



LDM-SYSTEMS

E-mail: info@ldm-systems.ru

URL: www.ldm-systems.ru

тел.: +7 (495) 500-89-20

факс: +7 (495) 536-46-67

Отладочная плата

LDM-K1986BE92QI



СДЕЛАНО В РОССИИ

LDM-SYSTEMS

Июнь 2014

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Основные характеристики микроконтроллеров серии 1986BE9x ..	4
2 Описание и работа.....	7
2.1 Принципиальная электрическая схема.....	7
2.2 Комплектация и опции	8
2.3 Инструкция по быстрому старту в среде Phyton.....	10
2.3.1 Установка программного обеспечения и драйверов.....	10
2.3.2 Подключение платы к компьютеру	13
2.3.3 Открытие демонстрационного проекта в среде Phyton	16
2.3.4 Создание простого проекта в среде Phyton	18
2.3.5 Загрузка прошивки в FLASH микроконтроллера.....	27
2.4 Создание простого проекта в среде Keil uVision.....	29
2.5 Полезные программы	46
2.6 Монтажные чертежи	48
2.7 Трассировка по слоям	49
3 Эксплуатация, хранение и транспортирование	50
Литература	51

ВВЕДЕНИЕ

Отладочная плата LDM-K1986BE92QI представляет собой печатную плату размером 127x102x22 мм. На плате установлен 32-битный микроконтроллер фирмы ЗАО «ПКК Миландр» с ядром ARM Cortex-M3. Отладочная плата LDM-K1986BE92QI предназначена для быстрого обучения азам проектирования электронных устройств на основе контроллера K1986BE92QI (MDR32F9Q2I), легкого старта новых проектов, сокращения времени выхода нового продукта на рынок.

В составе отладочного комплекта, как опции, могут устанавливаться интерфейсные микросхемы протоколов: CAN SN65HVD232D фирмы Texas Instruments Incorporated и RS485 MDRI4852SI (K5559ИИ10Б) фирмы ЗАО «ПКК Миландр».

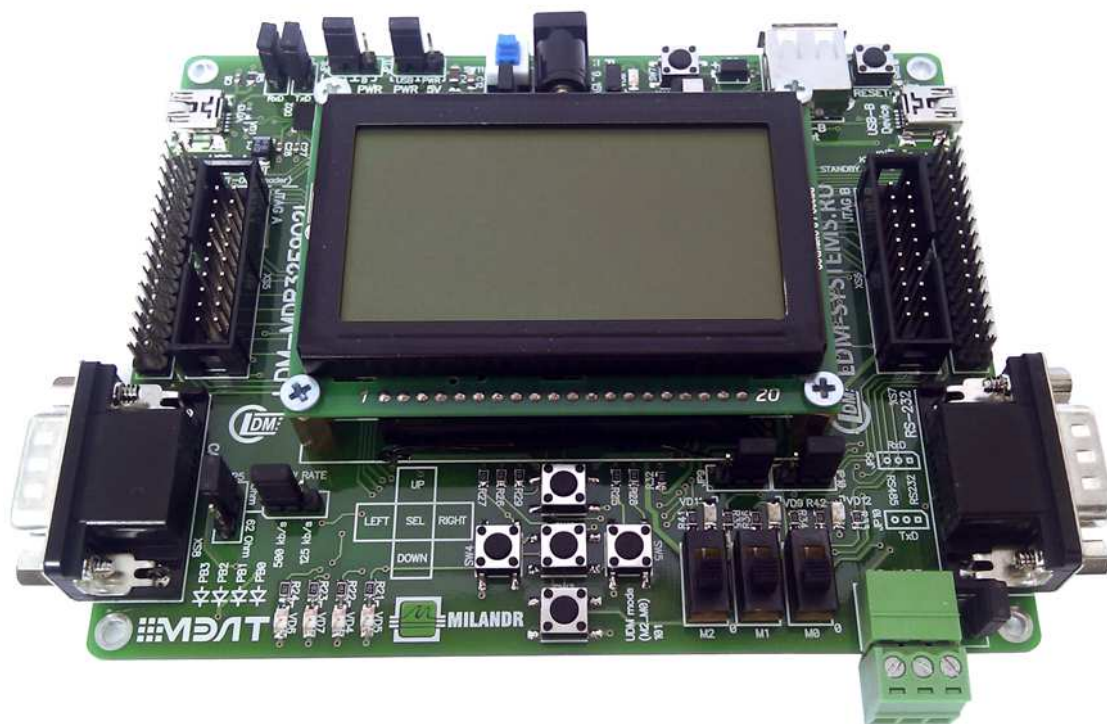


Рисунок 1. Общий вид отладочной платы LDM-K1986BE92QI-H в полной комплектации

1 Основные характеристики микроконтроллеров серии 1986BE9x

Ядро:

- ARM 32-битное RISC-ядро Cortex™-M3 ревизии 2.0, тактовая частота до 80 МГц, производительность 1.25 DMIPS/МГц (Dhrystone 2.1) при нулевой задержке памяти;
- блок аппаратной защиты памяти MPU;
- умножение за один цикл, аппаратная реализация деления.

Память:

- встроенная энергонезависимая Flash-память программ размером 128 Кбайт;
- встроенное ОЗУ размером 32 Кбайт;
- контроллер внешней шины с поддержкой микросхем памяти СОЗУ, ПЗУ, NAND Flash.

Питание и тактовая частота:

- внешнее питания 2,2÷3,6 В;
- встроенный регулируемый стабилизатор напряжения на 1,8 В для питания ядра;
- встроенные схемы контроля питания;
- встроенный домен с батарейным питанием;
- встроенные подстраиваемые RC генераторы 8 МГц и 40 кГц;
- внешние кварцевые резонаторы на 2÷16 МГц и 32 кГц;
- встроенный умножитель тактовой частоты PLL для ядра;
- встроенный умножитель тактовой частоты PLL для USB.

Режим пониженного энергопотребления:

- режимы Sleep, Deep Sleep и Standby;
- батарейный домен с часами реального времени и регистрами аварийного сохранения.

Аналоговые модули:

- два 12-ти разрядных АЦП (до 16 каналов);
- температурный датчик;
- двухканальный 12-ти разрядный ЦАП;
- встроенный компаратор.

Периферия:

- контроллер DMA с функциями передачи Периферия-Память, Память-Память;
- два контроллера CAN интерфейса;
- контроллер USB интерфейса с функциями работы Device и Host;
- контроллеры интерфейсов UART, SPI, I2C;
- три 16-ти разрядных таймер-счетчика с функциями ШИМ и регистрации событий;
- до 96 пользовательских линий ввода-вывода.

Отладочные интерфейсы:

- последовательные интерфейсы SWD и JTAG.

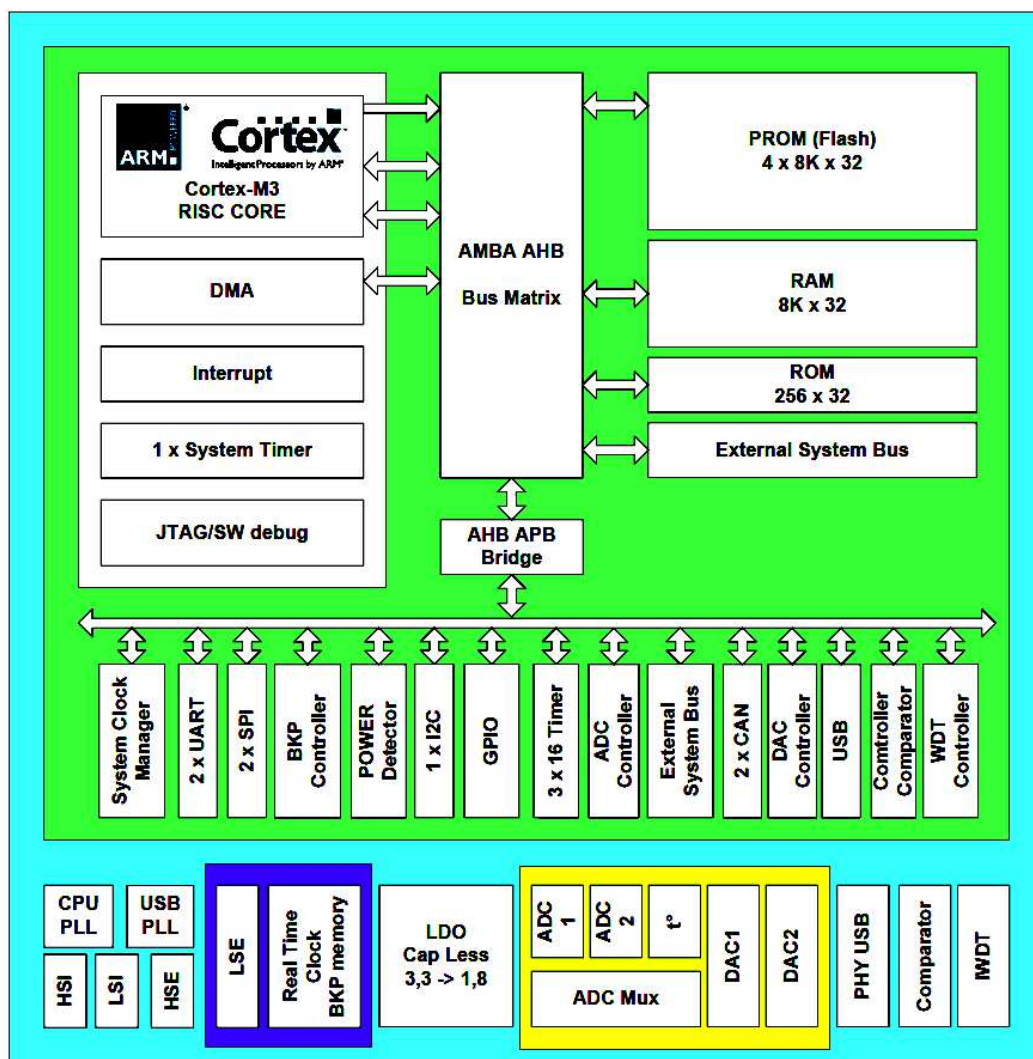


Рисунок 2. Структурная блок-схема микроконтроллера 1986BE9x

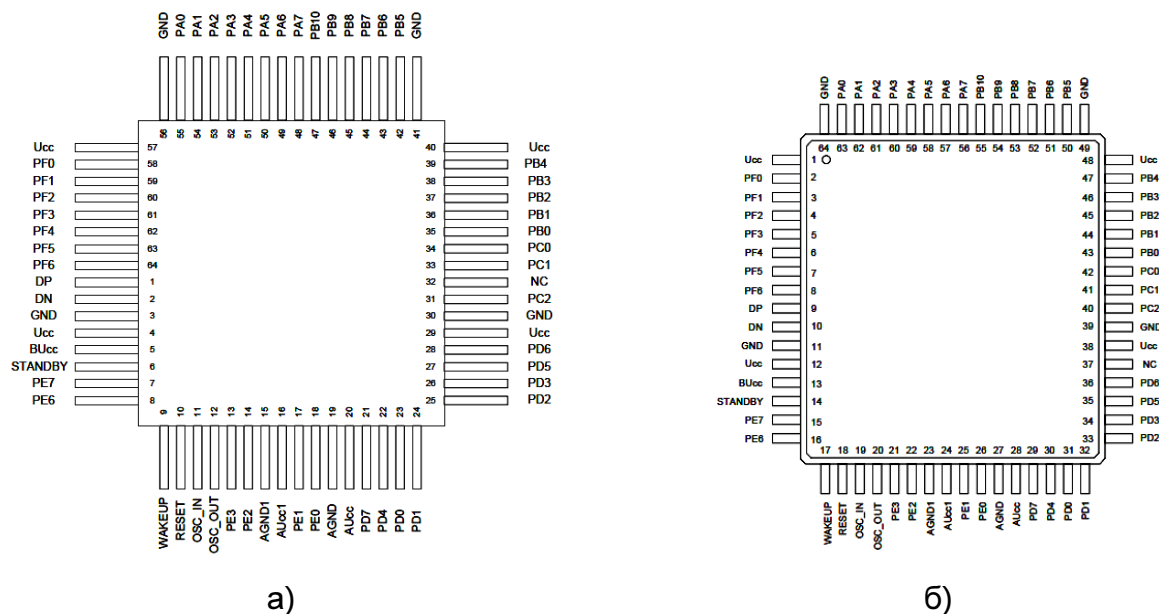
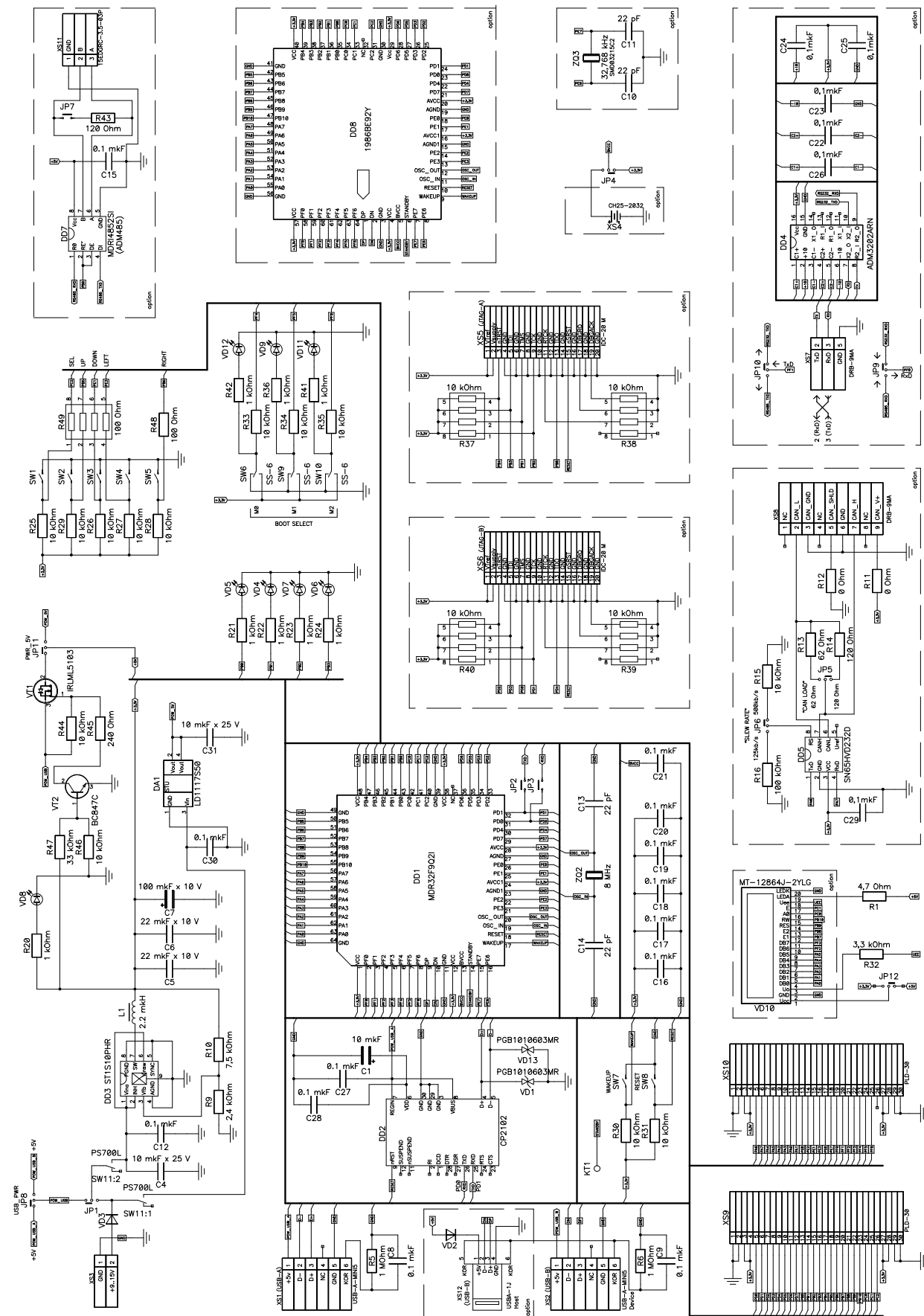


Рисунок 3. Расположение выводов: а) 64-х выводной металлокерамический корпус H18.64-1B, б) 64-х выводной пластиковый корпус LQFP64

2 Описание и работа

2.1 Принципиальная электрическая схема



2.2 Комплектация и опции

Отладочная плата LDM-K1986BE92QI поставляется в нескольких комплектациях. В базовой комплектации отсутствуют драйверы внешней периферии: RS232, CAN, RS485, разъем USB Host, площадка для литиевой батареи, генератор часов реального времени, ЖКИ индикатор, разъемы JTAG A и JTAG B. Общий вид базовой комплектации отладочной платы LDM-K1986BE92QI приведен на рисунке 4.

На плате в базовой комплектации имеются: 5 кнопок управления, 4 пользовательских светодиода, 3 движковых переключателя режима программирования FLASH, кнопки RESET и WAKEUP, разъем miniUSB USB-UART преобразователя (UART-загрузчик), разъем miniUSB интерфейса USB Device, преобразователи напряжения +5 В, 1 А; +3.3 В, 2.5 А, все выводы микроконтроллера выведены на 2 разъема XS9, XS10.

Питание платы осуществляется от постоянного стабилизированного источника с напряжением $+9 \div 15$ В, $0.5 \div 2$ А (XS3) или от miniUSB порта (XS1, XS2), подключенного кабелем к порту USB персонального компьютера.

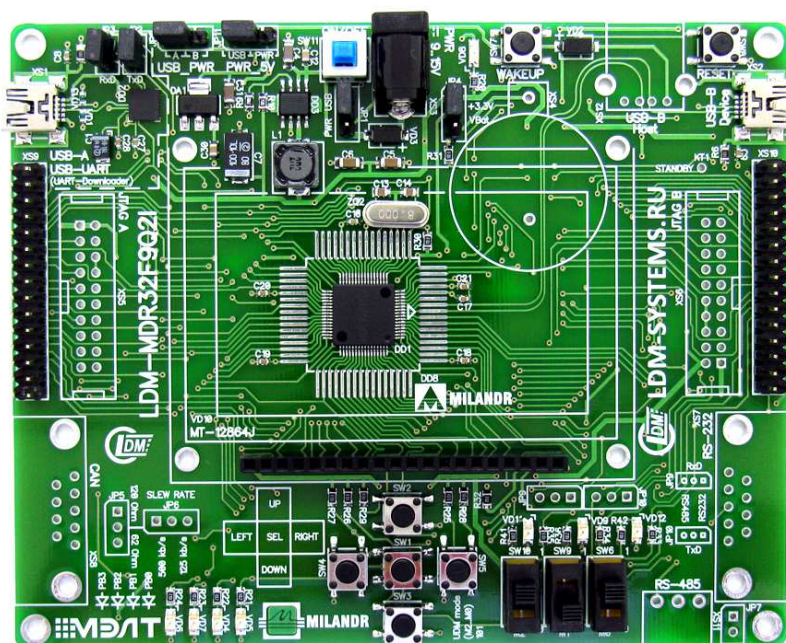


Рисунок 4. Общий вид базовой комплектации отладочной платы LDM-K1986BE92QI

Маркировка платы зависит от варианта комплектации изделия дополнительными опциями. При базовой комплектации маркировка соответствует LDM-K1986BE92QI. Дополнительные опции отражаются в маркировке в виде букв, идущих после основной надписи LDM-K1986BE92QI.

Например:

База + RS-485 + ЖКИ - **LDM-K1986BE92QI-CD**;

База + RS-232 + CAN + RS-485 + ЖКИ - **LDM-K1986BE92QI-ABCD**;

База + все опции - **LDM-K1986BE92QI-H**;

База + все опции + K1986BE92УК ОТК ("1") - **LDM-K1986BE92QI-HM**.

Варианты комплектации отладочной платы приведены в таблице 1.

Таблица 1

Варианты комплектации

№	Опция	Маркировка
1	RS-232 ADM3232 (Analog Devices)	A
2	CAN SN65HVD232D (Texas Instruments Incorporated)	B
3	RS-485 ADM485 или MDRI4852SI (K5559ИН10Б) (ЗАО «ПКС Миландр»)	C
4	ЖКИ МТ-12864J (ООО «МЭЛТ»)	D
5	Батарея + 32,768 кГц	E
6	JTAG A, JTAG B	F
7	USBA-1J Host	G
8	Все опции (ABCDEFGH)	H
9	K1986BE92УК ОТК ("1") (диапазон работы 0 ... +70 °С) H18.64-1В металлокерамический (ЗАО «ПКС Миландр»)	M

Отладочная плата упаковывается в фирменную коробку. В комплекте с платой идут:

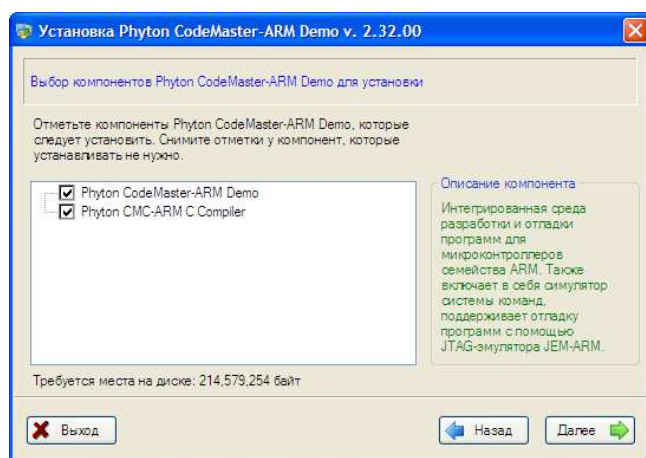
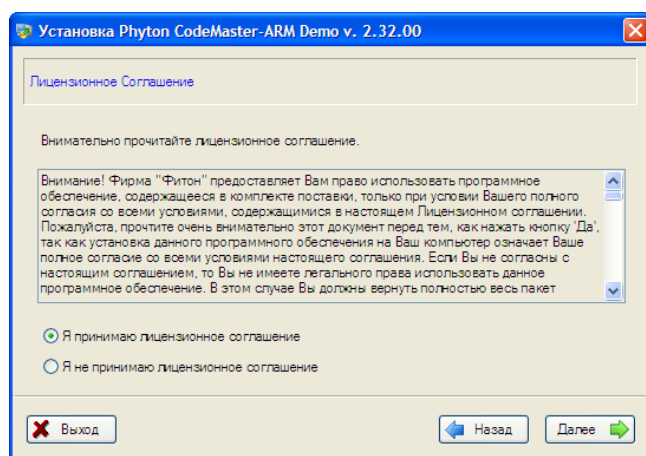
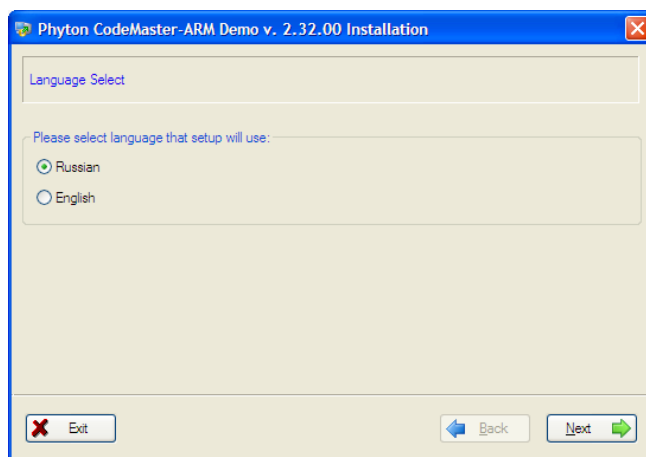
- CD-R диск со всей необходимой документацией и программным обеспечением;
- кабель USB-A – miniUSB (5 выводов).

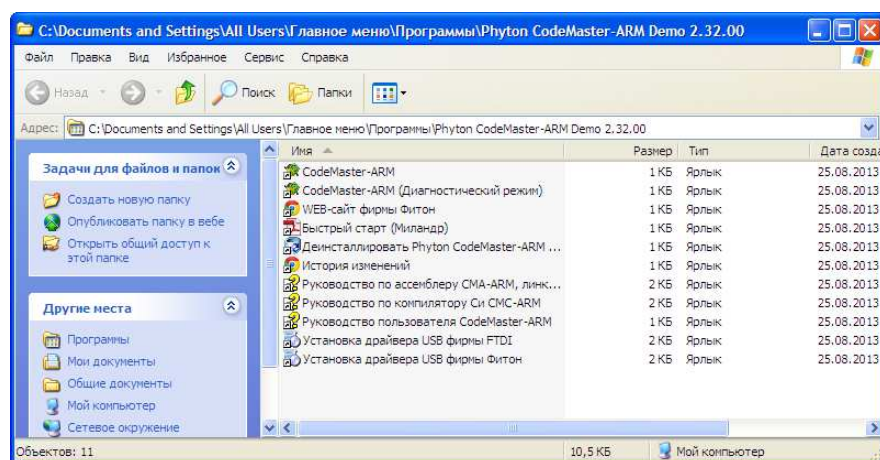
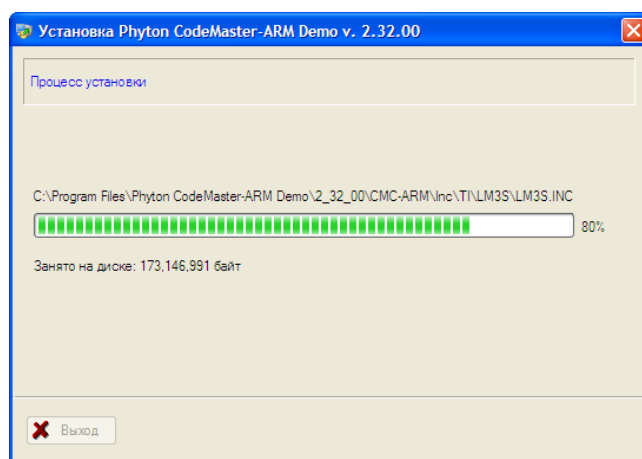
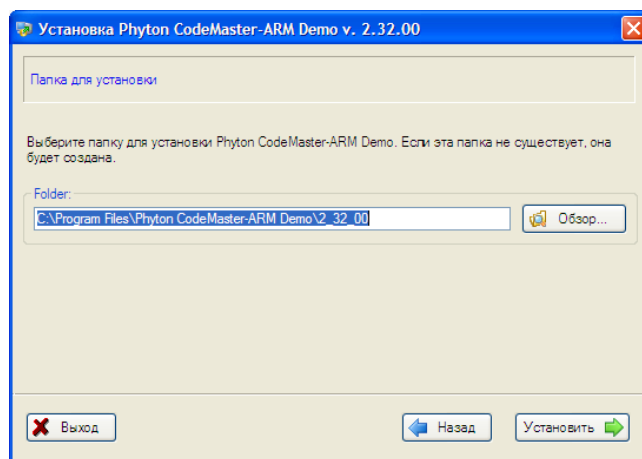
2.3 Инструкция по быстрому старту в среде Phytон

2.3.1 Установка программного обеспечения и драйверов

ШАГ 1: Устанавливаем компилятор Phytон CodeMaster-ARM Demo 2.

На CD-R диске запускаем файл по следующему адресу: CD-R\Компилятор\Phytон_codemaster-arm_demo.exe. Следуем графическим подсказкам ниже:

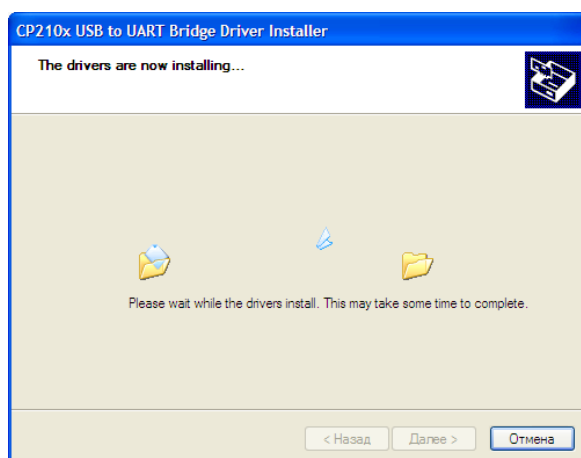
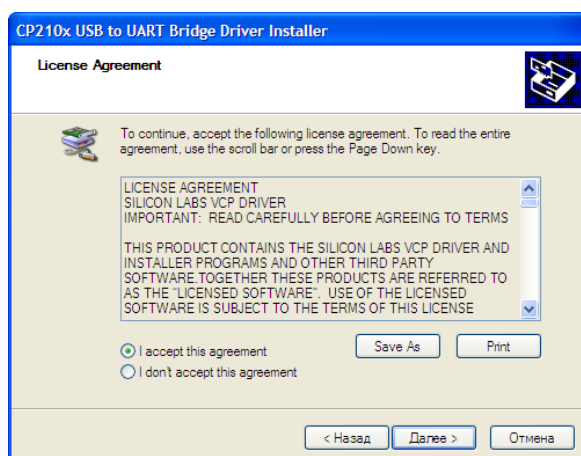
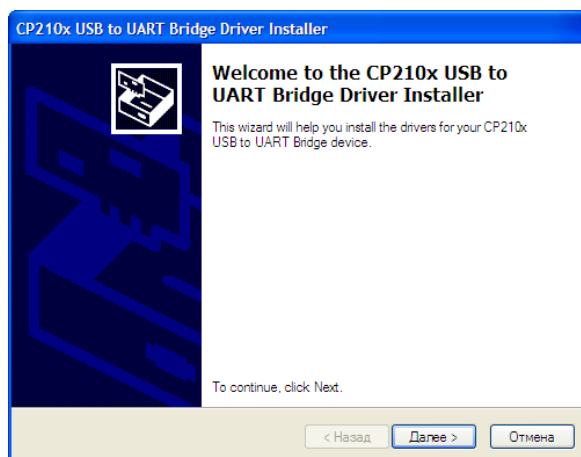


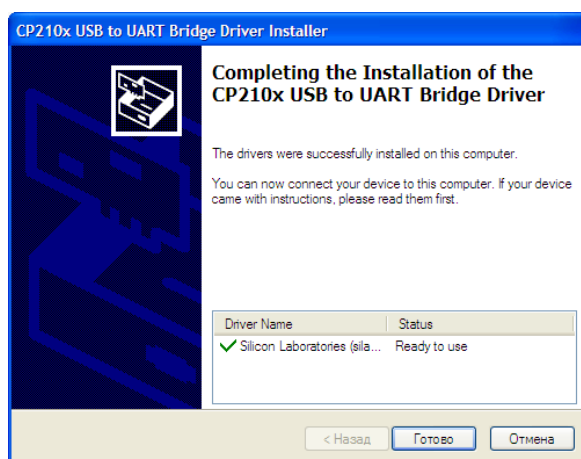


Процесс установки компилятора завершен. В разделе меню «Пуск» создана папка «Phyton CodeMaster-ARM Demo 2.32.00» с ярлыками для запуска компилятора. Запуск компилятора осуществляется утилитой «CodeMaster-ARM».

ШАГ 2: Устанавливаем драйвера для USB-загрузчика.

На CD-R диске запускаем файл по следующему адресу: CD-R\DRV\CP210x_VCP_Windows\CP210xVCPInstaller_x86.exe (для 32 битной операционной системы) или CD-R\DRV\CP210x_VCP_Windows\CP210xVCPInstaller_x64.exe (для 64 битной операционной системы). Следуем графическим подсказкам ниже:





Драйвера установлены.

2.3.2 Подключение платы к компьютеру

Питание отладочной плата LDM-K1986BE92QI осуществляется от постоянного стабилизированного источника с напряжением $+9 \div 15$ В, 0.5-2 А (XS3) или от miniUSB порта (XS1, XS2), подключенного кабелем к порту USB персонального компьютера. В таблице 2 приведены режимы включения джамперов, переключателей и их функции.

Таблица 2

Режимы включения джамперов и их функции

Джампер	Положение	Функционал
JP1	PWR	Питание от разъема XS3
	USB	Питание от USB XS1 или XS2
JP2	TxD	Перемычка линии TxD от Мк к USB-загрузчику
JP3	RxD	Перемычка линии RxD от Мк к USB-загрузчику
JP4	+3,3V	Если нет литиевой батареи
	VBat	Если литиевая батарея установлена
JP5	120 Ohm	Шунт 120 Ом на линии CAN
	62 Ohm	Шунт 62 Ом на линии CAN
JP6	500 kb/s	Выбор скорости CAN 500 кбит/с
	125 kb/s	Выбор скорости CAN 125 кбит/с
JP7	XS11	Шунт 120 Ом на линии RS485
JP8	A	Питание от USB разъема XS1
	B	Питание от USB разъема XS2
JP9	RxD RS232	Переключаем UART RxD на RS232 интерфейс
	RxD RS485	Переключаем UART RxD на RS485 интерфейс
JP10	TxD RS232	Переключаем UART TxD на RS232 интерфейс
	TxD RS485	Переключаем UART TxD на RS485 интерфейс

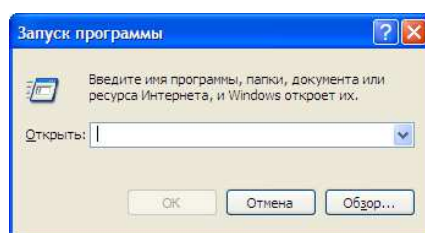
JP11	USB	Питание +5 В поступает от USB XS1 или XS2
	PWR	Питание +5 В стабилизируется от XS3
JP12	+5V	Питание ЖКИ дисплея от +5 В
	+3V	Питание ЖКИ дисплея от +3,3 В
SW6	M0	Выбор режима программирования
SW9	M1	Выбор режима программирования
SW10	M2	Выбор режима программирования

Шаг 1: Настройка джамперов и включение питания.

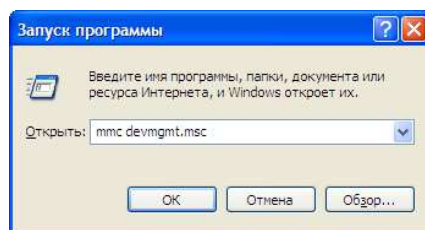
Воспользуемся схемой питания от USB XS1. Выставляем все джамперы в положение 1-2 (вывод 1 отмечен скошенным углом в слое маркировки). Устанавливаем переключатели XS6, XS9, XS10 в положение «0». Подключаем кабель USB к компьютеру и к разъему XS1 (подсветка ЖКИ засветится, если он установлен). Нажимаем кнопку SW11 (ON/OFF). Светодиод VD8 ярко засветится (на ЖКИ включится демонстративная анимация, если он установлен).

Шаг 2: Настройка номера виртуального COM-порта USB-загрузчика.

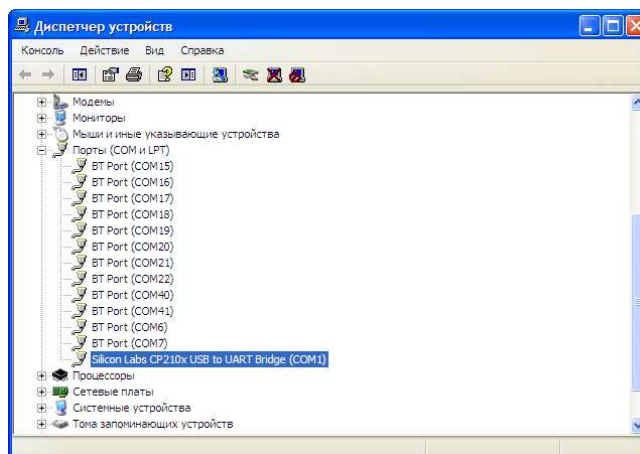
Открываем диспетчер устройств. Нажимаем на клавиатуре кнопку «Windows» , удерживаем ее и нажимаем кнопку «R». Откроется окно командной строки.



Вводим в поле командной строки текст «**mmc devmgmt.msc**» и нажимаем «**ENTER**».

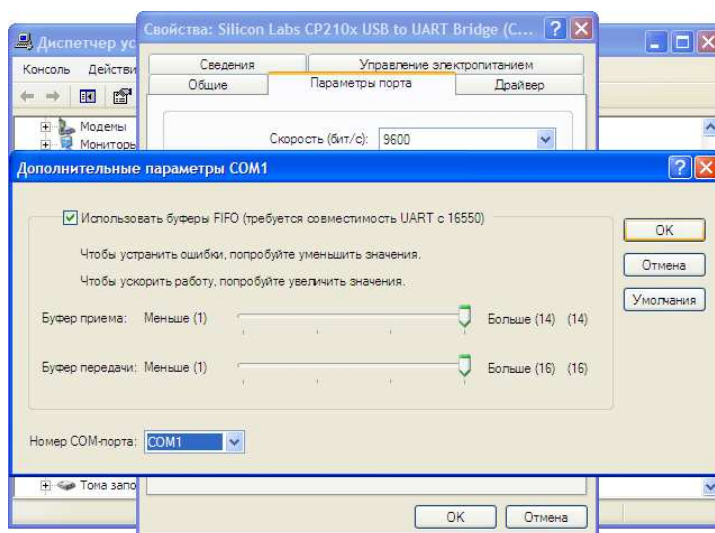


В открывшемся диспетчере устройств заходим в раздел «**Порты COM и LPT**».



Находим устройство «Silicon Labs CP210x USB to UART Bridge (COM x)». Если номер COM-порта устройства «x» равен 1, то закрываем диспетчер устройств. В противном случае либо запоминаем номер COM-порта, либо меняем его на 1.

Для смены номера COM-порта заходим в свойства устройства → Параметры порта → Дополнительно и выбираем «Номер COM-порта:» равным COM1. Нажимаем ОК.



P.S. Если у вас установлен ЖКИ дисплей, то вы можете ознакомиться с демонстрационным проектом. На первой ступени проекта производится вывод 4-х логотипов:

- LDM-SYSTEMS – разработчик отладочных средств;
- ЗАО «ПКС Миландр» – разработчик микроконтроллеров;

- ООО «Фирма Фитон» – разработчик программного обеспечения;
- ООО «МЭЛТ» – производитель ЖКИ дисплеев

Для перехода в основное меню демонстрационного проекта, нажмите на среднюю кнопку «SEL». На дисплее высветится меню с несколькими разделами и подразделами, которые демонстрируют работу контроллера LDM-K1986BE92QI и графические возможности ЖКИ дисплея MT-12864J.

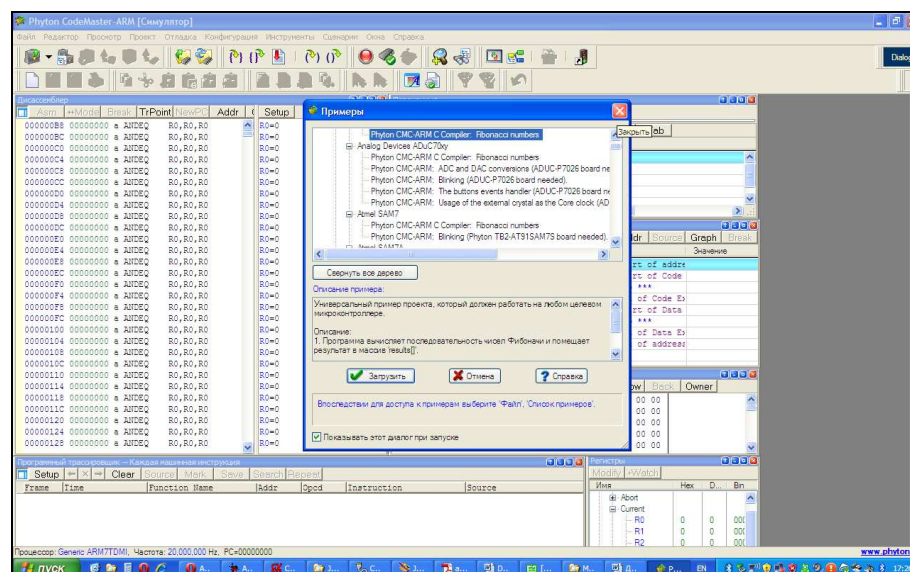
Если у вас не установлен ЖКИ дисплей, то работоспособность платы можно проверить следующим образом. После включения питания произведите последовательное нажатие на кнопки с периодичностью в 2 сек.: «SEL» → «DOWN» → «DOWN» → «SEL» → «SEL».

Если все было сделано верно, то на светодиоды VD5, VD4, VD7, VD6 будет производиться вывод запрограммированной комбинации световых импульсов. Если повторно нажать на «SEL», то светодиоды погаснут.

2.3.3 Открытие демонстрационного проекта в среде Phyton

Шаг 1: Запуск компилятора CodeMaster-ARM.

Заходим в меню «Пуск» → «Все программы» → «Phyton CodeMaster-ARM Demo 2.32.00» → CodeMaster-ARM. Откроется окно.



Закрываем окно «Примеры».

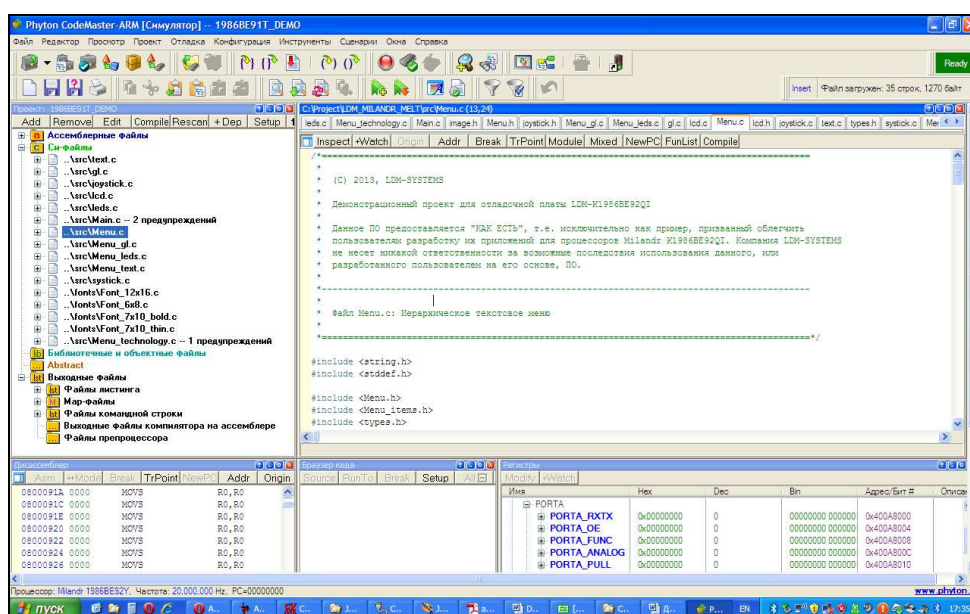
Шаг 2: Копирование папки с демонстрационным проектом с CD-R.

Произведем копирование папки с демонстрационным проектом с CD-R диска (CD-R диск\Примеры проектов\LDM_MILANDR_MELT) на жесткий диск.

Шаг 3: Открытие демонстрационного проекта, сборка проекта.

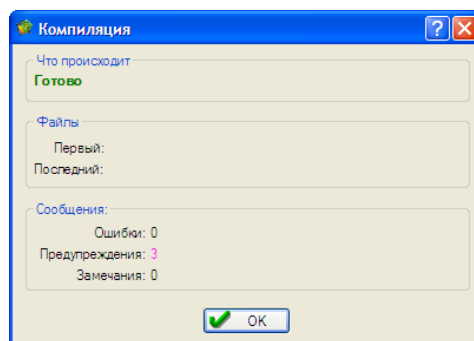
Нажимаем на кнопку «Открыть проект» . В открывшемся окне указываем путь к файлу *.IDE скопированного демонстрационного проекта: \LDM_MILANDR_MELT\CMCARM\1986BE91T_Demo.IDE.

Произойдет открытие проекта.



Шаг 4: Сборка (компиляция) проекта.

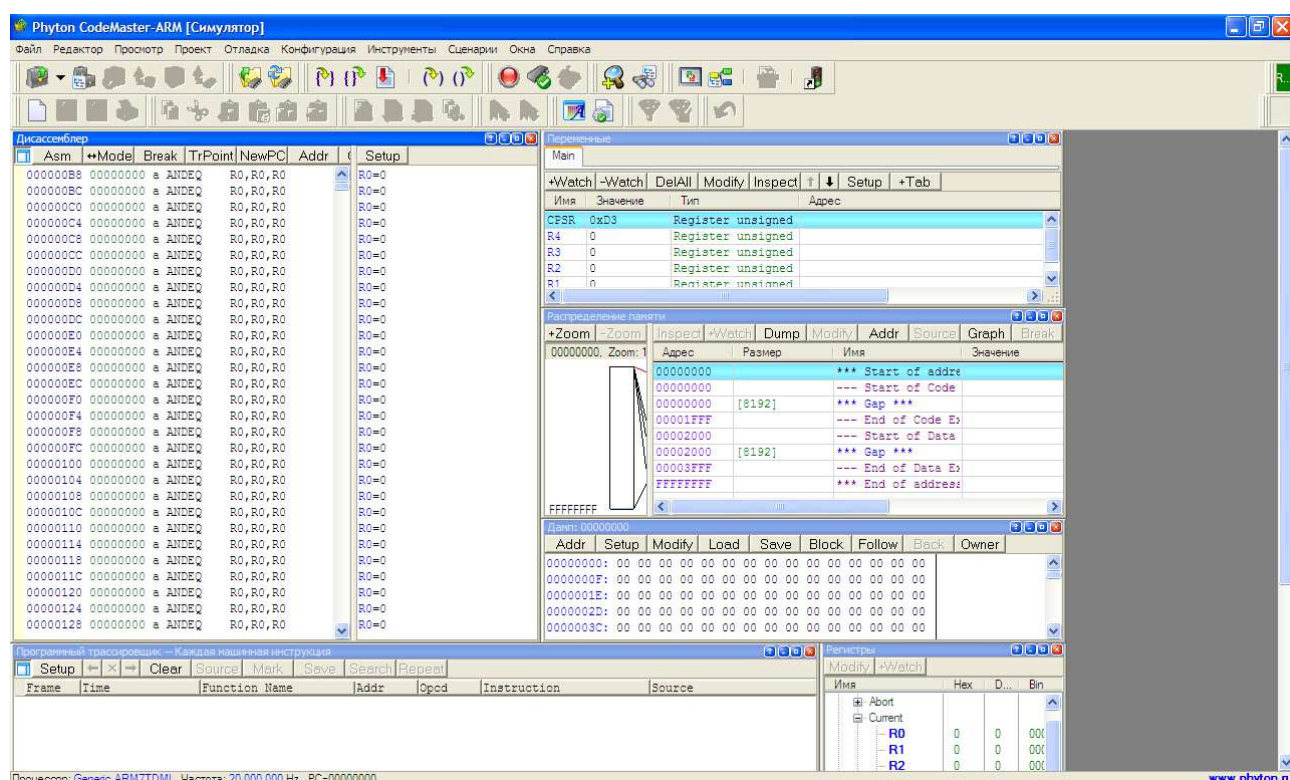
Нажимаем на кнопку «Пересобрать весь проект» . Ожидаем завершения сборки и нажимаем «ОК».



2.3.4 Создание простого проекта в среде Phytон

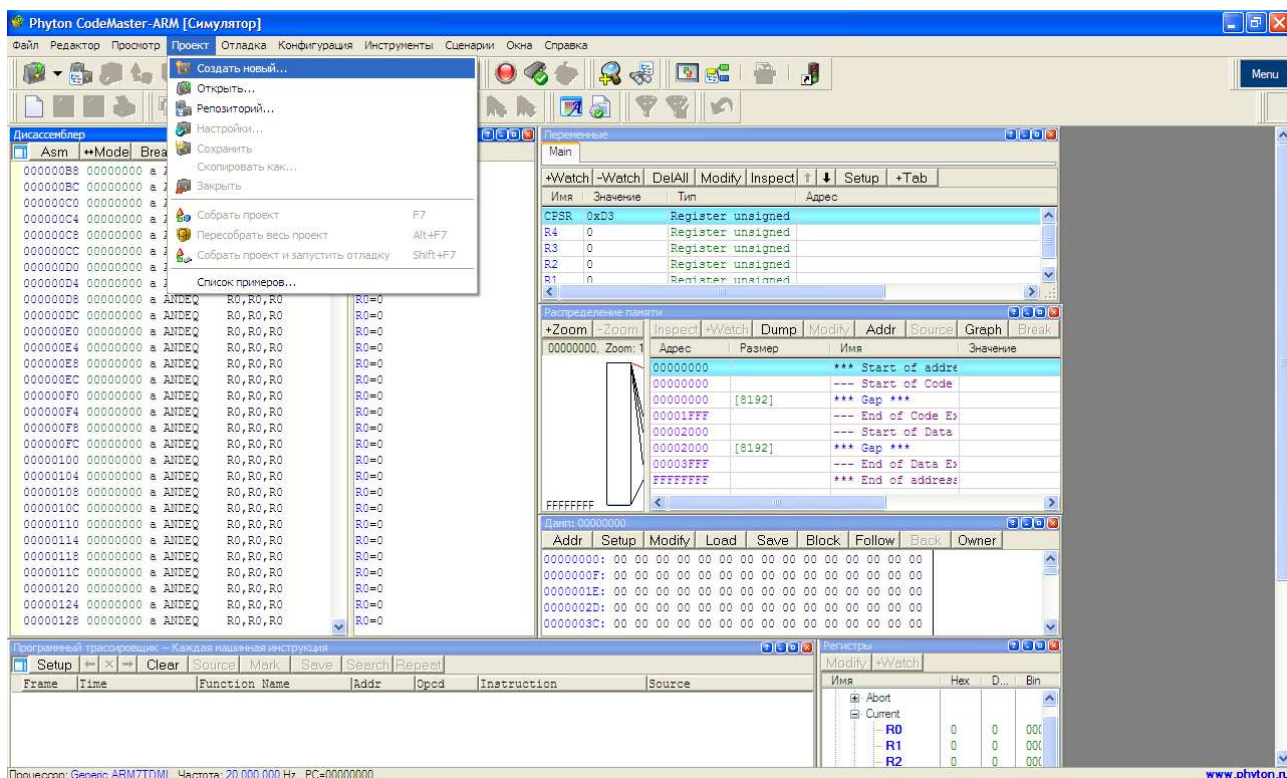
Цель проекта: Создадим проект в среде проектирования **Phyton**. Проект будет выполнять следующие задачи: при нажатии на кнопку «**SEL**» на плате **LDM-K1986BE92QI**, подключённой к порту **PC2**, должен загораться светодиод **VD5**, подключённый к порту **PB0**. При отпускании кнопки «**SEL**» светодиод **VD5** будет гаснуть.

Шаг 1: Запускаем компилятор из меню «**Пуск**». Откроется следующее окно:

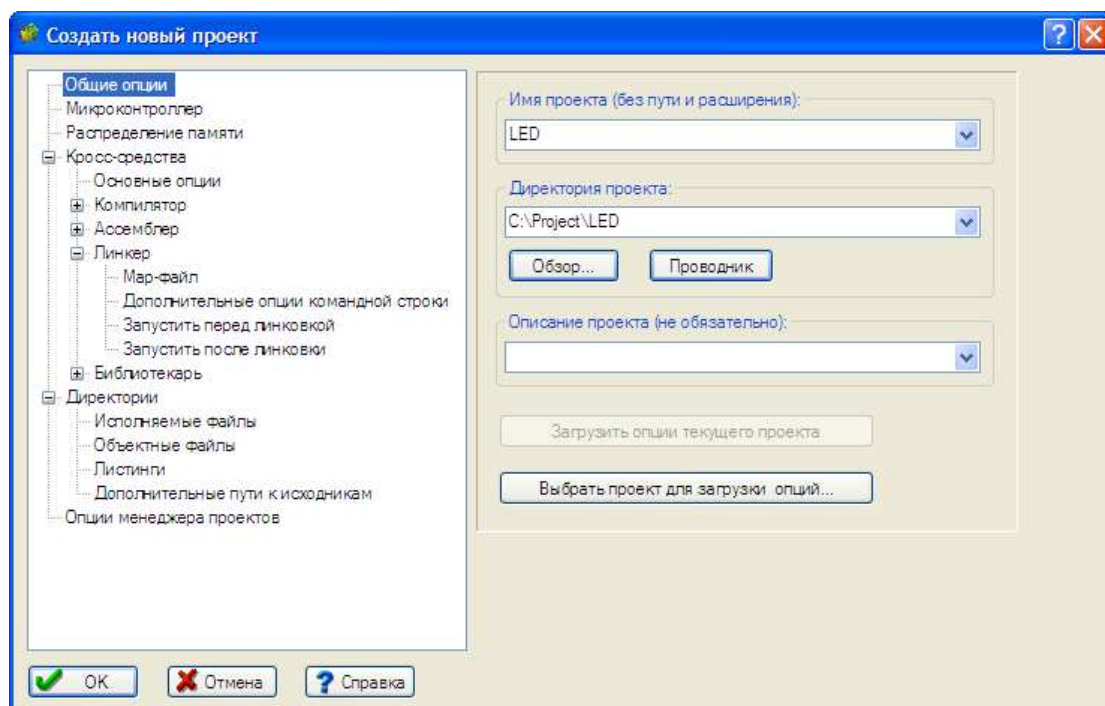


Шаг 2: Создаём новый проект.

Для этого на вкладке «**Проект**» программы выбираем «**Создать новый проект...**»

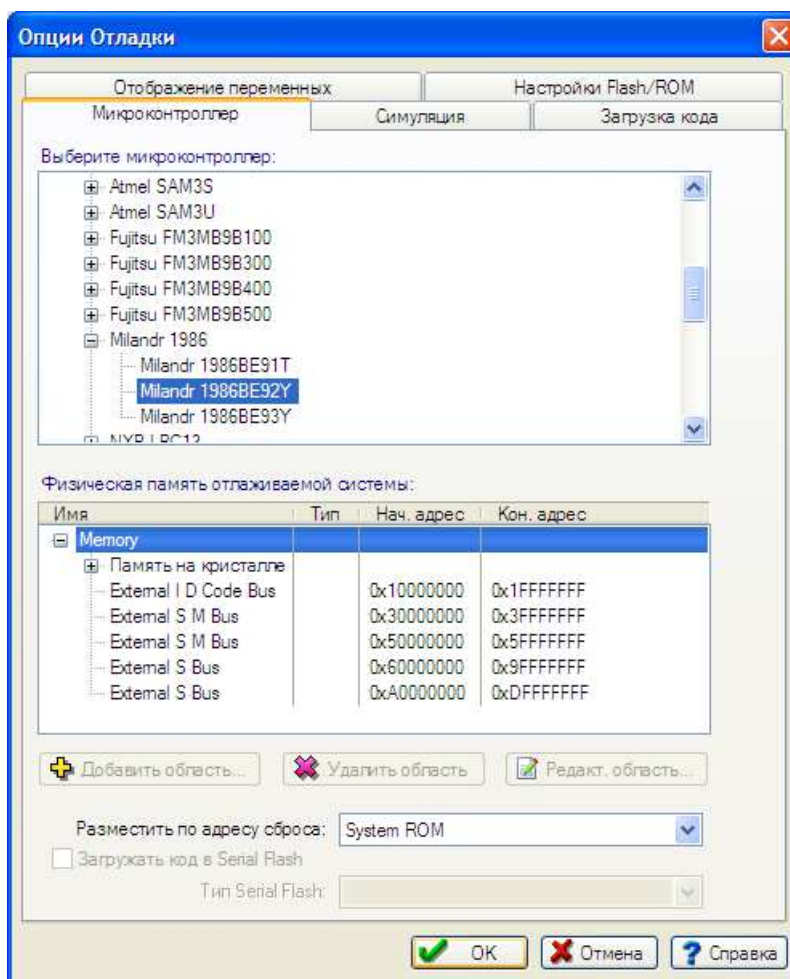


В открывшемся окне указываем: имя проекта **LED** и директорию проекта (Например: **C:\Project\LED**):

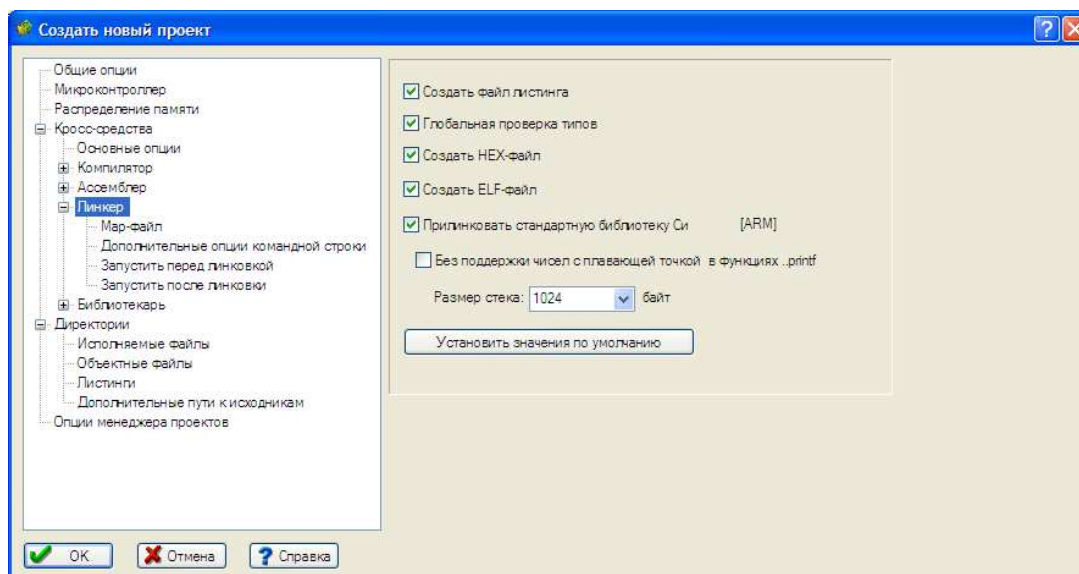


Далее выбираем в дереве настроек проекта позицию «Микроконтроллер» и нажимаем кнопку справа «Опции»

микроконтроллера...». В открывшемся окне выбираем раздел **Cortex-M3Milandr 1986Milandr 1986BE92Y**:



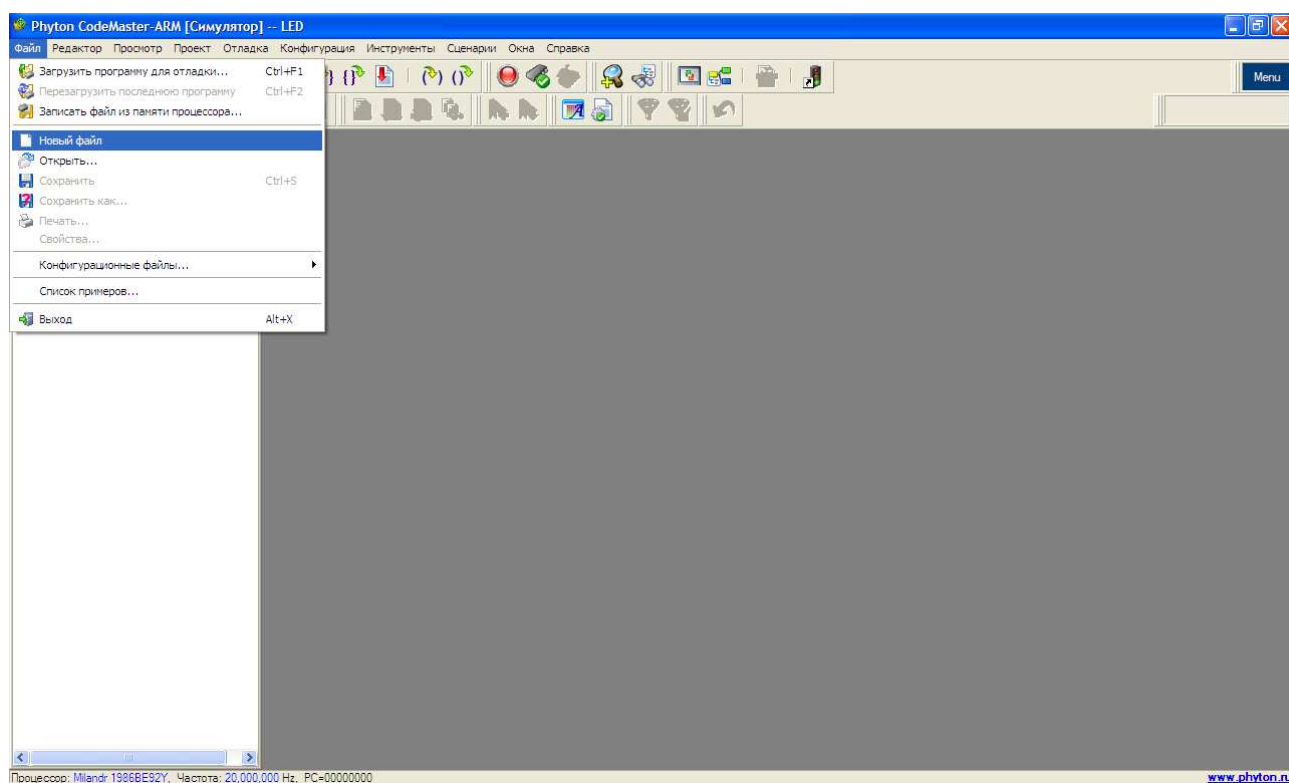
Нажимаем кнопку «**OK**» и переходим к пункту «**Линкер**». Ставим флаг «**Создать HEX-файл**»:



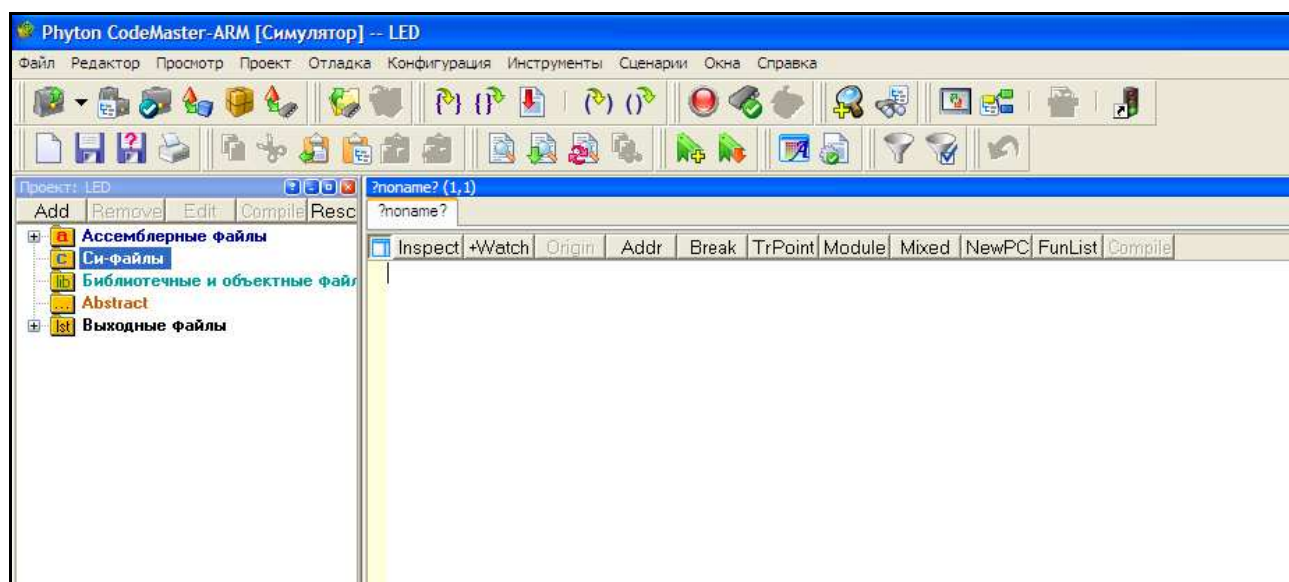
Нажимаем «**OK**».

Шаг 3: Создание главного файла проекта **main.c**

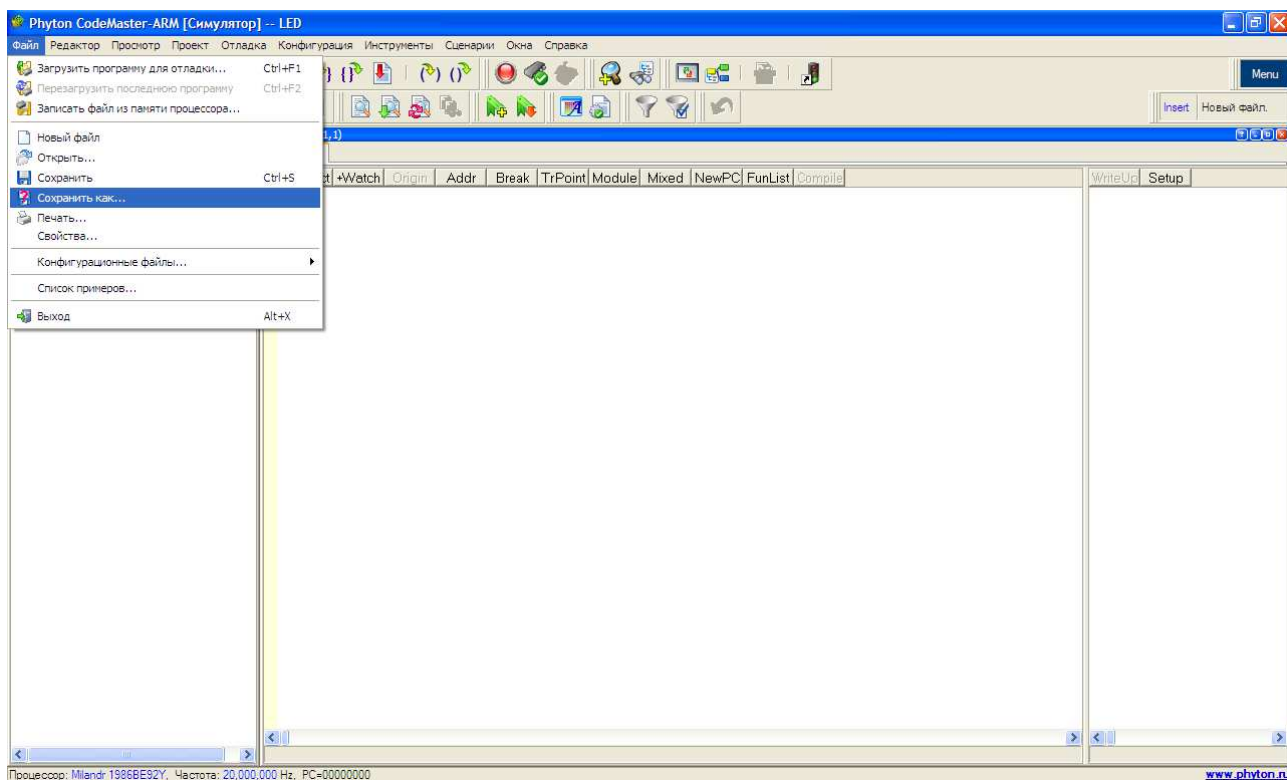
Создадим файл, в котором будет основной код программы. Для этого в меню «**Файл**» выбираем «**Новый файл...**»:



В окне слева появится белое поле с именем «**?noname?**»:

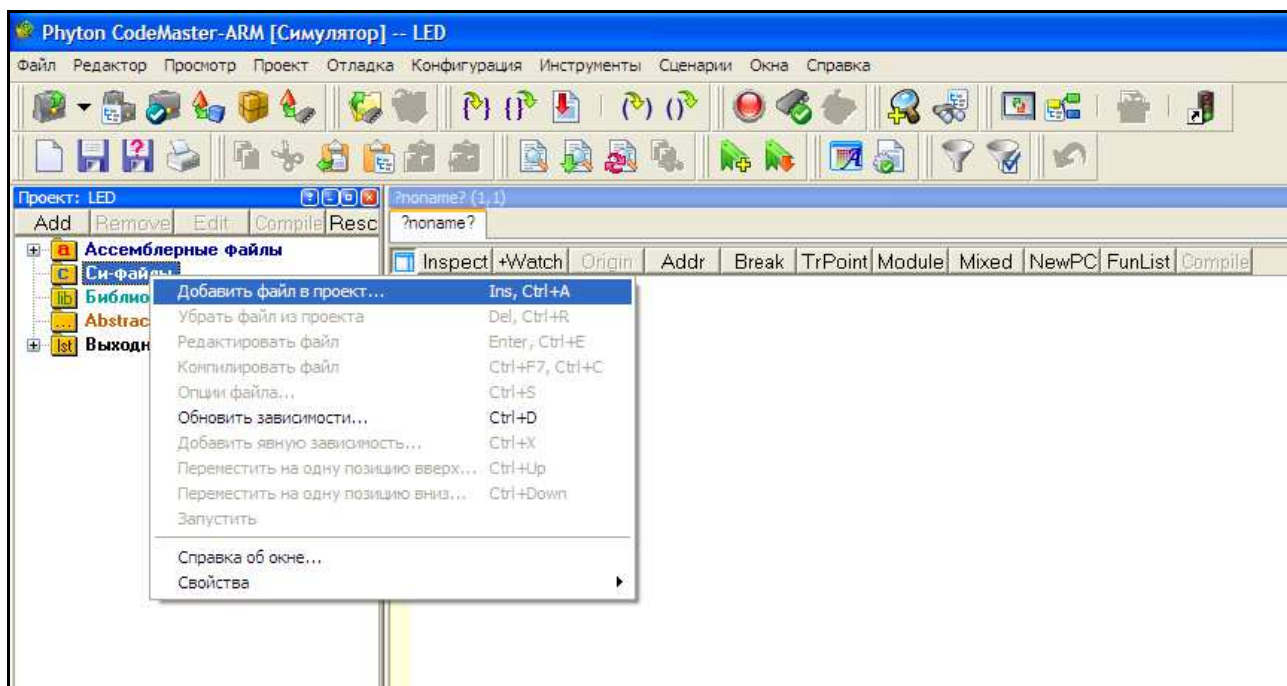


В меню «**Файл**» выбираем «**Сохранить как...**»:

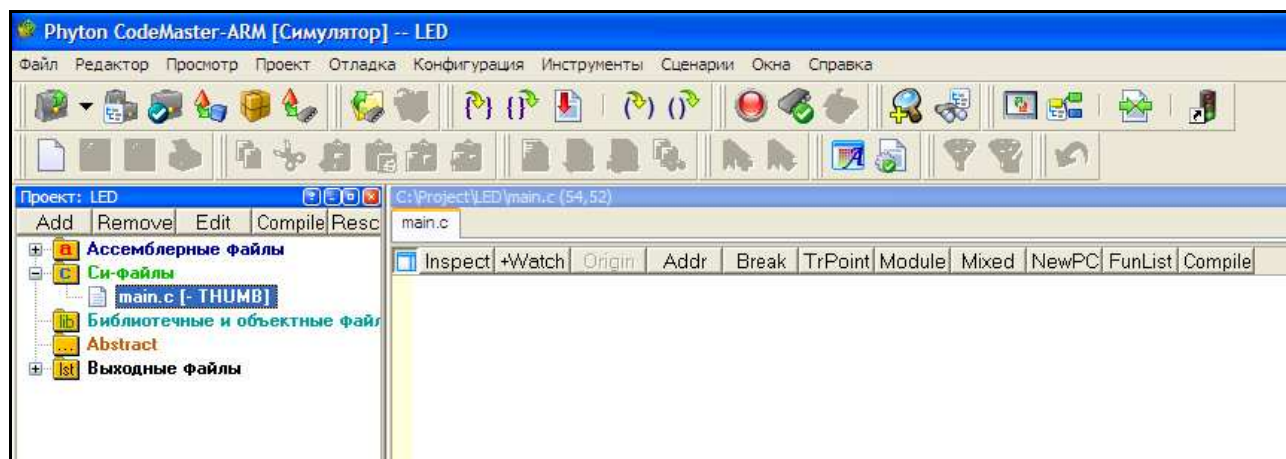


После чего появится окошко, в котором пишем «**main.c**» и нажимаем кнопку «**Сохранить**».

Теперь необходимо файл, сохраненный как «**main.c**», добавить в проект. Для этого жмем правой кнопкой мыши на папку «**Си-файлы**» и в выпадающем меню выбираем «**Добавить файл в проект...**»:

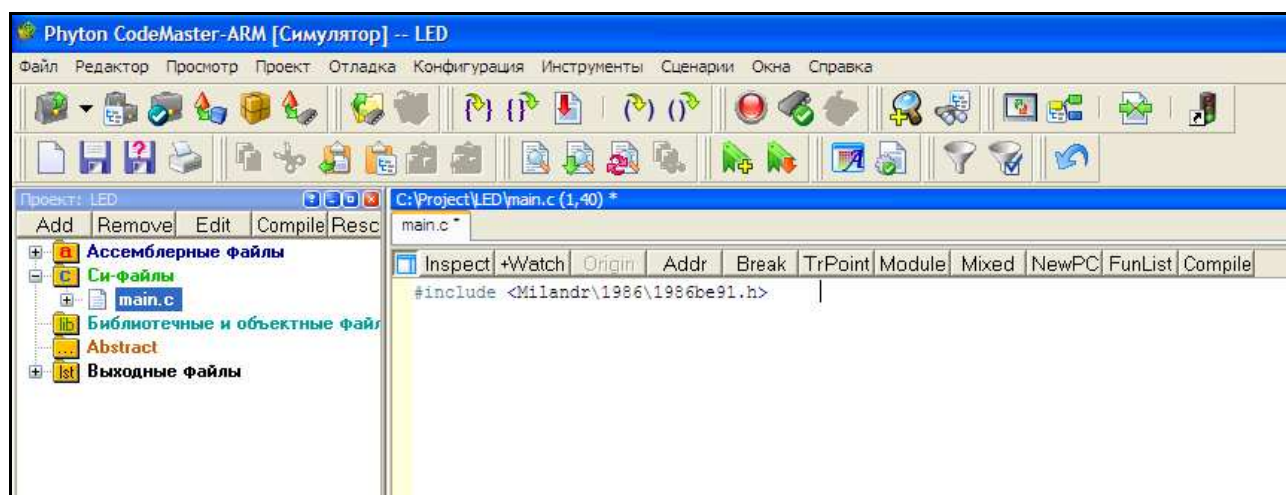


В открывшемся окне выбираем «**main.c**» и нажимаем кнопку «Открыть». В результате, в папке «**Си-файлы**» появится файл «**main.c**»:



Шаг 4: Создаем программный код проекта в файле **main.c**.

Добавляем в файл «**main.c**» заголовочный h-файл семейства микроконтроллера «**1986be91.h**». Введем текстовую строку в файл «**main.c**» **#include <Milandr\1986\1986be91.h>**:



Далее добавляем в наш файл main следующий код:

```
//=====
// Данный код предоставлен пользователем «редактор» на forum.milandr.ru
// и переделан для
// платы LDM-1986BE92QI компании LDM-SYSTEMS
//=====

#include <Milandr\1986\1986BE91.h>

#define BUTTON_SELECT (1<<2)           // Кнопка Select заведена на порт PC2
#define LAMP_VD5      (1<<0)           // Светодиод VD5 заведен на порт PB0
#define MDR_PORTC ((_port*) BASE_PORTC)
#define MDR_PORTB ((_port*) BASE_PORTB)
#define MDR_RST_CLK ((_rst_clk*) BASE_RST_CLK)

void main (void)
{
    // Разрешили работу всей периферии
    // Если этого не сделать, порты не будут инициализированы
    MDR_RST_CLK->PER_CLOCK = 0xFFFFFFFF;

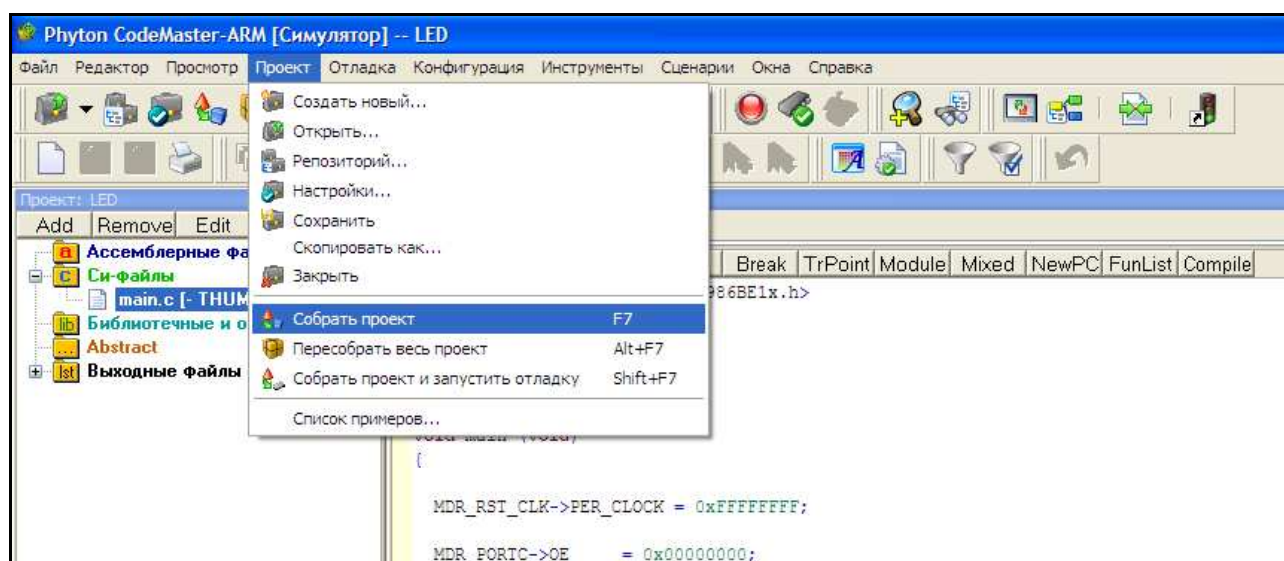
    // Настройка портов ввода-вывода
    // Настраиваем порт PC
    MDR_PORTC->OE      = 0x00000000;      // Все линии порта C настраиваем на ввод
    MDR_PORTC->FUNC     = 0x00000000;      // Все линии порта используются как порт ввода-вывода
    MDR_PORTC->ANALOG  = 0x0000FFFF;      // Все линии - цифровые
    MDR_PORTC->PULL    = BUTTON_SELECT << 16; // Линию PC2 притянули к положительному уровню питания
    MDR_PORTC->PD      = BUTTON_SELECT << 16; // Включили триггер Шмидта по входу линии PC2
    MDR_PORTC->PWR     = 0x55555555;      // Медленный фронт по входу всем линиям
    MDR_PORTC->GFEN    = 0;               // Фильтр по входу выключен по всем линиям

    // Настраиваем порт PB
    MDR_PORTB->OE      = LAMP_VD5;        // Линию порта PB0 настраиваем на вывод
    MDR_PORTB->FUNC     = 0x00000000;      // Все линии порта используются как порт ввода-вывода
    MDR_PORTB->ANALOG  = 0x0000FFFF;      // Все линии - цифровые
    MDR_PORTB->PULL    = LAMP_VD5 << 16;   // Линию PB0 притянули к положительному уровню питания
    MDR_PORTB->PD      = ~LAMP_VD5;        // Линия PB0 настраивается как управляемый драйвер
    MDR_PORTB->PWR     = 0x55555555;      // Медленный фронт по выходу линии PB0
    MDR_PORTB->GFEN    = 0;               // Фильтр по выходу выключен по всем линиям
    MDR_PORTB->RXTX    = 0xFFFF;         // Выставили все линии порта в высокое состояние

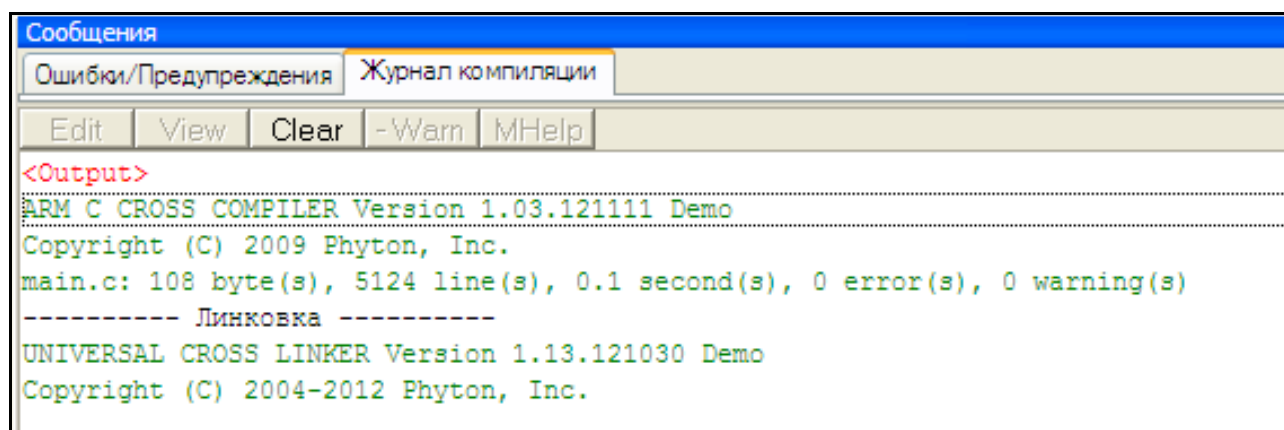
    while(1)                             // Основной цикл работы программы
    {
        if (MDR_PORTC->RXTX & BUTTON_SELECT) // Если бит установлен (кнопка отпущена)
        {
            MDR_PORTB->RXTX &= ~LAMP_VD5;    // Установили порт PB0 в 0 (низкий уровень выкл. светодиод)
        }
        else
        {
            MDR_PORTB->RXTX |= LAMP_VD5;      // Установили порт PB0 в 1 (высокий уровень вкл. светодиод)
        }
    }
}
```

Шаг 5: Проверка проекта.

В меню «Проект» выбираем «Собрать проект» или жмём на соответствующую иконку на интерфейсе программы:

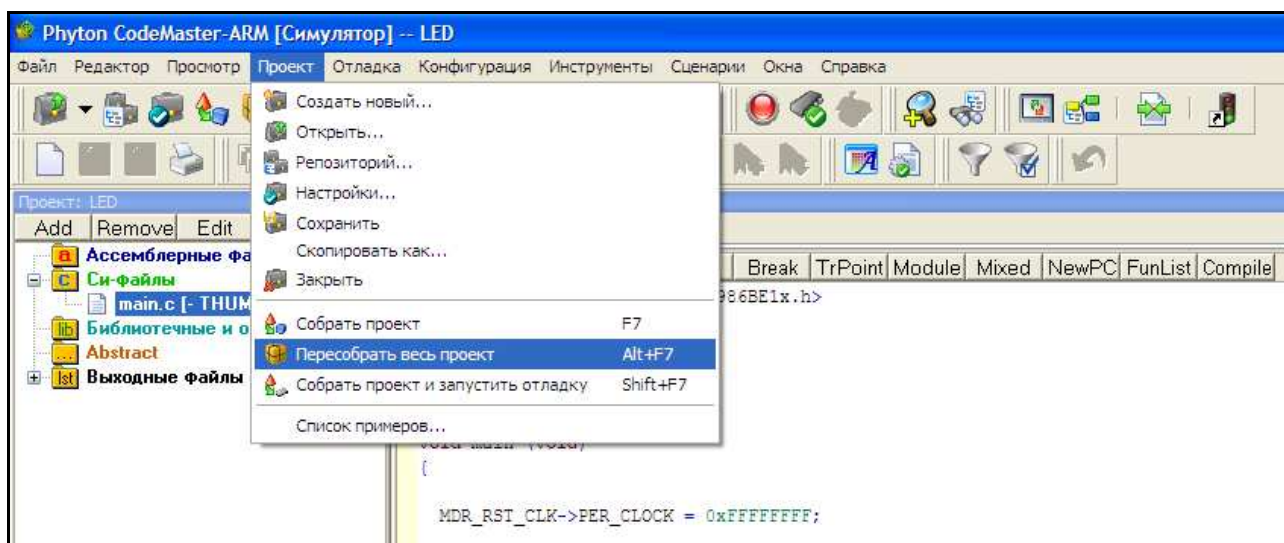


Наличие в проекте ошибки или предупреждения, будет выведено в нижнем окне программы «Сообщения»:



Шаг 6: Сборка проекта.

Для полной компиляции проекта необходимо в меню «Проект» выбираем «Пересобрать весь проект»:



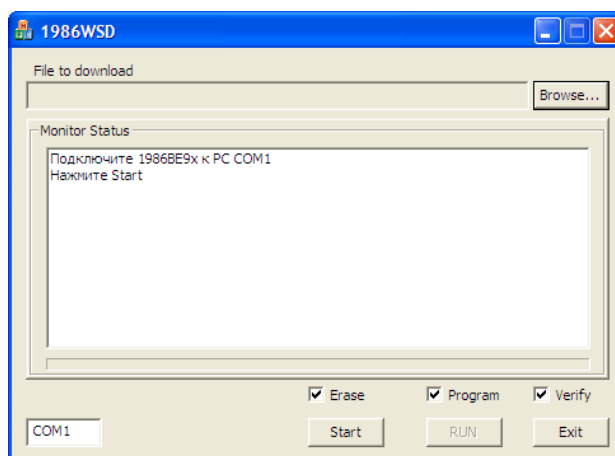
Для прошивки FLESH памяти микроконтроллера воспользуйтесь инструкцией раздела «2.3.5 Загрузка прошивки в FLASH микроконтроллера».

2.3.5 Загрузка прошивки в FLASH микроконтроллера

Шаг 1: Подготовительные работы.

Скопируем папку с CD-R диска «CD-R диск\USB-UART загрузчик» на жесткий диск. В скопированной папке запускаем файл «\1986UARTWSD\1986WSD.exe».

Если номер COM-порта не изменялся и не равен 1, то нужно указать номер порта, присвоенный устройству «Silicon Labs CP210x USB to UART Bridge (COM x)» в диспетчере устройств (см. п. 2.3.2, Шаг 2).



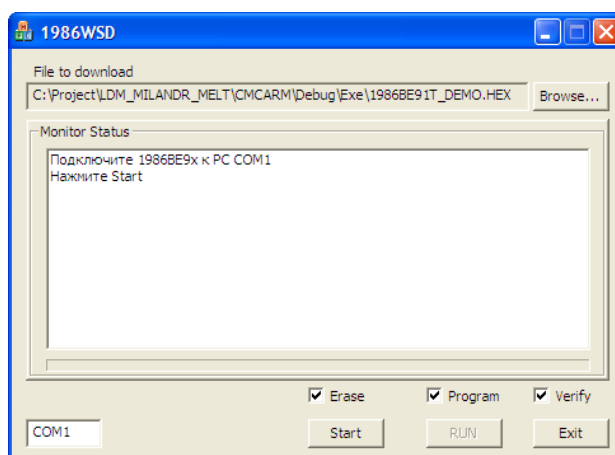
Шаг 2: Выбор файла и прошивка микроконтроллера.

Нажимаем кнопку «Browse...» и указываем путь к файлу *.HEX, созданного при компиляции:

«\LDM_MILANDR_MELT\CMCARM\Debug\Exe\1986BE91T_DEMO.HEX»

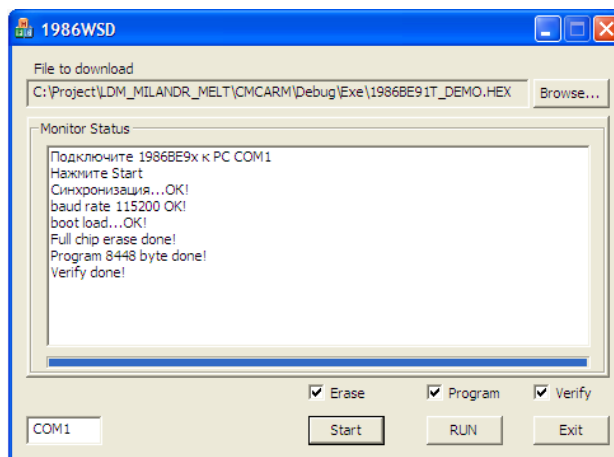
или

«\LED\LED.HEX».



Производим переключение контроллера в режим «UDM mode». Выставляем переключатели SW6 = 1, SW9 = 0, SW10 = 1. Выключаем и включаем питание при помощи кнопки SW11 (ON/OFF).

Нажимаем в программе «1986WSD» на кнопку «Start». Все сообщения о ходе прошивки будут выведены в поле «Monitor status».



После прошивки необходимо выключить питание при помощи кнопки SW11 (ON/OFF), выставить SW6 = 0, SW9 = 0, SW10 = 0 и включить питание при помощи кнопки SW11 (ON/OFF).

P.S. Если вы используете демонстрационную версию компилятора, то созданный вами проект будет неполным, т.к. у демонстрационной версии компилятора имеется ограничение по размеру кода программ менее 8 кбайт.

Если вы хотите загрузить полную версию демонстрационного проекта, то укажите при прошивке путь к файлу «\LDM_MILANDR_MELT\CMCARM\Debug\Exe\LDM-K1986BE92QI_FULL.HEX».

Поздравляем!

Вы освоили навыки работы с отладочной платой **LDM-K1986BE92QI** в среде проектирования **Phyton** и можете самостоятельно попробовать создать свой проект!

Enjoy!

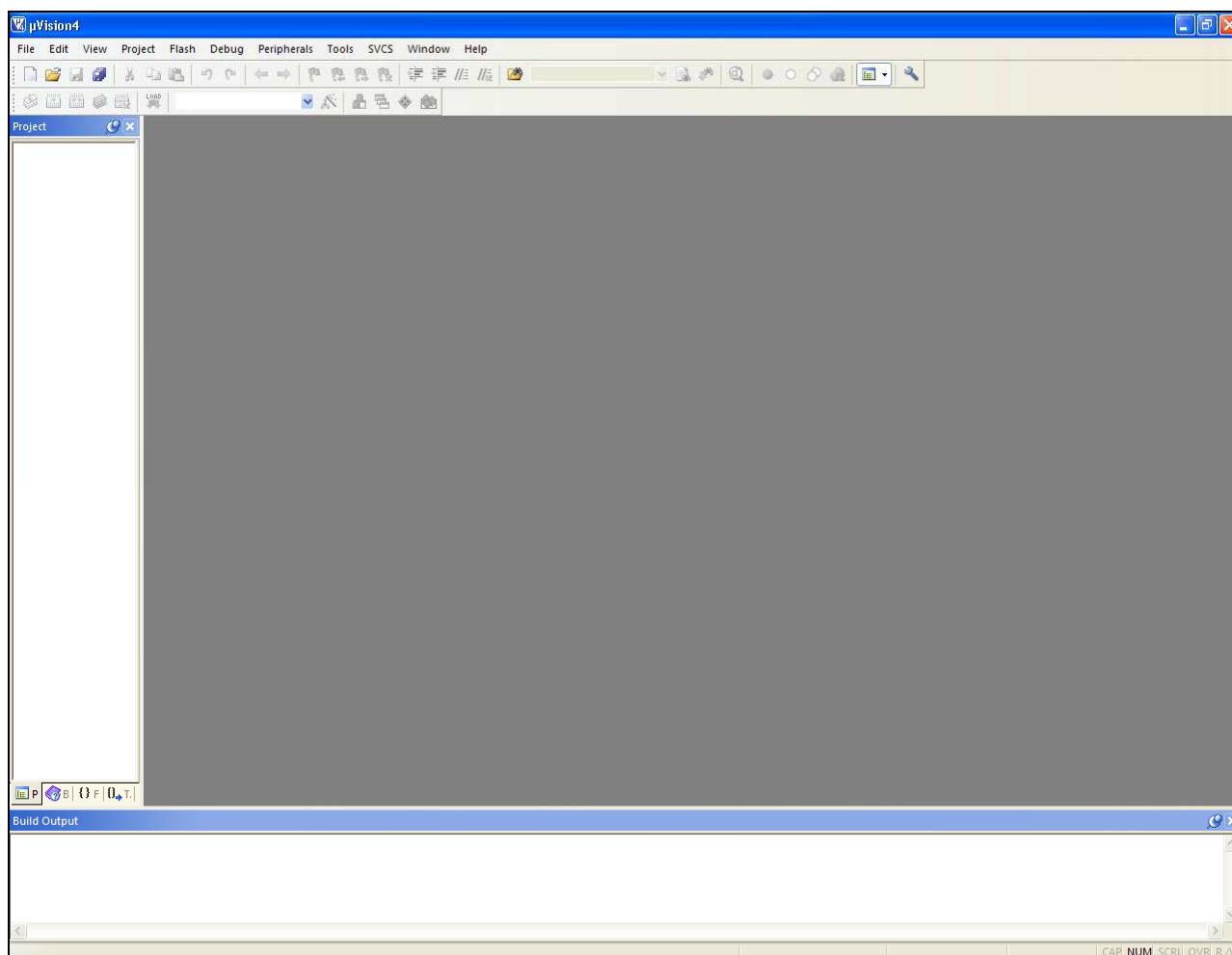
2.4 Создание простого проекта в среде Keil uVision

Цель проекта: Создадим проект в среде проектирования Keil uVision версии 4.72. Проект будет выполнять следующие задачи: при нажатии на кнопку «**SEL**» на плате **LDM-K1986BE92QI**, подключённой к порту **PC2**, должен загораться светодиод **VD5**, подключённый к порту **PB0**. При отпускании кнопки «**SEL**» светодиод **VD5** будет гаснуть.

Шаг 1: Устанавливаем компилятор Keil uVision версии 4.72.

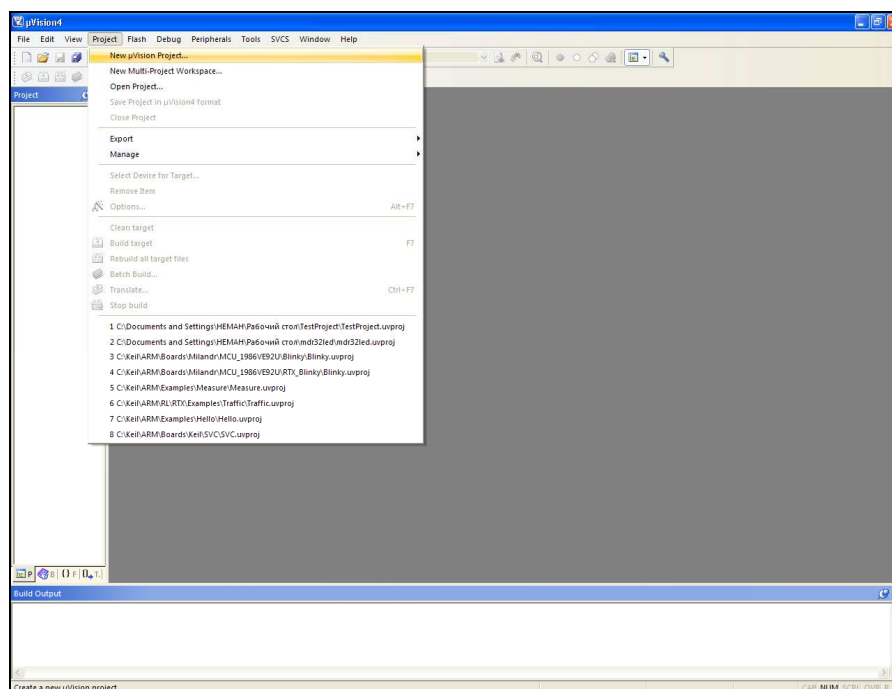
Дистрибутив компилятора можно найти на CD-R диске отладочной платы в разделе **Компилятор\Keil\mdk472_a.exe**. Запустите файл и произведите установку компилятора.

Шаг 2: Запускаем компилятор из меню «**Пуск**». Откроется следующее окно:

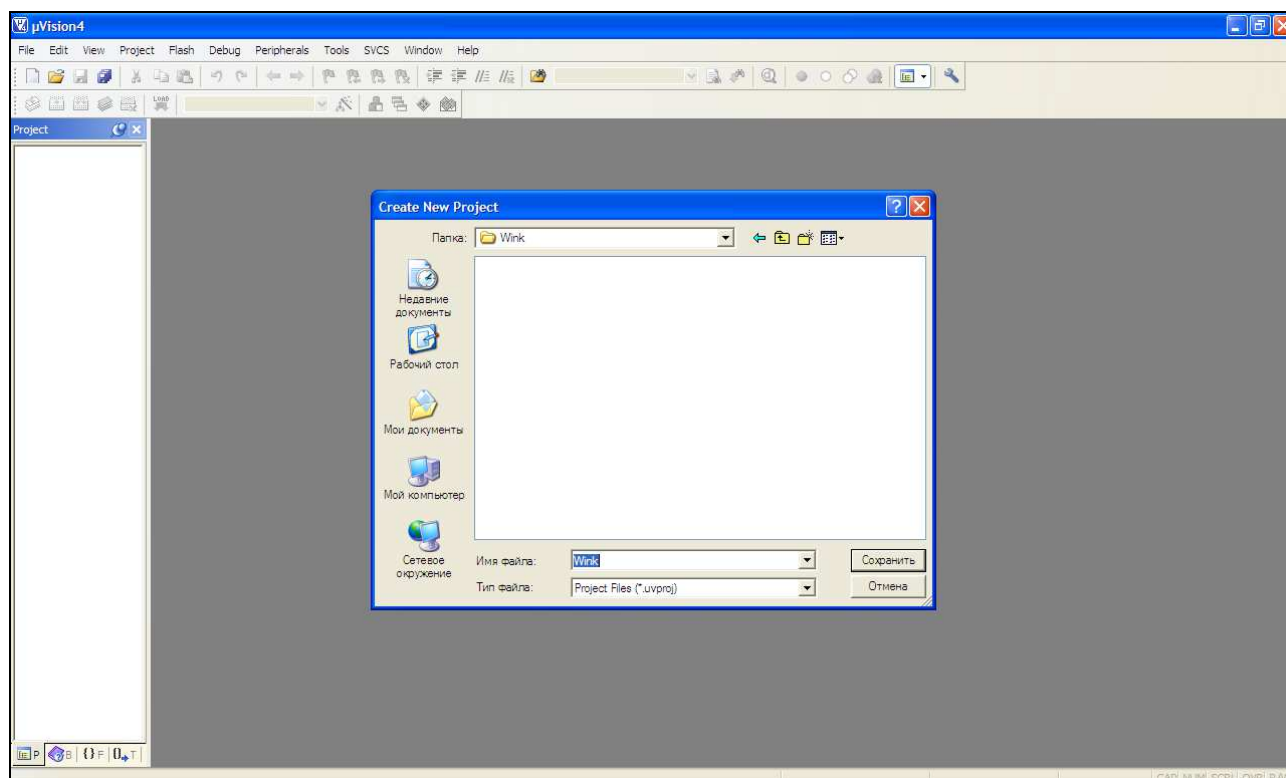


Шаг 3: Создаём новый проект.

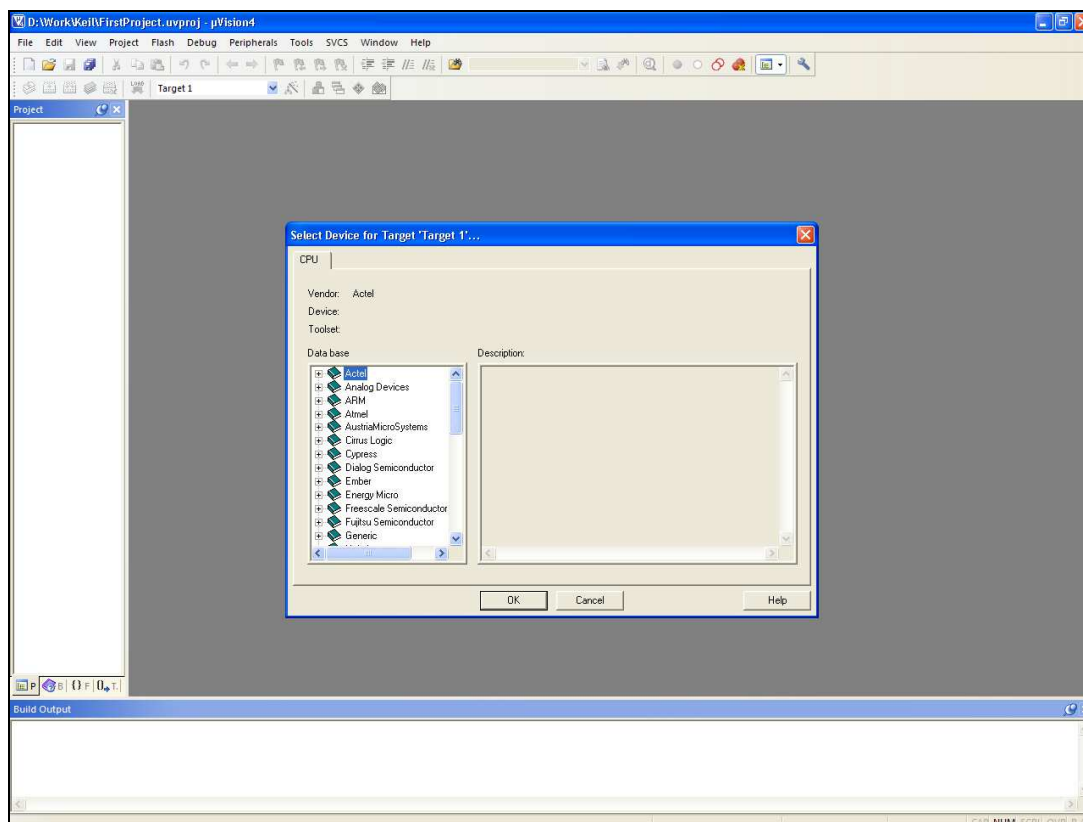
Для этого на вкладке «**Project**» программы выбираем «**New uVision Project...**»



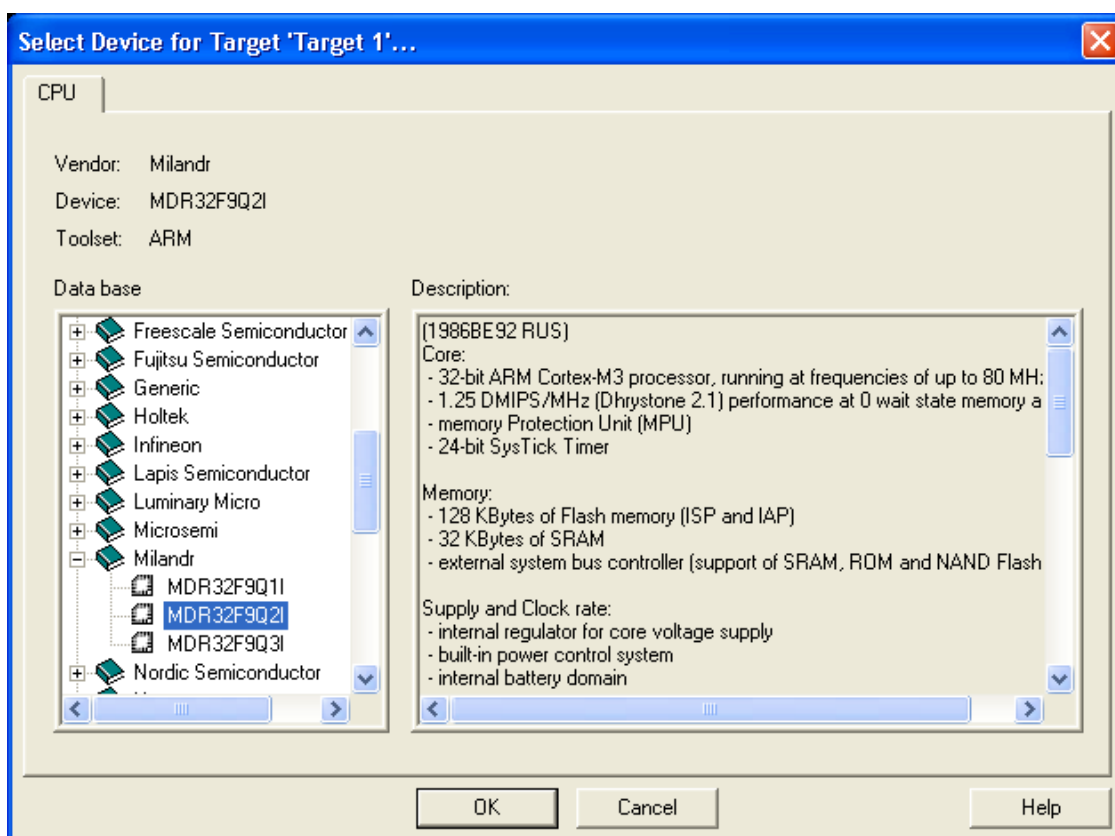
В открывшемся окне указываем: путь, где будет находиться проект (Например: **C:\Project\Wink**), название файла проекта **Wink.uvproj**. и нажимаем кнопку «**Сохранить**»:



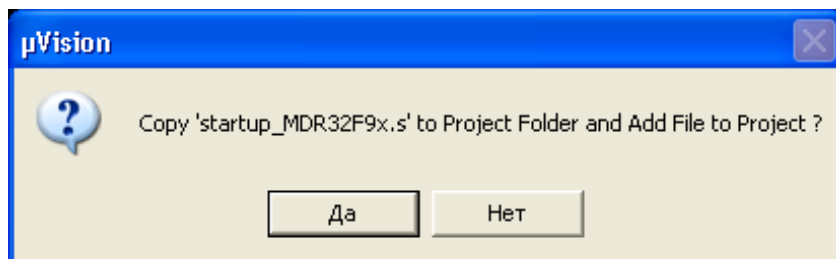
После этого появляется окно:



Выбираем производителя «Milandr» и микроконтроллер «MDR32F9Q2I».

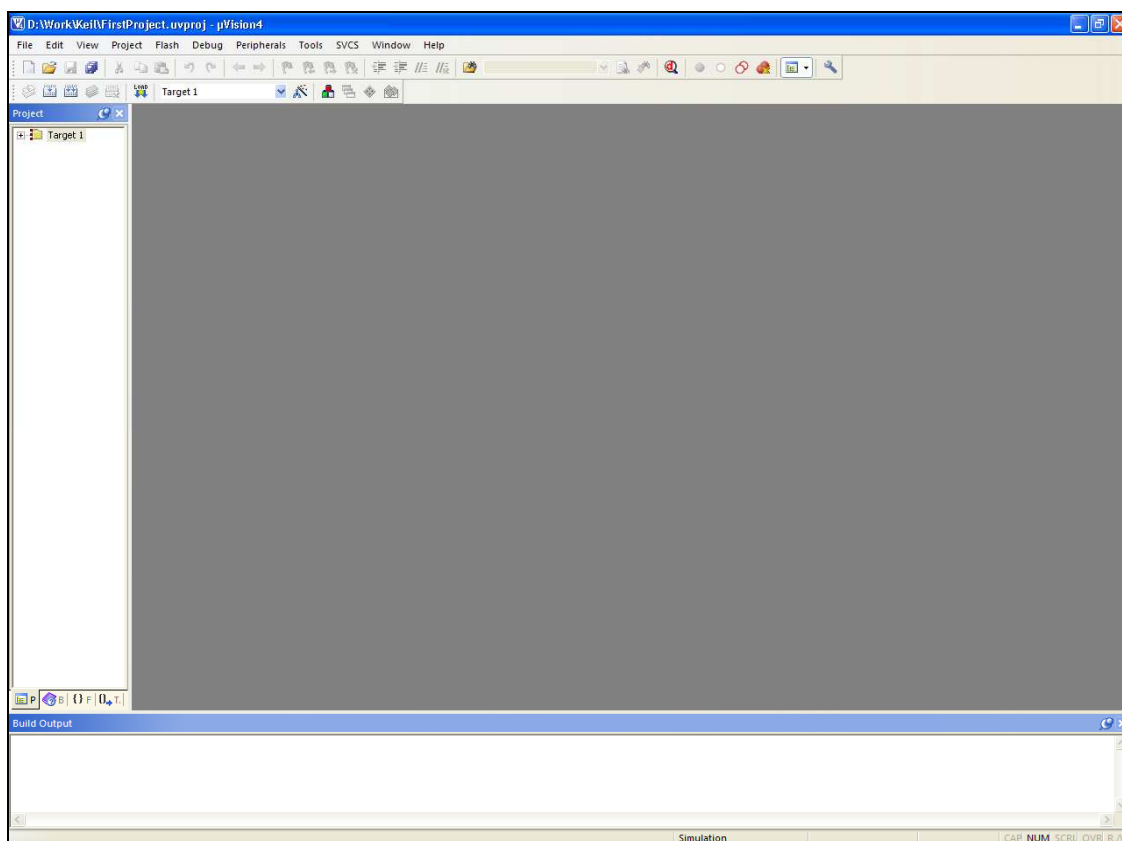


Нажимаем «**OK**», после чего появляется окно с вопросом:

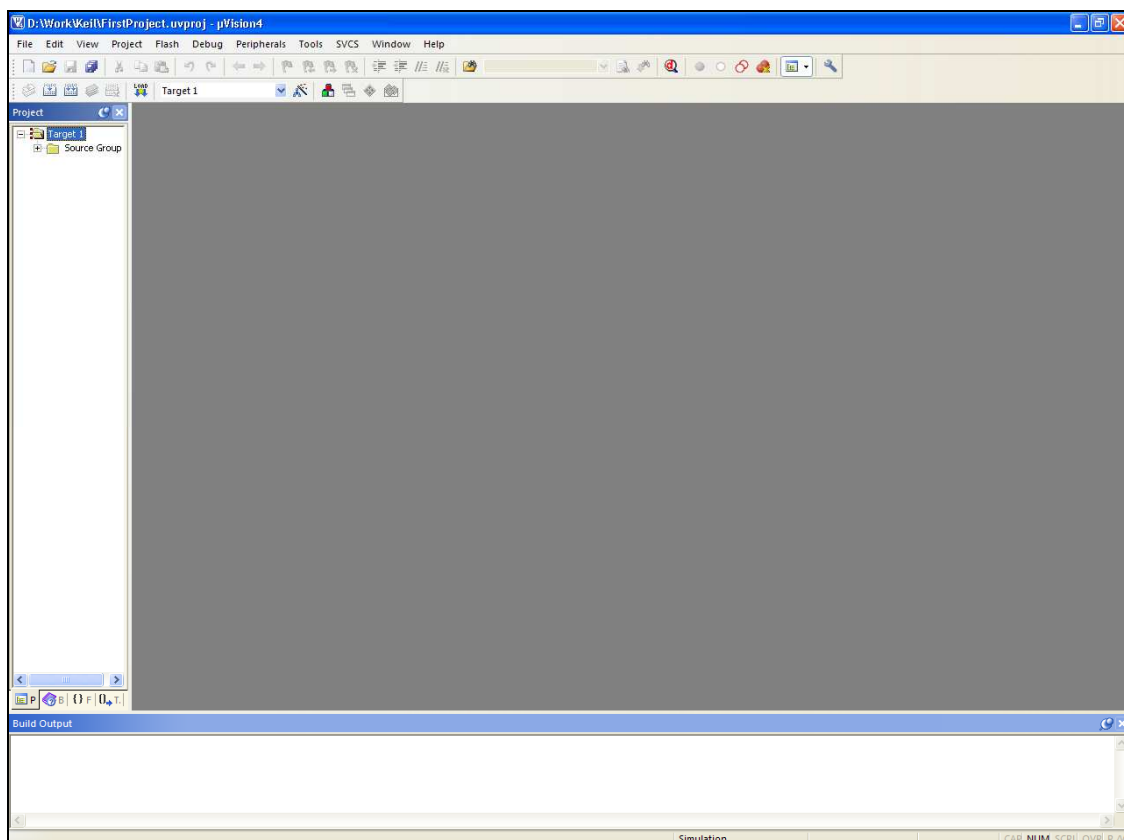


Компилятор предлагает создать файл инициализации контроллера, нажимаем кнопку «**Да**».

Слева, в окне программы «**Project**», появится папка «**Target 1**» с СИМВОЛОМ «**+**»:

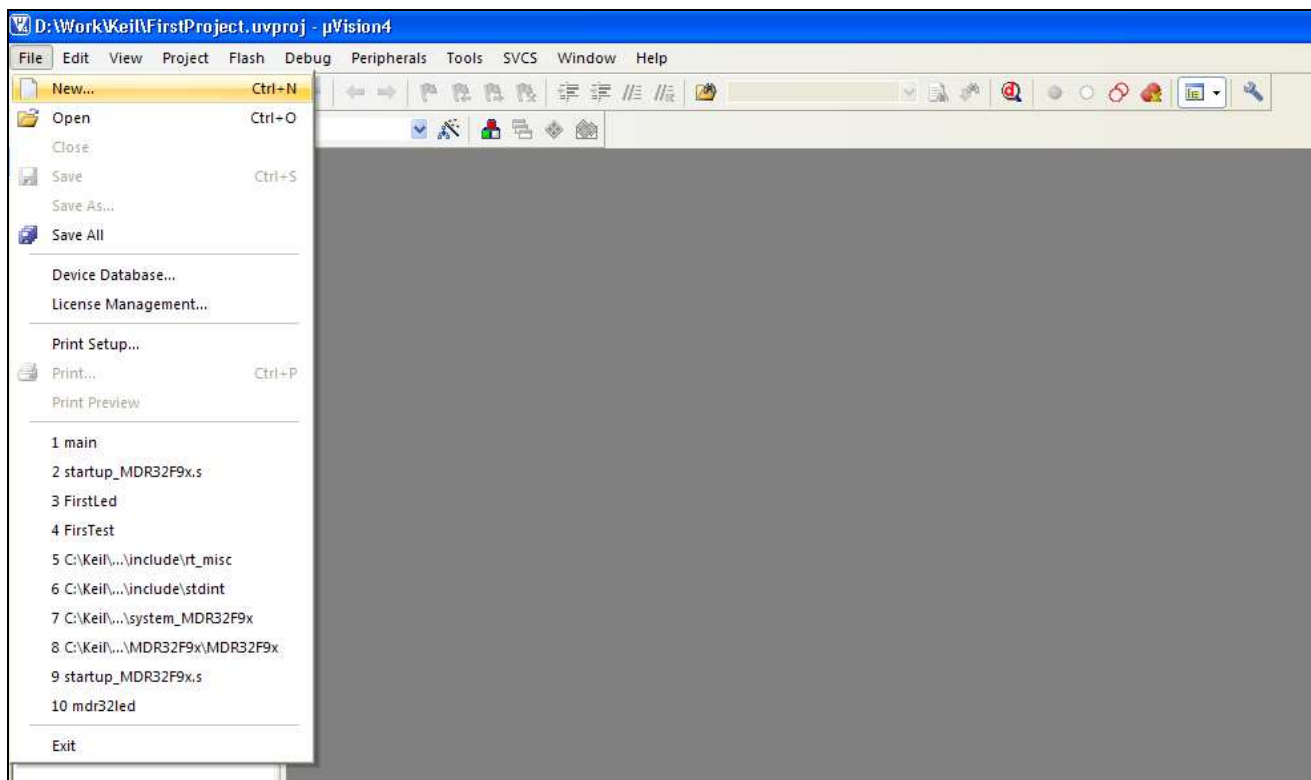


Нажимаем на символ «**+**», после чего, внутри папки «**Target 1**» появится папка «**Source Group 1**»:

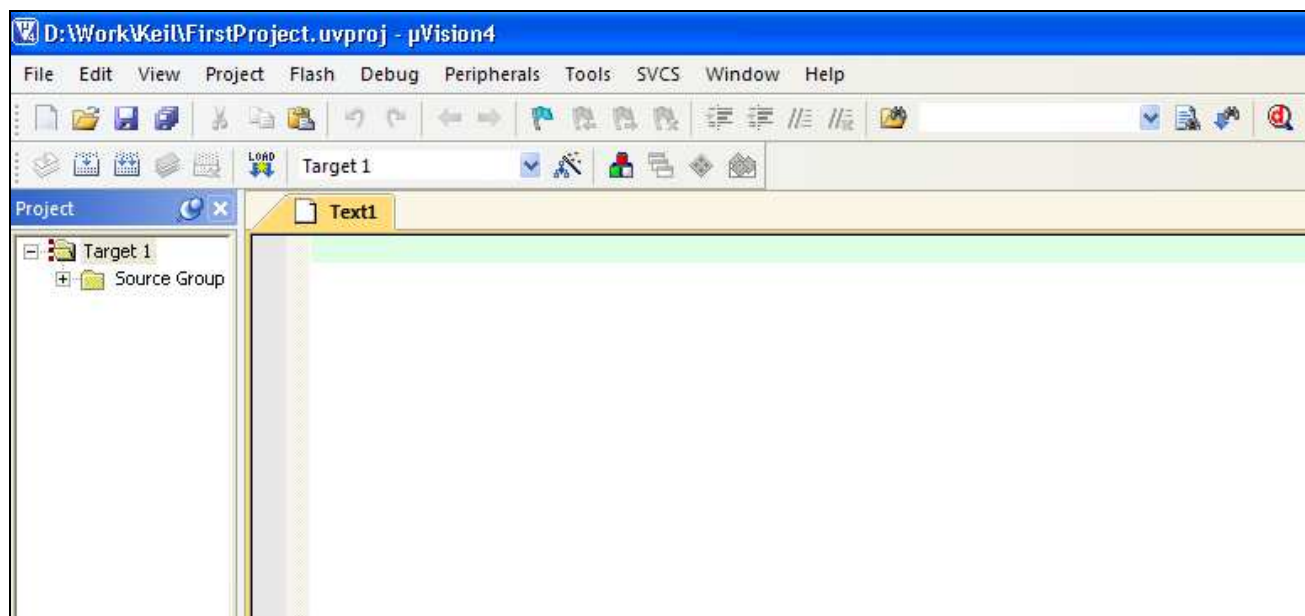


Шаг 4: Создание главного файла проекта **main.c**

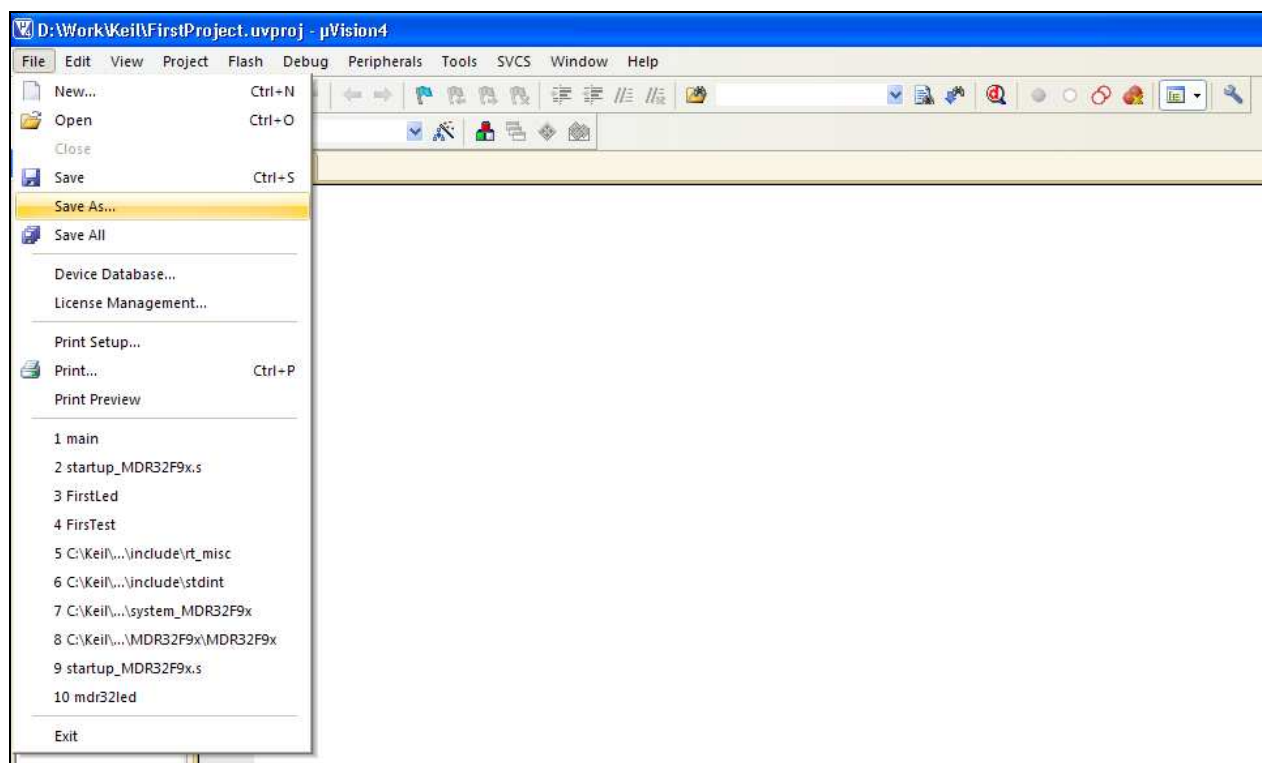
Создадим файл, в котором будет основной код программы. Для этого в меню «**File**» выбираем «**New...**»:



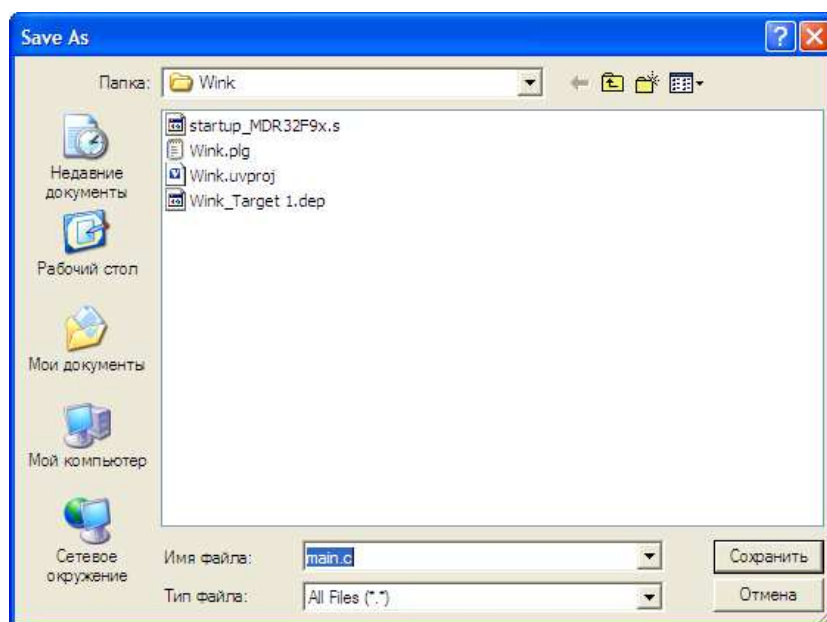
В окне слева появится белое поле с именем «**Text1**»:



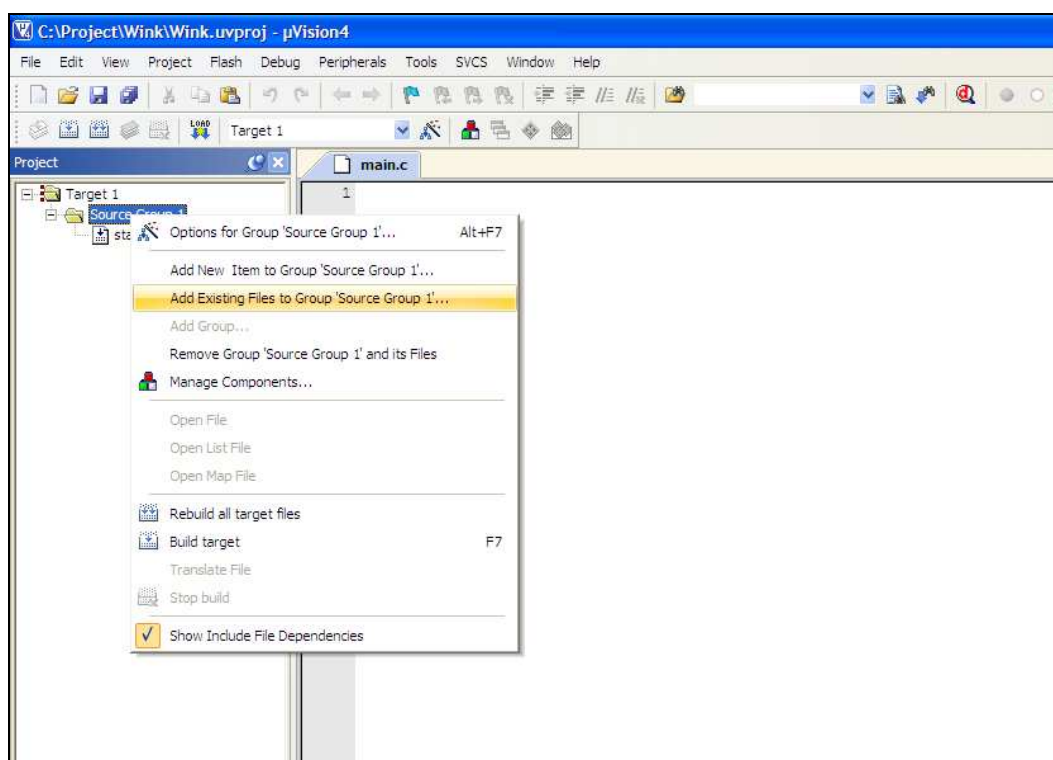
В меню файл выбираем «**Save As...**»:



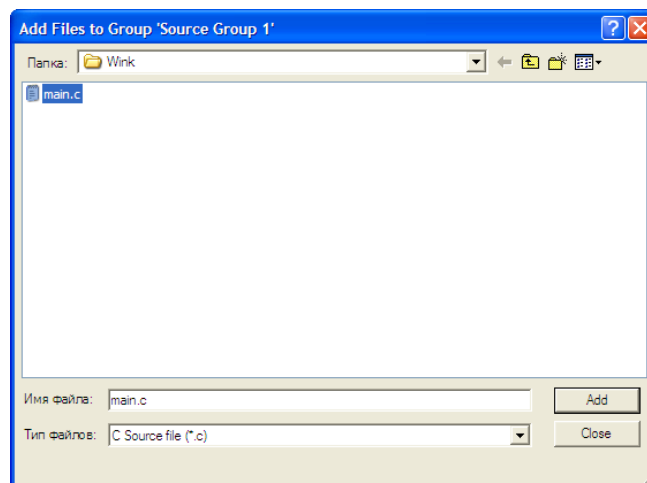
После чего появится окошко, в котором пишем «**main.c**»



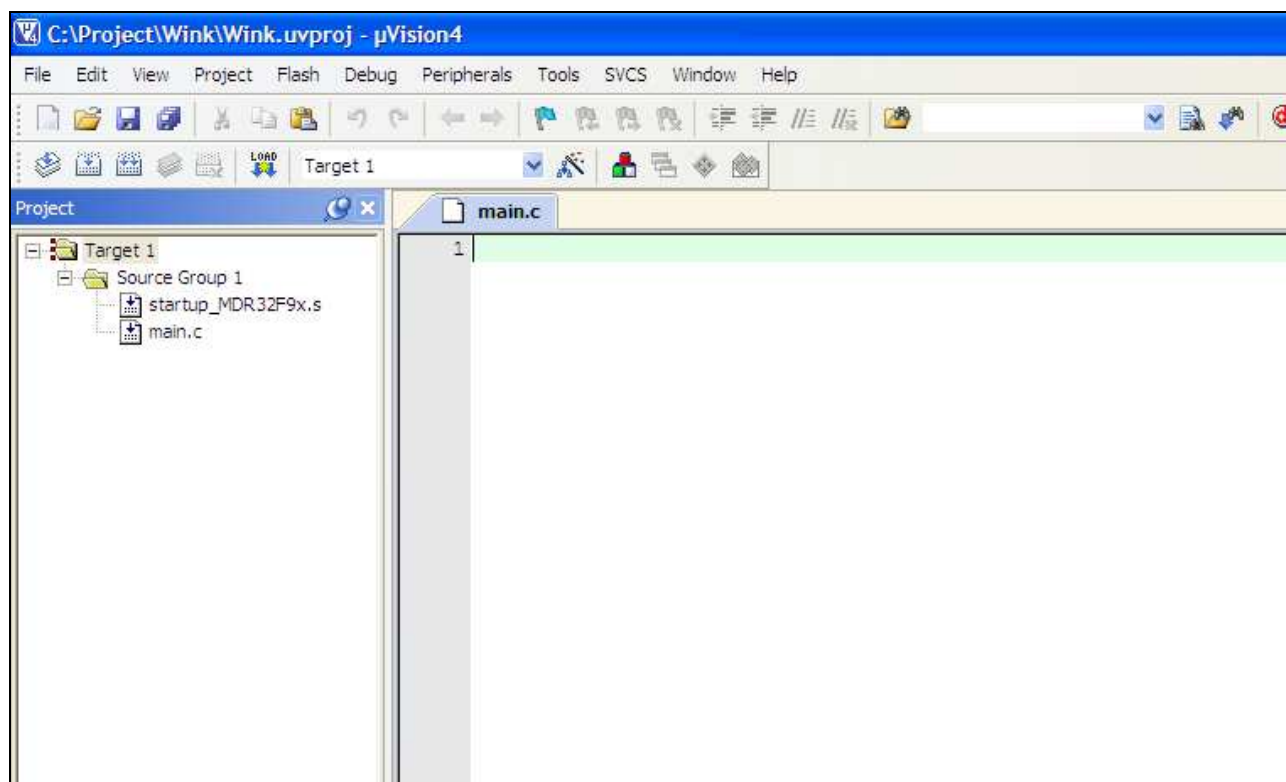
Теперь необходимо файл, сохранённый как «**main.c**», добавить в проект. Для этого жмём правой кнопкой мыши на папку «**Source Group 1**» и в выпадающем меню выбираем «**Add Existing Files to Group 'Source Group 1'**»:



Откроется окно:



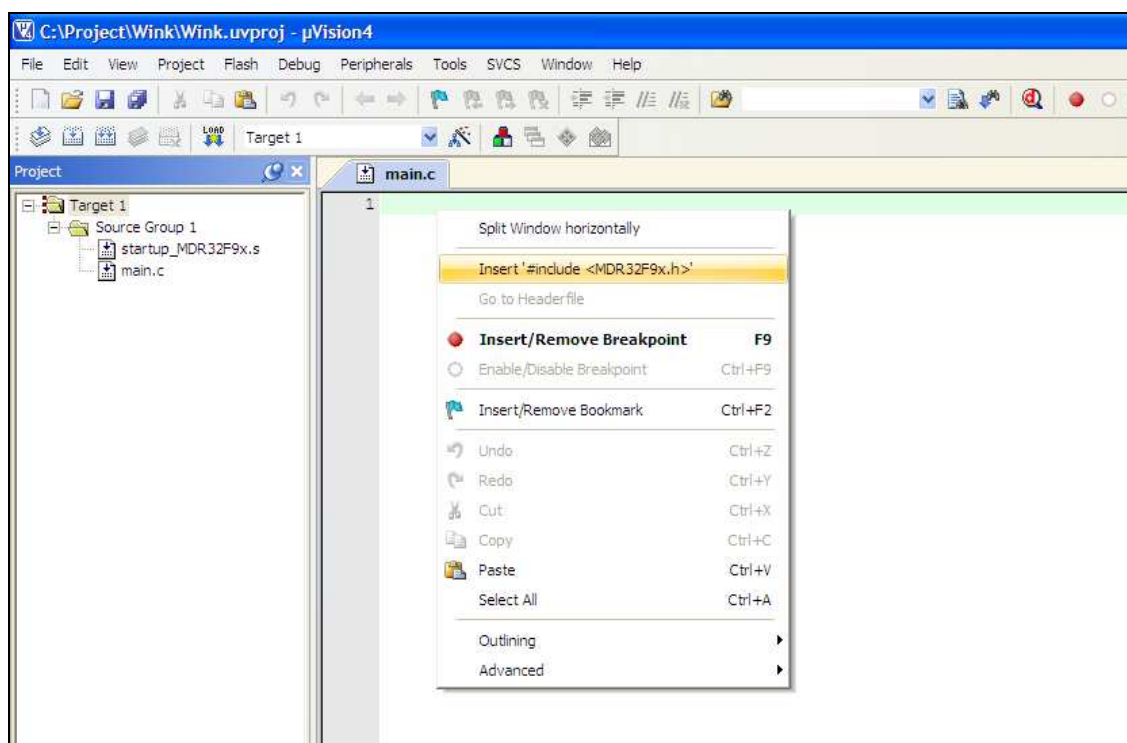
Выбираем «**main.c**» и нажимаем кнопку «**Add**». В результате, в папке «**Source Group 1**» видим уже два файла. Файл инициализации контроллера «**startup_MDR32F9x.s**» и созданный нами файл «**main.c**»:



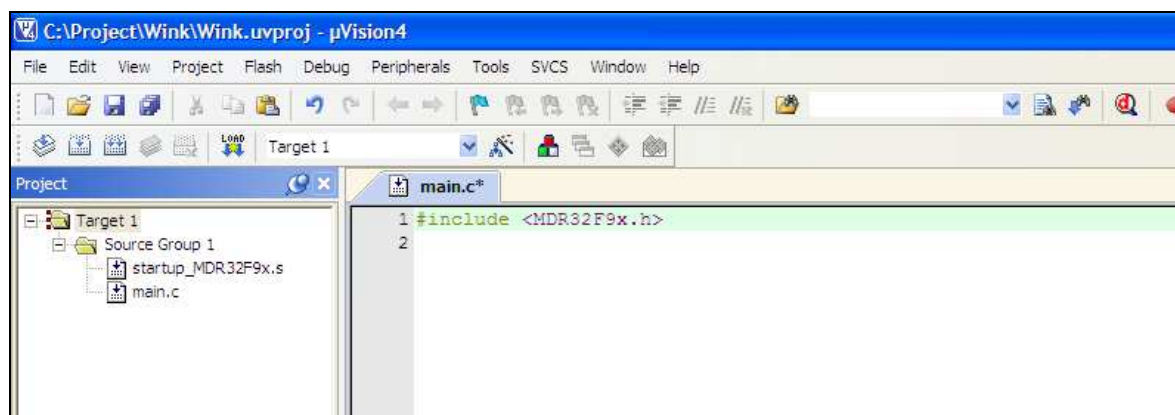
Шаг 5: Создаем программный код проекта в файле **main.c**.

Добавляем в файл «**main.c**» заголовочный h-файл семейства микроконтроллера «**MDR32F9x.h**». Нажимаем правой кнопкой мыши на

белое поле файла «**main.c**», где в выпадающем меню выбираем «**Insert '#include <MDR32F9x.h>'**»:



Появится строка в файле «**main.c**»:



Добавляем в проект файл «**system_MDR32F9x.c**» по аналогии с добавлением файла «**main.c**».

Файл «**system_MDR32F9x.c**» необходимо взять из библиотеки «**MDR32F9x Standart Peripheral Library**» (MDR_Library.rar).

Сама библиотека поставляется на CD-R диске, идущем в комплекте с платой **LDM-K1986BE92QI** «**CD-R\Примеры проектов\MDR_Library.rar**».

Её можно также взять на форуме компании ЗАО «ПКС Миландр»: <http://forum.milandr.ru> в разделе:

Интегральные микросхемы ЗАО "ПКС Миландр"→32-разрядные микроконтроллеры (1986BE9х, 1986BE1х, 1986BE2х) →32-разрядные микроконтроллеры серии 1986BE9х (ядро ARM Cortex-M3)→MDR32F9х Standart Peripheral Library.

Или на сайте компании ЗАО «ПКС Миландр»: <http://milandr.ru> в разделе «Программное обеспечение».

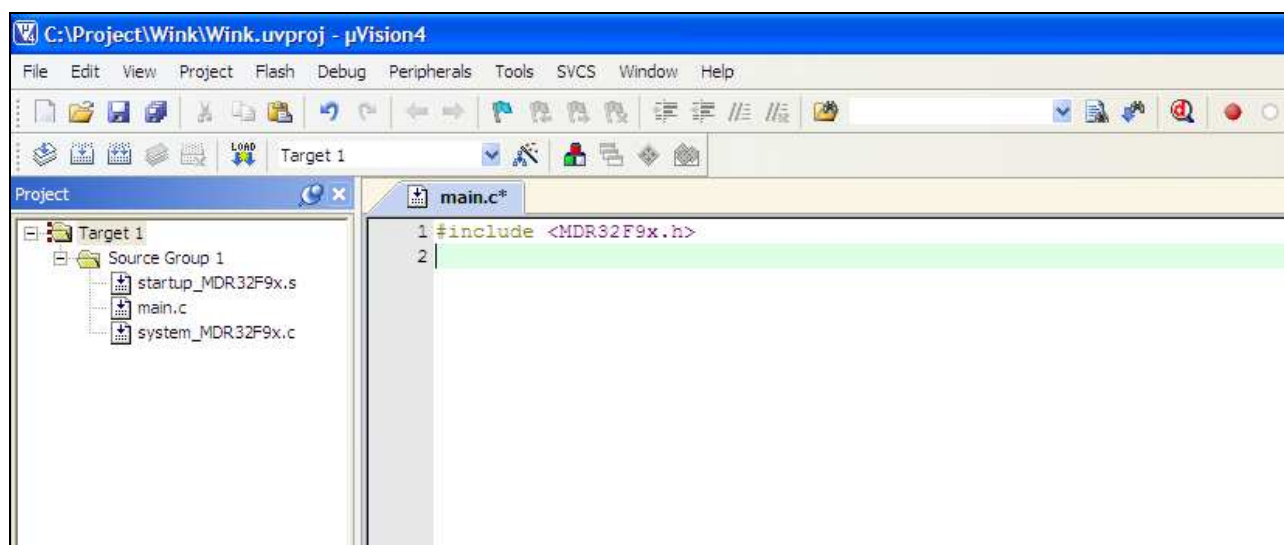
Или из установленной папки среды Keil: **\\Keil\\ARM\\Startup\\Milandr\\MDR32F9х**.

В результате в папке «**Source Group 1**» будут находиться уже три файла:

Файл инициализации контроллера - **startup_MDR32F9х.s**;

Системный - **system_MDR32F9х.c**;

Главный файл проекта – **main.c**.



Далее добавляем в наш файл main следующий код:

```
//=====
// Данный код предоставлен пользователем «редактор» на forum.milandr.ru
// и переделан для
// платы LDM-1986BE92QI компании LDM-SYSTEMS
//=====

#include <MDR32F9x.h>

#define BUTTON_SELECT (1<<2)          // Кнопка Select заведена на порт PC2
#define LAMP_VD5      (1<<0)          // Светодиод VD5 заведен на порт PB0

void main (void)
{

    // Разрешили работу всей периферии
    // Если этого не сделать, порты не будут инициализированы
    MDR_RST_CLK->PER_CLOCK = 0xFFFFFFFF;

    // Настройка портов ввода-вывода
    // Настраиваем порт PC
    MDR_PORTC->OE      = 0x00000000;    // Все линии порта C настраиваем на ввод
    MDR_PORTC->FUNC     = 0x00000000;    // Все линии порта используются как порт ввода-вывода
    MDR_PORTC->ANALOG  = 0x0000FFFF;    // Все линии - цифровые
    MDR_PORTC->PULL    = BUTTON_SELECT << 16; // Линию PC2 притянули к положительному уровню питания
    MDR_PORTC->PD      = BUTTON_SELECT << 16; // Включили триггер Шмидта по входу линии PC2

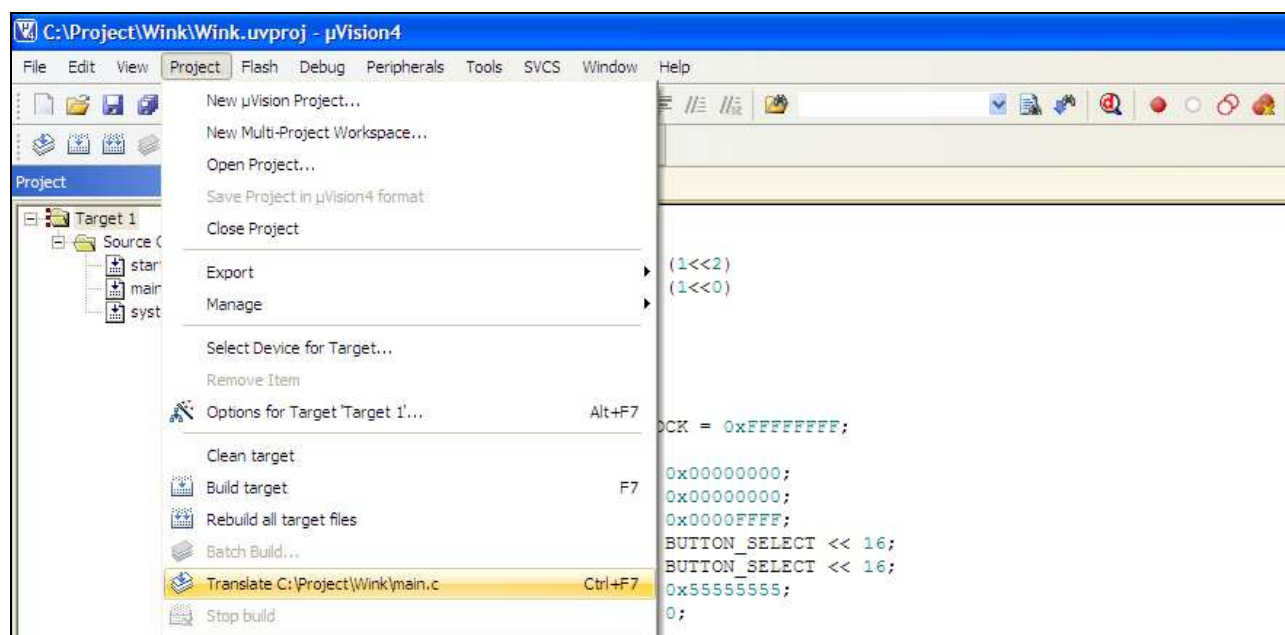
    MDR_PORTC->PWR     = 0x55555555;    // Медленный фронт по входу всем линиям
    MDR_PORTC->GFEN    = 0;             // Фильтр по входу выключен по всем линиям

    // Настраиваем порт PB
    MDR_PORTB->OE      = LAMP_VD5;      // Линию порта PB0 настраиваем на вывод
    MDR_PORTB->FUNC     = 0x00000000;    // Все линии порта используются как порт ввода-вывода
    MDR_PORTB->ANALOG  = 0x0000FFFF;    // Все линии - цифровые
    MDR_PORTB->PULL    = LAMP_VD5 << 16; // Линию PB0 притянули к положительному уровню питания
    MDR_PORTB->PD      = ~LAMP_VD5;     // Линия PB0 настраивается как управляемый драйвер
    MDR_PORTB->PWR     = 0x55555555;    // Медленный фронт по выходу линии PB0
    MDR_PORTB->GFEN    = 0;             // Фильтр по выходу выключен по всем линиям
    MDR_PORTB->RXTX    = 0xFFFF;       // Выставили все линии порта в высокое состояние

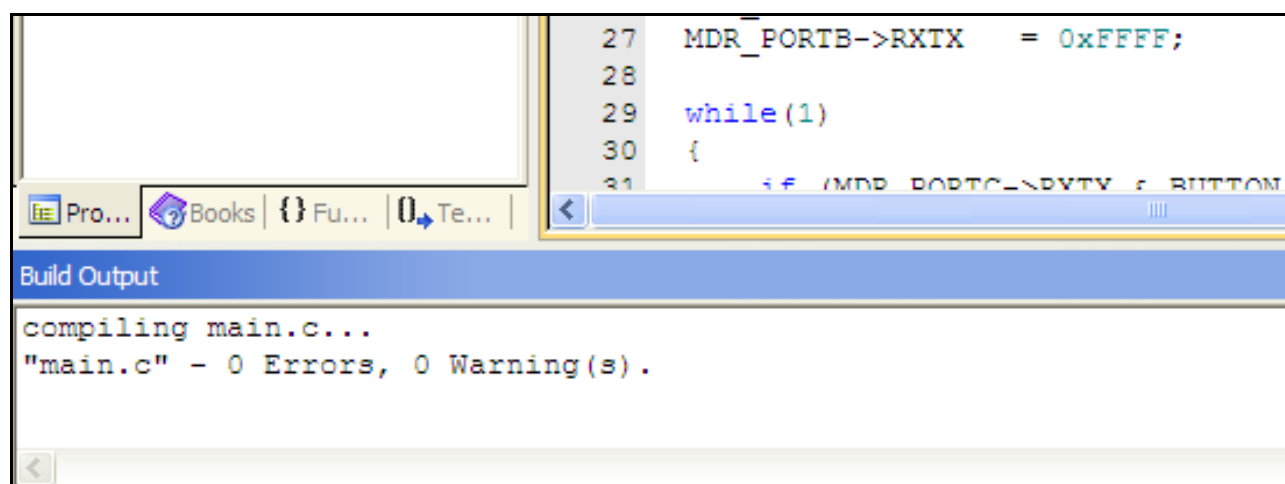
    while(1)                          // Основной цикл работы программы
    {
        if (MDR_PORTC->RXTX & BUTTON_SELECT) // Если бит установлен (кнопка отпущена)
        {
            MDR_PORTB->RXTX &= ~LAMP_VD5;    // Установили порт PB0 в 0 (низкий уровень выкл. светодиод)
        }
        else
        {
            MDR_PORTB->RXTX |= LAMP_VD5;      // Установили порт PB0 в 1 (высокий уровень вкл. светодиод)
        }
    }
}
```

Шаг 6: Проверка проекта.

В меню «Project» выбираем «Translate» или жмём на соответствующую иконку на интерфейсе программы.

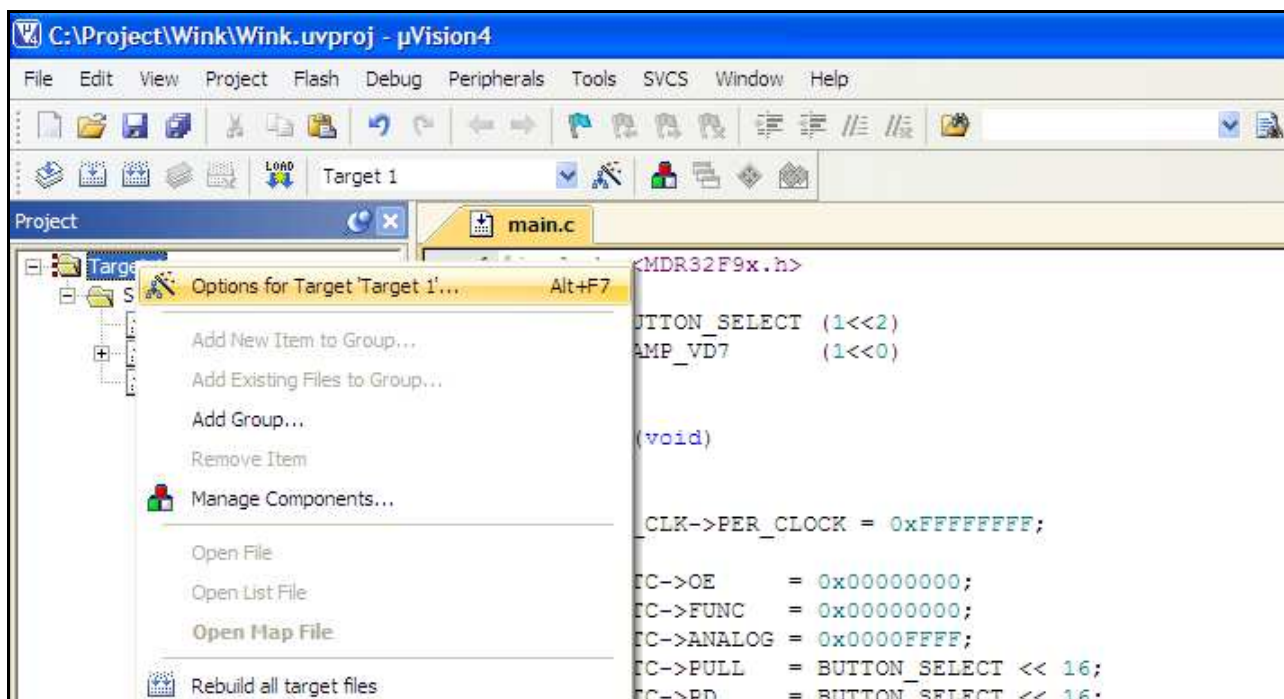


Наличие в проекте ошибки или предупреждения, будет выведено в нижнем окне программы «Build Output»:

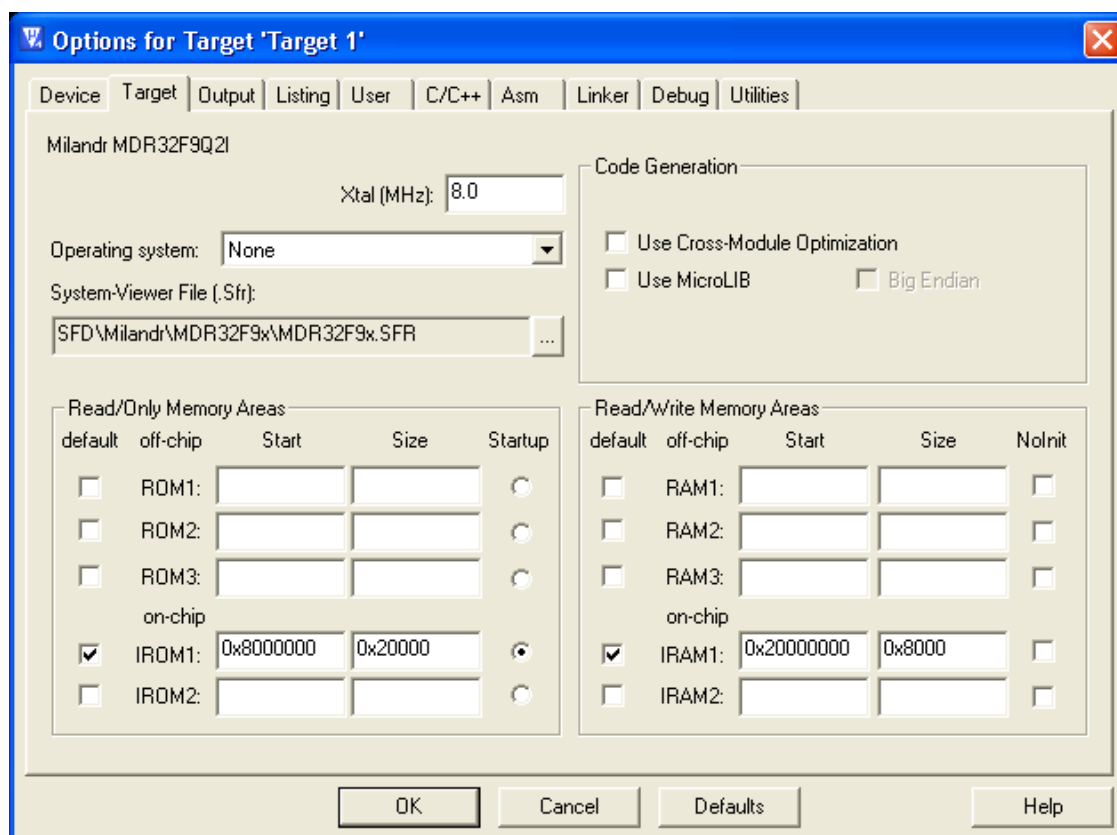


Шаг 7: Сборка проекта.

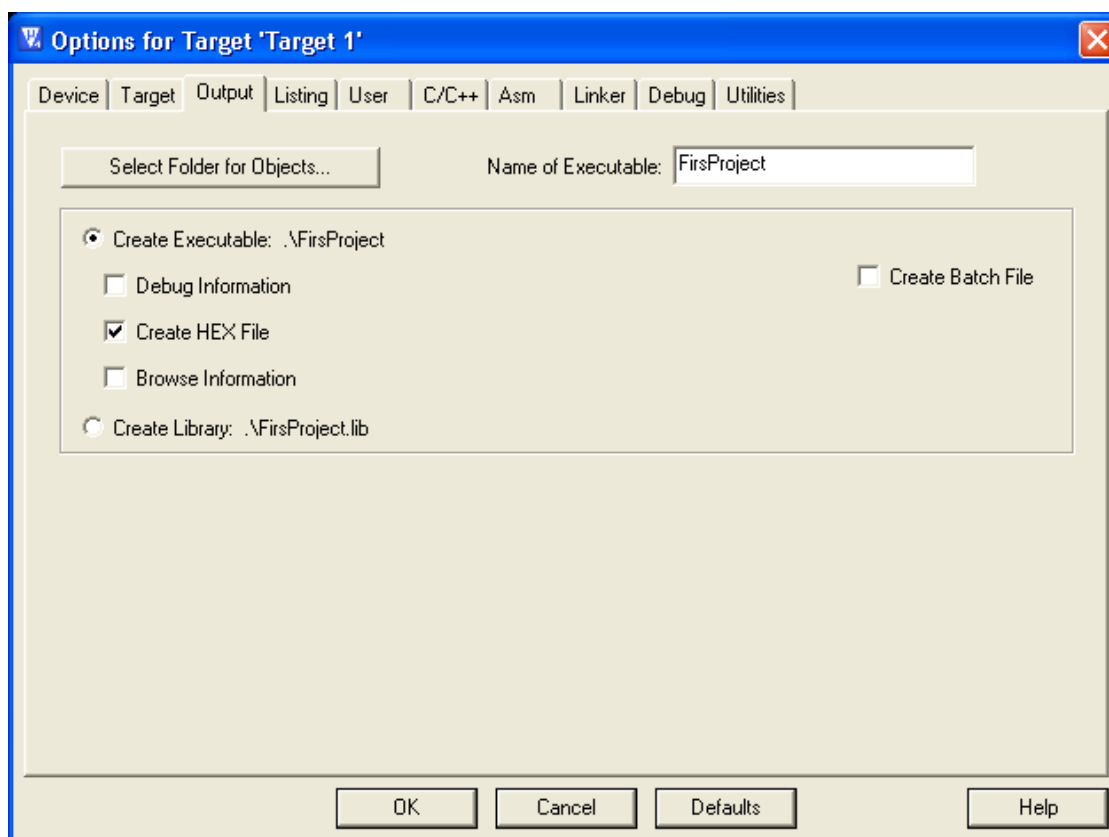
Настроим среду разработки на формирование hex-файла. Нажмем правой кнопкой мыши на папку «**Target 1**» в окне программы «**Project**» и в выпадающем меню выбираем «**Option for Target 'Target 1'**»:



Откроется окно «**Option for Target 'Target 1'**»:

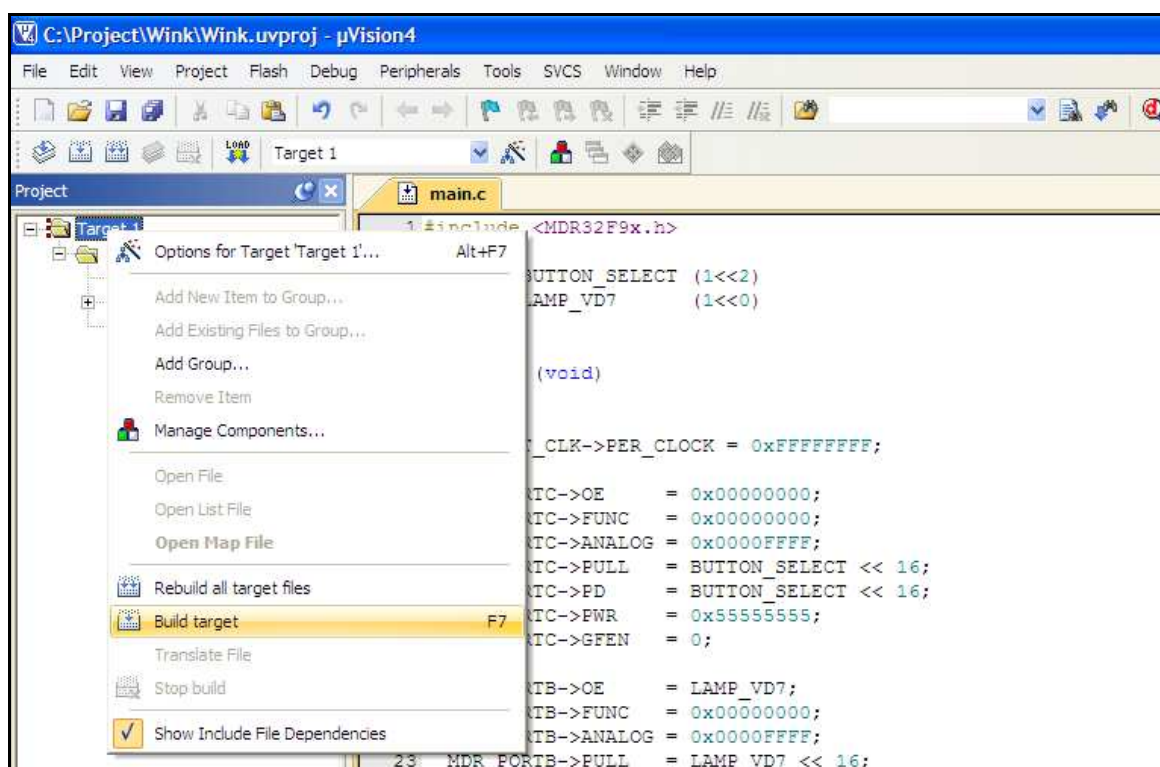


Перейдем на вкладку «**Output**» и поставим флаг «**Create HEX File**»:



Нажимаем «**OK**».

Для полной компиляции проекта необходимо в меню «**Project**» выбираем «**Build Target**»:



Шаг 8: Программирование FLASH памяти контроллера.

Программирование FLASH памяти контроллера можно осуществить при помощи встроенного в плату **LDM-K1986BE92QI** UART-загрузчик.

Программное обеспечение предоставлено пользователем «**vasili**» на **forum.milandr.ru**. в разделе: *Интегральные микросхемы ЗАО "ПКК Миландр" → 32-разрядные микроконтроллеры (1986BE9x, 1986BE1x, 1986BE2x) → 32-разрядные микроконтроллеры серии 1986BE9x (ядро ARM Cortex-M3) → AppNotes или примеры кода.*

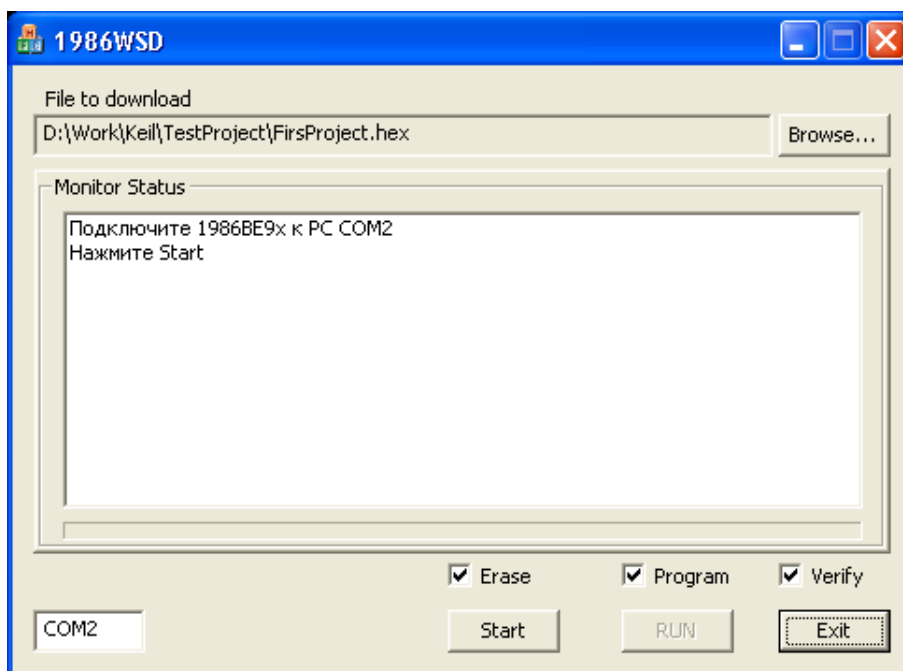
Программное обеспечение для UART-загрузчика можно скопировать на компьютер с диска для отладочного комплекта «**LDM-1986BE92QI**».

Перед тем, как запустить программу, необходимо установить на компьютер драйвер для микросхемы CP2102. Скачать драйвера можно непосредственно с сайта производителя, компании «**Silicon Labs**» <http://www.silabs.com>. Или непосредственно с диска к отладочному комплекту «**LDM-K1986BE92QI**».

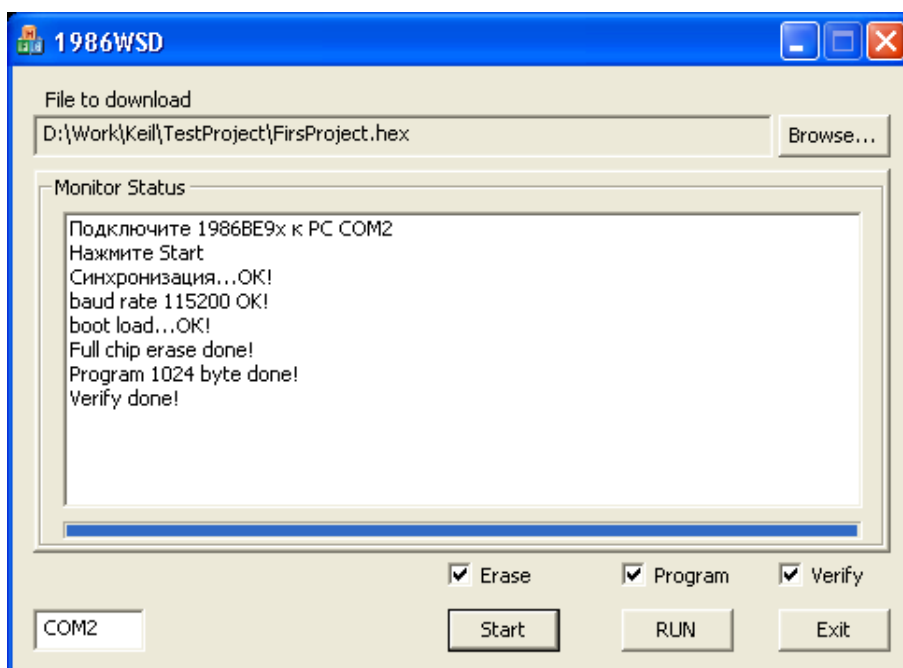
После установки драйвера в системе, при подключении платы кабелем к разъёму USB-A, появится COM-порт. Необходимо учитывать, что программа UART-загрузчик корректно работает с портами 1-9.

Далее, перед тем как включить питание платы, необходимо выставить режим загрузки микроконтроллера. Для этого переводим переключатели SW10 и SW6 в положение «1», а SW9 в «0». Подробнее о режимах загрузки микроконтроллера написано в спецификации на микроконтроллер **K1986BE92QI**.

Включаем питание и запускаем программу UART-загрузчик «1986WSD.exe». Нажимаем кнопку «Browse...» и указываем путь к файлу «Wink.hex», который находится в папке проекта «Wink»:

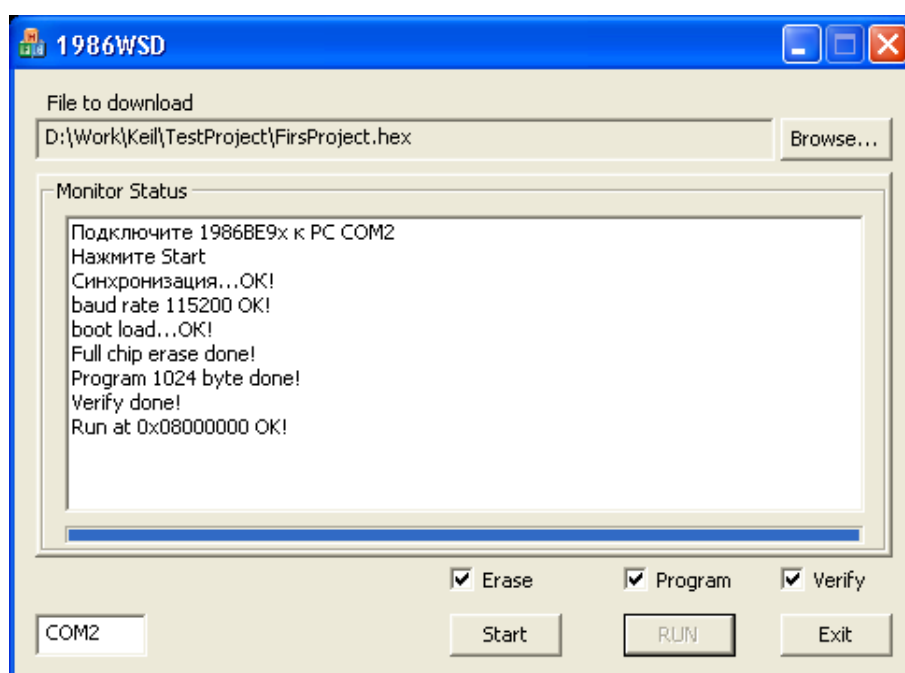


Нажимаем кнопку «Start», после чего в окне «Monitor Status» увидим соответствующую надпись об успешном выполнении загрузки программы во внутреннюю FLASH-память микроконтроллера.



Запуск программы на исполнение в контроллере можно осуществить нажатием кнопки «**RUN**» после окончания операции загрузки.

При использовании программы **1986WSD** не обязательно каждый раз после прошивки МК выключать/включать питание. Достаточно в окне программы нажать кнопку «**RUN**», после чего программа на МК начнёт выполняться. Для повторной перепрошивки можно нажать «**Reset**» на плате отладочного комплекта и МК вновь будет готов к перепрограммированию.



Об успешном выполнении программа также сообщит.

Поздравляем!

Вы освоили навыки работы с отладочной платой **LDM-K1986BE92QI** в среде проектирования **Keil uVision** и можете самостоятельно попробовать создать свой проект!

Enjoy!

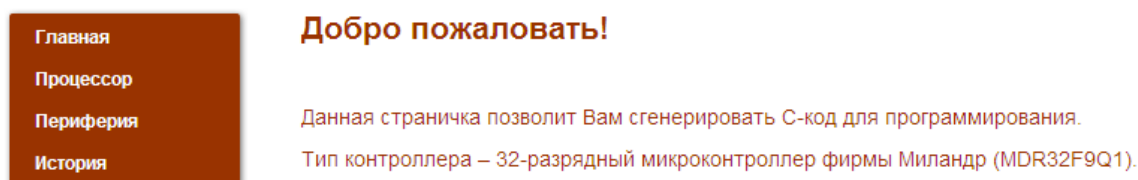
2.5 Полезные программы

На CD-R диске в папке «Полезное ПО» находится файл «Milandr_PLLv2.2.rar» (автор AntonAS форума Миландр: <http://forum.milandr.ru/viewtopic.php?f=33&t=977&p=9578#p8908>) в котором находятся утилиты, позволяющие быстро создать код настроек периферийных узлов: can, adc, cpu, port, tim, uart.

Для работы с утилитами произведите запуск файла «index.html» в интернет браузере.

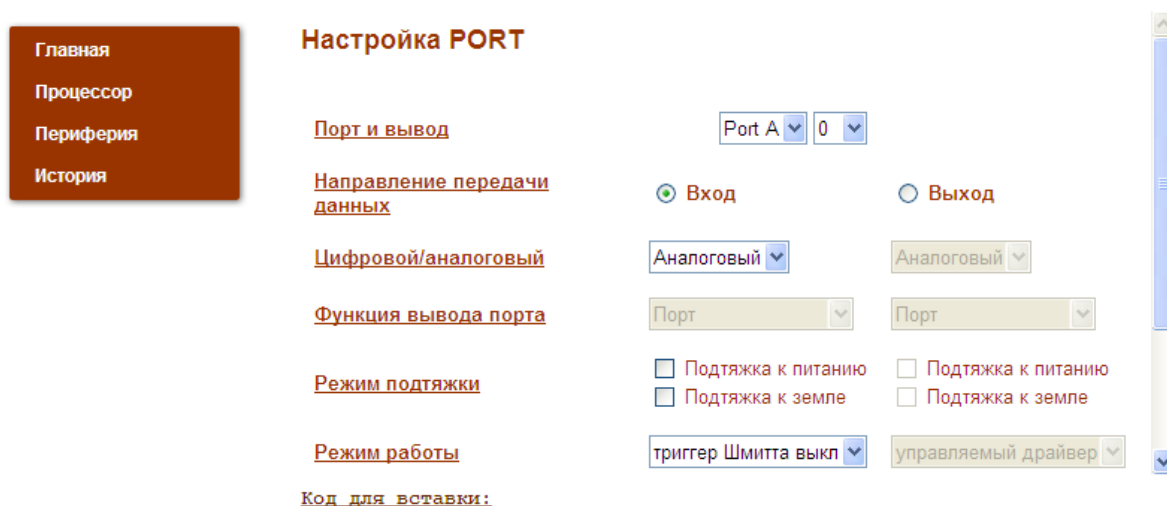
Появится основное окно ресурса:

Версия v2.1



В качестве примера настроим порт В0 на вывод цифровых данных. Выбираем закладку «Периферия/Port»:

Версия v2.1



Выбираем в поле «Настройка PORT»

Порт и вывод:	Port B, 0;
Направление передачи данных:	Выход;
Цифровой/аналоговый	Цифровой;
Функция вывода порта	Порт;
Режим подтяжки	Нет;
Режим работы	Управляемый драйвер;
Режим работы	Быстрый.

Версия v2.1

Главная
Процессор
Периферия
История

Настройка PORT

Порт и вывод Port B 0

Направление передачи данных ☐ Вход ☒ Выход

Цифровой/аналоговый Аналоговый Цифровой

Функция вывода порта Порт Порт

Режим подтяжки ☐ Подтяжка к питанию ☐ Подтяжка к земле ☐ Подтяжка к питанию ☐ Подтяжка к земле

Режим работы триггер Шмитта выкл управляемый драйвер

Режим вывода медленный быстрый

☐ Фильтр

Рассчитать Рассчитать

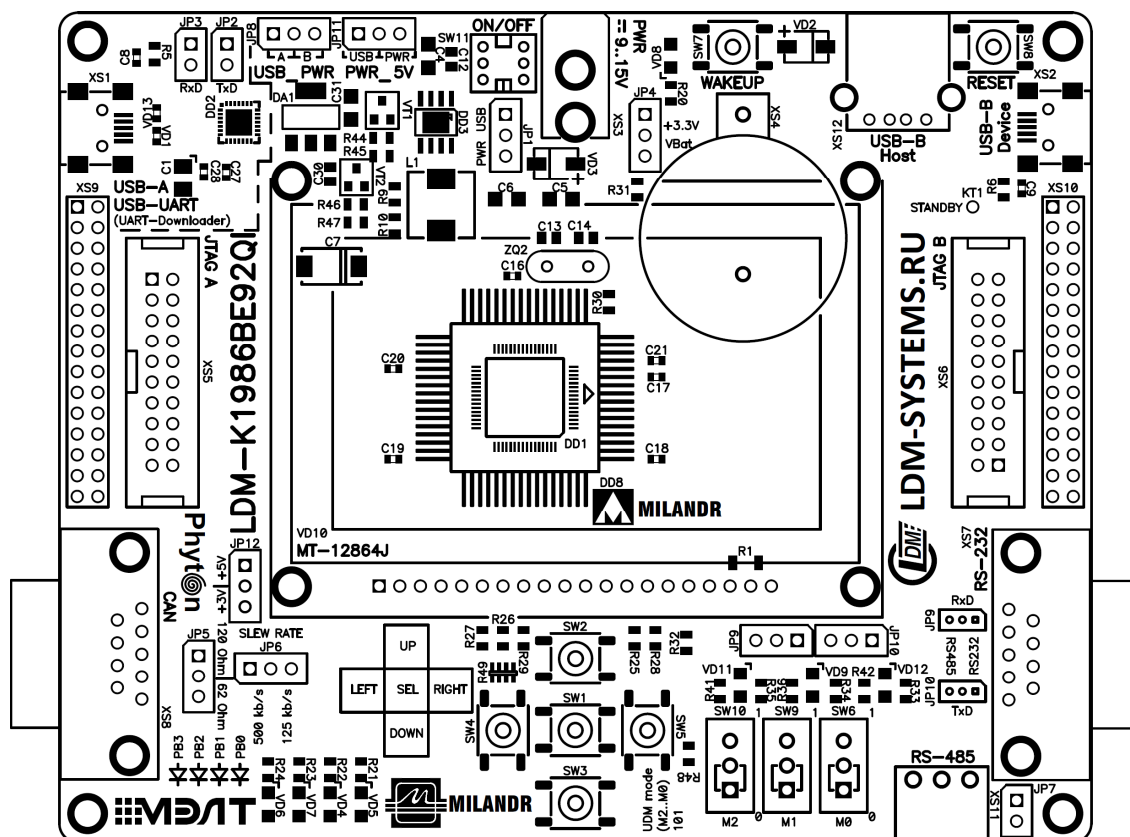
В результате в поле «Код для вставки» появится код программы для настройки периферийного модуля:

```
void Port_init(void)
{
    MDR_RST_CLK->PER_CLOCK |= (1UL << 22); // разрешение тактирования порта B
    MDR_PORTB->OE = ((1 << 0)); // направление передачи данных = Выход
    MDR_PORTB->ANALOG = ((1 << 0)); // режим работы контроллера = Цифровой
    MDR_PORTB->FUNC = ((0 << 0*2)); // режим работы вывода порта = Порт
    MDR_PORTB->PULL = ((0 << 0)); // запрещение подтяжки к GND
    MDR_PORTB->PULL = ((0 << (0 << 16))); // запрещение подтяжки к VCC
    MDR_PORTB->PD = ((0 << 0)); // режим работы выхода = управляемый драйвер
    MDR_PORTB->PWR = ((2 << 0*2)); // скорость фронта вывода = быстрый
}
```

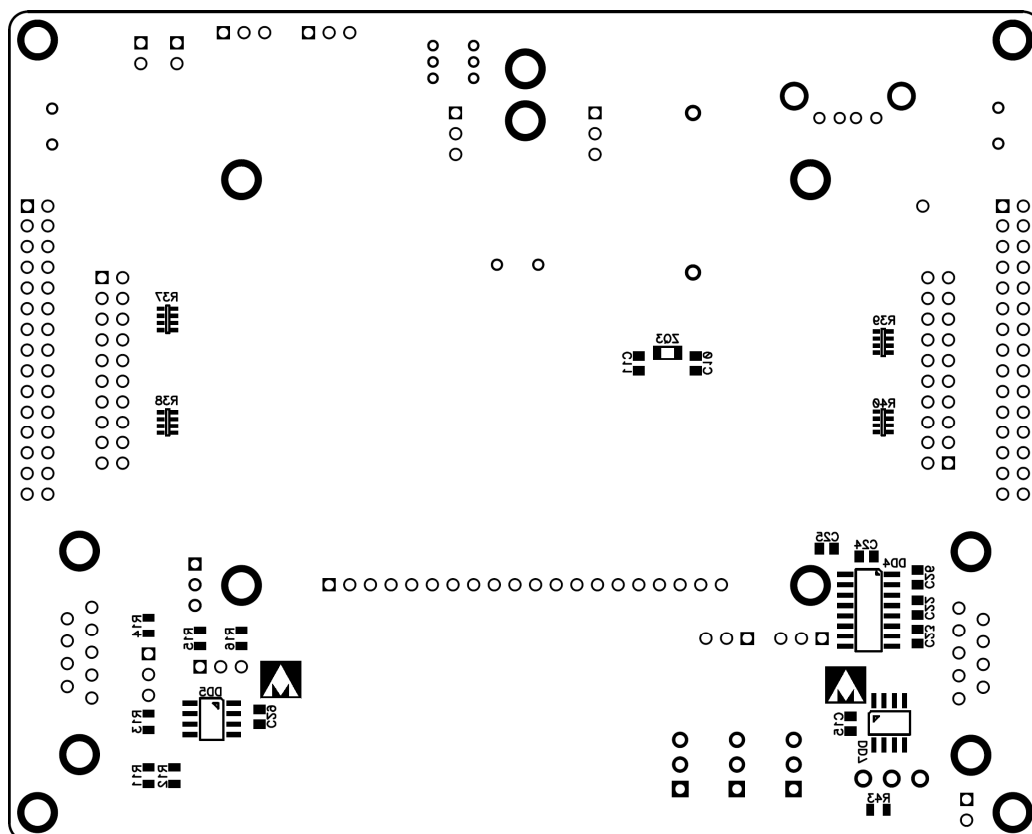
Представленное приложение позволит начинающему программисту быстро освоить принципы настройки контроллера K1986BE92QI (MDR32F9Q2I).

2.6 Монтажные чертежи

Слой TOP

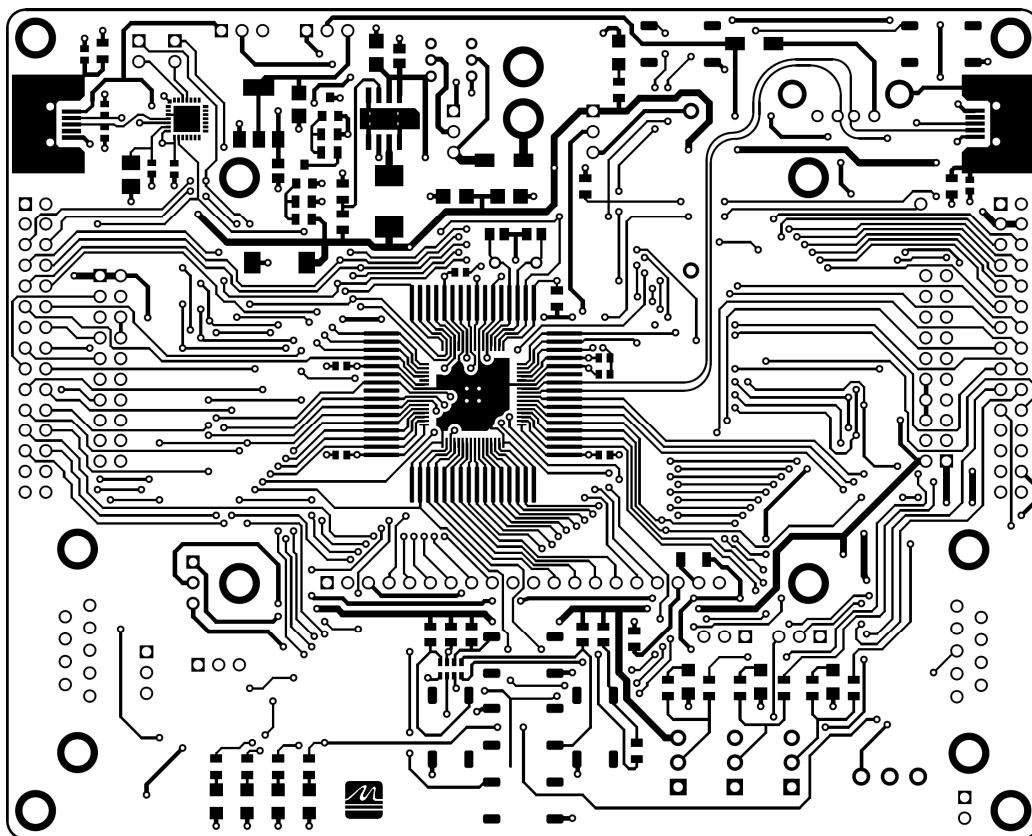


Слой BOTTOM

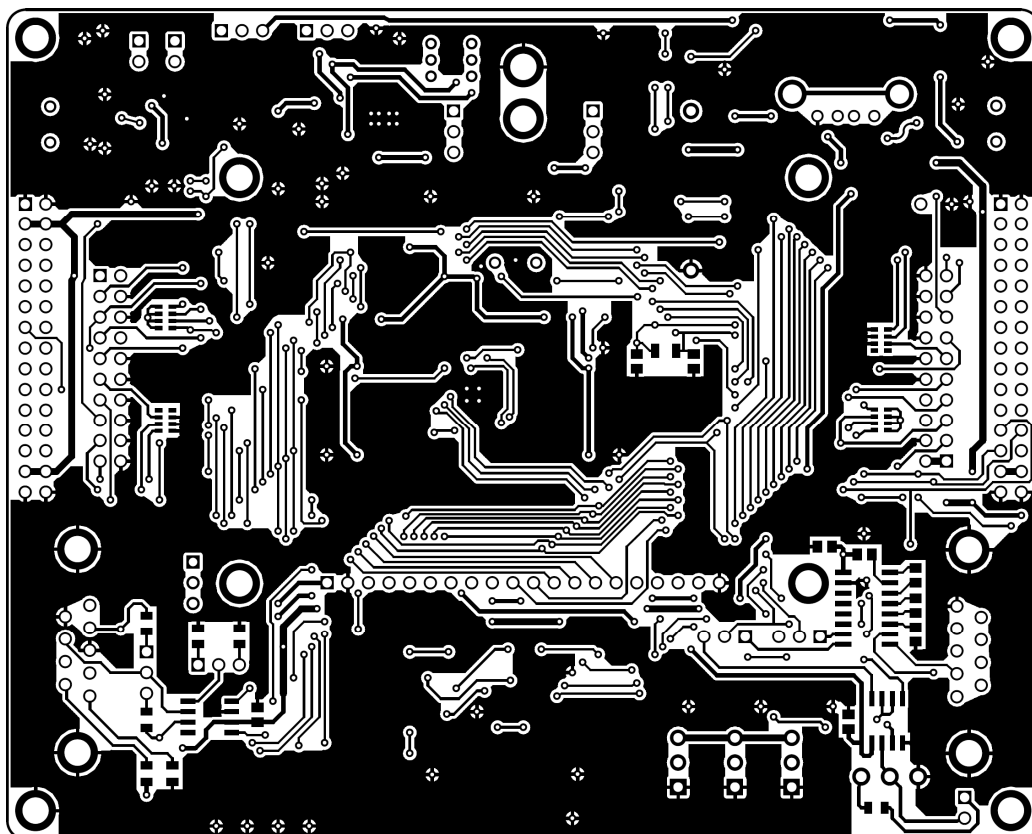


2.7 Трассировка по слоям

Слой TOP



Слой BOTTOM



3 Эксплуатация, хранение и транспортирование

Требования к условиям эксплуатации:

Изделие при испытаниях, перевозке, хранении и эксплуатации не наносит вреда окружающей среде и здоровью человека. Сохраняет свои параметры во всем диапазоне рабочих температур от 0 °С до +70 °С в закрытом помещении с относительной влажностью воздуха не более 80 %, без конденсата, при изменении напряжения первичного источника электропитания в допустимых пределах. По электромагнитной совместимости изделие соответствует всем требованиям для аппаратуры данного класса.

Требования к условиям хранения:

Изделие должно храниться в складских помещениях, защищенных от воздействий атмосферных осадков, на стеллажах в упаковке производителя при отсутствии в воздухе паров кислот, щелочей и других веществ, вызывающих коррозию. Условия хранения изделия по ГОСТ 15150-69: температура воздуха от 5 до 40 °С, относительная влажность до 80 % при температуре 25 °С. Предельный срок хранения в указанных условиях - три года.

Требования к условиям транспортирования:

Транспортирование изделия разрешается в упаковке производителя - всеми видами транспорта, за исключением негерметизированных отсеков самолета, без ограничения расстояния.

Транспортирование упакованных изделий может производиться в крытых вагонах и автомашинах, трюмах судов и герметичных кабинах самолетов при температуре воздуха от минус 20 до плюс 70 °С. При любом способе транспортирования необходимо предусмотреть крепление ящика к кузову (платформе) транспортного средства с помощью крепежной арматуры.

Литература

- 1) Спецификация микроконтроллеров серии 1986BE9х, K1986BE9х и MDR32F9Qх ЗАО «ПКК Миландр» ТСКЯ.431296.001СП Версия 3.2.0 от 20.09.2012.
- 2) Микросхема приемо-передатчика по стандарту RS-485/RS-422 5559ИН10АУ, K5559ИН10АУ, K5559ИН10АН4 5559ИН10БУ K5559ИН10БУ, K5559ИН10БН4 ЗАО «ПКК Миландр» ТСКЯ.431323.002СП Версия 2.5.0 от 29.05.2012.
- 3) Жидкокристаллический модуль МТ–12864J ООО «МЭЛТ», версия документа 1.2, 18/06/2007.