## Идеология mxsrclib

Библиотека mxsrclib должна компилироваться компилятором C++. Это общий принцип.

В корневой папке mxsrclib содержится архитектуро-независимые модули. Т. е. те которые компилируются без проблем на любой архитектуре.

Для того чтобы учесть особенности компиляторов и операционных систем была создана папка arch – архитектура, в которой содержится то, что характерно только для конкретной архитектуры. В ней содержатся подпапки, которые соответствуют каждой архитектуре со своим набором модулей. В каждой папке могут содержатся модули с одинаковыми названиями для всех архитектур. Эти модули, в общем случае, являются мостами из архитектуро-независимой части в архитектуро-зависимую. Также здесь могут содержаться специфические модули, которые используются этими мостами.

## Время компиляции

5-01 – Pentium 4; 3 ГГц; ОЗУ 2 Гб

5-04 – Athlon 64 X2; 4400+ (2,3 ГГц); ОЗУ 2,5 Гб

5-07 – Intel Core 2 Duo; E6550 2.33ГГц; ОЗУ 1Гб

Компиляция mxsrclib

IAR AVR 5.30; Ревизия mxsrclib 280; Оптимизация выключена

5-01 (Крашенинников) 00:03:34

5-04 (Лящов) 00:01:40

5-07 (Сергеев) 00:01:11

## Номера файлов

Внутри соответствующего файла пишется:

#define <имя файла вместе с расширением>\_ IDX <номер файла по таблице>

|  |  |
| --- | --- |
| № | Файл |
| 1 | mxdata.cpp |
| 2 | mxdata.h |
| 3 | irsdbgutil.cpp |
| 4 | irsdbgutil.h |
| 5 | irserror.cpp |
| 6 | irserror.h |
| 7 | mxnet.cpp |
| 8 | mxnet.h |
| 9 | irsalg.cpp |
| 10 | irsalg.h |
| 11 | irsfunnel.cpp |
| 12 | irsfunnel.h |
| 13 | irsfunnel.cpp |
| 14 | irsfunnel.h |
| 15 | irsstd.cpp |
| 16 | irsstd.h |
| 17 | mxnetc.cpp |
| 18 | mxnetc.h |
| 19 | irseeprom.cpp |
| 20 | irseeprom.h |

**class irs::pid\_regulator\_t**

sensor\_filter\_unit

sensor\_isodr\_unit

ref\_rate\_limiter\_unit

sync\_gains

dead\_band

prop\_gain

integ\_gain·Σe

deriv\_gain·Δe

## Сравнение компиляторов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Параметр | Builder | IAR AVR | Watcom | GCC PC | GCC ARM | GCC AVR32 | GCC NIOS |
| Версия | 2006 | 5.30 | 1.8 | 4.3.2 | 2.95 |  |  |
| Размер wchar\_t | 2 |  |  | 4 |  |  |  |
| Размер int | 4 | 2 | 4 | 4 | 4 | 4 | 4 |
| Знак.  char | знак. | беззн. (опция) | беззн. (опция) | знак. |  |  |  |
| Знак. wchar\_t | беззн. |  |  | знак. |  |  |  |
| string | шаблон | char | шаблон | шаблон |  | шаблон | шаблон |
| limits | есть | нет | есть | есть |  | есть | есть |
| stream | шаблон | char нов. | char стар. | шаблон | char стар. | шаблон | шаблон |
| complex | шаблон | не шаблон | шаблон | шаблон |  | шаблон | шаблон |

## Описание mx\_proc\_t

Класс mx\_proc\_t является абстрактным базовым классом для всех классов для длительной обработки. В нем содержится метод tick, предназначенный для вызова в цикле. Пока он возвращает истину, цикл выполняется. Как только выполнение завершено он возвращает ложь. Прервать выполнение можно методом abort. После этого следует дождаться, пока tick начнет возвращать ложь.

Так как mx\_proc\_t является абстрактным классом, то от него необходимо порождать потомков. Потомки будут работать через его интерфейс. Также можно порождать потомков со своими входными параметрами через оператор (). Сначала должен быть порожден абстрактный класс с определенный набором входных параметров. Потом от него уже порождаются классы которые можно создать.

**irs::handle\_t**

#include <mxdata.h>

Тип **irs::handle\_t** является умным указателем. Он управляет памятью подключенного к нему указателя на динамический выделенный объект с помощью подсчета ссылок. Ссылки подсчитываются при копировании объектов типа **irs::handle\_t**. При уничтожении последнего объекта типа **irs::handle\_t** уничтожается и подключенный к нему объект.

## Замечания по написанию программ

* Текст исходного кода не должен переходить за границу 80 символов. В случае если строка кода длиннее, то выполнять переход на следующую строку, с выполнением одинарного отступа. Последующие строки выравнивать по предыдущей строке. Отступы должны равняться 2 символам. И среда разработки должна быть настроена на то, чтобы выполнять отступы пробелами. Отступы должны правильно выполняться во всем исходном коде.
* В пустой реализации функции имя параметра не писать (или писать в виде комментария):

**void** get\_voltage(**double**\* /\*voltage\*/)

{

}

* При использовании класса ostrstream, после вызова функции str(), необходимо вызвать функцию freeze(**false**) буфера потока (не самого потока!):

irs::string f()

{

ostrstream strm;

strm << “ok”;

irs::string strg(strm.str(), strm.pcount());

strm.rdbuf()->freeze(**false**);

return strg;

}

* Подключать стандартную библиотеку C++ необходимо ТОЛЬКО с помощью **#include <irscpp.h>**. Квалификацию имен стандартной библиотек и с помощью **std::** не применять. Например:

cout << “example” << endl;

Вместо:

std::cout << “example” << std::endl;

* В аппаратно-независимом коде использование исключений запрещено. Вместо этого используем класс **irs::error\_trans\_base\_t**.
* Потоки ввода/вывода использовать нельзя. Можно использовать только потоки либо ввода, либо вывода. Например, можно использовать ifstream и ofstream, но нельзя fstream.
* Имена классов, структур, перечислений и **typedef**-определения вне класса писать с суффиксом **\_t**.
* Имена **typedef**-определений в **namespace**, в случае если оно похоже или совпадает с библиотечными, снабжать суффиксом **ns** и после него **\_t**. Например:

**typedef** size\_t sizens\_t;

* **typedef** для типа внутри класса писать с суффиксом **\_type**. Например:

**class** my\_data\_t

{

**public**:

**typedef** **int** value\_type;

<…>

**private**:

<…>

};

* Имена переменных членов класса писать с префиксом **m\_**, а имена переменных-указателей членов класса писать с префиксом **mp\_**.
* Имена параметров функций членов класса (методов) писать с префиксом **a\_**, а имена указателей параметров функций членов класса (методов) писать с префиксом **ap\_**.
* Не использовать **static const** для инициализации целочисленных констант непосредственно в описании класса. Вместо этого использовать неименованный **enum**.
* Константы **M\_PI** и другие **M\_** не использовать. Вместо этого брать **IRS\_PI** и **IRS\_** из файла **irsdefs.h**.
* Все семейство файлов **irsstd** следует включать с помощью #include <irsstd.h>. Включение #include <irsstdl.h> – НЕДОПУСТИМО!
* Ключевое слово **virtual** необходимо употреблять в описании виртуальных функций производного класса, которые переопределяют виртуальные функции базового. Несмотря на то, что компиляторы этого не требуют.
* При необходимости введения статической переменной в inline-функции, функция преобразуется в не-inline и размещается не в h-файле, а в cpp-файле. В h-файле помещается только ее описание.
* Переменные в C++ Builder типа **Variant** необходимо преобразовывать в **AnsiString** следующим образом:

AnsiString ansi\_str = **static\_cast**<**wchar\_t**\*>(variant\_variable);

* В соответствии со стандартом C++ (ISO/IEC 14882:2003(E) – 14.6.2) при наследовании классов-шаблонов от классов-шаблонов, если базовый класс зависит от параметров производного при наследовании, имена базового класса недоступны через производный класс:

**struct** A {

**struct** B { /\* ... \*/ };

**int** a;

**int** Y;

**int** z;

};

int a;

**template**<**class** T> **struct** Y: T {

**struct** B { /\* ... \*/ };

B b; // *The* B *defined in* Y

**void** f(**int** i) { a = i; } // ::a не T::a

Y\* p; // Y<T>

};

Y<A> ya;

**int** d = ya.z; // Ошибка в ya нет z

**int** r = ya.A::z; //правильно!

* В начале всех файлов h- и cpp-файлах обязательно создавать шапку (комментарии с помощью символов «//»):
* Одна или несколько строк описание (или название) на русском
* Дата последней модификации «// Дата: <Дата>». Здесь <Дата> заменить датой последней модификации. Например, «// Дата: 15.03.2010».
* Дата создания «// Дата создания: <Дата>». Здесь формат ввода <Дата> аналогично предыдущему пункту
* Пустая строка
* Относительно #include-определений во всех h- и cpp-файлах вводятся следующие правила:
* В h-файле #include <irsdefs.h>, в cpp-файле #include <irspch.h> (В главный cpp-файл в проекте C++ Builder #include <irspch.h> вставлять не надо, а оставить #include <vcl.h>)
* Только для cpp-файла:
* #ifdef \_\_BORLANDC\_\_
* #pragma hdrstop
* #endif // \_\_BORLANDC\_\_
* Пустая строка
* Аппаратно-зависимые #include-определения (WinAPI, Linux API, #include-определения микроконтроллеров и др.)
* Пустая строка
* Аппаратно-независимые стандартные #include-определения – стандартная библиотека C и C++
* Пустая строка
* #include-определения библиотеки mxsrclib
* Пустая строка (если есть следующий пункт)
* #include-определения проекта (если есть)
* Пустая строка
* #include <irsfinal.h>
* Пустая строка

Пример оформления cpp-файлов:

#include <irspch.h>

#ifdef \_\_BORLANDC\_\_

#pragma hdrstop

#endif // \_\_BORLANDC\_\_

#ifdef IRS\_STM32\_F2\_F4\_F7

# include <armcfg.h>

#endif // IRS\_STM32\_F2\_F4\_F7

#include <irsgpio.h>

#include <irserror.h>

#include <irsfinal.h>

Пример оформления h-файлов:

#include <irsdefs.h>

#ifdef \_\_BORLANDC\_\_

#include <SysUtils.hpp>

#include <System.hpp>

#include <vcl.h>

#endif //\_\_BORLANDC\_\_

#include <irscpp.h>

#include <irslocale.h>

#include <irsfinal.h>

* Если класс обращается к подклассу, содержащему функцию tick, то этот класс должен вызывать функцию tick подкласса в своей функции tick

## Обновление библиотеки

Если старая папка **mxsrclib** существует, то она удаляется. Вместо нее копируется новая папка.

Если папка **mxsrclib** не существует, то она создается на одном уровне с папкой проекта. Например, если папка проекта называется **my\_project**, то расположение должно быть таким:

<папка верхнего уровня>

mxsrclib

<файлы библиотеки mxsrclib>

my\_project

<файлы проекта my\_project>

## Настройка VisualSVN Server

Для того чтобы VisualSVN Server работал с Tortoise SVN необходимо в файле "<Папка Program Files> \VisualSVN Server\conf\httpd.conf" заменить строку:

SSLProtocol -ALL +SSLv3 +TLSv1

На строки:

#SSLProtocol -ALL +SSLv3 +TLSv1

SSLProtocol -ALL +SSLv3

Причем под Windows 7 или Vista с включенным UAC текстовый редактор, которым производится это изменение, должен быть запущен от имени администратора через правую кнопку мыши. Даже если вы уже находитесь под учетной записью администратора. Иначе Windows создает для редактора свою копию файла и его правит, а реальный файл остается неисправленным.

## Параметры Tortoise SVN

TortoiseSVN->Settings

* General
  + Global ignore pattern \*.o \*.lo \*.la \*.al .libs \*.so \*.so.[0-9]\* \*.a \*.pyc \*.pyo \_\_pycache\_\_ \*.rej \*~ #\*# .#\* .\*.swp .DS\_Store \*.obj \*.tds \*.~\* \*.il? \*.bak \*.\*~ \*.dcu ~W\*.tmp ~$\*.\* \*.map \*.pch irsconfig.h lsprst7.\* sysprs7.\* tmpprst.\* \*.tmp \*.csv irsconfig.h \*.r90 \*.s90 \*.pbi \*.pbd \*.SchDocPreview \*.dep

## Список игнорируемых SVN для проектов Quartus

В проектах Quartus следует добавлять список игнорируемых из файла: "mxsrclib\utils\info\svn quartus ignore.txt"

Делается это так:

«По правой кнопке мыши» → TortoiseSVN→Properties→Edit→Load→ «Выбор файла указанного выше»→«Подтвердить»

## Модули, потребляющие память даже когда они не используются:

avr\RTL8019AS = 42 byte

avr\UDP\_stack = 72 byte;

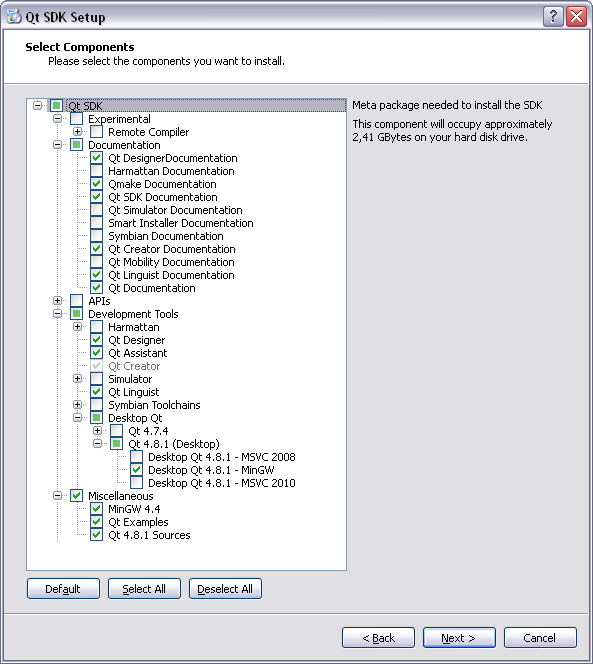
avr\counter = 12 byte;

avr\irsarchint = 21 byte;

avr\mxifa = 51 byte.

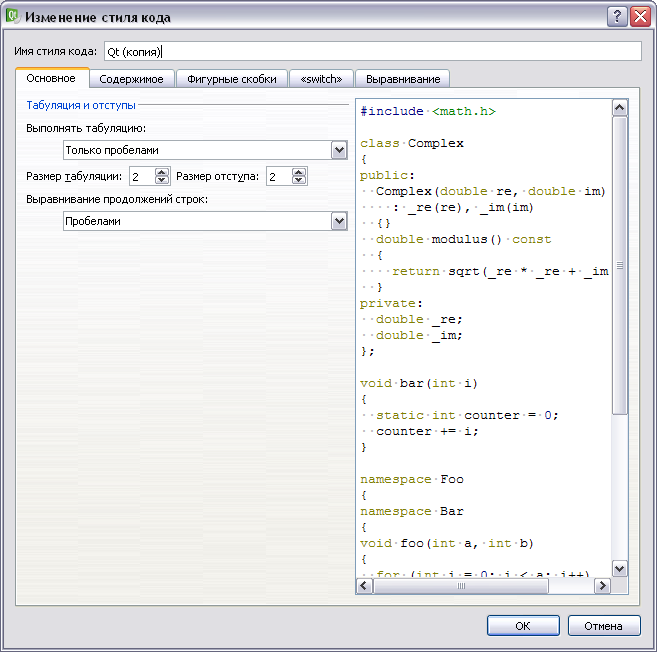
## Установка Qt SDK 1.2.1

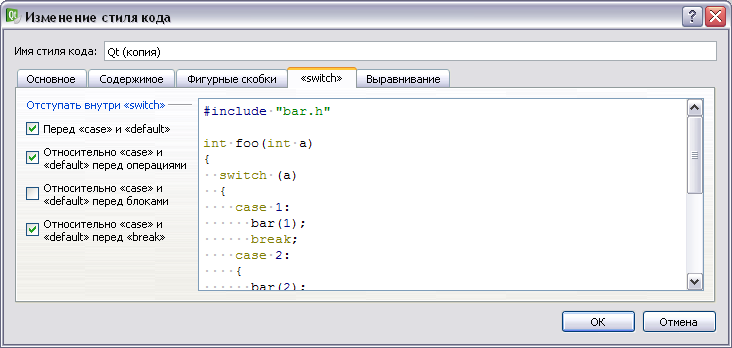
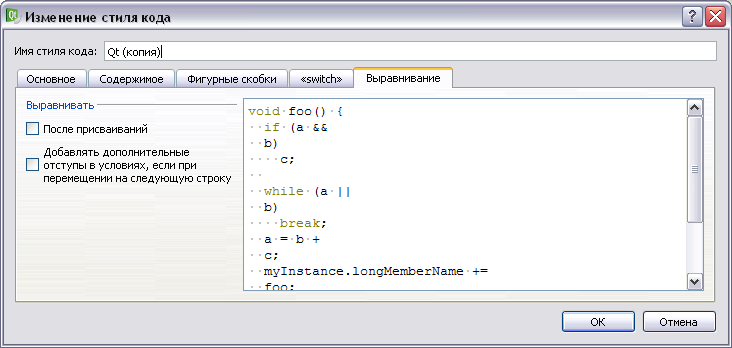
Рекомендуемые параметры:



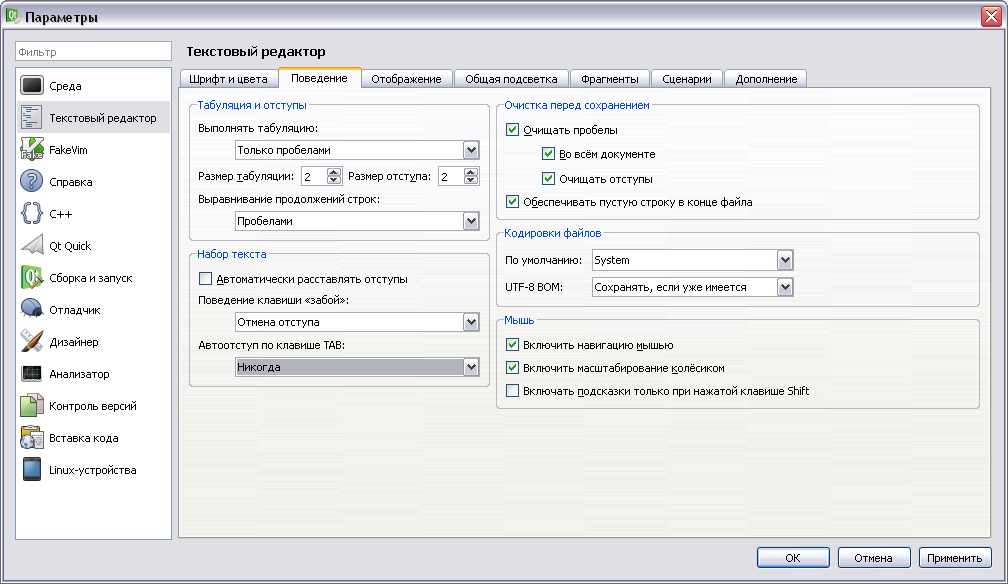
## Параметры проекта для Qt Creator

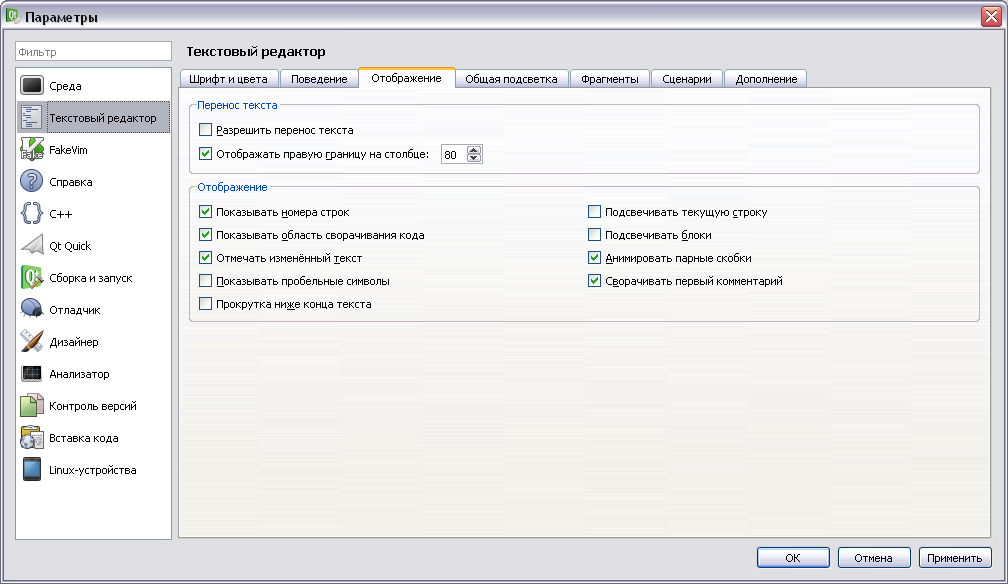
* Среда
  + Клавиатура
    - ProjectExplorer->Rebuild – Установить Ctrl+Shift+B
    - Debugger->Abort – Установить Ctrl+F2
* Отладчик
  + Основное
    - Переключится в предыдущий... – Установить
    - Сопоставление каталогов исходников->Добавить исходники Qt – Указать папку (Обычно QtSDK\QtSources. Пример полного пути: C:\Programs\QtSDK\QtSources)
  + GDB
    - ...qWarning – Снять
    - ...qFatal – Установить
* C++
  + Стиль кода (Копировать настройки Qt и нажать «Изменить»)



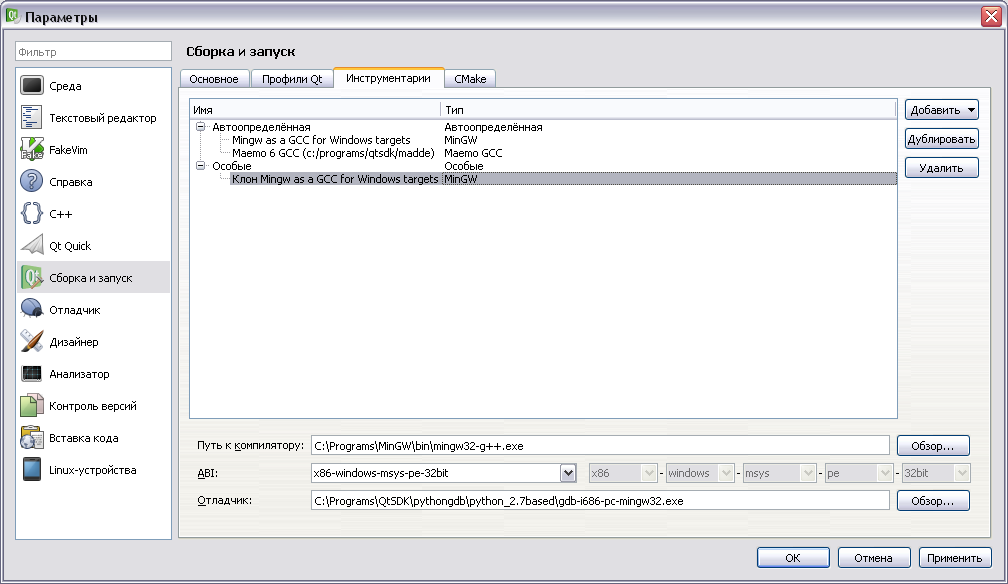
 

* Текстовый редактор





* Сборка и запуск.
  + Основное
    - Сохранять все файлы перед сборокой – Установить
    - Показывать вывод компилятора при сборке – Установить
    - Показывать вывод приложения при запуске - Снять
  + Инстументарии
    - MinGW должен быть поставлен отдельно по ссылке с его сайта (Есть также на Box). Чтобы подключить его к Qt необходимо выбрать конфигурацию Автоопределенная->MinGW…. Нажать кнопку «Дублировать».
    - В дублированной конфигурации указать путь к установленному MinGW.



* Редактор
  + В файле проекта нужно добавить:

INCLUDEPATH += ../mxsrclib/

INCLUDEPATH += ../mxsrclib/arch/qt/

* + - Для MINGW

QMAKE\_CXXFLAGS += -finput-charset=CP1251

QMAKE\_CXXFLAGS += -fno-strict-aliasing

QMAKE\_CXXFLAGS += -Wno-deprecated

QMAKE\_CXXFLAGS += -Wextra

QMAKE\_CXXFLAGS += -Woverloaded-virtual

QMAKE\_CXXFLAGS += -Wno-unused-parameter

* + - Для MSVC2008/MSVC2010

QMAKE\_CXXFLAGS -= -Zc:wchar\_t-

QMAKE\_CXXFLAGS -= -w34100

QMAKE\_CXXFLAGS += -wd4800

QMAKE\_CXXFLAGS += -w44355

QMAKE\_CXXFLAGS += -wd4996

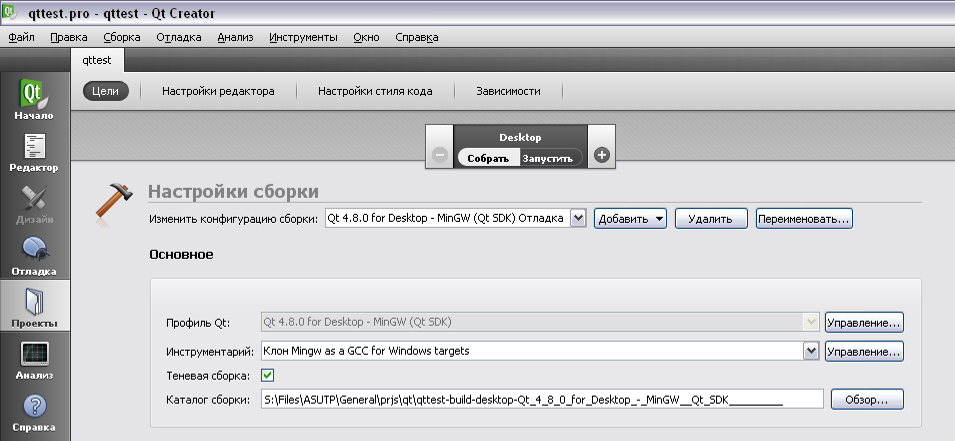
LIBS += -lws2\_32

LIBS += -lwinmm

* + В файле main.cpp первой строкой после фигурной скобки нужно добавить:

QTextCodec::setCodecForCStrings(QTextCodec::codecForName("UTF-8"));

* Проекты.
  + Когда проект загружен в интерфейсе нужно выбрать «Проекты». В группе «Основное», в выпадающем списке «Инструментарий» нужно выбрать вновь созданную конфигурацию «Клон MinGW…».



## Подготовка программы, написанной на Qt, для запуска вне среды Qt

Для запуска программы на Qt вне среды Qt в папку программы следует добавить следующие dll-файлы:

* Из папки MinGW\bin  
  (Пример полного пути: C:\Programs\MinGW\bin)
  + libgcc\_s\_dw2-1.dll
  + libstdc++-6.dll
  + mingwm10.dll
* Из папки QtSDK\Desktop\Qt\<Версия Qt>\mingw\bin  
  (Пример полного пути: C:\Programs\QtSDK\Desktop\Qt\4.8.1\mingw\bin)
  + Если программа скомпилирована как Release
    - QtCore4.dll
    - QtGui4.dll
  + Если программа скомпилирована как Debug
    - QtCored4.dll
    - QtGuid4.dll

## Настройка Comodo для работы c Qt

* Выбрать окне Comodo выбрать: Защита+->Политика безопасности компьютера->Защищенные файлы и папки->Группы…->Добавить->Новая группа…
* Вести: «Qt stuff» (без кавычек) и нажать «Применить»
* Не выходя из окна «Группы файлов» найти в списке «Qt stuff». Выделить его и по правой кнопке мыши выбрать «Добавить»
* В поле «Добавит новый элемент» ввести: <Папка QtSDK>\pythongdb\python\_2.7based\*   
  (\* без обратной косой черты перед ней это не ошибка, так и надо писать)
* Затем нажать кнопку «+» слева от поля ввода, а затем «Применить»
* Далее повторить добавление для следующих элементов:
  + <Папка Temp текущего пользователя>\\*.bat   
    (Для Windows 7 это папка вида: C:\Users\Maxim\AppData\Local\Temp; для Windows XP это папка вида: C:\Documents and Settings\Maxim\Local Settings\Temp)
  + <Папка MinGW>\bin\*  
    (Здесь папка именно того MinGW, который используется при компиляции. У нас для компиляции используется MinGW не входящий в комплект QtSDK, а устанавливаемый отдельно)
  + <Папка где будут хранится проекты Qt>\*
  + <Папка QtSDK>\QtCreator\bin\*
  + <Папка QtSDK>\Desktop\Qt\4.8.1\mingw\bin\*
* Нажать кнопку «Применить»
* Переходим на вкладку «Правила Проактивной Защиты»
* Нажимаем кнопку «Добавить…», далее Выбрать->Группы файлов->Qt stuff
* Выбрать радио-кнопку «Применять пользовательскую политику»
* Далее выбрать Настроить->Запуск приложения-> Изменить->Добавить->Группы файлов->Исполняемые файлы->OK->OK->Применить
* Перетащить Qt stuff с помощью мыши на самый верх списка
* Далее во вкладке «Правила Проактивной Защиты» находим в списке «COMODO Internet Security». Нажимаем: Править…->Настроить->Настройка защиты->Межпроцессорный доступ к памяти-> Изменить->Добавить->Группы файлов->Qt stuff->Применить-> OK->Применить->OK
* Далее идем из окна COMODO в Защита+->Настройки проактивной защиты-> Настройки контроля исполнения приложений->Исключения->Добавить->Группы файлов->Qt stuff->Применить->OK

## Параметры проекта для IAR AVR

* General Options
  + Target
    - Processor configuration Выставить
    - Memory model (для AVR128) Small
    - Use 64-bit doubles Отметить
  + Library Configuration
    - Library Full DLIB
  + Library Options
    - Printf formatter Large
    - Scanf formatter Full
  + Heap Configuration
    - NEAR\_HEAP 300
  + System
    - CSTACK 600
    - RSTACK 100
    - Enable bit definitions… Отметить

При подключении ОЗУ 64 кБ включить:

* + - Enable external memory bus Отметить
    - RAM Base 0x2200
    - Memory size 0xDD00
    - Остальное в External Mem… 0

Конец подключения ОЗУ

* С/С++ Compiler
  + Language Extended Embedded C++
  + Preprocessor
    - Additional include directories:

$PROJ\_DIR$\..\mxsrclib\arch\avr\

$PROJ\_DIR$\..\mxsrclib\

* + Diagnostics
    - Enable remarks Отметить
    - Suppress these diagnostics Pe161, Ta006, Pa089, Pe340, Pe261, Pe961, Pe381, Pa091
* Linker
  + Output
    - Output file – Override default Отметить
    - Output file $PROJ\_FNAME$.hex
    - Other Выбрано
    - Output format intel-extended
    - Format variant None
    - Module-local symbols Include all

## Fuses и LockBits для AVR2561 платы УПМС разработки осени 2010

Когда контроллер прошивается в первый раз, то он тактируется от внутреннего RC-генератора на частоте 1 МГц. Поэтому при первой прошивке Fuse необходимо выставлять частоту на программаторе 250 кГц, т. к. она должна быть в 4 раза ниже частоты контроллера.

Fuse программируются с помощью AVR Studio

Установлены только Fuse:

* Brown-out … 4.3 V …
* Preserve EEPROM …
* Boot Flash … 4096 …
* Clock output …
* Ext. Clock … 65 ms …

В LockBits устанавливаем «Mode3 …». Все остальное по умолчанию.

## Fuses и LockBits для AVR2561 платы ГТЧ

Fuse программируются с помощью AVR Studio

Установлены только Fuse:

* Brown-out … 4.3 V …
* Preserve EEPROM …
* Boot Flash … 4096 …
* Ext. Crystal … 8,0- MHz … 16 CK + 65 ms … (Последний пункт)

В LockBits устанавливаем «Mode3 …». Все остальное по умолчанию.

## Fuses и LockBits для AVR2561 платы Сигнал-1

Fuse программируются с помощью AVR Studio

Установлены только Fuse:

* Brown-out … 4.3 V …
* Preserve EEPROM …
* Boot Flash … 4096 …
* Ext. Crystal … 8,0- MHz … 16 CK + 65 ms … (Последний пункт)

В LockBits устанавливаем «Mode3 …». Все остальное по умолчанию.

## Предупреждения IAR

* Pe177: variable "<переменная>" was <путь> <строка> declared but never referenced.
* Remark[Pa091]: operator operates on value promoted to int (with possibly unexpected result)
* Error[Pe167]: argument of type "enum S:\Files\ASUTP\GTCH\prjs\avr\gtch\gtch.h 1267 gtch::app\_t<CFG>::<unnamed>" is incompatible with parameter of type "irskey\_t"

## Параметры проекта для IAR ARM

* General Options
  + Target
    - Processor variant Device
      * Device
        + Для отладочной тахометрической платы   
          TexasInstruments LM3Sx9xx
        + Для остальных плат   
          TexasInstruments LM3SxBxx
        + Для STM32VLDISCOVERY  
          ST STM32F100xB
        + Для ГТЧ   
          ST STM32F205rx
    - Endian Mode Little
    - FPU VFPv4
  + Library Configuration
    - Library Full
    - Library low-level interface implementation Semihosted
    - Stdout/stderr (не используется) Via semihosting
  + Library Options
    - Printf formatter Large
    - Scanf formatter Full
* С/С++ Compiler
  + Language
    - Language Auto (exceptions-based)
    - С++ dialect C++
      * With exceptions Отметить
      * With RTTI Отметить
    - Require prototypes Снято
    - Language conformation Allow IAR extensions
    - Plain ‘char’ is Unsigned
    - Enable multibyte support Снято
  + Preprocessor
    - Additional include directories:

$PROJ\_DIR$\..\mxsrclib\arch\arm\

$PROJ\_DIR$\..\mxsrclib\

* + - Defined symbols
      * Для отлад. тахометрической платы: \_\_LM3Sx9xx\_\_
      * Для остальных плат: \_\_LM3SxBxx\_\_
      * Для STM32VL-Discovery \_\_STM32F100RBT\_\_
  + Diagnostics
    - Enable remarks Отметить
    - Suppress these diagnostics Pa039, Pe261, Pe340, Pa050, Pa091, Pa089, Go005, Pe193, Pa137, Pe826, Pe237, Pe2349, Pe1873, Pe068, Pa109, Pe2330
* Output Converter
  + Generate additional output Отметить
  + Output format Binary
* Linker
  + Config
    - Linker configuration file
* Скопировать IAR-овский файл для Cortex-M3 ($IAR$\arm\config\generic\_cortex.icf) в директорию проекта, переименовать по своему усмотрению, заменить в нём фрагмент

«place at address mem:\_\_ICFEDIT\_intvec\_start\_\_ { readonly section .intvec };»

на фрагмент (в одну строку писать необязательно)

«place at address mem:\_\_ICFEDIT\_intvec\_start\_\_ { readonly section .intvec, section .periph\_intvec };»

Это нужно для того, чтобы добавить собственные вектора прерываний периферии.

* Установить галку «Override default» и указать свой файл
* По кнопке «Edit…» установить:
  + Для LM3Sx9xx:
    - .intvec start = 0x00000000
    - ROM Start = 0x00000000, End = 0x0003FFFF
    - RAM Start = 0x20000000, End = 0x2000FFFF
    - CSTACK = 0x600
    - HEAP = 0x800
  + Для LM3SxBxx:
    - .intvec start = 0x00000000
    - ROM Start = 0x00000000, End = 0x0003FFFF
    - RAM Start = 0x20000000, End = 0x20017FFF
    - CSTACK = 0x2000
    - HEAP = 0x4000
  + Для STM32F100RBT:
    - .intvec start = 0x08000000
    - ROM Start = 0x08000000, End = 0x0801FFFF
    - RAM Start = 0x20000000, End = 0x20001FFF
    - CSTACK = 0x400
    - HEAP = 0x800
  + Library
    - Automatic runtime library selection Отметить
    - Override default program entry Снять
  + Optimizations
    - Perform C++ Virtual Function Elumination Снять
  + Output
    - Output file $PROJ\_FNAME$.out
    - Include debug information in output Отметить

На IAR ARM 8.22 закомментировать «#pragma message("\nWarning: The header file 'cmsis\_iar.h' is obsolete…» (3 строки) в файле «C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.0\arm\inc\c\cmsis\_iar.h»

## Использование ST библиотек

**Библиотеки находятся в SVN по адресу** [**https://5-10/svn/extern\_libs/trunk**](https://5-10/svn/extern_libs/trunk)**.**

Подключите файлы с расширением «.c» к проекту из следующих каталогов, если вы используете контроллеры stm32f4xx (если вы используете stm32f2xx, подключайте файлы из каталогов с префиксом STM32F2x):

* Для STM32F2x:
  + extern\_libs\st\STM32F2xx\_StdPeriph\_Lib\Libraries\  
    STM32F4xx\_StdPeriph\_Driver\src
  + extern\_libs\st\STM32Fx7\_ETH\_Driver\src
* Для STM32F4x:
  + extern\_libs\st\STM32F4xx\_DSP\_StdPeriph\_Lib\Libraries\  
    STM32F4xx\_StdPeriph\_Driver\src
  + extern\_libs\st\STM32F4x7\_ETH\_Driver\src

Настройка параметров проекта.

* General Options => Library Configuration => CMSIS выберите Use CMSIS.
* General Options => С/С++ Compiler=> Preprocess => Additional include directories добавьте следующие строки, если вы используете контроллеры stm32f4xx (если вы используете stm32f2xx, подключайте библиотеки с префиксом STM32F2x):

$PROJ\_DIR$\..\extern\_libs\st\STM32F4xx\_DSP\_StdPeriph\_Lib\Libraries\STM32F4xx\_StdPeriph\_Driver\inc

$PROJ\_DIR$\..\extern\_libs\st\STM32F4xx\_DSP\_StdPeriph\_Lib\Libraries\CMSIS\Device\ST\STM32F4xx\Include

$PROJ\_DIR$\..\extern\_libs\st\STM32F4x7\_ETH\_Driver\inc

* General Options => С/С++ Compiler=> Preprocess => Defined symbols добавить строку USE\_STDPERIPH\_DRIVER

Обновление библиотек

Ниже перечислены ссылки, по которым можно искать библиотеки для обновления:

* STM32F2xx\_StdPeriph\_Lib <http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32f2xx_stdperiph_lib.zip>
* STM32F2x7\_ETH\_Driver

<http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32f2x7_eth_lwip.zip>

* STM32F4xx\_DSP\_StdPeriph\_Lib

<http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32f4_dsp_stdperiph_lib.zip>

* STM32F2x7\_ETH\_Driver

<http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32f4x7_eth_iap.zip>

В SVN не включены некоторые подкаталоги для экономии места и удобства обновления.

## Использование LWIP библиотеки

Библиотеки находятся в SVN по адресу <https://5-10/svn/extern_libs/trunk>.

1. Подключите файлы с расширением .c к проекту из следующих каталогов:

* extern\_libs\lwip\src\api
* extern\_libs\lwip\src\core\ipv4
* extern\_libs\lwip\src\core
* extern\_libs\lwip\src\netif
* extern\_libs\lwip\ports\STM32F2x7\arch

1. Настройте параметры проекта.

* General Options => С/С++ Compiler=> Preprocess => Additional include directories добавьте следующие строки:

$PROJ\_DIR$\..\extern\_libs\lwip\src\include\

$PROJ\_DIR$\..\extern\_libs\lwip\ports\STM32F2x7\

$PROJ\_DIR$\..\extern\_libs\lwip\src\include\ipv4\

* General Options => С/С++ Compiler=> Preprocess => Defined symbols добавить строку USE\_LWIP

1. В настройках проекта С/С++ Compiler-> Diagnostics-> Suppress these diagnostics необходимо прописать следующие опции:

Pa039, Pa050, Pa089, Pa091, Pe068, Pe193, Pe261, Pe340, Go005

Обновление библиотек

<http://download.savannah.gnu.org/releases/lwip/>

**Использование настроек библиотек на уровне проекта**

Скопируйте файлы конфигураций из mxsrclib\utils\libconfig в каталог проекта.

В настройках проекта определите define’ы, соответствующие используемым библиотекам:

IRS\_USE\_LWIP\_PROJECT\_CONFIGURATION=1

IRS\_USE\_STM32F2x7\_ETH\_PROJECT\_CONFIGURATION=1

IRS\_USE\_STM32F2xx\_PROJECT\_CONFIGURATION=1

IRS\_USE\_STM32F4x7\_ETH\_PROJECT\_CONFIGURATION=1

IRS\_USE\_STM32F4xx\_PROJECT\_CONFIGURATION=1

## Использование HID WinApi (HID.DLL) библиотеки

Часть mxsrclib содержит код, использующий hid.dll. Чтобы использовать этот код, необходимо в настройках проекта выполнить следующие действия:

1. Добавить каталог «..\extern\_libs\winapi\hid\» в пути поиска include-файлов.
2. Добавить в список defines «IRS\_USE\_HID\_WIN\_API=1».

## Программатор ARM MT-Link

***При установке драйверов JLink, не заменять библиотеку JLinkARM.dll в IAR* (проверено для IAR 6.0)!** И не запускать JLinkDLLUpdater.exe, т.к. это тоже приводит к замене JLinkARM.dll.

Прошивать прошивку с расширением bin или hex можно используя программу

"%ProgramFiles%\SEGGER\JLinkARM\_V420n\JFlashARM.exe"

Папка JLinkARM\_V420n в пути меняется в зависимости от версии драйверов JLink

После первого запуска необходимо вызвать настройки из меню «Options» -> «Project Settings…» . Во вкладке CPU в поле Device нужно выставить правильный тип микроконтроллера. Для У309М по состоянию на 27.03.2012 это: «Luminary LM3S9B95».

Если программатор не определяется, то скорей всего слетела прошивка. Для восстановления необходимо выполнить следующие действия:

* устанавливаем [SAM-PROG](file:///\\box\Distrib\SOFT\Devices\SAM-PROG) ([сайт](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3883&category_id=163&family_id=605&subfamily_id=1586)) (при установке откроется инструкция по настройке драйвера)
* разрываем термоусадку линка
* подключаем USB кабель
* замыкаем-размыкаем (на 1 сек) J1
* отключаем USB кабель
* замыкаем J2
* подключаем USB кабель
* ждем 20 секунд
* отключаем USB кабель
* размыкаем J2
* подключаем USB

Если Windows автоматический не нашла драйверы, то нужно выбрать драйвер вручную. При этом драйверы JLinkARM должны стоять. В окне выбора драйвера изготовитель – SEGGER, модель – JLink CDC UART Port.

После этого устройство должно определиться (или создаться заново).

Инструкция [от сюда](http://starterkit.ru/html/index.php?name=forum&op=view&id=11569&num=1).

Далее запускаем SAM-PROG и качаем [прошивку](http://starterkit.ru/html/index.php?name=files&op=view&id=4).

При запуске выбираем контроллер программатора в нашем случае это at91sam7s64-ek все остальное по умолчанию.

Далее на вкладке Flash в строке Send File Name указываем расположения скаченной прошивки. После этого жмем кнопку Send File. Далее все делаем без lock'а, т.е. в нашей ситуации (Yes потом No). После этого отключаем - подключаем USB!

Программатор MT-Link предположительно является почти полной копией Segger JLink.

Если новый процессор LM3S9B95 не прошивается, можно попробовать сделать следующее:

* запустить ..\Program Files\SEGGER\JLinkARM\_V\*\*\*\jlink.exe
* выполнить команду unlock LM3Sxxx
* во время выполнения команды на выводе RESET – низкий логический уровень
* выключить-включить питание

## Замечания по Watcom C++ 1.8

Настройки в Options->C++ Compiler Switches…

* 1. File Option Switches
  + Include directories "$(%watcom)/h;  
    $(%watcom)/h/nt";..\mxsrclib\;..\mxsrclib\arch\dos32
* 2. Diagnostic Switches
  + Disable Warning message
    - 367 368 013 690

## Замечания по C++ Builder 4

При добавлении новых файлов проект после добавления файлов mxsrclib может возникать непонятная ошибка. Лечится удалением всех файлов mxsrclib, добавлением нового файла в проект и после этого вновь необходимо добавить mxsrclib.

Настройки в Project->Options

* Compiler
  + Warnings Selected
  + Кнопка Warning. Все установлено кроме:
    - 'identifier' is assigned a value that is never used (Default ON) (**-w-aus**, -w-8004)
    - Functions containing certain constructs are not expanded inline (Default ON) (-w-8026). Functions containing reserved words are not expanded inline (Default ON) (-w-8027). (**-w-inl**)
    - Non-ANSI Keyword Used: 'keyword' (Default OFF) (**-wnak**, -w8036)
    - Parameter 'parameter' is never used (Default ON) (**-w-par**, -w-8057)
    - Unreachable code (Default ON) (**-w-rch**, -w-8066)
* Directories/Conditionals
  + Include path должно быть только:
    - ..\mxsrclib\arch\cbuilder
    - ..\mxsrclib
    - $(BCB)\include
    - $(BCB)\include\vcl
  + Library path
    - $(BDS)\lib
    - $(BDS)\lib\obj
* Advanced Compiler
  + Data alignment Byte
* Packages
  + Build with runtime packages Снять
* Linker
  + Use dynamic RTL Снять

## Замечания по C++ Builder 6

Настройки в Project->Options

* Compiler
  + Warnings Selected
  + Кнопка Warning. Все установлено кроме:
    - 'identifier' is assigned a value that is never used (Default ON) (**-w-aus**, -w-8004)
    - Functions containing certain constructs are not expanded inline (Default ON) (-w-8026). Functions containing reserved words are not expanded inline (Default ON) (-w-8027). (**-w-inl**)
    - Non-ANSI Keyword Used: 'keyword' (Default OFF) (**-wnak**, -w8036)
    - Parameter 'parameter' is never used (Default ON) (**-w-par**, -w-8057)
    - W8059 Structure packing size has changed (-w-pck)
    - Unreachable code (Default ON) (**-w-rch**, -w-8066)
    - W8080 'identifier' is declared but never used (-wuse)
    - W8091 ‘type’ argument ‘specifier’ passed to 'function' is a ‘iterator category’ iterator: ‘iterator category’ iterator required (-wstl)
    - W8092 ‘type’ argument ‘specifier’ passed to 'function' is not an iterator: ‘iterator category’ iterator required (-wstl)
* Directories/Conditionals
  + Include path должно быть только:
    - ..\mxsrclib\arch\cbuilder
    - ..\mxsrclib
    - $(BCB)\include
    - $(BCB)\include\vcl
  + Library path
    - $(BDS)\lib
    - $(BDS)\lib\obj
* Advanced Compiler
  + Data alignment <оставить по умолчанию> (Double word)
* Packages
  + Build with runtime packages Снять
* Linker
  + Use dynamic RTL Снять

Настройки Tools->Editor Options

* General
  + Use tab character Снять
  + Undo after save Отметить
  + Block indent 2
  + Tab stops 2
* Code Insight (не обязательно)
  + Code completion Снять
  + Code parameters Снять
  + Tooltip symbol insight Снять

Настройки Tools->Environment Options (не обязательно)

* Preferences
  + Editor files Отметить
  + Project desktop Отметить
* C++ Builder Direct
  + … poll Network Снять
* Class Explorer
  + … show Explorer Снять

Исправить файл \_str.h, строка 189. К функции \_\_strcmp\_\_ добавить слово inline

## Замечания по C++ Builder 2006

Настройки в Project->Options (Перед настройкой выставить выпадающий список Build Configuration в значение All Configurations)

* C++ Compiler
  + Paths and Defines
    - ..\mxsrclib\arch\cbuilder
    - ..\mxsrclib
    - $(BCB)\include
    - $(BCB)\include\vcl
    - $(BDS)\include\dinkumware
  + Warnings (Selected) Все установлено кроме:
    - 'identifier' is assigned a value that is never used (Default ON) (**-w-aus**, -w-8004)
    - Functions containing certain constructs are not expanded inline (Default ON) (-w-8026). Functions containing reserved words are not expanded inline (Default ON) (-w-8027). (**-w-inl**)
    - Non-ANSI Keyword Used: 'keyword' (Default OFF) (**-wnak**, -w8036)
    - Parameter 'parameter' is never used (Default ON) (**-w-par**, -w-8057)
    - Unreachable code (Default ON) (**-w-rch**, -w-8066)
  + Data alignment 1 Byte …
* Linker
  + Paths and Defines
    - $(BDS)\lib
    - $(BDS)\lib\obj
    - $(BDS)\lib\psdk
  + Linking
    - Use dynamic RTL Снять
* Packages
  + Build with runtime packages Снять

Настройки в Tools->Options

* Editor Options
  + Undo after save Отметить
  + Source options
    - Use tab character Снять
    - Show tab character Отметить
    - Block indent 2
    - Tab stops 2

## Замечания по C++ Builder 2007

Настройки в Project->Options (Перед настройкой выставить выпадающий список Build Configuration в значение Base)

Настройки в Tools->Options

* C++ Compiler
  + Warnings (Enable selected Warnings) Все True кроме:
    - 'identifier' is assigned a value that is never used (Default ON) (**-w-aus**, -w-8004)
    - Functions containing certain constructs are not expanded inline (Default ON) (-w-8026). Functions containing reserved words are not expanded inline (Default ON) (-w-8027). (**-w-inl**)
    - Non-ANSI Keyword Used: 'keyword' (Default OFF) (**-wnak**, -w8036)
    - Parameter 'parameter' is never used (Default ON) (**-w-par**, -w-8057)
    - Unreachable code (Default ON) (**-w-rch**, -w-8066)
* Editor Options
  + Undo after save Отметить
  + Source options
    - Use tab character Снять
    - Show tab character Отметить
    - Block indent 2
    - Tab stops 2

## Замечания по C++ Builder 2010

Звездочкой помечены разделы и параметры рекомендуемые, но не обязательные

Настройки в Project->Options (Перед настройкой выставить выпадающий список Build Configuration в значение Base)

* Directories and Conditionals
  + Final output directory final
  + Include path – здесь обязательно должно быть приведенное ниже, эти папки там появляются после подключения файлов библиотеки, остальное оставляем как было по умолчанию
    - ..\mxsrclib\arch\cbuilder
    - ..\mxsrclib
* C++ Compiler
  + Warnings (Enable selected Warnings) Все True кроме:
    - 'identifier' is assigned a value that is never used (Default ON) (**-w-aus**, -w-8004)
    - Functions containing certain constructs are not expanded inline (Default ON) (-w-8026). Functions containing reserved words are not expanded inline (Default ON) (-w-8027). (**-w-inl**)
    - Non-ANSI Keyword Used: 'keyword' (Default OFF) (**-wnak**, -w8036)
    - Parameter 'parameter' is never used (Default ON) (**-w-par**, -w-8057)
    - Unreachable code (Default ON) (**-w-rch**, -w-8066)
    - W8080 Identifier declared but never used
    - **W8104** Local Static with constructor dangerous for multi-threaded apps
* C++ Linker
  + Непосредственно пункт «C++ Linker»
    - Dynamic RTL False
* Packages
  + Build with runtime packages Снять

Настройки в Tools->Options

* Environment Options
  + \*Autosave options
    - Editor files Установить
    - Project desktop Установить
* Editor Options
  + Undo after save Установить
  + Source options
    - Use tab character Снять
    - Show tab character Установить
    - Block indent 2
    - Tab stops 2
  + \*Code Insight
    - Code completion Снять
    - Code parameters Снять
    - Tooltip expr… Установить
    - Tooltip symb… Снять
    - Block completion Снять
    - Code template completion Снять
* Debugger Options
  + Embarcadero Debuggers
  + Native OS Exceptions – выделить весь список «Exceptions» и в группе «Hanled by» выбрать «Debugger»

## Замечания по C++ Builder XE Update 1

Звездочкой помечены разделы и параметры рекомендуемые, но не обязательные

Настройки в Project->Options (Перед настройкой выставить выпадающий список Build Configuration в значение Base)

* Directories and Conditionals
  + Include path – здесь обязательно должно быть приведенное ниже, эти папки там появляются после подключения файлов библиотеки, остальное оставляем как было по умолчанию
    - ..\mxsrclib\arch\cbuilder
    - ..\mxsrclib
  + Warnings (Enable selected Warnings) Все True кроме:
    - 8004 Identifier assigned a value that is never used
    - 8017 Redefinition of ‘macro’ is not identical
    - 8026-8027 Functions are not expanded inline
    - 8036 Non-ANSI Keyword Used
    - 8057 Parameter is never used
    - 8066 Unreachable code
    - 8080 Identifier declared but never used
    - 8087, 8089-8092 STL warnings (operator visibility, iterators)
    - 8104 Local Static with constructor dangerous for multi-threaded apps
    - 8118 Inline member function in Package class
    - 8123 Path not found in option argument
* C++ Linker
  + Непосредственно пункт «C++ Linker»
    - Link with Dynamic RTL False
* Packages
  + Build with runtime packages Снять

Настройки в Tools->Options

* Environment Options
  + Environment Variables Выберите переменную «BDSCOMMONDIR» и нажмите кнопку «Add Override». Проверьте, содержит ли путь русские буквы. Если содержит, создайте альтернативный каталог, путь к которому не содержит русских букв, и впишите этот путь. Например, если указан каталог «C:\Documents and Settings\All Users\Документы\RAD Studio\8.0», то альтернативного каталог может выглядеть так: «C:\Documents and Settings\All Users\Shared Documents\RAD Studio\8.0». В случае указанного выше альтернативного каталога настройте права доступа в каталог Shared Documents, разрешающие полный доступ для всех пользователей.
* Editor Options
  + Undo after save Установить
  + Source options
    - Use tab character Снять
    - Show tab character Установить
    - Block indent 2
    - Tab stops 2
  + \*Code Insight
    - Code completion Снять
    - Code parameters Снять
    - Tooltip expression evalution… Установить
    - Tooltip symbol insight Снять
    - Block completion Снять
    - Code template completion Снять

## Замечания по C++ Builder 10.3

19.03.2025 Первоначально Крашенинников

Звездочкой помечены разделы и параметры рекомендуемые, но не обязательные

Настройки в Project->Options (Перед настройкой выставить выпадающий список Target в значение All configurations)

* Building->C++ Compiler->Directories and Conditionals
  + Include file search path – здесь обязательно должно быть приведенное ниже, эти папки там появляются после подключения файлов библиотеки, остальное оставляем как было по умолчанию
    - ..\mxsrclib\arch\cbuilder
    - ..\mxsrclib
* Building->C++ Compiler->Warnings (Enable selected Warnings) Все True кроме:
  + - 8004 Identifier assigned a value that is never used
    - 8017 Redefinition of ‘macro’ is not identical
    - 8026-8027 Functions are not expanded inline
    - 8036 Non-ANSI Keyword Used
    - 8057 Parameter is never used
    - 8064, 8065 Call to function with no prototype
    - 8066 Unreachable code
    - 8080 Identifier declared but never used
    - 8087, 8089-8092 STL warnings (operator visibility, iterators)
    - 8104 Local Static with constructor dangerous for multi-threaded apps
    - 8111 Accessing deprecated entity
    - 8118 Inline member function in Package class
    - 8121 Found invalid character from source code in the current text locale
    - 8123 Path not found in option argument
    - 8126 Base class of exported class should have exported constructor
    - 8135 Unknown #pragma
* Building->C++ Linker
  + Непосредственно пункт «C++ Linker»
    - Link with Dynamic RTL False
    - Link with the Delphi Runtime Library False
* Packages->Runtime Packages
  + Link with runtime packages False

Настройки в Tools->Options

* IDE
  + Environment Variables Выберите переменную «BDSCOMMONDIR» и нажмите кнопку «Add Override». Проверьте, содержит ли путь русские буквы. Если содержит, создайте альтернативный каталог, путь к которому не содержит русских букв, и впишите этот путь. Например, если указан каталог «C:\Documents and Settings\All Users\Документы\RAD Studio\8.0», то альтернативного каталог может выглядеть так: «C:\Documents and Settings\All Users\Shared Documents\RAD Studio\8.0». В случае указанного выше альтернативного каталога настройте права доступа в каталог Shared Documents, разрешающие полный доступ для всех пользователей.
  + Saving and Desktop
    - \*Save project desktop when closing
* User Interface->Editor Options
  + Undo after save Установить
  + Source options
    - Use tab character Снять
    - Show tab character Установить
    - Block indent 2
    - Tab stops 2
  + Display
    - Right margin 100
  + \*Code Insight
    - Code completion Снять
    - Code parameters Снять
    - Tooltip expression evalution… Установить
    - Tooltip symbol insight Снять
    - Block completion Снять
    - Code template completion Снять

## Обработка ошибок

Основой для обработки ошибок является класс приемо-передатчик ошибок:

**irs::error\_trans\_base\_t**.

Для получения указателя на него необходимо вызвать функцию:

**irs::error\_trans\_base\_t \*irs::error\_trans();**

При первом вызове эта функция создает экземпляр класс **irs::error\_trans\_base\_t** и возвращает указатель на него. При последующих вызовах экземпляр класса не создается, а возвращается указатель на экземпляр класса, созданный при первом вызове функции. Таким образом, все модули приложения независимо друг от друга могут использовать один общий экземпляр этого класса.

Работа с этим классом может осуществляться по двум вариантам:

**DevExpress VCL**

Инструкция ниже проверена C++ Builder 2010

После установки DevExpress VCL следует зайти в меню Tools->Options->Environment…->C++ Options->Paths and Directories. Скопировать содержимое поля «Browsing path:» в текстовый редактор. В текстовом редакторе удалить из строки все что не касается DevExpress. Сделать копию получившейся строки. Выполнить поиск и замену слова «Sources» на слово «Packages» в одной копии строки. Объединить обе получившихся строки через «;» в одну строку. Получившуюся строку добавить к содержимому поля «Library path:» с учетом правильной расстановки «;».

## Наиболее часто возникающие ошибки

* Использование '=' вместо '==' и наоборот.
* Отсутствие инициализации переменных.
* Потеря единицы. Например, использование ‘<=’ вместо ‘<’ или наоборот.
* Выход за границу массива.
* Забывание выделить динамическую память. Забывание освободить выделенную память. Повторное освобождение уже освобожденной памяти.
* C++ Builder начиная с 6 версии не объединяет файлы .h и .cpp, если в h-файле не определена или неправильно определена защитная define-константа. Например, если файл называется irsmath.h, то define-константа должна быть IRSMATHH.
* В C++ Builder предупреждение “cannot create predcompiled headers” возникает, если в **.cpp** файле использующем VCL нет строк в начале:

#include <vcl.h>

#pragma hdrstop

Также, возможно, в h-файле присутствует определение функции без спецификатора **inline**.

* При передаче размера буфера С-строки **char\*** можно забыть о том, что он должен включать нуль-символ.
* При написании специализации шаблона функции в h-файле не забывайте поставить inline, если специализация по сути становится не шаблонной функцией. Тоже самое относится к функциям специализации класса, если они определены вне класса. В Builder все это может вызывать предупреждение “cannot create predcompiled headers” совершенно не в том месте где это подобная функция написана неправильно. Например:

template <class T>

const char\* type\_to\_string()

{

return "unknown";

}

template <>

inline const char\* type\_to\_string<float>()

{

return "float";

}

* При добавлении дополнительных ключевых слов (**inline**, **virtual**) к описанию функции легко забыть указать тип возвращаемого значения.
* **auto\_ptr** нельзя использовать в качестве объекта контейнера стандартной библиотеки, в частности контейнера vector. Объявление типа:

vector<**auto\_ptr**<my\_object\_t> > obj\_vec;

— недопустимо!

* При взятии типа из шаблона необходимо ставить ключевое слово **typename**. Например:

**typename** map<T, K>::const\_iterator it = map\_container.begin();

В Watcom при этом появляется сообщение типа: «symbol 'Type' has not been declared» для контейнеров стандартной библиотеки.

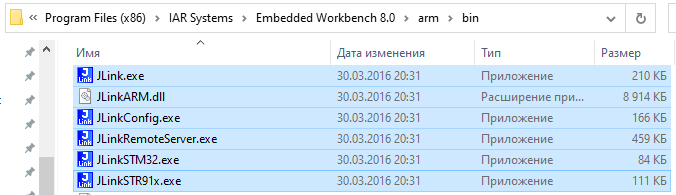
* В ОС Windows, при использовании сокетов необходимо различать константы

INVALID\_SOCKET и SOCKET\_ERROR. Например, результат функции socket необходимо сравнивать с INVALID\_SOCKET, а результат функций: send, recv и select необходимо сравнивать с SOCKET\_ERROR.

## Работа с клонами J-Link

Для работы с J-Link v10 требуется версия ПО Segger v5.12.

Чтобы внедрить его в IAR (v8.22), нужно в C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.0\arm\bin заменить файлы jlink\*.exe и jlink\*.dll на файлы из архива JLink\_V5\_12.exe (6 файлов).

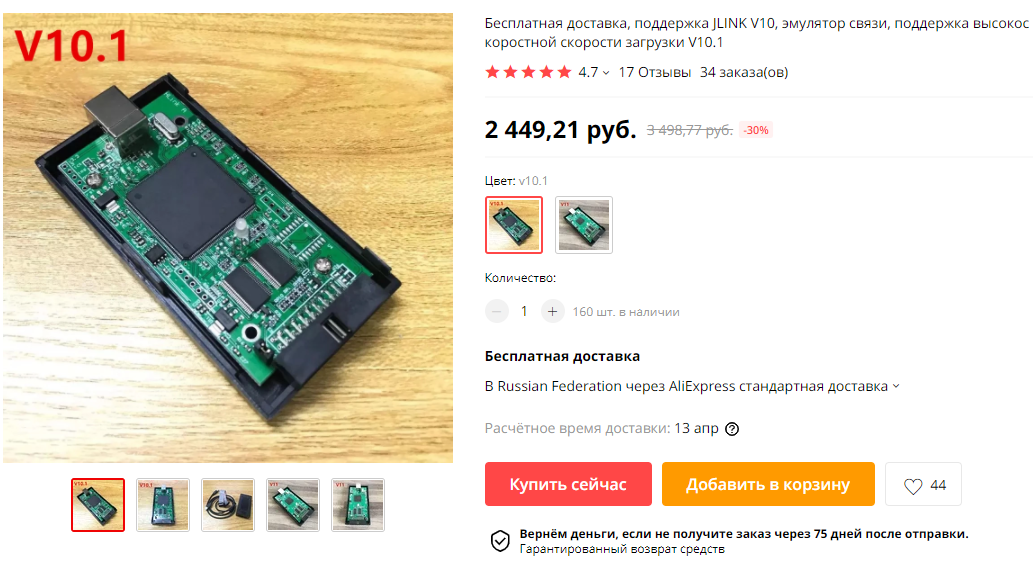


Архив JLink\_V5\_12.exe находится в [\\box\Distrib\SOFT\Compilers\J-Link\other\JLink\_V5\_12.exe](file:///\\box\Distrib\SOFT\Compilers\J-Link\other\JLink_V5_12.exe), и открывается с помощью 7-zip. Устанавливать через запуск exe-файла необязательно.

Старые файлы из папки IAR можно прибрать, но вообще они не нужны.

Проверено на китайском клоне

<https://aliexpress.ru/item/32974951095.html?spm=a2g0o.productlist.0.0.5daf3562uh3BlS&algo_pvid=e30c6d64-b863-426f-9dfa-fe3aa0b67cb4&algo_expid=e30c6d64-b863-426f-9dfa-fe3aa0b67cb4-10&btsid=0b8b035916133894934978386e112f&ws_ab_test=searchweb0_0,searchweb201602_,searchweb201603_&sku_id=12000016695113937>



Перед началом работы необходимо снять джампер, показанный на иллюстрации, иначе на схему будет подаваться питание +3,3 В с программатора.

<Конец>