

**Конструктор робот-манипулятор
Hiwonder LOBOT LeArm Single Robot**

Инструкция по эксплуатации

Оглавление

| | |
|--|----|
| 1. Сборка..... | 3 |
| 2. Калибровка | 4 |
| 3. Управление..... | 6 |
| 4. Главный компьютер (хост) | 7 |
| 5. Джойстик..... | 9 |
| 6. Управление через смартфон | 10 |
| 7. Пассивное управление | 10 |
| 8. Создание среды разработки и программирование..... | 11 |

1. Сборка

Для клиентов, которые приобрели робот в виде конструктора, первым делом предстоит его сборка. Подробное видео с процессом сборки вы можете найти у нас на сайте.

Перед сборкой настоятельно рекомендуем убедиться, что в комплекте присутствуют все детали, показанные на картинках ниже. Обратите внимание, что в приведенном списке показаны детали второго поколения.





Сборка делится на 3 этапа: сборка каркаса, крепление сервоприводов, подключение электрических контактов. При сборке каркаса разделите детали на передние, задние, правые, левые, верхние и нижние, внимательно следите за процессом сборки на видео.

При монтаже сервоприводов следите за положением главного и вспомогательного валов, не поворачивайте сервоприводы в процессе сборки. Исходное положение всех сервоприводов приблизительно одинаковое (90 градусов), на видео задействованы муляжи отцентрированных сервоприводов, поэтому, если вы повернете ось сервопривода, будет затруднительно повторить процесс сборки с видео. Сперва отцентрируйте сервопривод с помощью платы управления и лишь затем продолжайте сборку. Подробный алгоритм центрирования сервопривода представлен в справочных материалах. В конце, при подключении электрических контактов и платы управления, строго сверяйтесь с указаниями по подключению. Следите за соответствием маркировки проводов и портов на плате, неправильное подключение грозит порчей оборудования.

2. Калибровка

По окончании сборки манипулятора подключите его к ПК (хосту) через дата-кабель, включите ПК, выберите COM-порт, нажмите кнопку «Reset» сервопривода. В нормальном режиме робот-манипулятор должен вести себя так, как показано на рисунке 2, т.е. находиться в вертикальном положении. Если положение отличается, необходимо снять

сервопривод, отцентрировать его и повторить все заново. При наличии небольшого отклонения рекомендуется произвести калибровку сверяясь со справочными материалами.



Рисунок 1

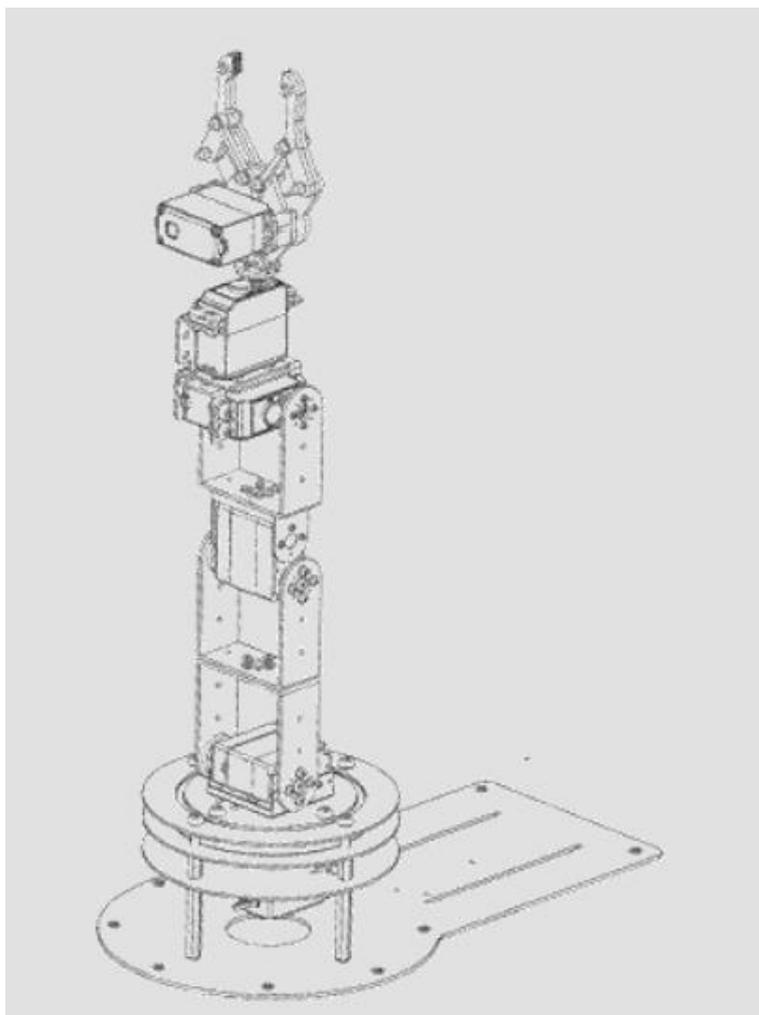


Рисунок 2

3. Управление

Робот-манипулятор LeArm предусматривает 4 способа управления: от ПК, от смартфона, от джойстика и пассивное управление. Пассивное управление и управление через приложение на смартфоне производится с помощью наборов действий, которые необходимо предварительно загрузить на плату управления. Подключившись к ПК, вы можете писать, редактировать или скачивать наборы действий. Управление через джойстик производится так же через наборы действий или напрямую, каждым сервоприводом по отдельности.

4. Главный компьютер (хост)

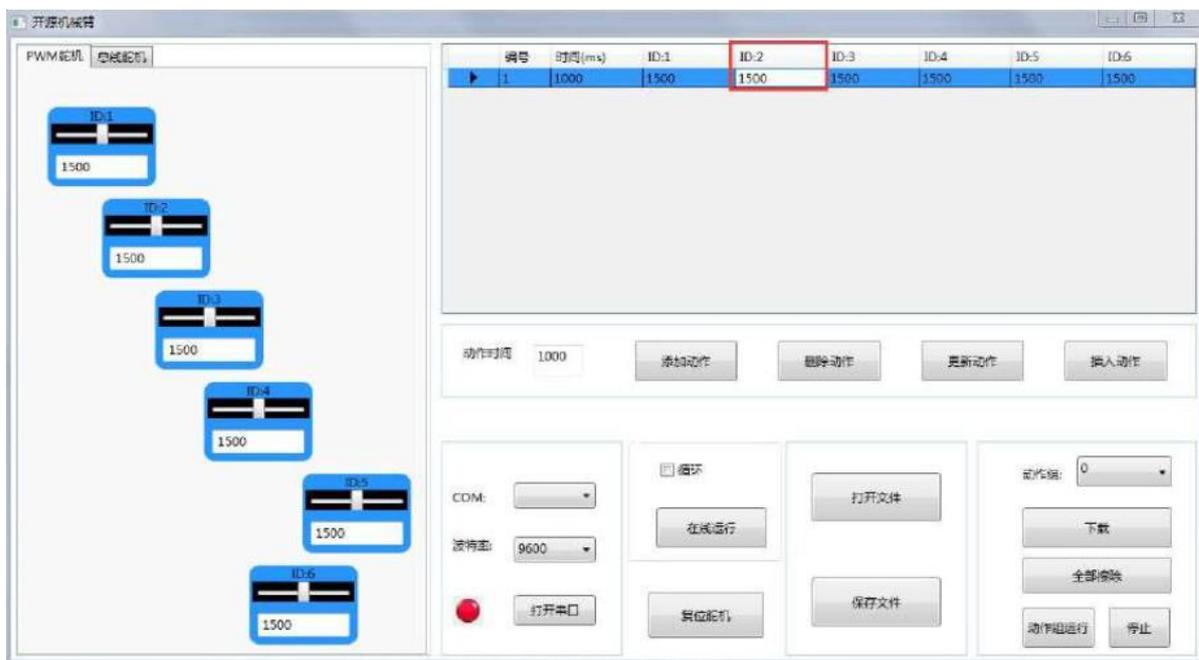


На скриншоте показано только то, на что преимущественно следует обратить внимание. Во-первых, в левом верхнем углу находятся две вкладки: ШИМ-сервоприводы и сервоприводы общей шины. На панели расширения LeArm есть порты для подключения сервоприводов общей шины. При покупке шины для сервоприводов нашего производства, вы можете переключиться на вторую вкладку и управлять роботом по общей шине.

Теперь взглянем на 6 голубых боксов, расположенных в левом окне, в каждом из них прописаны ID, обозначающие положение конкретного сервопривода относительно их подключения к плате расширения (на плате расширения имеется маркировка для подключения сервоприводов).

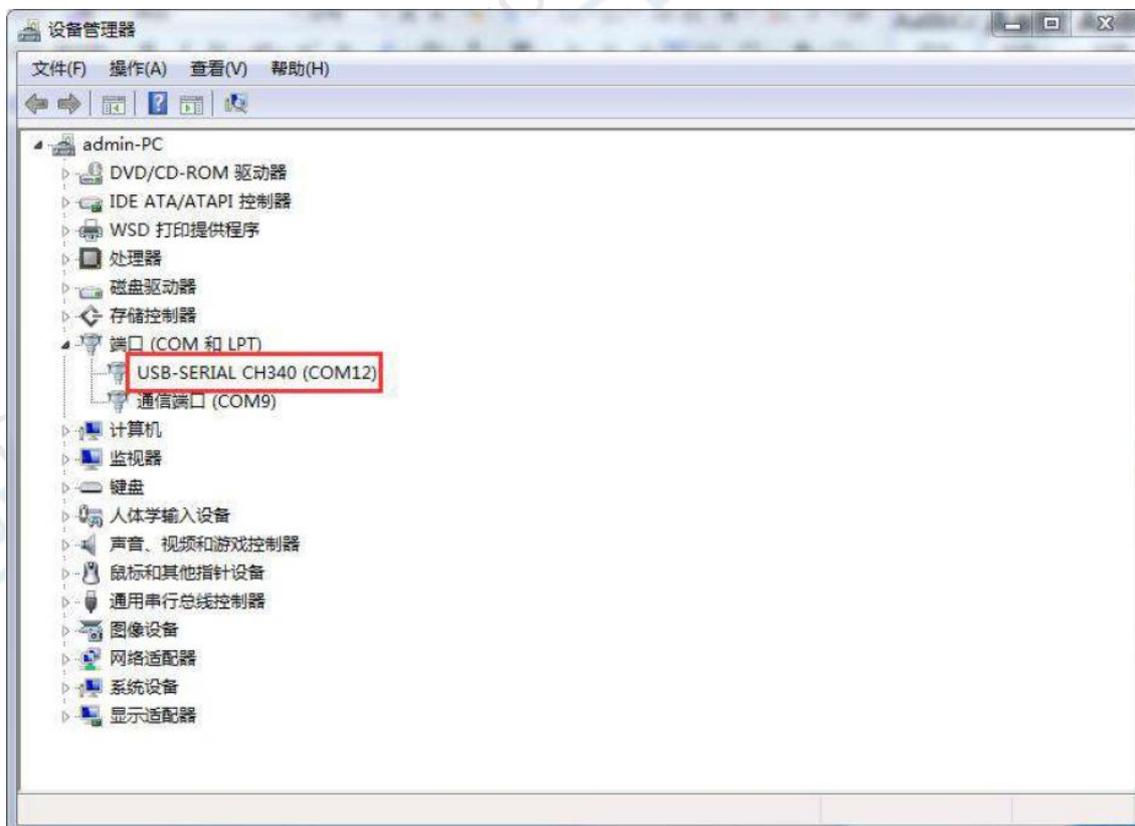
Итак, подключаемся к главному компьютеру. Если к нему в это время подключены другие устройства, они также будут отображаться при выборе COM-порта. Рекомендуется устанавливать нужный порт через диспетчер устройств.

SOM-порт с именем CN340 и есть порт нашего робота-манипулятора, если портов с таким названием несколько, просто выберите один из них. После установки порта передвиньте ползунки сервоприводов. Если рука будет двигаться, это означает, что порт выбран верно, если нет, выберите другой порт и попробуйте снова.



Каждый набор действий вмещает до 255 действий. Если вы хотите прописать больше действий, воспользуйтесь методом по их добавлению, который представлен в справочных материалах.

Обратите внимание, в случае, если вы не отключитесь от главного компьютера, а просто выключите манипулятор или отсоедините USB-кабель, подключение к COM-порту на главном компьютере будет отображаться как все еще активное. При последующем включении робота или соединении USB, отключитесь от COM-порта и подключитесь снова.



5. Джойстик

Подробнее об использовании джойстика вы можете узнать из справочных видео. Управление через джойстик делится на 2 типа: через наборы действий и прямое управление одиночным сервоприводом.

При включении LeArm, управление через набор действий выбрано по умолчанию. При нажатии различных клавиш и ввода переменных плата управления реализует загруженные в нее команды и отправляет на исполнение манипулятору. Переключиться на управление одиночным сервоприводом можно посредством комбинации клавиш SELECT+START, при успешном переключении прозвучит сигнал. Если сигнал не прозвучал, повторите попытку. В режиме управления сервоприводом можно осуществлять управление каждым сервоприводом с помощью кнопок или дистанционно. При необходимости переключиться на режим воспроизведения набора действий, зажмите кнопки SELECT+START. При успешном переключении прозвучит сигнал.

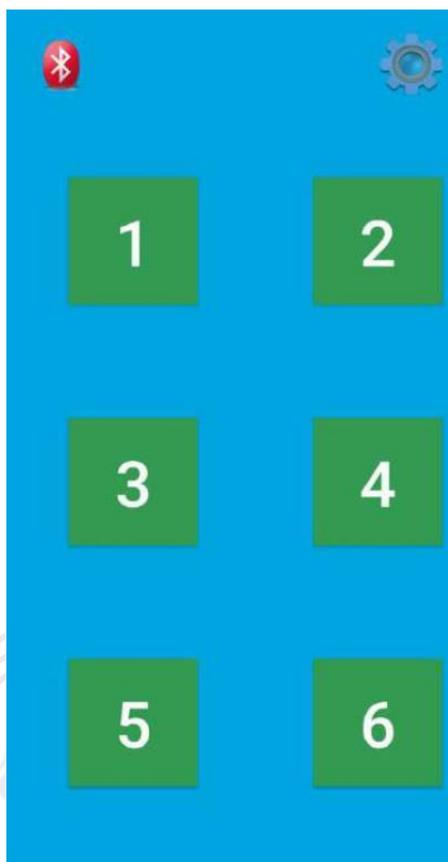
Описание кнопок джойстика представлено в таблице ниже.

| Клавиши | Режим выполнения набора действий | Режим управления одиночным сервоприводом |
|------------------------|--|---|
| START | Запустить 1 раз набор действий №0 | Установить все сервоприводы на 1500 |
| Вперед (левая клавиша) | Запустить 1 раз набор действий №1 | Управление вращением сервопривода №5 |
| Назад | Запустить 1 раз набор действий №2 | Управление вращением сервопривода №5 |
| Налево | Запустить 1 раз набор действий №3 | Управление вращением сервопривода №6 |
| Направо | Запустить 1 раз набор действий №4 | Управление вращением сервопривода №6 |
| △ | Запустить 1 раз набор действий №5 | Управление вращением сервопривода №4 |
| × | Управление вращением сервопривода №6 | Управление вращением сервопривода №4 |
| □ | Запустить 1 раз набор действий №7 | Управление вращением сервопривода №3 |
| ○ | Запустить 1 раз набор действий №8 | Управление вращением сервопривода №3 |
| L1 | Запустить 1 раз набор действий №9 | Управление вращением сервопривода №2 |
| R1 | Запустить 1 раз набор действий №10 | Управление вращением сервопривода №2 |
| L2 | Запустить 1 раз набор действий №11 | Управление вращением сервопривода №1 |
| R2 | Запустить 1 раз набор действий №12 | Управление вращением сервопривода №1 |
| SELECT+START | Переключение в режим одиночного сервопривода | Переключение в режим выполнения набора команд |

6. Управление через смартфон

Для управления через смартфон требуется установить специальное приложение. После установки запустите приложение, включить на смартфоне Bluetooth и подключитесь к роботу.

После открытия приложения нажмите на красный значок Bluetooth в верхнем левом углу, в всплывающем списке выберите устройство LOBOT, нажмите "подключиться". При успешном подключении красная иконка Bluetooth станет синей. При нажатии кнопок 1-6 осуществляется вызов соответствующих наборов действий (необходимо предварительно загрузить наборы действий на плату управления).



7. Пассивное управление

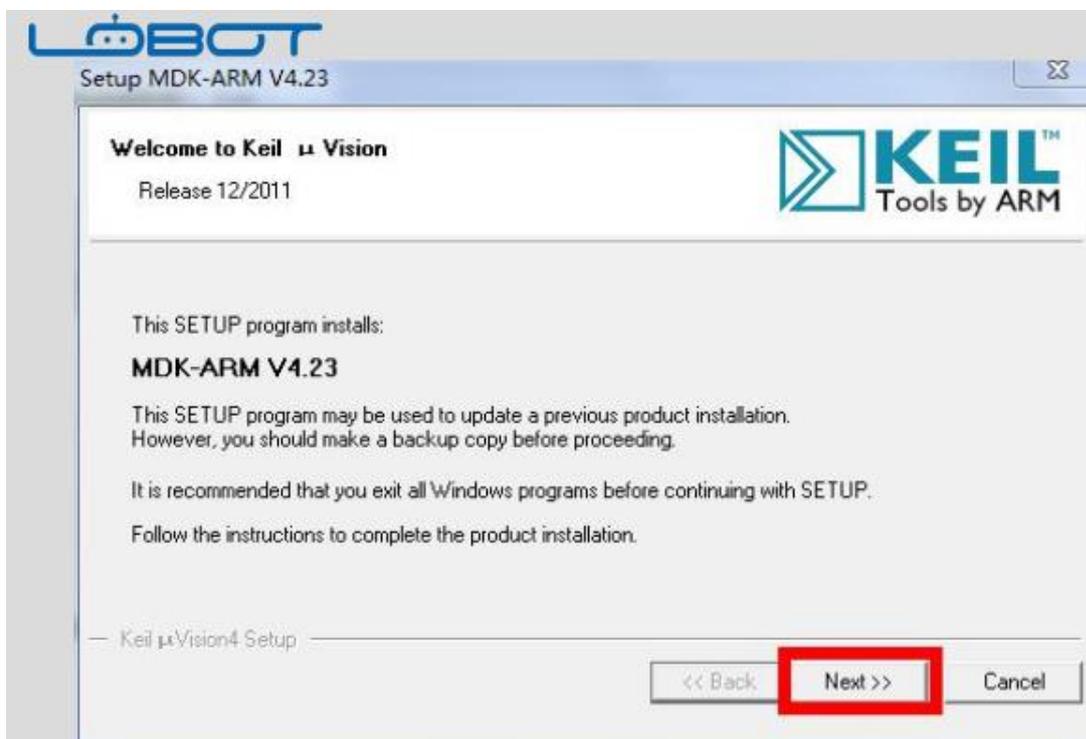
В нижнем углу контроллера LeArm есть кнопка пассивного управления 1. Загрузите на плату управления набор, включающий 100 действий, и нажмите эту кнопку. Манипулятор выполнит набор из 100 действий, а при зажатии кнопки на 3 робот будет выполнять цикл действий по кругу. Для остановки цикла следует отключить манипулятор.

8. Создание среды разработки и программирование

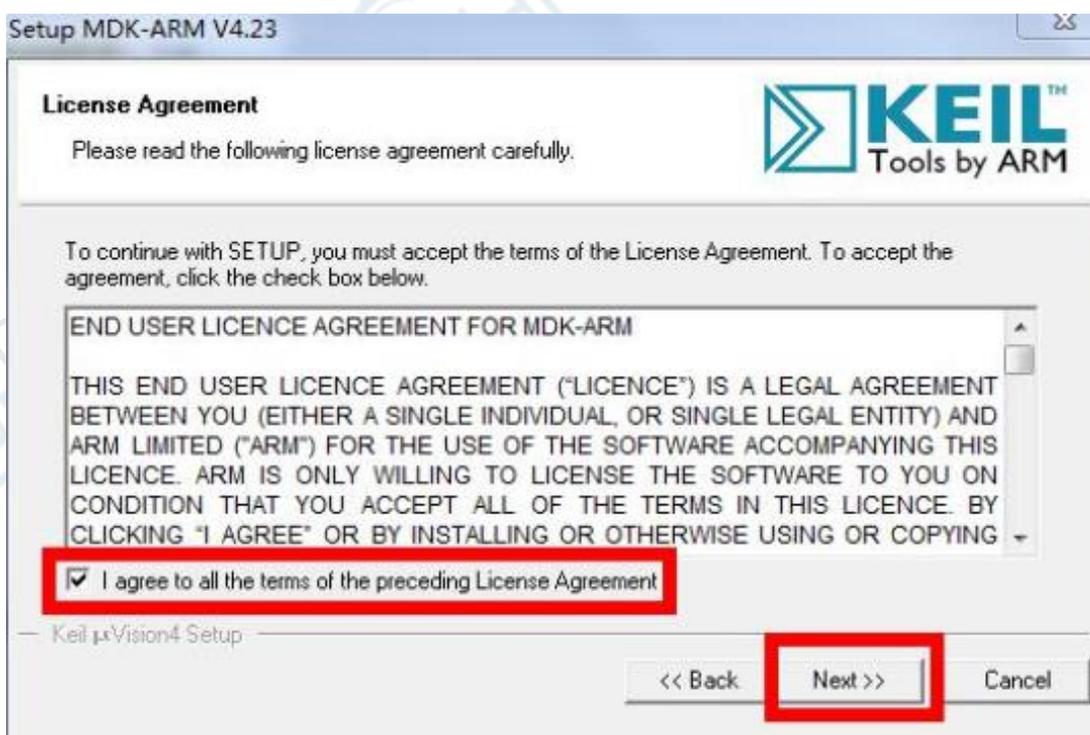
8.1 MDK-ARM установка и программирование

В независимости от версии, установка и программирование проходит одинаково.

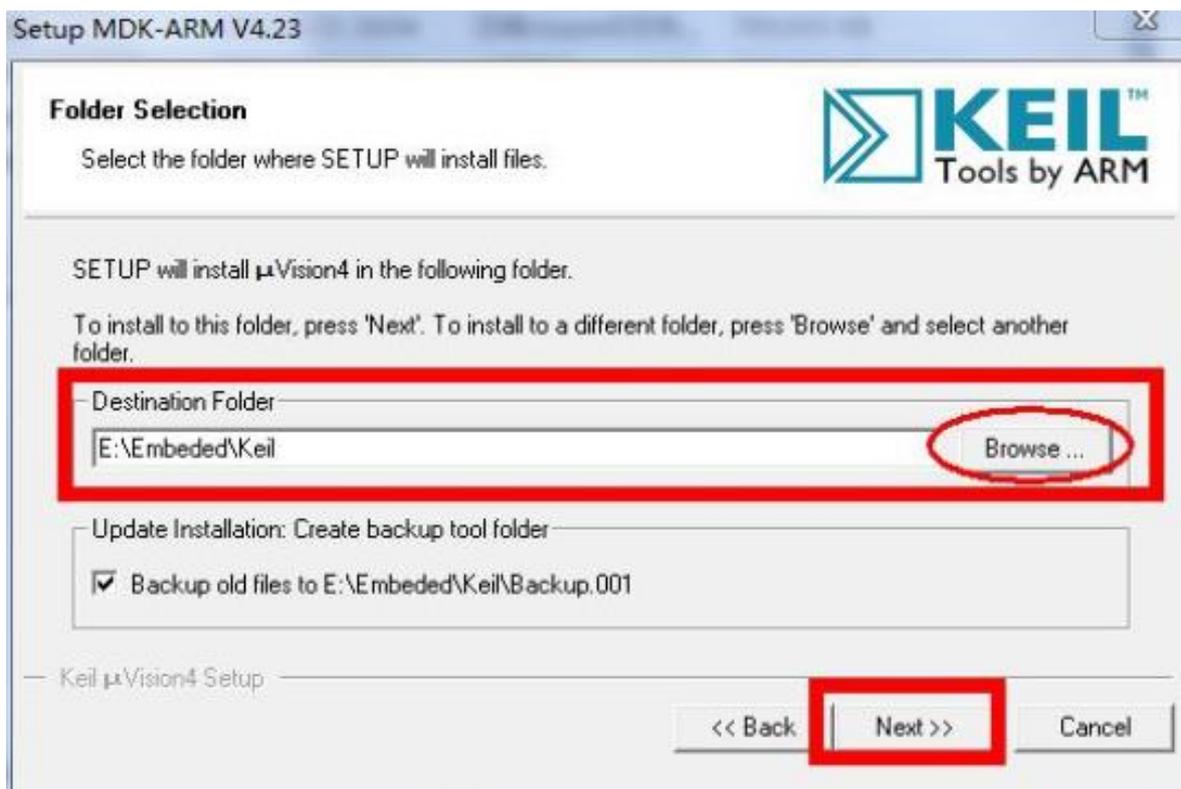
1. Запустите mdk.exe и подготовьтесь к установке MDK-ARM



2. Поставьте галочку на соглашение с правилами установки и использования.



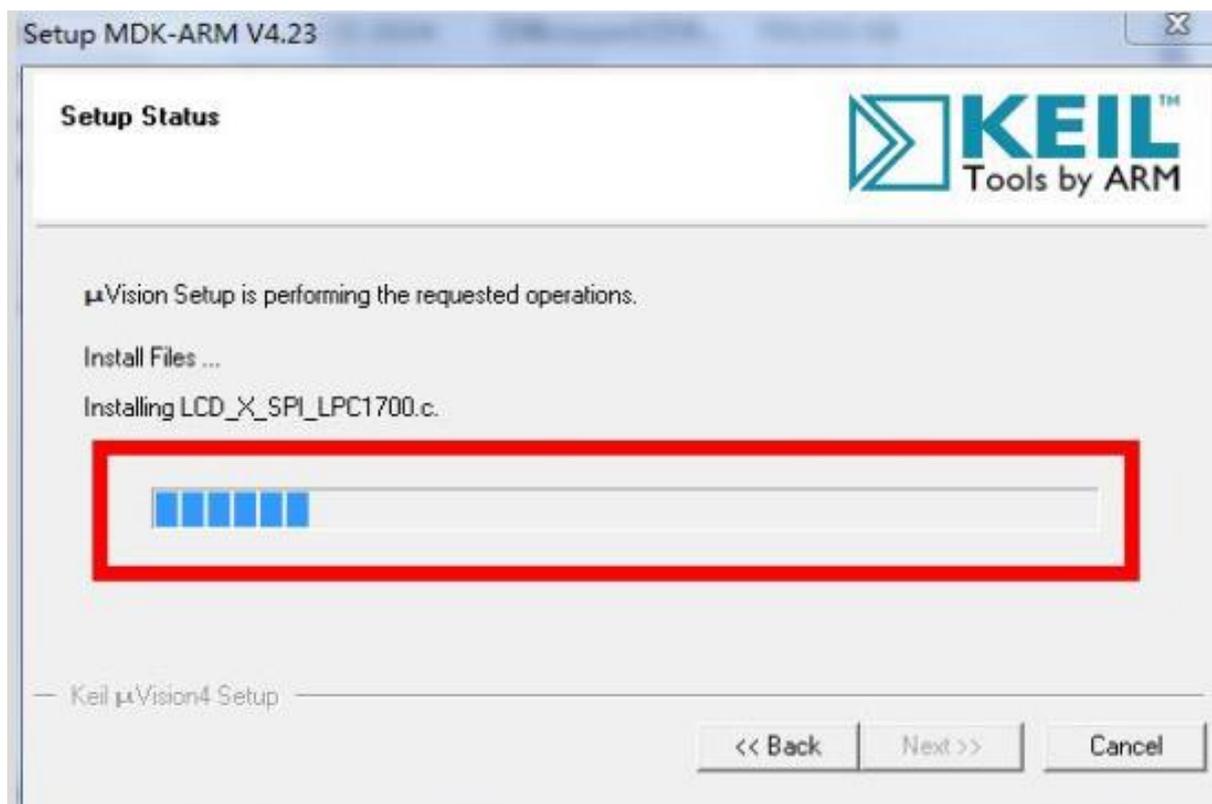
3. Выберите папку для установки



4. Введите данные пользователя (Не обязательно писать подлинные данные)



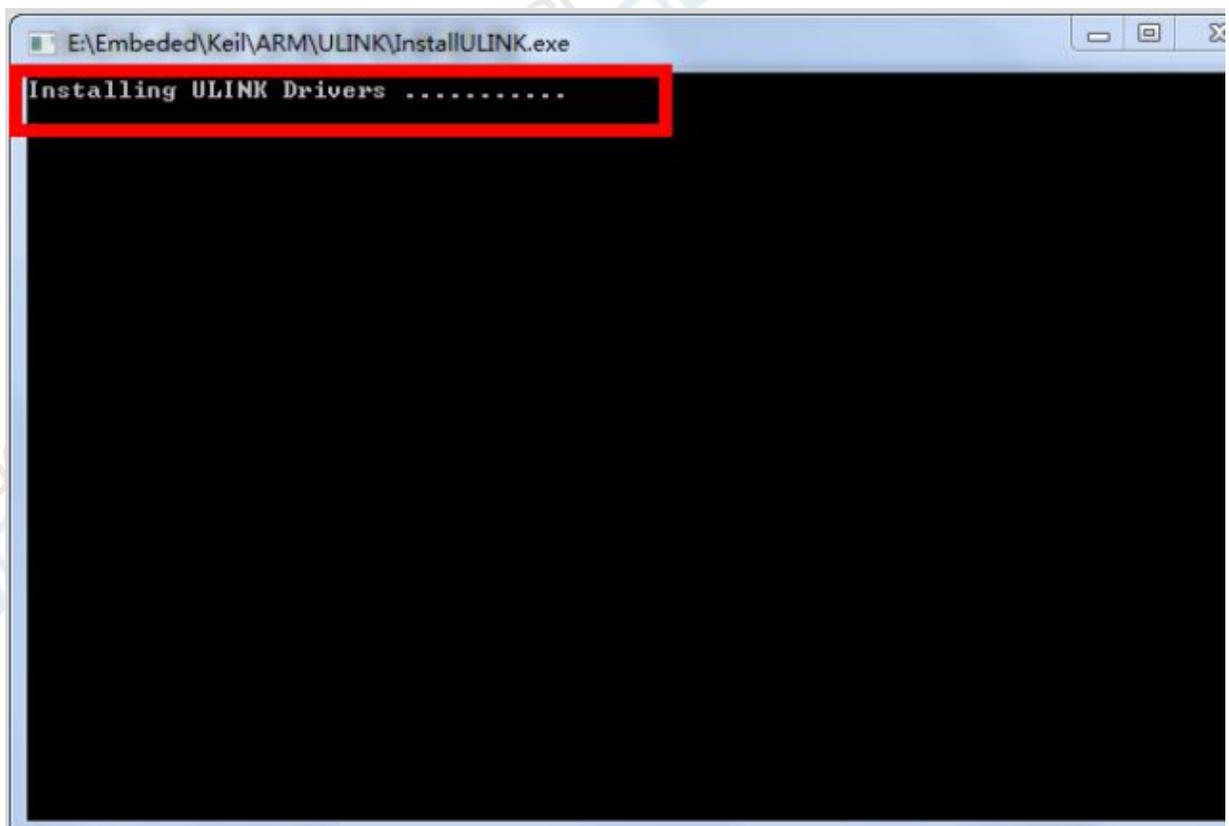
5. Установка



6. Завершение установки

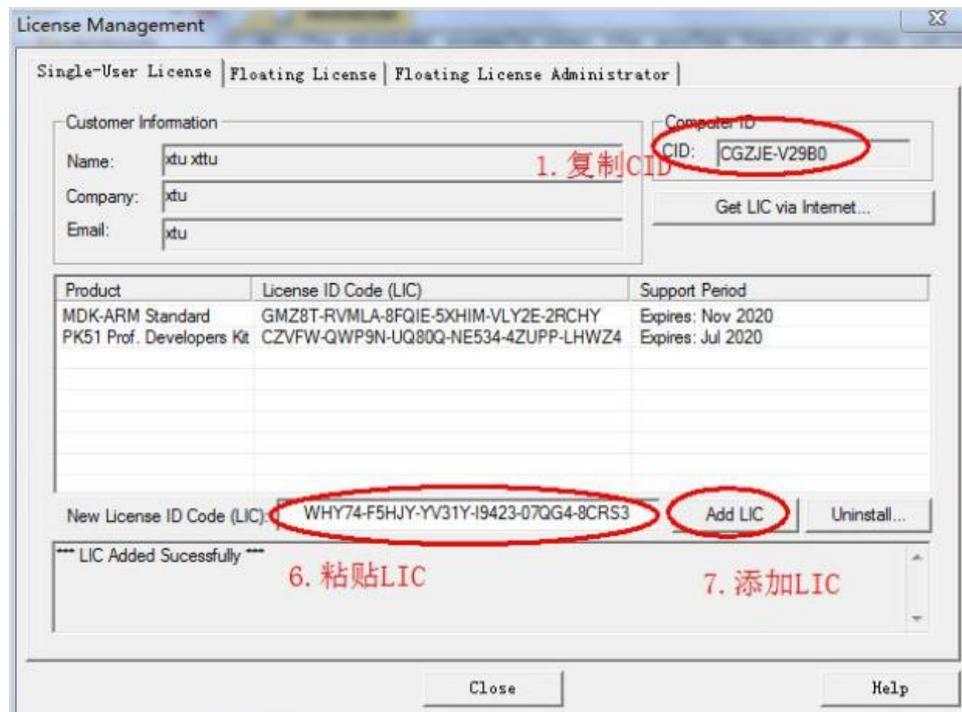


7. Установка драйвера Ulink



Программирование ARM-MDK V4.23

Запустите uVision4, нажмите File>>License Management, скопируйте CID, после чего откройте Keil_Lic, вставьте CID и выберите ARM. Нажмите generate для создания Lic.

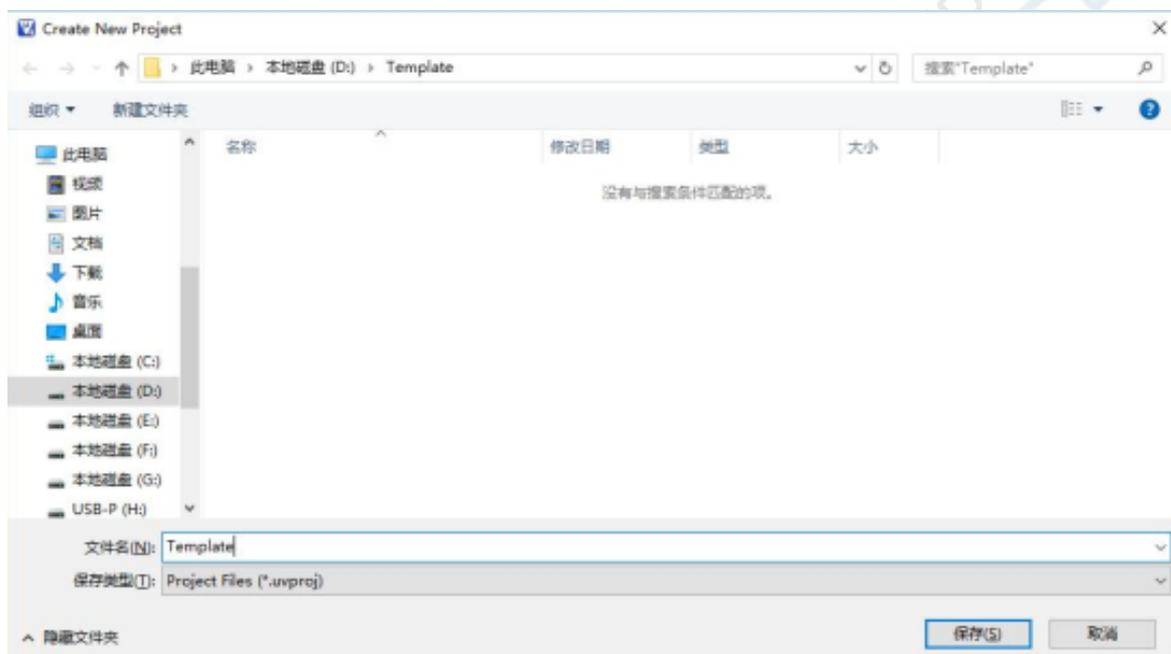


Создание нового проекта

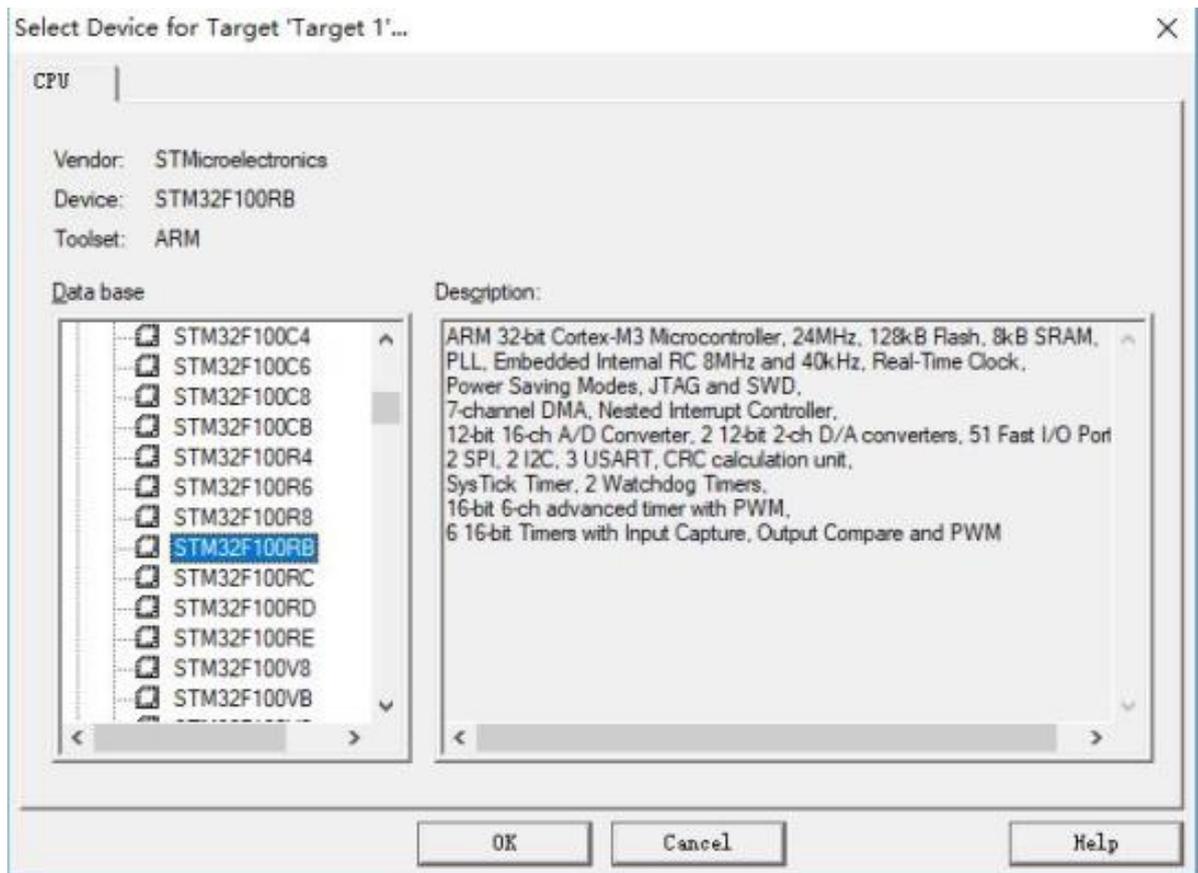
1. Для закрытия старого проекта откройте главную страницу MDK. На главной странице вы увидите список уже созданных проектов. Нажмите на название проекта и в меню выберите: Project->Close Project. Таким способом вы сможете закрыть все ненужные вам проекты, очистив главную страницу MDK. Далее мы рассмотрим, как создать новый проект.

2. Перед созданием проекта рекомендуем создать на вашем компьютере папку, для хранения всех необходимых для нового проекта данных. Для примера мы создали папку под названием STM32-Projects.

3. Откройте меню Keil: Project->NewUvisionProject, после выберите специально созданную папку STM32-Projects, и создайте в ней папку Template. Далее в меню выберите папку Template. Новый проект мы будем сохранять в данной папке. Новый проект назовём Template. Нажмите «Сохранить».

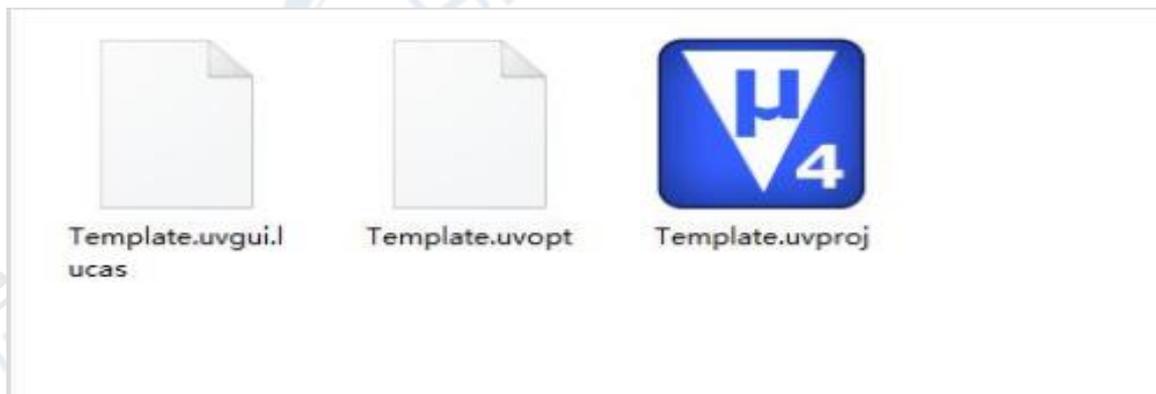


4. После появления окошко «Device», откройте диск где находится STMicroelectronics и выберите STM32F103RB.

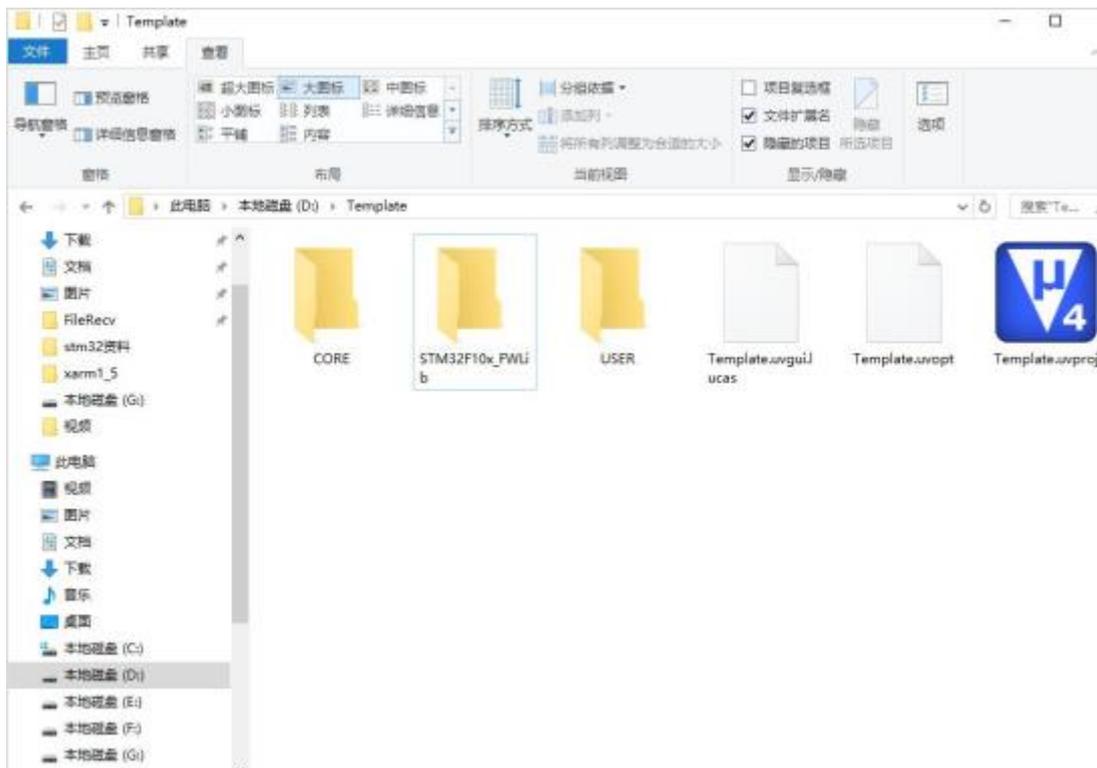


5. В сплывающем окне «CopySTM32StartupCodetoproject...» появится предложение «добавить код запуска?» выберите «нет». В нашем встроенном программном обеспечении уже встроен документ код запуска.

6. После возвращения в Template мы увидим, что там появились 3 файла:

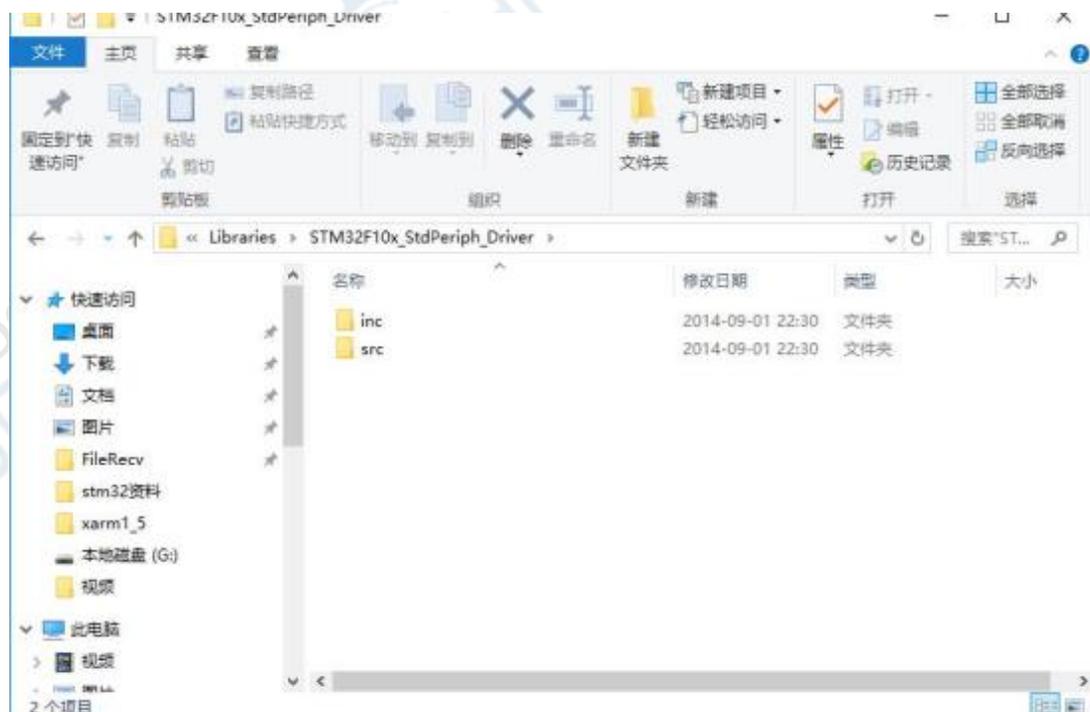


В Template необходимо создать три папки: CORE, USER, STM32F10x_FWLib.



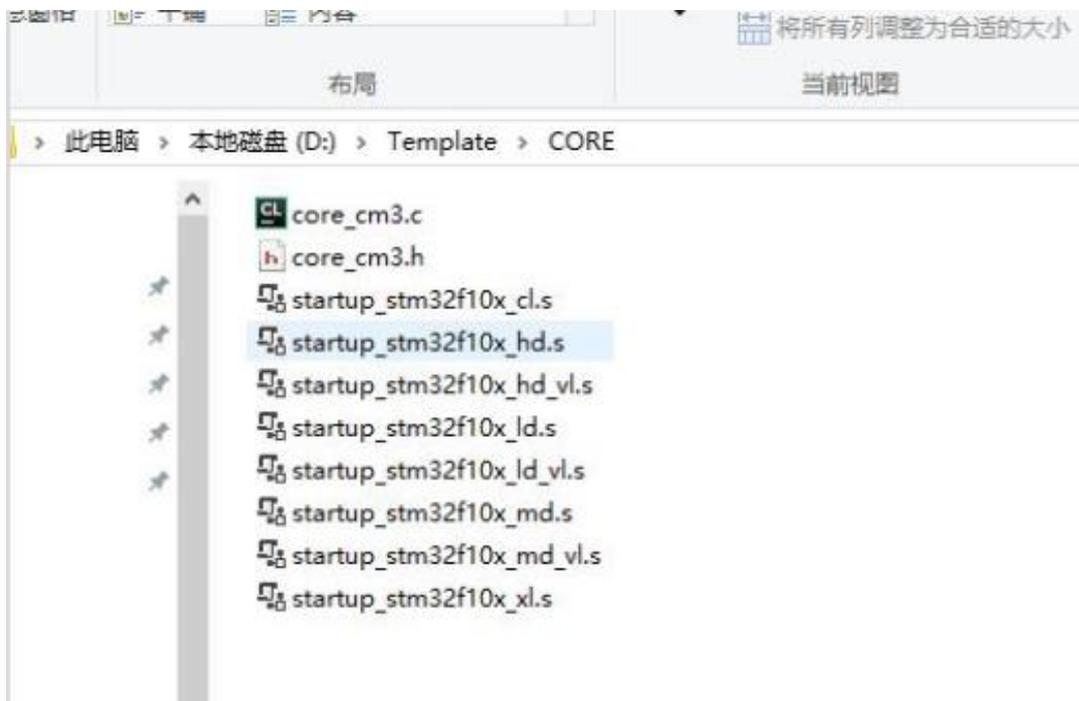
7. Далее необходимо из встроенного программного обеспечения скопировать исходный код и вставить его в наш новый проект. Для этого необходимо открыть папки со встроенным программным обеспечением: STM32F10x_StdPeriph_Lib_V3.5.0\Libraries\STM32F10x_StdPeriph_Driver. Скопировать src,inc и вставить в STM32F10x_FWLib.

8.



9. Сейчас нам необходимо скопировать из встроенного программного обеспечения все необходимые для движения папки в CORE.

Откройте встроенное программное обеспечение:
STM32F10x_StdPeriph_Lib_V3.5.0\Libraries\CMSIS\CM3\CoreSupport ,
скопируйте core_cm3.c и core_cm3.h и переместите их в CORE. После чего перейдите в
STM32F10x_StdPeriph_Lib_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\sta
rtup\arm скопируйте все находящиеся там элементы и перенесите в CORE.



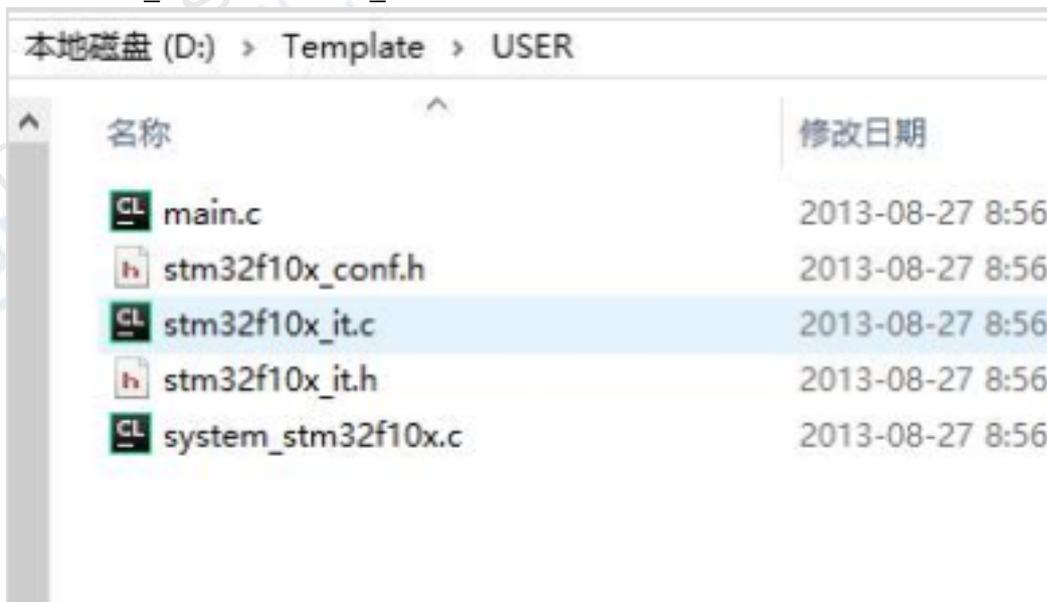
10. Перемещение в каталог

По данному адресу

STM32F10x_StdPeriph_Lib_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x
находятся три файла: stm32f10x.h, system_stm32f10x.c и system_stm32f10x.h .

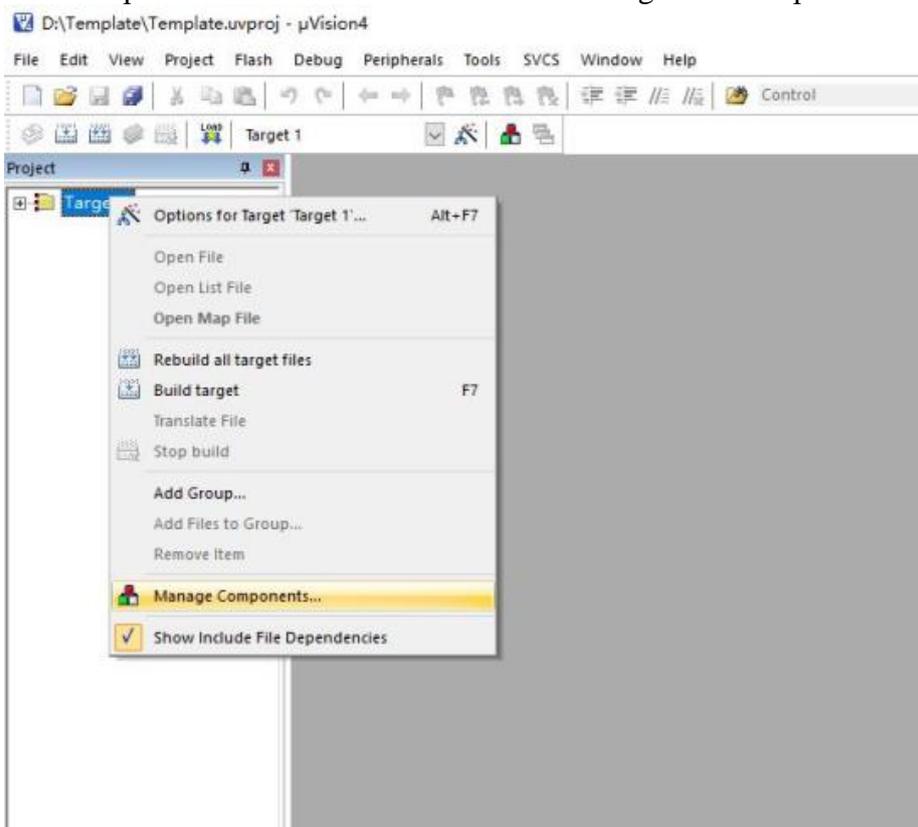
Их необходимо переместить в наш USER каталог.

После чего с STM32F10x_StdPeriph_Lib_V3.5.0\Project\STM32F10x_StdPeriph_Template
необходимо скопировать и вставить в USER 4 файла: main.c, stm32f10x_conf.h,
stm32f10x_it.c, stm32f10x_it.h



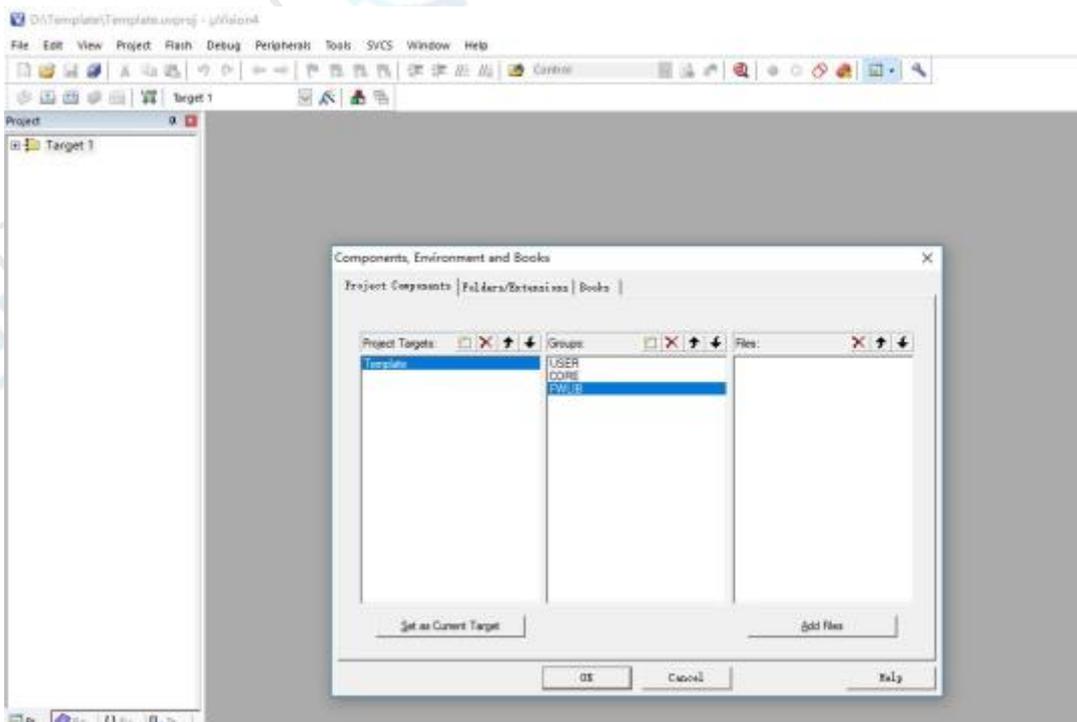
11. Все проделанные в основном программном обеспечении 10 этапов нам необходимо скопировать в наш новый проект.

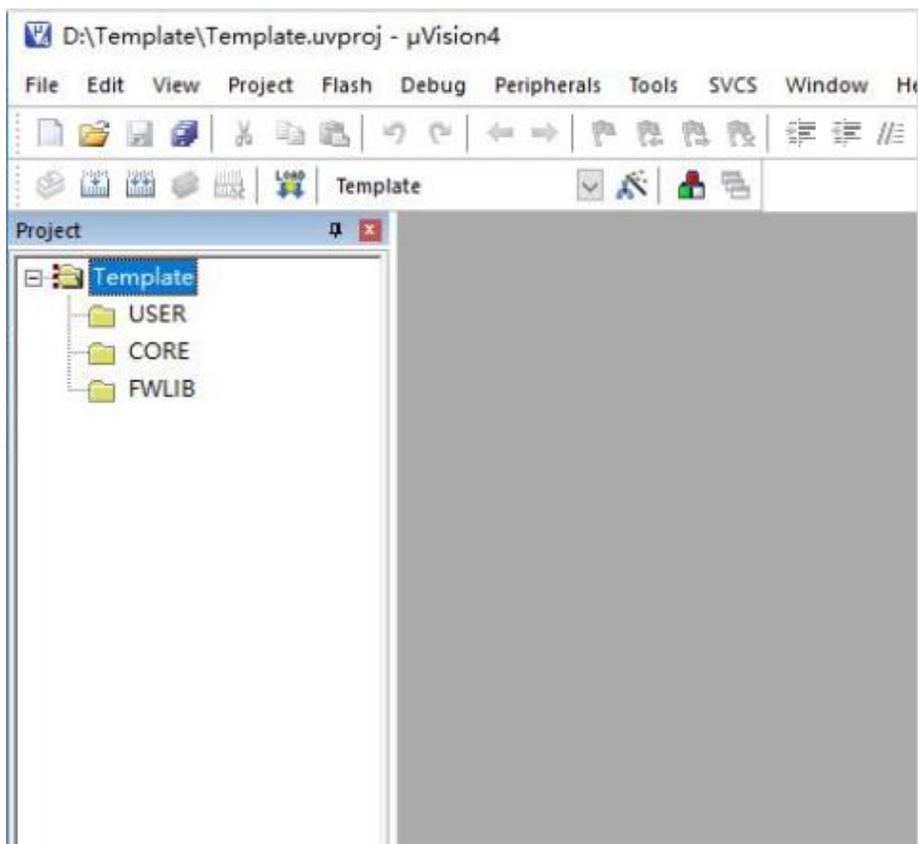
Правой кнопкой мыши нажмите на Target1 и выберите Manage Components.



12. Project Targets

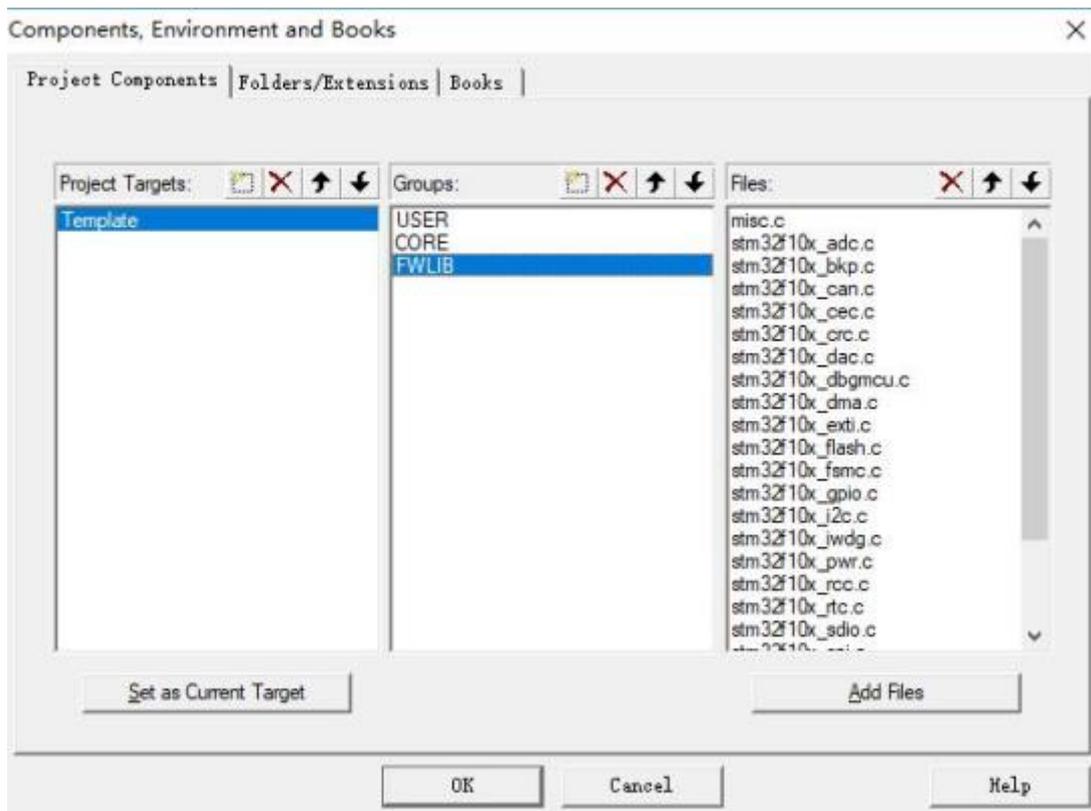
Переименуйте Target в Template, затем в Groups удалите один столбец и создайте три группы: USER, CORE, FWLIB, нажмите «ок».



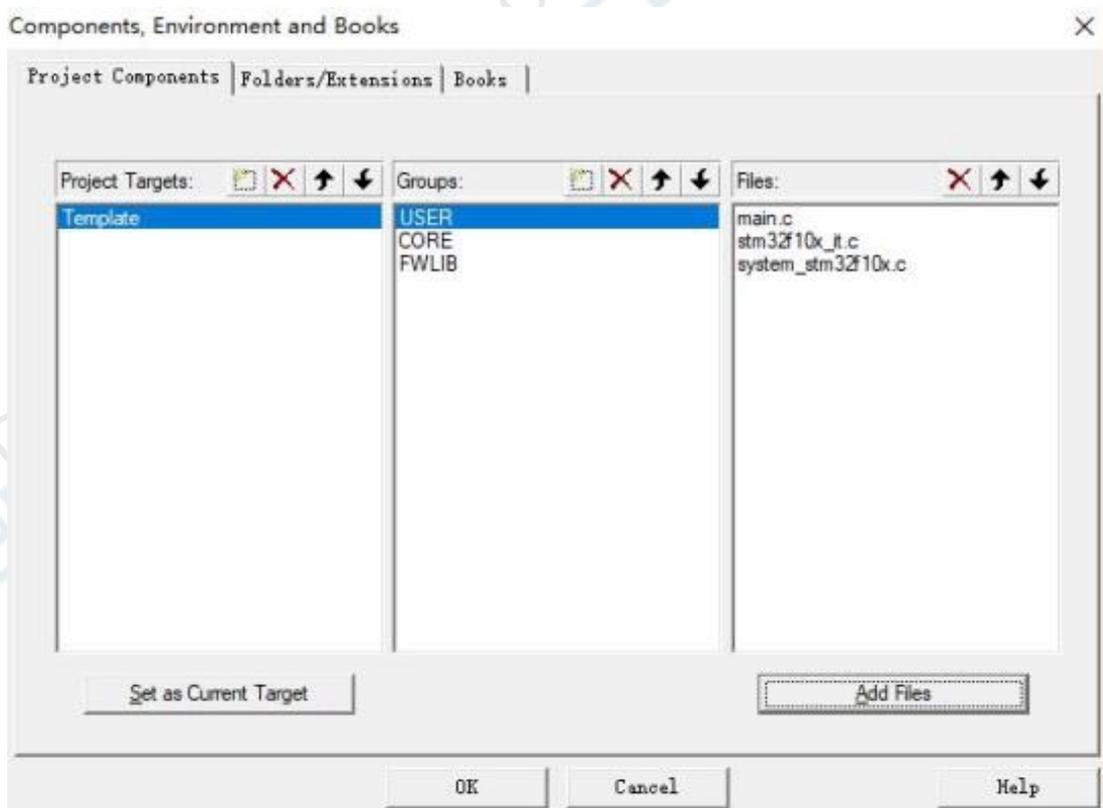


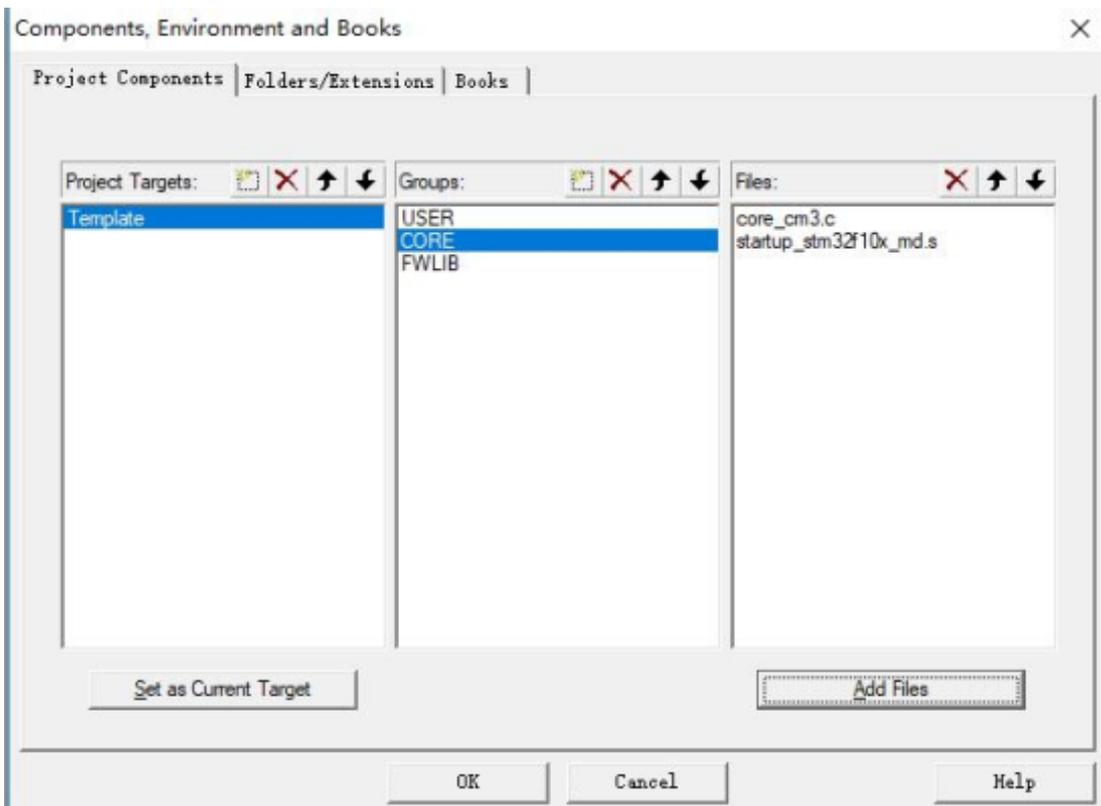
13. Добавляем в Group все необходимые нам файлы.

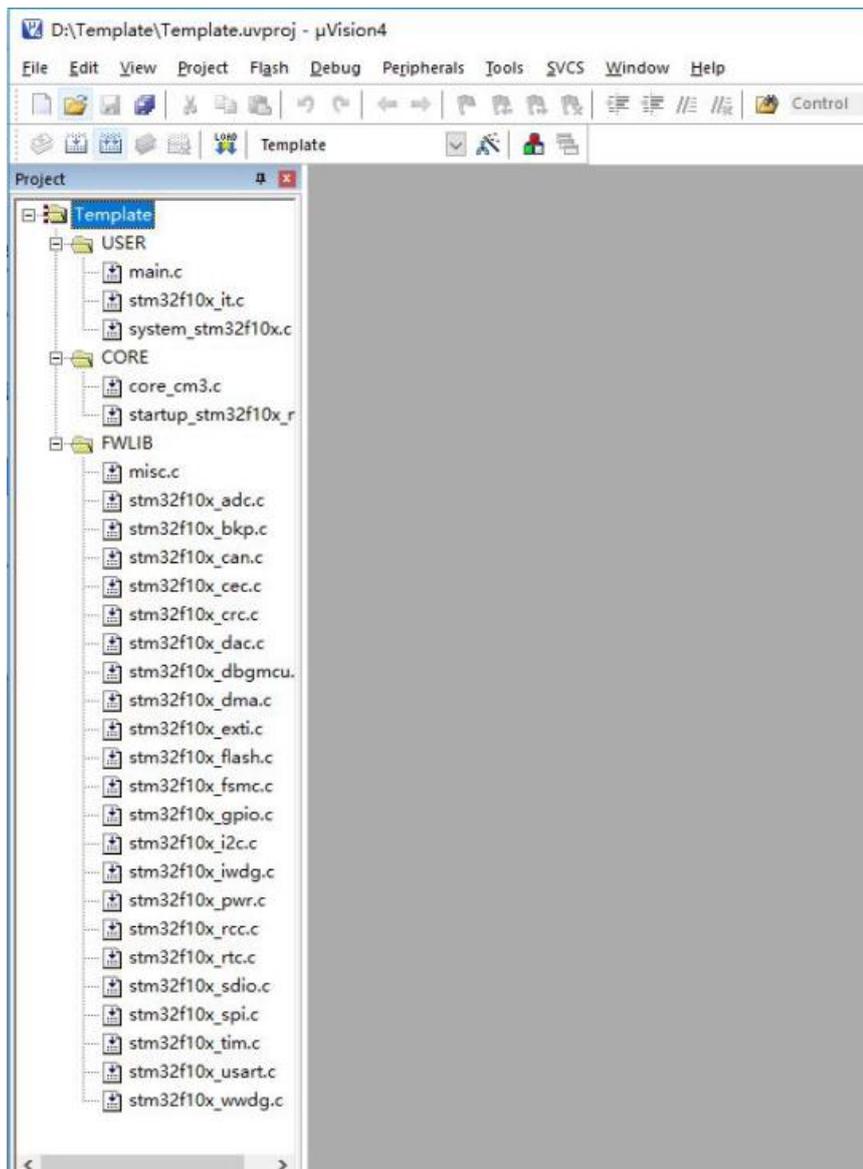
Правой кнопкой мыши нажимаем на Template, выбираем Manage Components и добавляем все необходимые нам файлы в Group. Во-первых выбираем FWLIB, нажимаем на Add Files, заходим в STM32F10x_FWLib/src, с помощью Ctrl+A выбираем все файлы и нажимаем Add и после Close.



14. С помощью аналогичного способа добавляем из Group в CORE и USER все необходимые нам файлы. В CORE добавляем core_cm3.c, startup_stm32f10x_md.s, USER, после добавляем main.c, stm32f10x_it.c, system_stm32f10x.c и нажимаем «ок».



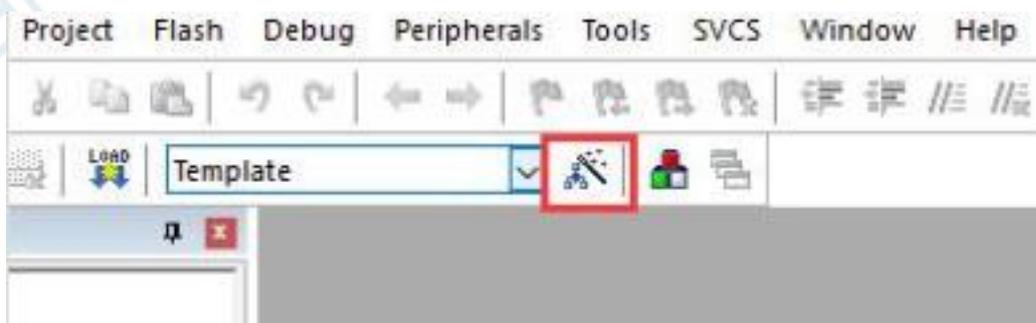


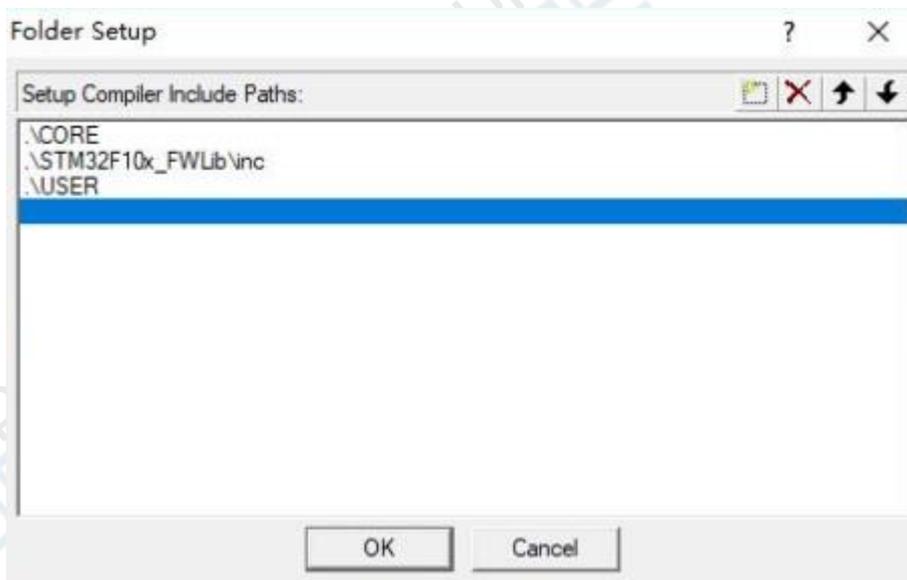
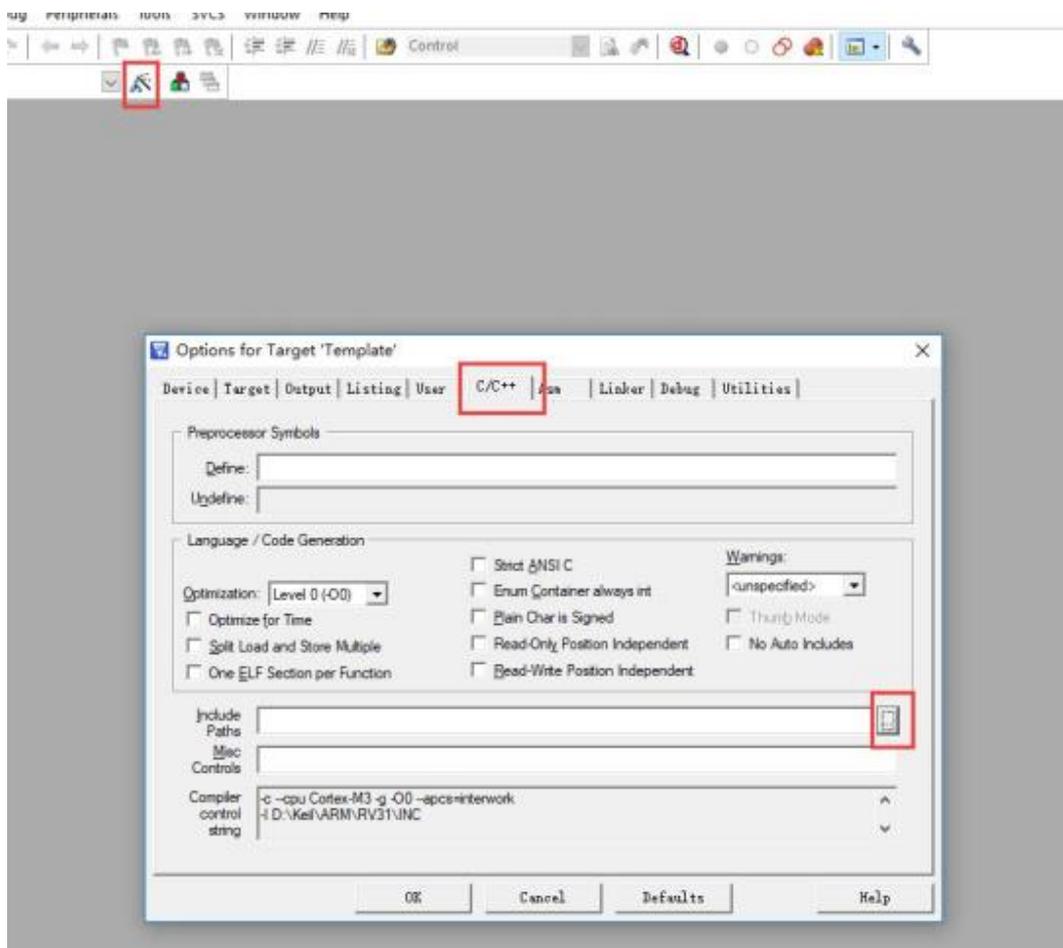


15. Далее нам необходимо запрограммировать МК на самостоятельное определение местонахождения необходимых файлов.

Вернитесь в главное меню нового проекта, нажмите на «волшебную палочку» и выберите c/c++, затем нажмите кнопку справа от IncludePaths и добавьте три проекта (как на картинке).

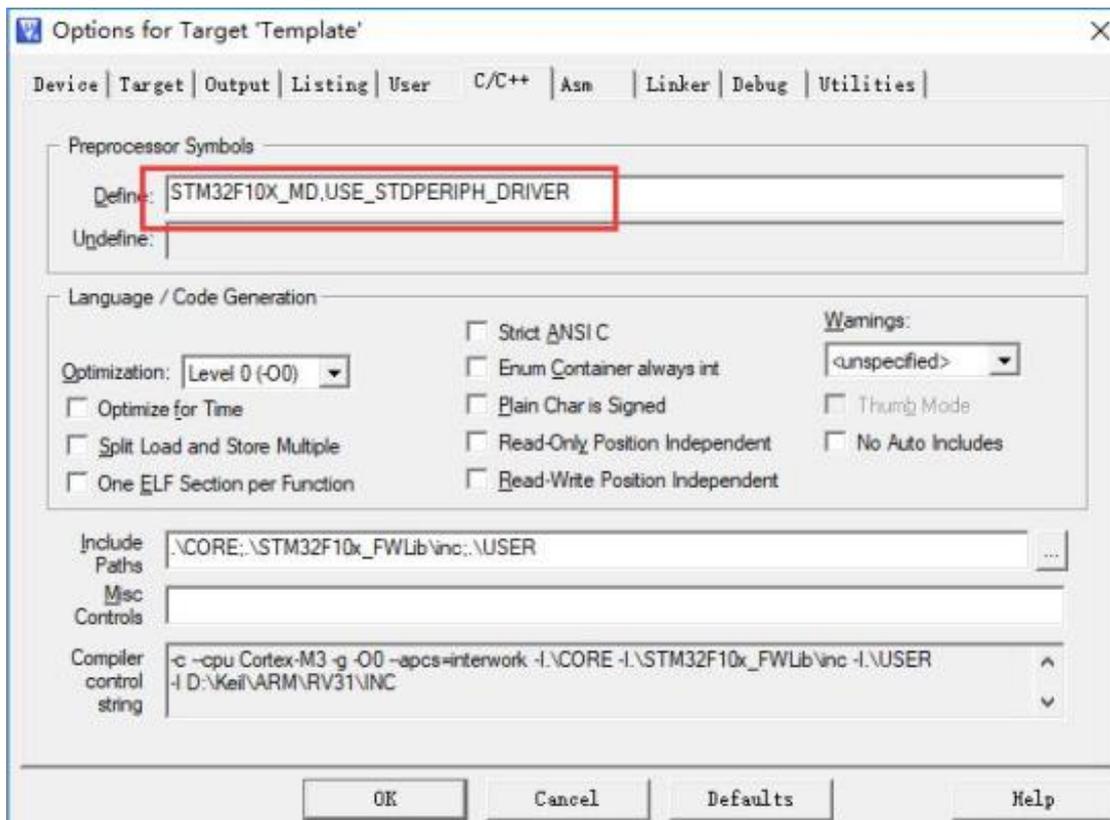
Keil должен находиться только в главном каталоге, а path в последнем.





16. Библиотека ST поддерживает несколько моделей микроконтроллеров

Как и в 15 шаге, нажмите на c/c++ и скопируйте вставьте: STM32F10X_MD,USE_STDPERIPH_DRIVER в Define.

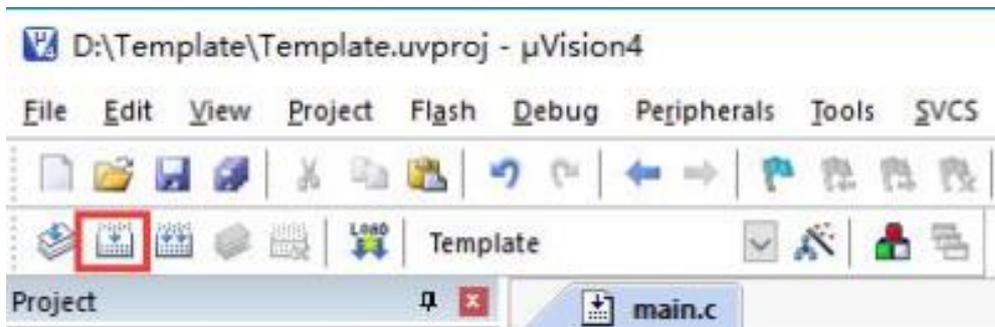


17. Зайдите в USUR проекта и скопируйте и вставьте код в main.c ,

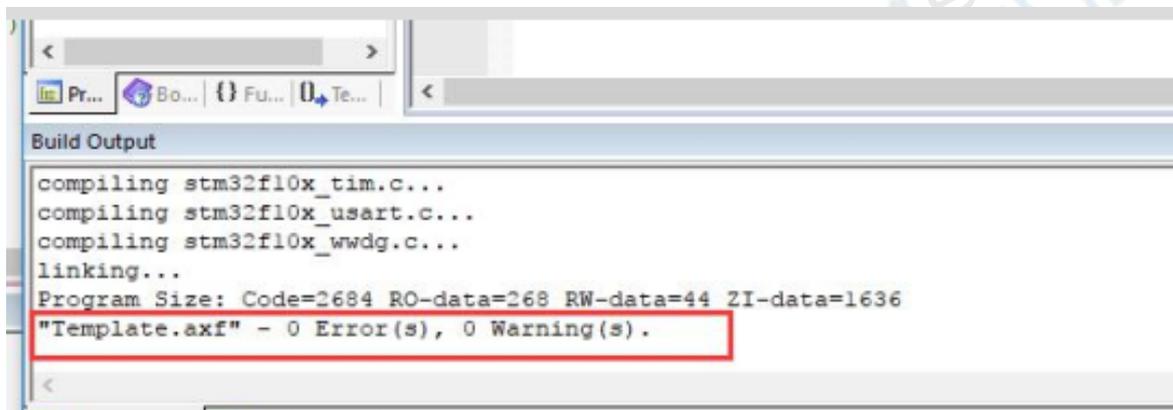
```
#include"stm32f10x.h"
#include"stm32f10x_gpio.h"
GPIO_InitTypeDefGPIO_InitStructure;
intmain(void)
{
SystemInit();
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB,ENABLE);
GPIO_InitStructure.GPIO_Pin=GPIO_Pin_9;
GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode=GPIO_Mode_Out_PP;
GPIO_Init(GPIOB,&GPIO_InitStructure);
while(1)
{
/*SetPB9*/
GPIO_SetBits(GPIOB,9);
/*ResetPB9*/
GPIO_ResetBits(GPIOB,9);
```

}
}

18. Для компиляции нажмите на кнопку как на картинке



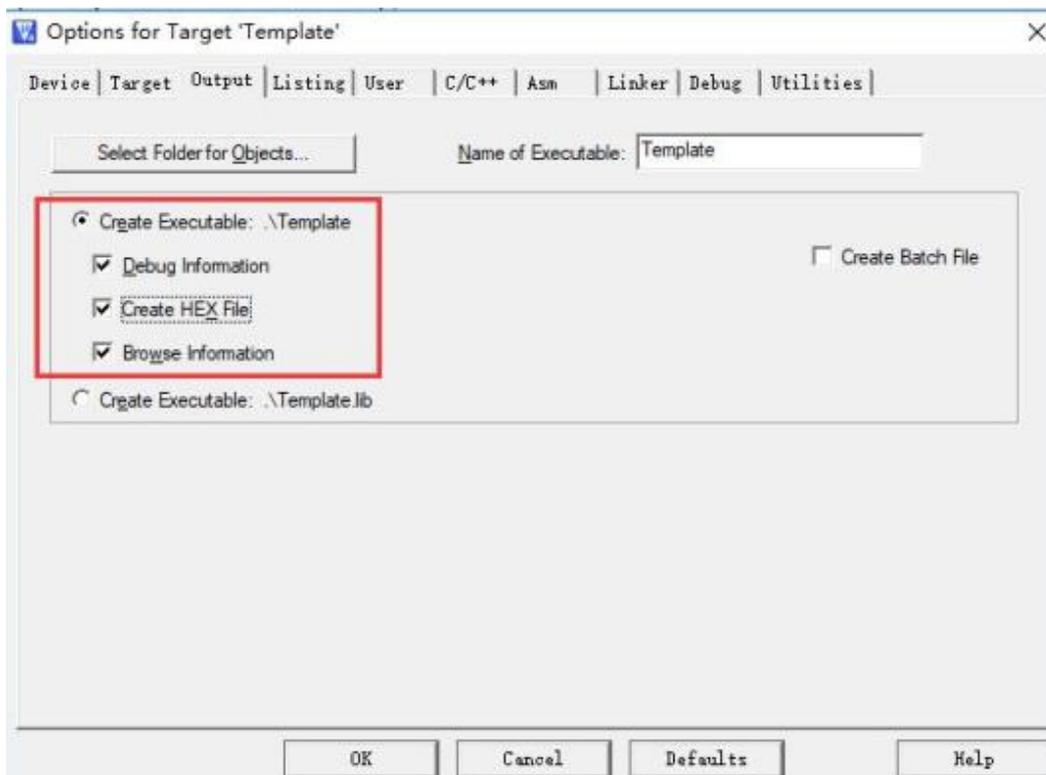
Результат компиляции



На картинке видно, что компиляция прошла успешно.

После успешной компиляции необходимо провести настройку и преобразование файлов в hex файл.

Нажмите на «волшебную палочку», в настройках выберите Output и поставьте галочки на всех вариантах выбора.



8.2 Толкование программы

На основании приведенной выше информации мы можем написать функцию инициализации порта ввода-вывода LED.

```
void InitLED(void)
{
    GPIO_InitTypeDef  GPIO_InitStructure;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE); //使能PB端口时钟
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9; //LED2 PB9 端口配置
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure); //配置IO
}

```

```
#define LED_ON    0
#define LED_OFF  1

```

Сейчас мы можем записать 0 или 1 в IO для управления включением или выключением светодиода.

```
LED = LED_ON; //亮灯

```

LED_ON это макрокоманда, которая обозначается 0.

Так же мы можем видеть, что просто LED это тоже макрокоманда, обозначающая PBout(9).

```
#define LED PBout(9)
```

```
#define BITBAND(addr, bitnum) ((addr & 0xF0000000)+0x2000000+((addr & 0xFFFF)<<5)+(bitnum<<2))
#define MEM_ADDR(addr) *((volatile unsigned long *) (addr))
#define BIT_ADDR(addr, bitnum) MEM_ADDR(BITBAND(addr, bitnum))

//IO口地址映射
#define GPIOA_ODR_Addr (GPIOA_BASE+12) //0x4001080C
#define GPIOB_ODR_Addr (GPIOB_BASE+12) //0x40010C0C
#define GPIOC_ODR_Addr (GPIOC_BASE+12) //0x4001100C
#define GPIOD_ODR_Addr (GPIOD_BASE+12) //0x4001140C
#define GPIOE_ODR_Addr (GPIOE_BASE+12) //0x4001180C
#define GPIOF_ODR_Addr (GPIOF_BASE+12) //0x40011A0C
#define GPIOG_ODR_Addr (GPIOG_BASE+12) //0x40011E0C

#define GPIOA_IDR_Addr (GPIOA_BASE+8) //0x40010808
#define GPIOB_IDR_Addr (GPIOB_BASE+8) //0x40010C08
#define GPIOC_IDR_Addr (GPIOC_BASE+8) //0x40011008
#define GPIOD_IDR_Addr (GPIOD_BASE+8) //0x40011408
#define GPIOE_IDR_Addr (GPIOE_BASE+8) //0x40011808
#define GPIOF_IDR_Addr (GPIOF_BASE+8) //0x40011A08
#define GPIOG_IDR_Addr (GPIOG_BASE+8) //0x40011E08

//io口操作,只对单一的io口!
//确保n的值小于16!
#define PAout(n) BIT_ADDR(GPIOA_ODR_Addr,n) //输出
#define PAin(n) BIT_ADDR(GPIOA_IDR_Addr,n) //输入

#define PBout(n) BIT_ADDR(GPIOB_ODR_Addr,n) //输出
```

Данные коды мы можем найти в include.h.

Преобразование LED в PBout(9)

PBout(9) в BIT_ADDR(GPIOB_ODR_Addr, 9)

BIT_ADDR(GPIOB_ODR_Addr, 9) в

MEM_ADDR(BITBAND(GPIOB_ODR_Addr,9)

MEM_ADDR(BITBAND(GPIOB_ODR_Addr, 9) в *((volatile

unsigned long*)(BITBAND(GPIOB_ODR_Addr, 9))) в *((volatile unsigned

long*)((GPIOB_ODR_Addr &

0xF0000000)+0x2000000

+((GPIOB_ODR_Addr & 0xFFFF)<<5)+(9 <<2)))

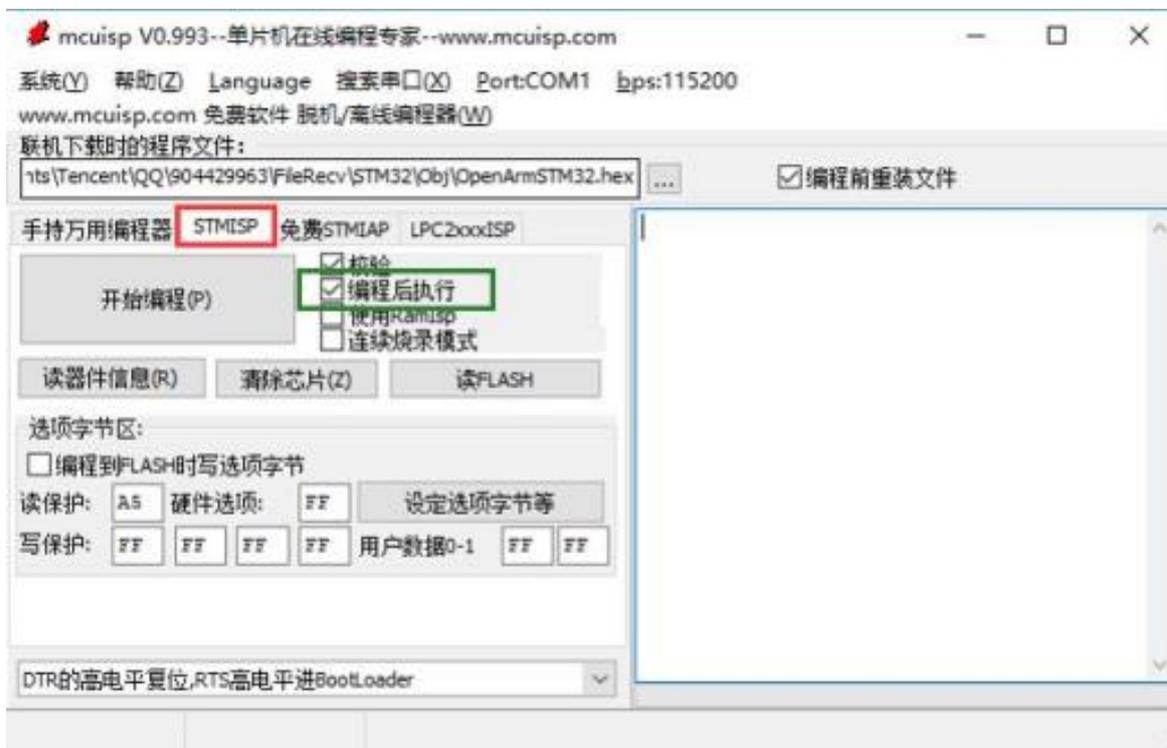
Здесь использовалась функция битового поля stm32. Регистры Stm32 не могут быть обработаны в битах. Чтобы управлять конкретным портом ввода-вывода, мы должны использовать комбинацию сдвига логических операций. Для удобства использования io портов, stm32 предоставляет функцию битового поля.

8.3 Порядок записи

1. Откройте программу для записи



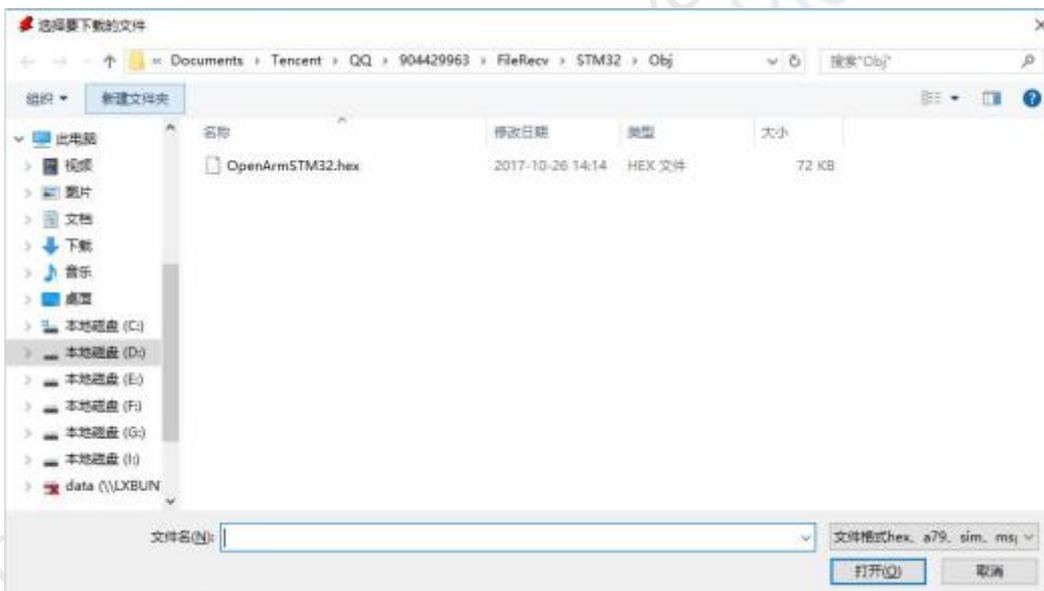
mcuisp.exe



Выберете STMISP, остальные пункты оставьте выбранными по умолчанию.

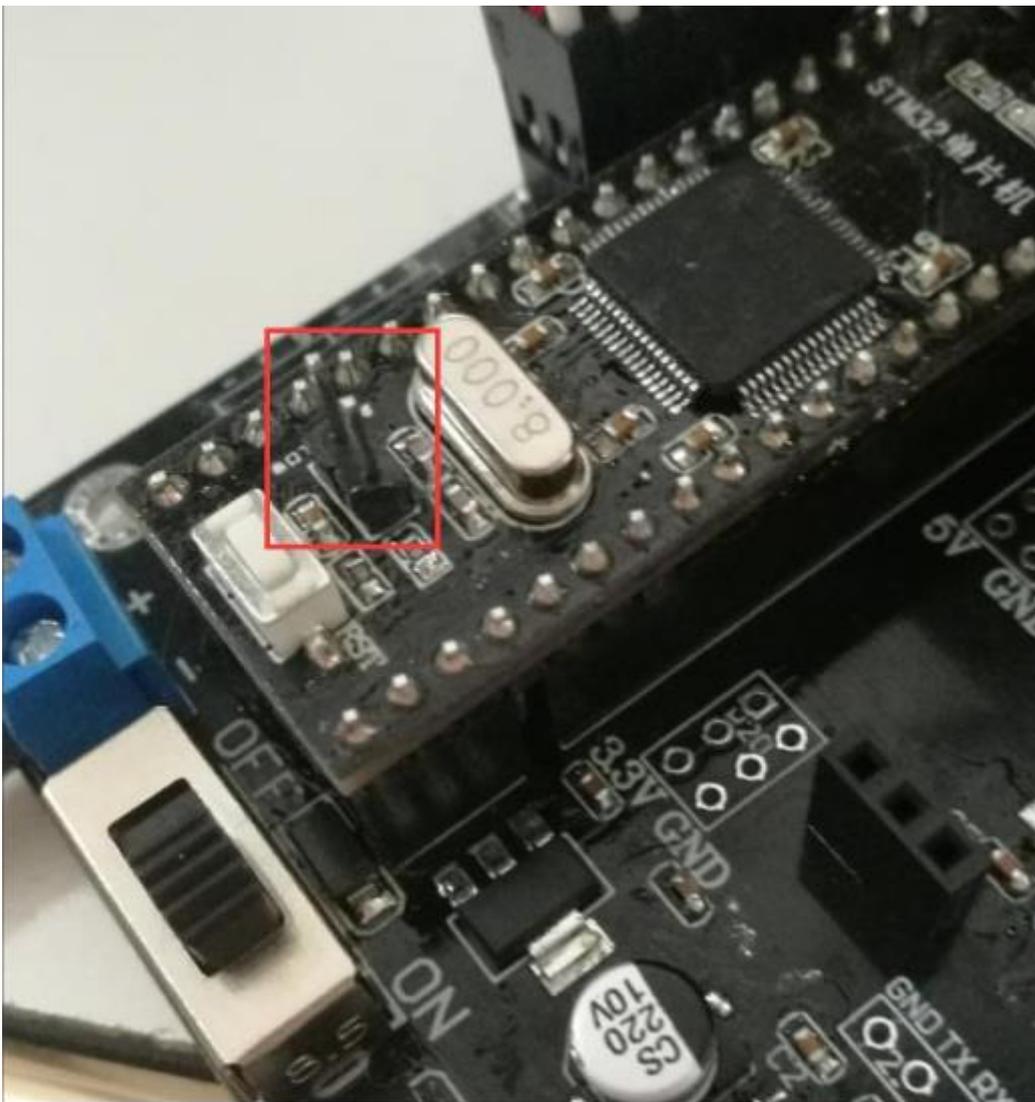


2. Выберите диск и папку для записи.



3. Откройте соответствующий файл.

Снимите заглушку на основной плате stm32. Снятая заглушка обозначает переход с обычного режима в режим записи.



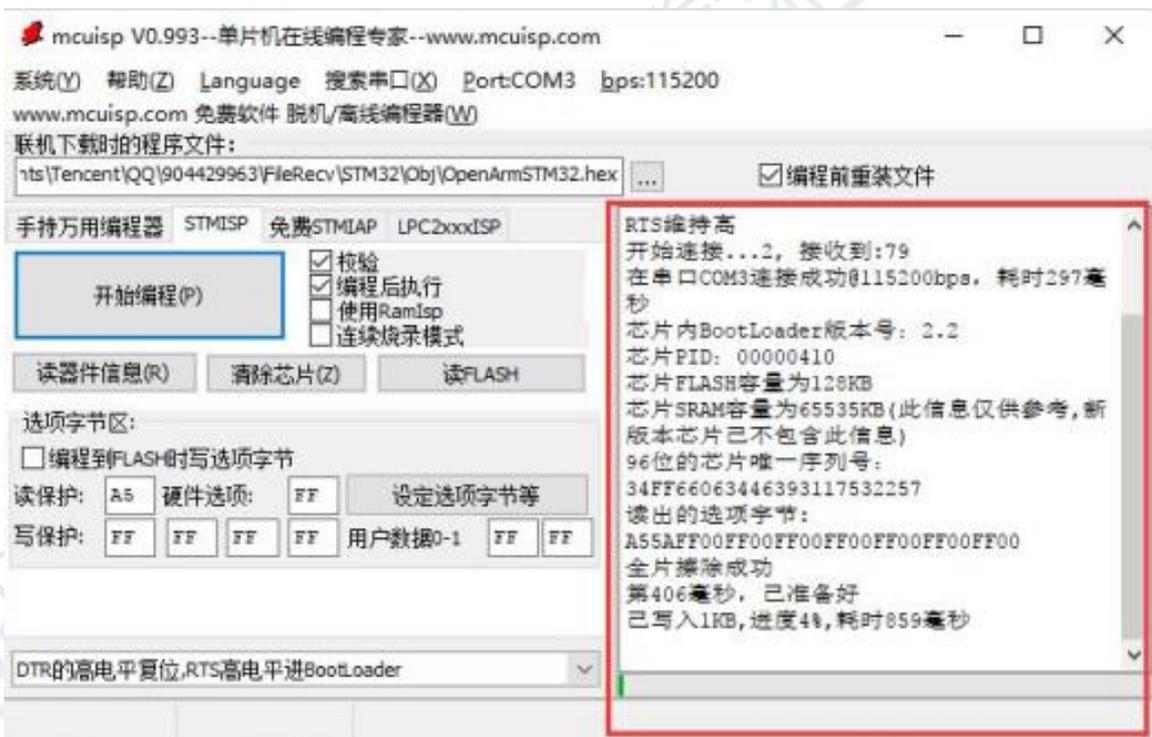
Соедините плату и компьютер с помощью USB кабеля и нажмите «Найти последовательный порт»



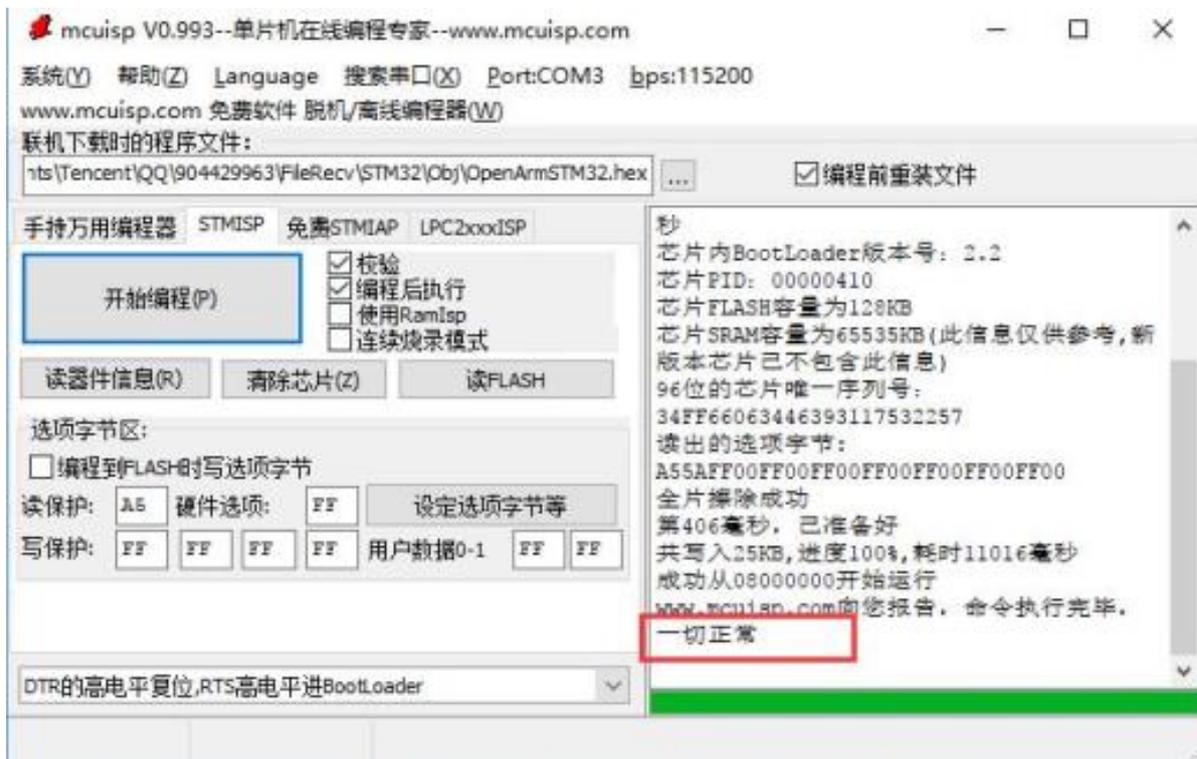
4. Выберите необходимый последовательный порт



5. Нажмите «Начать программировать»



6. После завершения появится надпись «программирование завершено»



Вы можете производить программирование удобным Вам способом.

**Скачать все исходные файлы можно по ссылке:*

https://www.lobot-robot.com/c_detail/15.html

Для скачивания файлов необходимо зарегистрироваться на сайте производителя и войти под своей учётной записью. При скачивании нужно ввести пароль bhп5