, на момент написания этой статьи последним поддерживаемым *GNU ARM Eclipse* релизом является *Eclipse 3.7 Indigo*, но в дальнейшем описании будем использовать *Eclipse 3.6 Helios*.

После установки *IDE Eclipse* следует добавить к ней *GNU ARM Eclipse* воспользовавшись встроенным механизмом. Остановимся на этом поподробнее.

Для начала необходимо запустить из распакованного *zip* архива *Eclipse* файл с именем *eclipse.exe*, после чего начнётся процесс загрузки среды (рекомендуется создать ярлык на рабочем столе этого файла, для ускорения доступа). В ходе загрузки, при первом старте (и при последующих), Вас спросят о местонахождении рабочего пространства (*Workspace*) откуда будут загружаться и где будут храниться файлы созданных проектов. Лучше сразу задать удобное для Вас место, чтобы потом не искать среди множества вложенных папок необходимый файл. Также необходимо избегать нелатинских символов и пробелов в строке пути рабочего пространства. После загрузки среды с пустой папкой *Workspace* появится стандартное приветственное окно среды представленное на рисунке 2.



Рис. 2 - Приветственное окно среды Eclipse Helios

На данном этапе нас будет интересовать только подпункт элемента главного меню «*Help*» - «*Install New Software...*», который показан на рисунке 3.

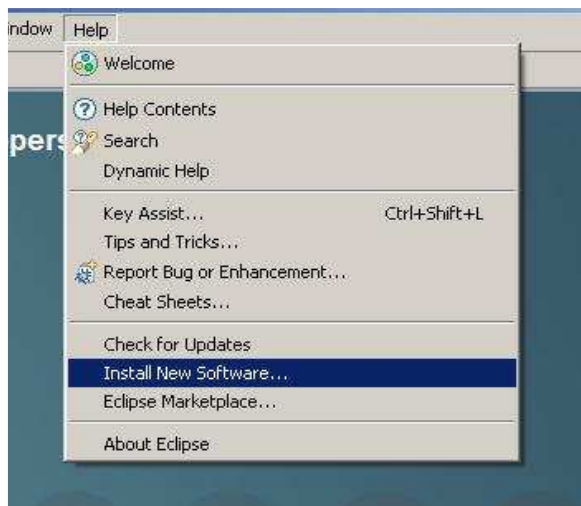


Рис. 3 - Место начала установки всех расширений Eclipse.

При выборе этого пункта мы переходим к следующему окну, которое позволяет загрузить нужное расширение среды — плагин.

Eclipse позволяет устанавливать расширения как напрямую с сайта разработчиков (при наличии интернет соединения), так и из zip архива хранящегося на жестком диске. В последнем случае, при наличии неисправностей в интернет соединении, рекомендуется его отключить, иначе процесс установки может остановиться посередине хода установки.

Итак, после выбора пункта «*Install New Software...*» должно появиться окно «*Install*», представленное на *рисунке 4*.

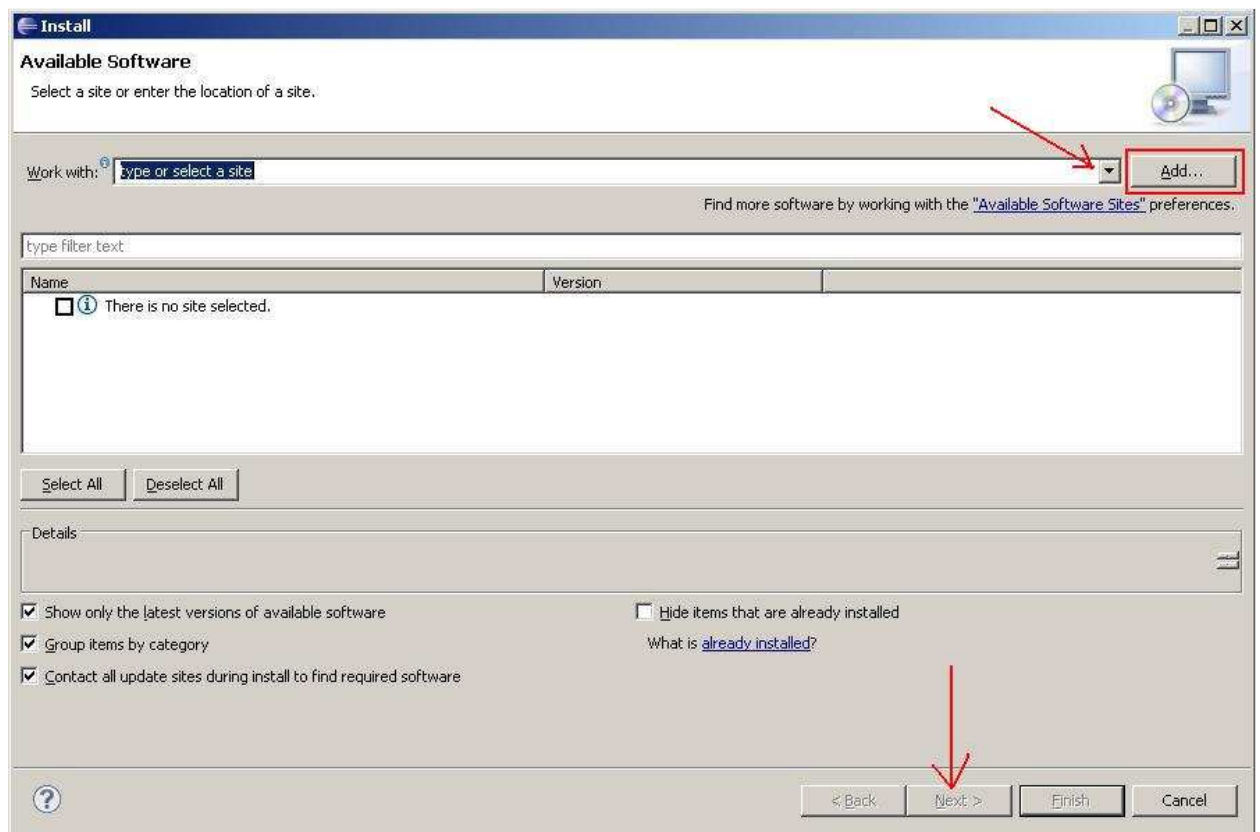


Рис. 4 - Выбор местонахождения устанавливаемого расширения Eclipse.

Здесь следует воспользоваться кнопкой «*Add...*», чтобы добавить необходимое расположение плагина.

Для *GNU ARM Eclipse* в поле *Location* следующего окна «*Add Repository*» следует ввести интернет адрес «<http://gnuarmeclipse.sourceforge.net/updates>» или воспользоваться кнопкой «*Archive...*», чтобы указать местонахождение zip архива на локальном диске. Нажатие кнопки «*OK*» подтверждает Вашу уверенность в введенных данных.

Возвратившись обратно к окну «*Install*» следует поставить галочку напротив имени *CDT GNU Cross Development Tools* — это приведет к разблокированию кнопки «*Next >*».

Дальнейший процесс установки плагина не

должен вызвать затруднений. Схожим образом устанавливаются другие расширения. Здесь нужно отметить, что, практически всегда, тот или иной плагин в интернете обладает своим сайтом, на страничках которого можно найти всю информацию о способах его установки. Так, например, процесс установки *GNU ARM Eclipse* описан по адресу [12].

После этого среда узнает о существовании в системе *Sourcery CodeBench Lite Edition* и автоматически производит все необходимые действия по интеграции с ним.

Теперь *Eclipse* может компилировать текст

программы, написанный на языке C/C++, в машинные коды ядра ARM Cortex M3.

СОЗДАНИЕ НОВОГО ПРОЕКТА

Следующим шагом на пути освоения отладочной платы STM32VLDISCOVERY является создание проекта, откомпилированный код, которого будет загружен во FLASH память микроконтроллера STM32F100RB.

STM32VLDISCOVERY продаются уже с запрограммированной в память программой, текст которой можно скачать с официального сайта фирмы ST Microelectronics [13]. Трудность заключается в том, что написанный проект не был рассчитан на среду Eclipse, поэтому придется произвести некоторые изменения в структуре проекта для адаптации его под данную среду.

Прежде всего, необходимо создать новый пустой проект. Для этого перейдем из приветственного окна в пустое рабочее пространство нажав кнопку «Go to the Workbench» как показано на рисунке 5.

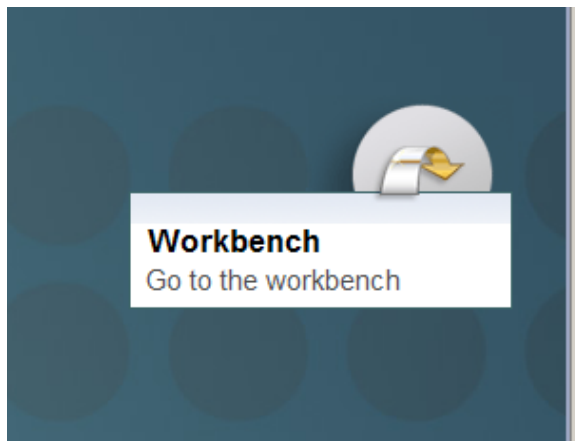


Рис. 5 - Переход в рабочее пространство проектов Eclipse

Затем создадим новый проект на языке C, выполнив команду *File* → *New* → *C Project*. В появившемся окне «C Project» в поле «Project Name:» следует указать имя нового проекта, пусть оно будет — DISCOVER. В поле «Project type:» следует выбрать *ARM Cross Target Application* → *Empty Project*, а в поле «Toolchains:» – «ARM Windows GCC (Sourcery G++ Lite)». Если все сделано, то нажимаем кнопку «Finish».

Далее нужно скопировать некоторые файлы из предлагаемого ST Microelectronics zip архива проекта в созданную Eclipse директорию нового проекта. В следующем тексте будут использоваться следующие сокращения <zip папка> - путь к файлам распакованного zip архива, <workspace> - путь к созданному рабочему пространству Eclipse. Итак, нужно скопировать в папку <workspace>\DISCOVER\ следующие:

1. файл STM32F100RB_FLASH.ld из <zip папка>\an3268\stm32vldiscovery_package\

Project\Demo\TrueSTUDIO\DISCOVER;

2. две папки (inc и src) вместе с вложенными в неё файлами из <zip папка>\an3268\stm32vldiscovery_package\Project\Demo;
3. папку Utilities из <zip папка>\an3268\stm32vldiscovery_package

Затем нужно создать в <workspace>\DISCOVER\ две папки с именем CMSIS и STM32F10x_StdPeriph_Driver. Папку CMSIS следом заполнить файлами:

1. «core_cm3.c» и «core_cm3.h» (из <zip папка>\an3268\stm32vldiscovery_package\Libraries\CMSIS\CMS3\CoreSupport);
2. stm32f10x.h, system_stm32f10x.c, system_stm32f10x.h (из <zip папка>\an3268\stm32vldiscovery_package\Libraries\CMSIS\CMS3\DeviceSupport\ST\STM32F10x);
3. startup_stm32f10x_md_vl.s (из <zip папка>\an3268\stm32vldiscovery_package\Libraries\CMSIS\CMS3\DeviceSupport\ST\STM32F10x\startup\TrueSTUDIO)

При этом у файла startup_stm32f10x_md_vl.s следует поменять расширение с s на S. Это необходимо для того чтобы среда Eclipse распознала этот файл как файл с кодом на языке ассемблер. Кроме этого, файл нужно дополнительно отредактировать: закомментировать или удалить строчку «bl __libc_init_array».

В папку STM32F10x_StdPeriph_Driver следует скопировать папки inc и src, вместе с вложенными в неё файлами, из <zip папка>\an3268\stm32vldiscovery_package\Libraries\STM32F10x_StdPeriph_Driver.

После этого, возвращаемся к Eclipse. В окне Project Explorer нужно выбрать папку с именем проекта (DISCOVER) и нажать на клавиатуре клавишу F5 или из контекстного меню выбрать пункт «Refresh». Это приведёт к считыванию новой структуры файлов и папок, созданной Вами на предыдущем шаге.

После наполнения проекта файлами произведём его настройку. Здесь опять следуют выбрать папку проекта и нажать комбинацию клавиш Alt+Enter или из контекстного меню пункт Properties. Должно появиться новое окно Properties for DISCOVER, как на рисунке 6.

В левой части окна следует выбрать необходимый пункт, как на рисунке 6. В правой части, на вкладке Tool Settings будем последовательно выбирать пункты, отмеченные на рисунке 6 красной линией.

Target Processor – нужно выбрать из выпадающего списка поля Processor cortex-m3 и задать необходимый набор инструкций ассемблера поставив галочку напротив пункта Thumb (-mthumb).

Preprocessor – необходимо добавить определения USE_STDPERIPH_DRIVER, и STM32F10X_MD_VL препроцессора языка C для нормальной работы файлов из папки

StdPeripheralDriver. Для этого в поле *Defined Symbols (-D)* (справа от этого названия) следует нажать кнопку *Add...* (картинка с плюсом) и ввести названия перечисленных определений.

Directories — необходимо указать пути где компилятор будет искать заголовочные файлы и файлы реализации. Здесь так же как и с препроцессором, следует воспользоваться кнопкой *Add...* Лучше дополнительно задействовать кнопку *Workspace...* в следующем окне, чтобы пути были относительными. Достаточно добавить следующие папки:

1. *workspace_loc:\${ProjName}/CMSIS;*
2. *workspace_loc:\${ProjName}/STM32F10x_*

3. *workspace_loc:\${ProjName}/inc;*
4. *workspace_loc:\${ProjName}/Utilities.*

Optimization — поставить галочки напротив *Function sections (-ffunction-sections)* и *Function sections (-fdata-sections)*.

General – в поле *Script file (-T)* ввести строку *\${workspace_loc:\${ProjName}/STM32F100RB_FLASH.ld}*, и поставить галочки напротив *Do not use standart start files (-nostartfiles)* и *Remove unused sections (-Xlinker —gc-sections)*.

Подтверждая все произведённые настройки нажимаем кнопку «Apply», в правом нижнем углу окна и далее «OK».

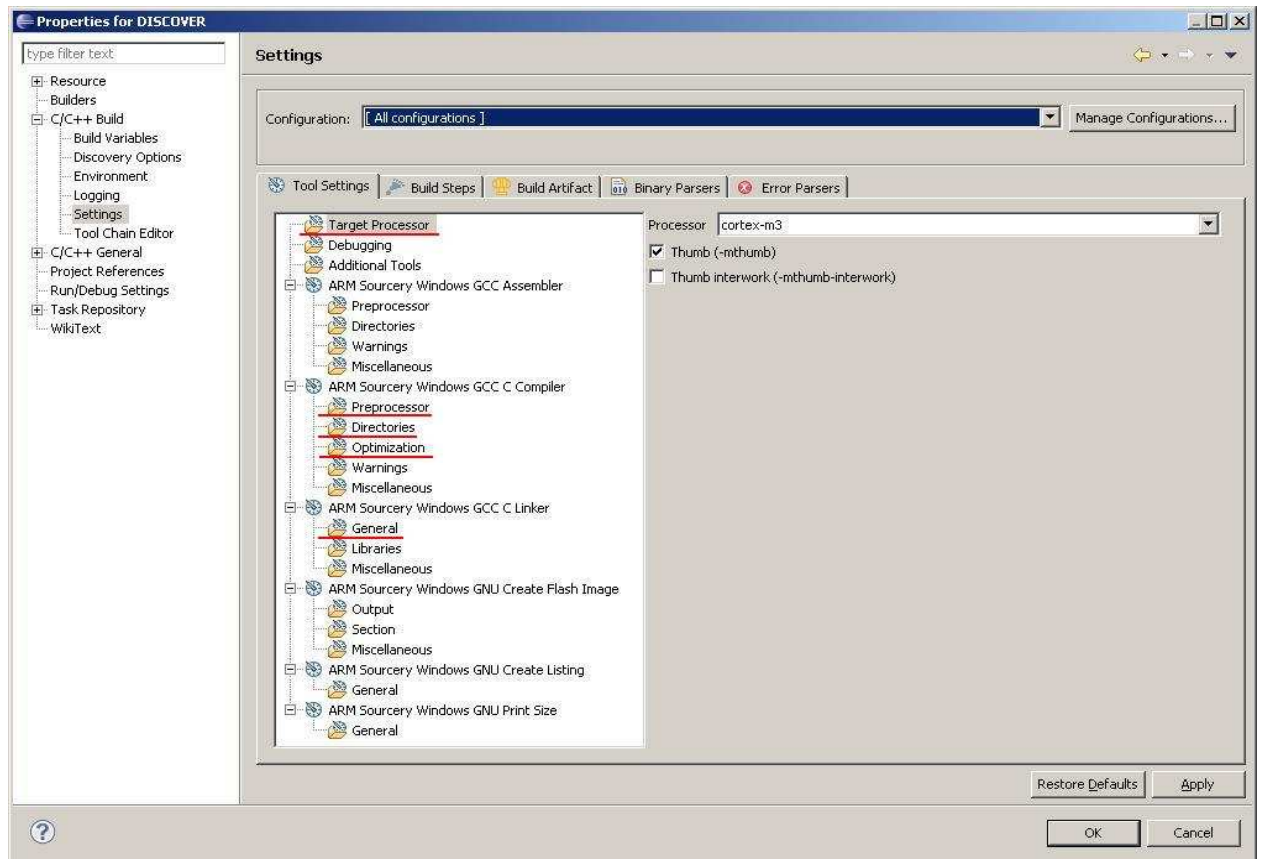


Рис. 6 - Настройка свойств проекта

Теперь все подготовительные операции произведены — можно запустить процесс компиляции. Делается это следующим способом: выделяется папка с именем проекта (*DISCOVER*), в окне *Project Explorer* главного окна *Eclipse* и из контекстного меню выбирается пункт *Build Project*. Если процесс компиляции пройдёт успешно, то на вкладке *Problems* внизу окна не должно быть ни каких записей.

ПРОГРАММИРОВАНИЕ STM32VLDISCOVERY

Настала пора увидеть работу программы. Для этого проверьте, замкнуты ли соседние выводы разъёма *CN3* и подключите отладочную плату *STM32VLDISCOVERY* к *USB* порту компьютера,

предварительно установив всё необходимое ПО (см. пункт выше). Если всё сделано правильно, то должно появиться окно съёмного диска, сигнализирующее о том, что *ST-LINK* определил запоминающее устройство развёрнутое на отладочной плате.

Теперь необходимо загрузить и установить программу *STM32 ST-LINK Utility* на персональный компьютер. Загрузить её можно с официального сайта пройдя по ссылке [14]. После установки, её следует запустить и выполнив команду из главного меню *Target* → *Automatic mode...* вызвать окно *Automatic mode* представленное на рисунке 7.

Теперь все подготовительные операции произведены — можно запустить процесс компиляции. Делается это следующим способом:

выделяется папка с именем проекта (*DISCOVER*), в окне *Project Explorer* главного окна *Eclipse* и из контекстного меню выбирается пункт *Build Project*. Если процесс компиляции пройдёт успешно, то на вкладке *Problems* внизу окна не должно быть ни каких записей.

По рассмотренному предмету авторами получены свидетельства на регистрацию программных продуктов {15 – 17}.

ПРОГРАММИРОВАНИЕ STM32VLDISCOVERY

Настала пора увидеть работу программы. Для этого проверьте, замкнуты ли соседние выводы разъёма *CN3* и подключите отладочную плату *STM32VLDISCOVERY* к *USB* порту компьютера, предварительно установив всё необходимое ПО (см. пункт выше). Если всё сделано правильно, то должно появиться окно съёмного диска, сигнализирующее о том, что *ST-LINK* определил запоминающее устройство развёрнутое на отладочной плате.

Теперь необходимо загрузить и установить программу *STM32 ST-LINK Utility* на персональный компьютер. Загрузить её можно с официального сайта пройдя по ссылке [14]. После установки, её следует запустить и выполнив команду из главного меню *Target* → *Automatic mode...* вызвать окно *Automatic mode* представленное на *рисунке 7*.

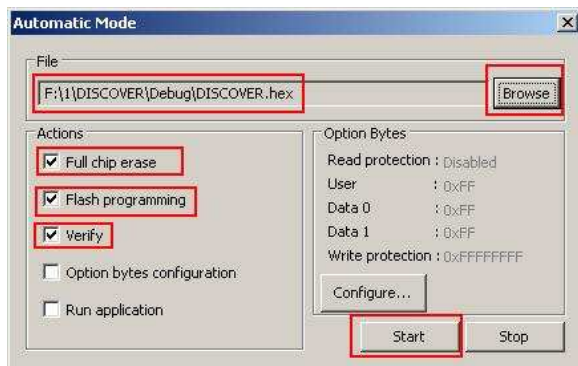


Рис. 7 - Окно Automatic Mode

Воспользовавшись кнопкой *Browse* выберите *hex-файл* созданный на стадии компиляции (он должен находится в папке *Debug* папки проекта). Установите галочки, как показано на *рисунке 7*, и нажмите кнопку *Start*. После чего, программа будет записана во внутреннюю *FLASH* память микроконтроллера отладочной платы. Далее нужно нажать на кнопке *Stop* и закрыть открытые окна. На этом процесс программирования закончен. Чтобы посмотреть работу программы следует перезагрузить микроконтроллер нажав чёрную кнопку *RST* или снять и снова подать напряжение питания на отладочную плату.

Таким образом, у Вас появился инструмент создания своих собственных и изменения

предлагаемых производителем программ для микроконтроллеров ARM. В этом нелегком, но интересном процессе Вам помогут широкие возможности среды *Eclipse*.

ВЫВОДЫ

- На рынке микроконтроллеров в настоящий момент времени существуют достаточно доступные решения для быстрого и удобного освоения открытой архитектуры ARM.
- При определённом уровне упорства, можно бесплатно получить полноценную интегрированную среду разработки программ с широким возможностями.
- Компания *ST Microelectronics* идёт на встречу своим клиентам предлагая, за небольшие деньги, программатор и микроконтроллер на одной плате, что позволяет быстро создавать разнообразное количество цифровых систем управления.

ЛИТЕРАТУРА

[1] ARM (архитектура) Материал из Википедии — свободной энциклопедии. URL:

http://ru.wikipedia.org/wiki/ARM_%28%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0%29#cite_note-Krazit-1 (дата обращения 3.10.13).

[2] Официальный сайт фирмы *STMicroelectronics* URL: <http://www.st.com/web/en/home.html> (дата обращения 3.10.13).

[3] *STM32 32-bit ARM Cortex MCUs* URL: <http://www.st.com/web/en/catalog/mmc/FM141/SC1169> (дата обращения 3.10.13).

[4] “Сравниваем микроконтроллеры до 50 рублей: ARM жаждет крови” URL: <http://habrahabr.ru/post/120611/> (дата обращения 3.10.13).

[5] Изображение отладочной платы *STM32VLDISCOVERY* на официальном сайте фирмы *STMicroelectronics* URL: http://www.st.com/st-web-ui/static/active/en/fragment/product_related/rpn_information/board_photo/board_stm32vl_discovery.jpg (дата обращения 3.10.13).

[6] *ST-LINK/V2 USB driver for Windows 7, Vista and XP* URL: <http://www.st.com/web/en/catalog/tools/PF258167> (дата обращения 3.10.13).

[7] *Eclipse* (среда разработки) Материал из Википедии — свободной энциклопедии. URL: http://ru.wikipedia.org/wiki/Eclipse_%28%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8%29 (дата обращения 3.10.13).

[8] *Eclipse downloads* URL: <http://www.eclipse.org/downloads/> (дата обращения 3.10.13).

[9] *Sourcery CodeBench Lite Edition for ARM EABI* URL: <http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/editions/lite-edition/request?id=e023fac2-e611-476b-a702-90eabb2aeca8&downloadlite=scblite2012&fmpath=/embedded-software/sourcery-tools/sourcery->

codebench/editions/lite-edition/form (дата обращения 3.10.13).

[10] Загрузить Java машину

http://www.java.com/ru/download/windows_xpi.jsp?locale=ru (дата обращения 3.10.13).

[11] GNU ARM Eclipse An Eclipse CDT Manged Build Extension for GNU ARM Toolchains URL:

<http://gnuarmeclipse.livius.net/blog/> (дата обращения 3.10.13).

[12] GNU ARM Eclipse | Downloads URL:

<http://gnuarmeclipse.livius.net/blog/downloads/> (дата обращения 3.10.13).

[13] STM32VLDISCOVERY firmware package (AN3268) URL:

<http://www.st.com/web/en/catalog/tools/PF257914> (дата обращения 3.10.13).

[14] STM32 ST-LINK utility URL:

<http://www.st.com/web/en/catalog/tools/PF258168> (дата обращения 3.10.13).

[15] Свидетельство о регистрации программы для ЭВМ № 2013617769 Bluetooth_HC-05_Driver-v001. Правообладатель: ФГБОУ ВПО НГТУ. Авторы: Ескин А.В., Жмудь В.А. Заявка № 2013615378 от 13 февраля 2013 г. Зарегистрировано в Реестре программ для ЭВМ 22 августа 2013 г.

[16] В.А. Жмудь, В.Г. Трубин. НС-SR04_Ultrasonic_Driver-v002 Свидетельство на регистрацию программы для ЭВМ. Заявка № 2013616195. Дата поступления 18 июля 2013. Дата гос. регистрации в реестре программ для ЭВМ 04 сентября 2013 г. Свид. № 2013618271. Правообладатель: ФГБОУ ВПО НГТУ, Россия.

[17] В.А. Жмудь, В.Г. Трубин. NXT_MotorDriver-v003a Заявка на регистрацию программы для ЭВМ.

Заявка № 2013616195. Дата поступления 26 апреля 2013. Дата гос. регистрации в реестре программ для ЭВМ 05 июня 2013 г. Свид. № 2013615343.

Правообладатель: ФГБОУ ВПО НГТУ, Россия.



Алексей Викторович Ескин - ведущий инженер ООО «КБ Автоматика»,

E-mail: kba-elma@bk.ru



Вадим Аркадьевич Жмудь – заведующий кафедрой Автоматики НГТУ, профессор, доктор технических наук, автор более 200 научных статей, включая 10 патентов и 6 учебных пособий. Область научных интересов и компетенций – теория автоматического управления, электроника, лазерные системы, оптимизация, измерительная техника.

E-mail: oao_nips@bk.ru



Виталий Геннадьевич Трубин - зав. лаб. кафедры Автоматики НГТУ, директор ООО «КБ Автоматика». Автор 18 научных статей. Область интересов – разработка специализированной электроники.

E-mail: trubin@ngs.ru