

2.4. Лабораторная работа: начинаем работать в Mbed Studio

Site: [Samsung Innovation Campus](#)

Course: Internet of Things

Book: 2.4. Лабораторная работа: начинаем работать в Mbed Studio

Printed by: Антон Файтельсон

Date: Saturday, 21 October 2023, 7:36 PM

Table of contents

2.4.1. Введение

2.4.2. Пример с миганием светодиода

2.4.3. Как правильно делать “мигалку”

2.4.4. Советы и замечания

2.4.1. Введение

Mbed Studio существует для Windows, Mac и Linux, скачивается по адресу: <https://os.mbed.com/studio/>

У этой среды множество возможностей:

- Полноценный компилятор от ARM
- Пошаговая отладка
- Автоопределение подключенной платы
- Загрузка программы прямо из IDE
- Контроль версий
- Окно терминала для коммуникации с платой
- Менеджмент устройств (об этом будет далее)

Но есть и свои недостатки:

- Малая библиотека стандартных примеров по сравнению с онлайн-средой. Но зато и примеры выбраны только самые важные, и в них не запутаешься
- Общая недоработанность среды (версия для Linux вышла совсем недавно), встречаются баги
- Самое главное: медленная компиляция! В онлайн-компиляторе, как ни странно, всё работает гораздо быстрее. Авторы называют в качестве причины то, что он работает на мощной распределенной серверной системе. Поэтому в учебных целях предпочтителен именно онлайн-компилятор для быстрой сборки и изменения примеров. Для серьезной разработки, где вы больше работаете с кодом и вам нужны все вышеупомянутые фишки, больше подойдет как раз офлайновая IDE.

Несмотря на указанные минусы, ознакомиться с Mbed Studio стоит, чтобы знать, чем пользоваться в случае, если нужен дебаггинг или более серьезная работа с кодом.

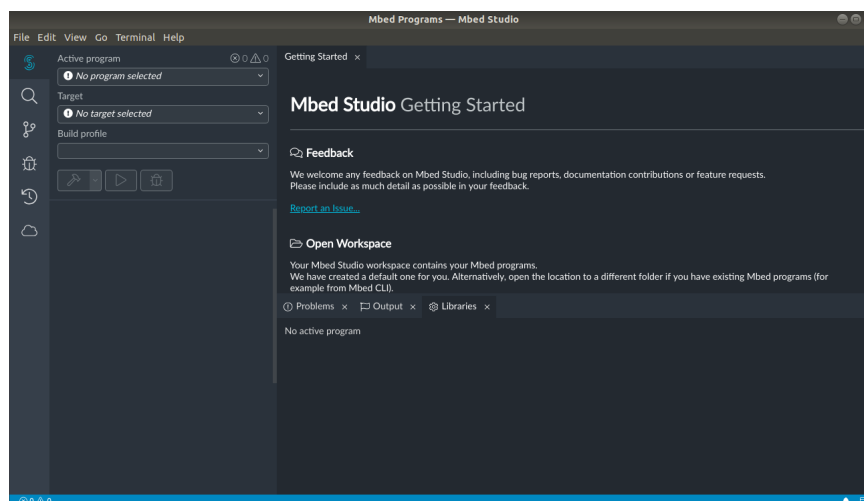
Установка не вызывает трудностей. Если вы работаете в Linux, то установка будет выглядеть так:

```
sudo chmod +x ./MbedStudio-0.9.1.sh
```

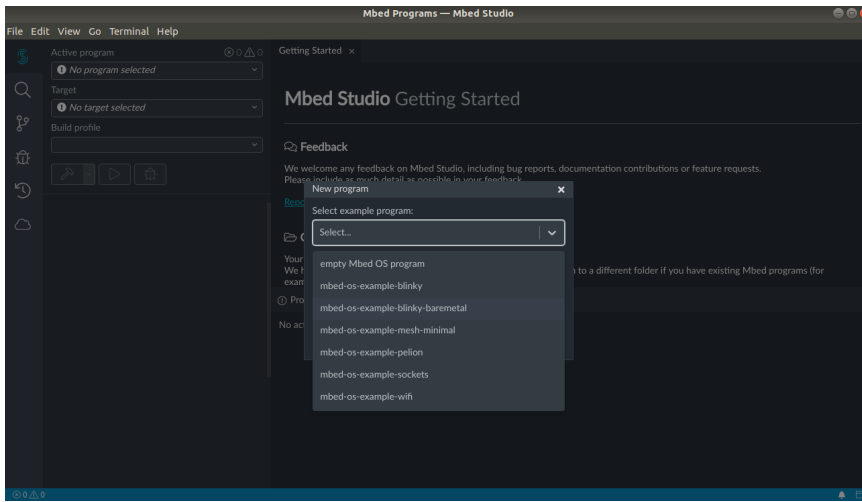
```
./MbedStudio-0.9.1.sh
```

Возможно, в процессе понадобится доустановить недостающие библиотеки.

Программа выглядит так:

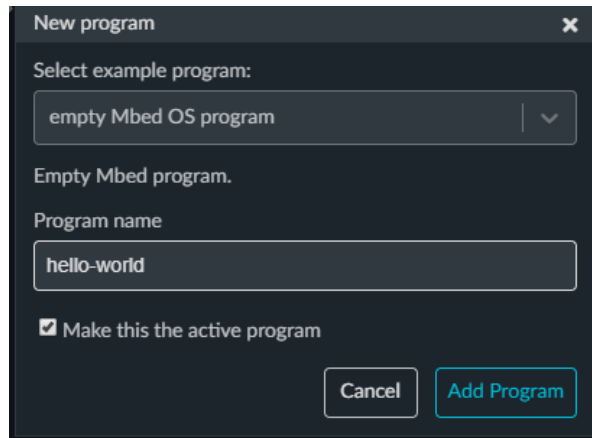


Можно создать пустой проект, либо выбрать один из готовых примеров.



2.4.2. Пример с миганием светодиода

Попробуем сделать новую программу. Выбираем File - New Program. Тип программы выбираем “empty Mbed OS program”, называем программу как угодно.



Вам по умолчанию сделают самый простой пустой main:

```
1  #include "mbed.h"
2
3  // main() runs in its own thread in the OS
4  int main()
5  {
6      while (true) {
7
8      }
9  }
10
11
```

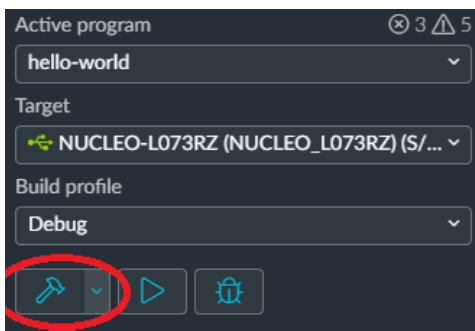
Давайте попробуем здесь простейшую “Мигалку”. Модифицируйте код согласно уже известной вам программе:

```
1  #include "mbed.h"
2  |
3  DigitalOut myled(LED1);
4
5  // main() runs in its own thread in the OS
6  int main()
7  {
8      while (true) {
9          myled = 1; // LED is ON
10         wait(0.5); // 200 ms
11         myled = 0; // LED is OFF
12         wait(0.5); // 1 sec
13     }
14 }
```

Функция wait выделяется желтым как предупреждение. Она по-прежнему работает, но ее не рекомендуется использовать. В дальнейшем будет сказано, как делать задержку корректнее, а пока можно собрать программу и с ней.

Сборка и загрузка программы не вызывает трудностей. Выберите в Target конкретную модель вашей платы. Дальше назначение кнопок

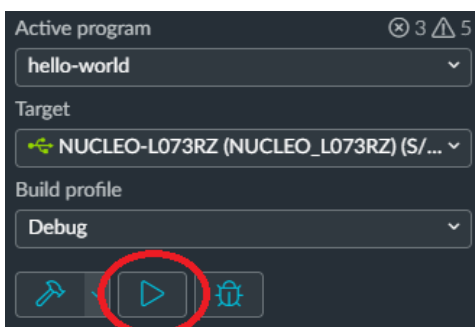
понятно: молоток - компиляция, стрелочка - загрузка программы в плату, жучок - отладка. Соберите вашу первую программу нажатием на “молоток”.



Пусть вас не смущает долгое время компиляции: происходит сборка всех вспомогательных библиотек, при дальнейших сборках этого же проекта всё будет гораздо быстрее. По итогам сборки увидите внизу лог:

```
Problems x Output x Libraries x
-----
Compile [ 99.5%]: sleep.c
Compile [ 99.7%]: stm_spi_api.c
Compile [ 99.8%]: trng_api.c
Compile [ 99.9%]: watchdog_api.c
Compile [100.0%]: us_ticker.c
Link: hello-world
[Warning] @0,0: L3912W: Option 'legacyalign' is deprecated.
Elf2Bin: hello-world
| Module | .text | .data | .bss |
|-----|-----|-----|-----|
| [Lib]\c_p.l | 11779(+11779) | 16(+16) | 348(+348) |
| [Lib]\fz_ps.l | 32(+32) | 0(+0) | 0(+0) |
| [Lib]\libcppabi_p.l | 44(+44) | 0(+0) | 0(+0) |
| [Lib]\m_ps.l | 44(+44) | 0(+0) | 0(+0) |
| anon$obj.o | 32(+32) | 0(+0) | 1024(+1024) |
| main.o | 136(+136) | 0(+0) | 28(+28) |
| mbed-os\drivers | 437(+437) | 0(+0) | 0(+0) |
| mbed-os\hal | 1920(+1920) | 8(+8) | 130(+130) |
| mbed-os\platform | 6564(+6564) | 64(+64) | 368(+368) |
| mbed-os\rtos | 9920(+9920) | 168(+168) | 7009(+7009) |
| mbed-os\targets | 9238(+9238) | 4(+4) | 767(+767) |
| Subtotals | 40146(+40146) | 260(+260) | 9674(+9674) |
Total Static RAM memory (data + bss): 9934(+9934) bytes
Total Flash memory (text + data): 40406(+40406) bytes
Image: BUILD/NUCLEO_L073RZ/ARMv6\hello-world.bin
```

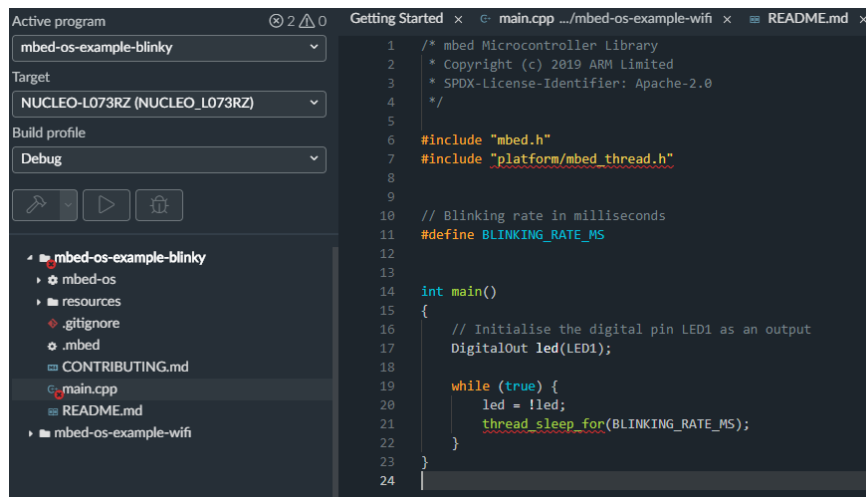
Теперь ничто не мешает загрузить программу в плату нажатием на “треугольник”.



Увидите, что плата начнет мигать встроенным светодиодом раз в секунду!

2.4.3. Как правильно делать “мигалку”

Теперь, раз вы уже знаете, как работать с потоками, запустите “правильную” мигалку в правильном стиле. Рассмотрите пример под названием mbed-os-example-blinky

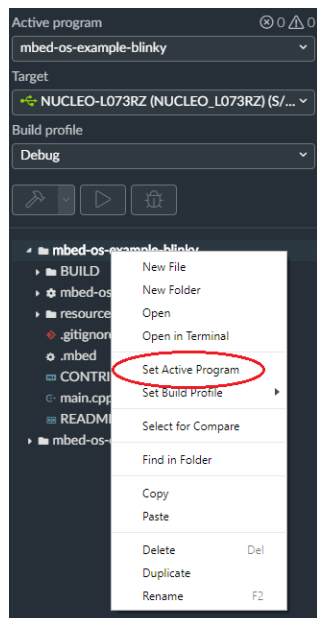


```
1  /* mbed Microcontroller Library
2  * Copyright (c) 2019 ARM Limited
3  * SPDX-License-Identifier: Apache-2.0
4  */
5
6  #include "mbed.h"
7  #include "platform/mbed_thread.h"
8
9
10 // Blinking rate in milliseconds
11 #define BLINKING_RATE_MS 500
12
13
14 int main()
15 {
16     // Initialise the digital pin LED1 as an output
17     DigitalOut led(LED1);
18
19     while (true) {
20         led = !led;
21         thread_sleep_for(BLINKING_RATE_MS);
22     }
23 }
24
```

Код достаточно понятен: основной поток программы засыпает на 500 миллисекунд, после чего переключает светодиод

2.4.4. Советы и замечания

1. Если работаете с несколькими проектами, то не забывайте указывать, какой из них активен в настоящий момент, нажатием правой кнопки мыши на проекте и выбором Set Active Program



2. Если среда не предлагает выбор цели и не дает скомпилировать программу, то скорее всего, она занята обновлением библиотек mbed. Среда так устроена, что она всегда стремится держать библиотеки в актуальном состоянии, и пока они не подгрузятся, ничего работать в принципе не будет.
3. На настоящий момент замечен баг - иногда не открывается терминал последовательного порта. Он должен автоматически открываться, но этого не происходит в некоторых случаях. Решение здесь простое - использовать внешний терминал при необходимости.
4. Проблема, похоже, связанная с п.2. - иногда не появляется кнопки "загрузка" - среда не распознает подключенную плату. В этом случае у вас всегда остается вариант загрузки программы через файловый менеджер, бросанием ее на подключенную "флэшку". Ваши программы хранятся в папке Mbed Programs в домашней директории, и там всегда можно взять файл .bin.

[Reset user tour on this page](#)