

Контроллеры ARMatura

© Dmitry Ponyatov <dponyatov@gmail.com>, SSAU ASCL

28 февраля 2013 г.

Оглавление

Оглавление	1
I Введение	5
II Железо	7
1 STM32VLDISCOVERY /STM32F100RBT6/	8
2 STM32F4DISCOVERY /SRM32F407VGT6/	10
3 ARMatura /STM32F417IGT/	12
4 PION /STM32F100C4T6B/	13
III Первые шаги	14
5 Установка пакета ПО STlink	15
6 Установка Keil MDK-ARM	21
7 Первый проект: blink	28
7.1 Настройки проекта в Keil	29
7.2 Структура файлов	42
8 Hell Of World	43

IV Средства разработки	44
9 Keil MDK-ARM	45
10 Компиляторы	46
10.1 GCC	46
10.2 KeilCC	46
10.3 IAR	46
11 IDE	47
11.1 Eclipse	47
11.2 Code::Blocks	47
11.3 gVim	47
11.4 Keil uVision	47
11.5 IAR	47
12 Программаторы	48
12.1 STlink	48
12.2 Serial Boot	48
13 Отладчики	49
13.1 JTAG	49
13.2 STM32 SWD	49
13.3 GDB	49
V Основы языка C⁺	50
14 Синтаксис	51
15 Типы данных	52
16 Стандартная библиотека libc	53
VI Отладка	54
17 JTAG	55
18 GDB	56
19 OpenOCD	57
VICMSIS	58
20 Startup	59
21 Стандартная библиотека STM32	60
22 USB client/host	61

<i>Оглавление</i>	3
VII Ядро Cortex-Mx	62
23 Режимы ARM и Thumb	63
24 DMA	64
25 DSP /Cortex-M3/	65
26 FPU /Cortex-M4F/	66
IX Интерфейсы	67
27 USB	68
28 UART	69
29 SPI	70
30 I2C	71
31 CAN	72
X Операционные системы OCPB	73
32 Keil RTX	74
33 FreeRTOS	75
34 eCos	76
35 Linux	77
XI Стек TCP/IP	78
36 Ethernet	79
37 PPP	80
XII Типовые применения	81
38 GPS	82
38.1 Протокол NMEA 0183	82
38.2 \$GPRMC — рекомендуемый минимум навигационных данных	83
38.3 Tistar15	84
38.4 WISMO228	84
39 GSM	85
39.1 WISMO228	85
40 шина Dallas 1Wire	86

40.1 RTC	86
40.2 Датчики температуры DS18x20	86
XIIИстраиваемый Linux	87
XIIIQA	88
XIVПриложения	90
41 Сводная таблица процессоров	91
41.1 STM32F10x	91

Часть I

Введение

Эта книга – набор методичек по разработке ПО для встраиваемых систем, написанных для Института космического приборостроения СГАУ.

Для применения в реальных проектах научной аппаратуры была разработана линейка унифицированных модулей:

1. ARMatura — модуль на мощном микропроцессоре STM32F417IGT: 1M Flash, 192K SRAM, TQFP176, DSP, FPU,.. [41](#)

предназначен для использования в качестве центрального процессора цифровой системы: обработка данных, сложные алгоритмы управления, ЦОС, вычисления, реализация протоколов передачи данных по интерфейсам USB, Ethernet, RS232/UART, SPI, I2C, CAN,..

2. PION [4](#) — модуль на самом простом и дешевом STM32F100C4T6B: 128K Flash, 8K SRAM, UART, SPI [41.1](#)

периферийный модуль для стыковки с аналоговыми датчиками и исполнительными устройствами, предварительная ЦОС обработка, передача данных на ARMatura-модули для дальнейшей обработки данных.

также модуль применим в качестве самостоятельного простого интерфейса при замене на чип STM32F103 с портом USB или установки внешних интерфейсных микросхем FT232RL (USB Serial), CP1202, MC1551 (CAN).

3. BACKPLANE — коммутационная плата межмодульного интерфейса
4. POWER — модуль импульсного источника питания
5. STEPPER — модуль управления двухфазным шаговым двигателем
6. WISMO — несущая плата для GPS/GSM модуля WISMO 228
7. QVGA — несущая плата для TFT touch-панели

В качестве базового микроконтроллера были выбраны чипы семейства STM32Fxxx с ядрами Cortex-M3, Cortex-M4F (ARM) как самые дешевые, и имеющие хорошую поддержку в виде отладочных плат линейки Discovery.

В общем, линейка модулей ARMatura может рассматриваться в качестве замены устаревшей линейки периферийных контроллеров Arduino на базе МК AVR8.

Проект размещен в репозитории <https://github.com/ponyatov/ARMatura.git> и предоставляется на условиях OpenHardware licence (за исключением прошивок и схем по тематике ИКП СГАУ).

Контакты разработчиков:

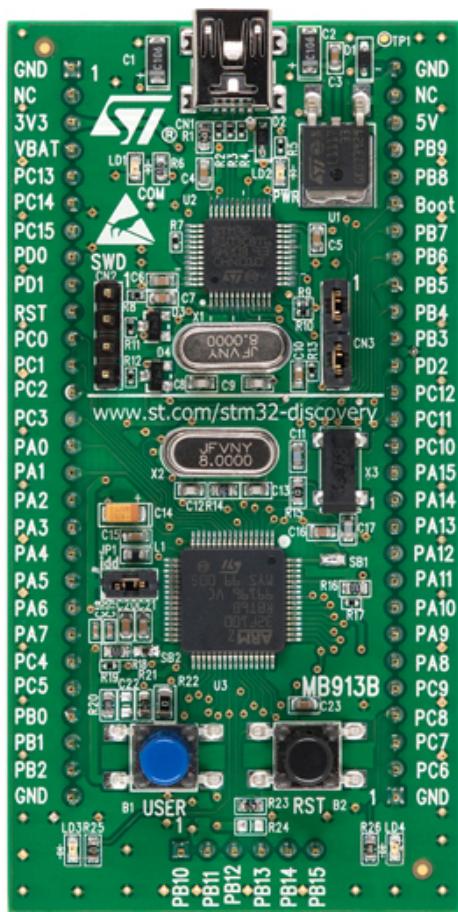
- ИКП СГАУ <semkin@ssau.ru>
- Дмитрий Понятов <dponyatov@gmail.com>

Часть II

Железо

Глава 1

STM32VLDISCOVERY /STM32F100RBT6/



- Микроконтроллер STM32F100RB [41](#), 128 KB Flash, 8 KB RAM in 64-pin LQFP
- Встроенный ST-Link с возможностью использования в режиме внешнего программатора (только с SWD коннектором)
- Разработана для питания как от USB, так и от внешнего источника 3.3 или 5 вольт
- Может обеспечить питание 3 и 5 вольт для внешних устройств
- Два пользовательских светодиода (зеленый и синий)
- Пользовательская кнопка USER

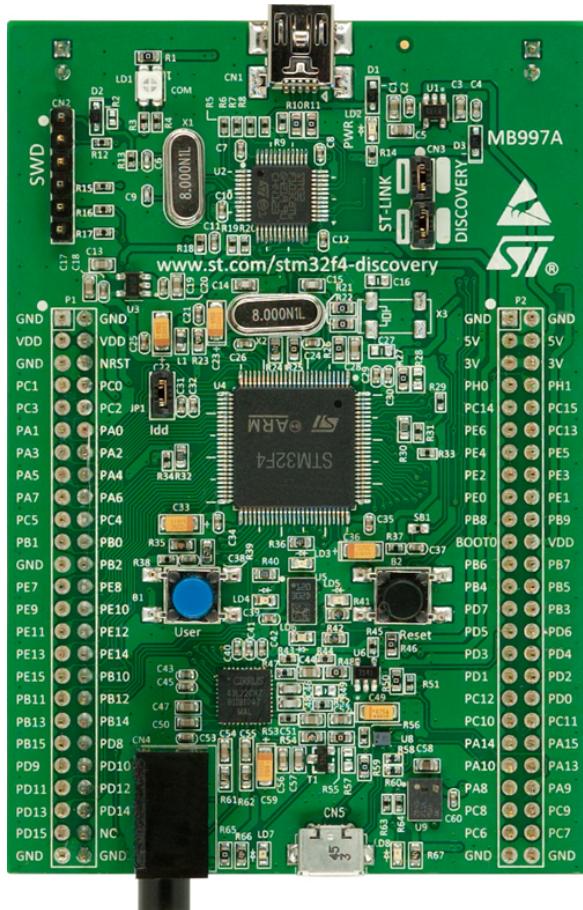
- Кнопка сброса RESET
- Контактные гребенки для всех выводов QFP64 для быстрого подключения и монтажа прототипа

Цена в розницу: 750 р.

<http://www.voltmaster.ru/cgi-bin/qwery.pl?id=127000573571&group=7000046>

Глава 2

STM32F4DISCOVERY /SRM32F407VGT6/



- микроконтроллер STM32F407VGT6 41 на базе 32-битного ядра Cortex-M4F, 1 MB Flash, 192 KB RAM в корпусе LQFP100
- встроенный ST-LINK/V2 с возможностью использования в режиме внешнего программатора (только с SWD коннектором)
- Разработана для питания как от USB, так и от внешнего источника 5 вольт
- Может обеспечить питание 3 и 5 вольт для внешних устройств
- LIS302DL, ST MEMS датчик движения, 3-осевой цифровой гироскоп
- MP45DT02, ST MEMS аудиодатчик, всенаправленный цифровой микрофон

- CS43L22, аудиоЖАП со встроенным аудиоусилителем класса D
- Восемь LEDs: LD1 (red/green) for USB communicationLD2 (red) for 3.3 V power onFour user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)2 USB OTG LEDs LD7 (green) VBus and LD8 (red) over-current
- Пользовательская кнопка USER
- Кнопка сброса RESET
- USB OTG FS with micro-AB connector
- Контактные гребенки для всех выводов LQFP100 ля быстрого подключения и мон-тажа прототипа

Цена в розницу: 1140 р.

<http://www.voltmaster.ru/cgi-bin/qwery.pl?id=127000854271&group=7000046>

Глава 3

ARMatura /STM32F417IGT/

Глава 4

PION /STM32F100C4T6B/

Модуль PION предназначен для мелких задач управления, первичной обработки данных,стыковки с устройствами измерения и исполнительными устройствами, т.е. для тех задач, для которых ранее использовались микроконтроллеры Atmel AVR8.

процессор	STM32F100C4T6B	41.1
ROM		16K
RAM		4K
шина	AUTObus	
интерфейсы	UART	1
	SPI	1
	AIЦП	10x12b
	ЦАП	2x12b
буфер	Parallel Flash	64K

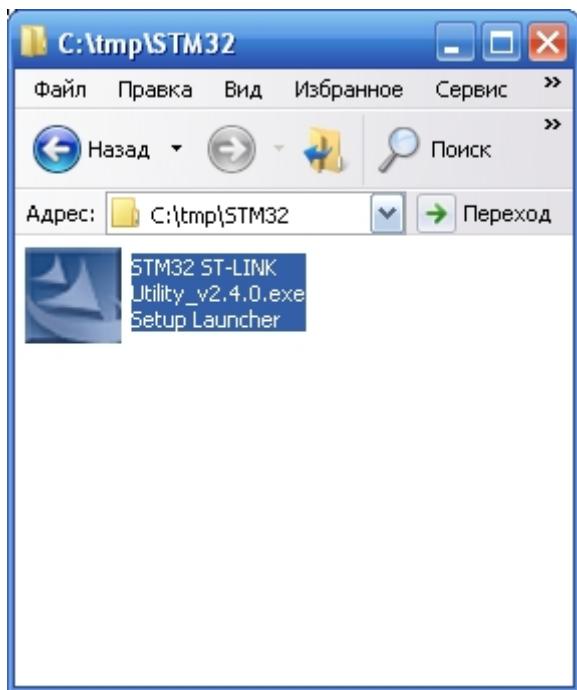
Часть III

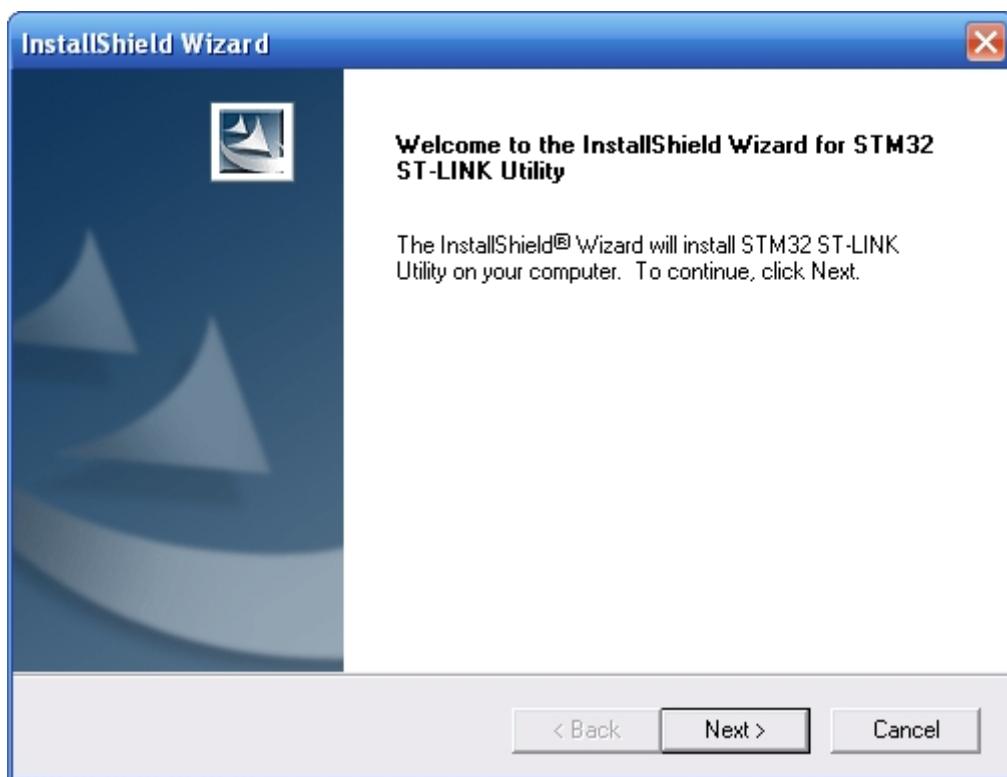
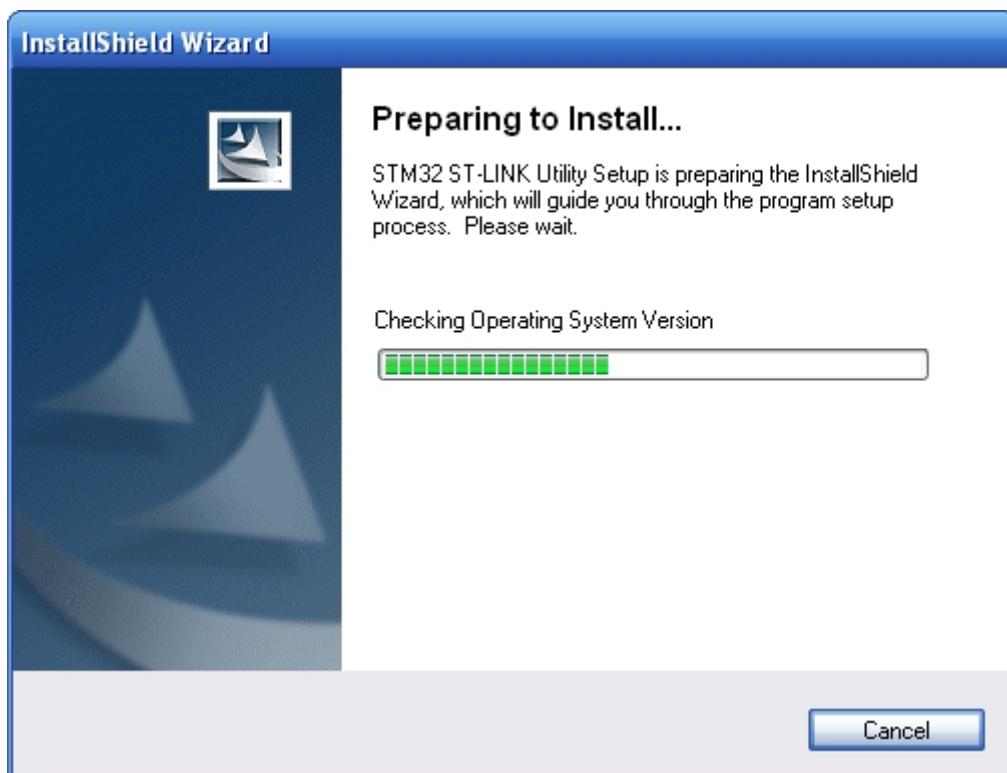
Первые шаги

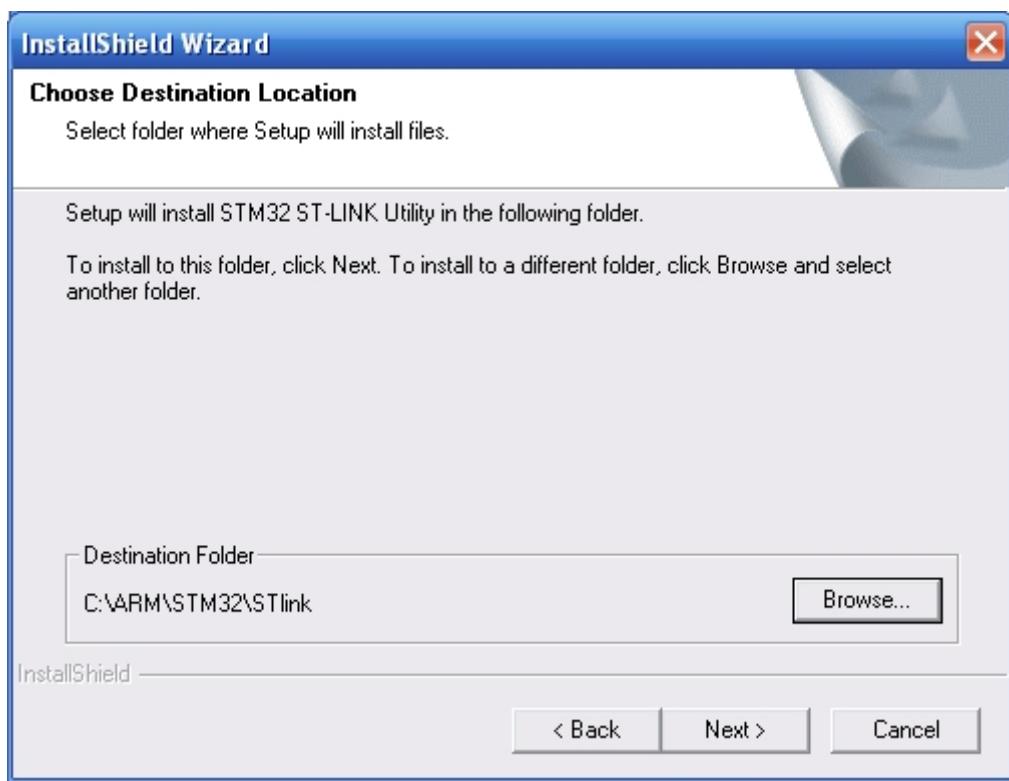
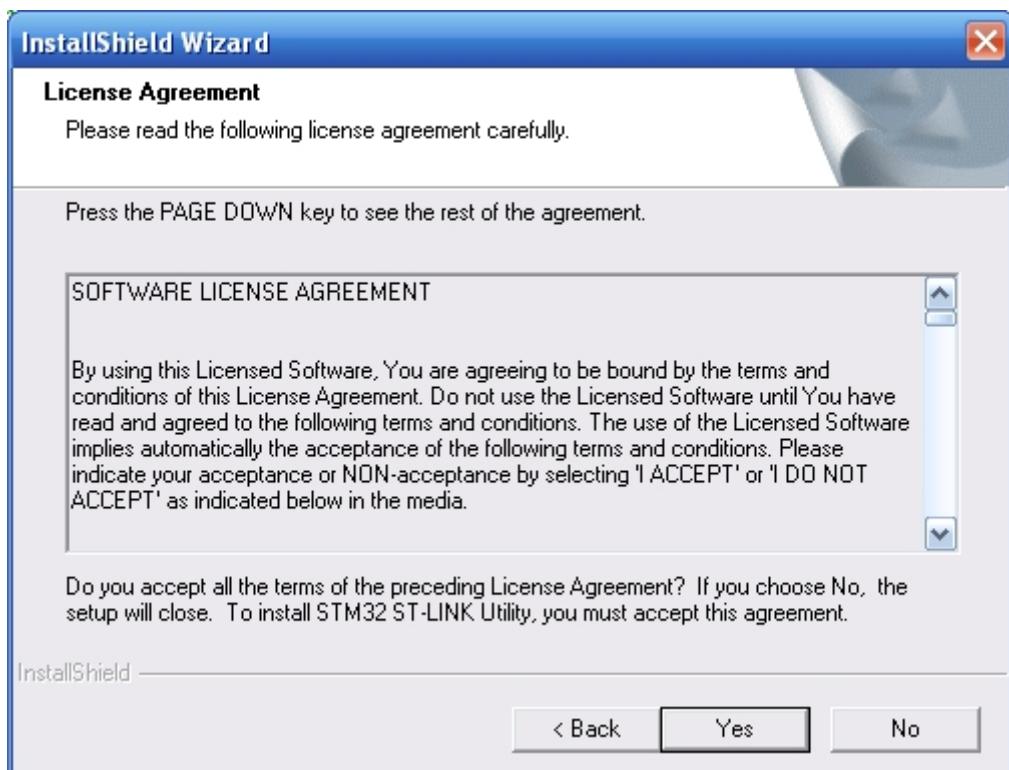
Глава 5

Установка пакета ПО STlink

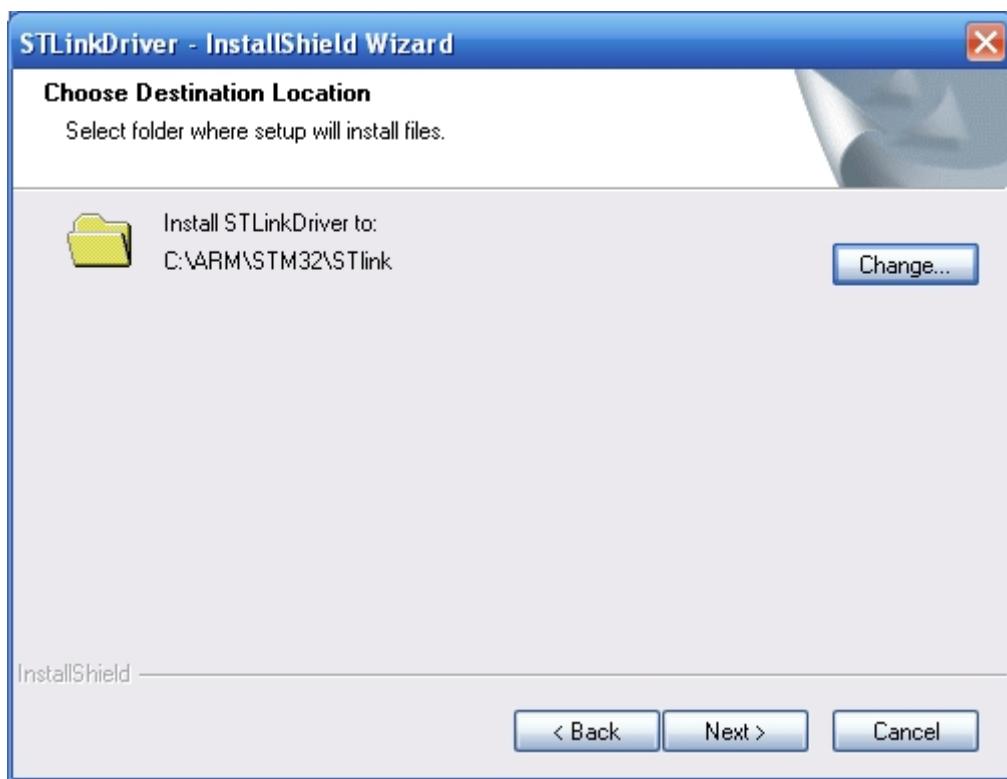
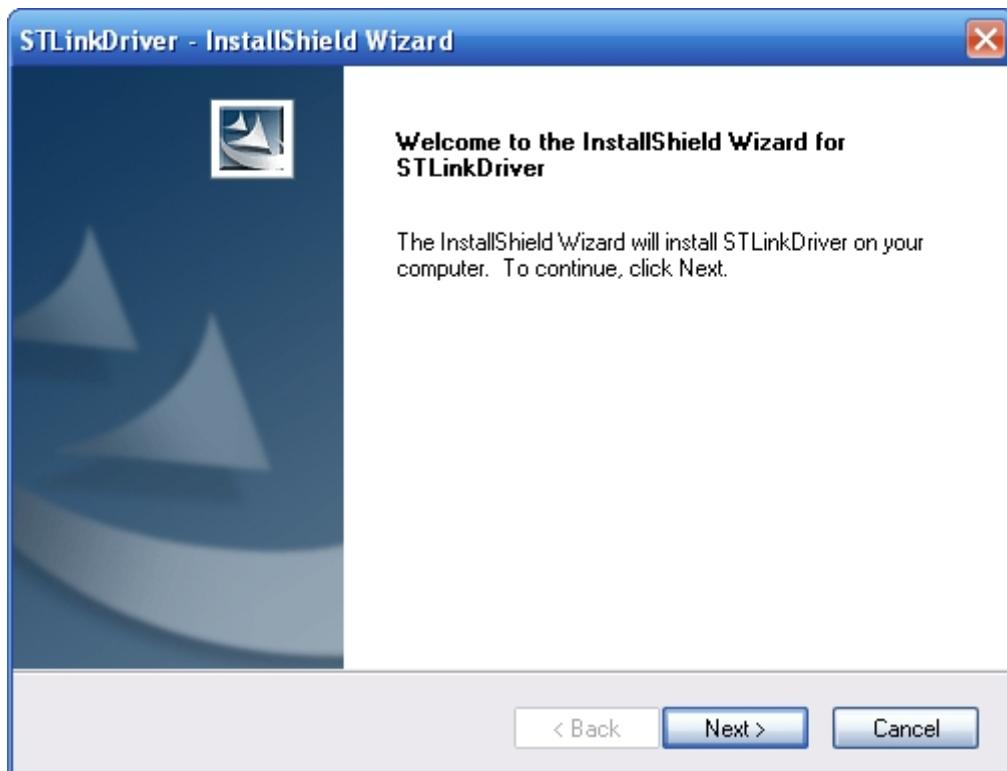
Для работы с чипами STM32 используя программаторы STlink, встроенные на демоплаты Discovery необходимо установить пакет STSW-LINK004 (STM32 ST-link Utility и драйвер), скачав его с <http://www.st.com/web/en/catalog/tools/PF258168>

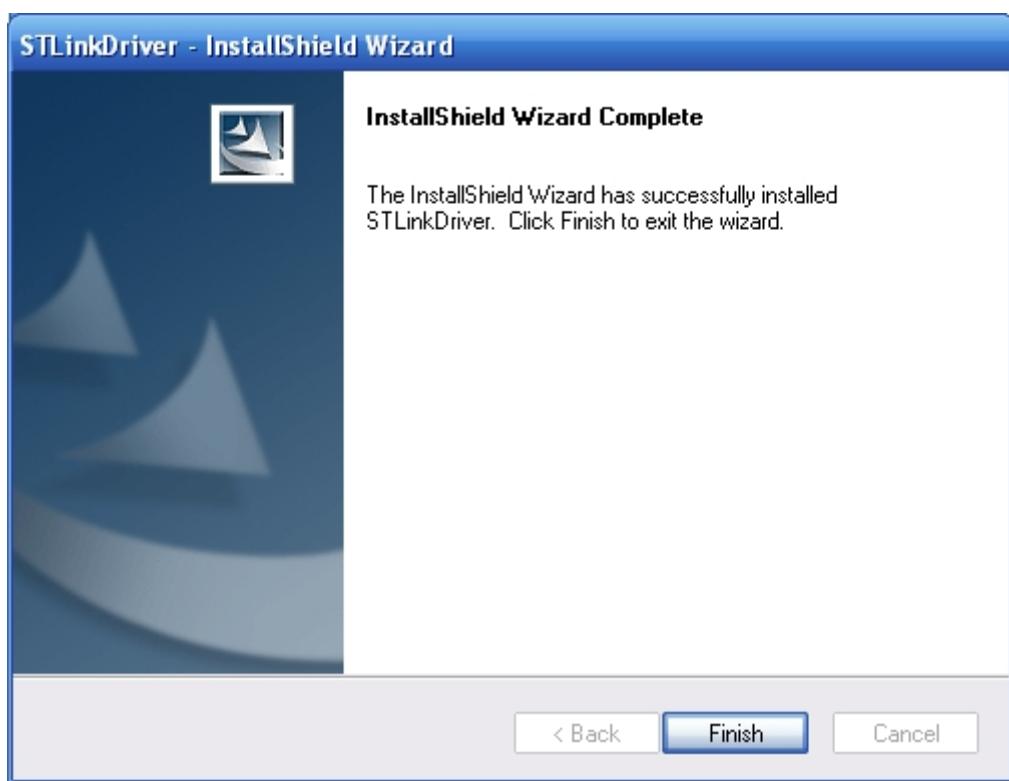
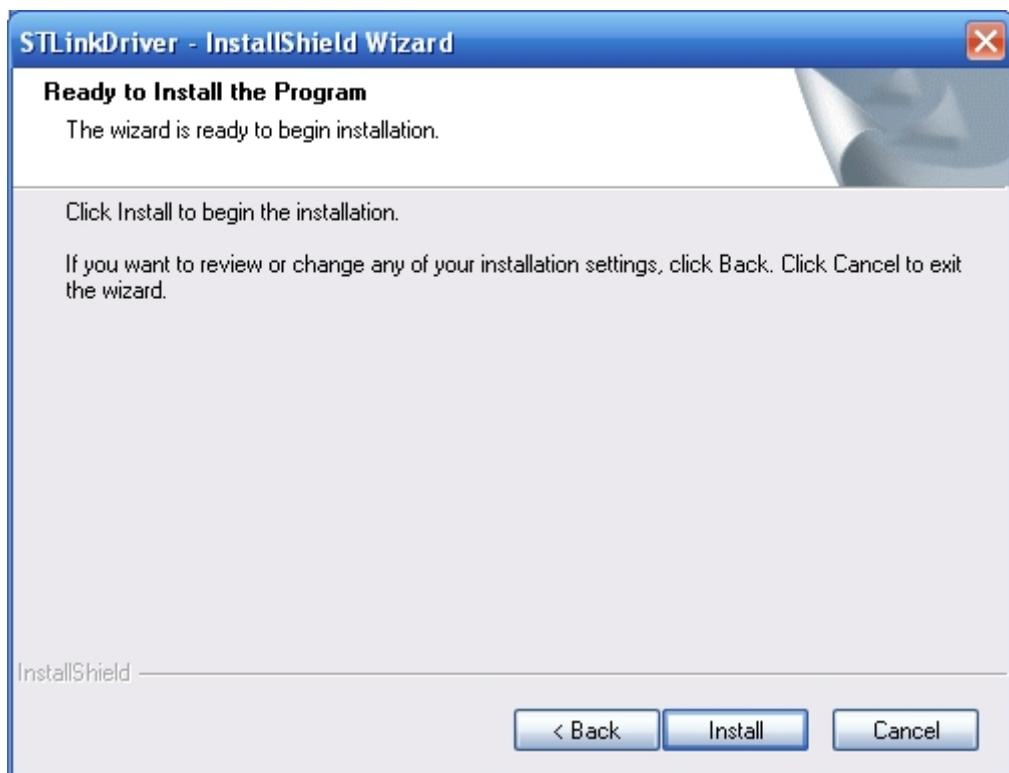


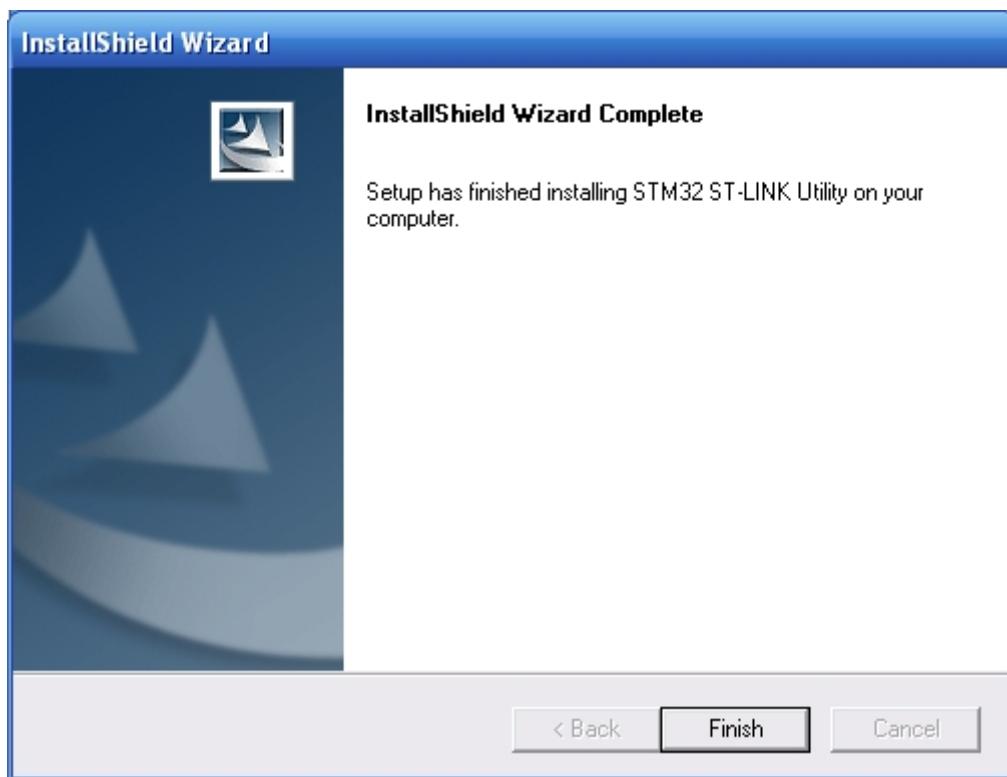




Пакет драйвера входит в пакет STSW-LINK004, и запускается автоматически







Глава 6

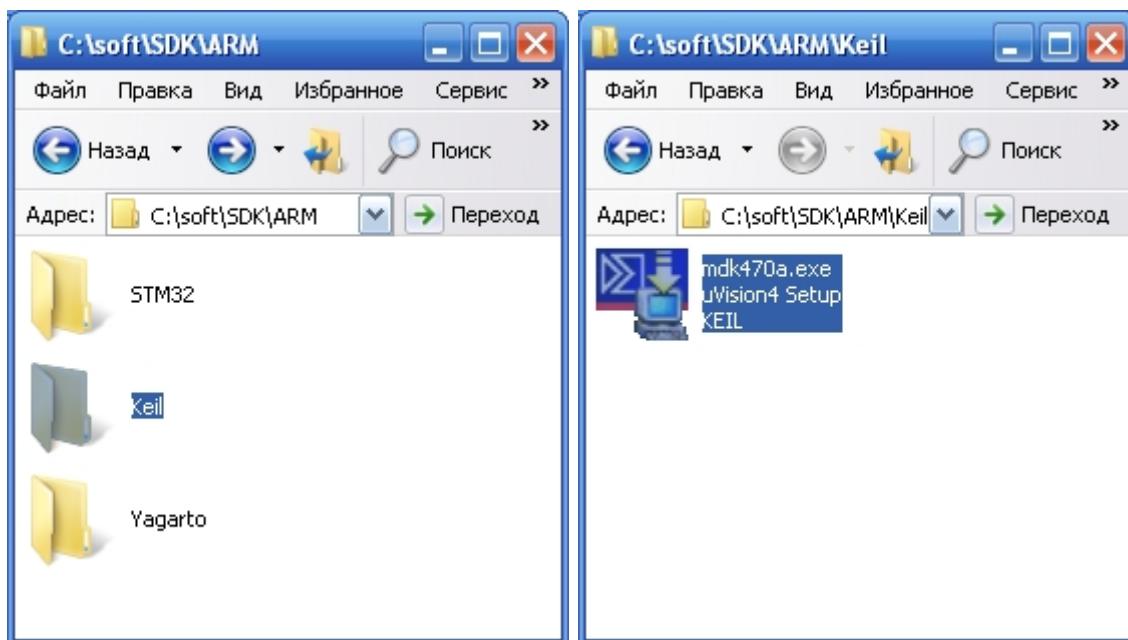
Установка Keil MDK-ARM

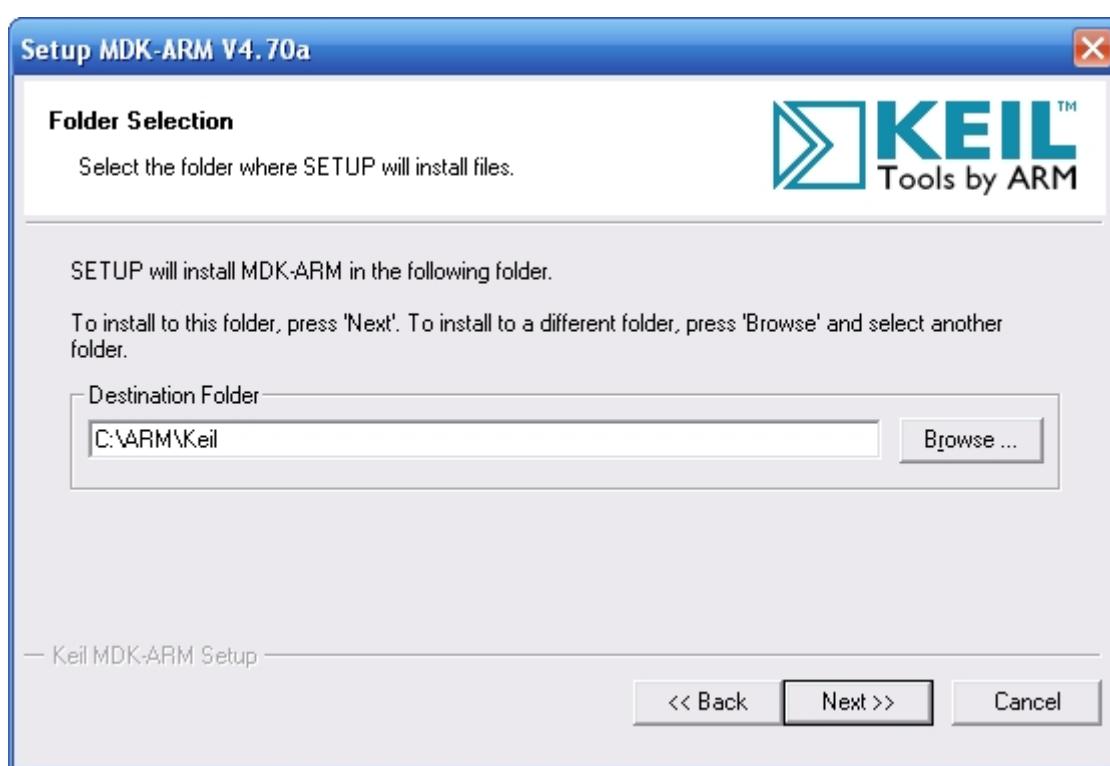
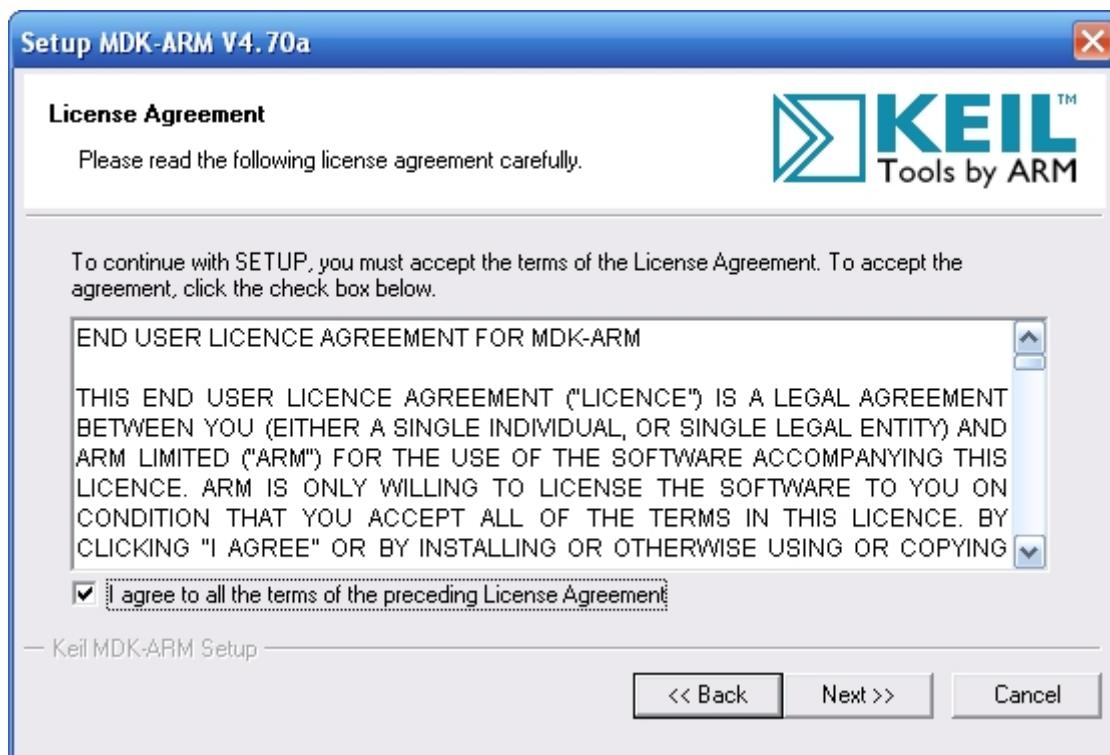


Для начала освоения программирования для ARM рекомендуем использовать бесплатный пакет от Keil: <http://www.keil.com/arm/mdk.asp> — ограничения бесплатной версии в 32К кода вполне достаточно для начального освоения программирования под процессоры семейства Cortex-Mx, а затем уже можно переползать на открытое ПО: GNU toolchain 10.1, Eclipse 11.1 и Linux XIII.

Процесс установки и первоначальной настройки описан в 9. В этом разделе будет рассмотрен только процесс установки и начальной настройки, подробно о пакете Keil MDK-ARM см.

Качаем пакет с официального сайта, заполнив анкету: <https://www.keil.com/demo/eval/arm.htm>.

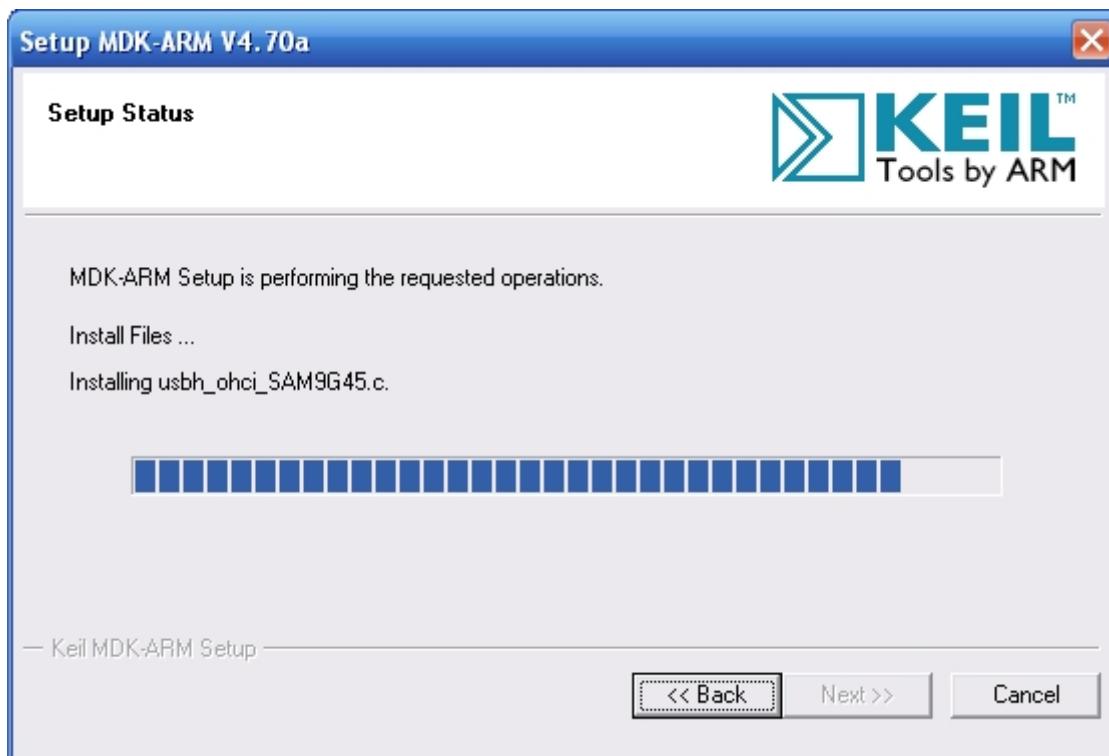


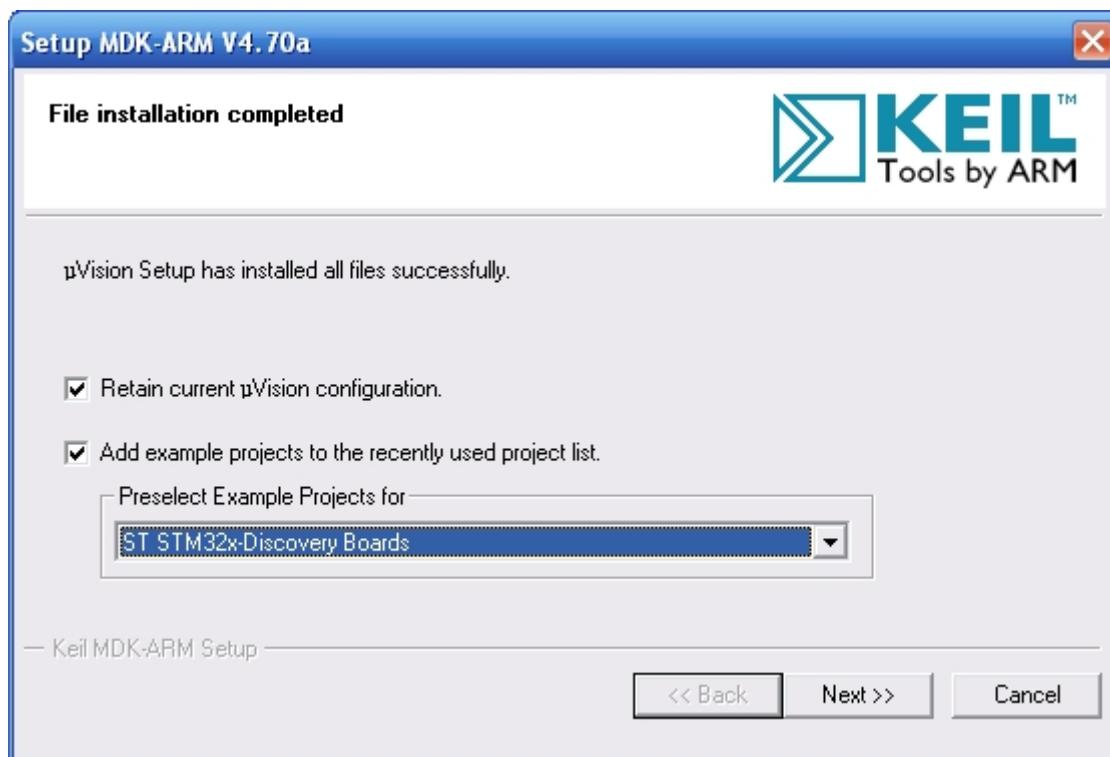


Путь установки пакета

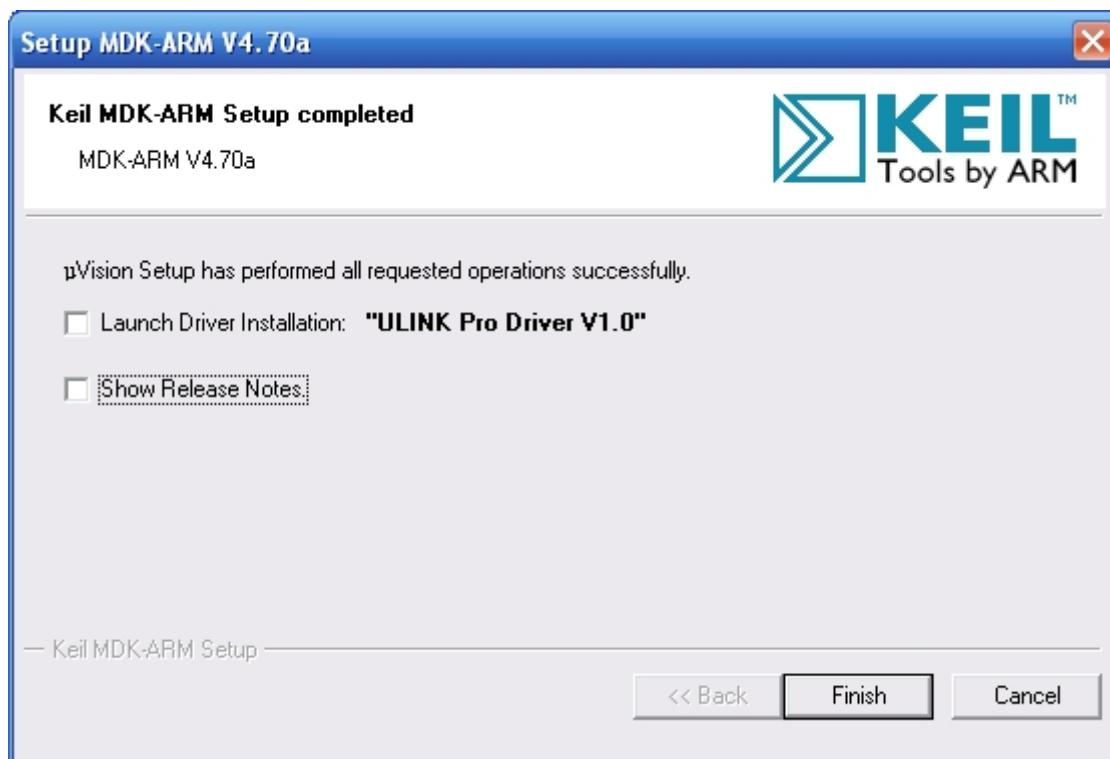


Личные данные: имя, название компании или hobbit, адрес электронной почты.





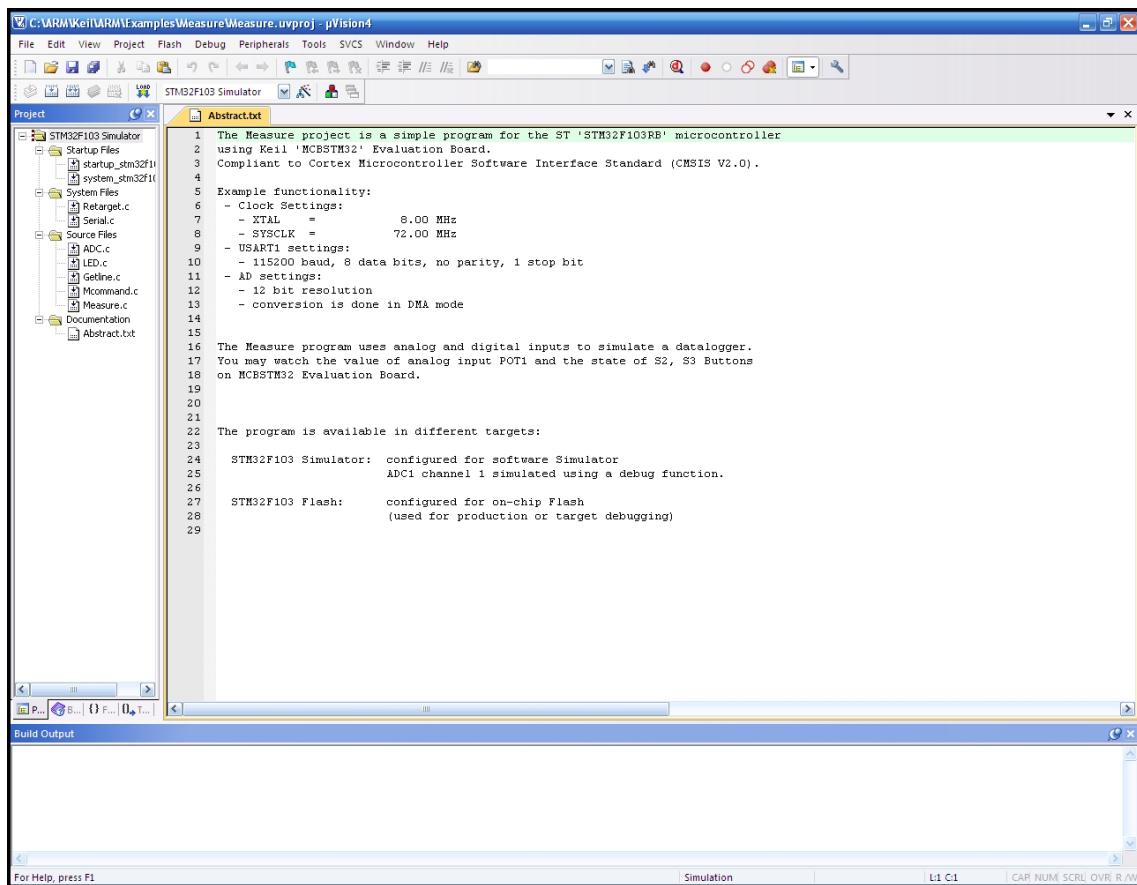
Укажите какие примеры кода добавить в список recently used project list: для работы с другими типами микропроцессоров выберете соответствующий раздел, или оставьте Simulation Hardware по умолчанию.



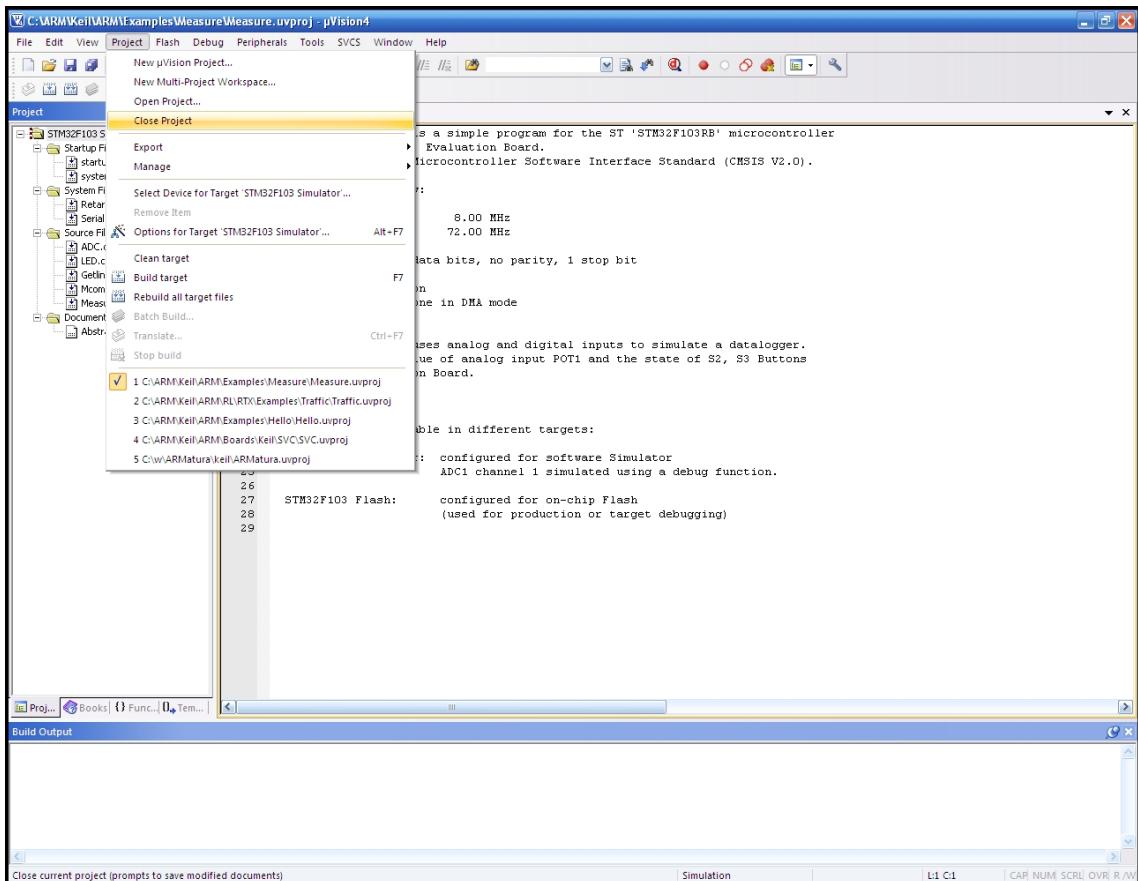
Снять установку драйвера программатора ULINK (если у вас его нет) и вывод текстового файла с последними изменениями Keil.



После запуска открывается проект по умолчанию, настроенный для программного симулятора STM32F103.



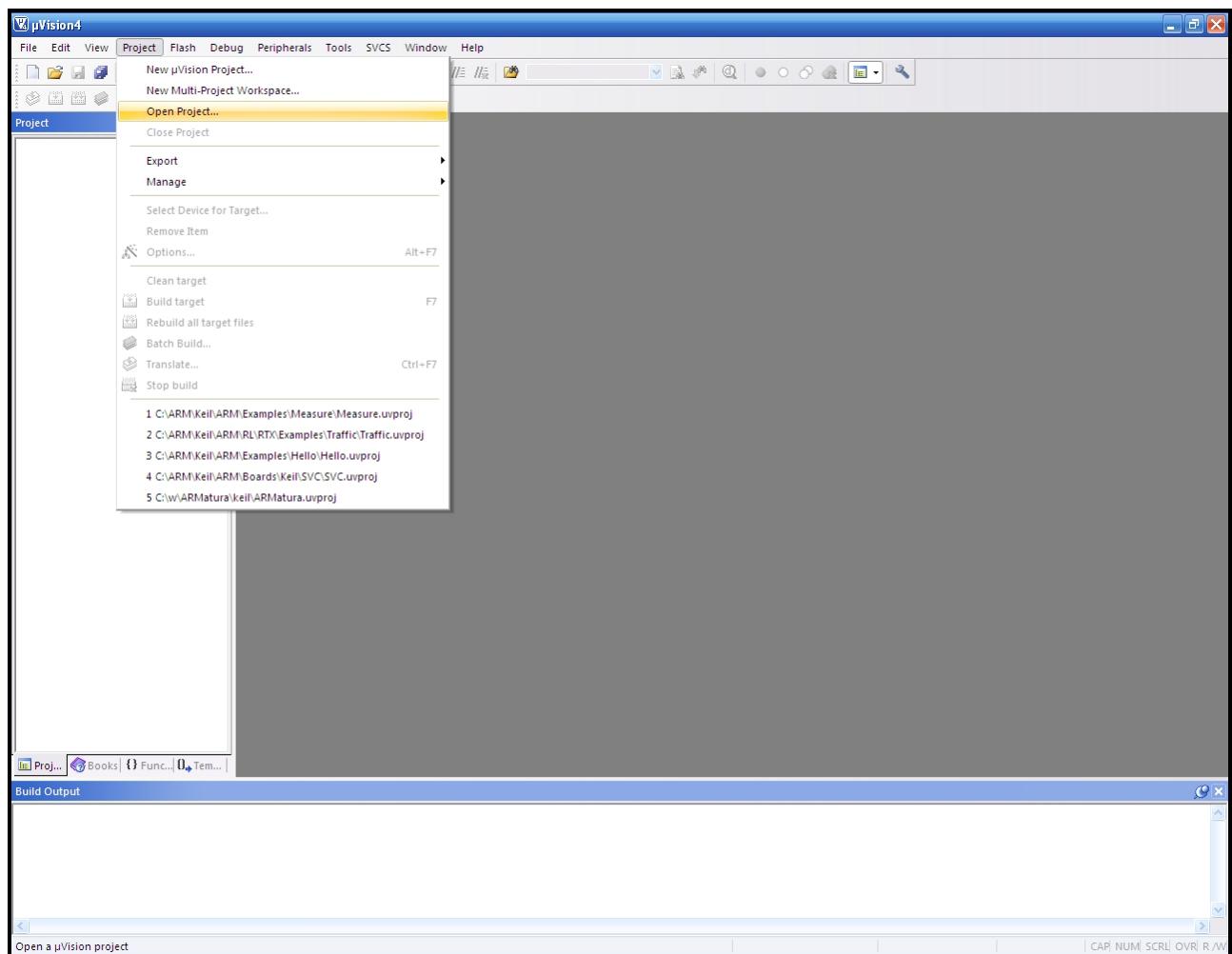
Нужно его закрыть,

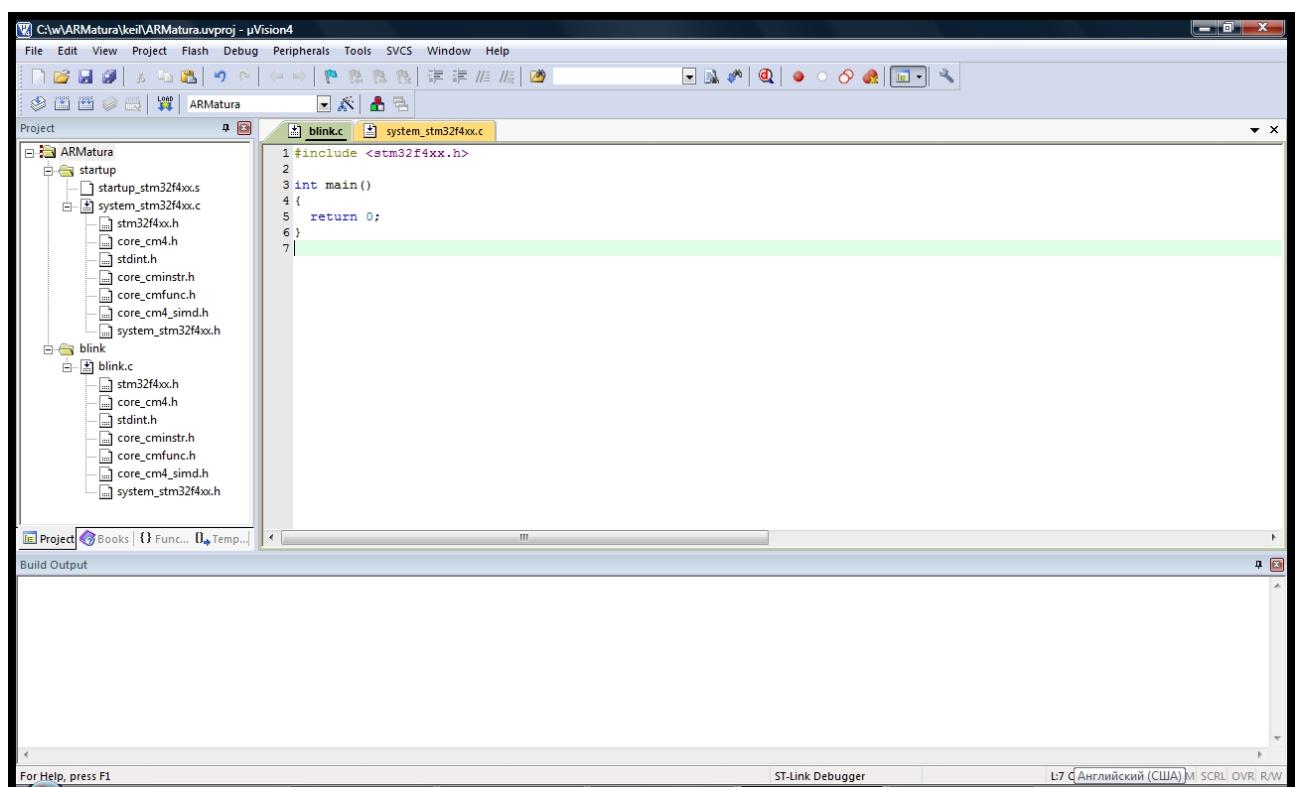
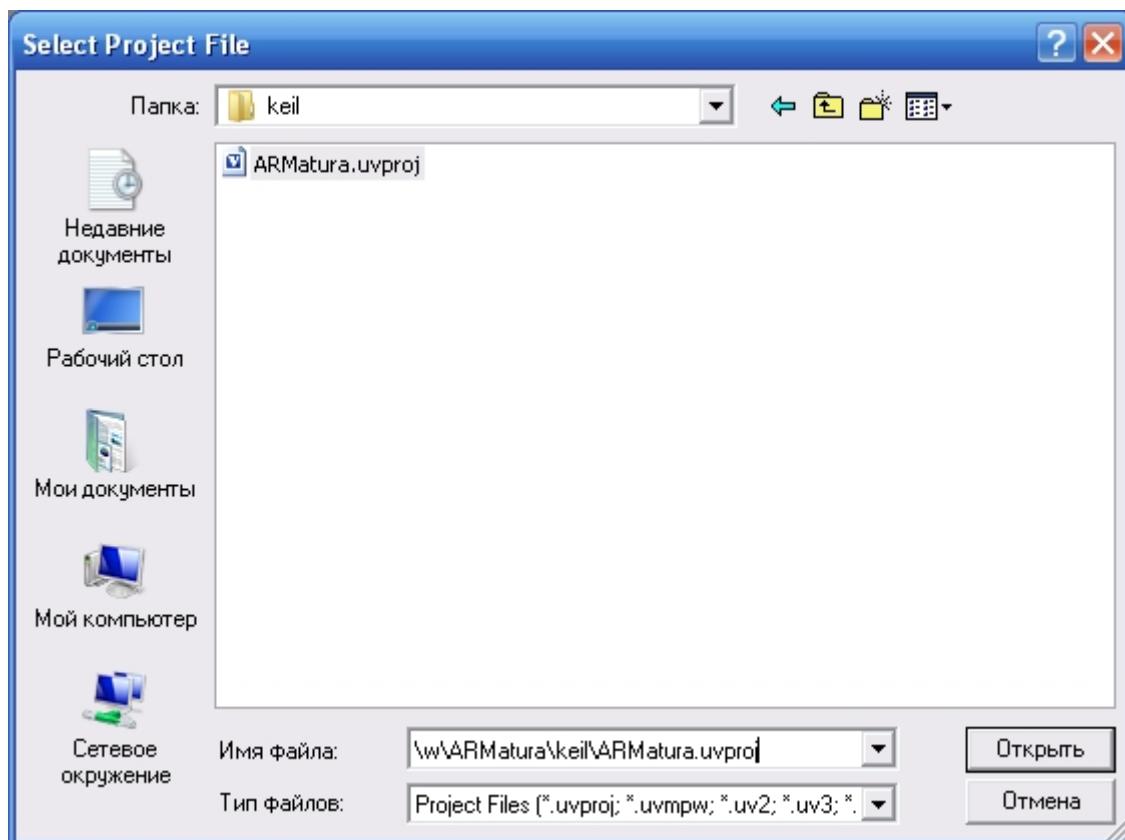


и открыть первый самый простой проект:

Глава 7

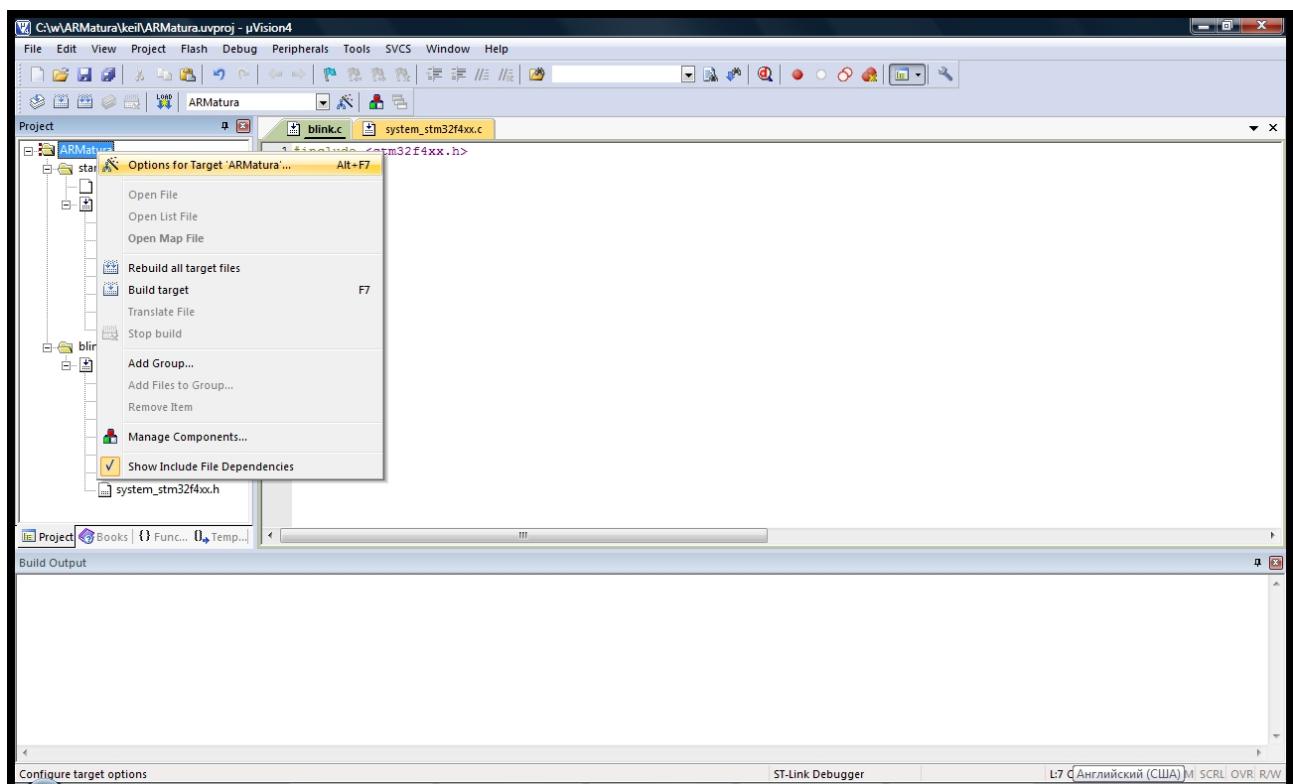
Первый проект: blink



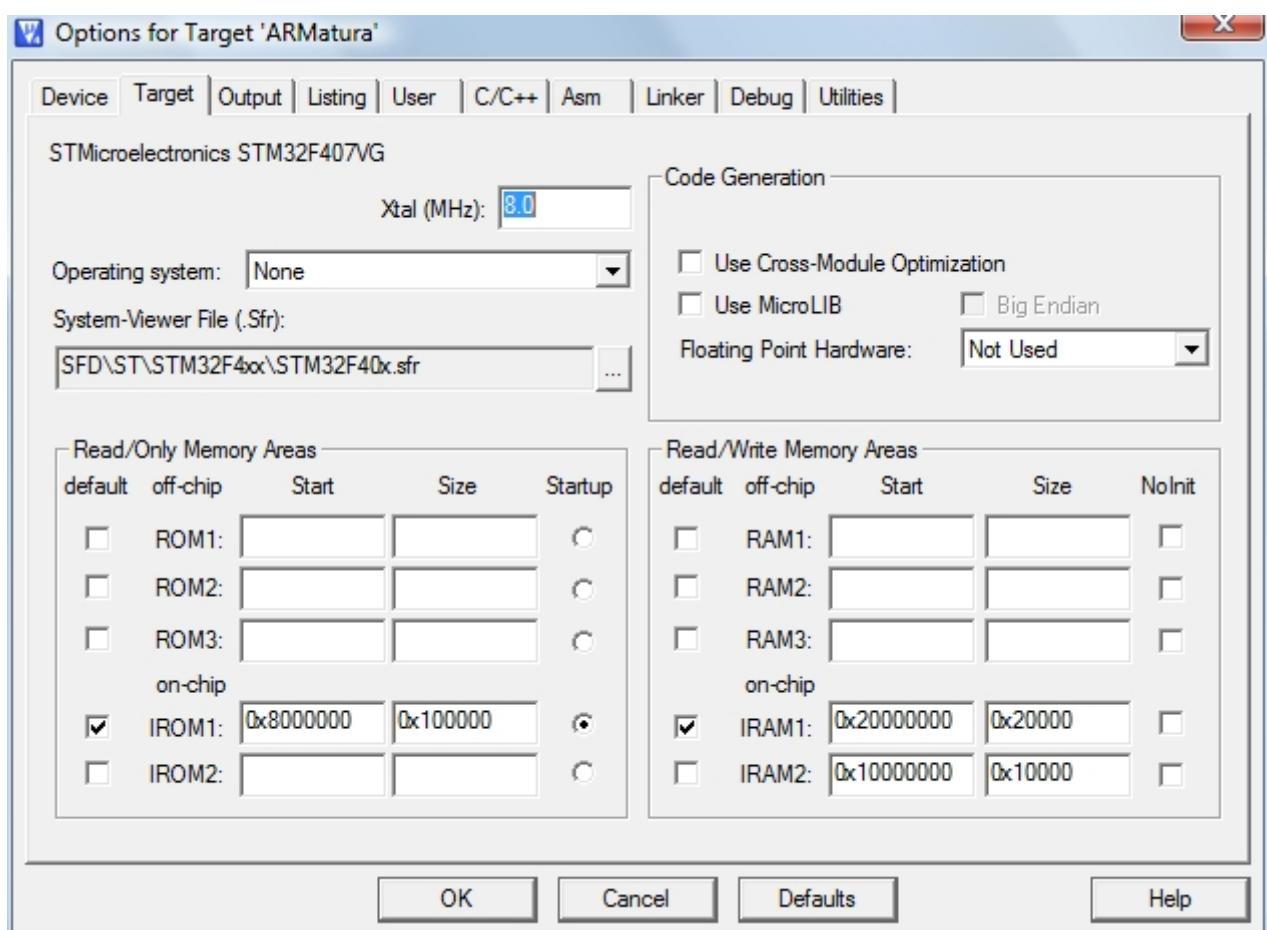


7.1 Настройки проекта в Keil

Настройки проекта вызываются из меню **Project** **Options for Target 'ARMatura' ...**, комбинацией клавиш **Alt** + **F7**, или выбором аналогичной опции из контекстного меню, вызываемого щелчком правой кнопкой мыши на корне дерева проекта **ARMatura** в левом окне **Project**.

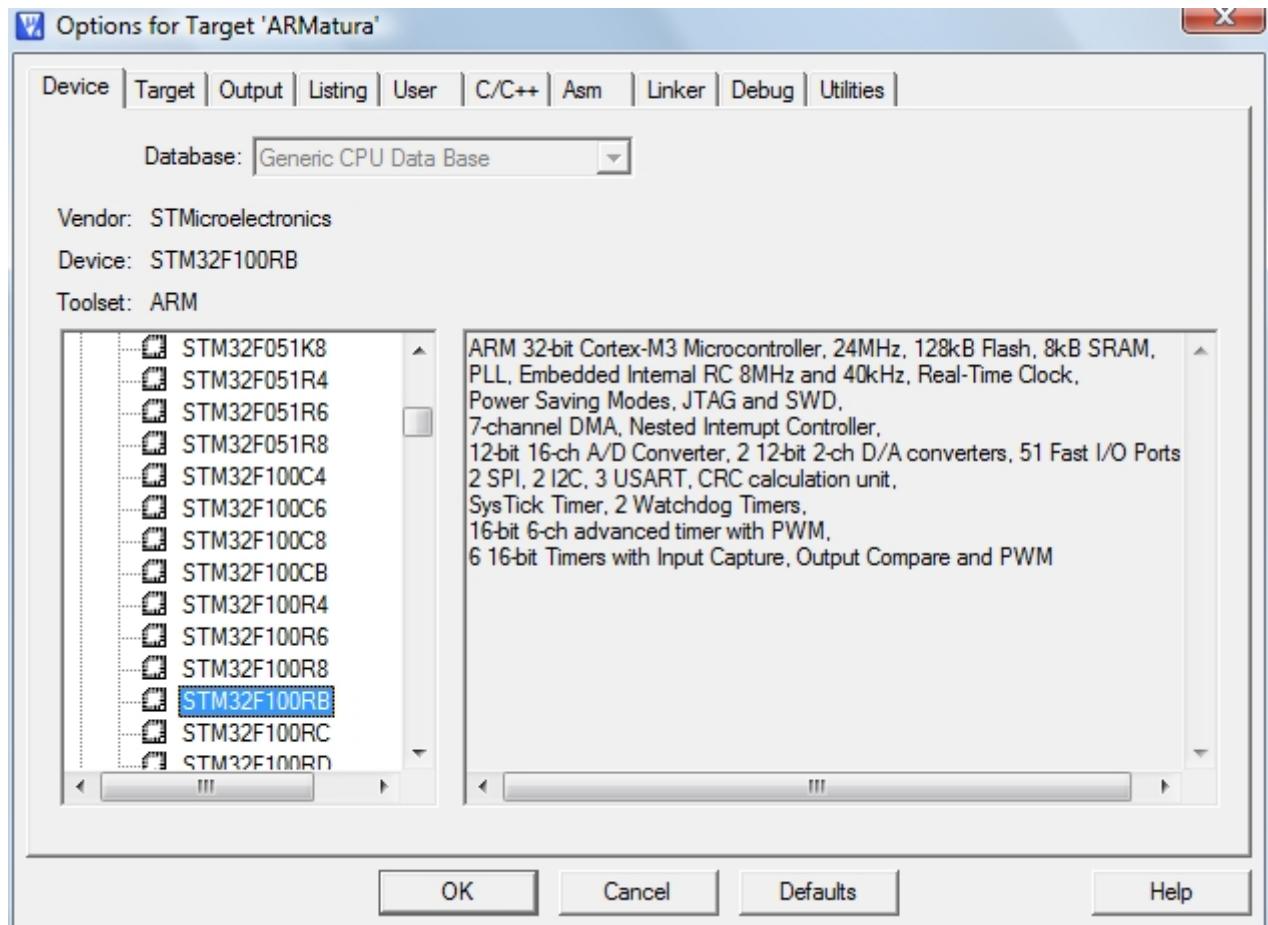


По умолчанию открывается вкладка **Target**:

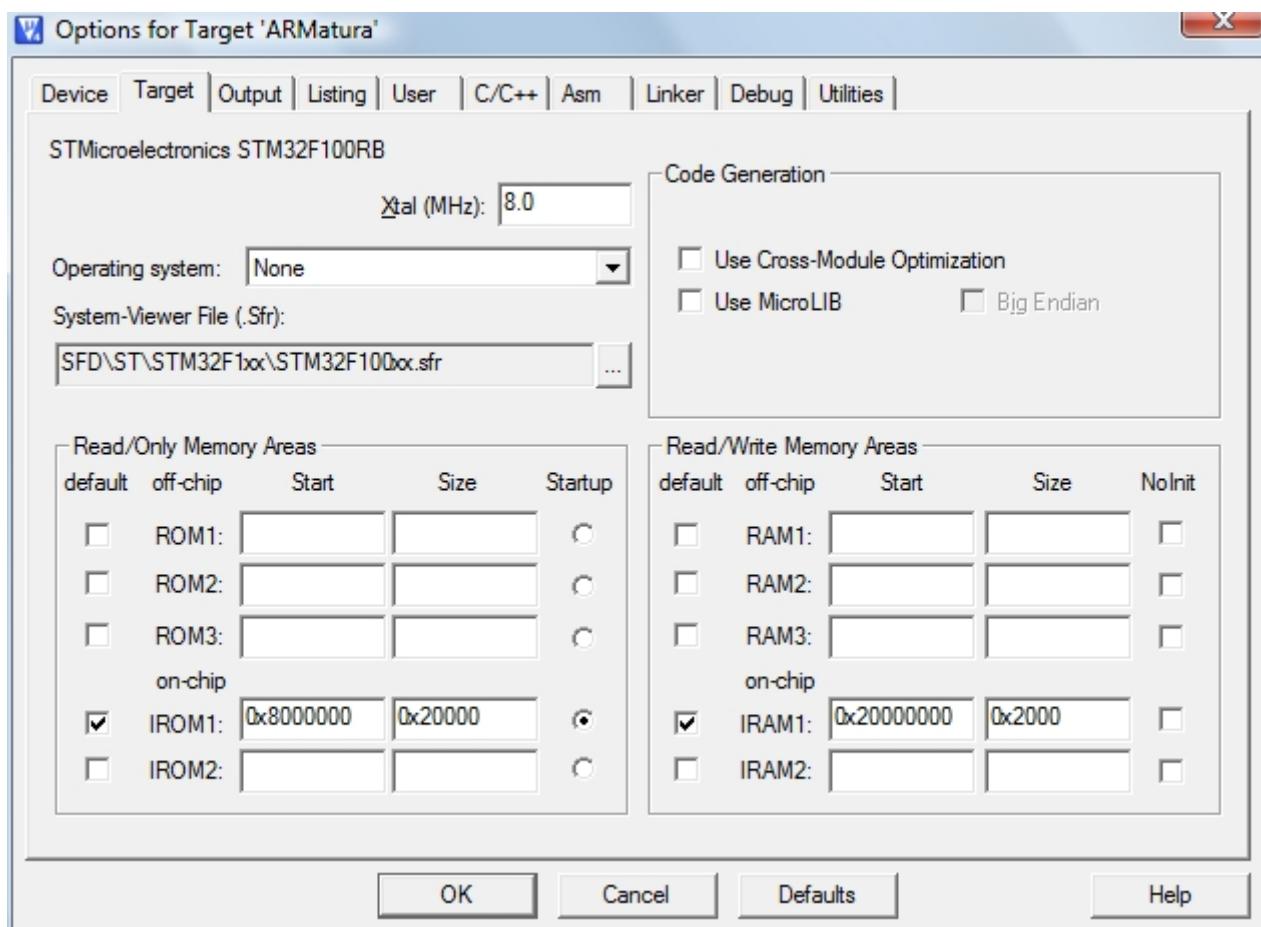


Xtal (MHz)	8.0	текущий процессор частота внешнего кварца на платах Discovery обычно стоит 8
Operating system:	None	или OCPB Keil RTX
Floating Point Hardware:	Not Used	использовать libc от Keil для Cortex-M4 доступен аппаратный FPU
		На этой вкладке также прописывается карта встроенной и внешней памяти Flash/SRAM.

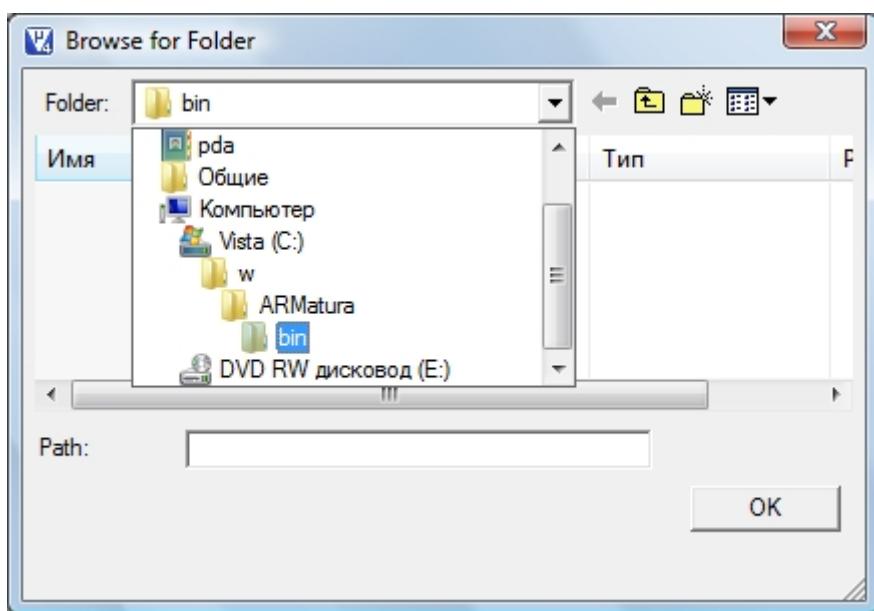
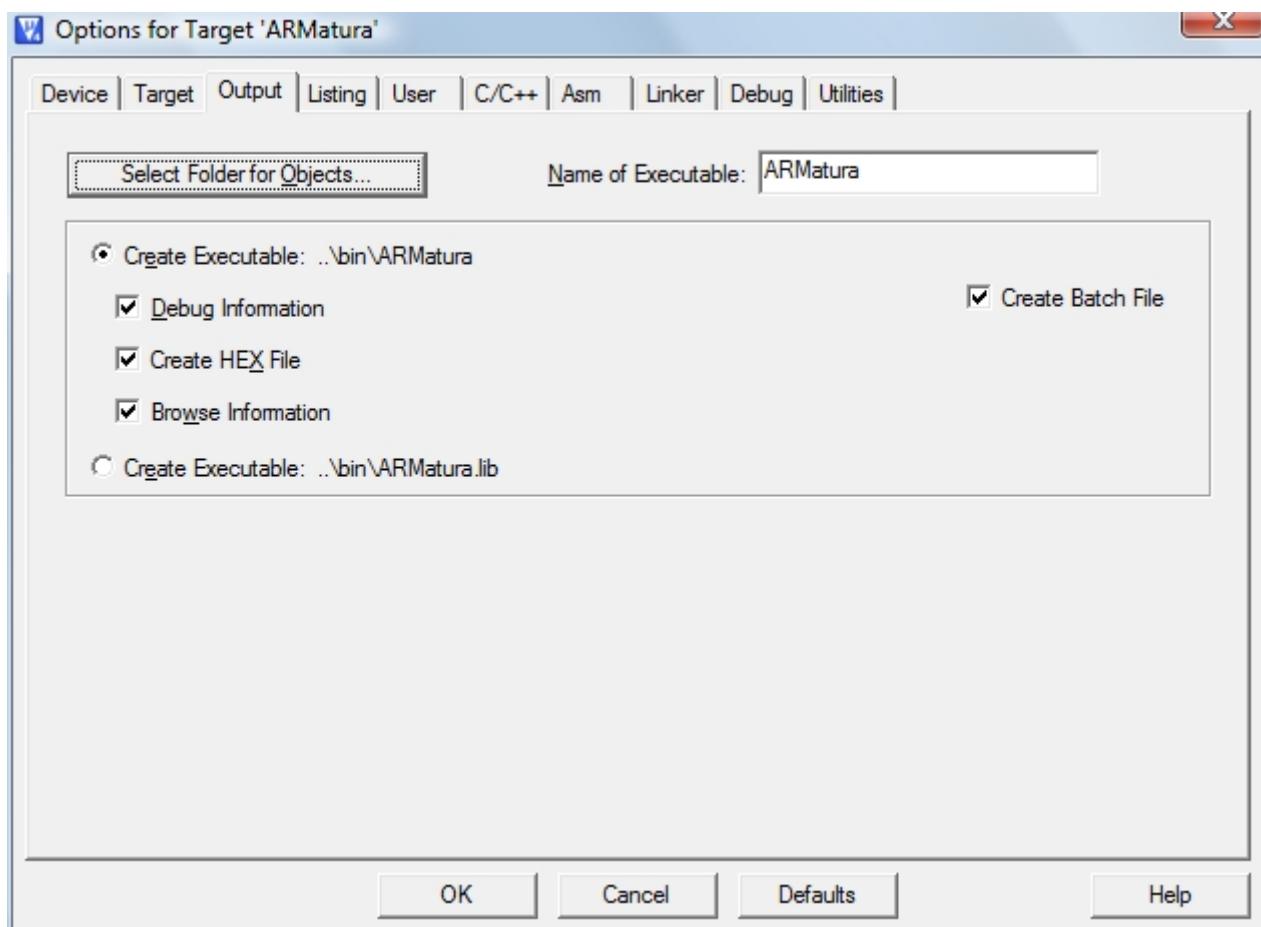
При необходимости собрать проект под другой процессор открываем вкладку **Device**, переконфигурируем проект под другую плату – STM32VLDISCOVERY 1:



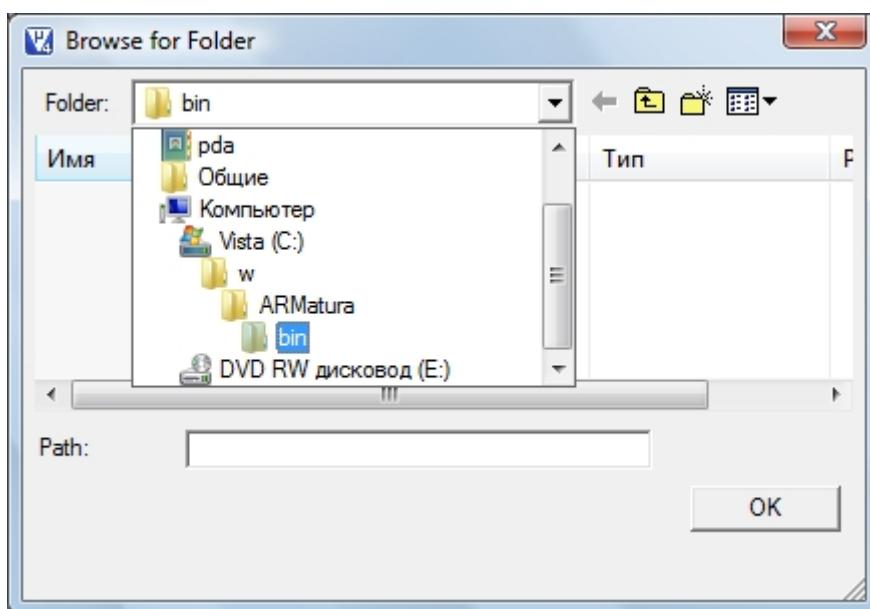
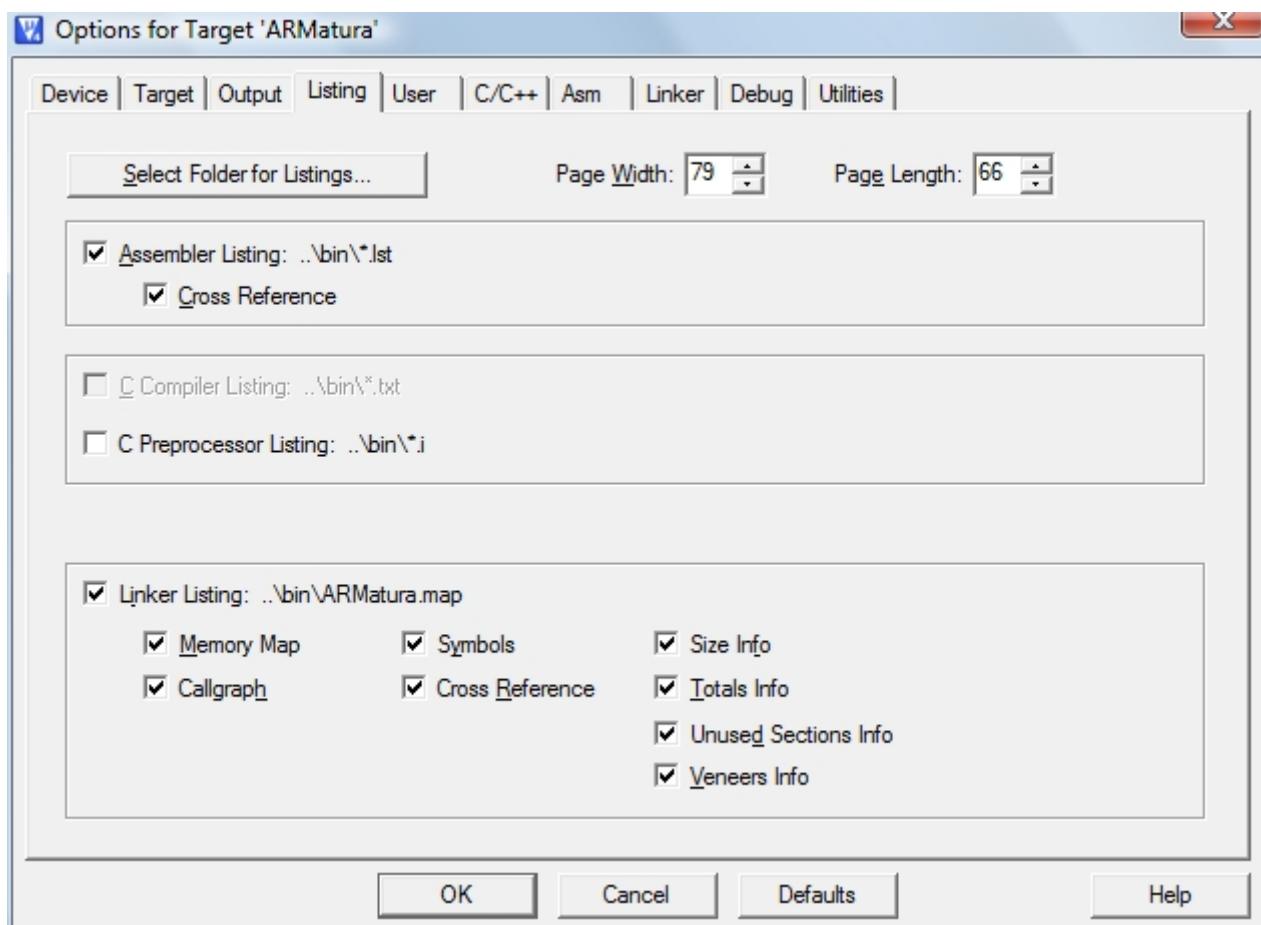
Обратите внимание что на вкладке **Target** изменился чип и настройки памяти:



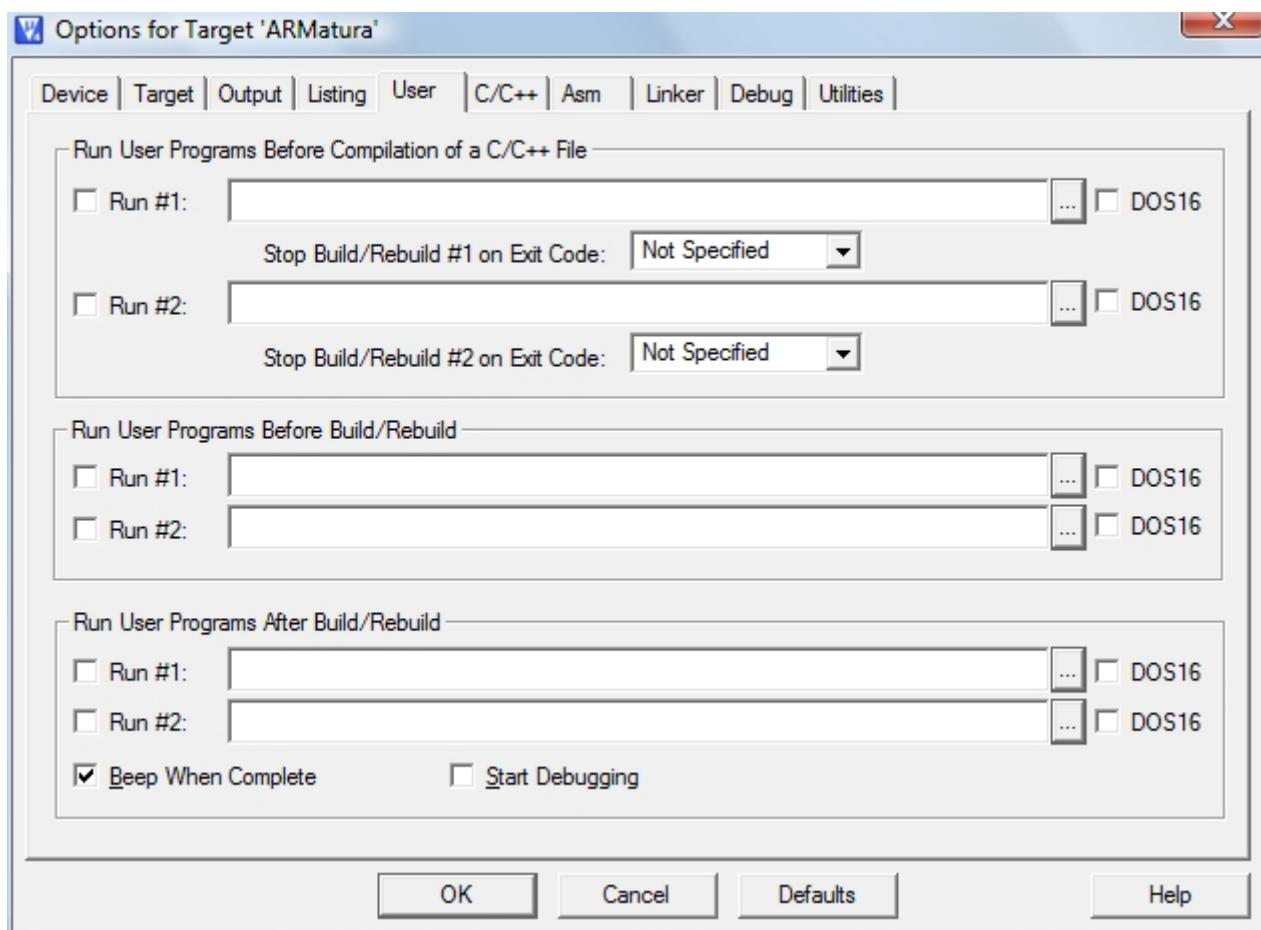
Output Настройки выходных файлов: каталог для бинарных файлов прошивки = firmware, при необходимости можно использовать .hex файл прошивки, имя файла прошивки



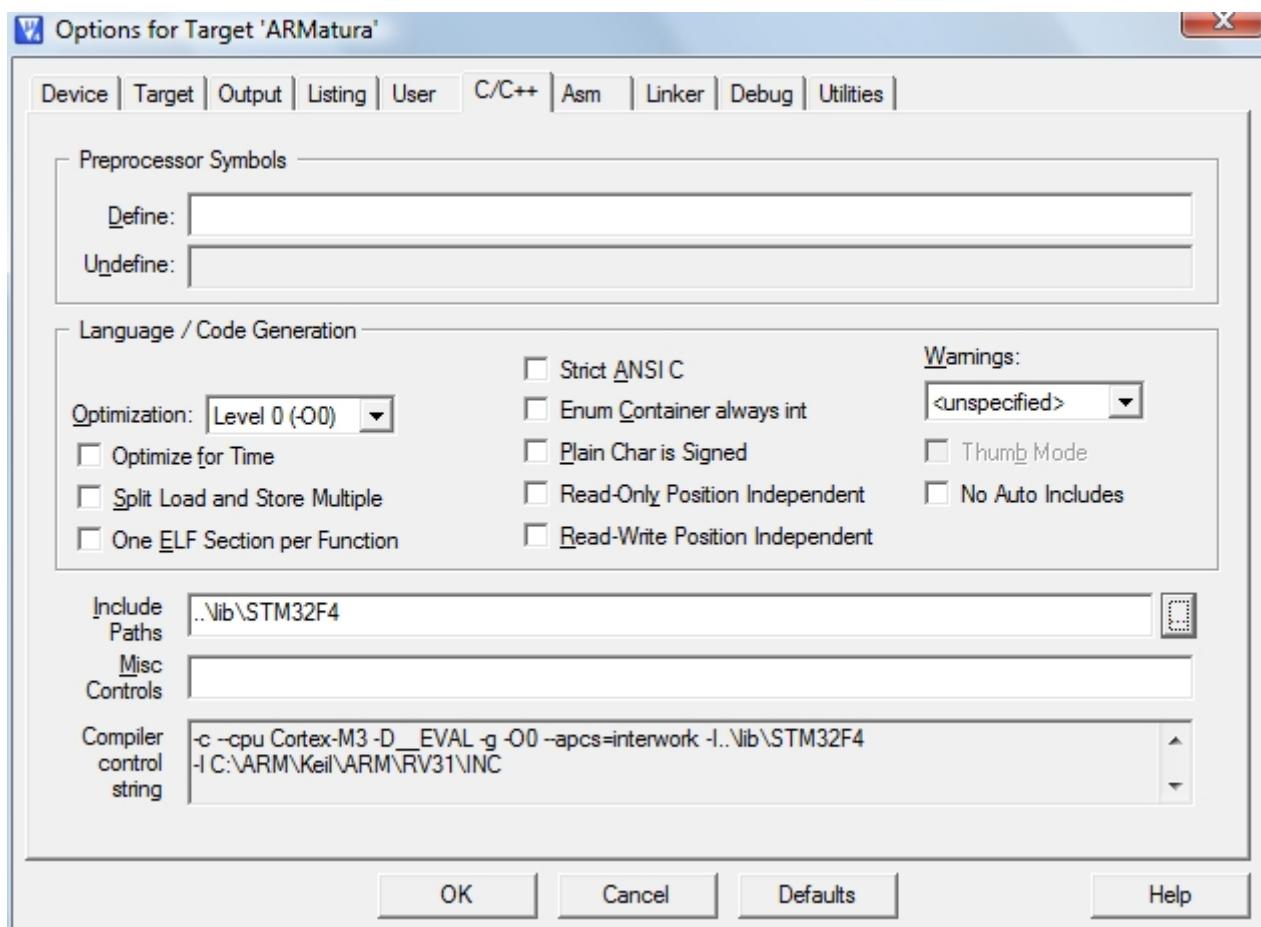
[Listing] Настройки генерации ассемблерных листингов в тот же каталог с прошивкой



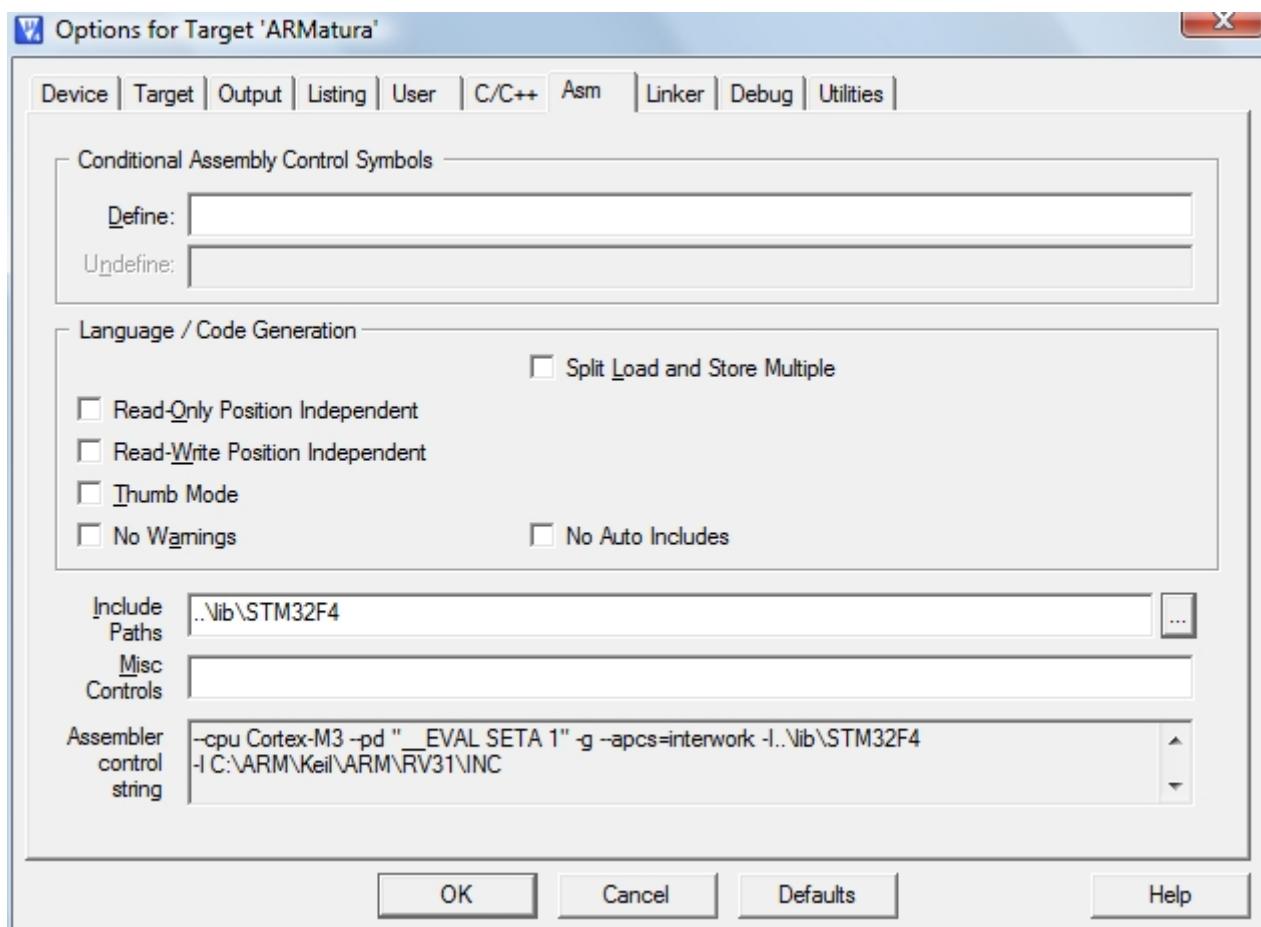
User Запуск пользовательских программ во время сборки проекта: пока не используем



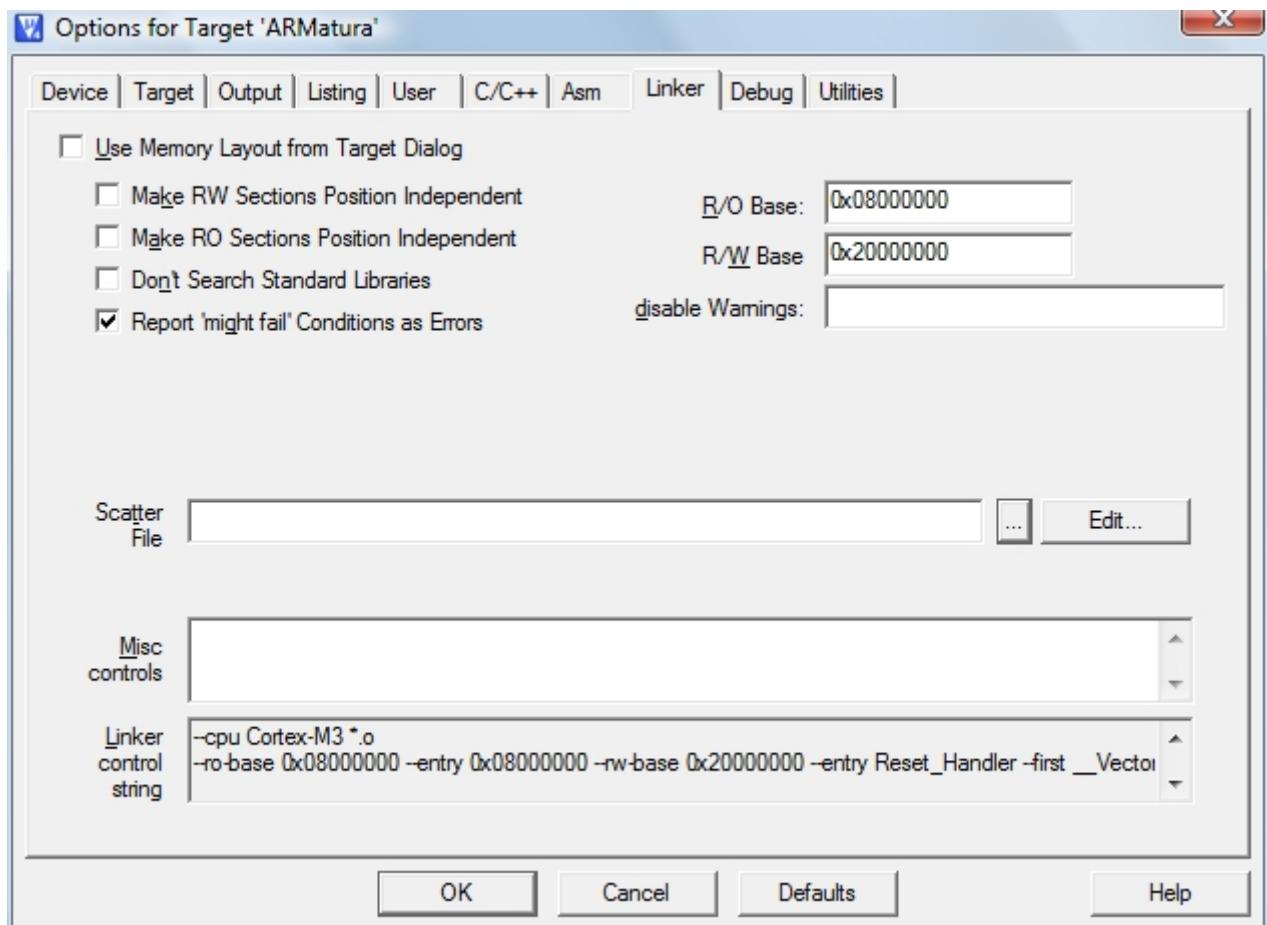
C/C++ Настройки компилятора C^{++} : опции оптимизации, каталоги библиотек и включаемых .h файлов, в нижнем поле прописывается полная командная строка вызова компилятора, которая может вам в дальнейшем понадобится, если потребуется компилировать без использования Keil IDE.



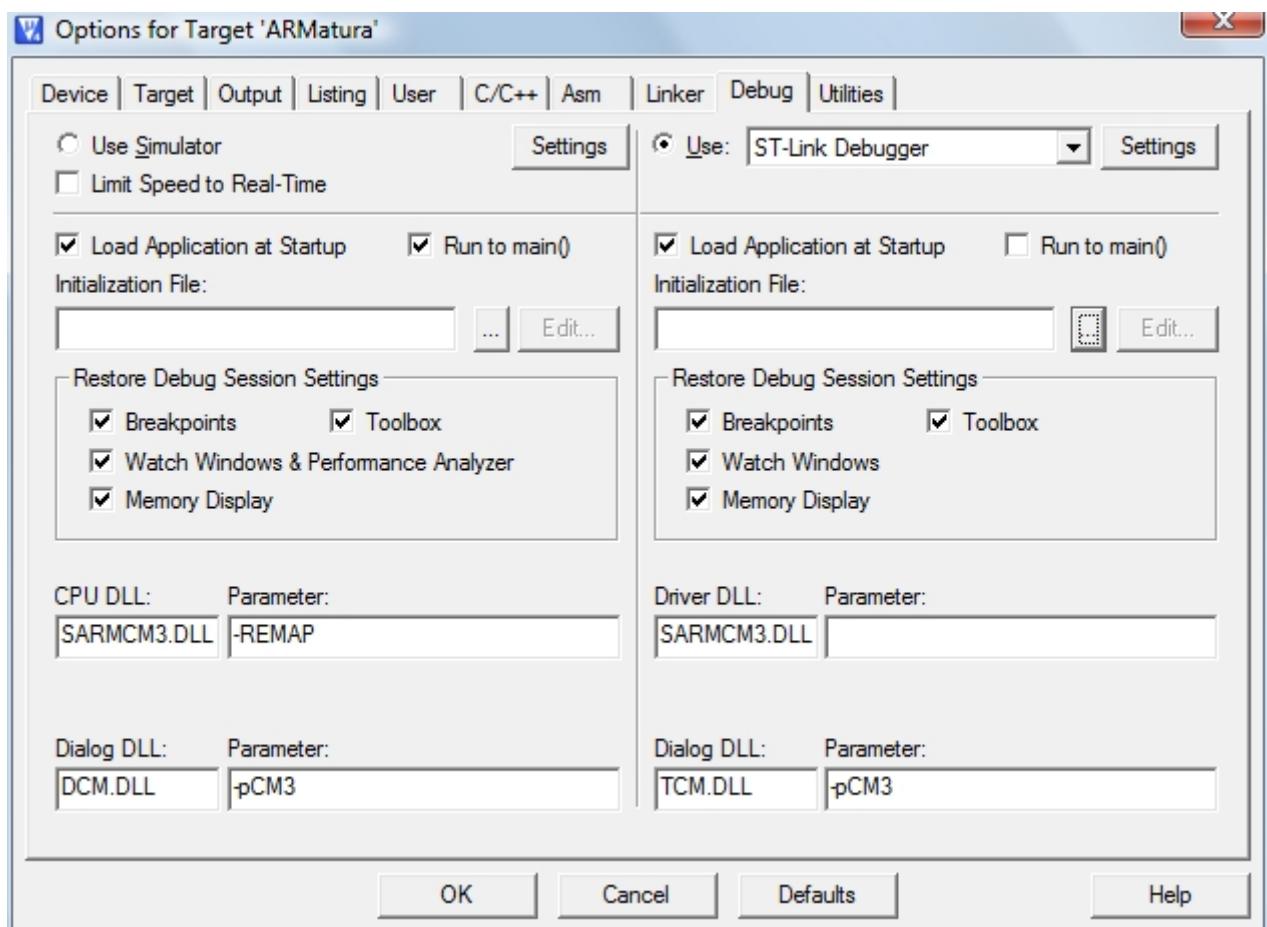
Asm Настройки ассемблера, приблизительно те же что и для C^{++}



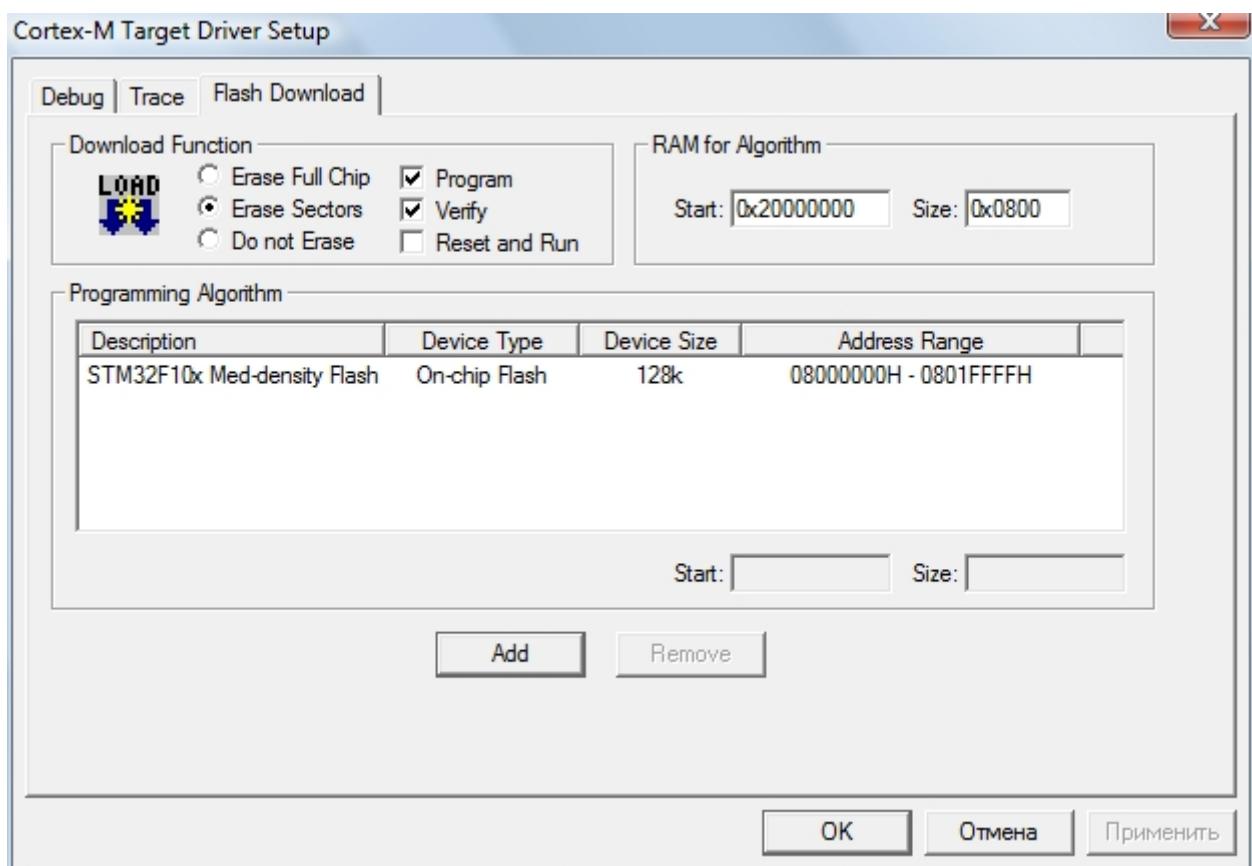
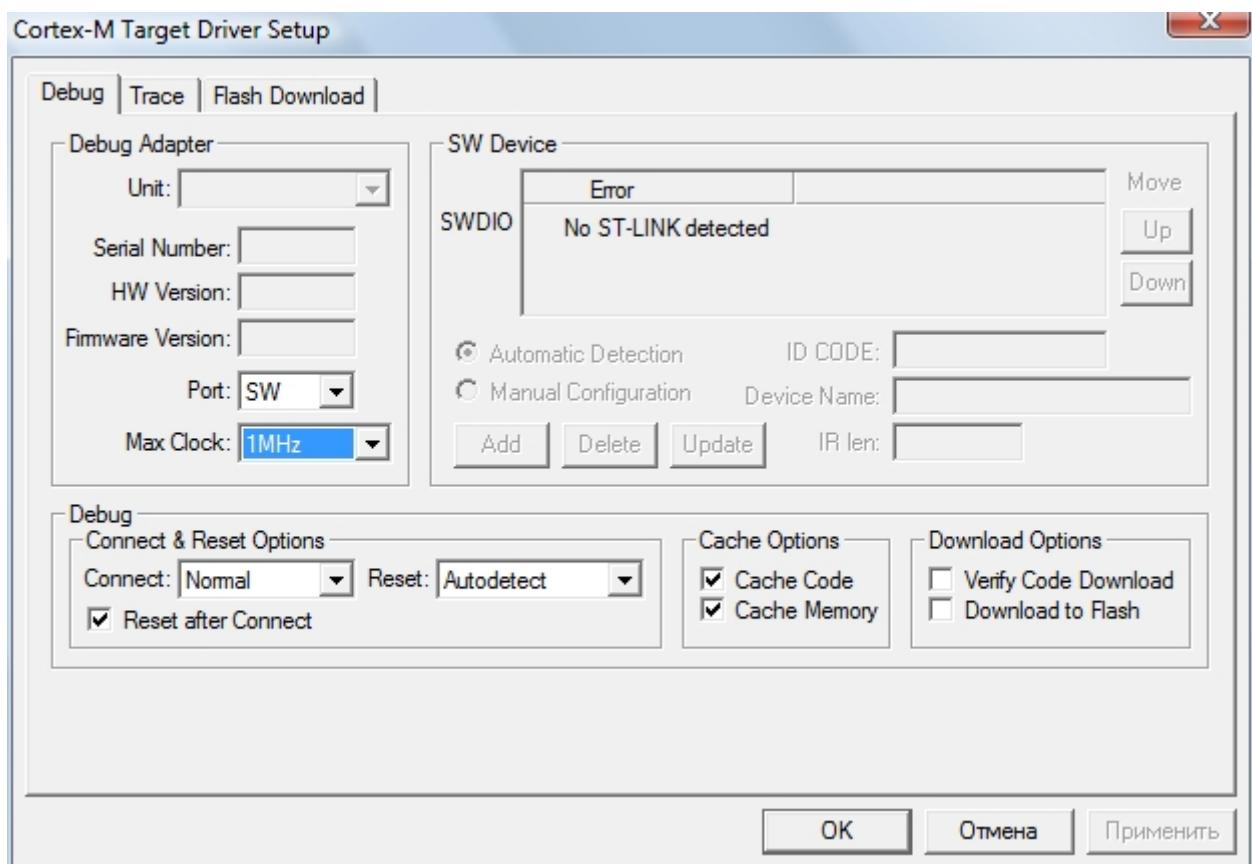
Linker Настройки линкера, который собирает объектные файлы .o, в которые компилируются каждый программный файл (модуль) по отдельности, в один готовый файл прошивки (настройки карты памяти контроллера, адреса точки входа программы и т.п.)



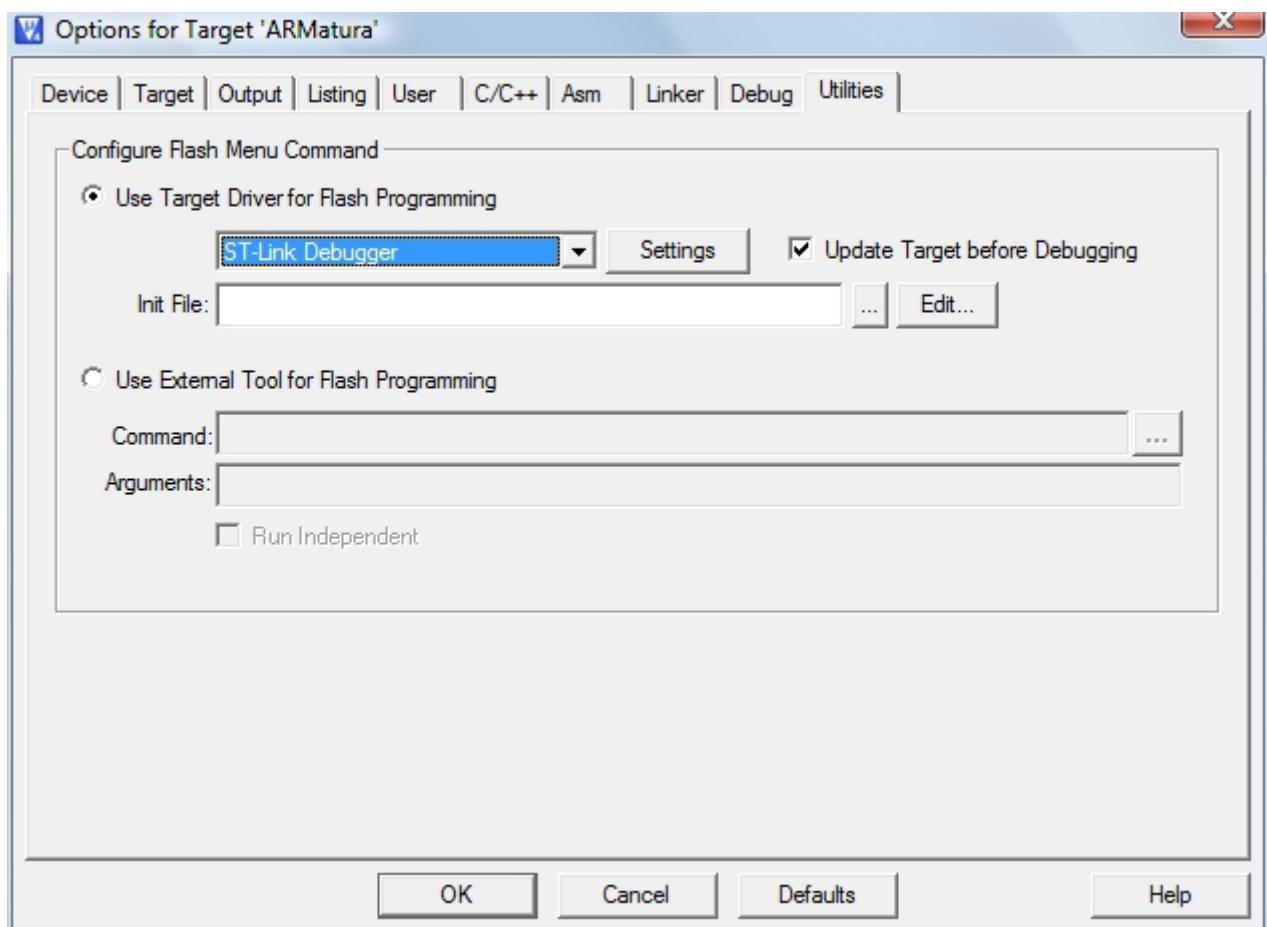
Debug Настройки отладки с помощью адаптера – STlink (внешний или встроенный в оценочную плату), JTAG.



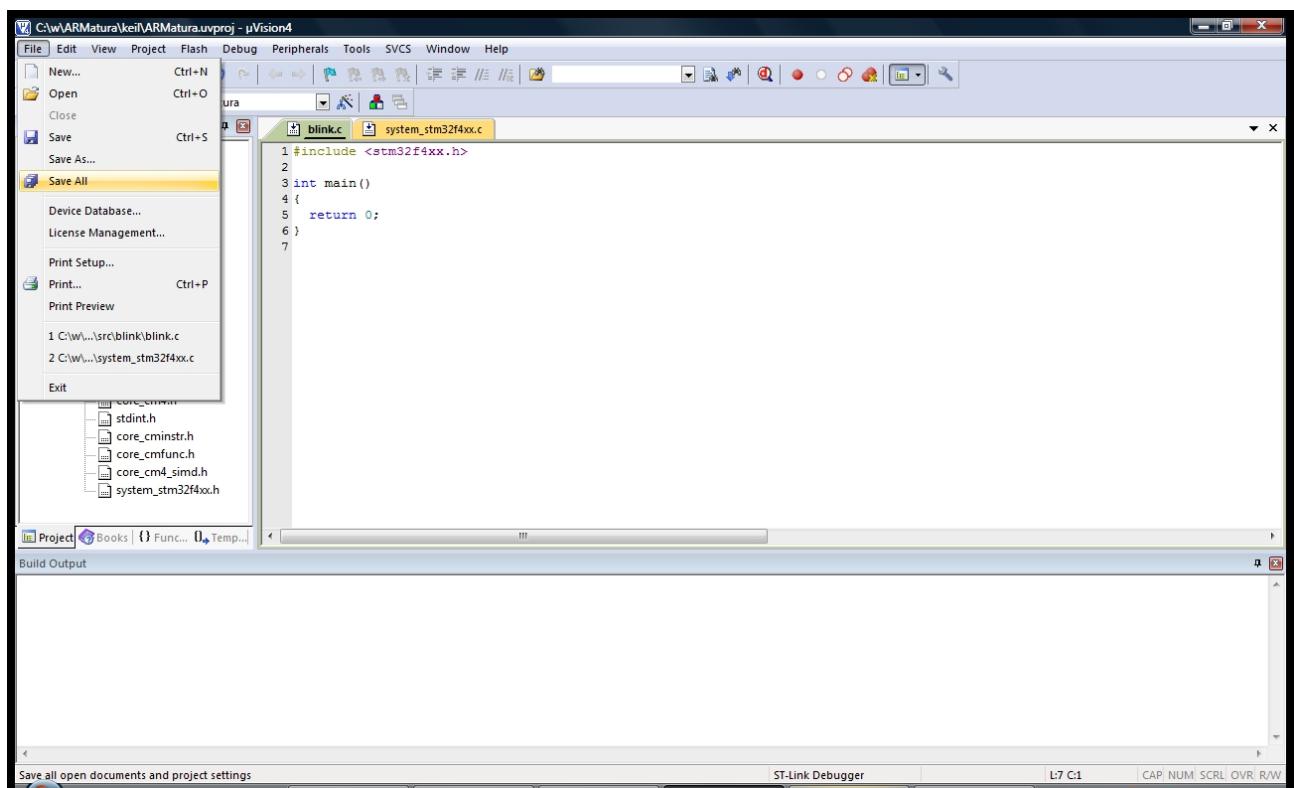
Настройки адаптера STlink – включаем режим SWD для варианта встроенного на оценочную плату, и обнаруживаем что не установили пакет поддержки STlink и драйвера.



Utilities Настраиваем в качестве программатора для прошивки тот же STlink



Сохраняем настроенный проект



7.2 Структура файлов

```
ARMatura
├── startup ..... код инициализации ядра процессора
│   └── keil_startup_stm32f4xx.s ..... ассемблерный код инициализации
└── system_stm32f4xx.c ..... синый код инициализации STM32F4
    ├── system_stm32f4xx.h ..... синый код инициализации STM32F4
    ├── stm32f4xx.h ..... STM32F4 CMSIS Cortex-M4 Device Peripheral
    │   ├── core_cm4.h ..... CMSIS библиотека поддержки ядра Cortex-M4
    │   ├── stdint.h ..... стандартные целые типы C+
    │   ├── core_cmInstr.h ..... CMSIS Core Instruction Interface
    │   ├── core_cmFunc.h ..... CMSIS Core Register Access Functions
    │   └── core_cm4_simd.h ..... CMSIS Cortex-M4 расширение SIMD/DSP
└── blink
    └── blink.c ..... простая программа мигания светодиодом или дрыгания ногой
        └── stm32f4xx.h
```

Глава 8

Hell Of World

Часть IV

Средства разработки

Глава 9

Keil MDK-ARM



Для начала освоения программирования для ARM рекомендуем использовать бесплатный пакет от Keil: <http://www.keil.com/arm/mdk.asp> — ограничения бесплатной версии в 32К кода вполне достаточно для начального освоения программирования под процессоры семейства Cortex-Mx, а затем уже можно переползать на открытое ПО: GNU toolchain 10.1, Eclipse 11.1 и Linux XIII.

Процесс установки и первоначальной настройки описан в 9.

Глава 10

Компиляторы

10.1 GCC

10.2 KeilCC

10.3 IAR

Глава 11

IDE

11.1 Eclipse

11.2 Code::Blocks

11.3 gVim

11.4 Keil uVision

11.5 IAR

Глава 12

Программаторы

12.1 STlink

12.2 Serial Boot

Глава 13

Отладчики

13.1 JTAG

13.2 STM32 SWD

13.3 GDB

STlink gdbserver

OpenOCD

Часть V

Основы языка C^{+^+}

Глава 14

Синтаксис

Глава 15

Типы данных

Глава 16

Стандартная библиотека libc

Часть VI

Отладка

Глава 17

JTAG

Глава 18

GDB

Глава 19

OpenOCD

Часть VII

CMSIS

Глава 20

Startup

Глава 21

Стандартная библиотека STM32

Глава 22

USB client/host

Часть VIII

Ядро Cortex-Mx

Глава 23

Режимы ARM и Thumb

Глава 24

DMA

Глава 25

DSP /Cortex-M3/

Глава 26

FPU /Cortex-M4F/

Часть IX

Интерфейсы

Глава 27

USB

Глава 28

UART

Глава 29

SPI

Глава 30

I2C

Глава 31

CAN

Часть X

Операционные системы ОСРВ

Глава 32

Keil RTX

Глава 33

FreeRTOS

Глава 34

eCos

Глава 35

Linux

подробно рассмотрен в отдельном разделе XIII

Часть XI

Стек TCP/IP

Глава 36

Ethernet

Глава 37

PPP

Часть XII

Типовые применения

Глава 38

GPS

38.1 Протокол NMEA 0183

NMEA 0183— текстовый протокол связи морского и навигационного оборудования.

http://www8.garmin.com/support/pdf/NMEA_0183.pdf
http://www8.garmin.com/support/pdf/NMEA_0183.pdf
http://ru.wikipedia.org/wiki/NMEA_0183
<http://www.robosoft.info/ru/technologies/knowledgebase/nmea0183>

Датум WGS84

http://ru.wikipedia.org/wiki/WGS_84

Большинство GPS приемников отдают данные по NMEA 0183, но вместо режима последовательного порта 4800 8N1 прописанного в протоколе, могут использовать другие режимы, характерные для портов RS232 (9600, 115200). Координаты передаются в датуме WGS84.

Формат сообщения NMEA¹:

\\$[A-Z]5(,<data>)+*[0-9A-F]2<CR><LF>

- символ \$
- 5-буквенный идентификатор сообщения. Первые две буквы — идентификатор источника сообщения, следующие три буквы — идентификатор формата сообщения
 - GPGGA — данные о последнем определении местоположения
 - GPGLL — координаты, широта/долгота
 - GPGSA — DOP (GPS) и активные спутники
 - GPGSV — наблюдаемые спутники
 - GPWPL — параметры заданной точки
 - GPBOD — азимут одной точки относительно другой
 - GPRMB — рекомендуемый минимум навигационных данных для достижения заданной точки

¹регулярные выражения http://en.wikipedia.org/wiki/Regular_expression

- GPRMC — рекомендуемый минимум навигационных данных: информацию о времени, местоположении (WGS84), курсе и скорости, передаваемые навигационным GPS приёмником. Контрольная сумма обязательна для этого сообщения, интервалы передачи не должны превышать 2 секунды.
- GPRTE — маршруты
- HCHDG — данные от компаса
- блоки данных, разделённых запятыми. Пустые данные, не заданные в середине строки, оставляют запятыю. Пустые данные в конце строки могут отбрасываться вместе с запятой.
- символ *
- двузначное hex число — контрольная XOR-сумма всех байт в строке между \$ и *.
- конец строки <CR><LF> = 0x0D 0x0A = \r\n

38.2 \$GPRMC — рекомендуемый минимум навигационных данных

\$GPRMC, hhmmss.ss, [AV], GGMM.MM, [NS], gggmm.mm, [EW], v.v, b.b, ddmmyy, x.x, n, m*hh<CR><LF>

- GP — приём сигналов GPS, GN — ГЛОНАСС
- RMC — Recommended Minimum sentence C
- hhmmss.ss — время фиксации местоположения по времени UTC
- A — данные достоверны, V — данные недостоверны
- GGMM.MM — широта, 2 цифры градусов, 2 цифры целых минут, точка и дробная часть минут;
- N/S — северная/южная широта
- gggmm.mm — долгота, 3 цифры градусов, 2 цифры целых минут, точка и дробная часть минут;
- E/W — восточная/западная долгота
- v.v — горизонтальная составляющая скорости в узлах
- b.b — путевой угол (направление скорости) в градусах: 0° север, 90° восток, 180° юг, 270° запад
- ddmmyy — дата
- x.x — магнитное склонение в градусах (часто отсутствует)
- n — направление магнитного склонения: для получения магнитного курса магнитное склонение необходимо вычесть (E) или прибавить (W) к истинному курсу
- m — индикатор режима: A — автономный, D — дифференциальный, E — аппроксимация, N — недостоверные данные (часто отсутствует, данное поле включая запятую отсутствует в старых версиях NMEA)

38.3 Tistar15



Цена: 250 руб. <http://www.voltmaster.ru/cgi-bin/qquery.pl?id=127000056703&group=700000>

38.4 WISMO228



Цена: 1070 руб. <http://www.voltmaster.ru/cgi-bin/qquery.pl?id=127000881980&group=700000>

Глава 39

GSM

39.1 WISMO228



Цена: 1070 руб. <http://www.voltmaster.ru/cgi-bin/qwery.pl?id=127000881980&group=700000>

Глава 40

шина Dallas 1Wire

40.1 RTC

40.2 Датчики температуры DS18x20

Часть XIII

Встраиваемый Linux

Часть XIV

QA

Где перевод пунктов меню Если вы собираетесь заниматься разработкой embedded ПО, знание английского на уровне пользования online словарями и Google Translate обязательно. Кому это не нравится, могут и дальше ждать перевод datasheetов и programmer's manualов на снимаемые с производства компоненты — перевод как раз появляются к этому времени.

Часть XV

Приложения

Глава 41

Сводная таблица процессоров

	ядро Cortex-	MHz	Flash	SRAM	корпус LQFP	USB	UART	SPI	CAN
STM32F100C4T6B	M3	24	16K	4K	48		2	1	
STM32F100RBT	M3	24	128K	8K	100		1		
STM32F103RBT	M3				100	1	1		
STM32F407VGT	M4F	168	1M	192K	144	2	6		2
STM32F407IGT	M4F	168	1M	192K	176	2	8		2
STM32F427IIT	M4F	168	2M	256K	176	2	8		2

41.1 STM32F10x

STM32F100C4T6B

Ядро	Cortex-M3
Flash	16K
SRAM	4K
16-битные таймеры	6
таймеры ШИМ	3
RTC	да
UART	2
SPI	1
I2C	1
DMA	1 канал
АЦП	10x12 бит
ЦАП	2x12 бит
корпус	LQFP48