

STM32CubeIDE を使用した STM32MP1 シリーズ向けプロジェクトの開始方法

概要

このアプリケーションノートでは、ST マイクロエレクトロニクス社の STM32CubeIDE 統合開発環境で STM32MP1 シリーズ 向けプロジェクトを開始する方法について説明します。



1 一般情報

STM32CubeIDE は、Arm® Cortex® プロセッサをベースとして STM32 32 ビット製品をサポートします。

注 Arm は、米国内およびその他の地域にある Arm Limited 社 (またはその子会社) の登録商標です。



1.1 前提条件

以下のツールは、本書のチュートリアルを理解し、STM32MP1 シリーズ に基づいてアプリケーションを開発するための前提条件です。

- STM32CubeIDE 1.1.0 以降
- STM32Cube_FW_MP 1.1.0 以降
- STM32CubeMX 5.4.0 以降

STM32MP1 シリーズ の文書の更新について、www.st.com/en/microcontrollers-microprocessors/stm32mp1-series で最新情報を確認することをお勧めします。

1.2 本書内のユースケース

STM32CubeIDE のコンテキストでは、STM32MP1 シリーズ に基づいてプロジェクトの開発を調査し、開始するためにさまざまな方法が用意されています。以下のリストから、検討されているユースケースに最適な説明を選択し、このアプリケーションノート内の該当するセクションを参照してください。

- ioc ファイルを含む SW4STM32 プロジェクトがすでにある。
参照先: [セクション 2.2 ioc ファイルを含む SW4STM32 プロジェクトのインポート](#)
- ioc ファイルを含まない SW4STM32 プロジェクトがすでにある。
参照先: [セクション 2.3 ioc ファイルを含まない SW4STM32 プロジェクトのインポート](#)
- サンプルプロジェクトを使用して学習し、調査したい。
参照先: [セクション 2.5 STM32CubeMP1 からのプロジェクトのインポート MCU パッケージ](#)
- 初めての STM32MP1 プロジェクトを開始したい。
 - 空のプロジェクト – STM32CubeMX のサポートなしで柔軟性を最大限にする。
参照先: [セクション 2.4 STM32CubeMP1 内のテンプレートに基づいた空のプロジェクトの作成 MCU パッケージ](#)
 - STM32CubeMP1 プロジェクト – STM32CubeMX で管理されたプロジェクト。
参照先: [セクション 2.1 新しい STM32 プロジェクトの作成](#)

1.3 STM32MP1 シリーズの特有の機能

ターゲットのブート方法は重要です。ブートピンは、ST マイクロエレクトロニクス のボード上でスイッチを使用して設定します。STM32MP157C-EV1 評価ボードに関しては、ユーザマニュアル (UM2535) の「Boot options」セクションで関連情報が提供されています。より一般的には、ST マイクロエレクトロニクス の MPU wiki (wiki.st.com/stm32mpu) で、使用しているボードの「Boot related switches」セクションからも情報を入手できます。

以下の 2 つのブートモードが考えられます。

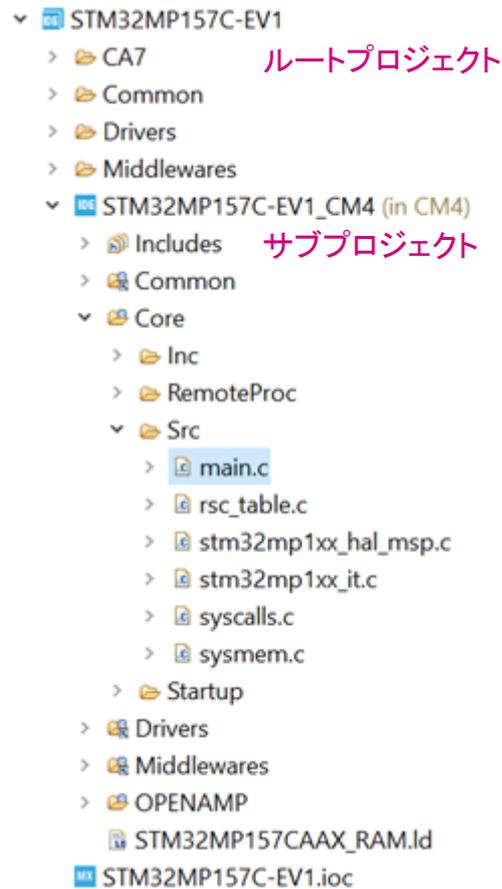
- プロダクションブートモード: Linux® は通常 SD カードでブートしますが、オンボードの NAND または NOR からブートすることも可能です。Cortex®-M4 elf はネットワーク経由でダウンロードされ、OpenAMP フレームワークによってロードされます。実行中のターゲットに接続することにより、JTAG/SWD 経由でアプリケーションをデバッグすることが可能です。
- エンジニアリングブートモード: Cortex®-A7 は事実上無効になり、アプリケーションは JTAG/SWD 経由で Cortex®-M4 に直接ダウンロードされます。このモードを使用すると、アプリケーションは標準的な Cortex®-M4 デバイスと同様にデバッグされます。

プロダクションモードかエンジニアリングモードかの選択によるその他の結果については、[セクション 3.1 デバッグモードと](#)
[セクション 3.2 ターゲットのステータス](#)で詳細に説明します。

1.3.1 STM32MP1 のプロジェクト構造

STM32MP1 プロジェクトを作成すると、その構造は自動的に階層化されます。単一コアプロジェクトのプロジェクト構造は平坦です。反対に、マルチコアプロジェクトでは、階層的なプロジェクト構造が使用されます。STM32MP1 プロジェクトは、作成時またはインポート時に、コアごとに 1 つのルートプロジェクトと複数のサブプロジェクト(マイクロコントローラプロジェクト)で構成されます。階層構造の例を図 1 に示します。

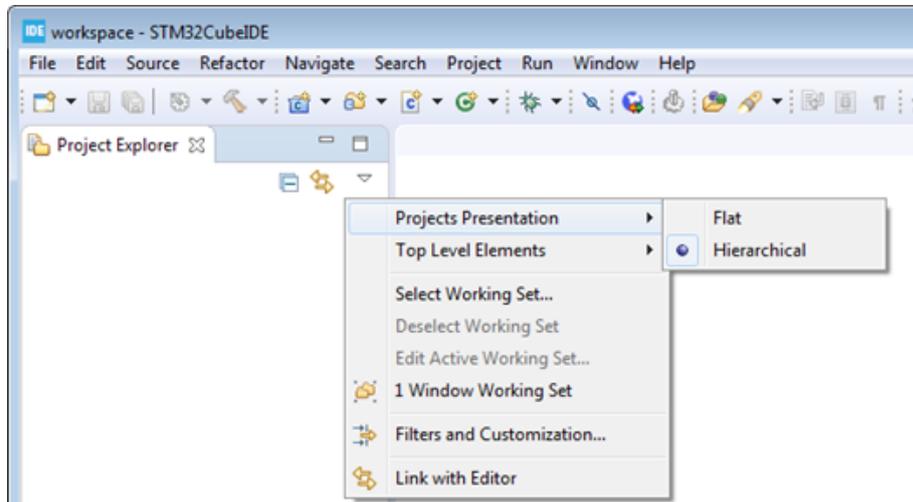
図 1. 階層的なプロジェクト構造



ルートプロジェクトは、コア間での共通コードの共有を可能にする単純なコンテナです。ルートプロジェクトには、ビルド設定もデバッグ設定も含めることができません。ただし、マイクロコントローラプロジェクトは実際の CDT プロジェクトであり、ビルド設定とデバッグ設定の両方を含めることができます。

プロジェクトが階層構造で表示されない場合は、図 2 に示すように変更できます。

図 2. プロジェクトの階層表示の設定



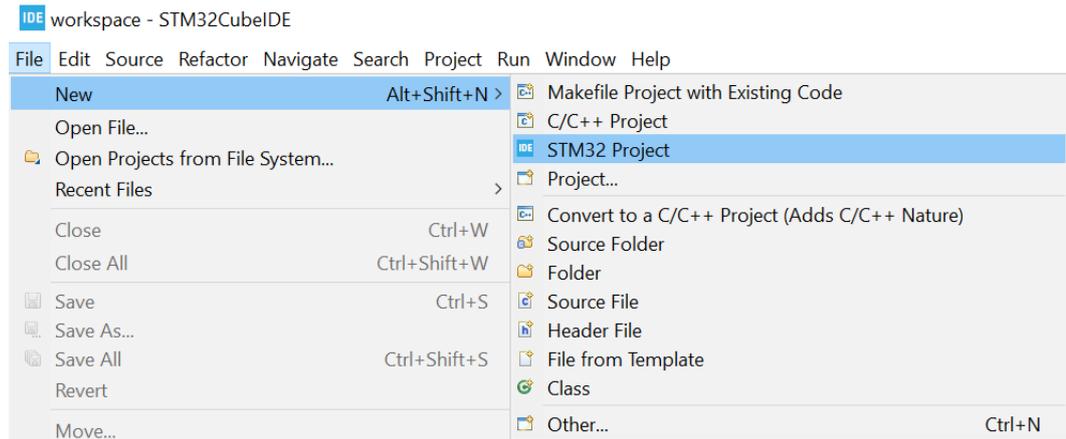
2 プロジェクトの作成とインポート

この章では、STM32MP1 シリーズ のプロジェクトを作成またはインポートする方法について説明します。

2.1 新しい STM32 プロジェクトの作成

新しいプロジェクトを開始するには、図 3 に示すように [File]>[New]>[STM32 Project] に移動します。

図 3. 新しい STM32 プロジェクト



目的のマイクロコントローラまたはボードを選択します。図 4 に示す例では、選択されているボードは STM32MP157C-EV1 です。[Next >] をクリックします。

図 4. ターゲットの選択

Target Selection
Select STM32 target

MCU/MPU Selector | Board Selector | Cross Selector

Board Filters

Part Number Search: STM32MP157C-EV1

Vendor: >

Type: >

MCU/MPU Series: >

Other: >

Price = 399.0

Oscillator Freq. = 0 (MHz)

Peripheral

<input checked="" type="checkbox"/>	Accelerometer	0	0
<input checked="" type="checkbox"/>	Analog I/O	0	0
<input checked="" type="checkbox"/>	Arduino Form Factor	0	0
<input checked="" type="checkbox"/>	Audio Line In	0	1
<input checked="" type="checkbox"/>	Audio Line Out	0	1
<input checked="" type="checkbox"/>	Battery		
<input checked="" type="checkbox"/>	Button	0	0
<input checked="" type="checkbox"/>	CAN	0	0
<input checked="" type="checkbox"/>	Camera	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	Compass		
<input checked="" type="checkbox"/>	Custom Form Factor	0	0
<input checked="" type="checkbox"/>	Digital I/O	0	0
<input checked="" type="checkbox"/>	Ethernet	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	Gyroscope		
<input checked="" type="checkbox"/>	I2C		
<input checked="" type="checkbox"/>	Joystick		
<input checked="" type="checkbox"/>	LCD Display (Graphics)	0	0
<input checked="" type="checkbox"/>	LCD Display (Segment)	0	1
<input checked="" type="checkbox"/>	LED	0	0

Features | Large Picture | Docs & Resources | Datasheet

★ STM32MP157C-EV1

STMicroelectronics STM32MP157C-EV1 Evaluation Board Support and Examples

ACTIVE Active
Product is in mass production

Unit Price (US\$): 399.0
Mounted device: STM32MP157CAAx

The STM32MP157A-EV1 and STM32MP157C-EV1 Evaluation boards are the full-feature demonstration and development platforms for STMicroelectronics Arm-based dual Cortex-A7 32 bits and Cortex-M4 32 bits MPUs in the STM32MP1 Series. They leverage the capabilities of STM32MP1 Series microprocessors to allow users develop applications using STM32 MPU OpenSTLinux Distribution software for the main processor and STM32CubeMP1 software for the co-processor. They include an ST-LINK embedded debug tool, LEDs, push-buttons, one joystick, 1-Gbps Ethernet, CAN FD, and USB OTG High Speed capabilities.

Boards List: 1 item

Overview	Part No	Type	Marketing Status	Unit Price (US\$)	Mounted Device
	STM32MP157C-EV1	Evaluation Board	Active	399.0	STM32MP157CAAx

< Back | **Next >** | Finish | Cancel

ターゲットを選択すると、図 5 に示すプロジェクトのセットアップ手順が表示されます。Targeted Project Type 設定は、プロジェクトが STM32CubeMX によって生成されるかどうかを決定します。Empty プロジェクトはビルドが必要なプロジェクトのスケルトンであり、一方で STM32Cube は STM32CubeMX で管理されるプロジェクトを示します。

図 5. プロジェクトのセットアップ

IDE STM32 Project

Project Setup

Setup STM32 project

Project Name: STM32MP157C-EV1

Use default location

Location: C:/Users/girdlanm/STM32CubeIDE/workspace_1.1.0 Browse...

Options

Targeted Language

C C++

Targeted Binary Type

Executable Static Library

Targeted Project Type

STM32Cube Empty

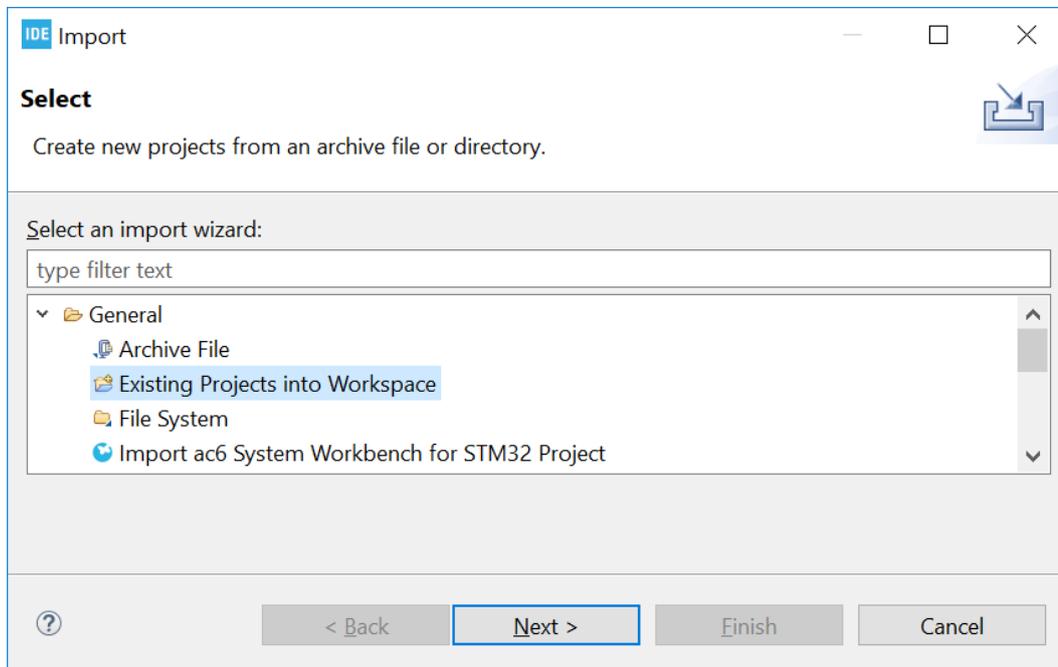
? < Back Next > Finish Cancel

2.2 ioc ファイルを含む SW4STM32 プロジェクトのインポート

プロジェクトに ioc ファイルがすでに含まれている場合、プロジェクトを作業中の STM32CubeIDE 環境にインポートするための最も簡単な方法は、プロジェクトをコピーして STM32CubeMX スタンドアロンで開き、その後 Project Manager で Toolchain / IDE を STM32CubeIDE に変更してプロジェクトを再生成することです。

プロジェクトが再生成されたら、[File]>[Import...] に移動して、図 6 に示すように Existing projects into workspace としてインポートすることを選択します。

図 6. ioc ファイルを含む既存プロジェクトのインポート

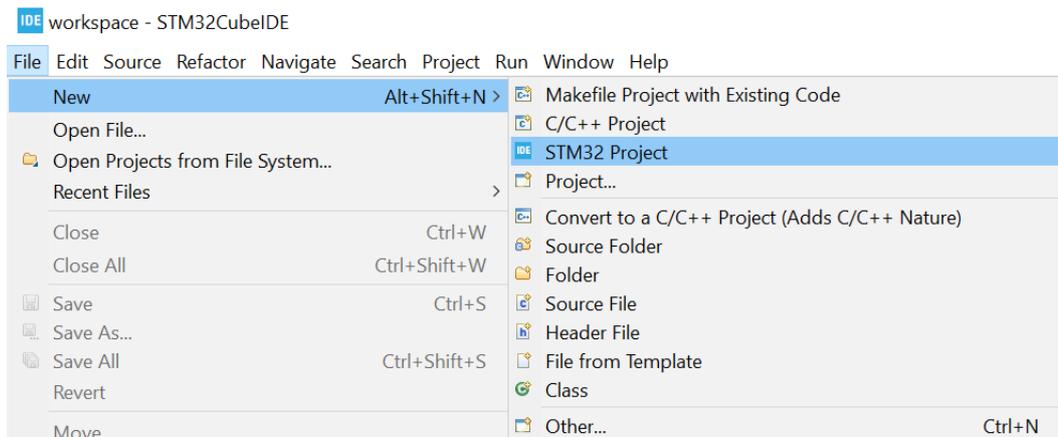


次に、プロジェクト内に存在するさまざまな `/* USER CODE BEGIN */` ブロック内のコードを新しい STM32CubeIDE 環境にコピーします。

2.3 ioc ファイルを含まない SW4STM32 プロジェクトのインポート

プロジェクトを確実に階層構造にするために推奨される方法は、図 7 に示すように [File]>[New]>[STM32 Project] に移動することです。

図 7. 新しい STM32 プロジェクト



インポートするプロジェクト用のデバイスを選択して [Next >] をクリックします。

図 8 に示すようにプロジェクトを設定するときに、Targeted Project Type が Empty に設定されていることを確認して [Finish] をクリックします。

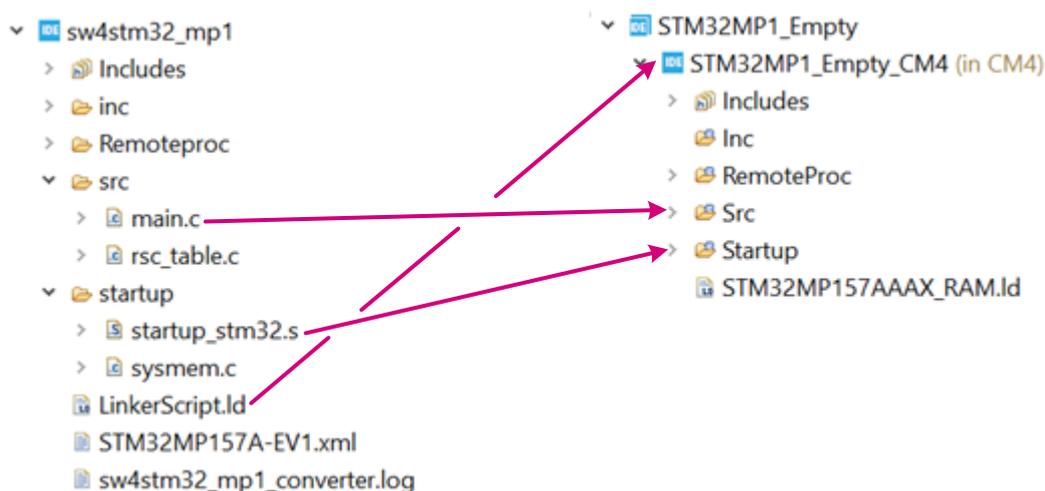
図 8. プロジェクトのセットアップ

The screenshot shows the 'Project Setup' dialog box for an STM32 project. The window title is 'IDE STM32 Project'. The main title is 'Project Setup' and the subtitle is 'Setup STM32 project'. The 'Project Name' field contains 'MP1_Empty'. The 'Use default location' checkbox is checked. The 'Location' field contains 'C:/Users/girdlanm/STM32CubeIDE/workspace' and has a 'Browse...' button. Under the 'Options' section, there are three groups of radio buttons: 'Targeted Language' with 'C' selected, 'Targeted Binary Type' with 'Executable' selected, and 'Targeted Project Type' with 'Empty' selected. At the bottom, there are buttons for '?', '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

空の階層プロジェクトが生成されたら、以下の手順を実行します。

1. [File]>[Import...] に移動します。
2. SW4STM32 プロジェクトを Import ac6 System Workbench for STM32 Project としてインポートします。
3. 図 9 に示すように、プロジェクトエクスプローラを使用して、プロジェクトの内容をコピーして空のプロジェクトのサブプロジェクトに貼り付けます。

図 9. プロジェクトの内容を空のサブプロジェクトにコピー



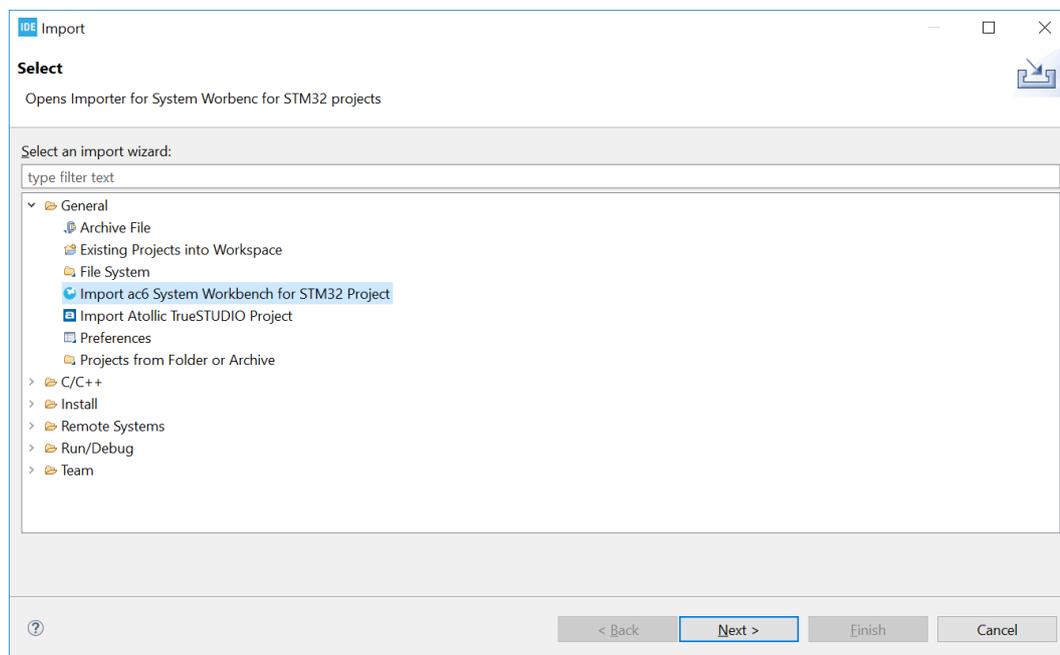
注 .cproject、.project、.settings のいずれかのファイルをインポートすることは推奨されません。プロジェクトが SW4STM32 環境にあったときに使用されていたものと同じビルド設定も忘れずに行うことが重要です。リンクされたリソースがプロジェクトに含まれている場合は、ファイルシステム内の正しいリソースを指すように更新する必要があります。このプロセスは必須です。特殊な処理がなく、ioc ファイルを含まない SW4STM32 からプロジェクトをインポートすると、平坦なプロジェクト構造の STM32CubeIDE にインポートされるからです。

2.4 STM32CubeMP1 内のテンプレートに基づいた空のプロジェクトの作成 MCU パッケージ
セクション 2.3 と同じ手順に従いますが、STM32CubeMP1 MCU パッケージ 内の STM32Cube_FW_MP ファームウェアを入力として使用します。

2.5 STM32CubeMP1 からのプロジェクトのインポート MCU パッケージ

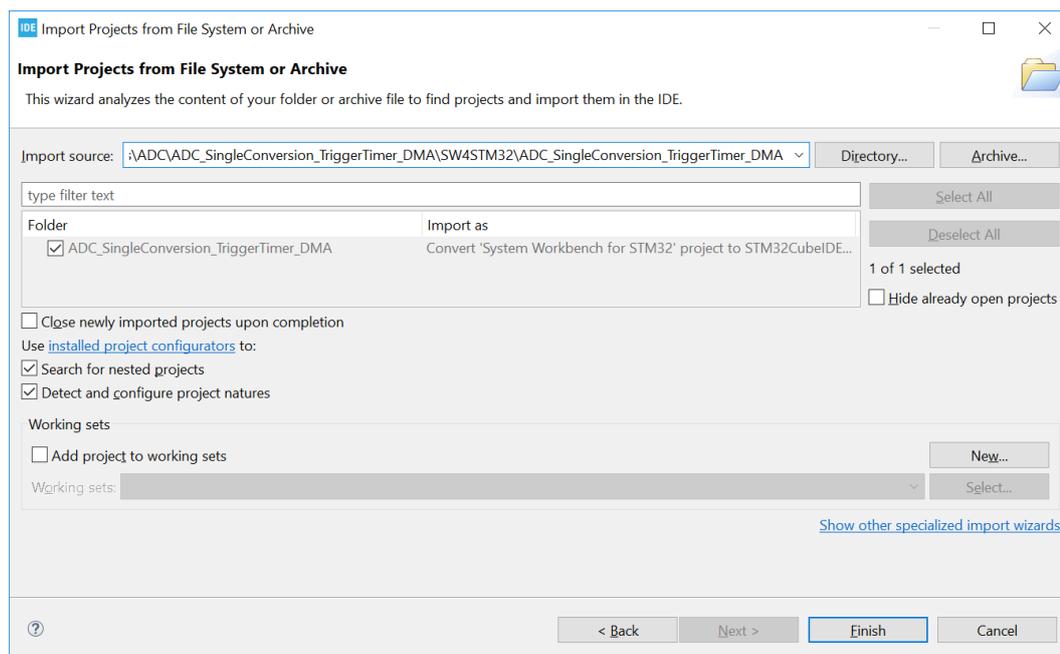
STM32Cube ファームウェアプロジェクトを STM32CubeIDE にインポートするには、[File]>[Import] に移動して Import ac6 System Workbench for STM32 Project を選択し(図 10 を参照)、[Next >] をクリックします。

図 10. STM32CubeIDE へのファームウェアプロジェクトのインポート



次に正しいプロジェクトを選択します。プロジェクトの例は、デフォルトでは `$HOME\STM32Cube\Repository\STM32Cube_FW_MP1_VX.X.X\Projects\STM32MP157C-EV1\Examples\ADC\ADC_SingleConversion_TriggerTimer_DMA\SW4STM32\ADC_SingleConversion_TriggerTimer_DMA` にあります。

図 11. ファームウェアプロジェクトの選択



プロジェクトを選択したら、[Finish] をクリックしてプロジェクトをインポートしてビルドします。

3 デバッグ

この章では、STM32MP1 シリーズでデバイスをデバッグする際に留意する点をいくつか提示します。

3.1 デバッグモード

STM32MP1 シリーズでデバイスをデバッグするには、プロダクションモードとエンジニアリングモードの 2 つのモードがあります。

プロダクションモード

プロダクションモードでは、Cortex[®]-A7 を含めることにより、マイクロコントローラの潜在能力をフルに活用します。以下の点を考慮することにより、Cortex[®]-A7 を使用してアプリケーションを駆動しながら、実行時の重要なタスクで Cortex[®]-M4 を使用可能にすることができます。

- プロダクションモードを有効にするには、ボード上のスイッチを正しく設定する必要があります。ST マイクロエレクトロニクスの MPU wiki (wiki.st.com/stm32mpu) で、使用しているボードの「Boot related switches」セクションを参照してください。STM32MP157C-EV1 評価ボードに関しては、ユーザマニュアル(UM2535)の「Boot options」セクションで関連情報が提供されています。
- ファームウェアは内蔵の Linux[®] ファイルシステムにダウンロードされてから、remoteproc フレームワークを介して Cortex[®]-M4 にアップロードされます。Cortex[®]-M4 コアは Linux[®] によって起動されるため、実行中のターゲットに接続する際のデバッグセッションの早期の起動フェーズを監視する方法はありません。アプリケーションの起動フェーズを監視する必要がある場合、1 つの可能性は、レジスタ値に基づいてビジー待機ループを含めるように Cortex[®]-M4 アプリケーションの起動コードを変更し、デバッグセッションを介して Cortex[®]-M4 を解放するための解放値をレジスタに対して手動で設定することです。
- ターゲットはネットワークに接続されている必要があり、Linux[®] が実行されている必要があります。接続が稼働していることを調べるには、ステータスライトが緑であり、IP アドレスが提示されていることを確認します(セクション 3.2 を参照)。
- このモードでは、Cortex[®]-A7 Linux[®] コアが Cortex[®]-M4 にコマンドを与えます。

エンジニアリングモード

- エンジニアリングモードを有効にするには、ボード上のスイッチを正しく設定する必要があります。ST マイクロエレクトロニクスの MPU wiki (wiki.st.com/stm32mpu) で、使用しているボードの「Boot related switches」セクションを参照してください。STM32MP157C-EV1 評価ボードに関しては、ユーザマニュアル(UM2535)の「Boot options」セクションで関連情報が提供されています。
- Cortex[®]-A7 がループに入り、Cortex[®]-M4 が通常の STM32 デバイスとしてデバッグされ、そこにデバッグ接続を使用してアプリケーションがロードされます。

3.2 ターゲットのステータス

プロダクションモードでは、STM32CubeIDE ウィンドウの右下にあるステータスライトによって、コンピュータと組み込み Linux[®] システムの間の接続の現在のステータスに関する情報が提供されます。

注 シリアルコンソールは共有リソースであり、ターゲットのウィジェットのステータスの優先順位は最低です。シリアルポート用のアクティブなコンソールビューがある場合は、シリアルポートがこのビューから切断されるまでの間、ターゲットのステータスの更新が妨げられます。

ターゲットのステータスライトのさまざまな値を表 1 に示します。

表 1. ターゲットのステータスライト

ステータスライト	アイコン	説明
黒	Stopped 	ライトが完全にオフの場合は、ウィジェットが無効であることを意味します。
赤	Status: offline 	STM32CubeIDE は接続を確立できず、ターゲットを検出できません。
黄	Serial console in use 	以下のような機能障害を示します。 1. コンピュータと MPU の間にネットワーク接続がない。 2. コンソールが開かれている。
緑	Status: idle 	接続され稼動しています。

3.3 シリアルコンソール

シリアルコンソールを開くには、このアイコンをクリックします  を参照してください。

いかなる場合も、単一のターゲットに対してアクティブなシリアル接続は 1 つに限られます。Remote System Explorer のパースペクティブを使用すると、ターゲットに対する 2 つ目の接続を SSH 経由で行うことができます。シリアルコンソール内でまだ実行中のアプリケーションがある状態でシリアルコンソールが閉じられた場合、そのアプリケーションは引き続き実行されます。その後新しいシリアルコンソールが開かれても、実行中のままとなります。この場合、ターゲットの IP アドレスを更新する必要があるときに、まだ実行中のアプリケーションが原因となって Target Widget Status への干渉が発生します。

改版履歴

表 2. 文書改版履歴

日付	版	変更内容
2019 年 10 月 29 日	1	初版発行

目次

1	一般情報	2
1.1	前提条件	2
1.2	本書内のユースケース	2
1.3	STM32MP1 シリーズの特有の機能	2
1.3.1	STM32MP1 のプロジェクト構造	3
2	プロジェクトの作成とインポート	5
2.1	新しい STM32 プロジェクトの作成	5
2.2	ioc ファイルを含む SW4STM32 プロジェクトのインポート	7
2.3	ioc ファイルを含まない SW4STM32 プロジェクトのインポート	8
2.4	STM32CubeMP1 内のテンプレートに基づいた空のプロジェクトの作成 MCU パッケージ	10
2.5	STM32CubeMP1 からのプロジェクトのインポート MCU パッケージ	11
3	デバッグ	12
3.1	デバッグモード	12
3.2	ターゲットのステータス	12
3.3	シリアルコンソール	13
	改版履歴	14
	表一覧	16
	図一覧	17

表一覧

表 1.	ターゲットのステータスライト.....	13
表 2.	文書改版履歴.....	14

図一覧

図 1.	階層的なプロジェクト構造	3
図 2.	プロジェクトの階層表示の設定	4
図 3.	新しい STM32 プロジェクト	5
図 4.	ターゲットの選択	6
図 5.	プロジェクトのセットアップ	7
図 6.	ioc ファイルを含む既存プロジェクトのインポート	8
図 7.	新しい STM32 プロジェクト	8
図 8.	プロジェクトのセットアップ	9
図 9.	プロジェクトの内容を空のサブプロジェクトにコピー	10
図 10.	STM32CubeIDE へのファームウェアプロジェクトのインポート	11
図 11.	ファームウェアプロジェクトの選択	11

重要なお知らせ(よくお読み下さい)

STMicroelectronics NV およびその子会社(以下、ST)は、ST 製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前に ST 製品に関する最新の関連情報を必ず入手してください。ST 製品は、注文請書発行時点で有効な ST の販売条件に従って販売されます。

ST 製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関して ST は一切の責任を負いません。

明示又は黙示を問わず、ST は本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件で ST 製品が再販された場合、その製品について ST が与えたいかなる保証も無効となります。

ST および ST ロゴは STMicroelectronics の商標です。ST の登録商標については ST ウェブサイトをご覧ください。www.st.com/trademarks その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

© 2022 STMicroelectronics – All rights reserved