



STマイクロエレクトロニクス STM32ファミリ開発環境のご紹介

1. STM32開発環境の紹介

- STM32の主な開発環境
- STM32Cube
- Nucleo
- X-Nucleo

2. Nucleoを使用した実習

- ツールチェーン&ライブラリの準備
- 実習1: ツールの動作確認
- 実習2: LED点滅

3. まとめ

- 開発ツールの使い分け

STM32の主な開発環境(1/2)

3

統合開発環境

- KEIL社 MDK-ARM



- IAR社 EWARM



- Atollic社 TrueSTUDIO



- ST社 System Workbench



デバgger

- ST社 ST-LINK/V2



- KEIL社 U-LINK



- IAR社 I-Jet



- Raisonance社 R-LINK



評価ボード

- ST社 標準評価ボード



- ST社 簡易ボード「Discovery」



- ST社 mbed対応ボード「Nucleo」



OS

- eForce社 uC3 (μITRON)



- ユーシーテクノロジー社 μT-kernel



STM32の主な開発環境(2/2)

4

ソフトウェア・ライブラリ

- 標準ペリフェラル
- USBデバイス／ホスト
- イーサネット
- DSP
- GUI
- オーディオ
- モーター
- 暗号化



その他の便利なツール

- STVP (FLASH書き込みソフト)
- Clock Configuration Tool (クロック設定ツール)
- MicroXplorer (MUXピン設定ツール)
- STM Studio (変数監視視覚化ツール)
- Java Evaluation Kit (Java環境構築キット)
- Matlab/Simulink Embedded Target
(Matlab/Simulink対応STM32用ブロックセット)

STM32Cube

Complete Embedded Software & Tools



STM32Cube

STM32Cubeの紹介(1/2)

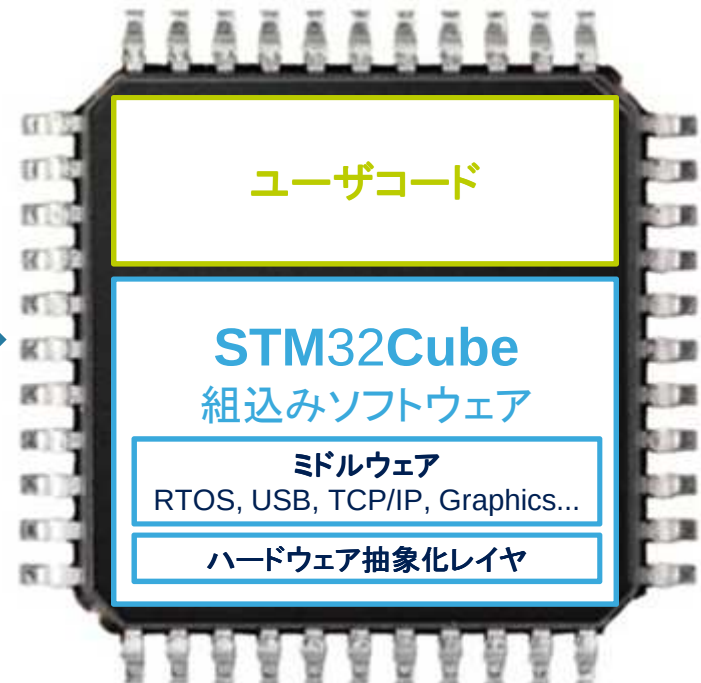
6

- STM32Cube™は100% 無償のソリューションで以下のものを含まれます:
 - ソフトウェア設定ツール(PC上で動作)
 - 様々なSTM32 用組み込みソフトブロック

STM32CubeMX
PC上の設定ツール

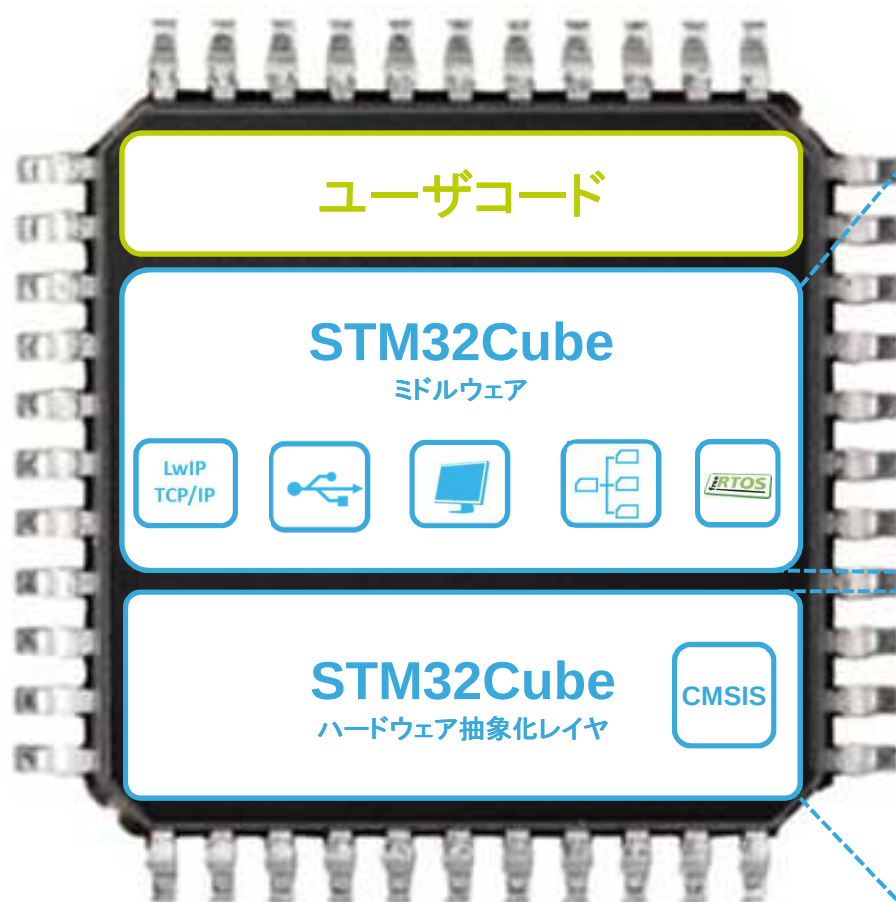


ユーザの設定に応じた
Cコードの初期化ファイルを
自動生成



STM32Cubeの紹介 (2/2)

7



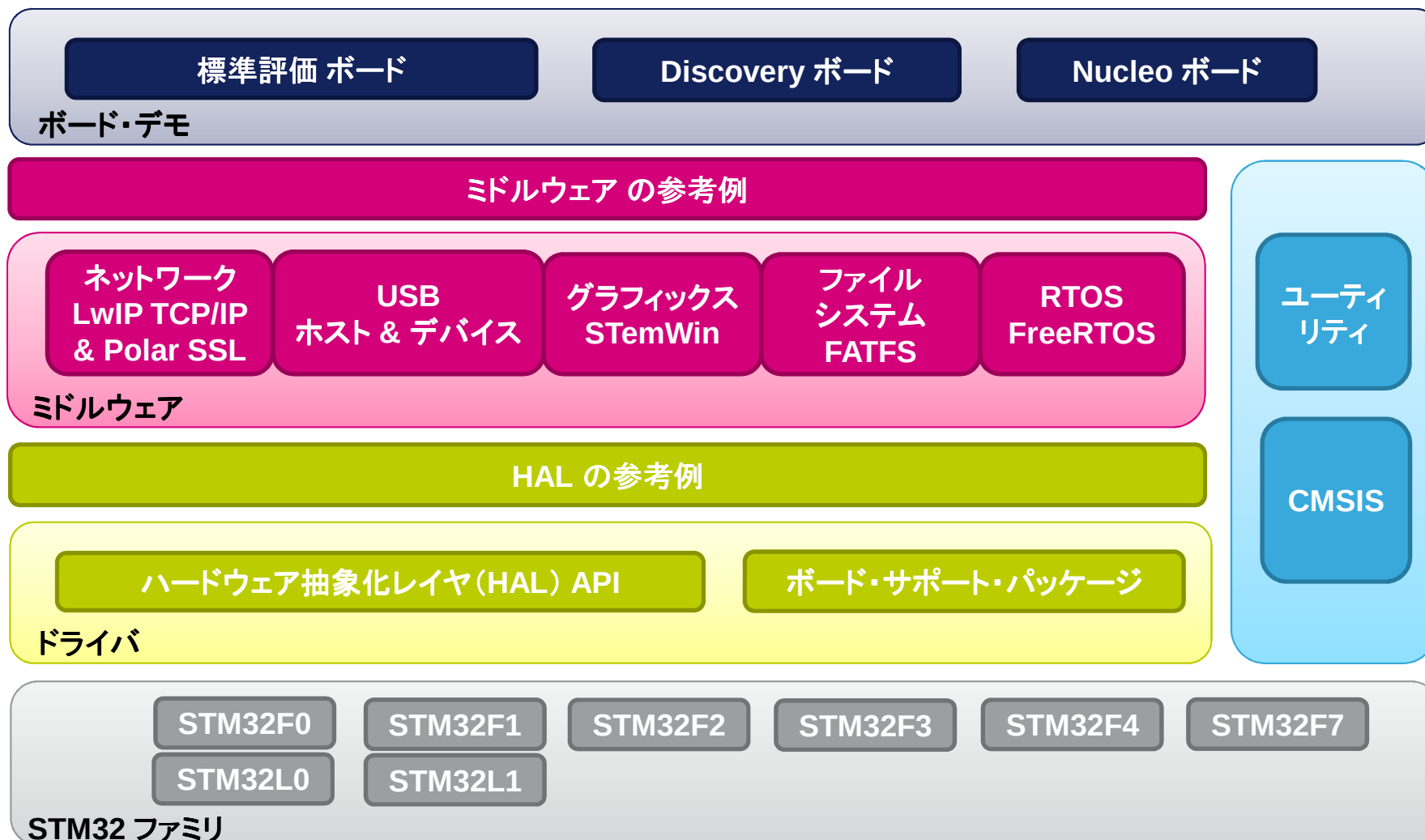
- TCP/IPスタック
 - LwIP オープンソース規格
- USBライブラリ
 - ホスト、デバイス・ライブラリ (ST)
- グラフィックス
 - STemWin (STおよびSEGGER)
- ファイルシステム
 - FatFS オープンソース規格
- リアルタイムOS
 - FreeRTOS オープンソース規格 (CMSIS-RTOSに準拠)
- サンプルプログラム
 - 50種類のサンプルプログラム

- STM32の抽象化
 - 移植が容易なAPI
- すべてのSTM32に対応
 - 全てのペリフェラルを網羅
- 量産対応
 - 品質: CodeSonar™ (高精度バグ検出ツール)
- サンプルプログラム
 - 150種類の使用例
- 自由なライセンス形態
 - オープンソースBSDライセンス

注意: 青色はオープンソフトウェア

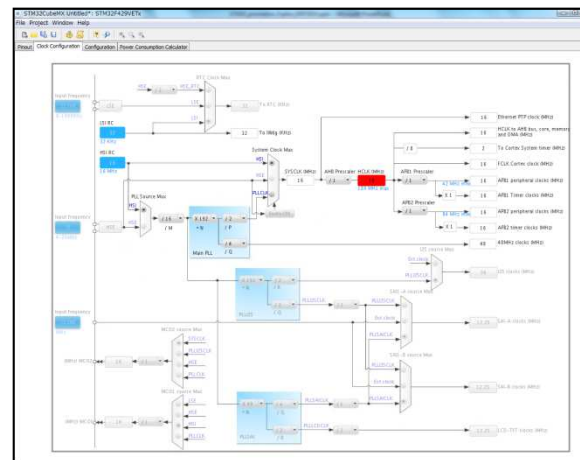
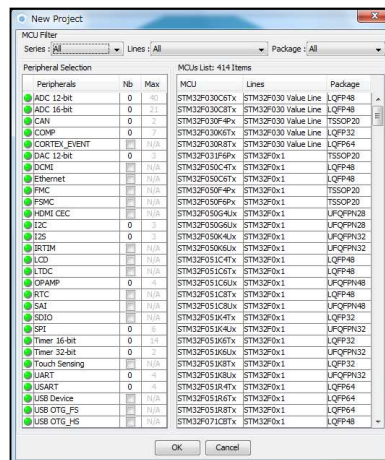
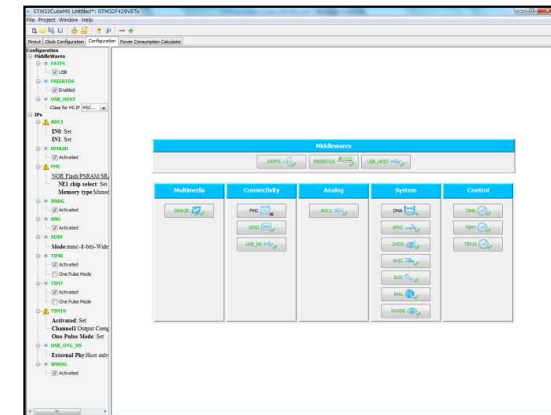
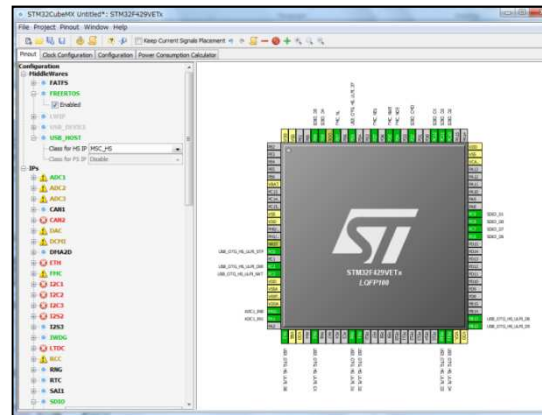
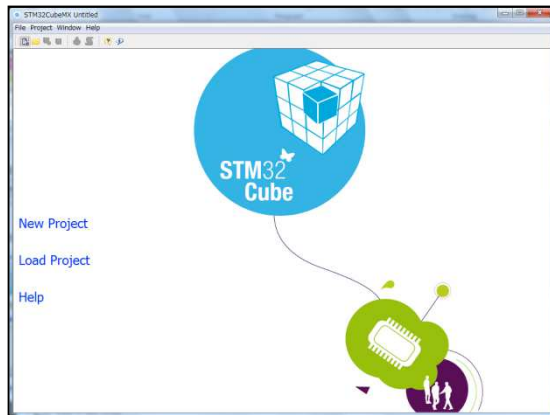
STM32Cube FW パッケージ ブロック図

8



STM32CubeMXの紹介

9



Microcontroller Selection

Family: STM32L1
Subfamily: STM32L151VC
MCU: STM32L151VCx
Datasheet: 027799_Rev2
Part Number: STM32L151VC

Parameter Selection

Ambient Temperature (°C): 25
Vdd Power Supply (V): 3.6

Battery Selection

Battery: Li-SOCL2(D19000)
Capacity: 19000.0 mAh
Self Discharge: 0.08 %/month
Nominal Voltage: 3.6 V
Max Cont Current: 230.0 mA
Max Pulse Current: 500.0 mA

Sequence

Load Save Delete

Step	M...	Vcore	M...	Pre...	CP...	Pe...	U...	Av...	Du...	O...
1	SLEEP	Range...	FLASH M...	4.2...	4.2...	MALL	0 mA	230...	5 ms	0.0
2	LOW	NoRange	RAM	131...	131...	MALL	0 mA	22.0...	4 ms	0.0
3	RUN	Range...	FLASH M...	16.0...	16.0...	ADC...	0 mA	7.23...	40 ms	16.8
4	LOW	NoRange	FLASH M...	131...	131...	CRC...	0 mA	407...	30 ms	0.16...

Step Add Delete Duplicate Up Down

Sequence Chart

Results

Total Sequence Time: 79.0 ms Average Consumption: 3.835 mA
Battery Life Estimation: 6 months Average DMPS: 9.67 DMPS

STM32Cubeの関連資料一覧

10

UM1709: イーサネットを使用したIAPの一例

UM1713: LwIP TCP/IPスタックを使用したアプリケーション開発

UM1720: STM32Cube USB ホストライブラリ

UM1721: FatFsを使用したアプリケーション開発

UM1722: RTOSを使用したアプリケーション開発

UM1723: PolarSSLの一例

ミドルウェア関連

UM1725: STM32F4xx HAL(ハードウェア抽象化レイヤ)ドライバ説明書

UM1730: STM32F4のための STM32CubeF4ファームウェアパッケージ導入ガイド

UM1734: STM32Cube USB デバイスライブラリ

ミドルウェア関連

UM1743: STM32CubeF4 デモプラットフォーム

デモソフト関連

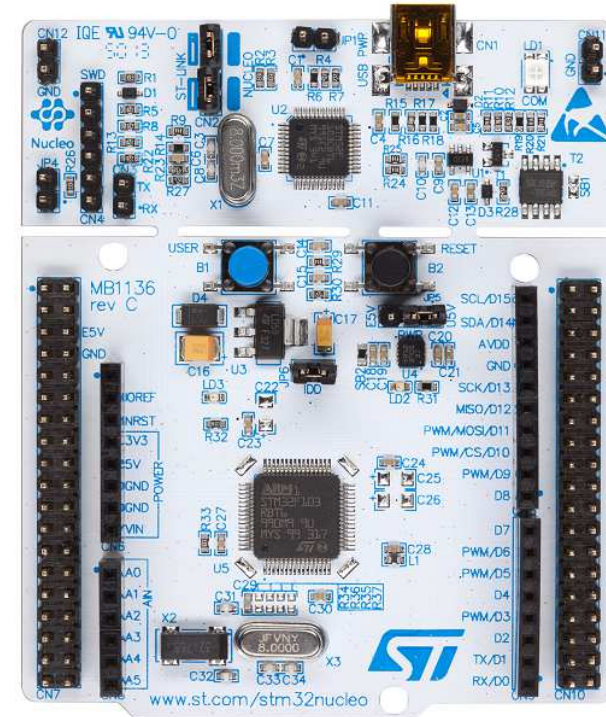


Nucleo

ARM mbed対応 Nucleoボード

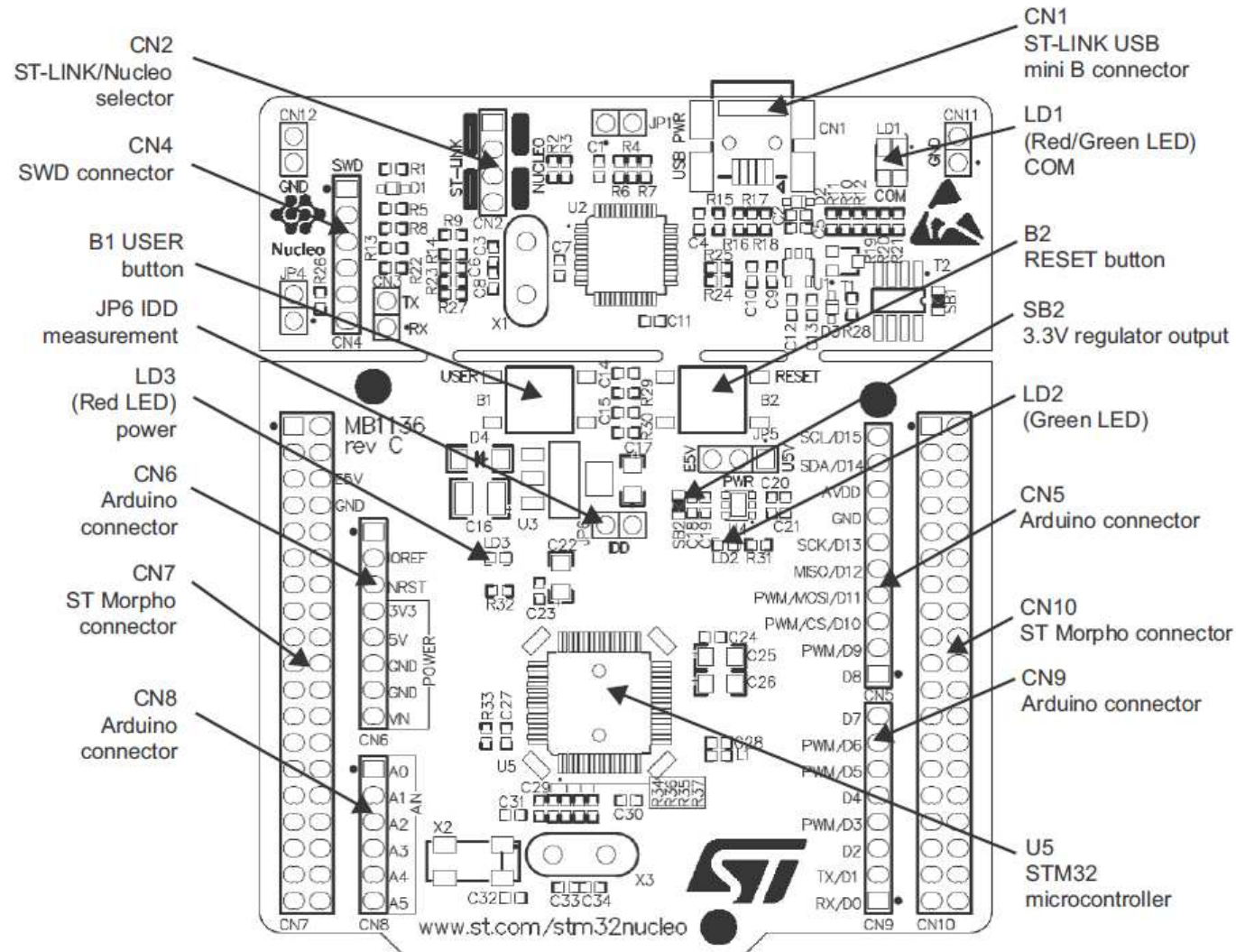
12

- Nucleoボードは、STM32マイコン間で共通
- 2種類の拡張ヘッダ:
 - **Arduino** Uno v3との接続
 - Morpho ヘッダにより全てのペリフェラルへのアクセスが可能
- オンボードST-LINK/V2-1デバッガとプログラマ:
 - mbedの利用が可能
 - ドラッグ&ドロッププログラミングに対応
 - Nucleoボード、および、外部のSTM32実装ボードをターゲットとして利用可能
- 試作に最適
 - 安価なSTM32搭載ボードはファミリ間で共通部品を実装
 - ハード/ソフトの両面での調整において費用的リスクを回避
- オープンコミュニティの利点を楽しむ
 - 制限の無い拡張性能
 - ペリフェラルシールドの巨大なエコシステム(Arduino)へ瞬時にアクセス可能
 - mbedのオンラインIDEを無償で利用可能



Nucleoボード 詳細

13

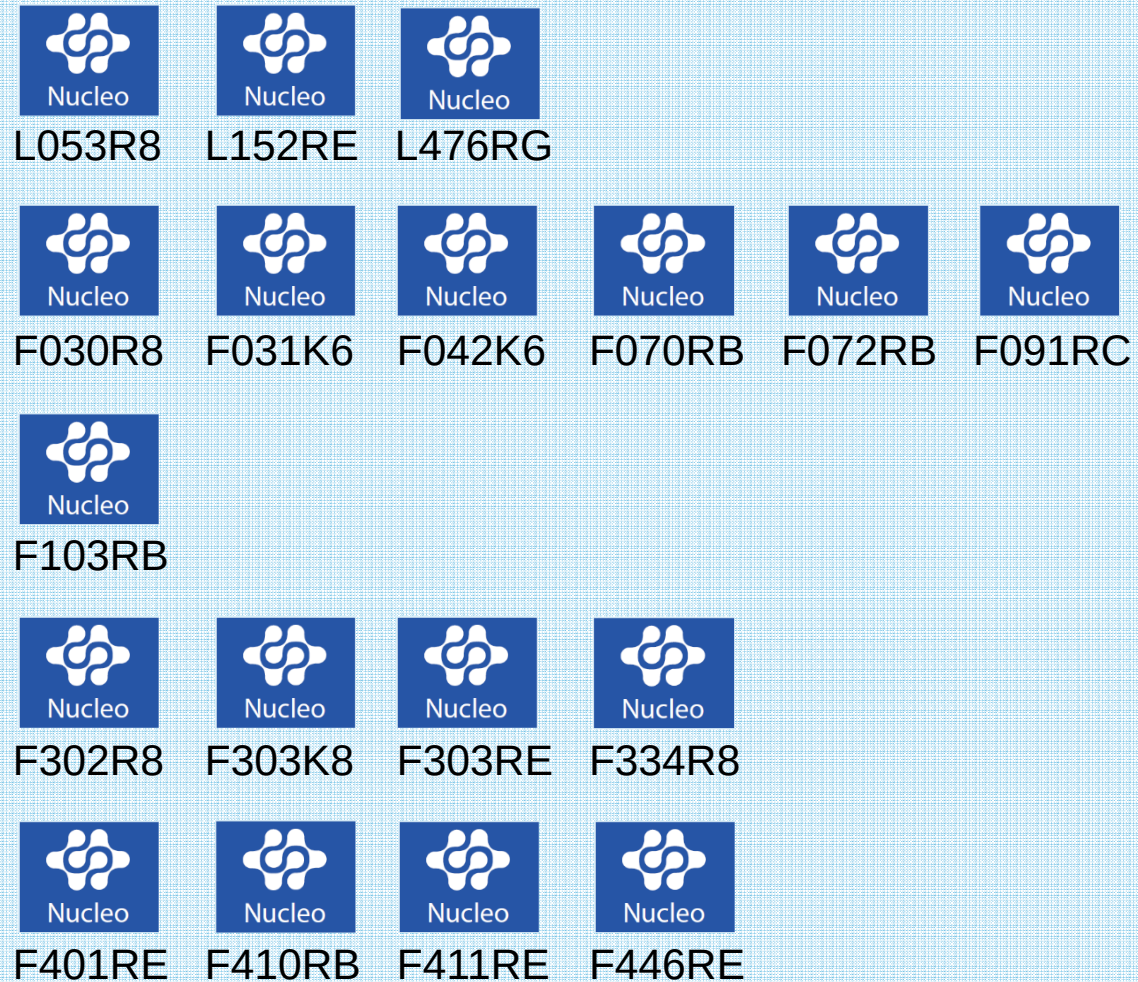


全18種類

Nucleo ボードのラインナップ

(2015年12月時点)

14





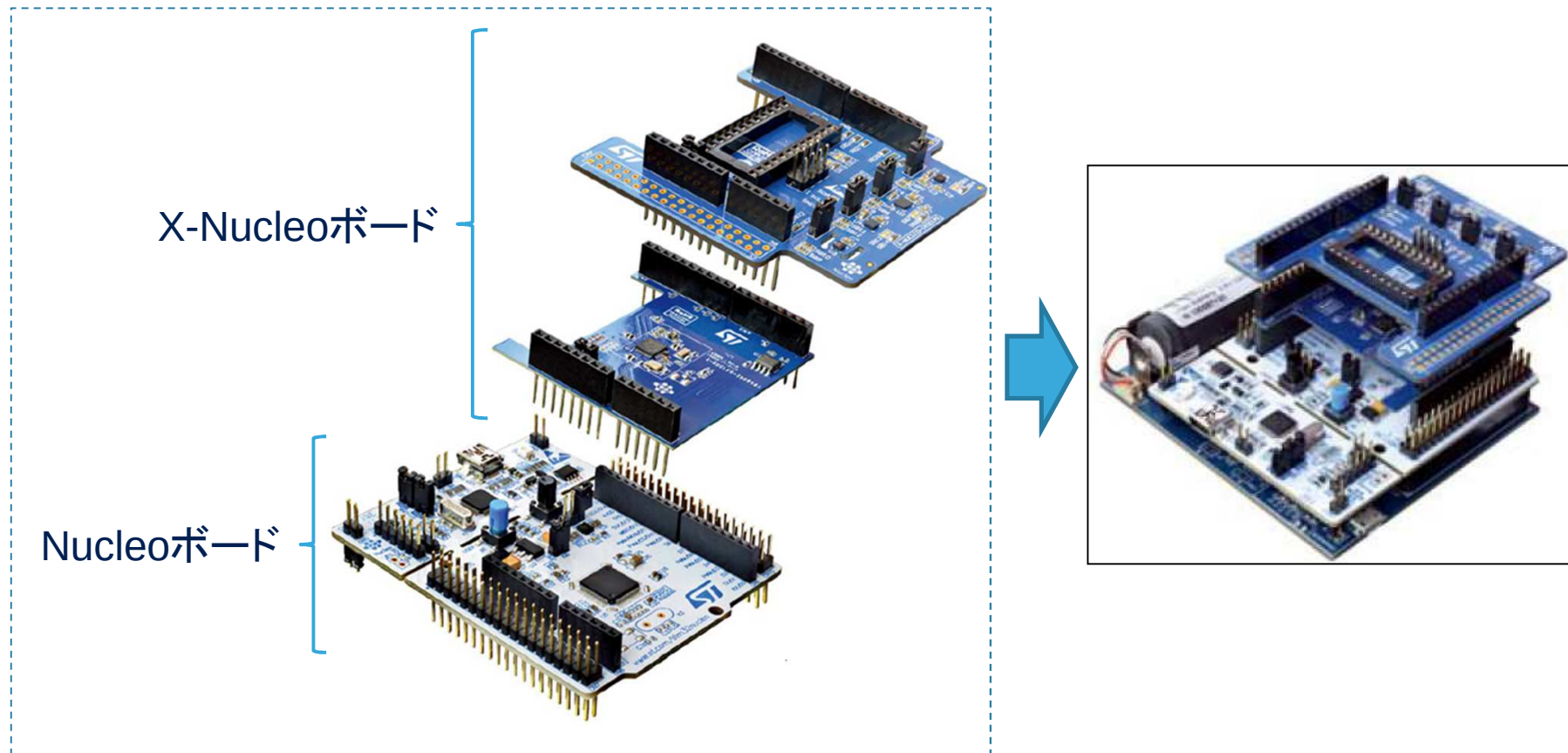
X-Nucleo

X-Nucleo ボードの紹介

16

X-Nucleo ボード

Nucleo ボードに重ねて機能する、マイコンのペリフェラルを搭載した拡張ボードです。



全18種類

X-Nucleoボードのラインナップ

(2015年12月時点)

17



X-NUCLEO-IKS01A1



X-NUCLEO-IDS01A5



X-NUCLEO-6180XA1



X-NUCLEO-NFC01A1



X-NUCLEO-CCA02M1



X-NUCLEO-IHM01A1



X-NUCLEO-IDB04A1



X-NUCLEO-IHM03A1



X-NUCLEO-IDS01A4



X-NUCLEO-IHM07A1

カテゴリ	機能	製品型番
センサ	モーション・センサ	X-NUCLEO-IKS01A1
	近接センサ	X-NUCLEO-6180XA1
	マイク	X-NUCLEO-CCA02M1
通信	Bluetooth Low Energy	X-NUCLEO-IDB04A1 X-NUCLEO-IDB05A1
	Sub-1GHz無線	X-NUCLEO-IDS01A4 X-NUCLEO-IDS01A5
	NFC	X-NUCLEO-NFC01A1
	WiFi	X-NUCLEO-IDW01M1
モータ制御	モータ・ドライバ	X-NUCLEO-IHM01A1 X-NUCLEO-IHM02A1 X-NUCLEO-IHM03A1 X-NUCLEO-IHM04A1 X-NUCLEO-IHM05A1 X-NUCLEO-IHM07M1 X-NUCLEO-IHM09M1
PLC	産業向け入出力	X-NUCLEO-PLC01A1
拡張	機能拡張	X-NUCLEO-IKA01A1



X-NUCLEO-PLC01A1



STM32Cubeを使用した Nucleo-F401の実習

事前に準備が必要なもの

19

- Windows7のインストールされたPC
- 管理者権限のあるユーザーアカウント
- IAR EWARM ver.7.40.5のインストールおよびライセンスの取得
 - コードサイズ制限版
- Java VMのインストール
- インターネット環境



ツールチェーン & ライブラリのインストール

サンプルプログラムを動かそう

21

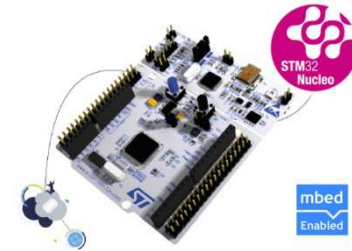
- STM32Cube F4に付属しているサンプルプログラムを動かしてみましょう
- 以下の手順を踏みます
 - 開発ツールをインストール
 - STLINKのドライバをインストール
 - NUCLEO-F401REを準備
 - STM32Cube F4を展開
 - サンプルプログラムを実行

本日使用する開発環境

22

評価ボード

NUCLEO-F401RE
(STM32F401搭載のmbed対応ボード)



統合開発環境

IAR社 EWARM v7.40.5



ソフトウェア・ライブラリ

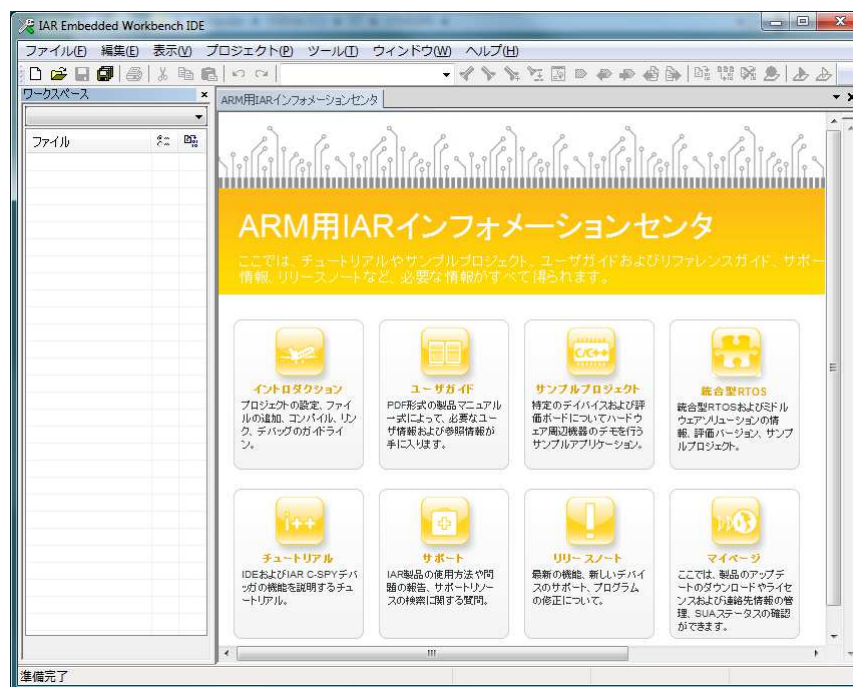
STM32CubeF4 v1.8.0
STM32CubeMX v4.10.0



開発ツールのインストール

23

- IAR 7.40.5をダウンロードします
- インストーラーを起動してインストールします
- 立ち上げて評価用ライセンスをインターネット経由で入手します



STLINKドライバインストール

24

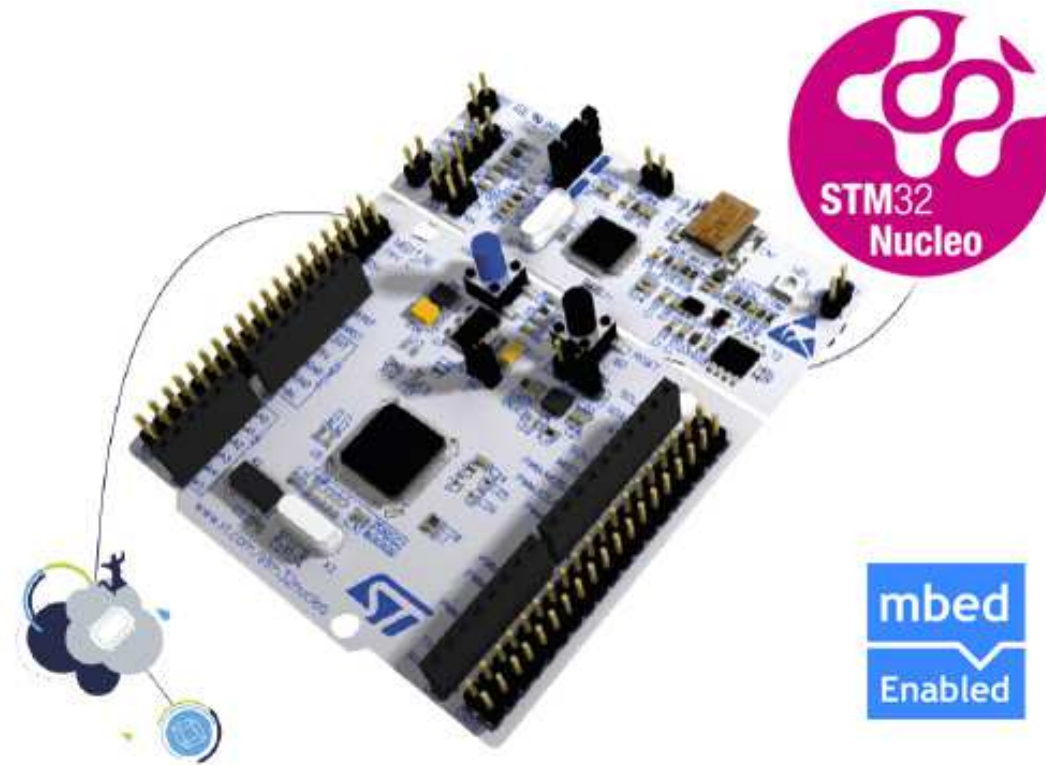
- エクスプローラでドライバの位置を開きます
- C:\Program Files(x86) \IAR Systems \Embedded Workbench 7.xxxx \arm
 \drivers \ST-Link \Win_7-8
- stlink_winusb_install.batを実行します

Name	Date modified
dpinst_amd64.exe	2014/03/22 12:38
dpinst_x86.exe	2014/03/22 12:38
readme.txt	2014/03/22 12:38
stlink_dbg_winusb.inf	2014/03/22 12:38
stlink_VCP.inf	2014/03/22 12:38
stlink_winusb.inf	2014/03/22 12:38
stlink_winusb_install.bat	2014/03/22 12:38
stlink_winusb_uninstall.bat	2014/03/22 12:38
stlinkdbgwinusb_x64.cat	2014/03/22 12:38
stlinkdbgwinusb_x86.cat	2014/03/22 12:38
stlinkvcp_x64.cat	2014/03/22 12:38
stlinkvcp_x86.cat	2014/03/22 12:38
stlinkwinusb_x64.cat	2014/03/22 12:38
stlinkwinusb_x86.cat	2014/03/22 12:38

Nucleo-F401REを準備

25

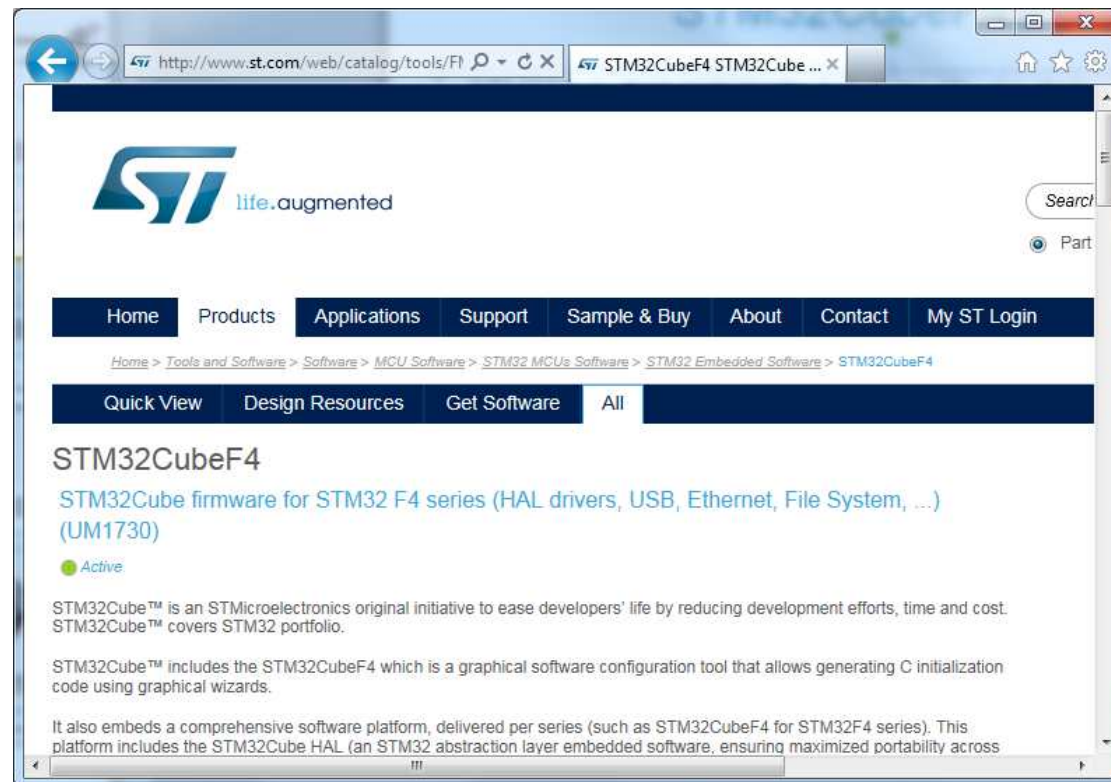
Nucleo-F401RE



STM32CubeF4をダウンロード

26

- STM32CubeF4を以下のURLからダウンロード
- <http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1743/PF259243>



STM32CubeF4を解凍(1/2)

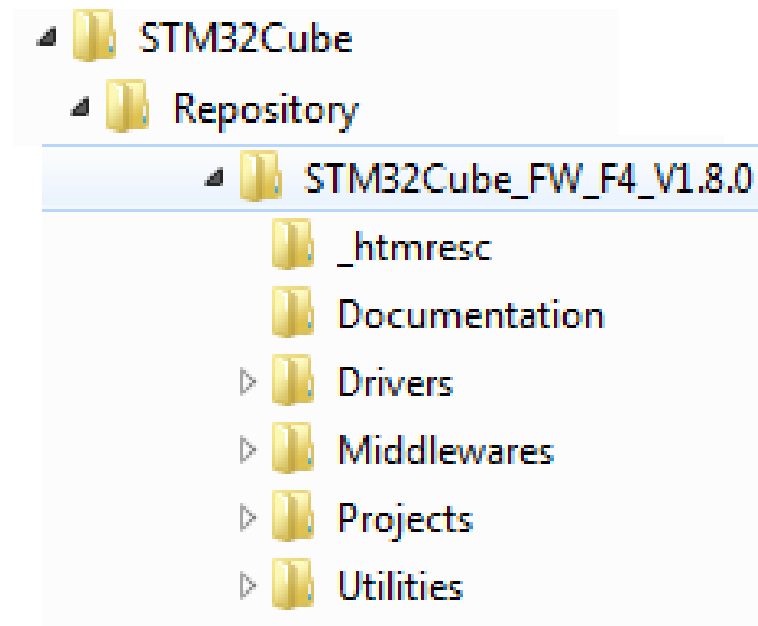
27

- ダウンロードしたstm32cubef4.zipを解凍します
- C:¥Users¥<ログインユーザー名>¥STM32Cube¥Repositoryに解凍します
- 注意: ログインユーザー名が日本語の方はここに解凍しても動作しません！
該当する方はc:¥STに解凍します

STM32CubeF4を解凍(2/2)

28

- 以下のようなディレクトリ構成にします



- 注意:この後、STM32CubeMXでこのディレクトリを参照しますので、必ずこのディレクトリに解凍してください

STM32CubeMXのインストール

29

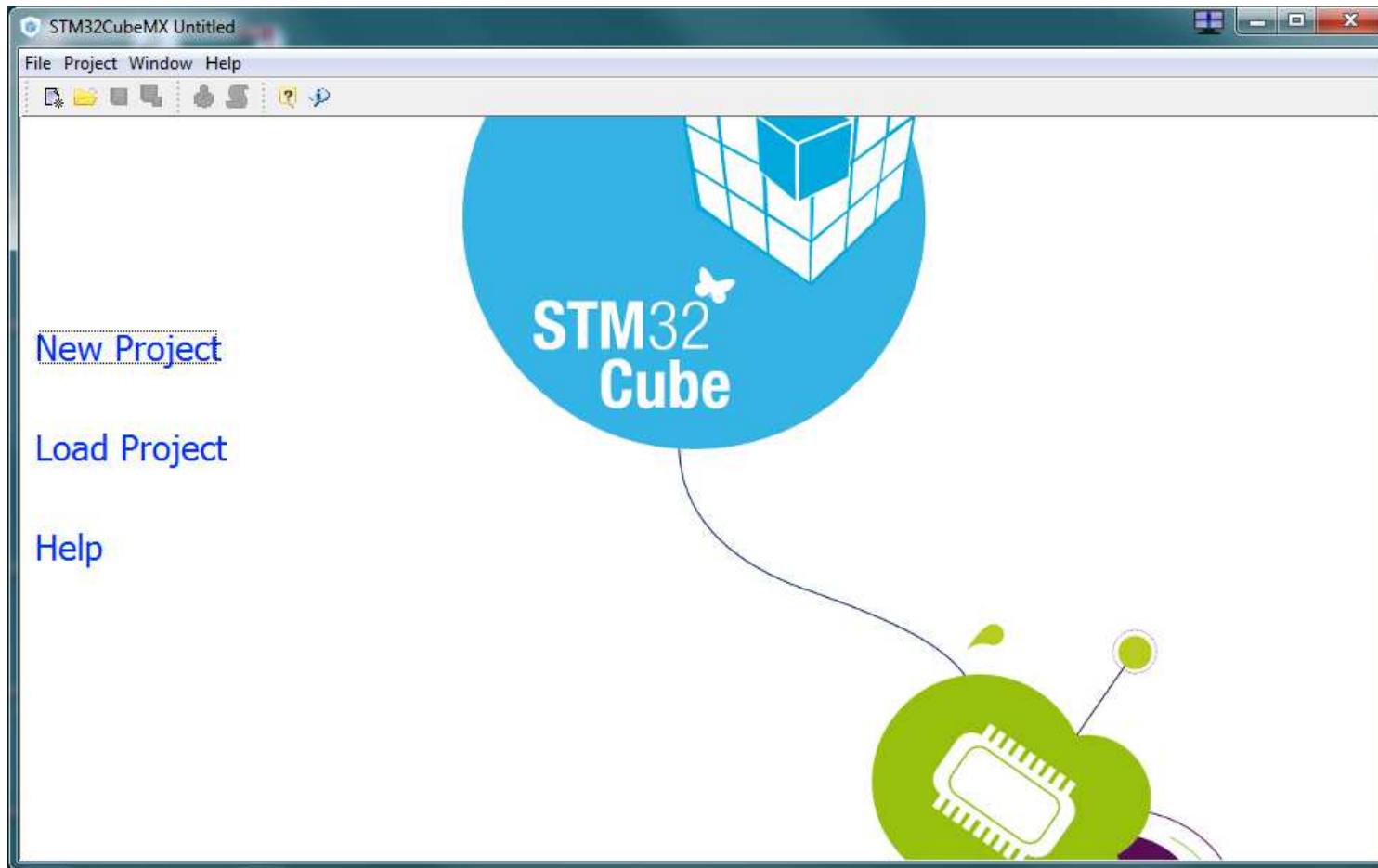
- <http://www.st.com/stm32cube>からSTM32CubeMXをダウンロードします
- 解凍すると、SetupSTM32CubeMX-4.10.0.exeが入っていますので、ダブルクリックして実行します
- 環境によってはJavaVMのダウンロードが始まります
- デフォルトのままインストールすると、デスクトップにアイコンができます



STM32CubeMXの起動

30

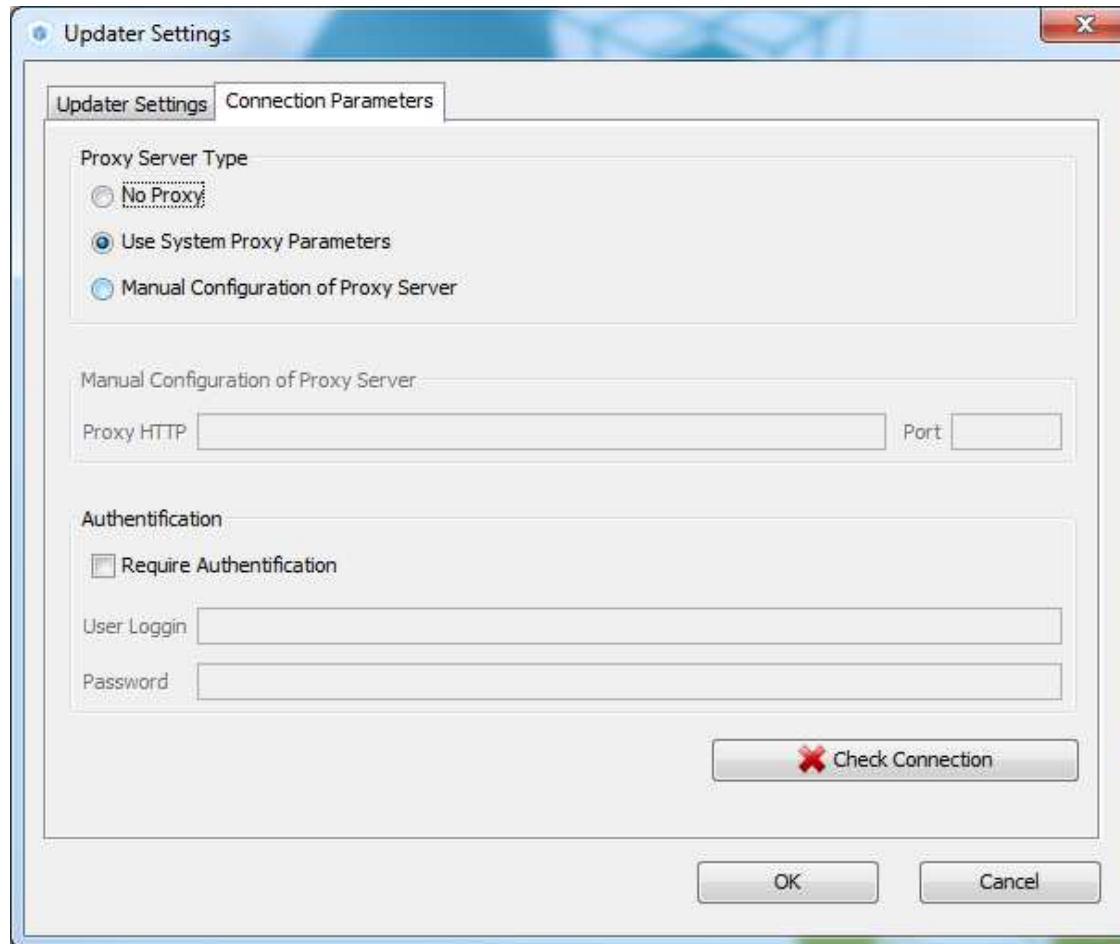
- 起動すると以下のような画面になります



インターネット接続の設定

31

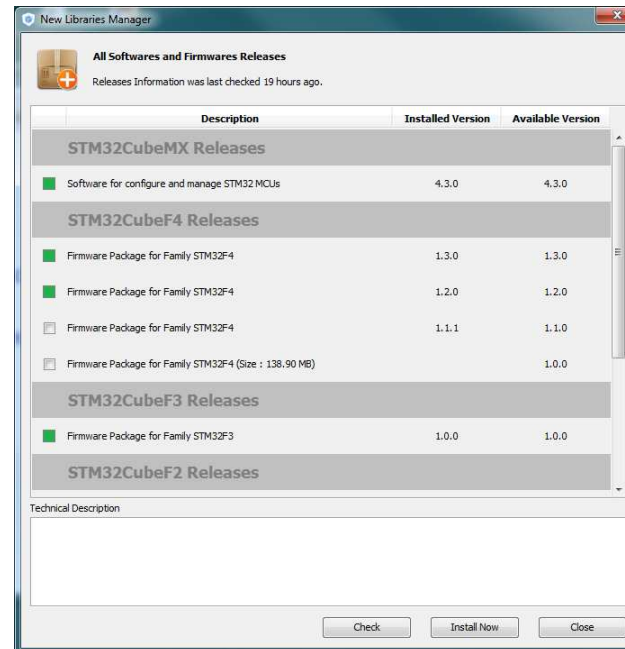
- Help->Updater Settingからインターネット接続の設定を行います



ライブラリのダウンロード

32

- Help->Install New Librariesから最新のライブラリをダウンロードします

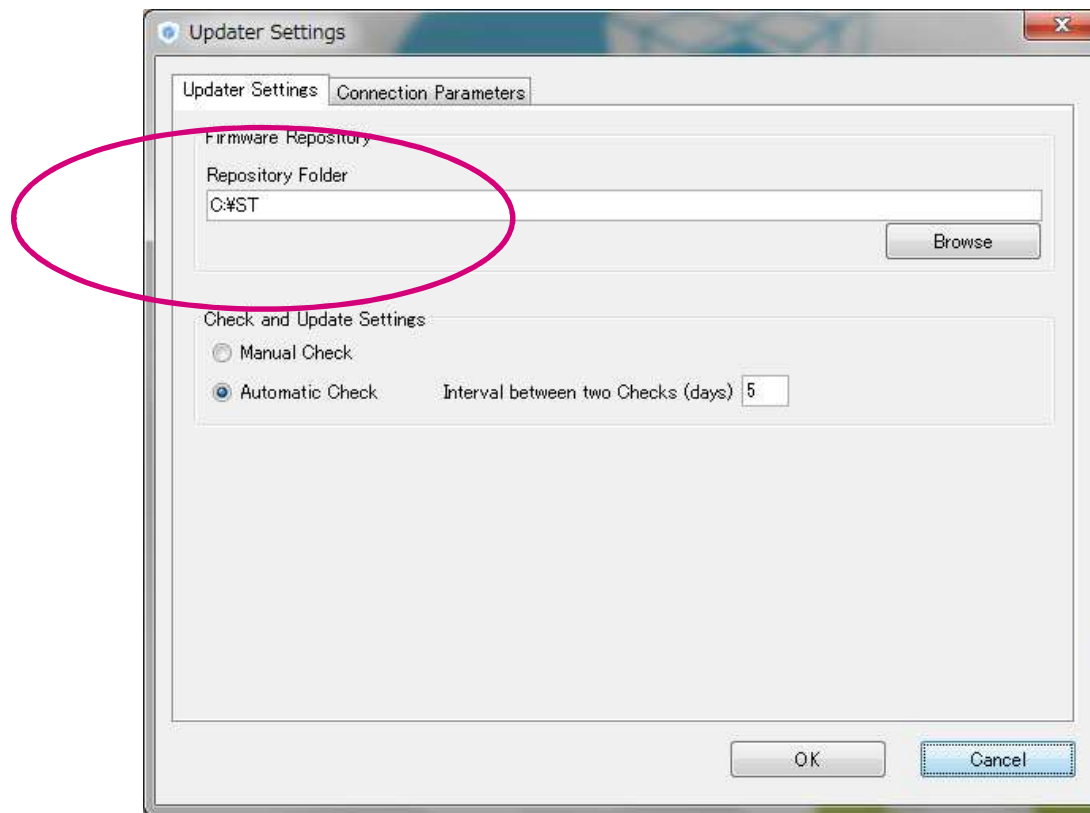


- 事前にzipファイルを所定の位置に解凍してある場合、チェックボックスが緑色になります
- 注意: ネットワークへの接続が必要です！

C:¥STに解凍した方

33

- Help->Updater SettingsのUpdater Settingsタブで、Repository Folderに“C:¥ST”を設定します



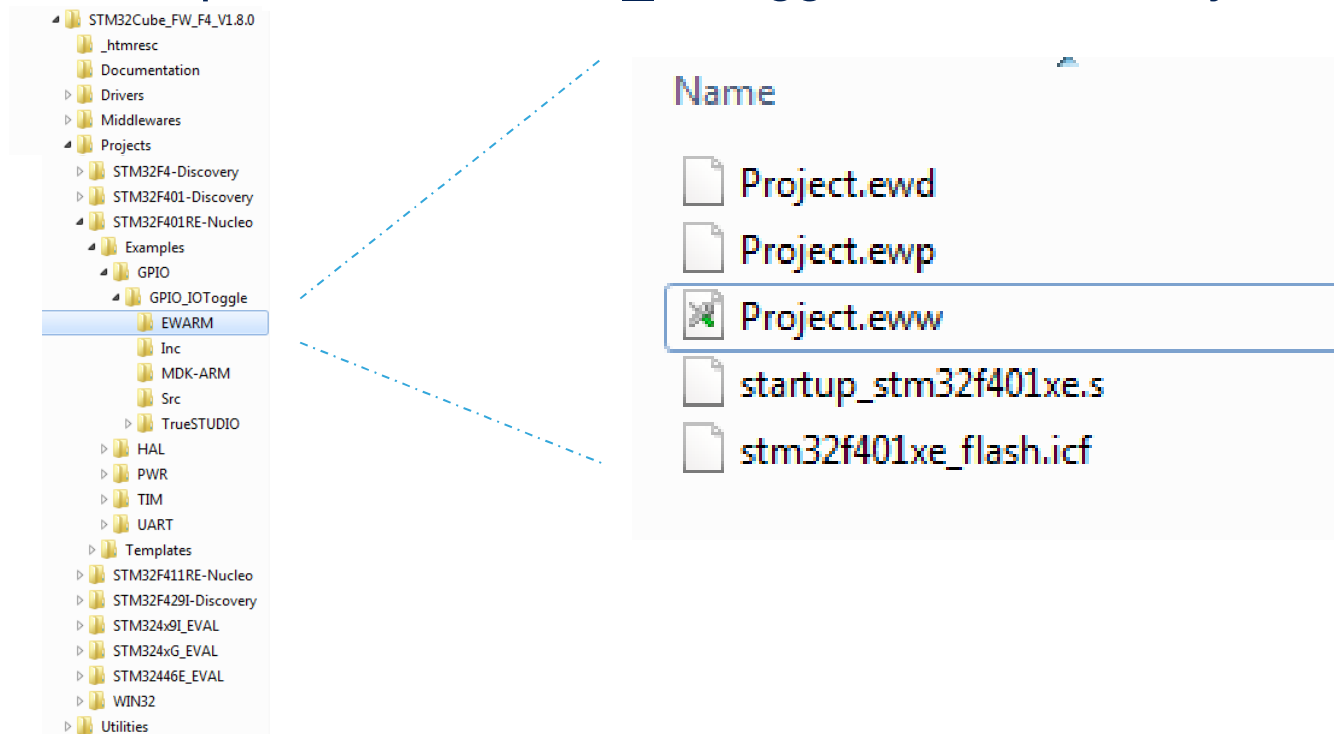


実習1: ツールの動作確認

サンプルプログラムを実行(1/7)

35

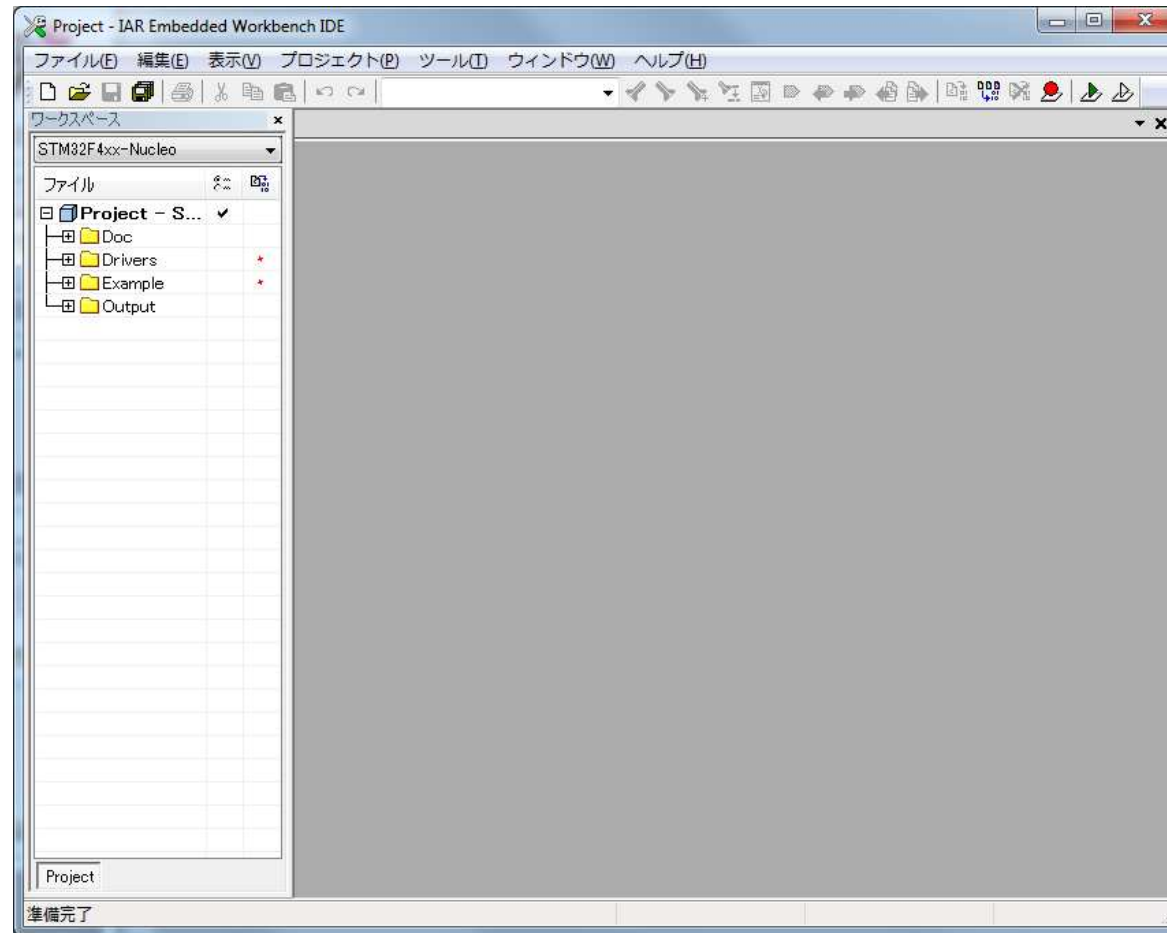
- 既存プロジェクトを開きます
- ファイル＞開く＞ワークスペース にて以下のファイルを指定
- STM32Cube_FW_F4_V1.8.0¥Projects¥STM32F401RE-Nucleo¥Examples¥GPIO¥GPIO_IOToggle¥EWARM¥Project.eww



サンプルプログラムを実行(2/7)

36

- Embedded Workbench起動画面

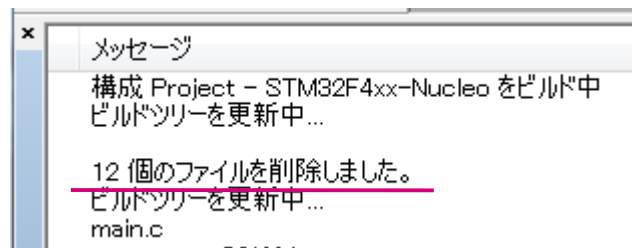


サンプルプログラムを実行(3/7)

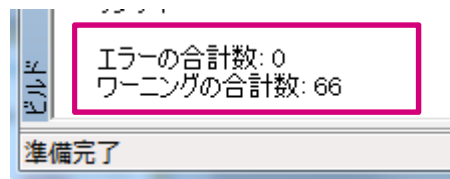
37

- プロジェクトの再ビルドを行います

最初に、オブジェクトコードが削除されます



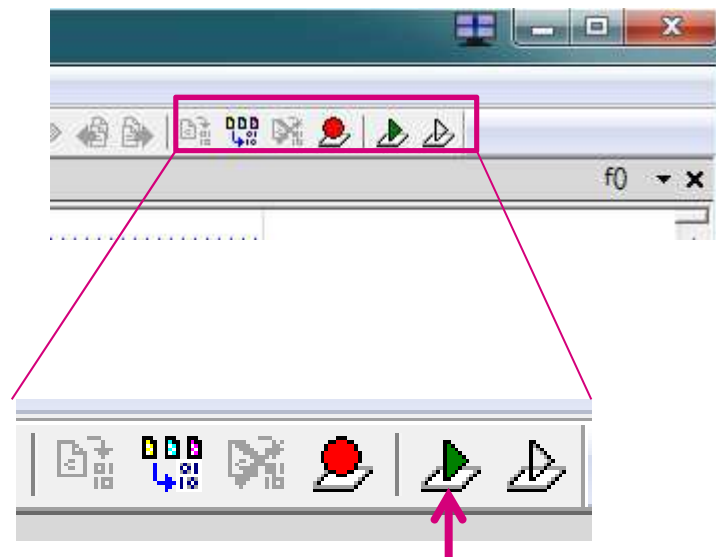
最後に、エラーとワーニングの数が表示されます



サンプルプログラムを実行(4/7)

38

- ターゲットにダウンロードしてデバッガを起動
 - プロジェクト > ダウンロードしてデバッグ



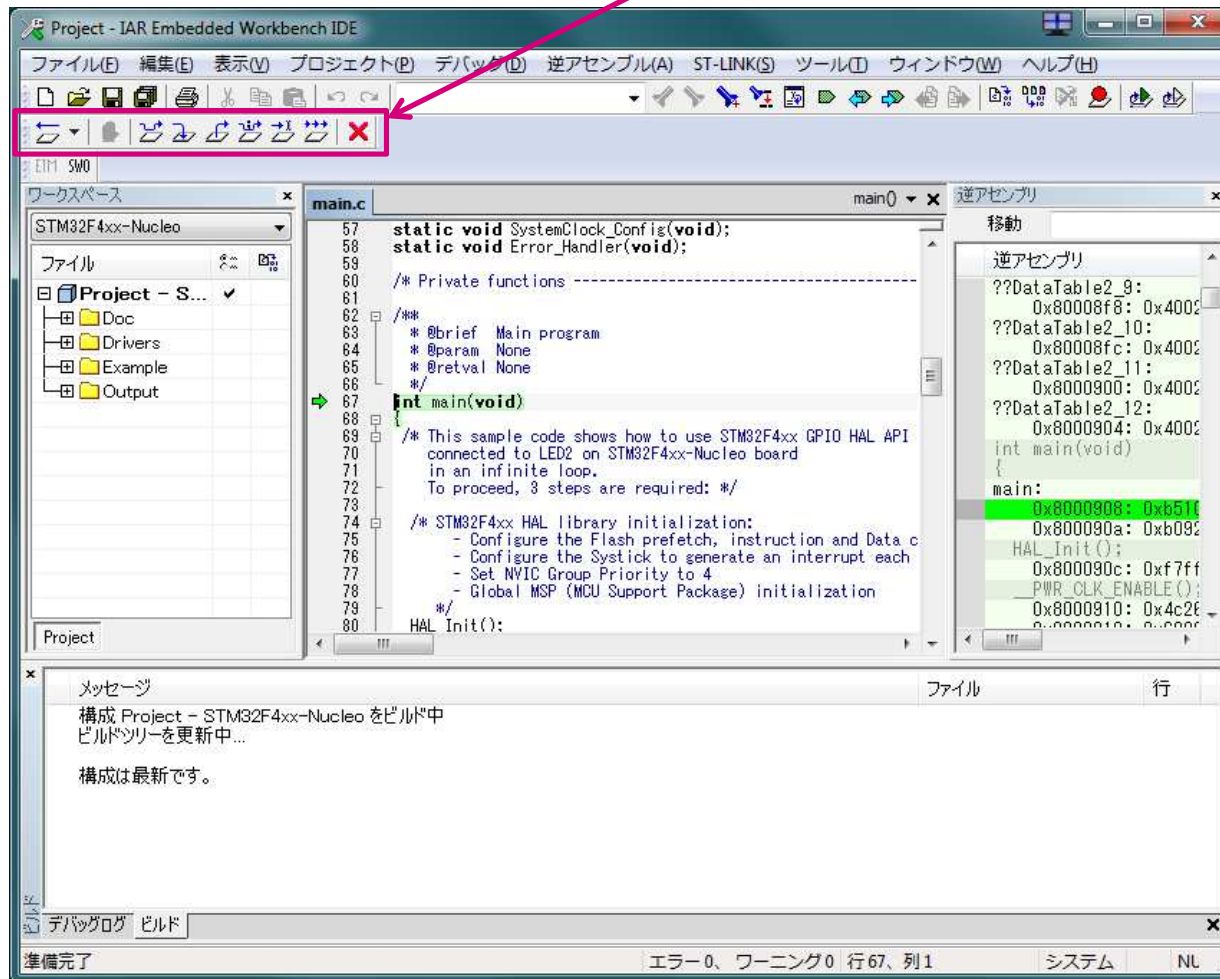
「ダウンロードしてデバッグ」
アイコン



サンプルプログラムを実行(5/7)

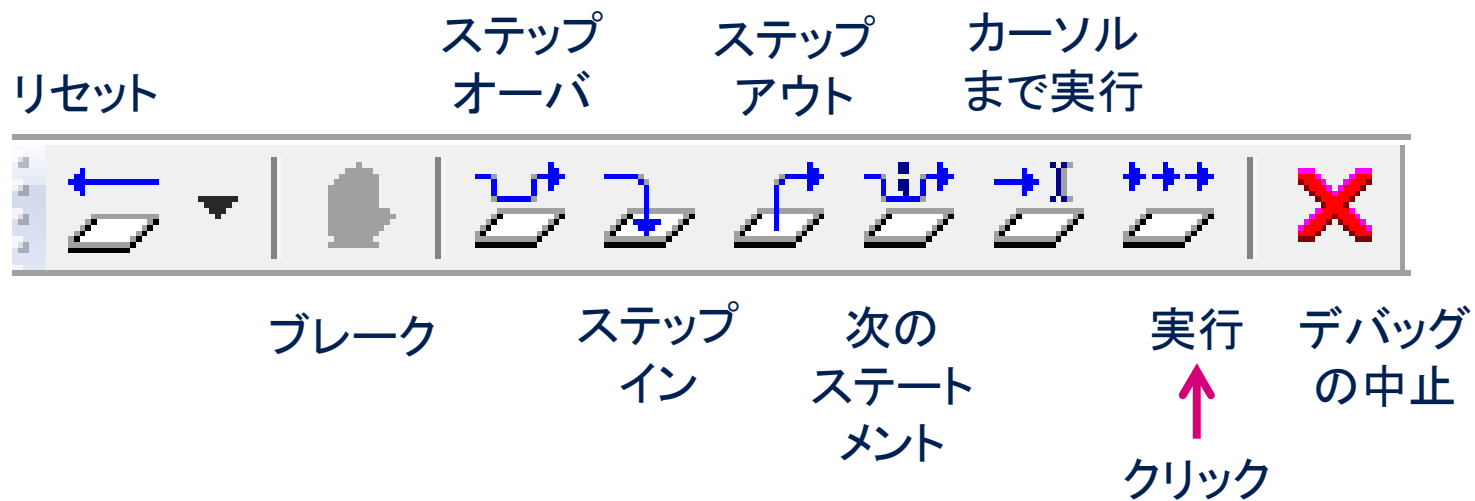
39

デバッグ用のツールバー



サンプルプログラムを実行(6/7)

40

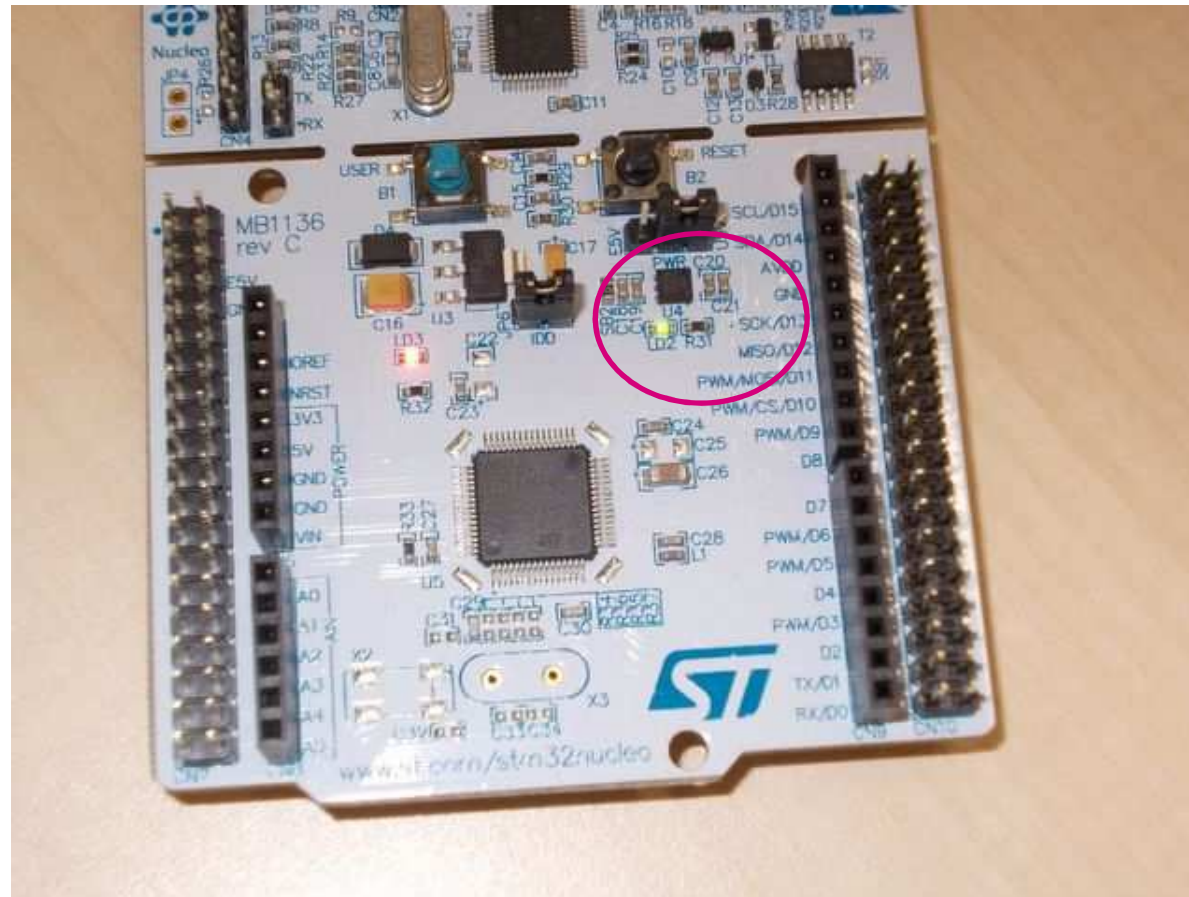


ここで、「実行」をクリック

サンプルプログラムを実行(7/7)

41

- 実行結果
 - 緑のLEDが点滅します





実習2: LED点滅

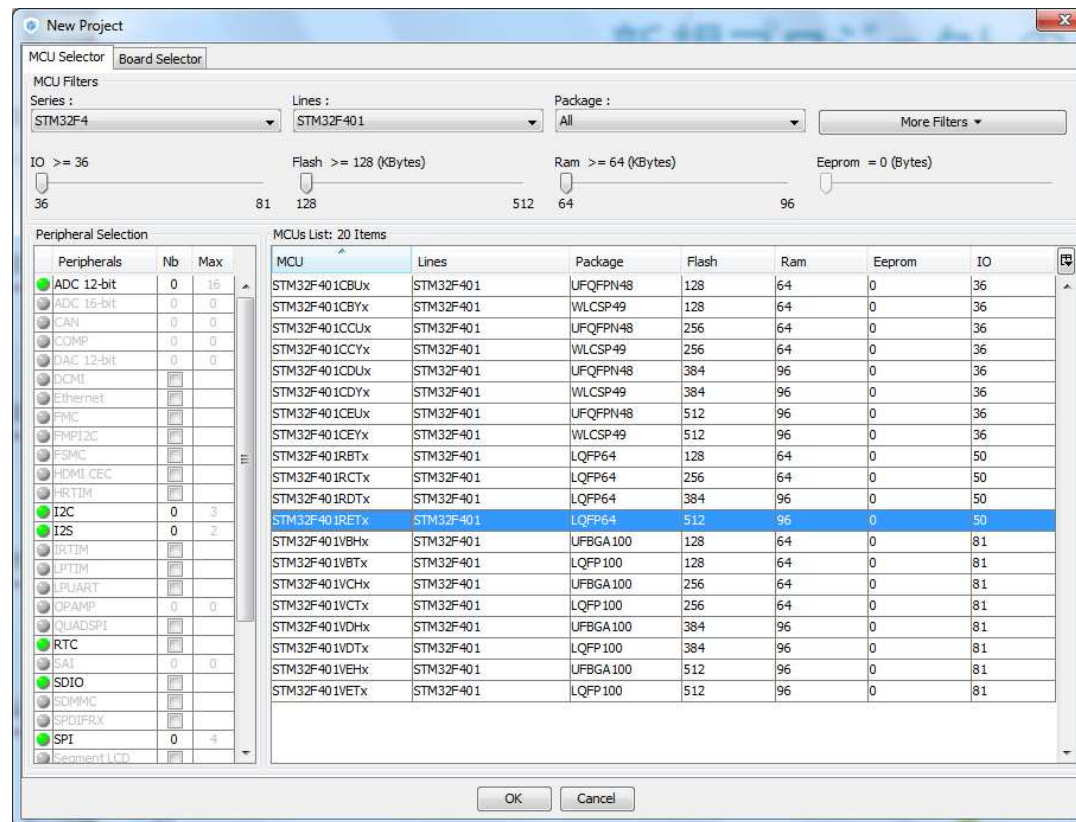
- NUCLEO-F401REを使用します
- NUCLEO-F401REにはあまり部品が載っていないので、USERボタンを押すたびに、LD2をトグルするという仕様にします

- USERボタンは割り込みで検知します
- メインループは以下のようにコーディングします
 - While (1) {
 - USERボタンが押されるのを待つ
 - LD2をトグルする
 - }
- 割り込みは以下のようにコーディングします
 - If (割り込みがあった) {
 - 変数に1を代入
 - }

新規プロジェクトの作成

45

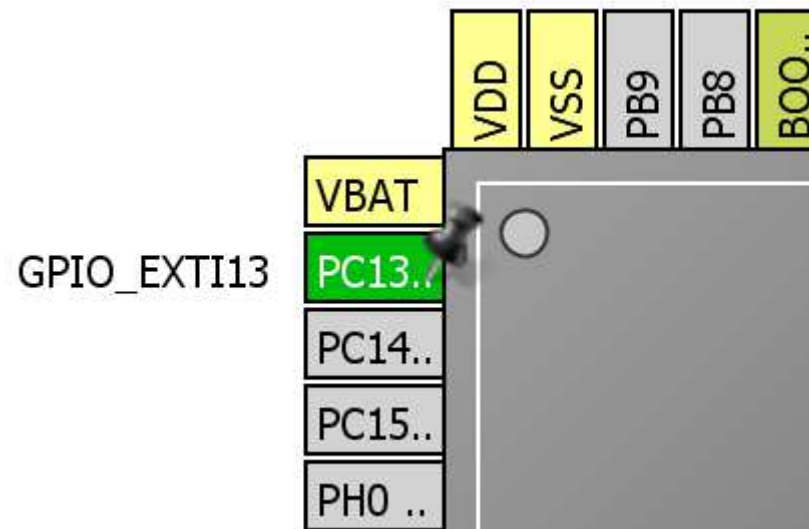
- STM32CubeMXを起動して、**New Project**を選択します
- Series->F4、Line->F401、STM32F401RETxを選択します



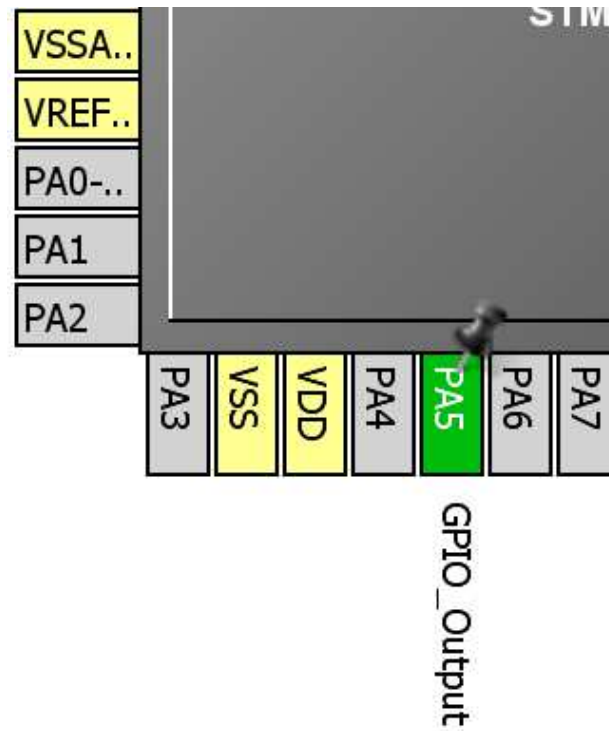
USERボタン入力

46

- PC13をGPIO割り込み入力に設定します



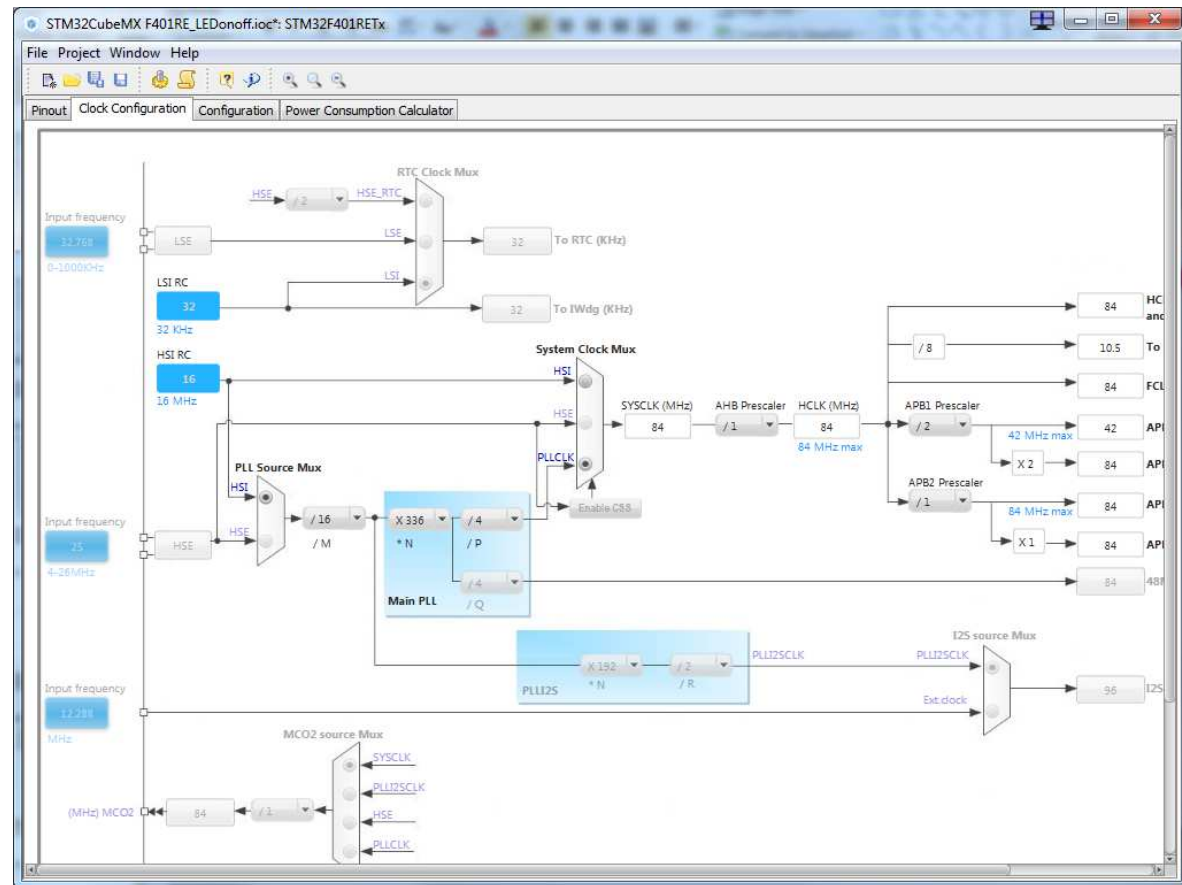
- PA5をGPIO出力に設定します



クロック設定

48

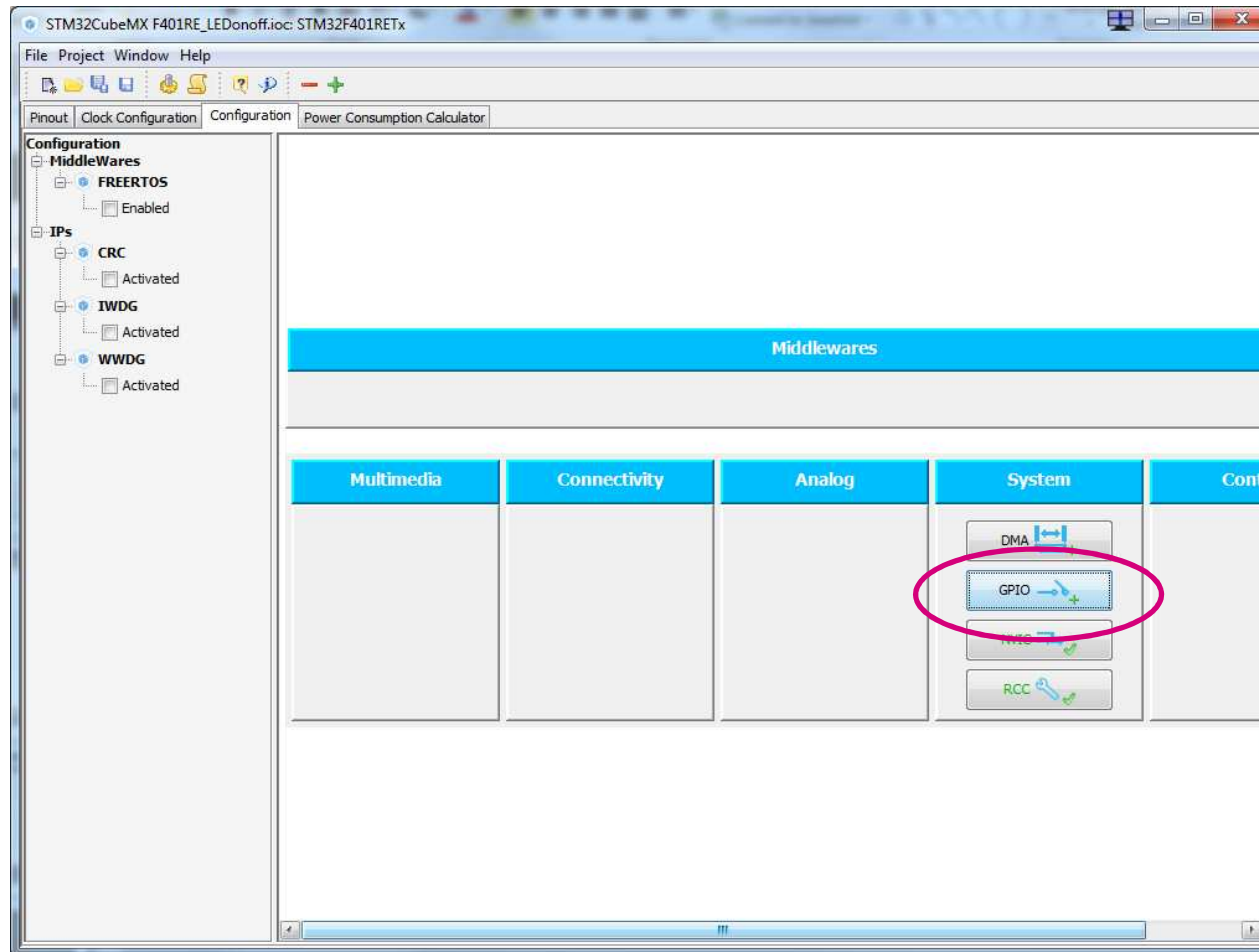
- 以下のように設定します
 - System CLK Mux -> PLLCLK
 - PLL source -> HSI
 - PLL N -> 336
 - PLL P -> 4
 - APB1 Prescaler -> 2



GPIO設定(1/2)

49

- ConfigurationタブのGPIOボタンを押します

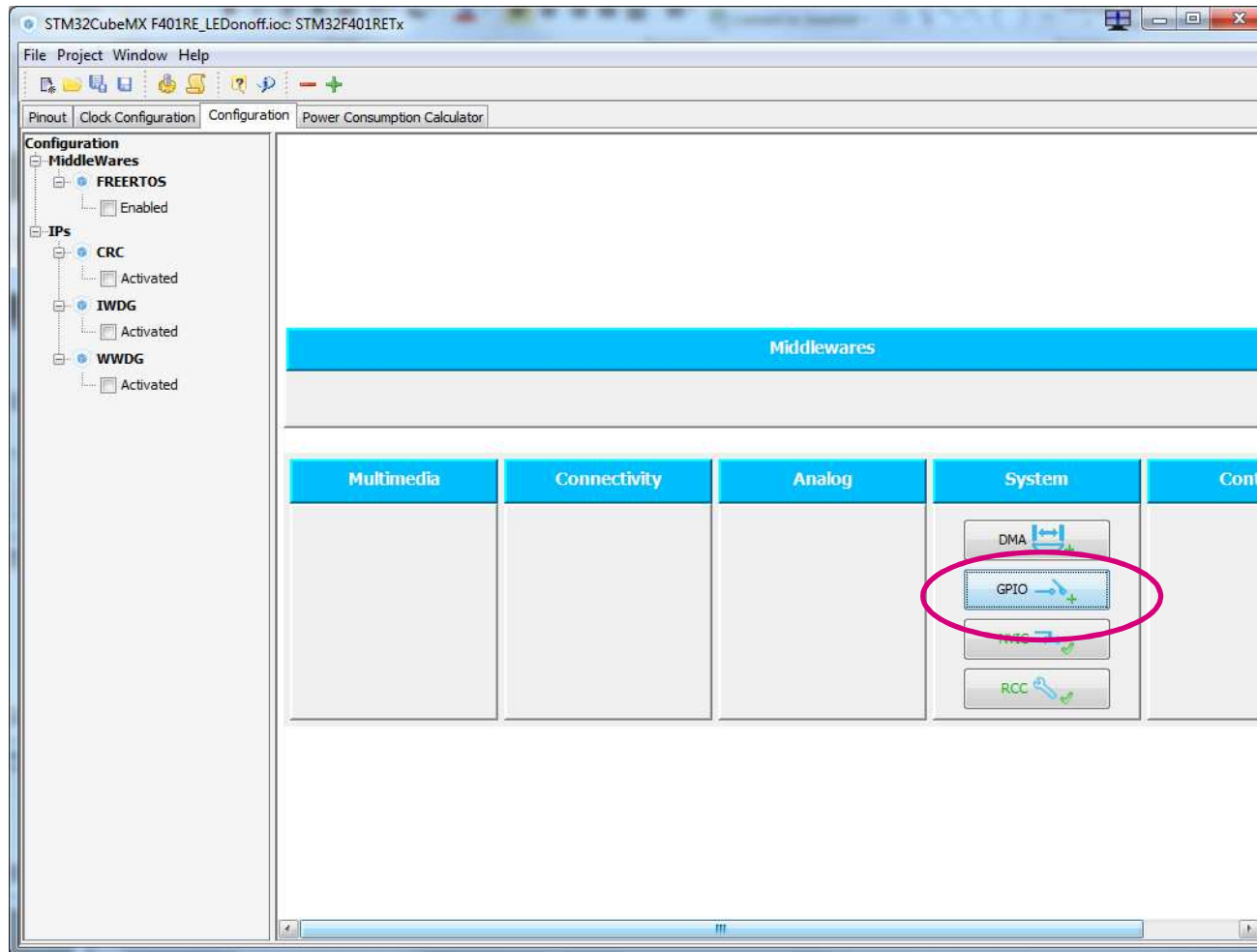


- 以下のように設定します
- PA5
 - Output push-pull
 - No pull-up and no pull-down
 - High
- PC13
 - External Interrupt Mode with Falling edge trigger detection
 - No pull-up and no pull-down

NVIC設定(1/2)

51

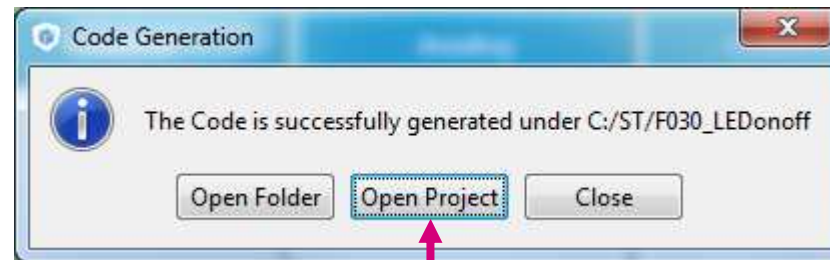
- ConfigurationタブのNVICボタンを押します



- 以下のように設定します
- Priority Group
 - 4bits for pre-emption priority 0 bits for subpriority
- System tick timer
 - Enabled: Checked
 - Preemption Priority: 0
 - Sub Priority: 0
- EXTI Line[15:10] interrupts
 - Enabled: Checked
 - Preemption Priority: 1
 - Sub Priority: 0

- Project->Settingを選択します
- Projectタブは以下のように設定します
 - Project Name -> F401RE_LEDOnoff
 - Project Location -> C:¥ST¥
 - Toolchain Folder Location -> C:¥ST¥F401RE_LEDOnoff
 - Tool chain -> EWARM
- Code Generatorタブは以下のように設定します
 - Copy only the necessary library files
 - Keep User Code when re-generating
 - Delete previously generated files when not re-generated
- OKを押してダイアログを閉じます
- Project->Generate Codeを選択します

- C言語の生成が終わると以下のようなダイアログがでます



ここを押すとIARが起動してプロジェクトを開きます

- または、C:\¥ST¥F401RE_LEDOnoff¥Projects¥EWARM¥Project.eww を開きます
- ビルドしてエラーなくコンパイルが通るのを確認します
 - このとき、プロジェクト>メイクも使用できます

ユーザー処理をいれる(1/3)

55

- “ボタンを押すと、LEDが点滅する”処理をいれます
- main.cに以下の処理を追加します

```
/* USER CODE BEGIN 0 */  
__IO uint8_t Pushed;  
/* USER CODE END 0 */
```

ユーザー処理をいれる(2/3)

56

- main.cに以下の処理を追加します

```
/* USER CODE BEGIN 3 */  
/* Infinite loop */  
while (1)  
{  
    Pushed = 0;  
    while (Pushed == 0)  
        ;  
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);  
}  
/* USER CODE END 3 */
```


ユーザー処理をいれる(3/3)

57

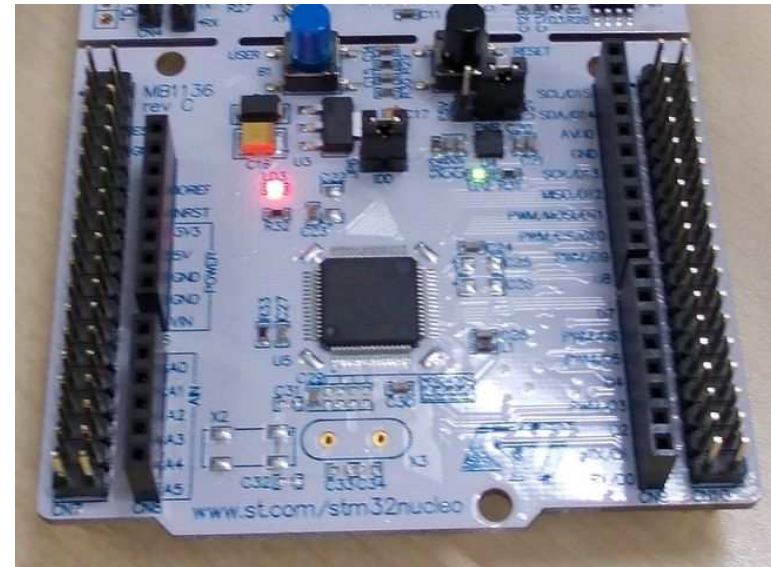
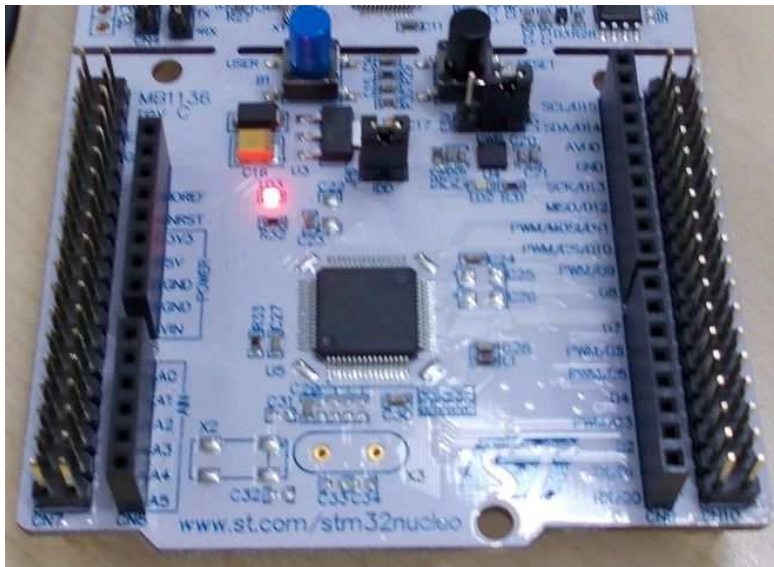
- main.cに以下の処理を追加します

```
/* USER CODE BEGIN 4 */  
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)  
{  
    if (GPIO_Pin == GPIO_PIN_13) {  
        Pushed = 1;  
    }  
}  
/* USER CODE END 4 */
```

IARでコンパイルおよび実行

58

- ビルドしてエラーなくコンパイルが通るのを確認します
- 実行して、USERボタンを押すたびに、LEDのON/OFFが切り替わるのを確認します



- IARをデバッグ停止、ファイル>ワークスペースの保存、ファイル>ワークスペースを閉じる、を行い終了します



まとめ

今回ご紹介したツールは開発方法に応じて、以下の様に使い分けて頂く事をお薦めします。

スクラッチからコーディングする場合：

⇒ STM32CubeMX を使用

サンプルコードを修正する場合：

⇒ STM32CubeF4のサンプルプログラム を使用