

# Начинаем работу с Nuvoton микроконтроллерами

Автор: Юрий Гаман, ООО «Гамма»

## 1. Создание проекта в Keil uVision IDE

Для начала работы нужно загрузить среду разработки Keil uVision. Для этого заходим на страницу <https://www.keil.com/download/product/> и загружаем MDK-ARM v5, куда входят uVision и программные библиотеки. Keil uVision также можно установить с CD-диска, который поставляется с отладочными платами Nuvoton.

Для начала работы с микроконтроллером нужно добавить библиотеки периферийных устройств. Библиотеки доступные для свободного скачивания на сайте [www.nuvoton.com](http://www.nuvoton.com) в разделе «ARM Cortex-M0 NuMicro Family->Development Resources->Development Tools->Device Driver and Software Library». Также их можно скопировать с компакт диска, который поставляется с отладочными платами Nuvoton. На странице загрузки выбираем CMSIS библиотеку, например для микроконтроллера M0516LBN серии M051, на примере которого будет приводиться пример программы, загружаем файл **M051 SeriesBSP\_CMSIS\_V2.01.002.zip**. Создаем директорию Nuvoton в C:/Keil, и распаковываем туда загруженную библиотеку. Самый простой способ создать проект это отредактировать шаблон проекта, который находится в директории с данным уроком. Копируем всю папку с проектом M051Demo в удобную для Вас директорию, два раза нажимаем на файле проекта M051Demo.uvproj, после чего должен открыться uVision.

Теперь можно приступать к написанию программы для Nuvoton микроконтроллера. Ниже приведенная программа мигает светодиодом на отладочной плате NuTiny-SDK-M0516LDN. Для написания программы нужно перейти к файлу main.c. Для этого на панели Project в группе Source files нажимаем левом кнопкой мыши на main.c (рис.1.1).

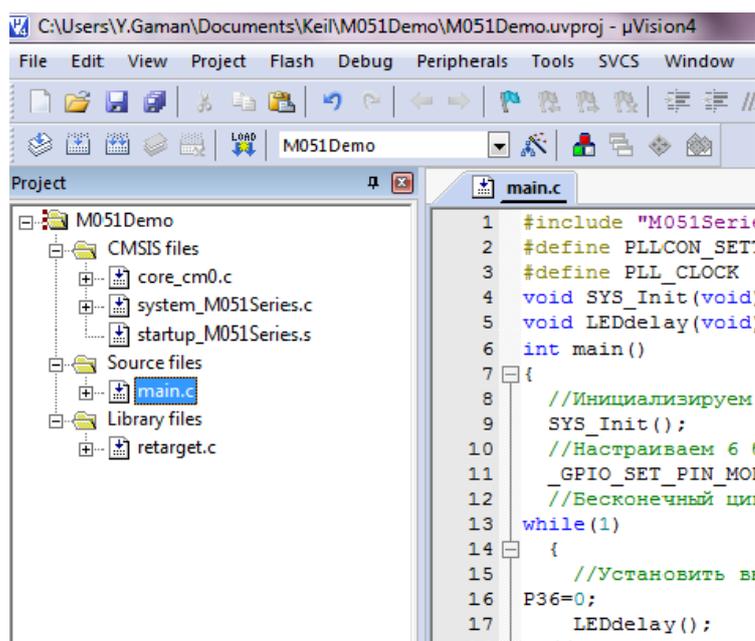


Рисунок 1.1 – Выбор файла для редактирования

```
#include "M051Series.h"
```

```

#define PLLCON_SETTING      SYSCLK_PLLCON_50MHz_XTAL
#define PLL_CLOCK           50000000

void SYS_Init(void); //Функция для настройки системного генератора
void LEDdelay(void); //Функция задержки

int main()
{
    SYS_Init(); //Инициализируем системный генератор
    GPIO_SET_PIN_MODE(P3, 6, GPIO_PMD_OUTPUT); //Настраиваем 6 бит порта №3 на вывод
    //Бесконечный цикл
    while(1)
    {
        P36=0; //Установить вывод P3.6 в «0»
        LEDdelay();
        P36=1; //Установить вывод P3.6 в «1»
        LEDdelay();
    }
}

void LEDdelay(void)
{
    int i;
    for (i=0; i<1000000; i++);
}

void SYS_Init(void)
{
    SYS_UnlockReg(); //Разблокировать защищенные регистры
    //Настроить внешний кварц как источник системного генератора
    SYSCLK->PWRCON |= SYSCLK_PWRCON_XTL12M_EN_Msk;
    SYSCLK->PLLCON = PLLCON_SETTING; //Включить PLL
    //Подождать пока стабилизируется системный генератор
    SYS_WaitingForClockReady(SYSCLK_CLKSTATUS_PLL_STB_Msk
    SYSCLK_CLKSTATUS_XTL12M_STB_Msk);
    //Обновить ядро системного генератора (для использования значений тактовой
    частоты в программе пользователя)
    SystemCoreClockUpdate();
}

```

```
//Заблокировать защищенные регистры
SYS_LockReg();
}
```

Перед компилированием программы в настройках проекта (Меню Project->Options For Target) во вкладке Device выбираем нужный микроконтроллер, для данного примера M0516LBN (рис 1.2). Далее не закрывая диалоговое окно Options for Target, во вкладке Debug выбираем программатор Nuvoton Nu-Link M0 Debugger в комбинированном списке Use (рис. 1.3). В этой вкладке нажимаем кнопку Settings и настраиваем программатор-отладчик на нужную серию микроконтроллеров в комбинированном списке ChipSelect, например для серии M051 необходимо выбрать M05x. Во вкладке Utilities (диалоговое окно Options for Target) ставим галочку “Use Debug Driver” (рис. 1.4).

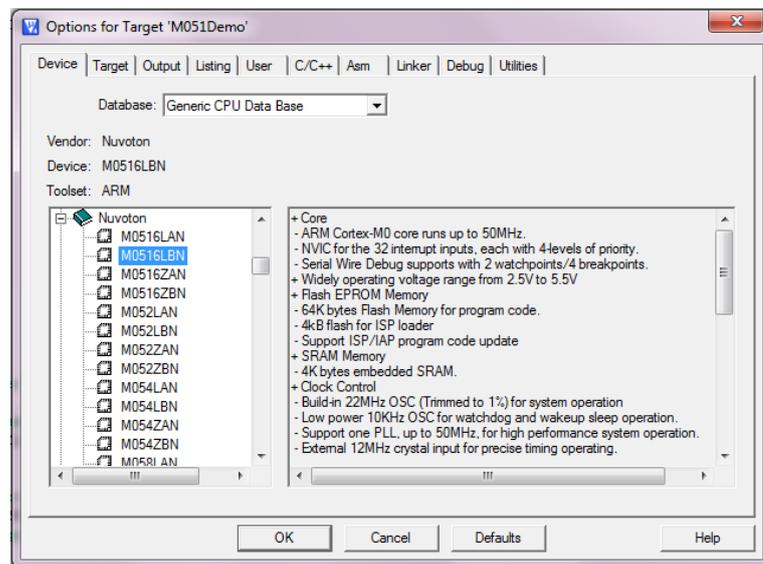


Рисунок 1.2 – Выбор микроконтроллера

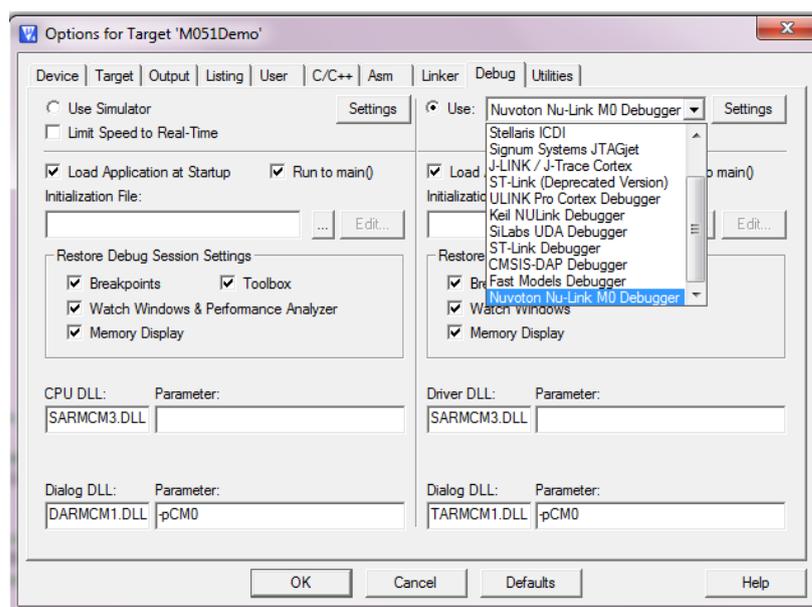


Рисунок 1.3 – Выбор программатора-отладчика

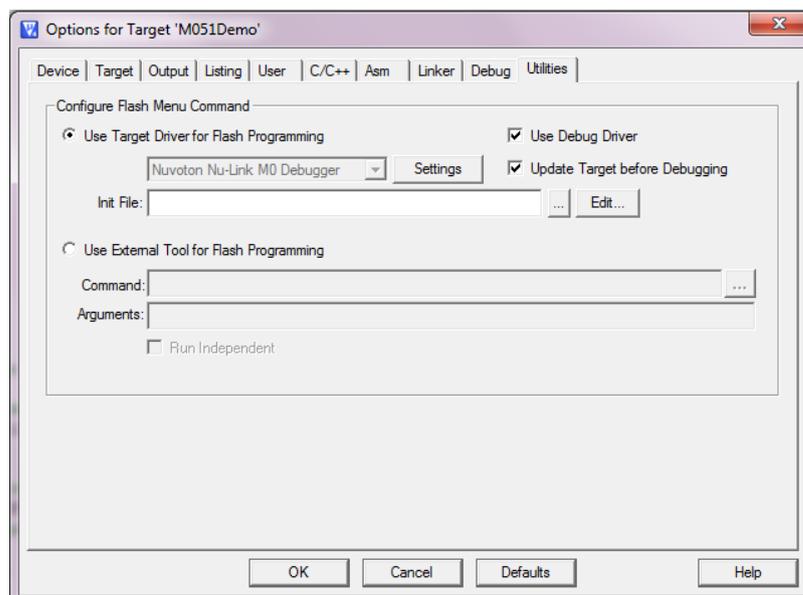


Рисунок 1.4 – Настройка программатора

Для компиляции программы нужно нажать кнопку Build или нажать клавишу F7 (рис. 1.5). Для загрузки программы нажимаем на кнопку Load, которая находится справа от кнопки Build. Для того, чтобы микроконтроллер начал работу нужно сбросить микроконтроллер.

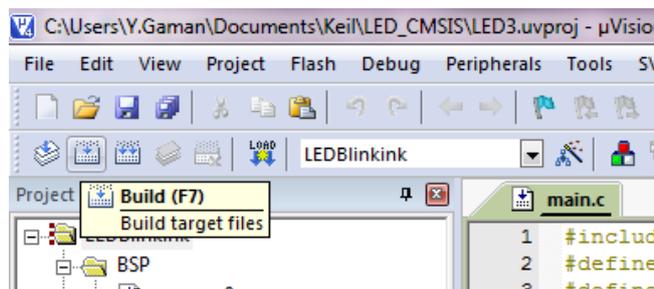


Рисунок 1.5 – Компилирование программы

## 2. Создание проекта в CoCoX CoIDE

Среда разработки CoIDE является бесплатной и использует компилятор GCC. Программу можно загрузить с официального сайта <http://coocox.org>. Компилятор для ARM микроконтроллеров доступный для свободной загрузки на веб странице <https://launchpad.net/gcc-arm-embedded/+download>.

После установки CoIDE необходимо установить компилятор и указать его путь в среде разработки. Для этого в меню Project нажимает на "Select Toolchain Path". После нажатия в диалоговом окне нужно указать папку bin из директории компилятора, например, C:\Program Files (x86)\GNU Tools ARM Embedded\4.7 2013q3\bin.

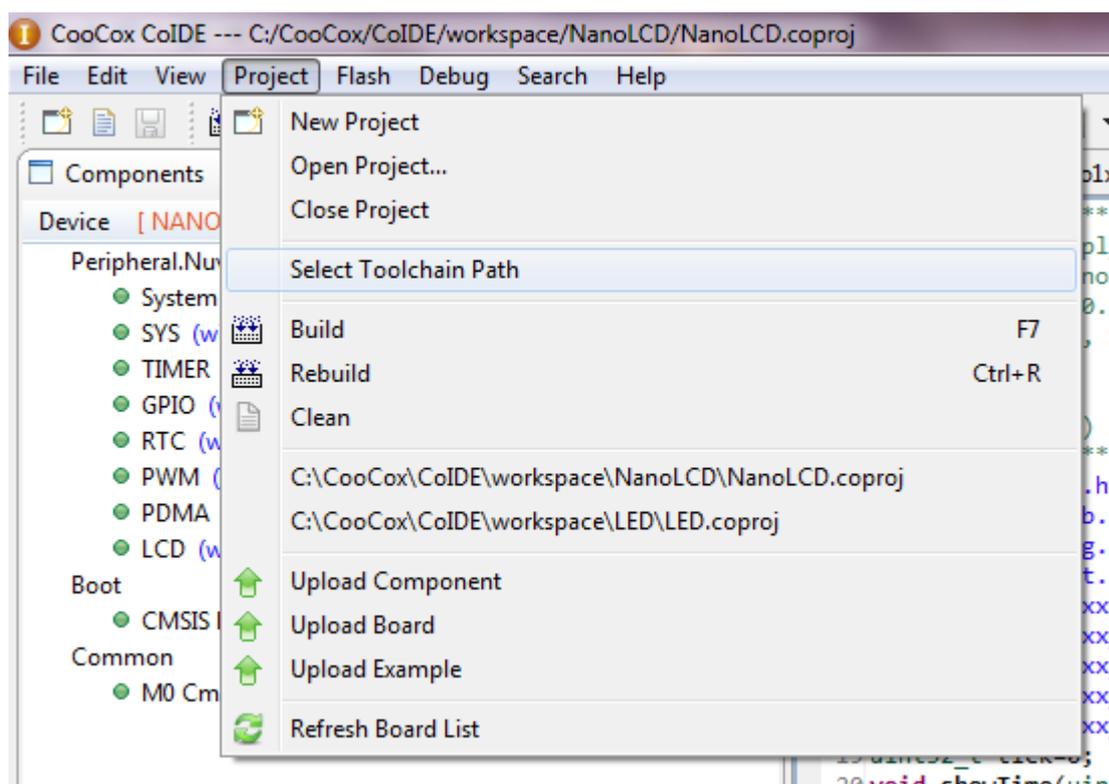


Рисунок 2.1 – Указание пути к компилятору

Для создания проекта нужно выбрать в меню Project пункт New Project. Указываем имя проекта и нажимаем Next. Далее наводим на раздел Chip (рисунок 2.2), выбираем микросхему в разделе Nuvoton и нажимаем кнопку Finish. После этого на экране должна появиться вкладка Repository (Рисунок 2.3), где нужно выбрать необходимые программные библиотеки. Для примера со светодиодом достаточно подключить библиотеку GPIO, остальные необходимые компоненты (SYS, SystemDefinitions) подключаться автоматически. После настройки библиотек можно закрыть вкладку Repository и переходить к написанию программы.

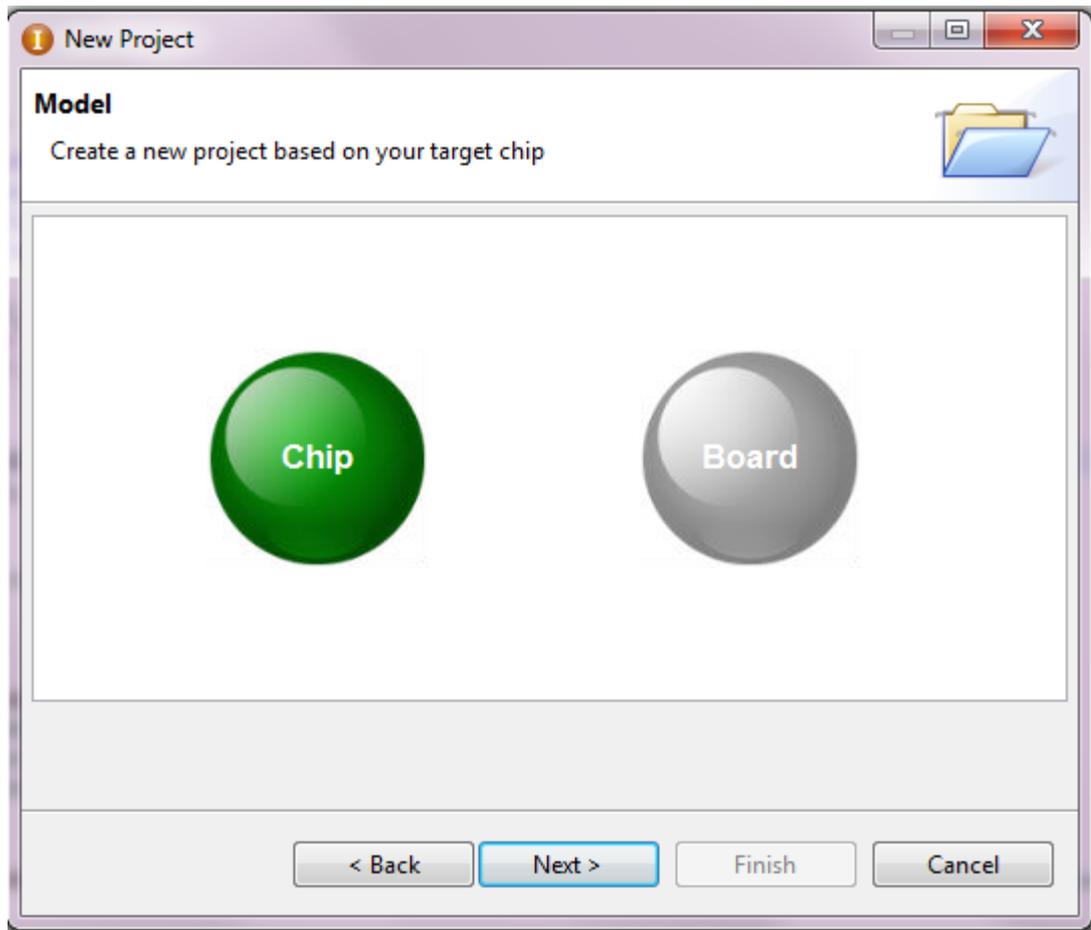


Рисунок 2.2 – Создание проекта для микроконтроллера

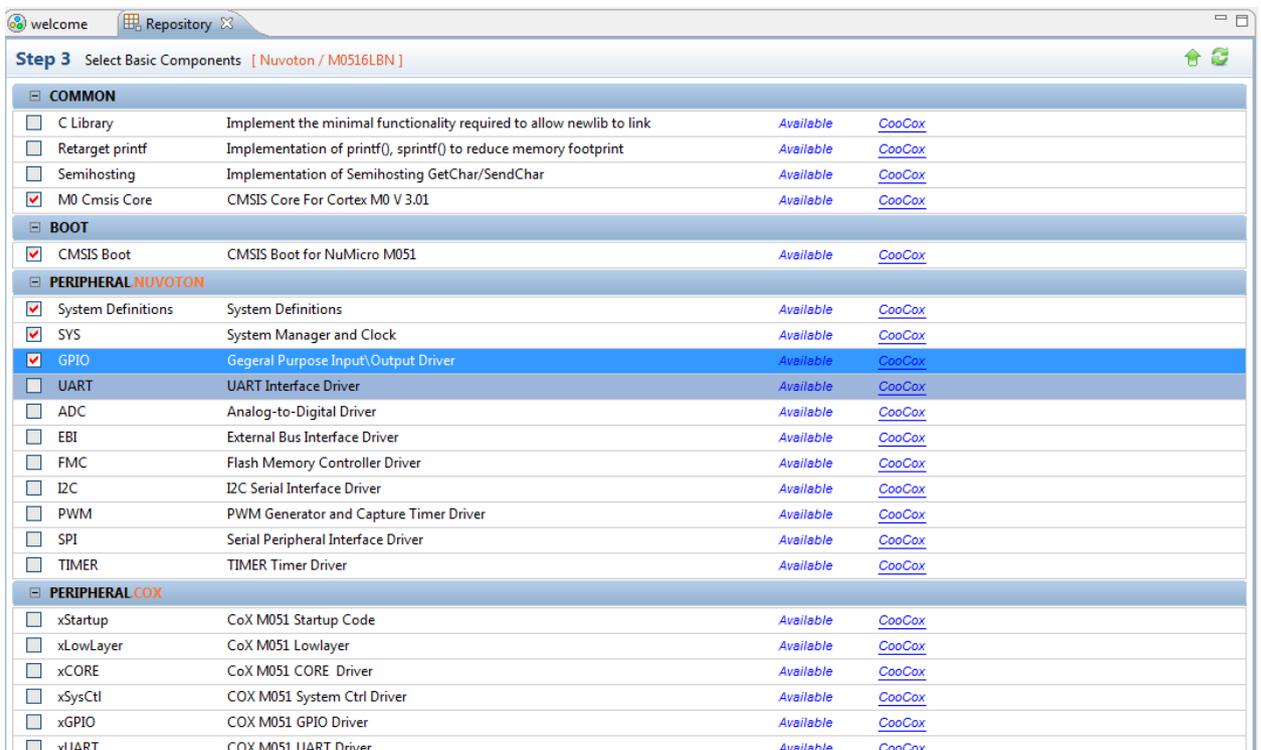


Рисунок 2.3 – Подключение библиотек

Ниже приведенная программа написана для отладочной платы NuTiny-SDK-M0516LDN

```
#include "DrvSYS.h"
#include "DrvGPIO.h"//Библиотека для входов/выходов
void LEDdelay(void);
int main()
{
    DrvSYS_UnlockProtectedReg(); //Разблокировать защищенные регистры
    DrvSYS_Open(50000000); //Настроить системный генератор на частоту 50 МГц
    DrvGPIO_Open(E_PORT3,6,E_IO_OUTPUT); //Настроить 6 бит порта P3 на вывод
    while(1)
    {
        P36_DOUT=0; //Установить вывод P3.6 в «0»
        LEDdelay(); //Задержка
        P36_DOUT=1; //Установить вывод P3.6 в «1»
        LEDdelay(); //Задержка
    }
}
void LEDdelay(void) //Функция задержки
{
    int i;
    for (i=0; i<1000000; i++);
}
```

После написания программы компилируем ее с помощью кнопки Build (рисунок 2.4). и в случае удачного компилирования зашиваем программу в микроконтроллер кнопкой «Download Code To Flash» (рис. 2.4 – справа от кнопки Build).

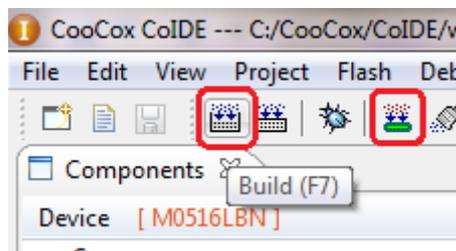


Рисунок 2.4 – Компилирование программы