

ユーザー (/activity/) » hudakz (/users/hudakz/) » コード (/users/hudakz/code/) »
STM32F103C8T6_Hello (/users/hudakz/code/STM32F103C8T6_Hello/)



ゾルタン・フダック (/users/hudakz/) / OS 2
(/users/hudakz/code/STM32F103C8T6_Hello/)

STM32F103C8T6 Hello

(/users/hudakz/code/STM32F103C8T6_Hello/) ✓ 特徴

mbedで低コストのBlue Pill (STM32F103C8T6) ボードを使用します。

🔗 依存関係 : (/users/hudakz/code/STM32F103C8T6_Hello/dependencies) ⚙️ mbed (/users/mbed_official/code/mbed/)

👤 扶養家族 : (/users/hudakz/code/STM32F103C8T6_Hello/dependents) 📦 STM32F103C8T6 Chicharronera 4CV2
(/teams/Que-elegancia-la-de-Francia/code/STM32F103C8T6_Chicharronera_4CV2/) 📦 STM32F103C8T6 Portada
(/teams/Que-elegancia-la-de-Francia/code/STM32F103C8T6_Portada/)

ホーム (/users/hudakz/code/STM32F103C8T6_Hello/)	歴史 (/users/hudakz/code/STM32F103C8T6_Hello/shortlog)
グラフ (/users/hudakz/code/STM32F103C8T6_Hello/graph)	
APIドキュメント (/users/hudakz/code/STM32F103C8T6_Hello/docs/)	
Wiki (/users/hudakz/code/STM32F103C8T6_Hello/wiki/)	
プルリクエスト (/users/hudakz/code/STM32F103C8T6_Hello/pull-requests/)	

STM32F103C8T6ボード、エイリアスブルーピル

手頃な価格 (eBayで約2ドル) で、ユーザーが新しいアイデアを試してプロトタイプを作成するための柔軟な方法を提供します。このボードには、NUCLEO-F103RB (<https://developer.mbed.org/platforms/ST-Nucleo-F103RB/>) プラットフォームと互換性のあるSTM32F103C8T6 (http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f1-series/stm32f103/stm32f103c8.html) マイクロコントローラーが搭載されています。
(<https://developer.mbed.org/platforms/ST-Nucleo-F103RB/>)

マイクロコントローラー機能

- LQFP48パッケージのSTM32F103C8T6
- ARM®32ビットCortex®-M3CPU
- 最大72 MHzのCPU周波数
- 2.0 Vから3.6 VのVDD

リポジトリツールボックス

[コンパイラーにインポート \(https://os.mbed.com/ide/import-program/use\)](https://os.mbed.com/ide/import-program/use)

📁 デスクトップIDEにエクスポート (/users/hudakz/code/ST

🔗 リポジトリを構築 (/users/hudakz/code/STM32F10

+ フォロー

🔗 埋め込みURL : <<program /users/h

📄 リポジトリをデスクトップに複製 : hg clone

リポジトリの詳細

📄 タイ プログラム

プ :

版 : Mbed OS 2

作成 2016年7月2日 (2016年7月2日)

し
た :

輸 📄 2668 (/users/hudakz/code/STM32F103C8T6_Hello/)

入 :

フォ 📄 7 (/users/hudakz/code/STM32F103C8T6_Hello/forks/)

ー
ク :

コミ 📄 14 (shortlog)

ッ
ト :

- 20 KB SRAM
- 外部割り込み機能を備えたGPIO (32)
- 10チャンネルの12ビットADC (2)
- RTC
- タイマー (4)
- I2C (2)
- USART (3)
- SPI (2)
- USB 2.0フルスピード
- できる

家族:

依存関係:

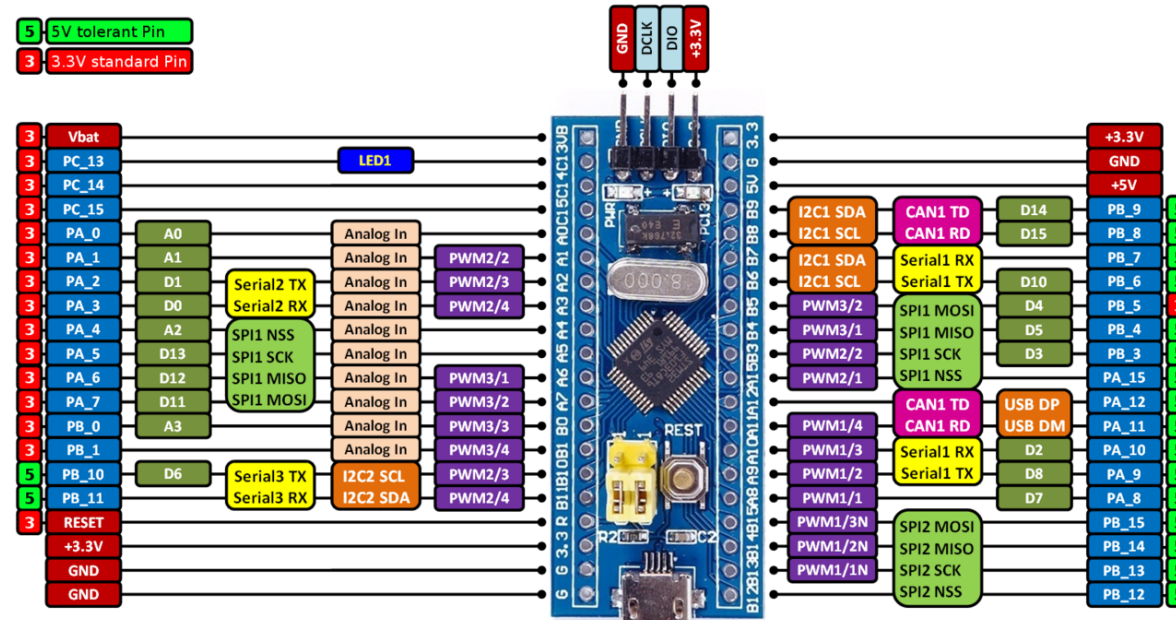
フォロワー:

ボードの機能

- 小さな足跡
- 柔軟なボード電源：USB VBUSまたは外部ソース (3.3V、5V)
- ユーザーLED：LED1
- ワンブッシュボタン：リセット
- プログラミング/デバッグポート
- Micro-B USBコネクタ

ボードのピン配列

(/media/uploads/hudakz/stm32f103c8t6_pinout_voltage01.png)



ピンの横にある (/media/uploads/hudakz/stm32f103c8t6_pinout_voltage01.png)最大許容I/O電圧レベルのズームインは、



(./media/uploads/hudakz/stm32f103c8t6_pinout_voltage01.png)ボードの損傷を防ぐためにThor Stenの厚意
(https://developer.mbed.org/users/Thorsten/)により提供されています。ピン定義の詳細については、データシートの
(http://www.st.com/content/ccc/resource/technical/document/datasheet/33/d4/6f/1d/df/0b/4c/6d/CD00161566.pdf/files/CD00161566.pdf/jcr:content/translations/en.CD00161566.pdf)
表5を参照してください。



i 情報

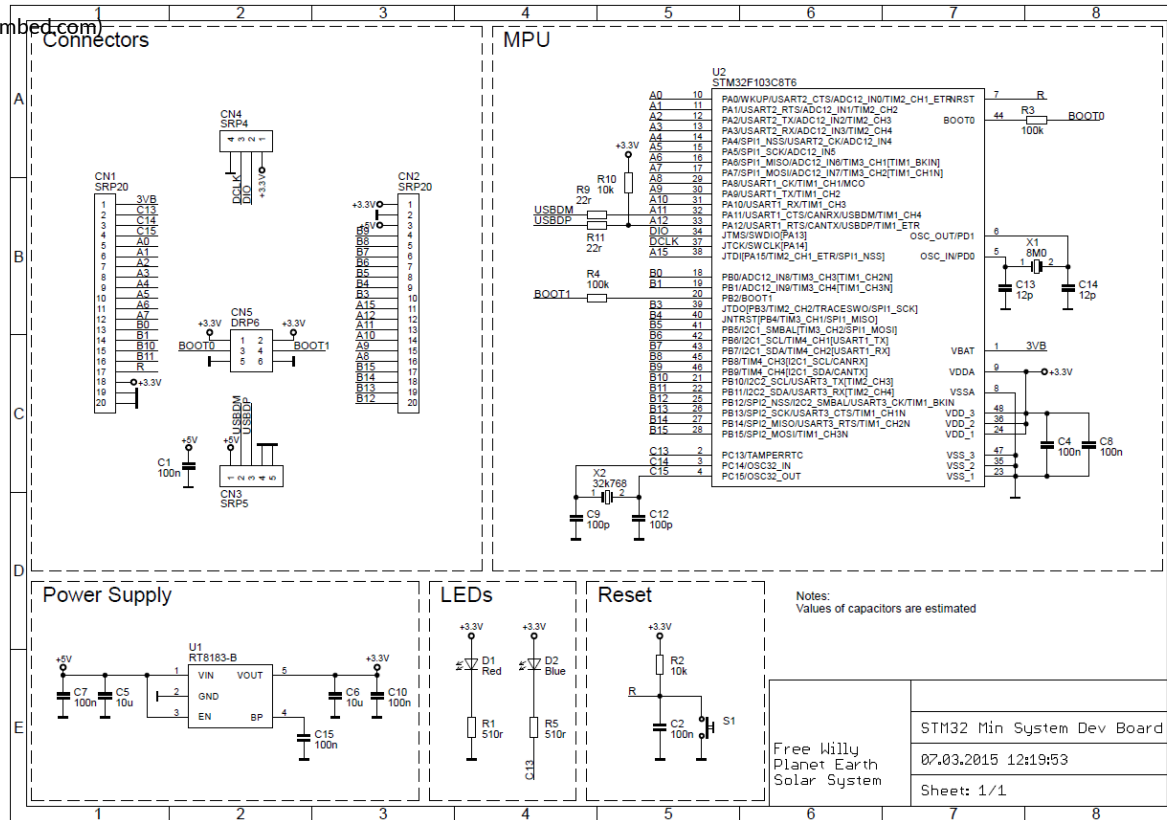
青/白または緑/白で印刷されたラベル（PC_13、PB_9、A0、D14など）のみをコードで使用する必要があります。他のラベルは情報として提供されます（代替機能、電源ピン、...）。これらの追加のピン名も使用できます。

i

```
1 LED1 = PC_13 SERIAL_TX = PA_2 I2C_SCL = PB_8 SPI_MOSI = PA_7 PWM_OUT = PB_3
2 SERIAL_RX = PA_3 I2C_SDA = PB_9 SPI_MISO = PA_6
3 SPI_SCK = PA_5
4 SPI_CS = PB_6
```

小さいサイズのボードに合わせるために、先頭の「P」と「_」の文字がボードに表示されているラベルから省略されていることに注意してください（たとえば、「PA_1」の代わりにボードに「A1」というラベルがあります）。
Arduino（緑/白）および追加の命名ラベルは、ボードに示されていません。
また、オンボードLEDはピンPC_13に接続されており、抵抗を介して+ 3.3Vに接続されています。そのため、LEDをオンにするには、☐ on または ☐ off DigitalOutを ☐ 0 またはに設定する必要があり ☐ 1 ます。

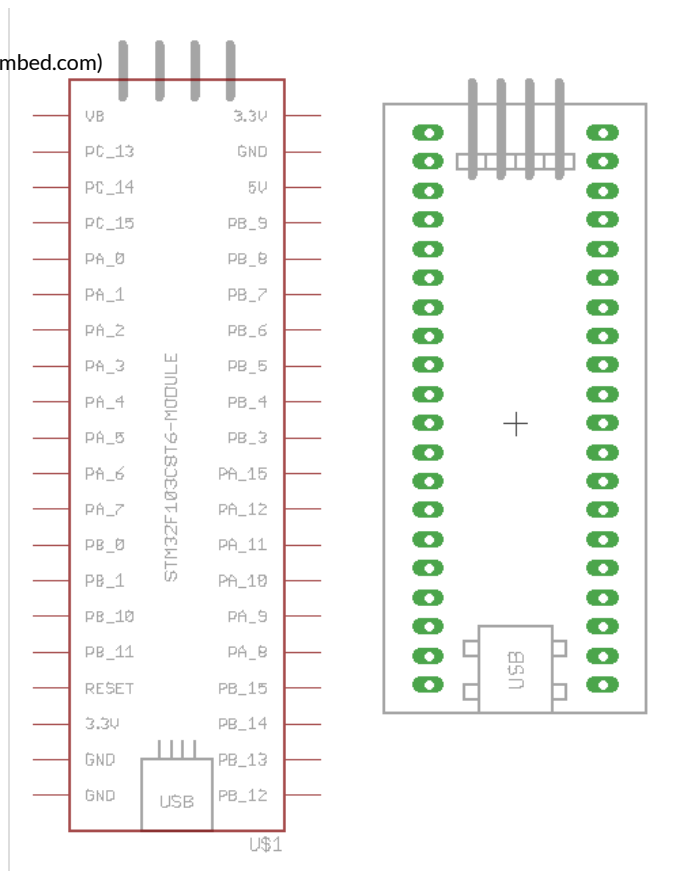
回路図



イーグル図書館

回路図

ボード



どちらのファイル [lbr](#) も [zip](#) mbed wikiページにアップロードしてイーグルライブラリ
([/media/uploads/hudakz/stm32f103c8t6_module.png](#))をダウンロードできないため、リンクを右クリックしてを選択しま
す [Save link as...](#)。ローカルドライブに保存したら、拡張子をから [png](#) に変更し [lbr](#) ます。

mbedオンラインコンパイラを使用して STM32F103C8T6ボード用のプログラムを構築す る

NUCLEO-F103RBボード用のプログラムを作成します（オンラインコンパイラのターゲットプラットフォームとして
NUCLEO-F103RBを選択します）。
または、[ここをクリックし](https://developer.mbed.org/compiler/#import/users/hudakz/code/STM32F103C8T6_Hello/)
て、このデモをオンラインコンパイラにインポートします。

オンボードLEDの点滅：



arm

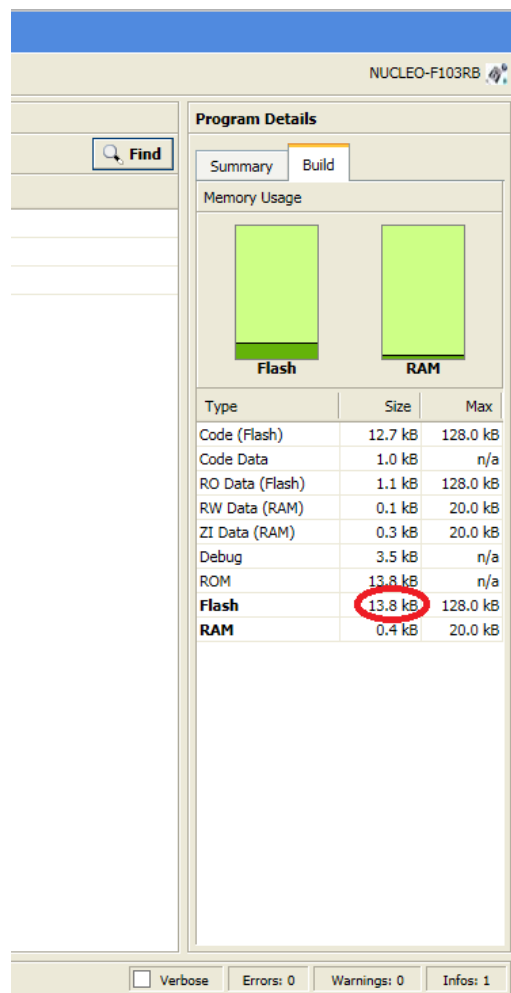
MBED (https://www.mbed.com)

```
1 #include "mbed.h"
2
3 // デジタル I/O
4 DigitalOut myled (PC_13); // オンボード LED
5
6 int main ()
7 {
8     while (1) {
9         // オンボード LED は、抵抗を介して + 3.3V (GND ではなく) に接続されます。
10        // したがって、LED は 0
11        myled = 0 でアクティブになります。 //
12        wait_ms (200) の LED をオンにします; // 200 ミリ秒
13        myled = 1 待機します; //
14        PC の LED をオフにします。 printf ("Blink \ r \ n");
15        wait_ms (1000); // 1000 ミリ秒待機します
16    }
17 }
```



警告

オンラインコンパイラは、128kB の最大フラッシュサイズをチェックしていることに注意してください。ただし、STM32F103C8T6 には 64kB しか装備されていません。オンラインで販売されている Blue Pill ボードの大半は、通常 64kB ではなく 128kB のフラッシュを搭載しているようです。コンパイルが完了すると (コンパイルドロップリストの [ビルドのみ] ボタンをクリックするか、**Ctrl + B** を押すことで開始)、[プログラムの詳細 - ビルド] タブで使用済みフラッシュメモリのサイズを視覚的に確認できます。STM32F103C8T6 ボードに適合するために、使用されるフラッシュは 64kB を超えてはなりません (実際のボードによって異なります)。64kB 未満のフラッシュメモリを使用するまでプログラムを最適化してください。見てください MBED-STM32F030F4 (<https://developer.mbed.org/users/mega64/code/mbed-STM32F030F4/>) と アンディ のヒント (<https://developer.mbed.org/questions/72252/mbed-dev-Minimum-application-size-using-/>) いくつかの良いヒントのために。



STM32F103C8T6ボードのプログラミング

NUCLEO ST-LINK / V2-1およびドラッグアンドドロップ

NUCLEO仮想ディスクを使用して、STM32F103C8T6ボードをプログラミングできます（ドラッグアンドドロッププログラミング）。そのためには、追加のNUCLEOボードが必要です（ST-LINK / V2-1を装備しているどのタイプでも可能です）。

- 図8に示すように、CN2コネクタから2つのジャンパーを取り外します。

Figure 8. Using ST-LINK/V2-1 to program the STM32 on an external application

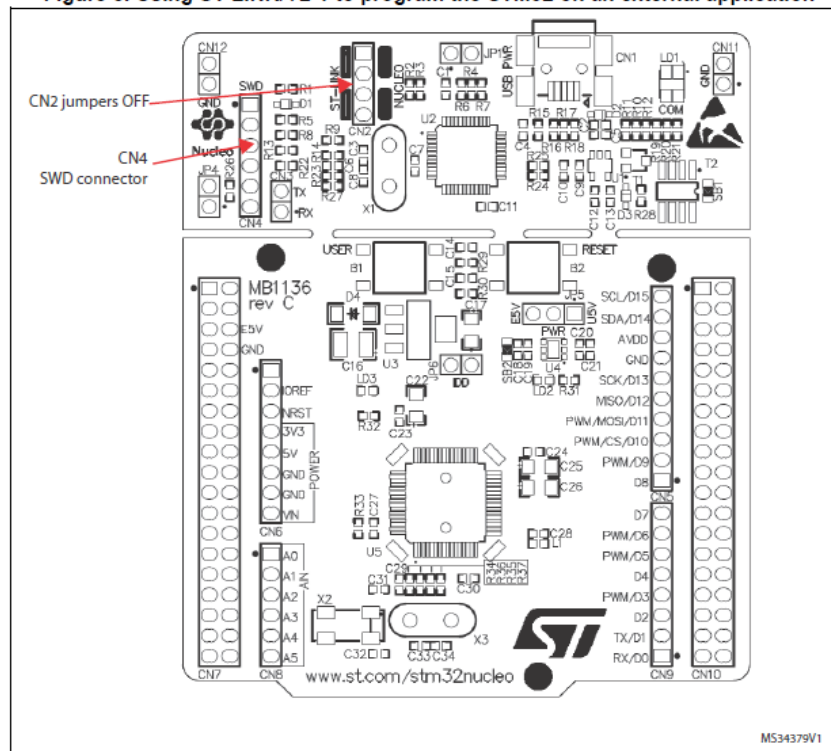
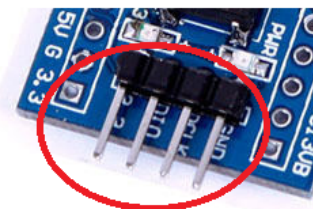


Table 4. Debug connector CN4 (SWD)

Pin	CN4	Designation
1	VDD_TARGET	VDD from application
2	SWCLK	SWD clock
3	GND	Ground
4	SWDIO	SWD data input/output
5	NRST	RESET of target MCU
6	SWO	Reserved

STM32F103C8T6 debug connector



3V3, DIO, DCLK, GND

- 次のようにフラッシングワイヤを使用して、NUCLEOボードCN4コネクタをSTM32F103C8T6ボードに接続します。

NUCLEOボード CN4コネクタ		STM32F103C8T6 デバッグコネクタ		STM32F103C8T6 ボード
SWCLK	<=>	DCLK		
GND	<=>	GND		
SWDIO	<=>	DIO		
NRST			<=>	リセット

- 3.3Vピン、5Vピン、またはUSBケーブルを介してSTM32F103C8T6ボードに電力を供給します。（NUCLEOボードCON4のVDD_TARGETピンは、電源として機能しません）。
- NUCLEOボードをUSBケーブルでPCに接続します。
- STM32F103C8T6ボードをプログラムするには、[**コンパイル**] ボタンをクリックして、バイナリをNUCLEO仮想ディスクに保存します。
詳細については、[ユーザーマニュアルの](#)
(http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf)
6.2.4章「*ST-LINK/V2-1*を使用して外部STM32アプリケーションをプログラムおよびデバッグする」
(http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf)
を参照してください。

ST-Link V2 USB ドングルおよびSTM32 ST-LINKユーティリティ

ボードをプログラムするためにST-Link V2 USB ドングル（別名ST-Link V2プログラミングユニット）を使用する場合は、上記と同じ配線を適用します。まだ行われていない場合は、PCにST-Link/V2ドライバー
(<https://os.mbed.com/teams/ST/wiki/ST-Link-Driver>)をインストールします。ST-Link V2ドングルをPCに差し込みます。次に、**Compile** ボタンをクリックして、バイナリをローカルディスクに保存します。**STM32 ST-LINKユーティリティ**
(<http://www.st.com/en/development-tools/stsw-link004.html>)をインストールして実行します
(<http://www.st.com/en/development-tools/stsw-link004.html>)。プログラムが実行されたら、オンラインコンパイラでビルド



STM32 USARTシステムメモリブートローダーおよびフラッシャーSTM32

見ていSTM32組み込みブートローダを (<https://scienceprog.com/flashing:programs-to-stm32-embedded-bootloader/>)。

ダウンロード : FLASHER-STM32 (<http://www.st.com/en/development-tools/flasher-stm32.html>)

詳細については、Application Note AN2606を

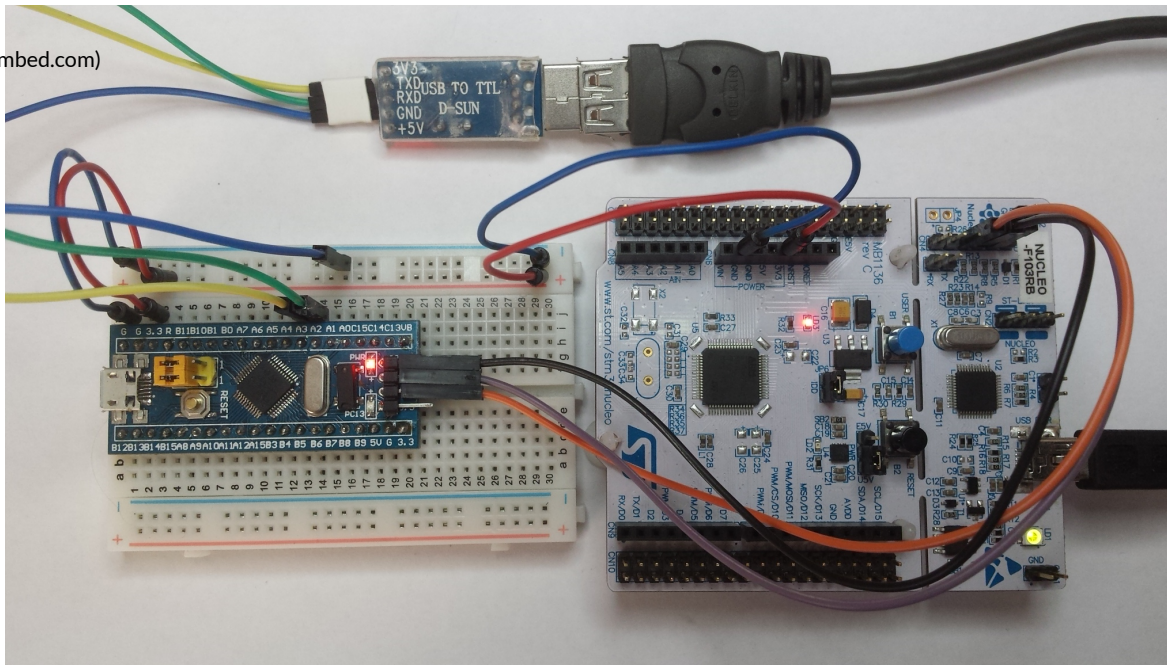
(http://www.st.com/content/ccc/resource/technical/document/application_note/b9/9b/16/3a/12/1e/40/0c/CD00167594.pdf/files/CD00167594.pdf/jcr:content/translations/en.CD00167594.pdf)

参照してください

(http://www.st.com/content/ccc/resource/technical/document/application_note/b9/9b/16/3a/12/1e/40/0c/CD00167594.pdf/files/CD00167594.pdf/jcr:content/translations/en.CD00167594.pdf)。

シリアルポートの使用（デバッグだけでなく）

- Connect an FTDI or similar USB to Serial TTL converter to your PC and to an on-board serial port (for example PA_2, PA_3). Make sure you connect the on-board TX pin to the converter's RX pin and the on-board RX pin to the converter's TX pin.



Zoom in (/media/uploads/hudakz/stm32f103c8t6_hookup.jpg).

- In your code, create a `Serial` object (using TX and RX pin names of the connected serial port).
- Use `printf` function to send serial messages to the connected PC.

i Sending debug messages over the ST-Link virtual com port

In case you would like to spare the external USB-Serial converter for other purposes then there is available an alternative solution proposed by [X M \(bitman\)](https://developer.mbed.org/users/bitman/). You can use the ST-Link virtual com port also for debugging of programs running on the STM32F103C8T6 board. However, that will require a soldering iron (and probably some soldering skills). According to the [User Manual](http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/f) (http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/f) chapter 6.8 "USART communication", solder bridges (on the back side of the NUCLEO board) SB62 and SB63 should be ON, SB13 and SB14 should be OFF. In such case it is possible to connect another USART to the NUCLEO (ST-Link) CN3 connector using flying wires. For instance on STM32F103C8T6 board it is possible to use USART2 available on PA_2 (TX) and PA_3 (RX). Two flying wires shall be connected as follows:

STM32F103C8T6 board, pin PA_2 (Serial2 TX) <=> NUCLEO board CN3 connector, pin RX

STM32F103C8T6 board, pin PA_3 (Serial2 RX) <=> NUCLEO board CN3 connector, pin TX

A smart trick proposed by [Nothing Special](https://developer.mbed.org/users/mega64/) makes even soldering needless.

The point is to redirect the UART on the NUCLEO board by software (without modifying the solder bridges on the back side of the NUCLEO board) and convert it into a "Debugger". On the NUCLEO board that you are going to use as programmer/



debugger, choose any Serial port other than Serial2 (other than the default port used for standard UART) to be initialized as standard UART. In the program below (using NUCLEO-F103RB as programmer/debugger) Serial1 (PA_9, PA_10) was selected.

Debugger

```
1 #include "mbed.h"
2
3 // declarations needed to change the parameters of stdio UART
4 extern serial_t      stdio_uart;
5 extern int           stdio_uart_initd;
6
7 int main() {
8     serial_init(&stdio_uart, PA_9, PA_10); // other than Serial2
9     stdio_uart_initd = 1;
10    printf("Ready for debugging\r\n");
11 }
```

Once compiled (remember to select the NUCLEO board used for programing/debugging as target for the online compiler), download the "Debugger" program to the NUCLEO board. Please make sure you have the two jumpers in place on the CN2 connector when programming the NUCLEO board. Once the "Debugger" binary has been downloaded to the NUCLEO board, remove the two jumpers again.



Additional examples

- [Reading analog input \(https://os.mbed.com/users/hudakz/code/STM32F103C8T6_AnalogIn/\)](https://os.mbed.com/users/hudakz/code/STM32F103C8T6_AnalogIn/)
- [DS18B20 temperature sensor \(https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_DS18B20/\)](https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_DS18B20/)
- [Driving Text LCD \(https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_TextLCD/\)](https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_TextLCD/)
- [Driving Nokia 5110 LCD \(https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_Nokia5110/\)](https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_Nokia5110/)
- [Driving ILI9341 TFT Display \(https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_ILI9341/\)](https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_ILI9341/)
- [A USB serial device \(https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_USBSerial/\)](https://developer.mbed.org/users/hudakz/code/STM32F103C8T6_USBSerial/)
- [CAN bus with mbed boards \(https://os.mbed.com/users/hudakz/code/CAN_Hello/\)](https://os.mbed.com/users/hudakz/code/CAN_Hello/)
- [Tiny HTTP server with ENC28J60 Ethernet module \(https://os.mbed.com/users/hudakz/code/WebSwitch_ENC28J60/\)](https://os.mbed.com/users/hudakz/code/WebSwitch_ENC28J60/)
- [Using SD Card \(https://os.mbed.com/users/hudakz/code/STM32F103C8T6_SDCard/\)](https://os.mbed.com/users/hudakz/code/STM32F103C8T6_SDCard/)

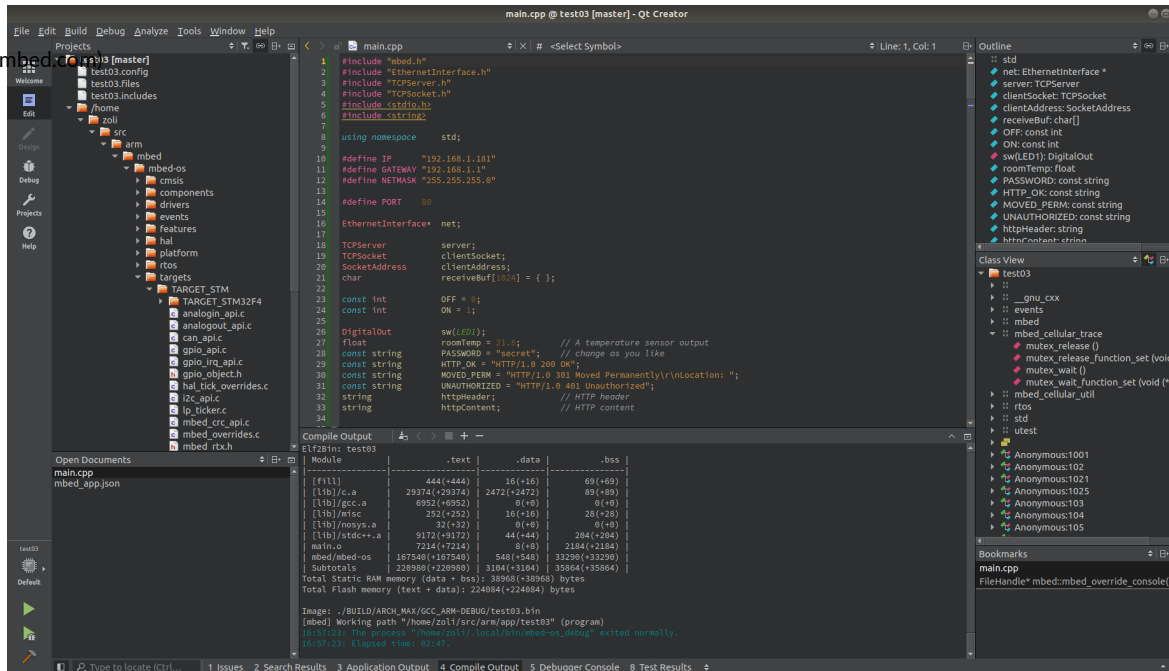
Working offline

The [Mbed Studio \(https://os.mbed.com/blog/entry/Mbed-Studio-Public-Beta/\)](https://os.mbed.com/blog/entry/Mbed-Studio-Public-Beta/) IDE is a great tool but unfortunately supports only Mbed OS 5 (with built-in Mbed RTOS threads, mutexes, semaphores ...). That however requires decent SRAM and Flash memory. Since the [Blue Pill](#) is rather suitable for small or medium size projects which require less RAM and Flash I'd recommend to stay with **Mbed 2** (aka Mbed classic) and use the [QtCreator IDE \(https://os.mbed.com/users/hudakz/notebook/building-offline-with-qt-creator/\)](#) for building offline.



arm

MBED(https://www.mbed.com/)



Works great on Linux (Ubuntu) and MS Windows. It's fully integrated with [Mbed CLI](https://os.mbed.com/docs/mbed-os/v5.11/tools/developing-mbed-cli.html) and supports all the Mbed enabled targets. To build for the Blue Pill you can select `BLUEPILL_F103C8` or `NUCLEO_F103RB` as your target, depending on the Flash size actually available on your Blue Pill board. To reduce the size of used static RAM and Flash try to select the `small` library for your project rather than the `standard` one. That can be achieved by modifying the `mbed_app.json` file (Ctrl+O) as below:

```
1 {
2     "target_overrides": {
3         "**": {
4             "target.default_lib": "small"
5         }
6     }
7 }
```

詳細については、[Qt Creator IDEを使用したオフラインビルドを](https://os.mbed.com/users/hudakz/notebook/building-offline-with-qt-creator/)ご覧ください。

ハッピーコーディング:-)

リポジトリのダウンロード：[zip \(/users/hudakz/code/STM32F103C8T6_Hello/archive/485613003845.zip\)](https://users.hudakz/code/STM32F103C8T6_Hello/archive/485613003845.zip) [tar.gz \(/users/hudakz/code/STM32F103C8T6_Hello/archive/485613003845.tar.gz\)](https://users.hudakz/code/STM32F103C8T6_Hello/archive/485613003845.tar.gz)

リビジョン13：485613003845のファイル

/ デフォルトの ヒント

名前	サイズ	行動
↑ アップ (/users/hudakz/code/STM32F103C8T6 Hello/file/485613003845/)		
main.cpp	493	🔄 改訂 (/users/hudakz/code/STM32F103C8T6 Hello/log/485613003845/main.cpp) 🖋 注釈を付ける (/users/hudakz/code/STM32F103C8T6 Hello/annotate/485613003845/main.cpp)
mbed.bld	69	🔄 改訂 (/users/hudakz/code/STM32F103C8T6 Hello/log/485613003845/mbed.bld) 🖋 注釈を付ける (/users/hudakz/code/STM32F103C8T6 Hello/annotate/485613003845/mbed.bld)