

Контроллеры ARMatura

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

2 марта 2013 г.

Оглавление

Оглавление	1
I Введение	6
II Железо	8
1 STM32VLDISCOVERY /STM32F100RBT6/	9
2 STM32F4DISCOVERY /SRM32F407VGT6/	11
3 ARMatura /STM32F407IGT/	13
4 PION /STM32F103C8/	14
III Первые шаги	15
5 Установка ПО	16
6 Первый проект: blink	17
6.1 Структура файлов	19
6.2 blink.c	20
7 Hell Of World	21
IV Средства разработки	22
8 IDE	23

8.1 Keil	23
8.2 Настройки среды	29
8.3 Настройки проекта	30
8.4 Eclipse	43
8.5 Code::Blocks	44
8.6 gVim	44
8.7 IAR	44
9 Компиляторы	45
9.1 GCC	45
9.2 KeilCC	45
9.3 IAR	45
10 Адаптеры	46
10.1 STlink	46
10.2 JTAG	53
11 Отладка	54
11.1 STM32 SWD	54
11.2 GDB	54
11.3 JTAG	54
V Основы языка C⁺	55
12 Синтаксис	56
13 Типы данных	57
14 Стандартная библиотека libc	58
VICMSIS	59
15 Startup: файлы стартового кода	61
16 Стандартная библиотека STM32	75
16.1 stm32f4xx.h	75
17 USB client/host	76
VII Ядро Cortex-Mx	77
18 Режимы ARM и Thumb	78
19 DMA	79
20 DSP /Cortex-M3/	80
21 FPU /Cortex-M4F/	81

VIIИнтерфейсы	82
22 USB	83
23 UART	84
24 SPI	85
25 I2C	86
26 CAN	87
IXОперационные системы ОСРВ	88
27 Keil RTX	89
28 FreeRTOS	90
29 eCos	91
30 Linux	92
X Стек TCP/IP	93
31 Ethernet	94
32 PPP	95
XIТиповые применения	96
33 GPS	97
33.1 Протокол NMEA 0183	97
33.2 \$GPRMC — рекомендуемый минимум навигационных данных	98
33.3 Системы координат (датум)	99
33.4 Методы преобразования систем координат	100
33.5 Геоид и эллипсоиды	101
33.6 Tistar15	102
33.7 WISMO228	102
34 GSM	103
34.1 WISMO228	103
35 шина Dallas 1Wire	104
35.1 RTC	104
35.2 Датчики температуры DS18x20	104

XI Встраиваемый Linux	105
XII ПДА	106
XIII Приложения	108
36 Сводная таблица процессоров	109

Listings

6.1	<code>blink.c</code>	20
15.1	<code>keil_startup_stm32f4xx.s</code>	62

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,.. [36](#)

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 [36](#)

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:

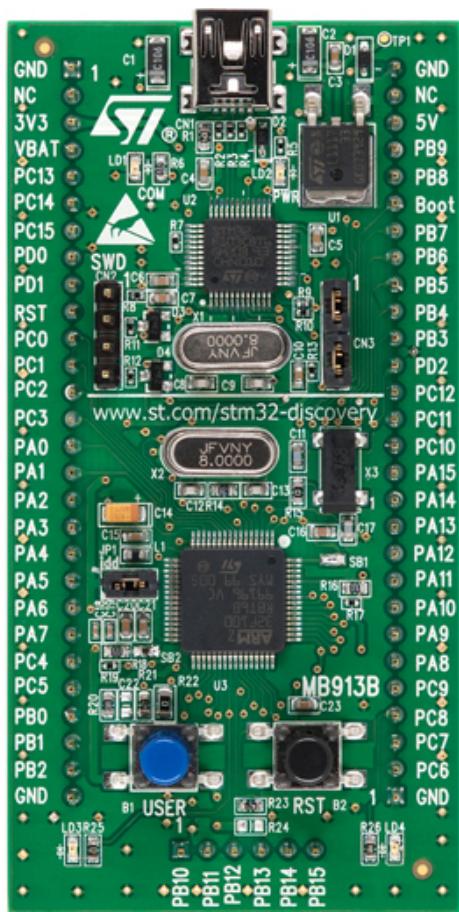
- ИКП СГАУ <semkin@ssau.ru>
- Дмитрий Понятов <dponyatov@gmail.com>

Часть II

Железо

Глава 1

STM32VLDISCOVERY /STM32F100RBT6/



- Микроконтроллер STM32F100RBT6 **36**, 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.

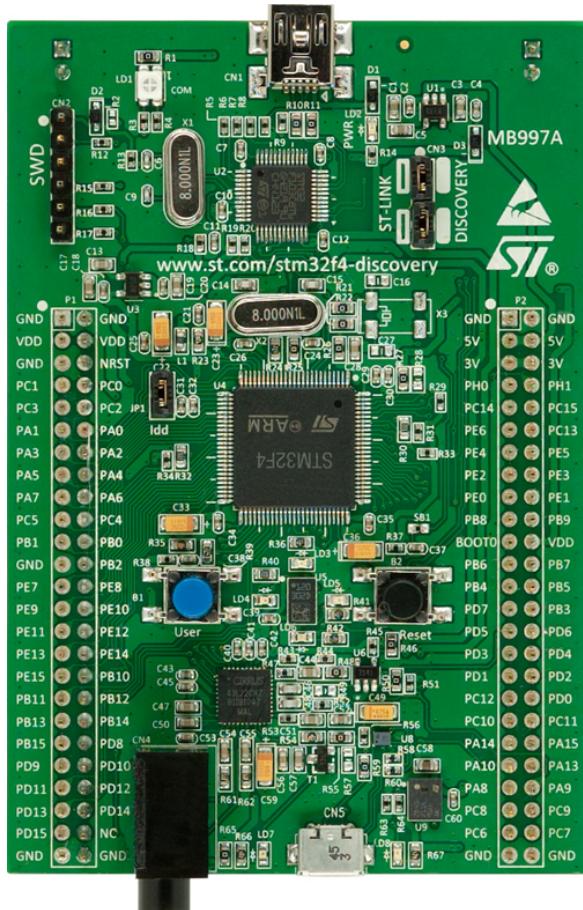
<http://www.voltmaster.ru/cgi-bin/qwery.pl?id=127000573571&group=7000046>

<http://we.easyelectronics.ru/STM32/stm32vldiscovery-the-grabli-ili-dlya-samyh-nach.html>

- Контроллер STM32F100RBT6 (128KB Flash, 8KB RAM, up to 24 MHz)
- Кварц 8MHz, установлен на цангах, так что можно ставить свой
- Кварц 32.768KHz **X3**
- Встроенный программатор-отладчик STlink^{10.1}, только с выходом SWD, можно его использовать и для прошивки своих девайсов. Что характерно, отладчик выполнен на более серьёзном STM32F103CBT6³⁶.
- Две кнопки (правда одна из них — RESET).
- Два светодиода **PC8,PC9**

Глава 2

STM32F4DISCOVERY /SRM32F407VGT6/



- микроконтроллер STM32F407VGT6 **36** 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 /STM32F407IGT/

Глава 4

PION /STM32F103C8/

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	STM32F103C8	??
тактовая	72Mhz	
Flash	64K	
SRAM	20K	
шина	ARMAbus	
интерфейсы	UART	3
	USB	1
	CAN	1
	SPI	2
	АЦП	10x12b
	ЦАП	2x12b
буфер	Parallel Flash	64K

Выбор процессора определялся ценой, наличием CAN интерфейса (требование для шины ARMAbus), USB (возможность использования в виде изолированного контроллера для ПК), объемом SRAM для хранения рабочих данных, и количеством ног ввода/вывода.

Часть III

Первые шаги

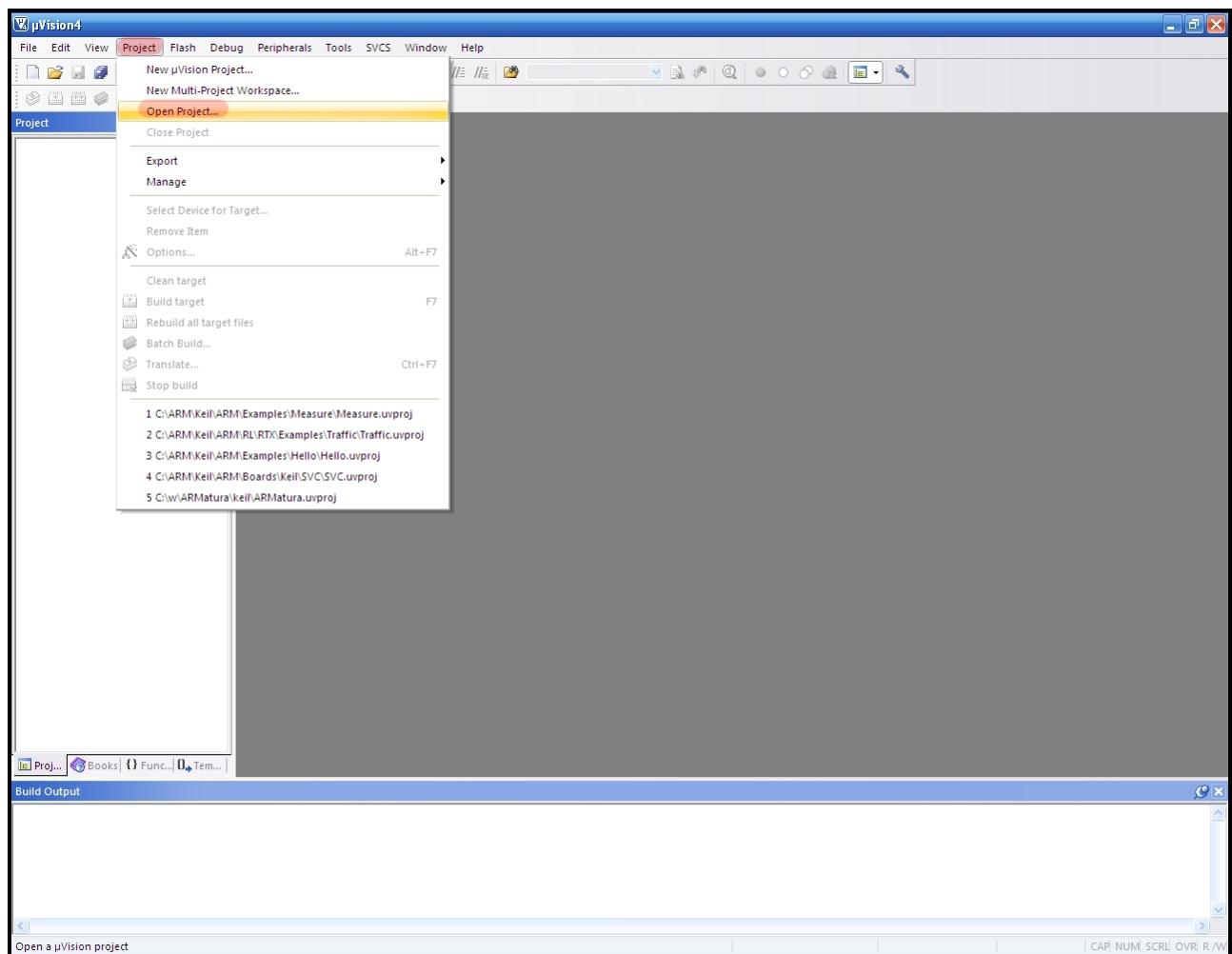
Глава 5

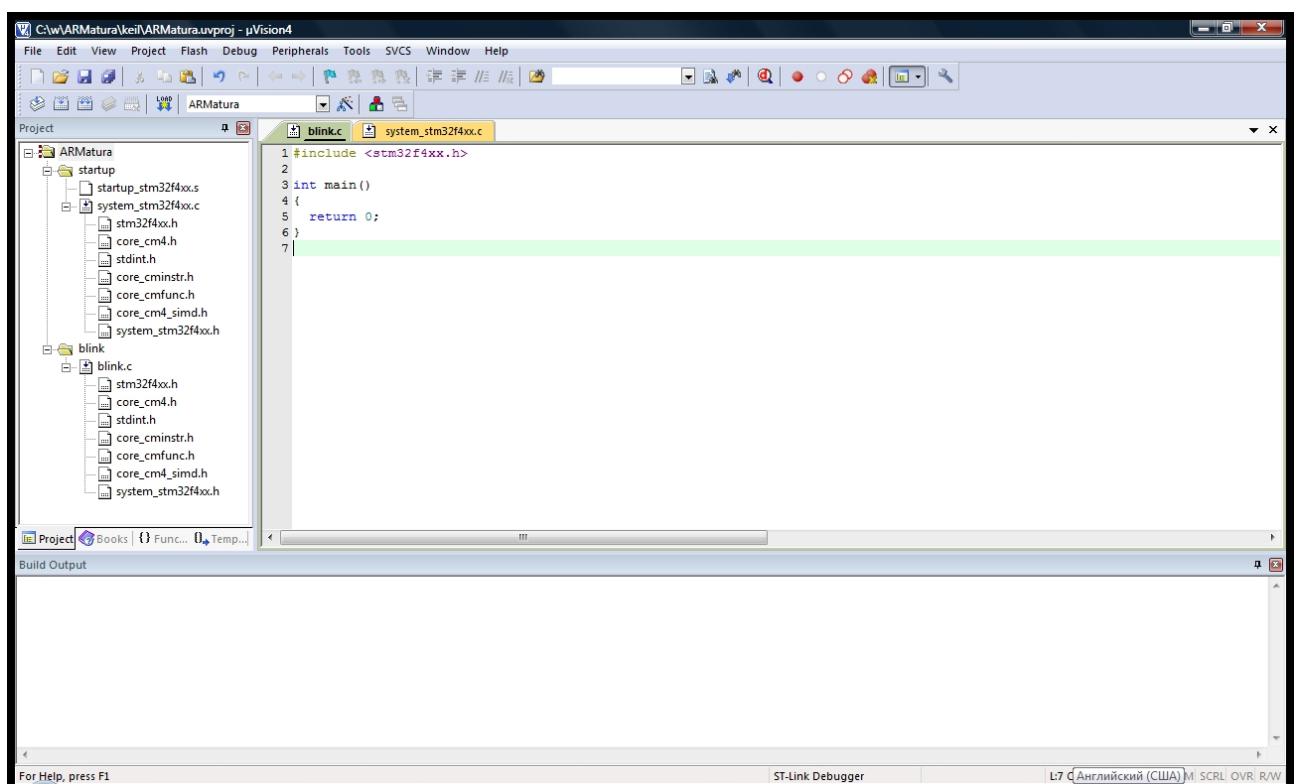
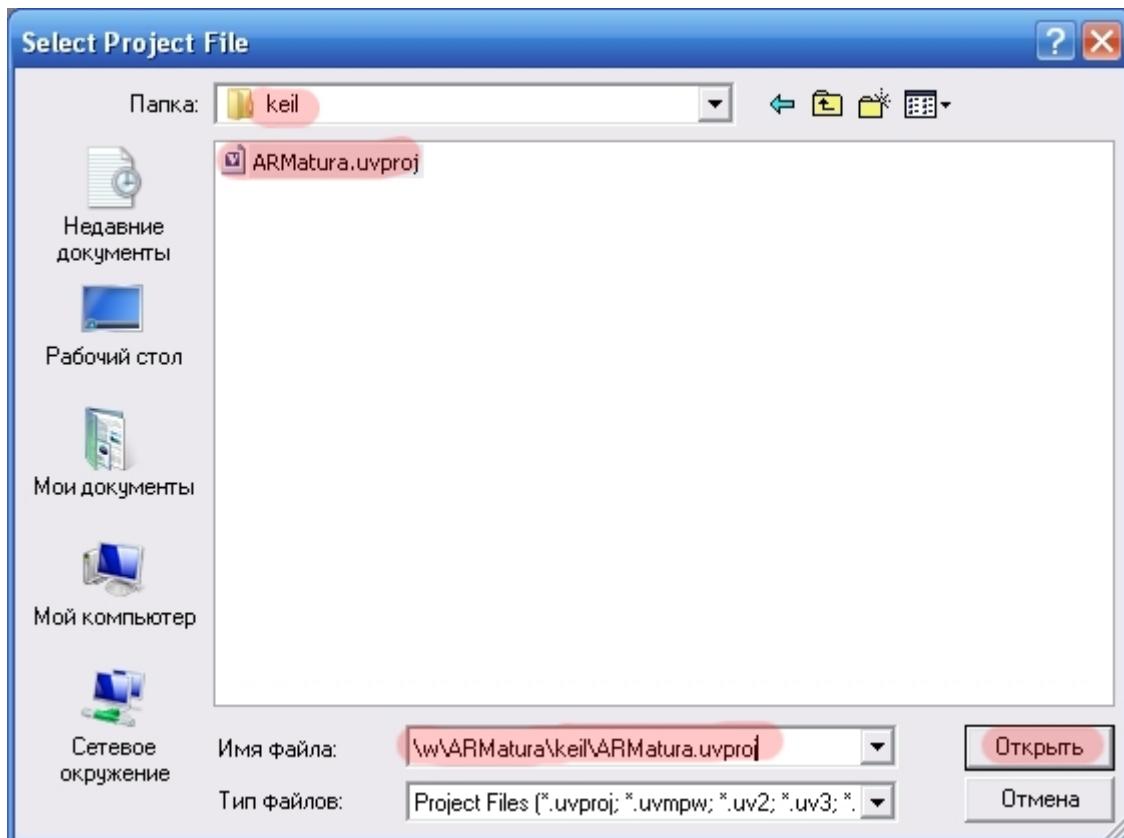
Установка ПО

1. установка пакета ПО STlink [10.1](#)
2. установка Keil MDK-ARM [8.1](#)

Глава 6

Первый проект: blink





При необходимости можно изменить настройки проекта см. 8.3. Перенастройка может понадобиться если вы пытаетесь работать не с STM32F4DISCOVERY², а с другой платой или самопальной железкой. Также перенастройка может понадобится если почему-то не срабатывает подключение к адаптеру STlink^{10.1}.

Собираем проект, нажав клавишу **F7**:

6.1 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 CMSISVI Cortex-M4 Device Peripheral
                    └── core_cm4.h ..... CMSISVI библиотека поддержки ядра Cortex-M4
                    └── stdint.h ..... стандартные целые типы C+
                    └── core_cmInstr.h ..... CMSISVI Core Instruction Interface
                    └── core_cmFunc.h ..... CMSISVI Core Register Access Functions
                    └── core_cm4_simd.h ..... CMSISVI Cortex-M4 расширение SIMD/DSP
└── blink
    └── blink.c ..... простая программа мигания светодиодом или дрыгания ногой
        └── stm32f4xx.h

```

В отличие от кристаллов типа Atmel AVR, при программировании для архитектуры Cortex-Mx принято использовать готовую библиотеку CMSIS^{VI}, для которой проводится активная стандартизация среди производителей процессоров.

Cortex > Microcontroller > Software > Interface > Standard

<http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php>

Кроме основной части CMSIS^{VI}, общей для всех производителей МК Cortex-Mx, каждый подставщик предоставляет аппаратно-зависимую часть библиотеки:

- стартовый код инициализации ядра процессора
- библиотеку периферии например STM32F Standard Peripheral Library, включающую код обеспечивающий доступ к устройствам, встроенным в кристалл в т.ч.
- USB OTG Host & Device
- DSP /SIMD Extension (для Cortex-M4F) системо-зависимую часть расширения CMSIS^{VI} DSP

Библиотеки доступны для загрузки тут:

STSW-STM32065 STM32F4 DSP and standard peripherals library, including 82 examples for 26 different peripherals and template project for 5 different IDEs

http://www.st.com/st-web-ui/static/active/en/st_prod_software_internet/resource/technical/software/firmware/stm32f4_dsp_stdperiph_lib.zip

6.2 blink.c

Сначала посмотрим код основной программы:

Listing 6.1: /src/blink/blink.c

```
/*
 * ARMatura book sample
 * https://github.com/ponyatov/ARMatura
 */

#include <stm32f4xx.h> // загрузить CMSIS-библиотеку STM32Fx

int main()
{
    return 0;
// проверка отображения строк в LaTeX: char X []="Hello World !";
}
```

Как видим — в ней нет абсолютно ничего сложного: только подключение библиотеки нашего МК и единственная функция `int main(void)`, содержащая только возврат в стартовый код CMSISVI значения по умолчанию — 0.

Файл `stm32f4xx.h` подробно рассмотрен в ?? — на текущем слое знаний не стоит в него углубляться: он слишком большой и содержит практически весь список функционала, обеспечиваемый CMSISVI.

То же самое можно сказать и о других файлах — все они относятся либо к библиотеке CMSISVI, либо к файлам поддержки производителя чипов.

Подробности будут рассмотрены далее в разделе VI, а пока важнее получить практический результат.

Глава 7

Hell Of World

Часть IV

Средства разработки

Глава 8

IDE

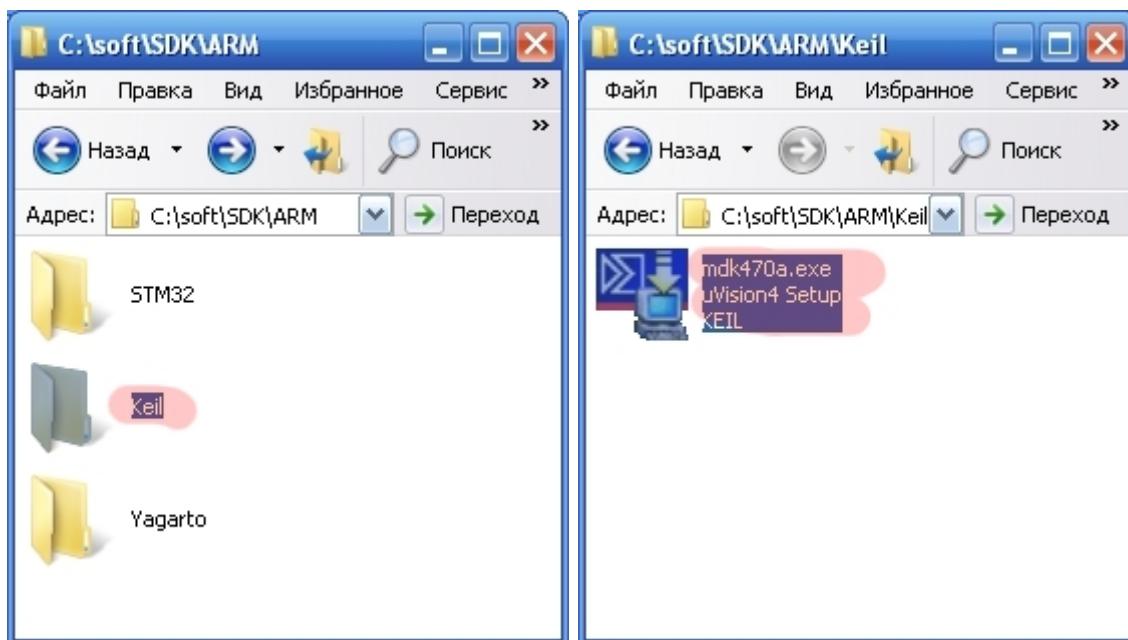
8.1 Keil

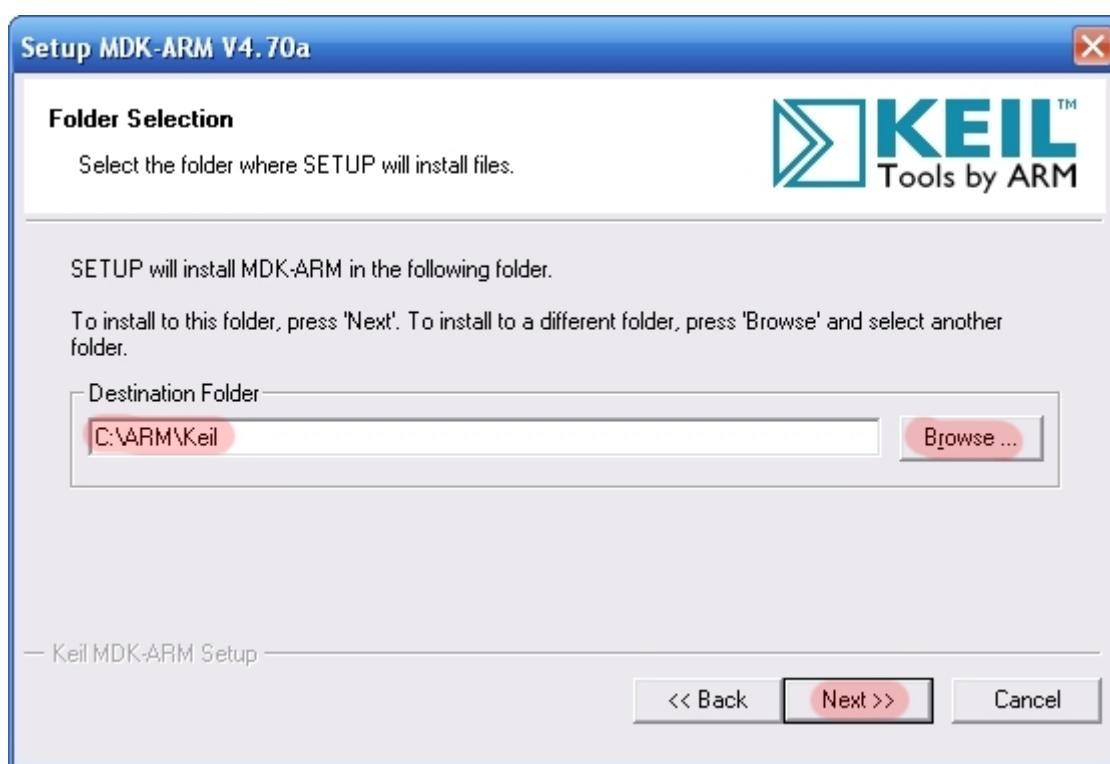
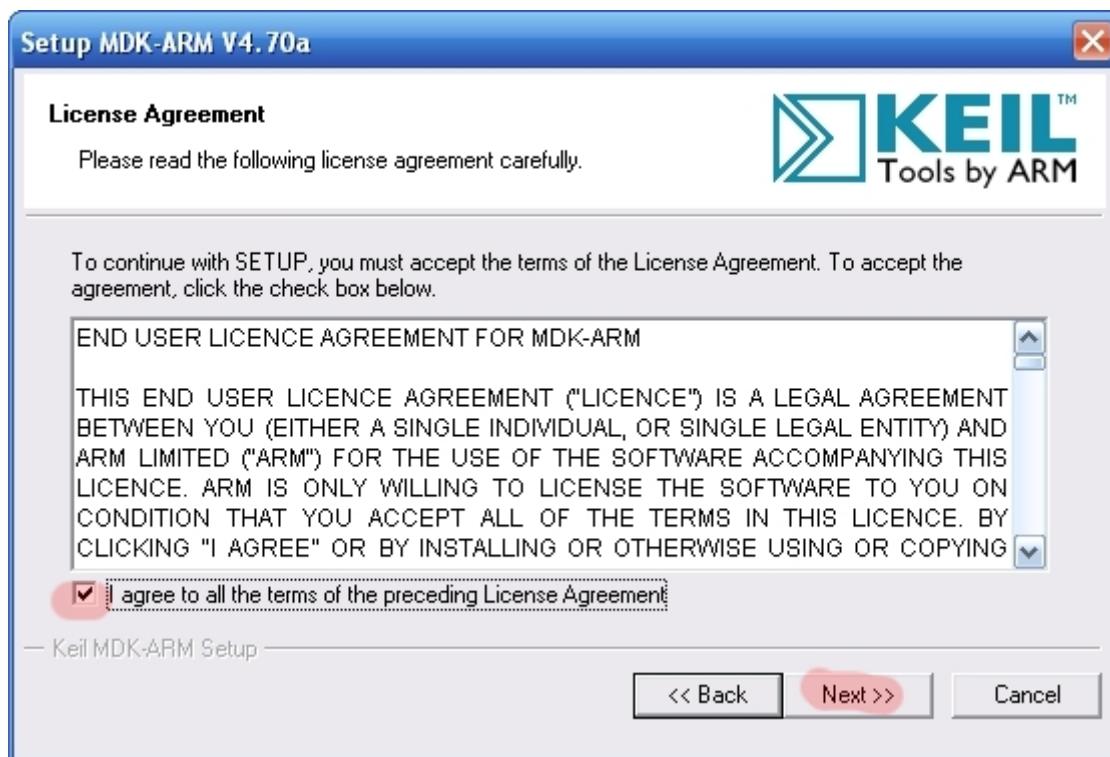


Для начала освоения программирования для ARM рекомендуем использовать бесплатный пакет от Keil: <http://www.keil.com/arm/mdk.asp> — ограничения бесплатной версии в 32К кода вполне достаточно для начального освоения программирования под процессоры семейства Cortex-Mx, а затем уже можно переползать на открытое ПО: GNU toolchain 9.1, Eclipse 8.4 и Linux XII.

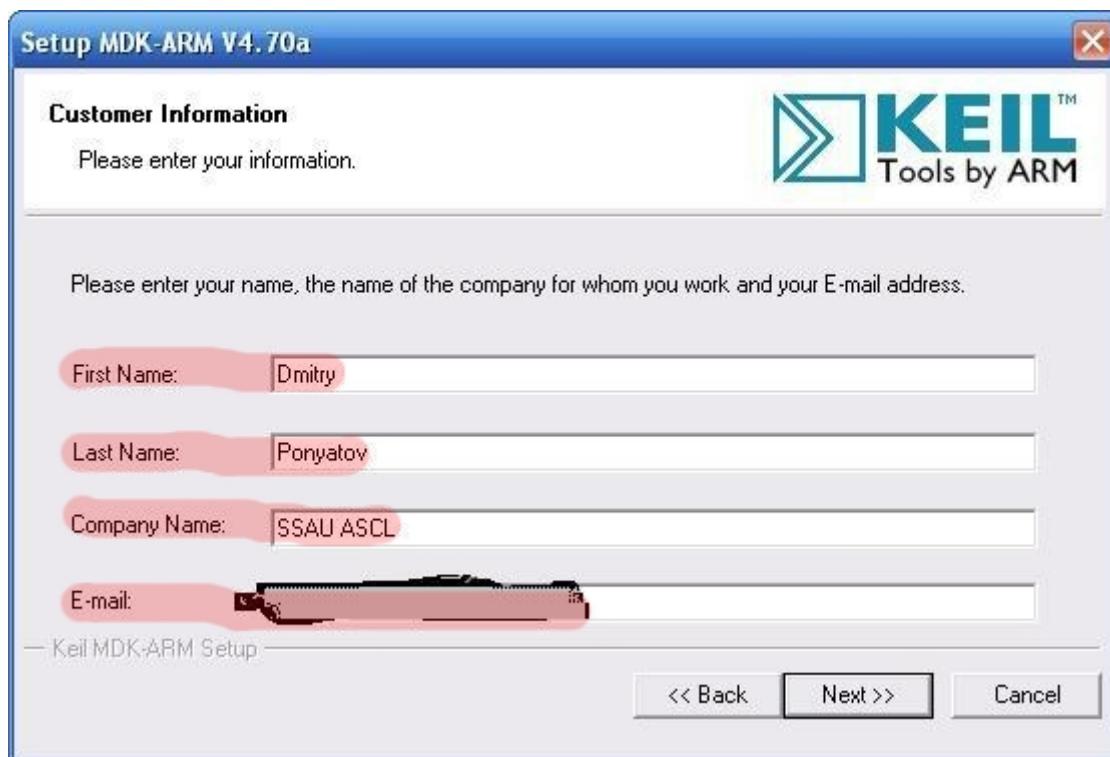
Установка

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

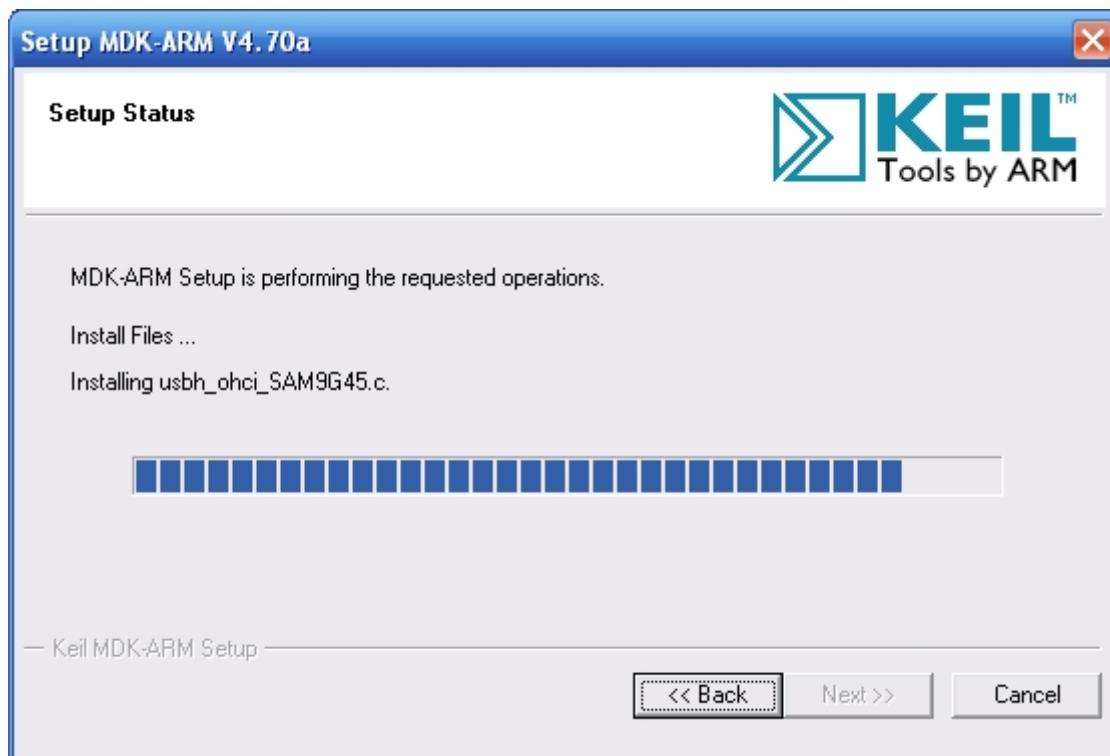


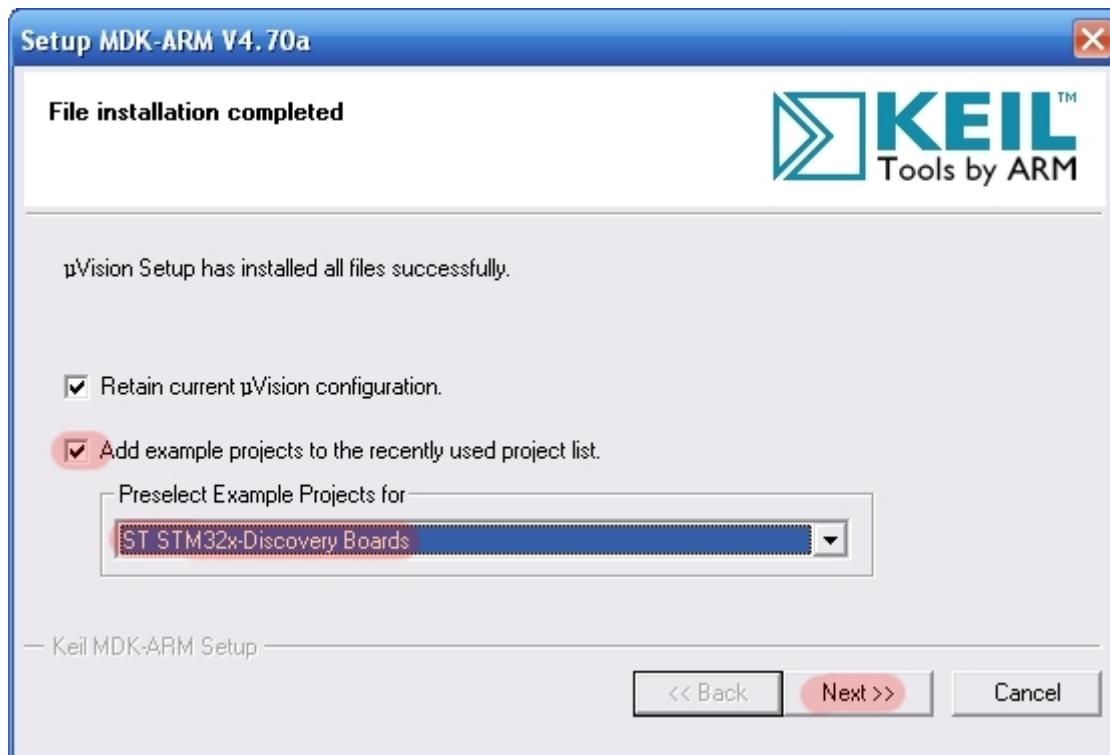


Путь установки пакета



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





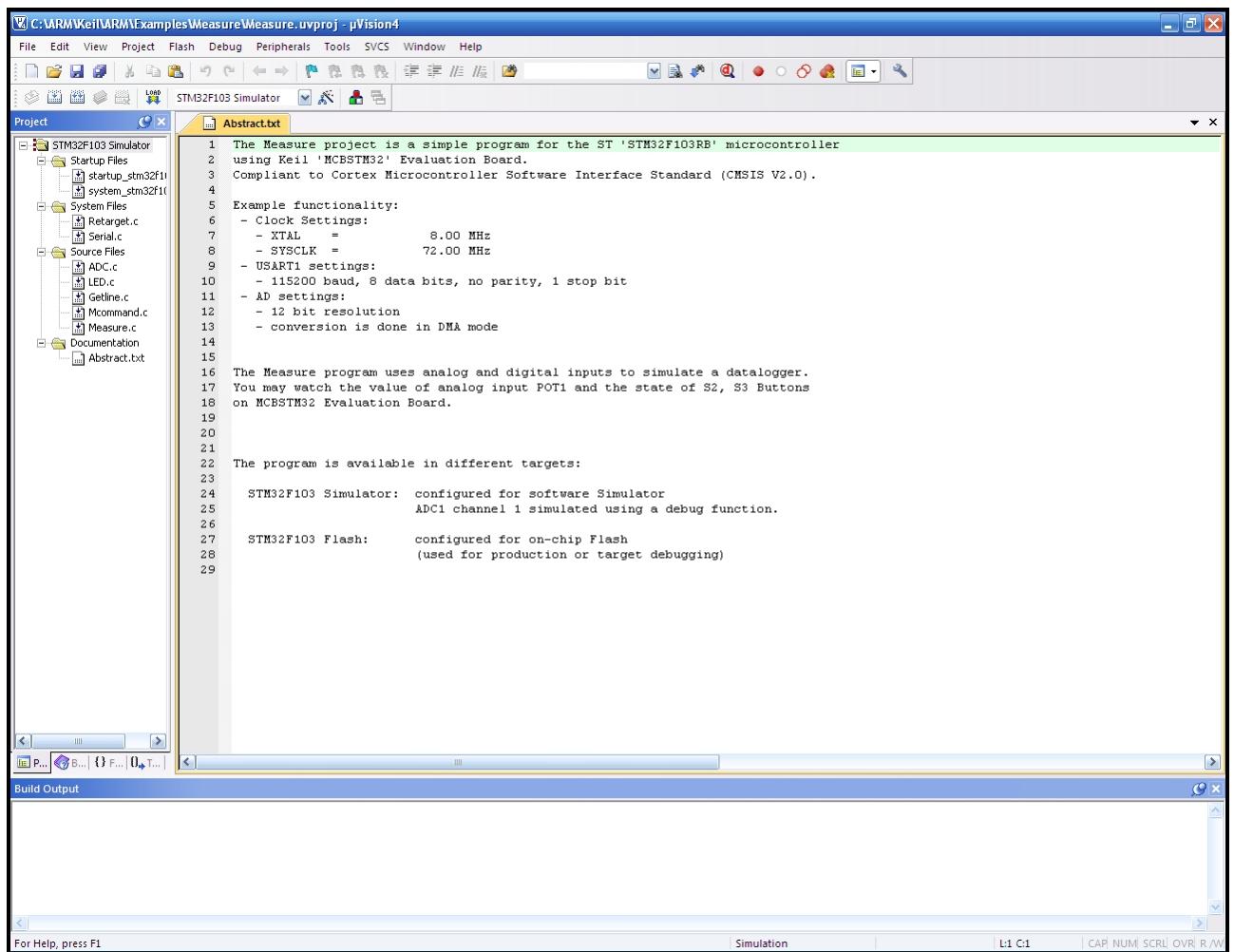
Укажите какие примеры кода добавить в список recently used project list: для работы с другими типами микропроцессоров выберете соответствующий раздел, или оставьте Simulation Hardware по умолчанию.



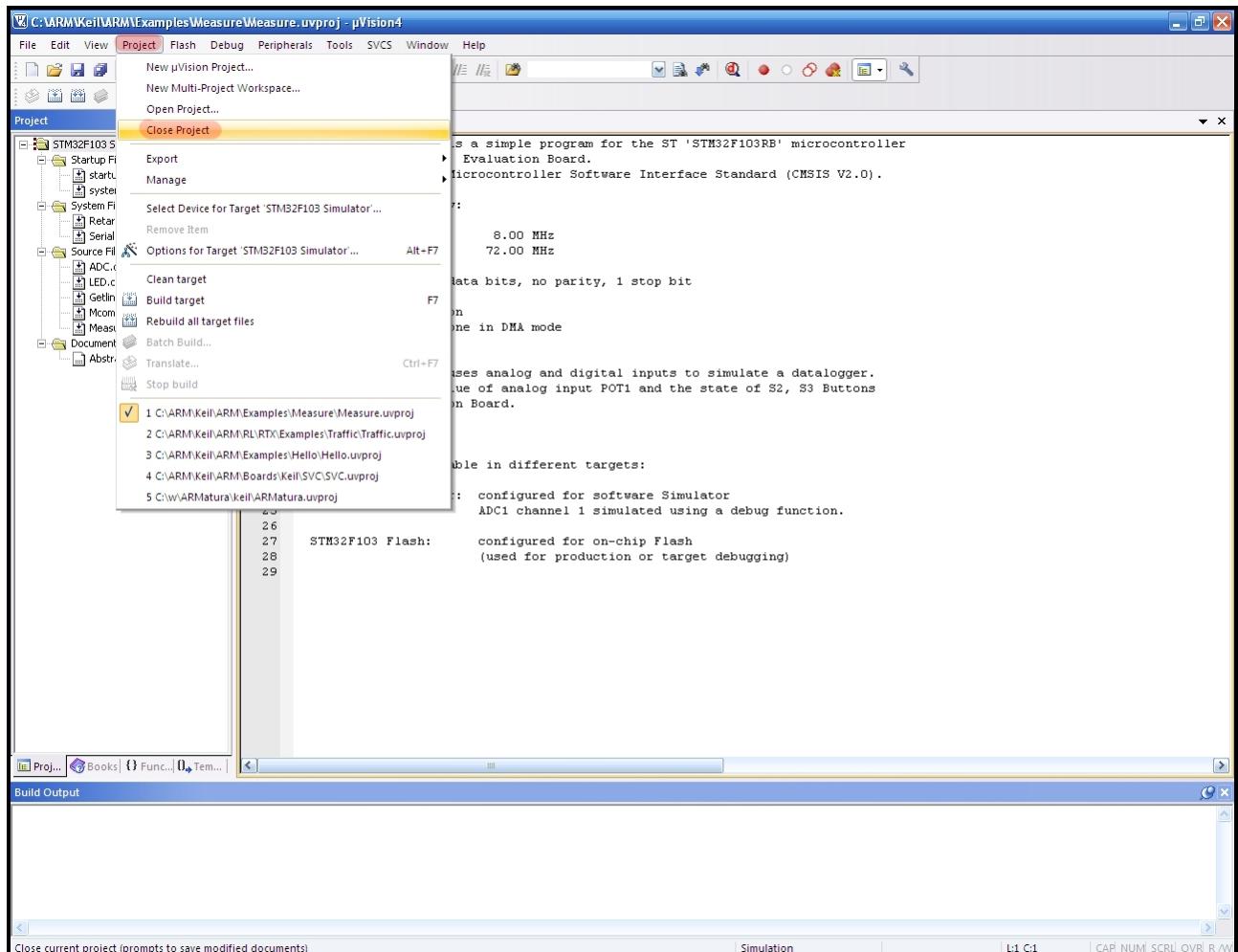
Снять установку драйвера программатора ULINK (если у вас его нет) и вывод текстового файла с последними изменениями Keil.



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



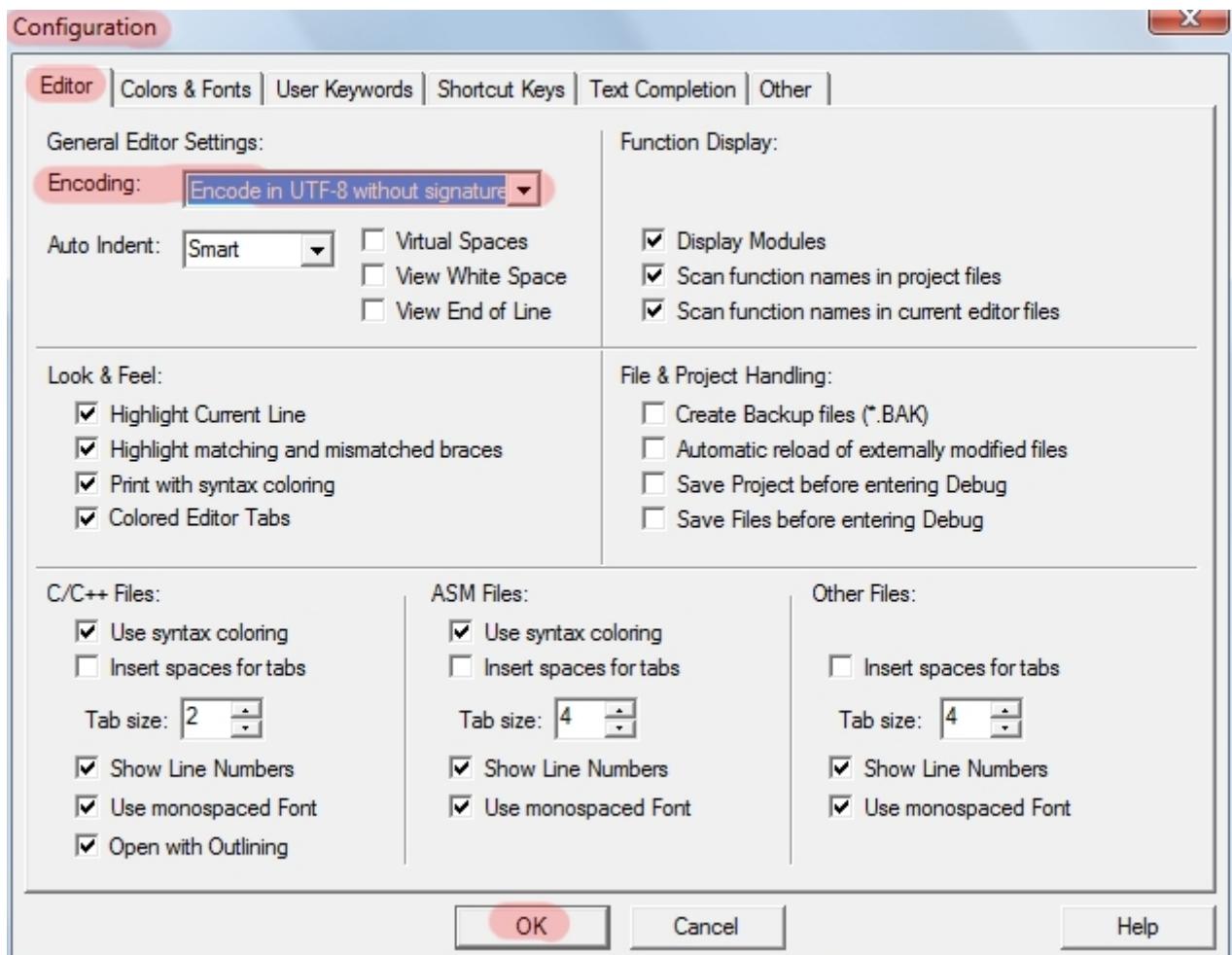
Нужно его закрыть,



и вернуться в раздел III и открыть первый проект 6.

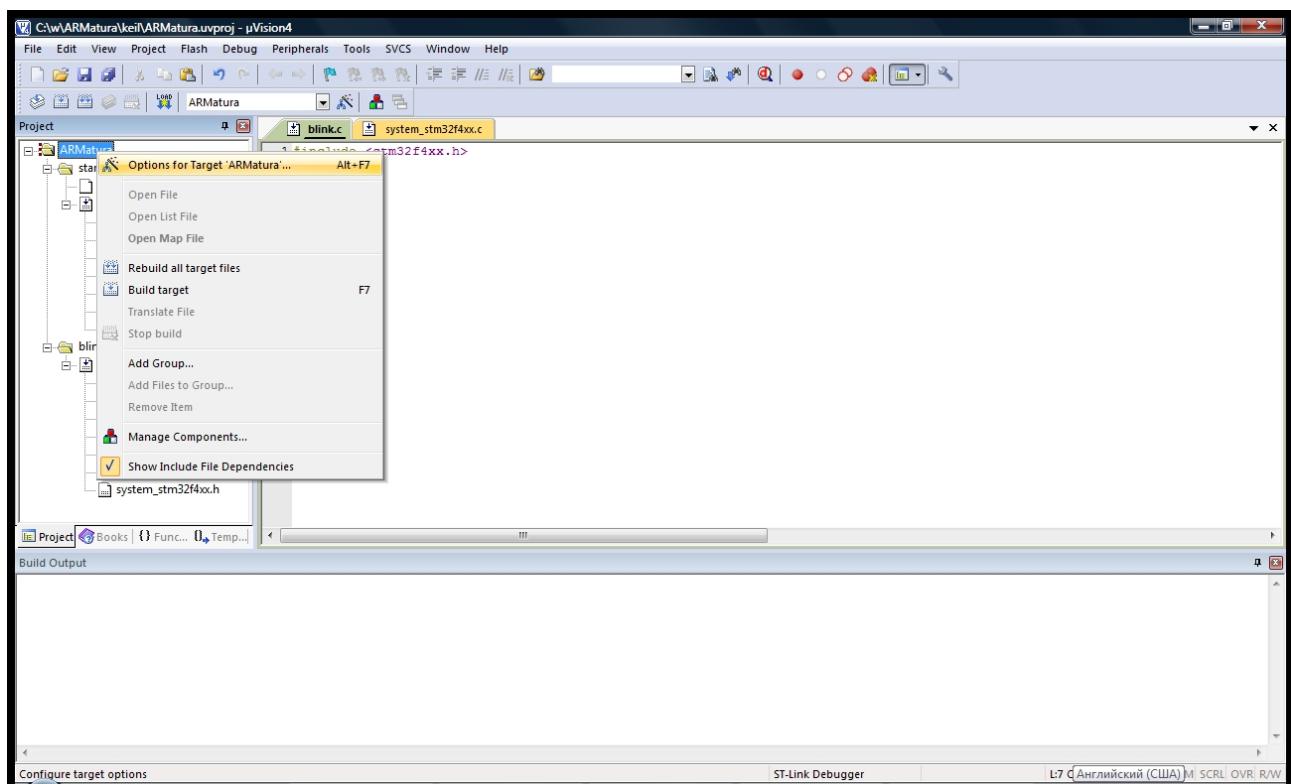
8.2 Настройки среды

По умолчанию Keil 8.1 косячит с кириллицей, поэтому идем в меню **Edit** > **Configuration** > **Editor** и устанавливаем кодировку UTF8

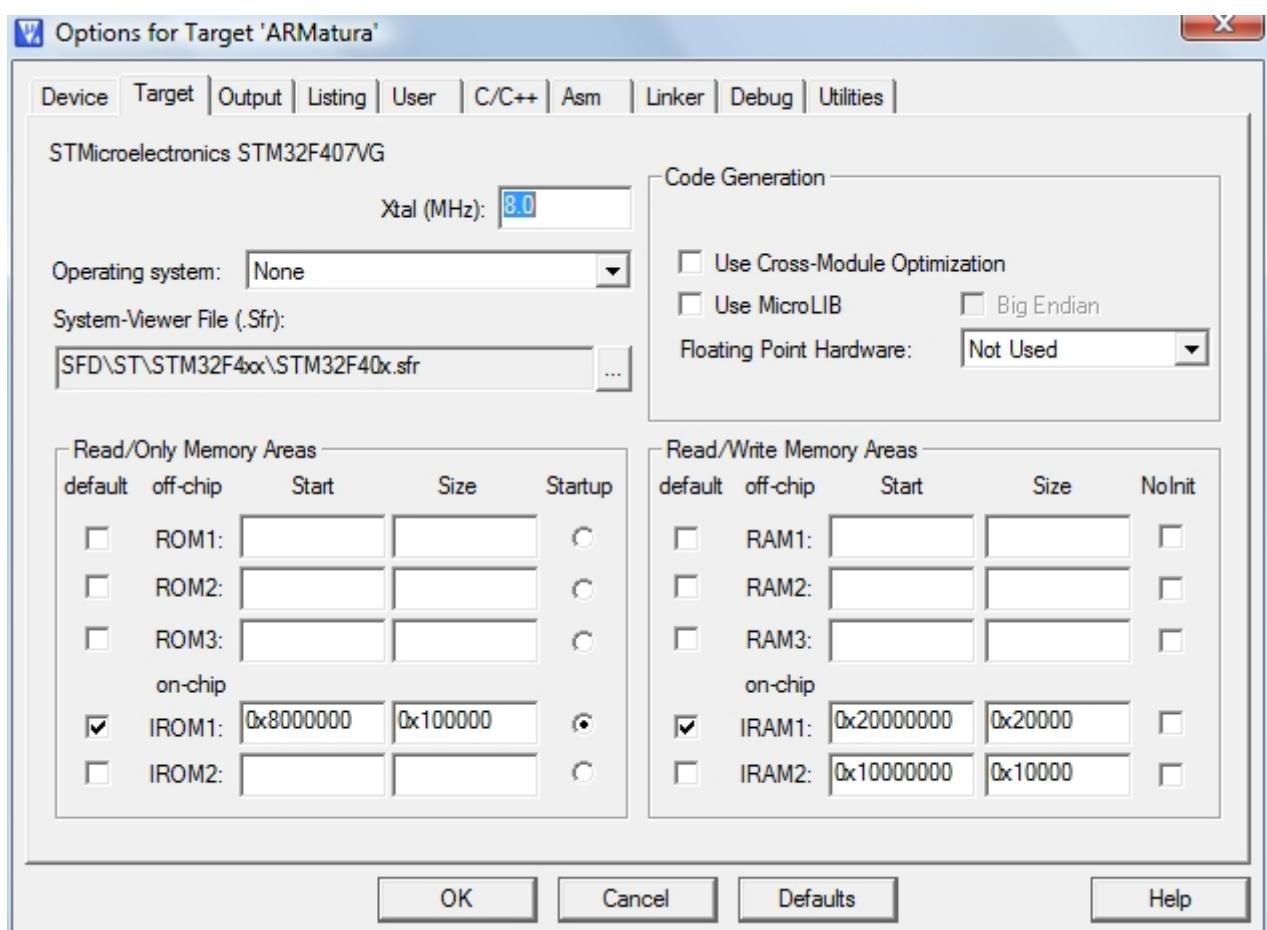


8.3 Настройки проекта

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`.

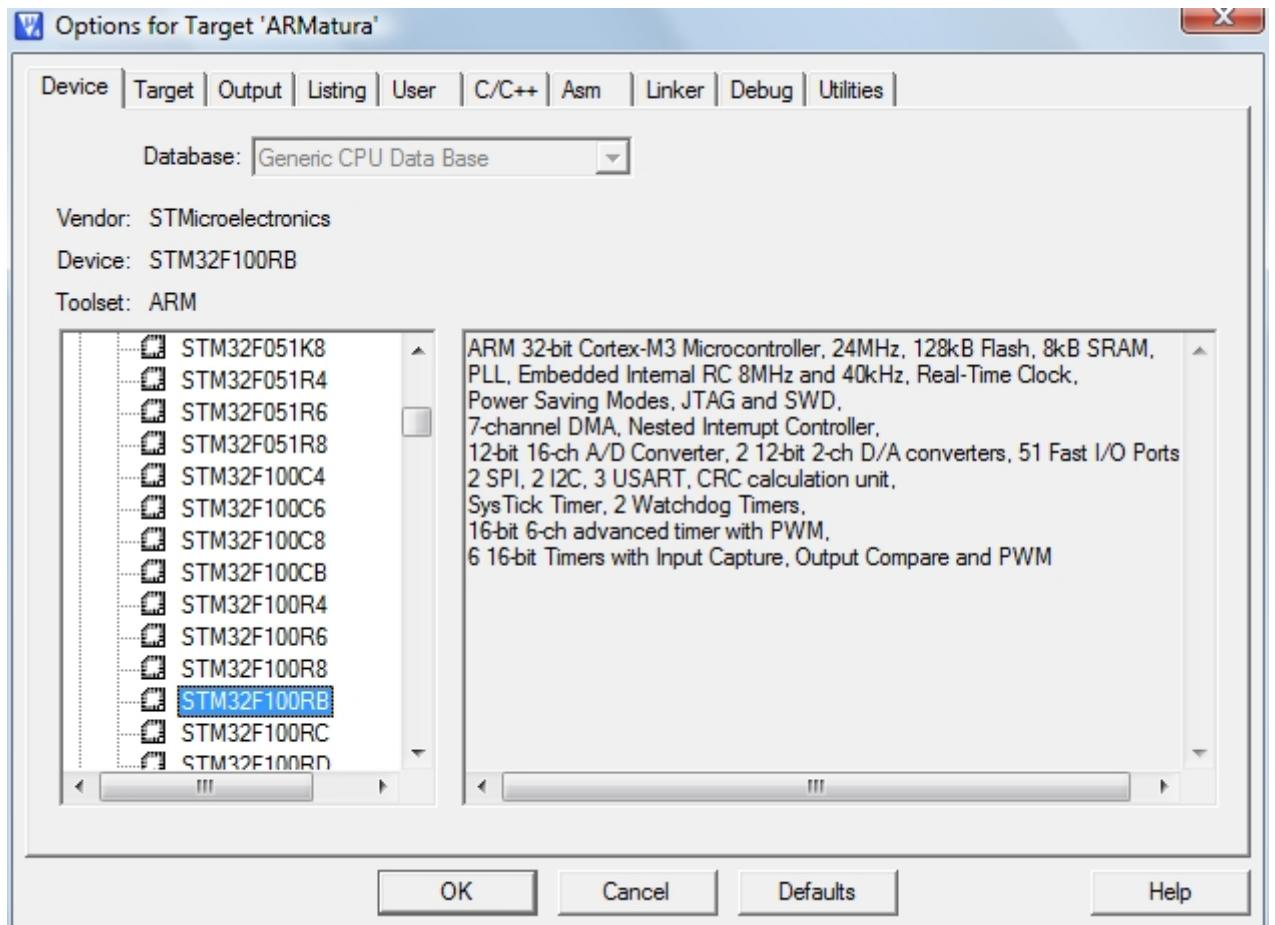


По умолчанию открывается вкладка **Target**:

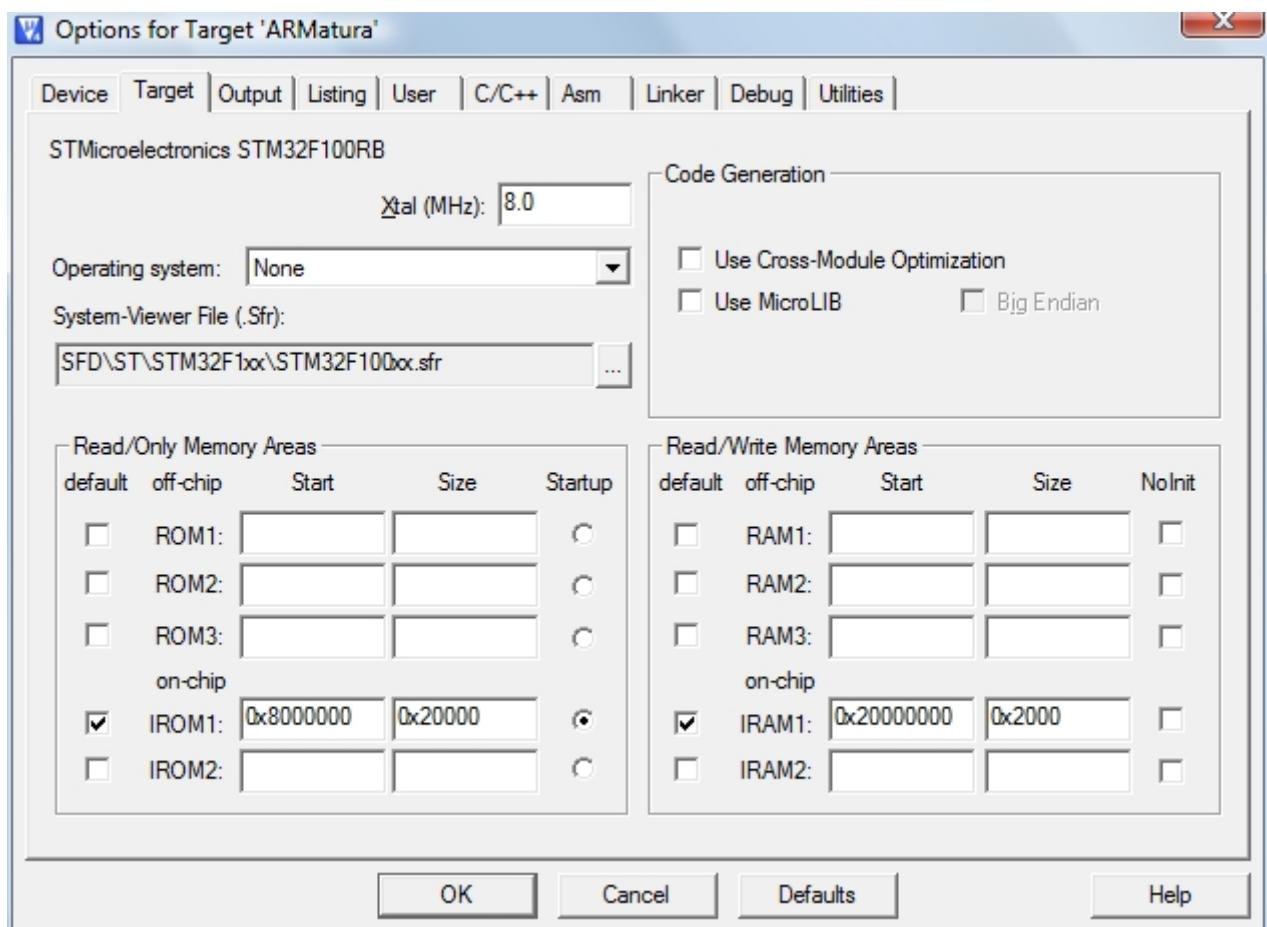


Xtal (MHz)	8.0	текущий процессор частота внешнего кварца на платах Discovery обычно стоит 8
Operating system:	None Use MicroLIB	или OCPB Keil RTX использовать libc от Keil
Floating Point Hardware:	Not Used	для Cortex-M4 доступен аппаратный FPU
	На этой вкладке также прописывается карта встроенной и внешней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

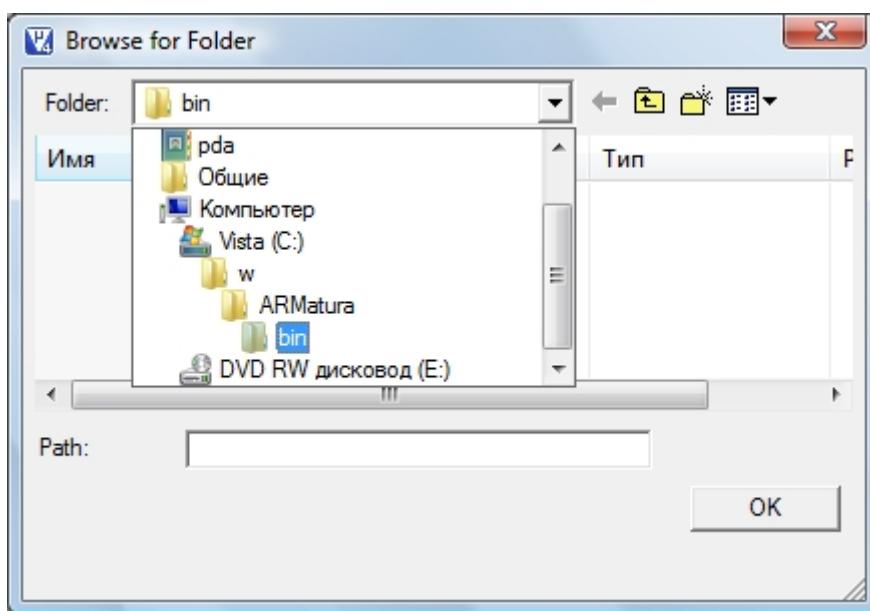
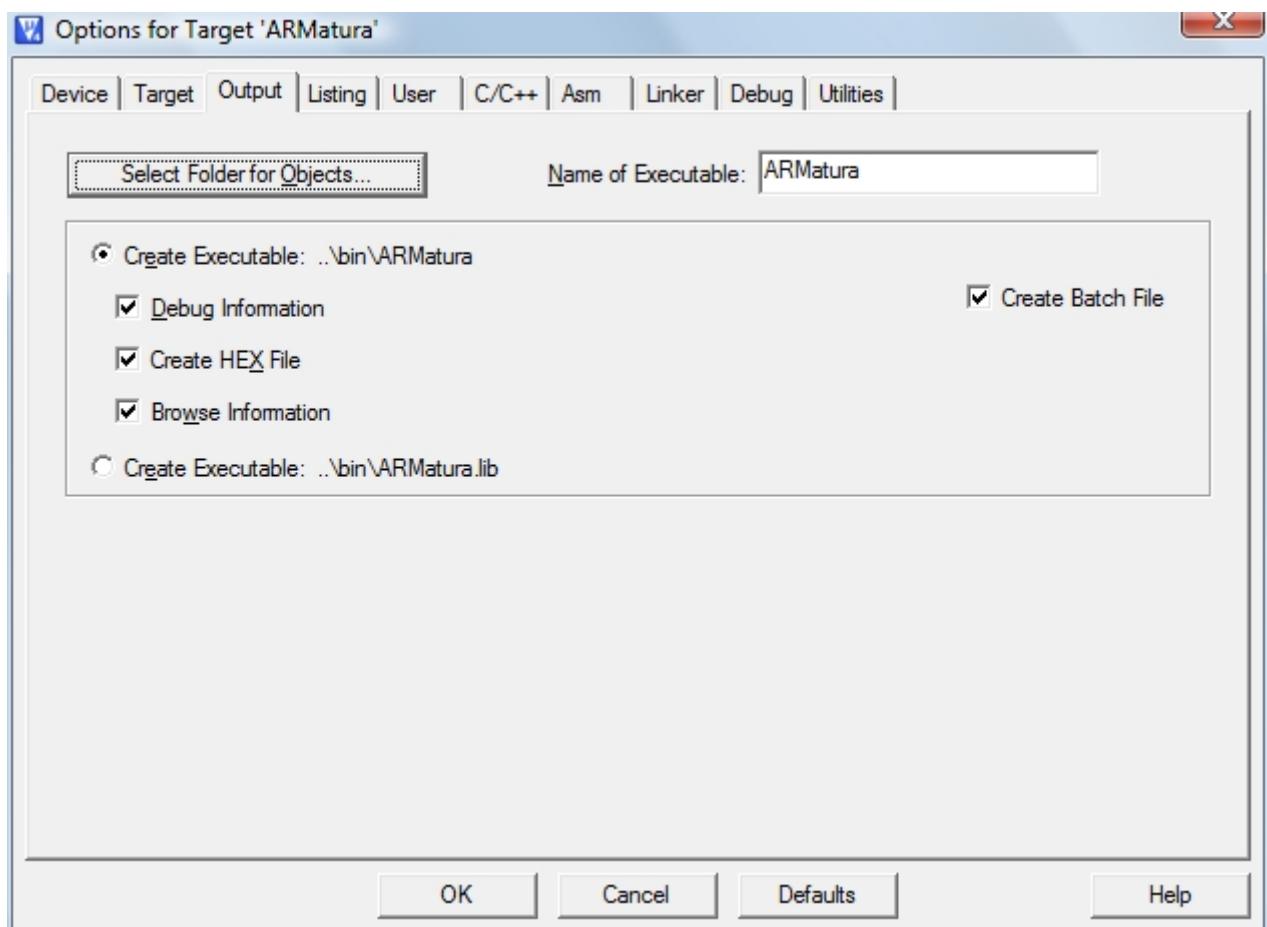
При необходимости собрать проект под другой процессор открываем вкладку **Device**, переконфигурируем проект под другую плату – STM32VLDISCOVERY 1:



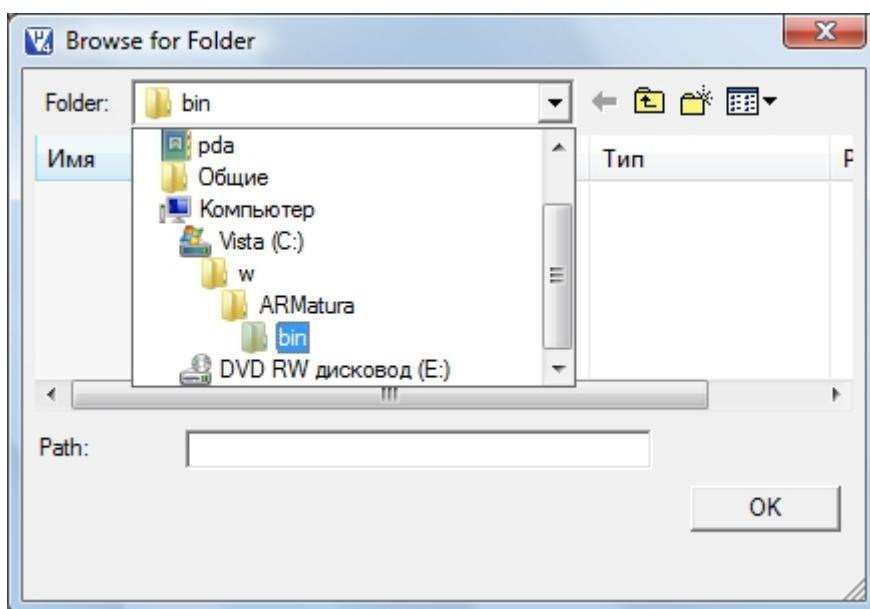
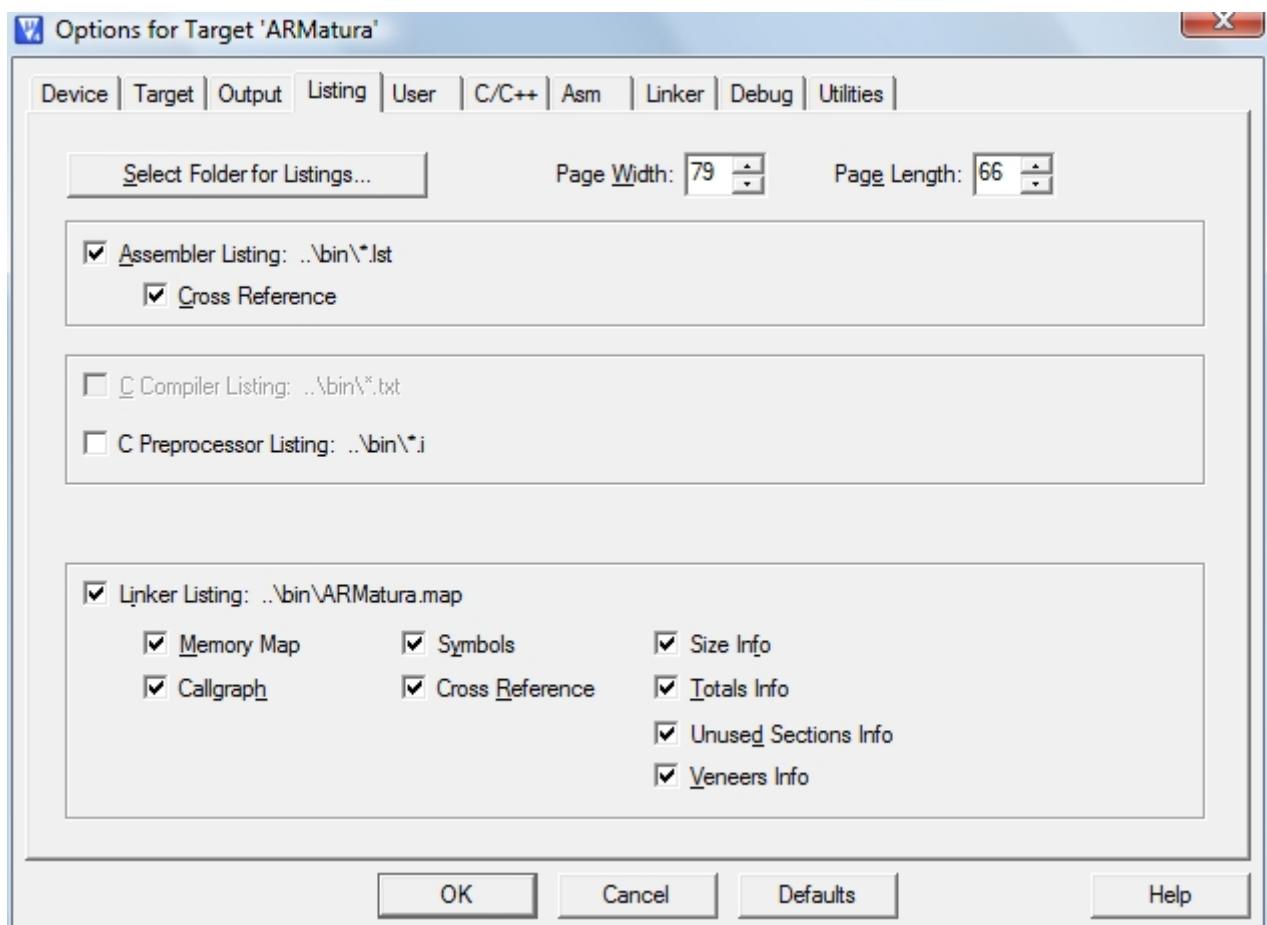
Обратите внимание что на вкладке **Target** изменился чип и настройки памяти:



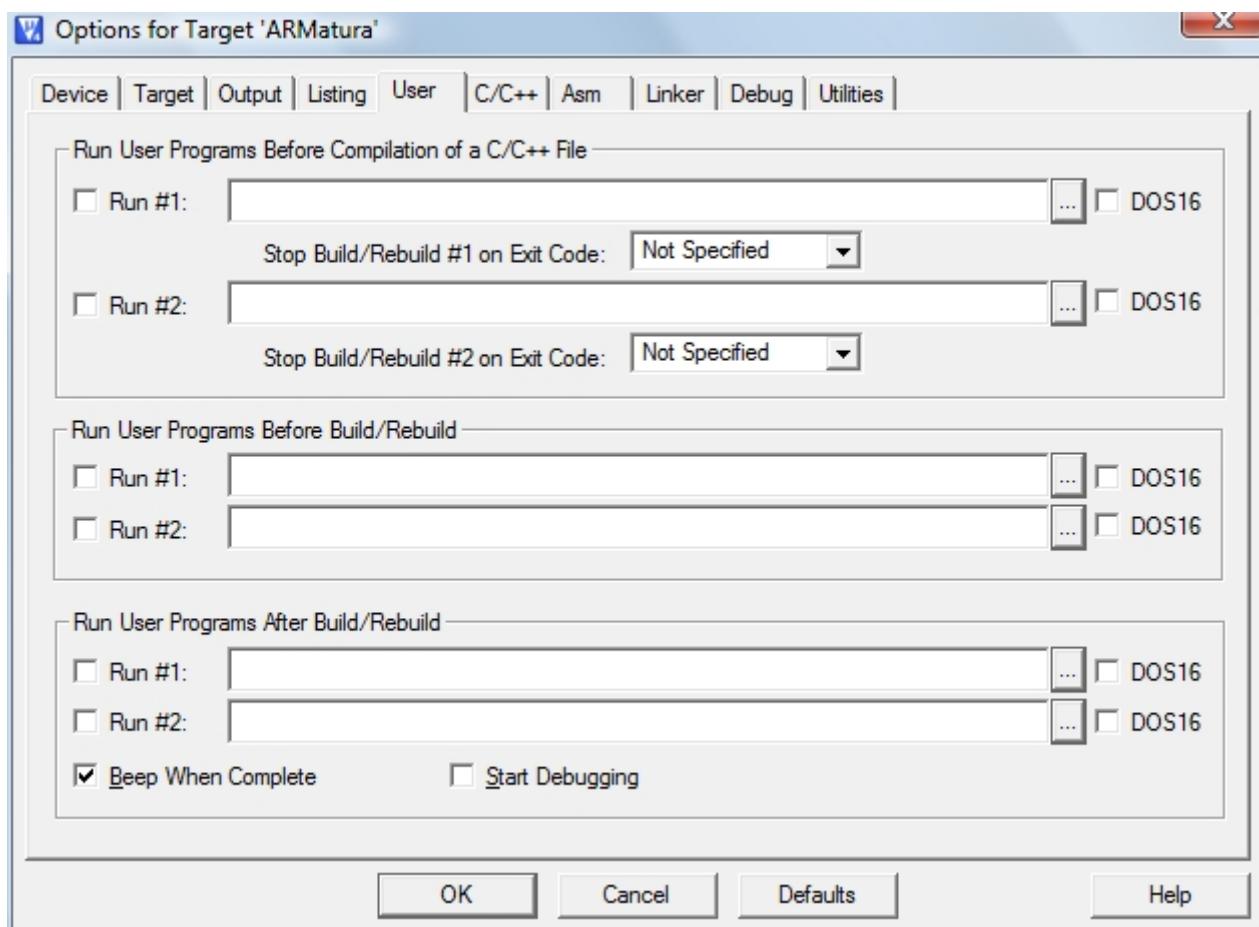
Output Настройки выходных файлов: каталог для бинарных файлов прошивки = firmware, при необходимости можно использовать .hex файл прошивки, имя файла прошивки



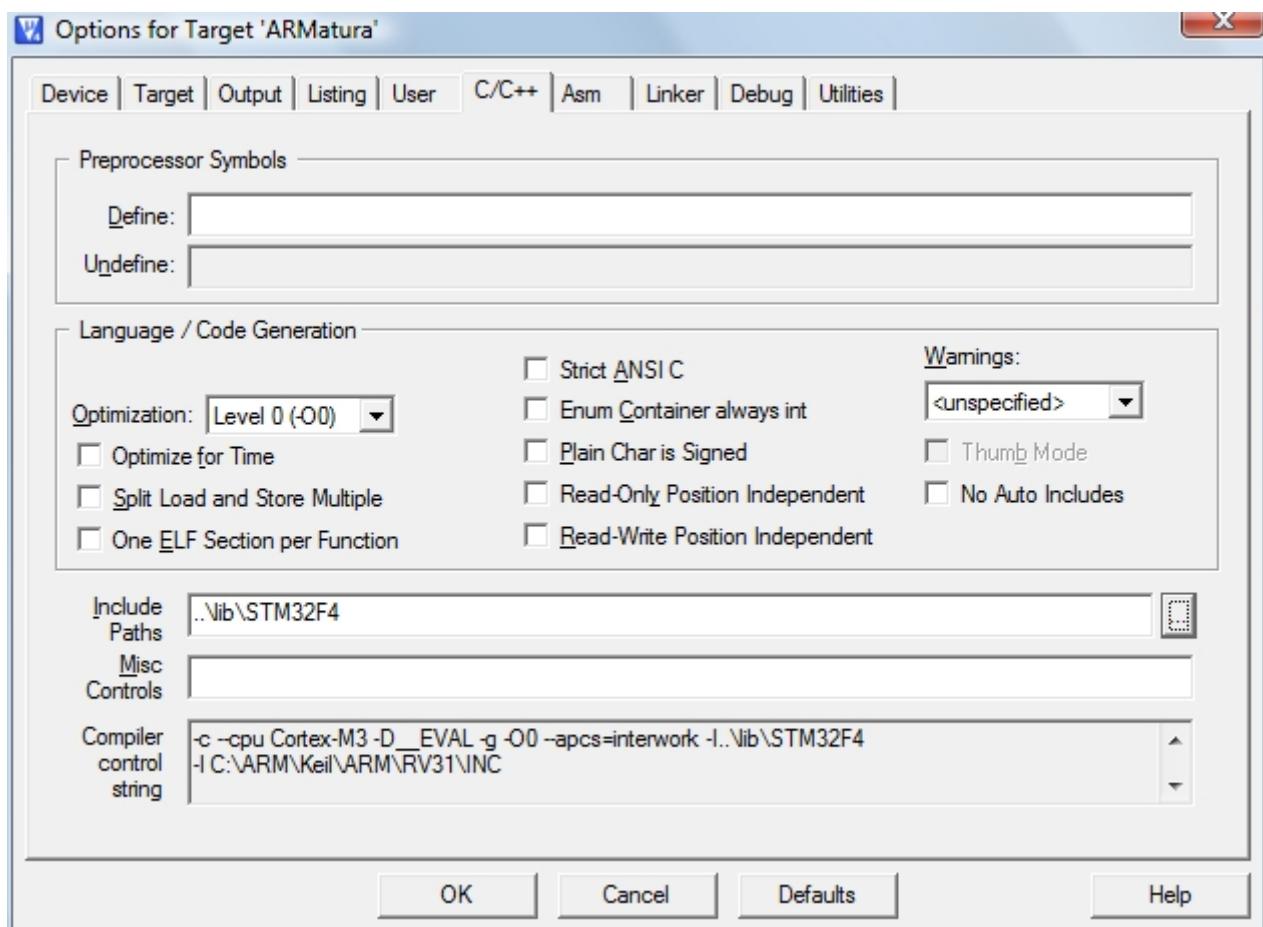
[Listing] Настройки генерации ассемблерных листингов в тот же каталог с прошивкой



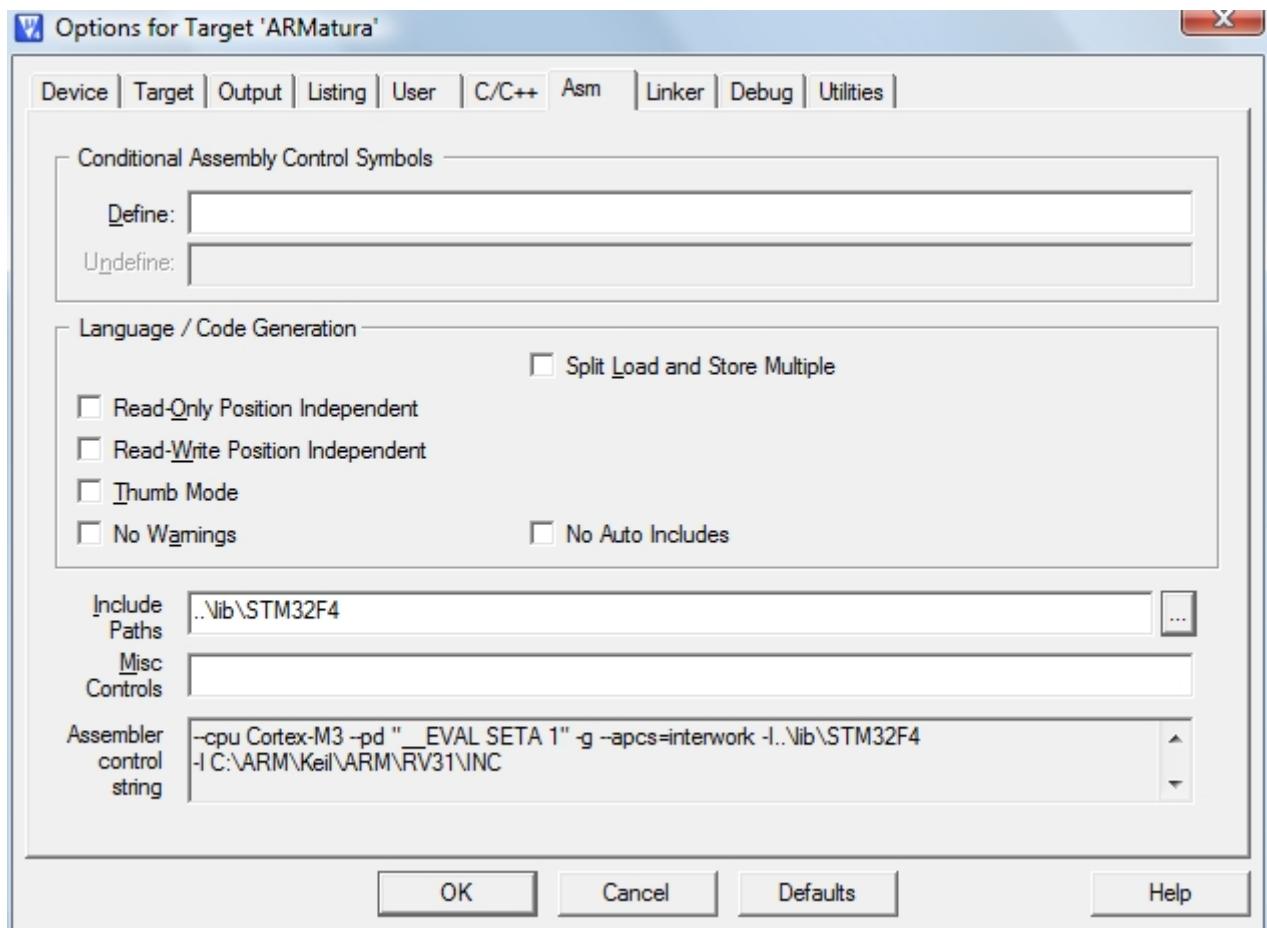
User Запуск пользовательских программ во время сборки проекта: пока не используем



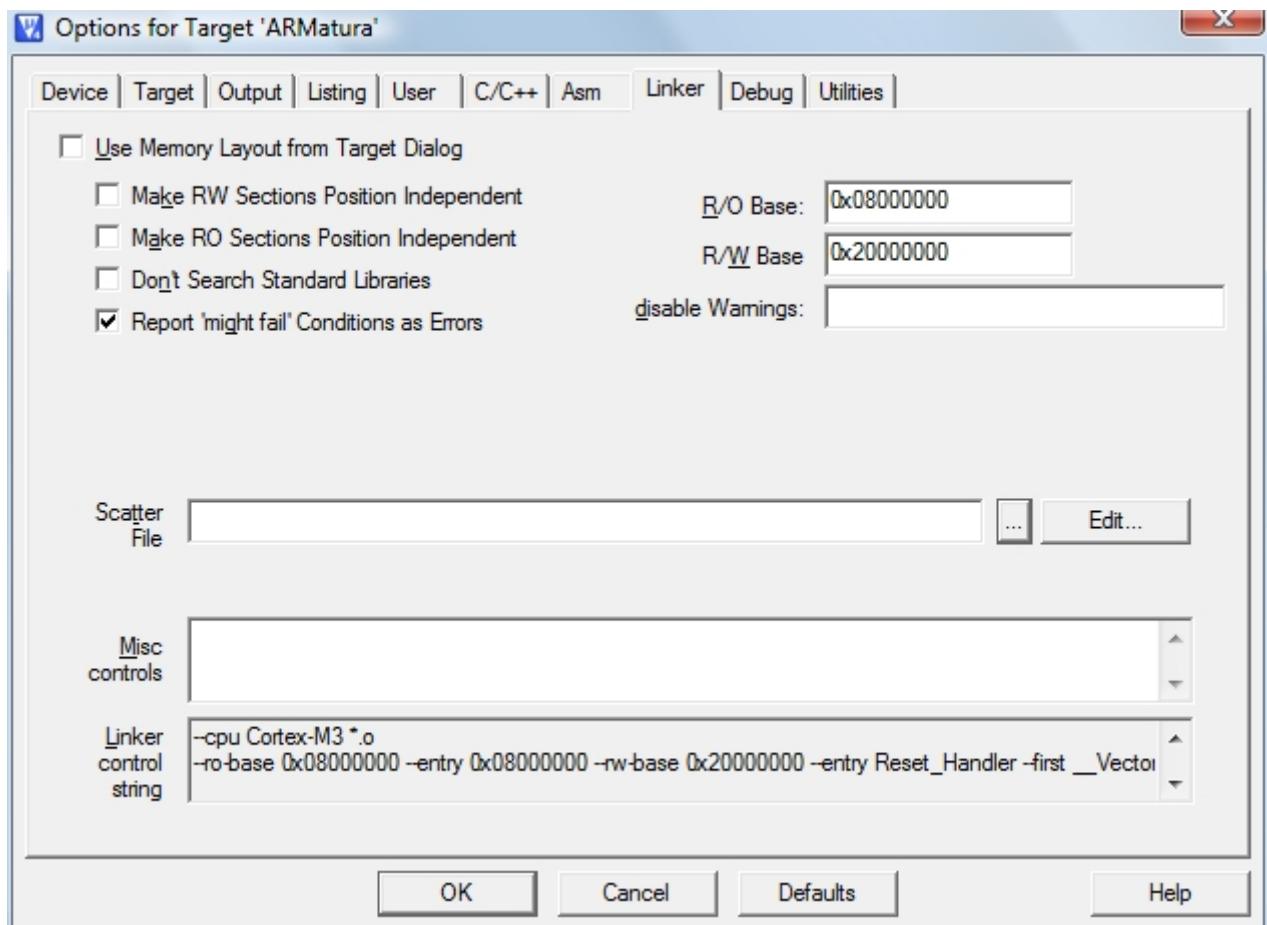
C/C++ Настройки компилятора C^{++} : опции оптимизации, каталоги библиотек и включаемых .h файлов, в нижнем поле прописывается полная командная строка вызова компилятора, которая может вам в дальнейшем понадобится, если потребуется компилировать без использования Keil IDE.



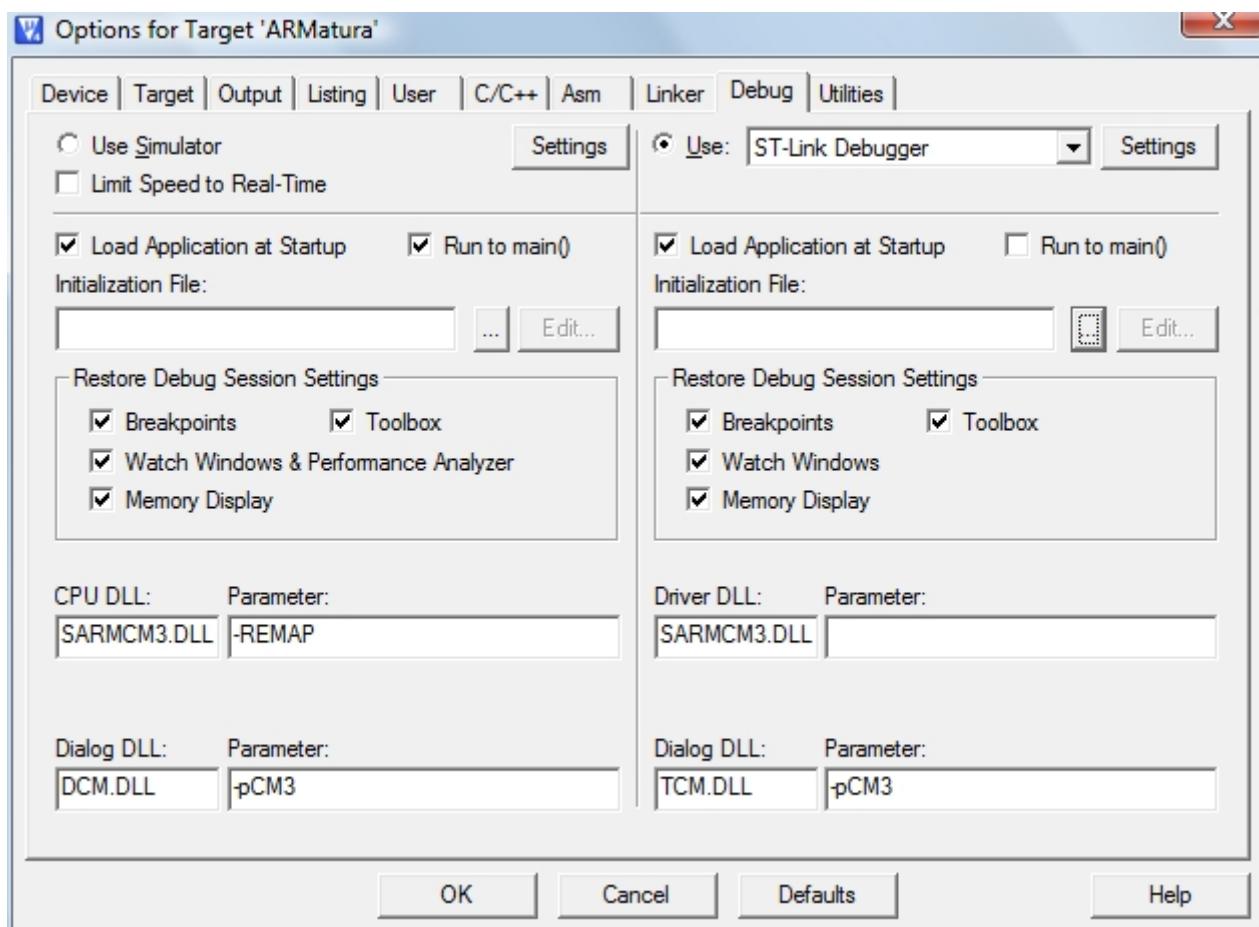
Asm Настройки ассемблера, приблизительно те же что и для C^{++}



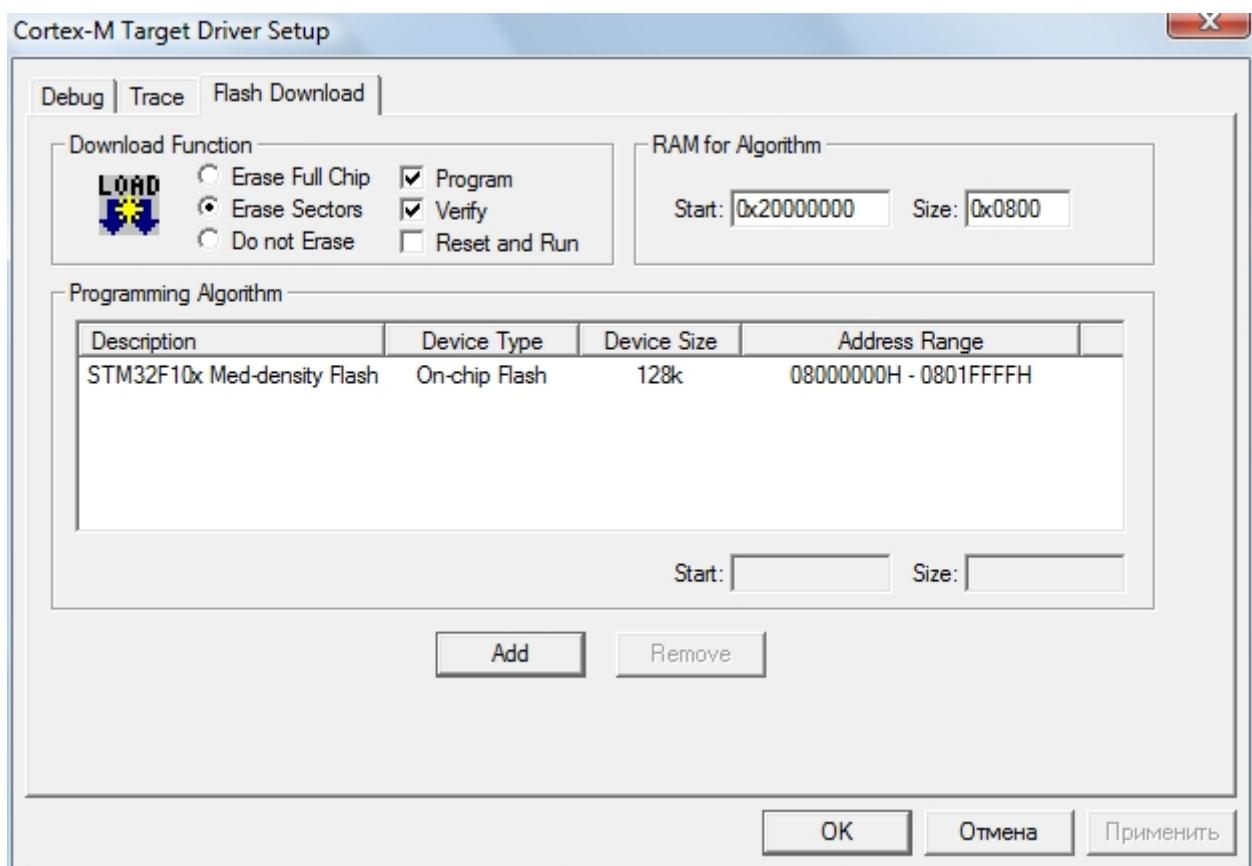
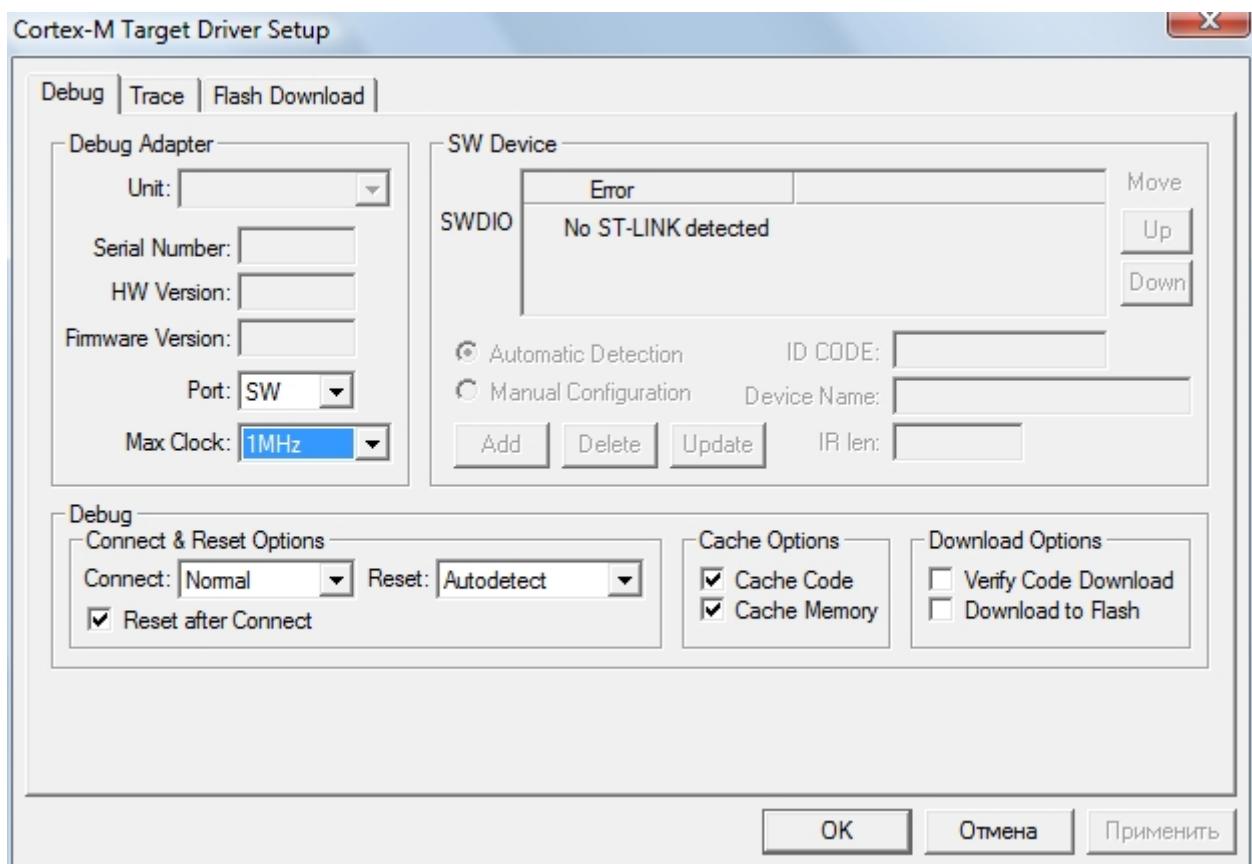
Linker Настройки линкера, который собирает объектные файлы .o, в которые компилируются каждый программный файл (модуль) по отдельности, в один готовый файл прошивки (настройки карты памяти контроллера, адреса точки входа программы и т.п.)



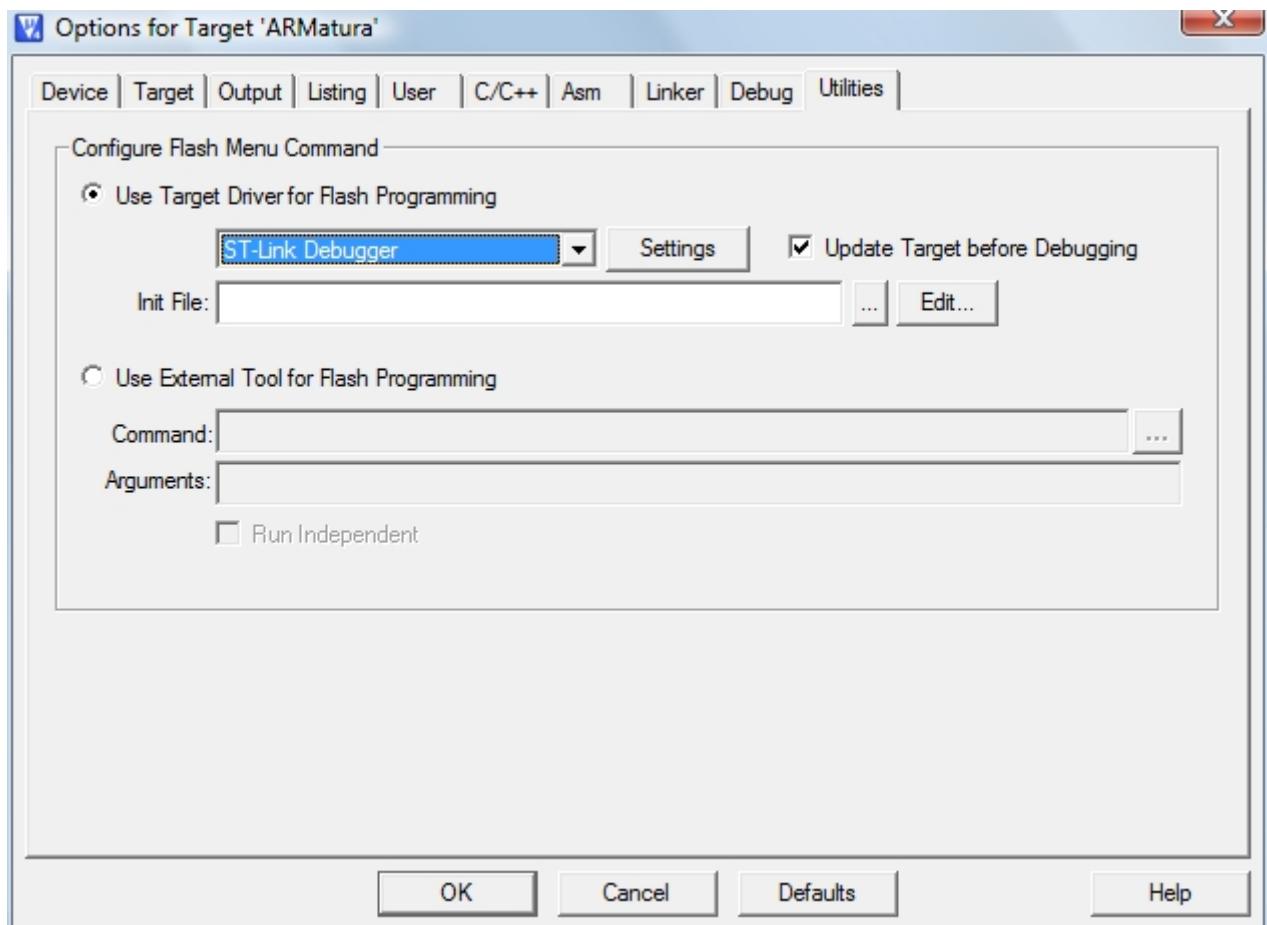
Debug Настройки отладки с помощью адаптера – STlink (внешний или встроенный в оценочную плату), JTAG.



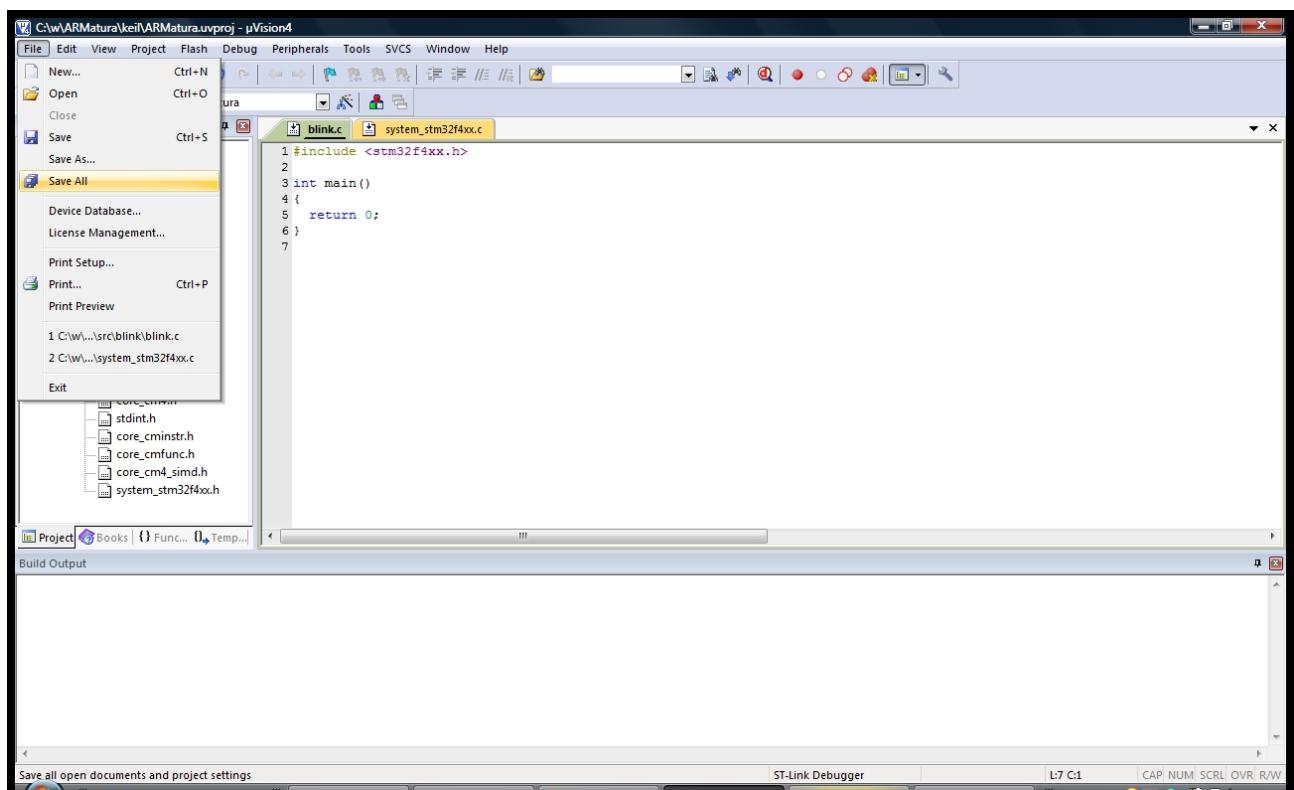
Настройки адаптера STlink – включаем режим SWD для варианта встроенного на оценочную плату, и обнаруживаем что не установили пакет поддержки STlink и драйвера.



Utilities Настраиваем в качестве программатора для прошивки тот же STlink



Сохраняем настроенный проект



8.4 Eclipse

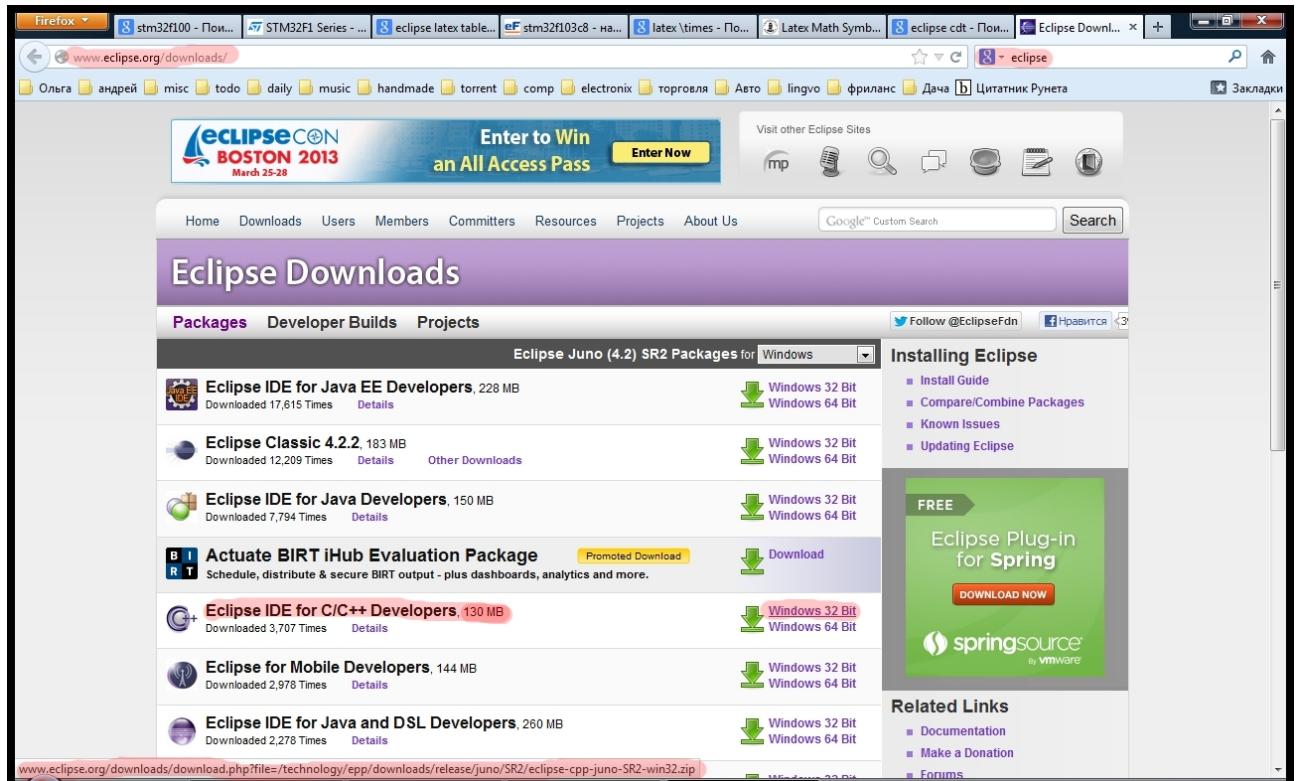
Установка

Для работы нам потребуется сборка Eclipse 8.4 в варианте

```
http://www.eclipse.org/downloads/
http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/
release/juno/SR2/eclipse-cpp-juno-SR2-win32.zip
```

Пакет поставляется в виде zip-архива, который достаточно распаковать в любое место, например в C:\Eclipse\.

После запуска нужно будет указать каталог для хранения проектов: C:\w\.

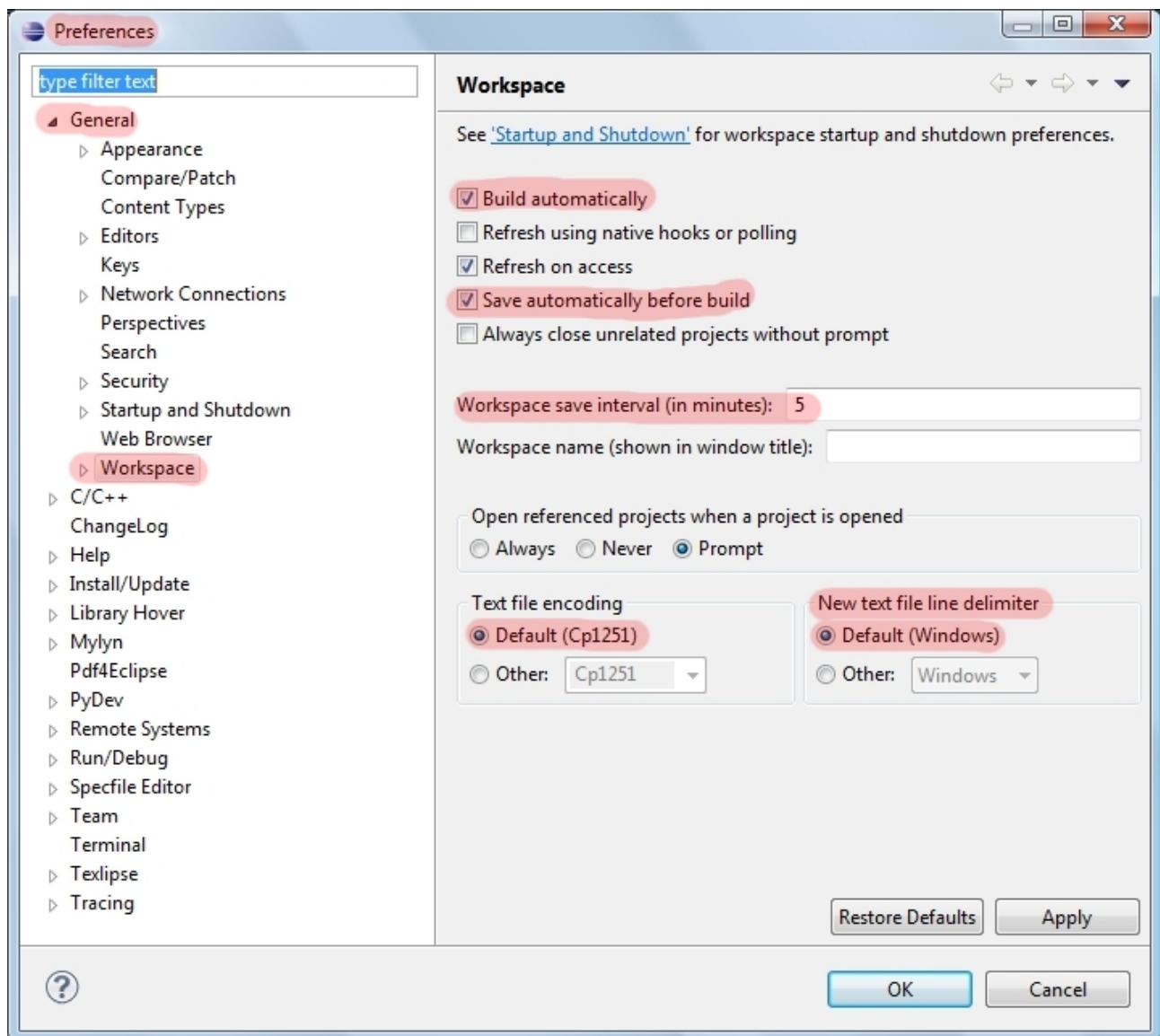


Настройка глобальных свойств

Глобальные свойства настраиваются в меню `Windows > Preferences`.

Прежде всего настроим автосохранение при запуске Build `Ctrl + B`:

- запуск сборки при сохранении файла
- автоматическое сохранение всех измененных файлов перед сборкой
- автосохранение рабочих файлов
- кодировка файлов с кириллицей (может потребоваться корректировка при импорте файлов в кодировках KOI8, UTF8)
- конец строки в стиле dos/windows: <CR><LF> (в UNIX используется только <LF>)



8.5 Code::Blocks

8.6 gVim

8.7 IAR

Глава 9

Компиляторы

9.1 GCC

9.2 KeilCC

9.3 IAR

Глава 10

Адаптеры

Адаптер = программатор = внутрисхемный отладчик = ... — аппаратное устройство, подключаемое одним концом в компьютер (обычно по USB), а другим в отлаживаемое устройство с микроконтроллером.

По подключению к устройству разделяют JTAG-адAPTERЫ, адAPTERЫ, впариваемые производителями конкретных чипов (Atmel, STmicroelectronics), и коммерческих IDE (Keil Ulink), и разнообразный самопал.

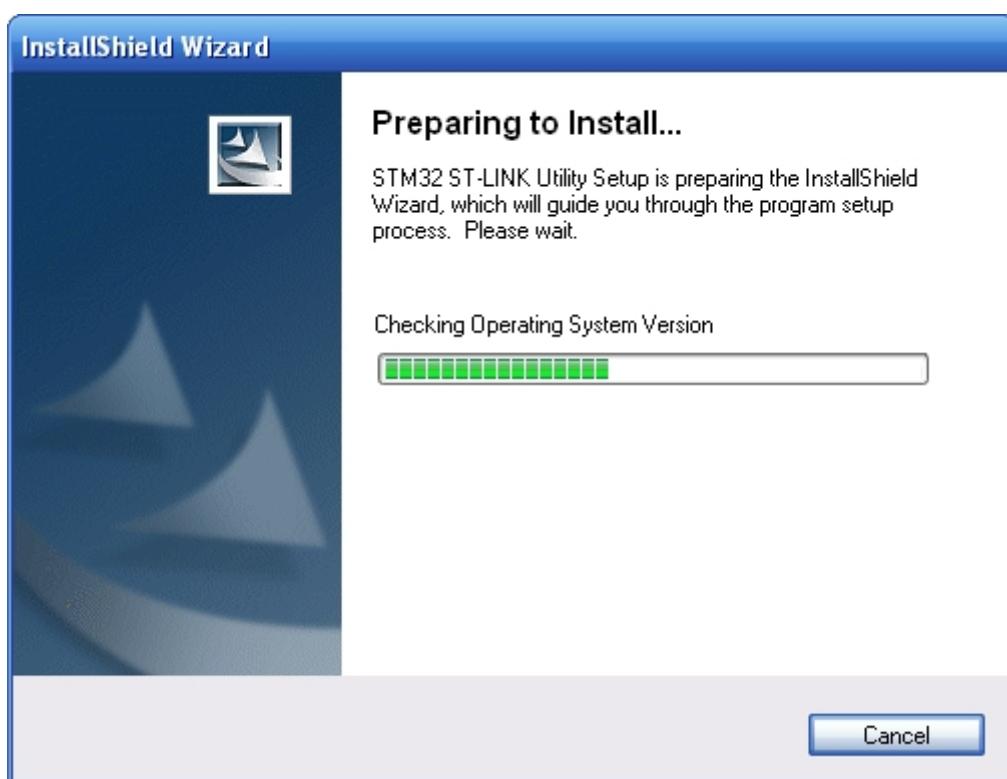
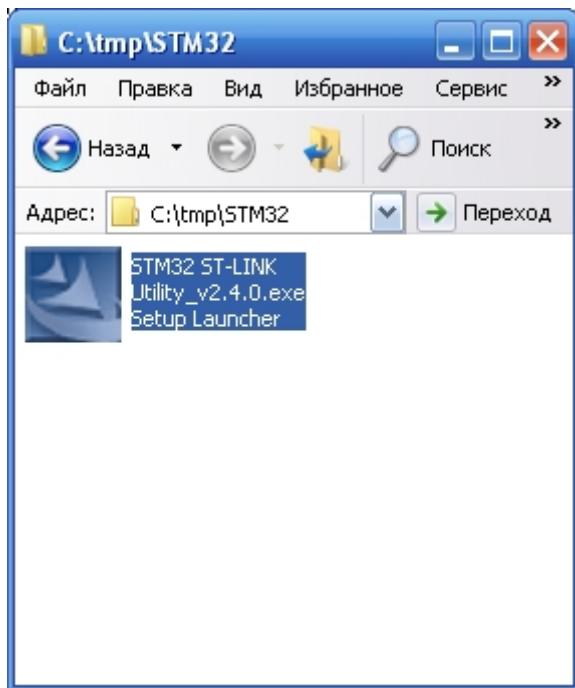
АдAPTER обеспечивает

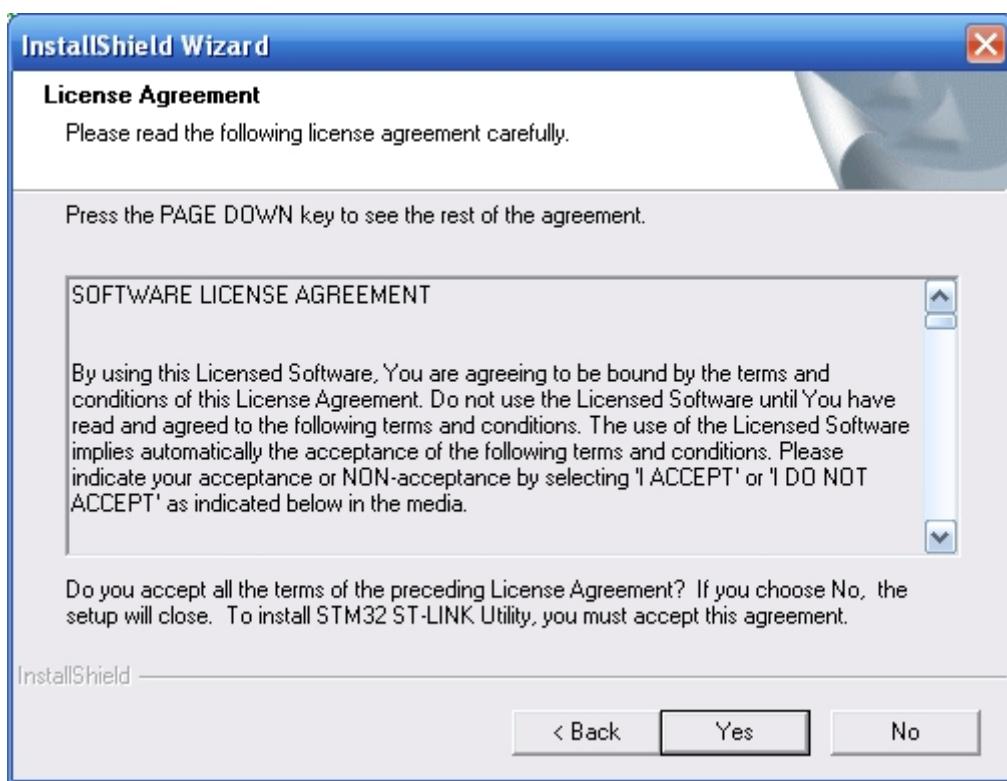
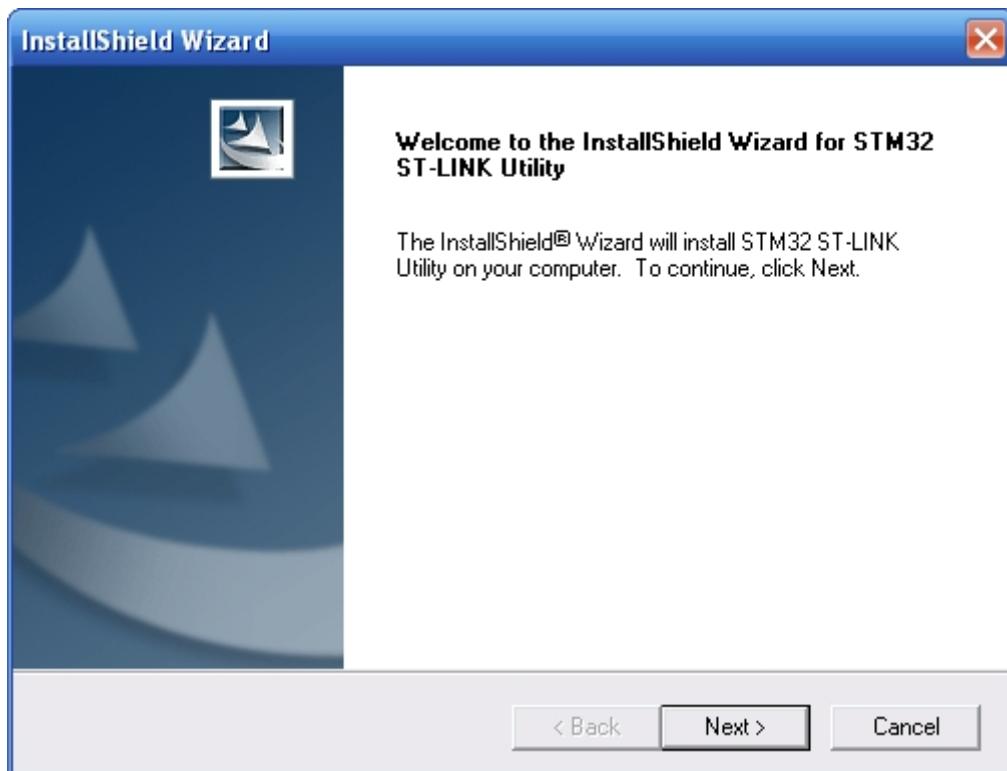
- загрузка прошивки во встроенную Flash (обеспечивается всеми адAPTERами)
- чтение/запись областей памяти
- интерактивную отладку в процессе работы программы ??

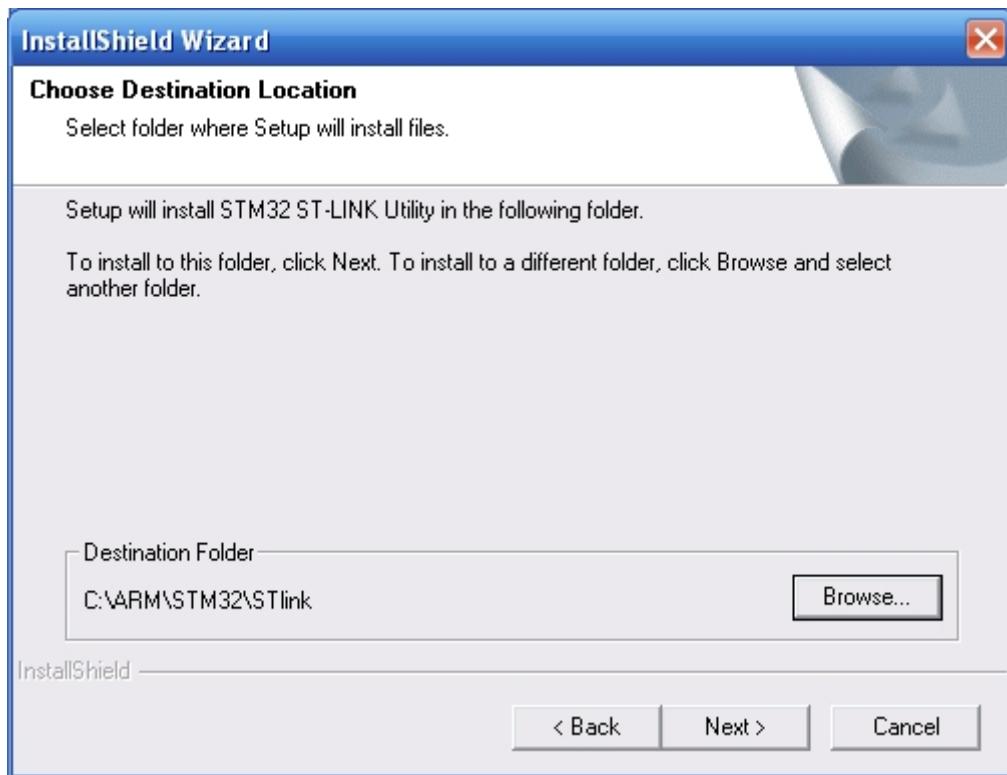
10.1 STlink

Установка ПО

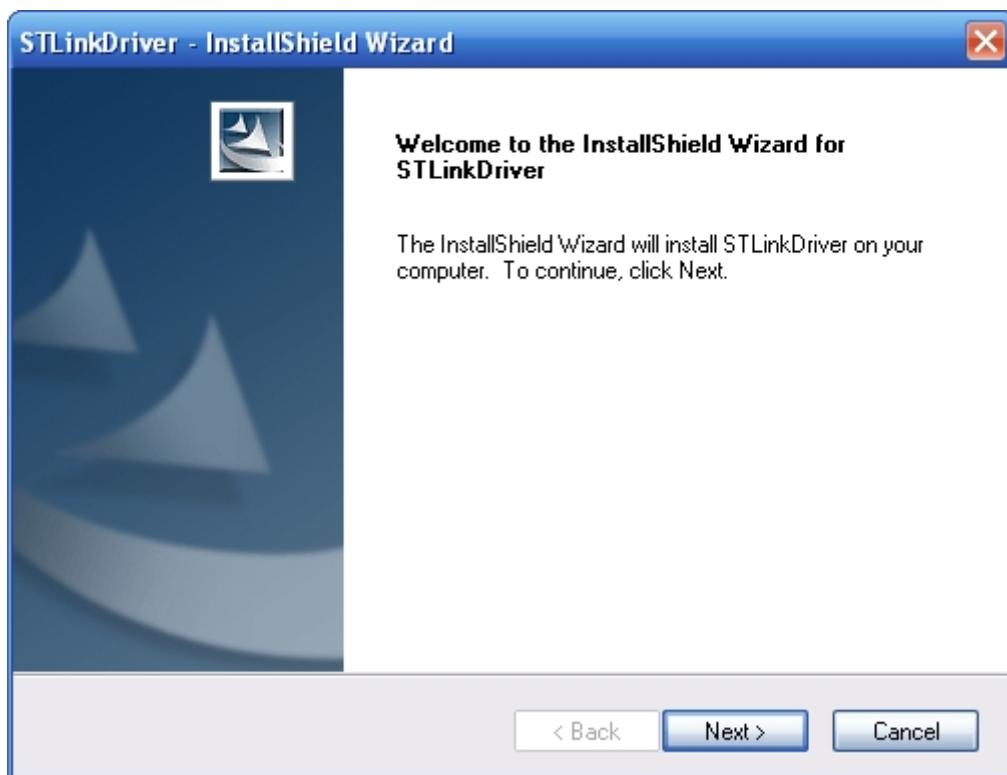
Для работы с чипами STM32 используя программаторы STlink, встроенные на демоплаты Discovery необходимо установить пакет STSW-LINK004 (STM32 ST-link Utility и драйвер), скачав его с <http://www.st.com/web/en/catalog/tools/PF258168>

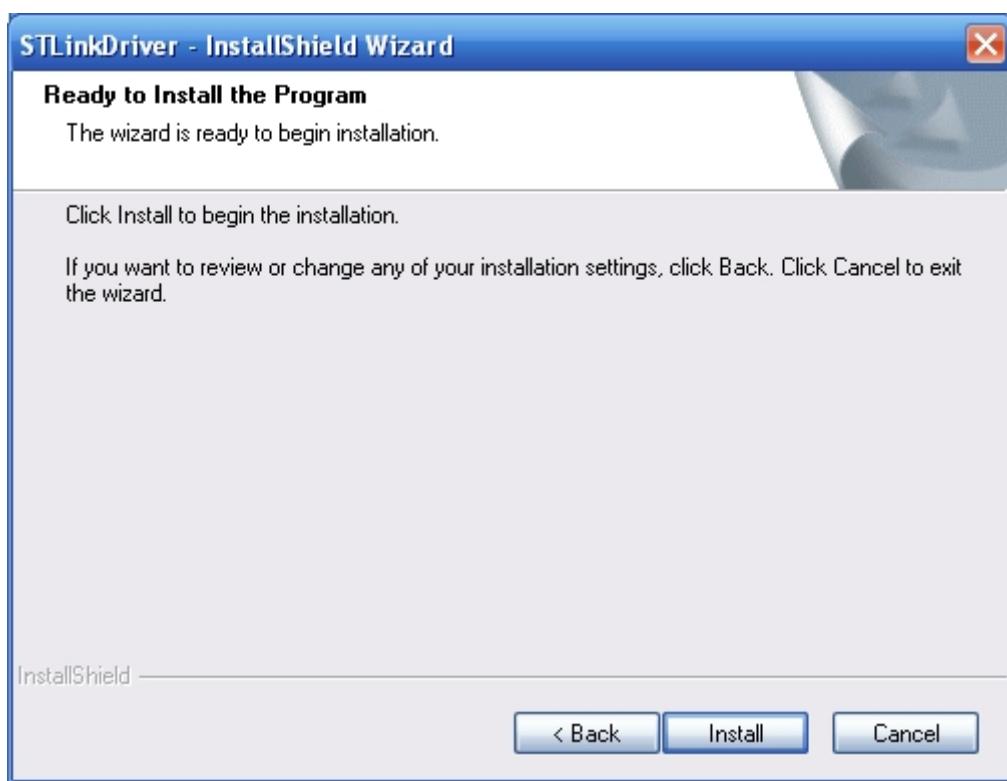
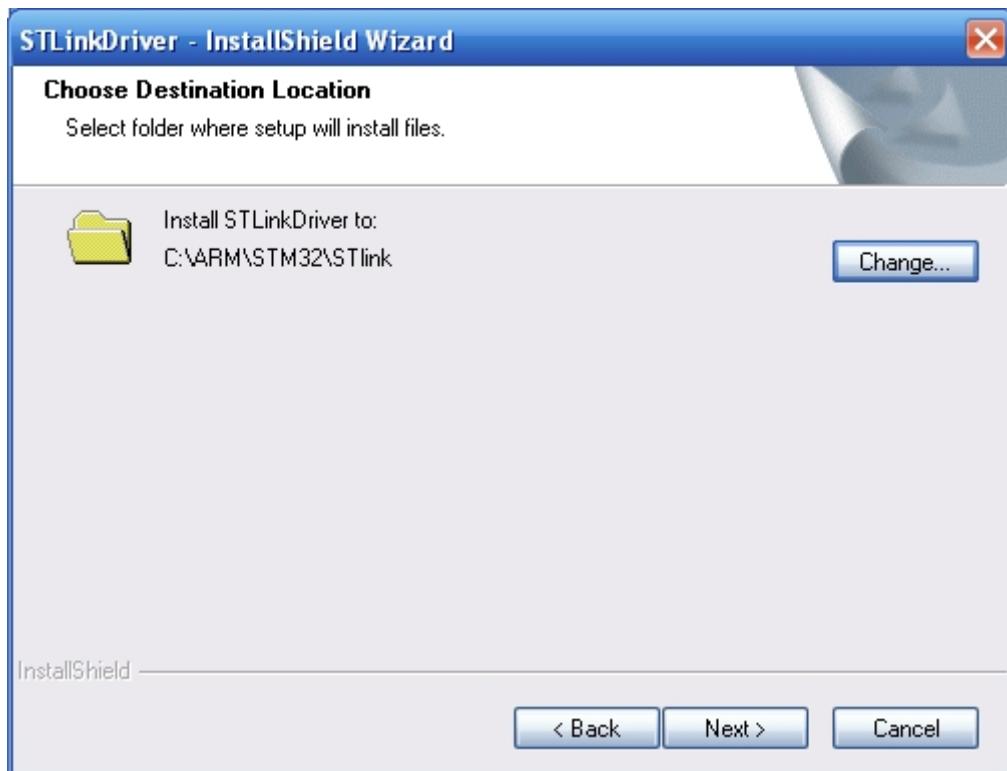


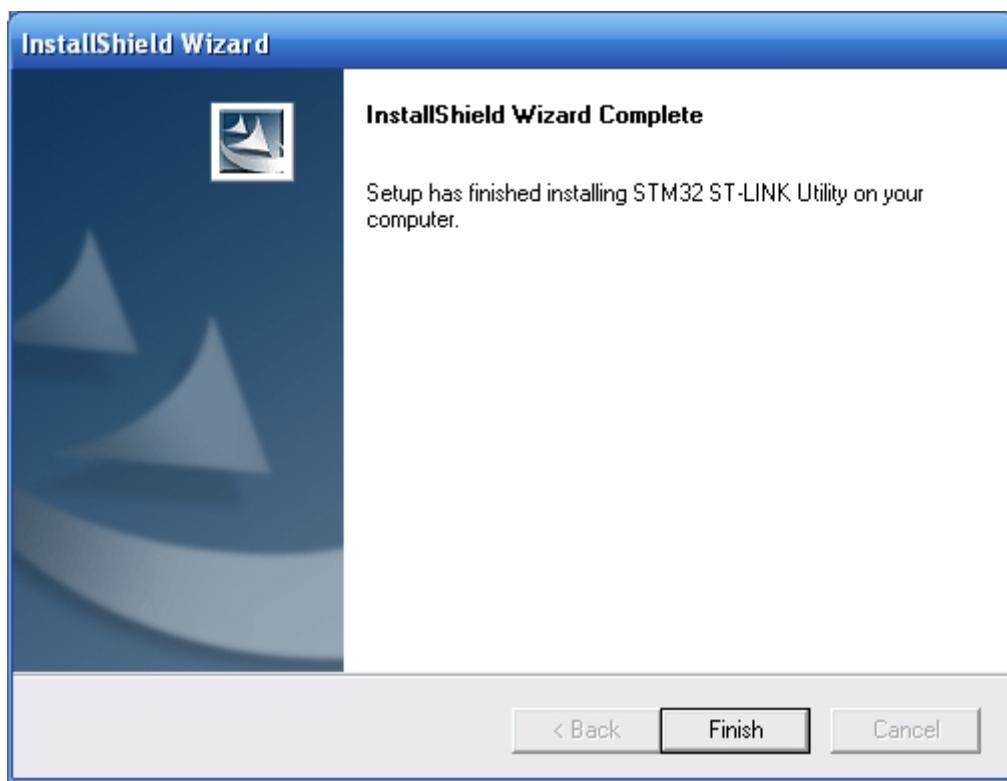
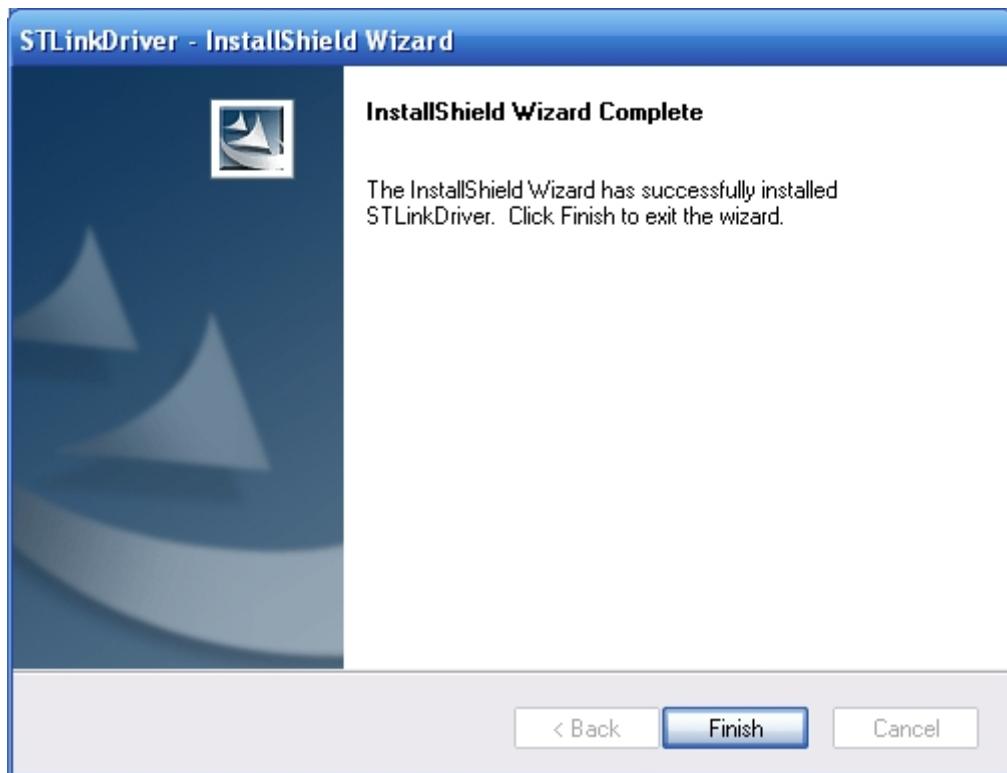




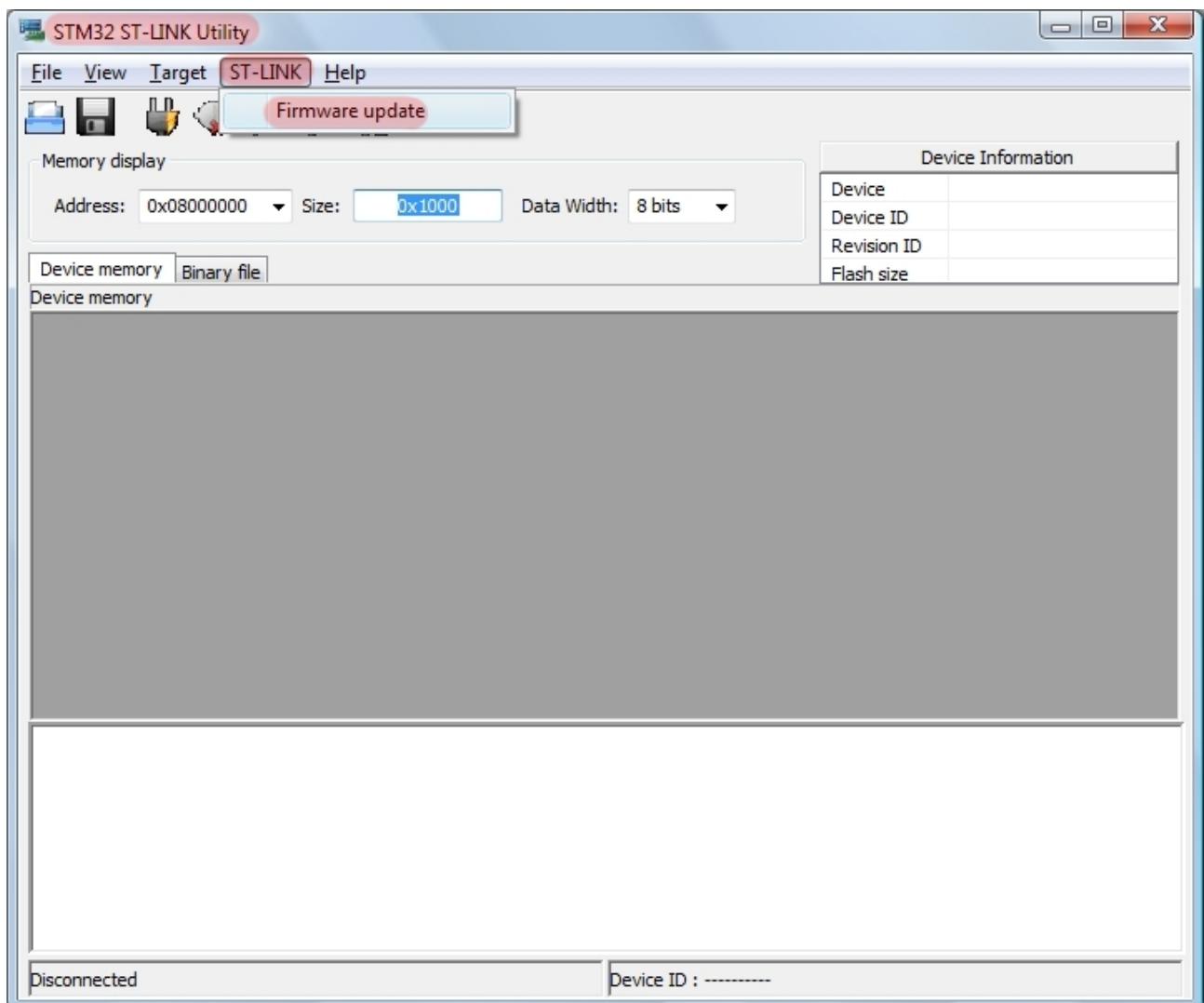
Пакет драйвера входит в пакет STSW-LINK004, и запускается автоматически



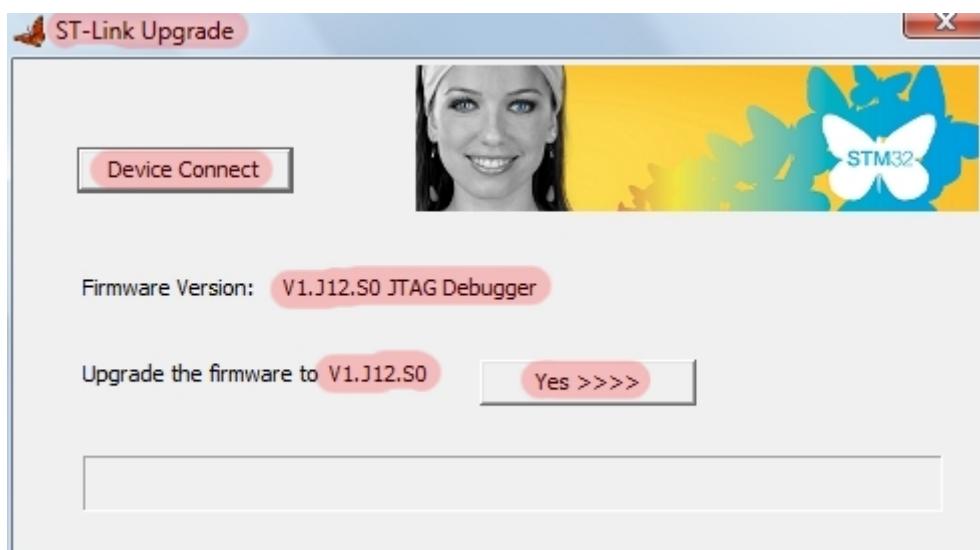




Подключаете отладочную плату, запускаете STlink10.1 Utility [Пуск](#) > [Программы](#) > [STMicroelectronics](#) > [STM32 ST-LINK Utility](#)



Запускаем **ST-LINK** > **Firmware update** > **Device Connect** и проверяем что прошивка STlink **10.1** не устарела (ПК должен быть подключен к Сети), при необходимости нажимаем **Yes** и обновляемся.



STlink gdbserver

Необходим для отладки через GDB[11.2](#).

10.2 JTAG

Глава 11

Отладка

11.1 STM32 SWD

11.2 GDB

OpenOCD

11.3 JTAG

Часть V

Основы языка C^{+^+}

Глава 12

Синтаксис

Глава 13

Типы данных

Глава 14

Стандартная библиотека libc

Часть VI

CMSIS

В этом разделе будут описаны только *некоторые* части библиотеки CMSIS^{VI} для STM32F, которые используются в коде ARMatura. Объяснить это очень просто — попытавшись включить в книгу полный листинг `stm32f4xx.h` получил 205 страниц листинга 😊

Глава 15

Startup: файлы стартового кода

`keil_startup_stm32f4xx.s`

Listing 15.1: /lib/STM32F4/keil_startup_stm32f4xx.s

```
***** (C) COPYRIGHT 2012 STMicroelectronics *****
;* File Name      : startup_stm32f4xx.s
;* Author         : MCD Application Team
;* Version        : V1.0.2
;* Date          : 05-March-2012
;* Description    : STM32F4xx devices vector table for MDK-ARM toolchain.

This module performs :
- Set the initial SP
- Set the initial PC == Reset_Handler
- Set the vector table entries with the exceptions ISR address
- Configure the system clock and the external SRAM mounted on
  STM324xG-EVAL board to be used as data memory (optional,
  to be enabled by user)
- Branches to __main in the C library (which eventually
  calls main()).
After Reset the CortexM4 processor is in Thread mode,
priority is Privileged, and the Stack is set to Main.
<<< Use Configuration Wizard in Context Menu >>>
***** (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at :
http://www.st.com/software-license-agreement-liberty-v2

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*****
```

```
; Amount of memory (in bytes) allocated for Stack
; Tailor this value to your application needs
; <h> Stack Configuration
;   <o> Stack Size (in Bytes) <0x0-0xFFFFFFFF:8>
;   </h>

Stack_Size      EQU      0x000000400

Stack_Mem        AREA     STACK, NOINIT, READWRITE, ALIGN=3
Stack_Size

--initial_sp

; <h> Heap Configuration
;   <o> Heap Size (in Bytes) <0x0-0xFFFFFFFF:8>
;   </h>

Heap_Size        EQU      0x000000200

Heap_Mem         AREA     HEAP, NOINIT, READWRITE, ALIGN=3
Heap_Size

--heap_base
--Heap_Mem
--heap_limit

PRESERVE
THUMB

; Vector Table Mapped to Address 0 at Reset
; AREA    RESET, DATA, READONLY
; EXPORT -- Vectors
; EXPORT -- Vectors_End
; EXPORT -- Vectors_Size
; DCD    --initial_sp
; Top of Stack
```

```

; Reset Handler          ; Reset Handler
DCD NMI_Handler          ; NMI Handler
DCD HardFault_Handler    ; Hard Fault Handler
DCD MemManage_Handler    ; MPU Fault Handler
DCD BusFault_Handler     ; Bus Fault Handler
DCD UsageFault_Handler   ; Usage Fault Handler
DCD Reserved             ; Reserved
DCD Reserved             ; Reserved
DCD Reserved             ; Reserved
DCD Reserved             ; Reserved
DCD SVC_Handler           ; SVCAll Handler
DCD DebugMon_Handler      ; Debug Monitor Handler
DCD Reserved             ; Reserved
DCD Reserved             ; Reserved
DCD PendSV_Handler        ; PendSV Handler
DCD SysTick_Handler       ; SysTick Handler

; Window WatchDog          ; Window WatchDog
DCD PVD_IRQHandler        ; PVD through EXTI Line detection
DCD TAMP_STAMP_IRQHandler ; Tamper and TimeStamps through the EXTI line
DCD RTC_WKUP_IRQHandler   ; RTC Wakeup through the EXTI line
DCD FLASH_IRQHandler       ; FLASH
DCD RCC_IRQHandler          ; RCC
DCD EXTI0_IRQHandler       ; EXTI Line0
DCD EXTI1_IRQHandler       ; EXTI Line1
DCD EXTI2_IRQHandler       ; EXTI Line2
DCD EXTI3_IRQHandler       ; EXTI Line3
DCD EXTI4_IRQHandler       ; EXTI Line4
DCD DMA1_Stream0_IRQHandler ; DMA1 Stream 0
DCD DMA1_Stream1_IRQHandler ; DMA1 Stream 1
DCD DMA1_Stream2_IRQHandler ; DMA1 Stream 2
DCD DMA1_Stream3_IRQHandler ; DMA1 Stream 3
DCD DMA1_Stream4_IRQHandler ; DMA1 Stream 4
DCD DMA1_Stream5_IRQHandler ; DMA1 Stream 5
DCD DMA1_Stream6_IRQHandler ; DMA1 Stream 6

; External Interrupts
DCD WWDG_IRQHandler        ; WWDG_IRQHandler
DCD PVD_IRQHandler          ; PVD_IRQHandler
DCD TAMP_STAMP_IRQHandler   ; TAMP_STAMP_IRQHandler
DCD RTC_WKUP_IRQHandler     ; RTC_WKUP_IRQHandler
DCD FLASH_IRQHandler         ; FLASH
DCD RCC_IRQHandler           ; RCC
DCD EXTI0_IRQHandler        ; EXTI0_IRQHandler
DCD EXTI1_IRQHandler        ; EXTI1_IRQHandler
DCD EXTI2_IRQHandler        ; EXTI2_IRQHandler
DCD EXTI3_IRQHandler        ; EXTI3_IRQHandler
DCD EXTI4_IRQHandler        ; EXTI4_IRQHandler
DCD DMA1_Stream0_IRQHandler ; DMA1_Stream0_IRQHandler
DCD DMA1_Stream1_IRQHandler ; DMA1_Stream1_IRQHandler
DCD DMA1_Stream2_IRQHandler ; DMA1_Stream2_IRQHandler
DCD DMA1_Stream3_IRQHandler ; DMA1_Stream3_IRQHandler
DCD DMA1_Stream4_IRQHandler ; DMA1_Stream4_IRQHandler
DCD DMA1_Stream5_IRQHandler ; DMA1_Stream5_IRQHandler
DCD DMA1_Stream6_IRQHandler ; DMA1_Stream6_IRQHandler

```

```

ADC ; ADC1, ADC2 and ADC3s
DCD ; CAN1_TX_IRQHandler
DCD ; CAN1_RX0_IRQHandler
DCD ; CAN1_RX1_IRQHandler
DCD ; CAN1_SCE_IRQHandler
EXTI9_5_IRQHandler ; External Line [9:5] s
DCD ; TIM1_BRK_TIM9_IRQHandler
DCD ; TIM1_UP_TIM10_IRQHandler
DCD ; TIM1_TRG_COM_TIM11_IRQHandler
DCD ; TIM1_CC_IRQHandler
DCD ; TIM2_IRQHandler
DCD ; TIM3_IRQHandler
DCD ; TIM4_IRQHandler
DCD ; I2C1_EV_IRQHandler
DCD ; I2C1_ER_IRQHandler
DCD ; I2C2_EV_IRQHandler
DCD ; I2C2_ER_IRQHandler
DCD ; SPI1_IRQHandler
DCD ; SPI2_IRQHandler
DCD ; USART1_IRQHandler
DCD ; USART2_IRQHandler
DCD ; USART3_IRQHandler
EXTI15_10_IRQHandler ; External Line [15:10] s
DCD ; RTC_Alarm_IRQHandler
DCD ; OTG_FS_WKUP_IRQHandler
DCD ; TIM8_BRK_TIM12_IRQHandler
DCD ; TIM8_UP_TIM13_IRQHandler
DCD ; TIM8_TRG_COM_TIM14_IRQHandler
DCD ; TIM8_CC_IRQHandler
DMA1_Stream7_IRQHandler ; DMA1 Stream7
FSMC ; FSMC_IRQHandler
SDIO ; SDIO_IRQHandler
TIM5_IRQHandler ; TIM5
SPI3_IRQHandler ; SPI3
UART4_IRQHandler ; UART4

```

```

; UART5
;   DCD          UART5_IRQHandler
;   DCD          TIM6_DAC_IRQHandler
;   DCD          TIM7_IRQHandler
;   DCD          DMA2_Stream0_IRQHandler
;   DCD          DMA2_Stream1_IRQHandler
;   DCD          DMA2_Stream2_IRQHandler
;   DCD          DMA2_Stream3_IRQHandler
;   DCD          DMA2_Stream4_IRQHandler
;   DCD          ETH_IRQHandler
;   DCD          ETH_WKUP_IRQHandler
;   DCD          CAN2_Tx_IRQHandler
;   DCD          CAN2_Rx0_IRQHandler
;   DCD          CAN2_Rx1_IRQHandler
;   DCD          CAN2_Sce_IRQHandler
;   DCD          OTG_FS_IRQHandler
;   DCD          DMA2_Stream5_IRQHandler
;   DCD          DMA2_Stream6_IRQHandler
;   DCD          DMA2_Stream7_IRQHandler
;   DCD          USART6_IRQHandler
;   DCD          I2C3_EV_IRQHandler
;   DCD          I2C3_ER_IRQHandler
;   DCD          OTG_HS_EP1_OUT_IRQHandler
;   DCD          OTG_HS_EP1_IN_IRQHandler
;   DCD          OTG_HS_WKUP_IRQHandler
;   DCD          OTG_HS_IRQHandler
;   DCD          DCMI_IRQHandler
;   DCD          CRYP_IRQHandler
;   DCD          HASH_RNG_IRQHandler
;   DCD          FPU_IRQHandler

--Vectors_Size    EQU    --Vectors_End - --Vectors

AREA    .text|, CODE, READONLY

```

```
; Reset handler
Reset_Handler PROC
    EXPORT Reset_Handler [WEAK]
    IMPORT SystemInit
    IMPORT __main
    LDR    RO, =SystemInit
    BLX    RO
    LDR    RO, =__main
    BX    RO
ENDP

; Dummy Exception Handlers (infinite loops which can be modified)

NMI_Handler PROC
    EXPORT NMI_Handler [WEAK]
    B
ENDP

HardFault_Handler \
PROC
    EXPORT HardFault_Handler [WEAK]
    B
ENDP

MemManage_Handler \
PROC
    EXPORT MemManage_Handler [WEAK]
    B
ENDP

BusFault_Handler \
PROC
    EXPORT BusFault_Handler [WEAK]
    B
ENDP

UsageFault_Handler \
PROC
    EXPORT UsageFault_Handler [WEAK]
    B
ENDP
```

```
PROC          EXPORT UsageFault_Handler    [WEAK]
B
ENDP
PROC          EXPORT SVC_Handler        [WEAK]
B
ENDP
SVC_Handler
DebugMon_Handler \
PROC          EXPORT DebugMon_Handler   [WEAK]
B
ENDP
PROC          EXPORT PendSV_Handler    [WEAK]
B
ENDP
PendSV_Handler
SysTick_Handler \
PROC          EXPORT SysTick_Handler    [WEAK]
B
ENDP
SysTick_Handler
Default_Handler \
PROC          EXPORT WWDG_IRQHandler   [WEAK]
EXPORT PVD_IRQHandler      [WEAK]
EXPORT TAMP_STAMP_IRQHandler [WEAK]
EXPORT RTC_WKUP_IRQHandler  [WEAK]
EXPORT FLASH_IRQHandler     [WEAK]
EXPORT RCC_IRQHandler       [WEAK]
EXPORT EXTI_IRQHandler      [WEAK]
EXPORT EXTIO_IRQHandler     [WEAK]
EXPORT EXTI1_IRQHandler    [WEAK]
EXPORT EXTI2_IRQHandler    [WEAK]
EXPORT EXTI3_IRQHandler    [WEAK]
EXPORT EXTI4_IRQHandler    [WEAK]
```

```
EXPORT DMA1_Stream0_IRQHandler [WEAK]
EXPORT DMA1_Stream1_IRQHandler [WEAK]
EXPORT DMA1_Stream2_IRQHandler [WEAK]
EXPORT DMA1_Stream3_IRQHandler [WEAK]
EXPORT DMA1_Stream4_IRQHandler [WEAK]
EXPORT DMA1_Stream5_IRQHandler [WEAK]
EXPORT DMA1_Stream6_IRQHandler [WEAK]
EXPORT ADC_IRQHandler [WEAK]
EXPORT CAN1_Tx_IRQHandler [WEAK]
EXPORT CAN1_Rx0_IRQHandler [WEAK]
EXPORT CAN1_Rx1_IRQHandler [WEAK]
EXPORT CAN1_Sce_IRQHandler [WEAK]
EXPORT EXTI9_5_IRQHandler [WEAK]
EXPORT TIM1_Brk_TIM9_IRQHandler [WEAK]
EXPORT TIM1_UP_TIM10_IRQHandler [WEAK]
EXPORT TIM1_Trig_Comp_TIM11_IRQHandler [WEAK]
EXPORT TIM1_CC_IRQHandler [WEAK]
EXPORT TIM2_IRQHandler [WEAK]
EXPORT TIM3_IRQHandler [WEAK]
EXPORT TIM4_IRQHandler [WEAK]
EXPORT I2C1_Ev_IRQHandler [WEAK]
EXPORT I2C1_Er_IRQHandler [WEAK]
EXPORT I2C2_Ev_IRQHandler [WEAK]
EXPORT I2C2_Er_IRQHandler [WEAK]
EXPORT SPI1_IRQHandler [WEAK]
EXPORT SPI2_IRQHandler [WEAK]
EXPORT USART1_IRQHandler [WEAK]
EXPORT USART2_IRQHandler [WEAK]
EXPORT USART3_IRQHandler [WEAK]
EXPORT EXTI15_10_IRQHandler [WEAK]
EXPORT RTC_Alarm_IRQHandler [WEAK]
EXPORT OTG_FS_Wkup_IRQHandler [WEAK]
EXPORT TIM8_Brk_TIM12_IRQHandler [WEAK]
EXPORT TIM8_UP_TIM13_IRQHandler [WEAK]
EXPORT TIM8_Trig_Comp_TIM14_IRQHandler [WEAK]
```

```
[WEAK]
EXPORT TIM8_CC_IRQHandler
EXPORT DMA1_Stream7_IRQHandler
EXPORT FSMC_IRQHandler
EXPORT SDIO_IRQHandler
EXPORT TIM5_IRQHandler
EXPORT SPI3_IRQHandler
EXPORT UART4_IRQHandler
EXPORT USART5_IRQHandler
EXPORT TIM6_DAC_IRQHandler
EXPORT TIM7_IRQHandler
EXPORT DMA2_Stream0_IRQHandler
EXPORT DMA2_Stream1_IRQHandler
EXPORT DMA2_Stream2_IRQHandler
EXPORT DMA2_Stream3_IRQHandler
EXPORT DMA2_Stream4_IRQHandler
EXPORT ETH_IRQHandler
EXPORT ETH_WKUP_IRQHandler
EXPORT CAN2_TX_IRQHandler
EXPORT CAN2_RXO_IRQHandler
EXPORT CAN2_RX1_IRQHandler
EXPORT CAN2_SCE_IRQHandler
EXPORT OTG_FS_IRQHandler
EXPORT DMA2_Stream5_IRQHandler
EXPORT DMA2_Stream6_IRQHandler
EXPORT DMA2_Stream7_IRQHandler
EXPORT USART6_IRQHandler
EXPORT I2C3_EV_IRQHandler
EXPORT I2C3_ER_IRQHandler
EXPORT OTG_HS_EP1_OUT_IRQHandler
EXPORT OTG_HS_EP1_IN_IRQHandler
EXPORT OTG_HS_WKUP_IRQHandler
EXPORT OTG_HS_IRQHandler
EXPORT DCM1_IRQHandler
EXPORT CRYP_IRQHandler
EXPORT HASH_RNG_IRQHandler
```

[WEAK]

EXPORT FPU_IRQHandler

WWDG_IRQHandler
PVD_IRQHandler
TAMP_STAMP_IRQHandler
RTC_WKUP_IRQHandler
FLASH_IRQHandler
RCC_IRQHandler
EXTIO_IRQHandler
EXTI1_IRQHandler
EXTI2_IRQHandler
EXTI3_IRQHandler
EXTI4_IRQHandler
DMA1_Stream0_IRQHandler
DMA1_Stream1_IRQHandler
DMA1_Stream2_IRQHandler
DMA1_Stream3_IRQHandler
DMA1_Stream4_IRQHandler
DMA1_Stream5_IRQHandler
DMA1_Stream6_IRQHandler
ADC_IRQHandler
CAN1_TX_IRQHandler
CAN1_RXO_IRQHandler
CAN1_RX1_IRQHandler
CAN1_SCE_IRQHandler
EXTI9_5_IRQHandler
TIM1_BRK_TIM9_IRQHandler
TIM1_UP_TIM10_IRQHandler
TIM1_TRG_COM_TIM11_IRQHandler
TIM1_CC_IRQHandler
TIM2_IRQHandler
TIM3_IRQHandler
TIM4_IRQHandler
I2C1_EV_IRQHandler
I2C1_ER_IRQHandler

```
I2C2_EV_IRQHandler
I2C2_ER_IRQHandler
SPI1_IRQHandler
SPI2_IRQHandler
USART1_IRQHandler
USART2_IRQHandler
USART3_IRQHandler
EXTI15_10_IRQHandler
RTC_Alarm_IRQHandler
OTG_FS_WKUP_IRQHandler
TIM8_BRK_TIM12_IRQHandler
TIM8_UP_TIM13_IRQHandler
TIM8_TRG_COM_TIM14_IRQHandler
TIM8_CC_IRQHandler
DMA1_Stream7_IRQHandler
FSMC_IRQHandler
SDIO_IRQHandler
TIM5_IRQHandler
SPI3_IRQHandler
UART4_IRQHandler
UART5_IRQHandler
TIM6_DAC_IRQHandler
TIM7_IRQHandler
DMA2_Stream0_IRQHandler
DMA2_Stream1_IRQHandler
DMA2_Stream2_IRQHandler
DMA2_Stream3_IRQHandler
DMA2_Stream4_IRQHandler
ETH_IRQHandler
ETH_WKUP_IRQHandler
CAN2_TX_IRQHandler
CAN2_RXO_IRQHandler
CAN2_RX1_IRQHandler
CAN2_SCE_IRQHandler
OTG_FS_IRQHandler
```

```
DMA2_Stream5_IRQHandler
DMA2_Stream6_IRQHandler
DMA2_Stream7_IRQHandler
USART6_IRQHandler
I2C3_EV_IRQHandler
I2C3_ER_IRQHandler
OTG_HS_EP1_OUT_IRQHandler
OTG_HS_EP1_IN_IRQHandler
OTG_HS_WKUP_IRQHandler
OTG_HS_IRQHandler
DCMI_IRQHandler
CRYPT_IRQHandler
HASH_RNG_IRQHandler
FPU_IRQHandler

B
.
ENDP

ALIGN

; *****
; User Stack and Heap initialization
; *****

IF :DEF :__MICROLIB
EXPORT __initial_sp
EXPORT __heap_base
EXPORT __heap_limit
ELSE
IMPORT __use_two_region_memory
EXPORT __user_initial_stackheap
```

```
-- user_initial_stackheap

    LDR    R0, = Heap_Mem
    LDR    R1, =(Stack_Mem + Stack_Size)
    LDR    R2, =(Heap_Mem + Heap_Size)
    LDR    R3, = Stack_Mem
    BX    LR

ALIGN
ENDIF
END

;***** (C) COPYRIGHT STMicroelectronics ***** END OF FILE*****
```

Глава 16

Стандартная библиотека STM32

Для работы необходимо скачать библиотеки поддержки от ST:

STSW-STM32078: STM32VLDISCOVERY1 firmware package (AN3268)

http://www.st.com/st-web-ui/static/active/en/st_prod_software_internet/resource/technical/software/firmware/stsw-stm32078.zip

STSW-STM32068: STM32VLDISCOVERY1 firmware package (AN3268)

http://www.st.com/st-web-ui/static/active/en/st_prod_software_internet/resource/technical/software/firmware/stsw-stm32068.zip

STSW-STM32065: STM32F4 DSP and standard peripherals library, including 82 examples

http://www.st.com/st-web-ui/static/active/en/st_prod_software_internet/resource/technical/software/firmware/stm32f4_dsp_stdperiph_lib.zip

STSW-STM32078 STM32VLDISCOVERY1 firmware package (AN3268)

STSW-STM32068 STM32F4DISCOVERY2 board firmware package, including 22 examples

STSW-STM32065 STM32F4 DSP and standard peripherals library, including 82 examples

16.1 stm32f4xx.h

Глава 17

USB client/host

Часть VII

Ядро Cortex-Mx

Глава 18

Режимы ARM и Thumb

Глава 19

DMA

Глава 20

DSP /Cortex-M3/

Глава 21

FPU /Cortex-M4F/

Часть VIII

Интерфейсы

Глава 22

USB

Глава 23

UART

Глава 24

SPI

Глава 25

I2C

Глава 26

CAN

Часть IX

Операционные системы ОСРВ

Глава 27

Keil RTX

Глава 28

FreeRTOS

Глава 29

eCos

Глава 30

Linux

подробно рассмотрен в отдельном разделе XII

Часть X

Стек TCP/IP

Глава 31

Ethernet

Глава 32

PPP

Часть XI

Типовые применения

Глава 33

GPS

33.1 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

33.2 \$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)

33.3 Системы координат (датум)

<http://ne-grusti.narod.ru/Glossary/datums.html>

Системы координат (datums) можно разделить на геоцентрические и топоцентрические.

В геоцентрической системе размеры эллипсоида, ориентация и положение его центра выбираются следующим образом:

- объем эллипсоида предполагается равным объему геоида;
- большая полуось эллипсоида лежит в плоскости экватора геоида;
- малая полуось направлена по оси вращения Земли;
- среднеквадратичное отклонение поверхности эллипсоида от поверхности геоида
- минимально по всей территории земного шара.

WGS⁷² и сменившая ее *WGS⁸⁴*, а также российская *SGS⁸⁵* являются геоцентрическими системами координат на эллипсоидах *WGS72*, *GRS80* и *SGS85* соответственно. В системе GPS/NAVSTAR используется *WGS⁸⁴*, а в системе GLONASS — *SGS⁸⁵*.

Топоцентрическая (национальная) система координат появляется так: вы берете некоторый эллипсоид и располагаете его таким образом, чтобы для заданной территории среднеквадратичное отклонение поверхности эллипсоида от поверхности геоида было минимальным. При этом остальная часть мира вас не интересует: отклонения на другой стороне Земли может быть сколь угодно велико.

В России используются несколько геодезических систем координат: Пулково¹⁹⁴² Пулково 1942 г., 1963 г. и 1991 г. Система координат 1963 г. используется военными и ее параметры преобразования засекречены. Обычно мы пользуемся картами, составленными в системе Пулково¹⁹⁴². Она базируется на эллипсоиде Красовского.

Параметры преобразования для Пулково¹⁹⁴²:

Преобразование Bursa-Wolf (Position Vector Transformation)

<http://ne-grusti.narod.ru/Glossary/transformations.html#pvt>

направление	dX	dY	dZ	rX	rY	rZ	M	источник
<i>WGS84</i> → 1942	-27.0	+135.0	+84.5	0.0	0.0	0.554	-0.2263	Data+
1942 → <i>WGS84</i>	+25.0	-141.0	-78.5	0.0	0.35	0.736	0.0	Stefan A. Voser

Преобразование Молоденского

<http://ne-grusti.narod.ru/Glossary/transformations.html#molodensky>

направление	dX	dY	dZ	da	df	источник
1942 → <i>WGS84</i>	+28.0	-130.0	-95.0	-108.0	+0.00480795	Vladimir Zh. Boston-PC Forum
1942 → <i>WGS84</i>	+28.0	-130.0	-95.0	-108.0	+0.00480795	MADTRAN, Peter H. Dana
1942 → <i>WGS84</i>	+24.0	-123.0	-94.0	-108.0	+0.00480795	Stefan A. Voser

33.4 Методы преобразования систем координат

<http://ne-grusti.narod.ru/Glossary/transformations.html>

Часто требуется перейти от одной системы координат (datum) к другой. Например, вы хотите данные с GPS в системе координат WGS^{84} наложить на карту в системе координат Пулково¹⁹⁴².

Произодить преобразование проще (математически), если сначала перевести координаты из географических (широта и долгота) в прямоугольные (XYZ).

Чтобы получить точные прямоугольные координаты точки на поверхности Земли, необходимо знать не только широту и долготу, но и ее высоту над поверхностью эллипсоида. GPS показывает высоту над эллипсоидом WGS^{84} . Можно вычислить прямоугольные координаты только по широте и долготе, считая, что точка лежит на поверхности эллипса, но при этом точность понизится.

Высоту точки на другими эллипсоядами получить сложнее. Обычно мы знаем высоту в национальной системе высот. За нулевую высоту, как правило, принимается среднее значение уровня моря в определенной точке побережья по результатам многолетних наблюдений. Высоты в этой системе измеряются геодезическими методами относительно поверхности геоида. Поэтому, чтобы узнать высоту точки над эллипсоядом, нужно знать возвышение геоида над эллипсоядом в данном месте, которое трудно вычислить с хорошей точностью. Существуют различные математические модели геоида для разных территорий и для всего земного шара, которые постоянно уточняются.

На заре спутниковой геодезии, когда взаимосвязи между системами координат не были четко определены, а данные исследований были не очень точны, применяли простой сдвиг начала координат dX , dY , dZ для перехода от одной системы координат к другой. Это предполагало, что направления осей двух эллипсоядов параллельны (что во многих случаях не соответствует действительности). Для работ на небольшой территории погрешности, вносимые этим предположением, были меньше, чем точность самих данных. Однако, по мере накопления и уточнения данных и повышения точности измерений, стало очевидно, что преобразование по трем параметрам не подходит для больших территорий и глобального использования, если требуется максимальная точность и единый набор параметров преобразования.

Простейший метод — сдвиг центра координат прямоугольной системы, предполагая, что оси исходной и целевой систем координат параллельны.

$$\begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix} = \begin{pmatrix} X_A \\ Y_A \\ Z_A \end{pmatrix} + \begin{pmatrix} dX \\ dY \\ dZ \end{pmatrix} \quad (33.1)$$

Молоденский разработал формулы для применения параметров сдвига к географическим координатам.

$$\Delta\varphi'' = \frac{206265}{\rho} (-dX \sin \varphi \cos \lambda - dY \sin \varphi \sin \lambda + dZ \cos \varphi + [a\Delta f + f\Delta a] \sin 2\varphi) \quad (33.2)$$

$$\Delta\lambda'' = \frac{206265}{\nabla \cos \varphi} (-dX \sin \lambda + dY \cos \lambda) \quad (33.3)$$

$$\Delta h = dX \cos \varphi \cos \lambda + dY \cos \varphi \sin \lambda + dZ \sin \varphi + (a\Delta f + f\Delta a) \sin^2 \varphi - \Delta a \quad (33.4)$$

здесь dX , dY , dZ — сдвиг по осям, м; a и f — большая полуось и сжатие исходного эллипса; da и df — разности между большой полуосью и сжатием исходного эллипса и целевого эллипса.

Повышенная точность достигается преобразованием Хелмерта с семью параметрами. Есть две его разновидности, различающиеся присвоением знака для параметров поворота.

1. Position Vector Transformation (Bursa-Wolf)

$$\begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix} = M \cdot \begin{pmatrix} 1 & -Rz & +Ry \\ +Rz & 1 & -Rx \\ -Ry & +Rx & 1 \end{pmatrix} \begin{pmatrix} X_A \\ Y_A \\ Z_A \end{pmatrix} + \begin{pmatrix} dX \\ dY \\ dZ \end{pmatrix} \quad (33.5)$$

2. Coordinate Frame Transformation

$$\begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix} = M \cdot \begin{pmatrix} 1 & +Rz & -Ry \\ -Rz & 1 & +Rx \\ +Ry & -Rx & 1 \end{pmatrix} \begin{pmatrix} X_A \\ Y_A \\ Z_A \end{pmatrix} + \begin{pmatrix} dX \\ dY \\ dZ \end{pmatrix} \quad (33.6)$$

Здесь M — это масштаб (scale), параметры берутся из описания системы координат.

33.5 Геоид и эллипсоиды

<http://ne-grusti.narod.ru/Glossary/ellipsoid-geoid.html>

Геоид — фигура сложной формы, образованная поверхностью уровня вод Мирового океана, продолженной под материками. Эта поверхность во всех точках перпендикулярна (нормальна) вектору силы тяжести. Отвес направлен перпендикулярно поверхности геоида, а не к центру Земли! Это связано с тем, что плотность Земли распределена неравномерно.

Эллипсоид — тело, полученное вращением эллипса вокруг его малой оси. Размеры подбирают так, чтобы среднеквадратичное отклонение от поверхности геоида было минимально либо по всей поверхности Земли, либо для заданной территории.

Параметры некоторых эллипсоидов

эллипсоид	использование	большая полуось a, м	малая полуось b, м	сжатие $f = (a - b)/a$
Красовского (1940)	Россия и др. Пулково ¹⁹⁴²	6378245	6356863	1/298.3
GRS80	международный <i>WGS⁸⁴</i>	6378137	6356752.31425	1/298.25722356
SGS85	ГЛОНАСС SGS85			
Бесселя	СССР до 1942 г.			

В таблицах эллипсоидов часто указывается не полярное сжатие f , а обратная величина $1/f$, например, для эллипсоида Красовского $1/f = 298.3$.

Отклонения эллипсоида Красовского от геоида на территории СНГ не превышают 150 м.

33.6 Tistar15



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

33.7 WISMO228



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

Глава 34

GSM

34.1 WISMO228



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

Глава 35

шина Dallas 1Wire

35.1 RTC

35.2 Датчики температуры DS18x20

Часть XII

Встраиваемый Linux

Часть XIII

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 XIV

Приложения

Глава 36

Сводная таблица процессоров

[STM32F] > core > crypto > model > pins > flash > package > n]

core

- 1 Cortex-M3
- 3 Cortex-M4
- 4 Cortex-M4F с расширениями FPU (плавающая точка) и SIMD
для приложений цифровой обработки сигналов

crypto

- 0 Cortex-M3 только CRC
- 1 Crypto расширение с функционалом вычислений MD5, SHA,..
- 2 Cortex-M4 CRC
- 3 Crypto

model

- 0
- 3 включена поддержка CAN, USB, SDIO
- 7

package

- T LQFP

pins

- C 48
- R 64
- V 100
- I 176

flash

- 8 64K
- B 128K
- G 1M
- I 2M

	ядро Cortex-	MHz	Flash	SRAM	корпус LQFP	GPIO	ST family
STM32F100C4T6B	M3	24	16K	4K	48		LD
STM32F100RBT	M3	24	128K	8K	100		MD
STM32F103C8T	M3	72	64K	20K	48	37	MD
STM32F103CBT	M3	72	128K	20K	48	37	MD
STM32F103RBT	M3	72	128K	20K	64	51	MD
STM32F103VBT	M3	72	128K	20K	100	80	MD
STM32F407VGT	M4F	168	1M	192K	144		
STM32F407IGT	M4F	168	1M	192K	176		
STM32F427IIT	M4F	168	2M	256K	176		
	CAN	USB	UART	I2C	SPI	ADC	DAC
STM32F100C4T6B	○	○	2		1		
STM32F100RBT	○	○	1				
STM32F103C8T	●	●	3	2	2	2x12b (10ch)	
STM32F103C8T	●	●	3	2	2	2x12b (10ch)	
STM32F103RBT	●	●	3	2	2	2x12b (16ch)	
STM32F103VBT	●	●	3	2	2	2x12b (16ch)	
STM32F407VGT	2	2	6				
STM32F407IGT	2	2	6				
STM32F427IIT	2	2	8				
	таймеры IC/OC/PWM	PWM таймер	DMA	WDT	SysTick 24b	RTC	
STM32F100C4T6B							
STM32F100RBT							
STM32F103C8T	3x16b	1x16b	7ch	2	●	●	
STM32F103C8T	3x16b	1x16b	7ch	2	●	●	
STM32F103RBT	3x16b	1x16b	7ch	2	●	●	
STM32F103VBT	3x16b	1x16b	7ch	2	●	●	
STM32F407VGT					●	●	
STM32F407IGT					●	●	
STM32F427IIT					●	●	