



IAR
SYSTEMS

IAR
DEVCON

Armv8-Mセキュアマイコンプログラミングテクニック

技術チーム / 殿下 信二



このセッションの目的



- Armv8-Mセキュアマイコンの使い方の基礎を学ぶ
 - ✓ Cortex-MマイコンとArmv8-Mセキュアマイコンの違い
 - ✓ 簡単です、Armv8-Mセキュアマイコンプログラミング

なぜセキュアマイコンが必要ですか？



- 製品が偽造・模造・過剰生産されるリスクの低減
- IoT製品のメリット(コネクティビティ)を最大限活用
- 攻撃を防御するためには、プロセッサからソフトウェアにいたる、システム全体の対策が必要

Armv8-Mセキュアマイコンとは？

- Armv8-Mマイコンは、2種類に分類

Armv6-Mアーキテクチャ

Armv8-Mベースライン



Armv7-Mアーキテクチャ

Armv8-Mメインライン

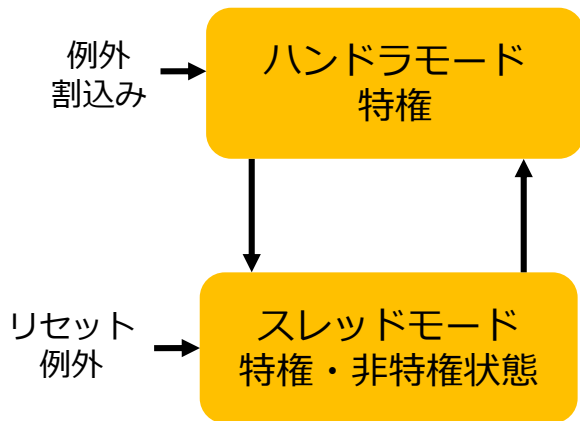


- TrustZone技術を備えたシステム全体に渡るセキュリティ機能の提供
- セキュアステートと非セキュアステートによる保護
- セキュリティデバッグ機能提供

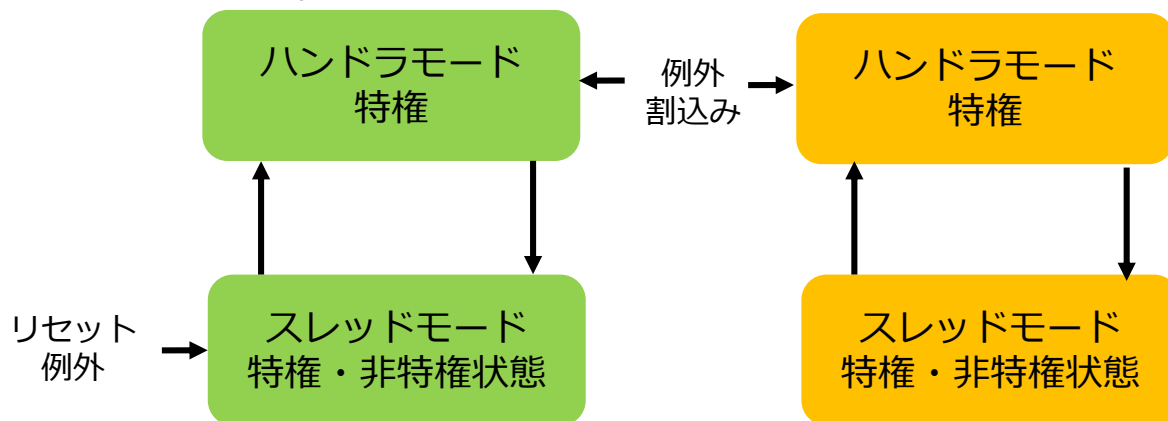
プロセッサモードの追加

- Armv7-M(Cortex-M3/M4)マイコンは、スレッドモード(非特権)とハンドラモード(特権)
- Armv8-Mセキュアマイコンは、**非セキュアステート**と**セキュアステート**が追加

Armv7-M(Cortex-M3/M4)

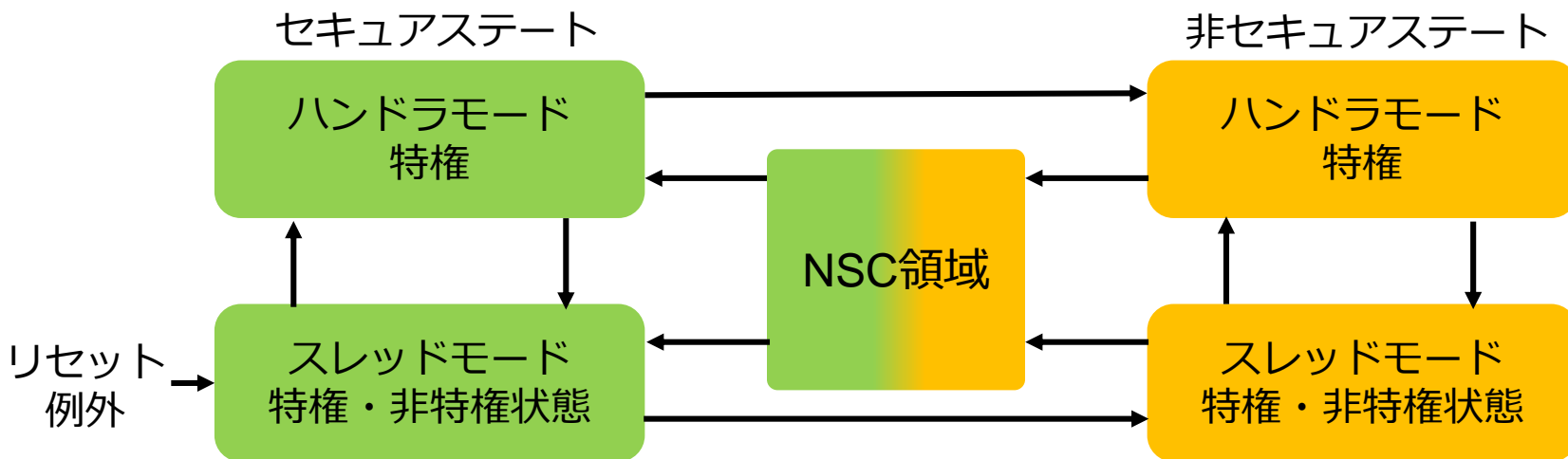


Armv8-Mセキュリティオプション(TrustZone)搭載時
セキュアステート
非セキュアステート



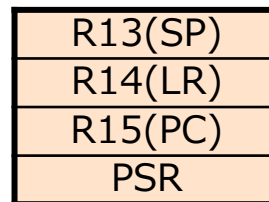
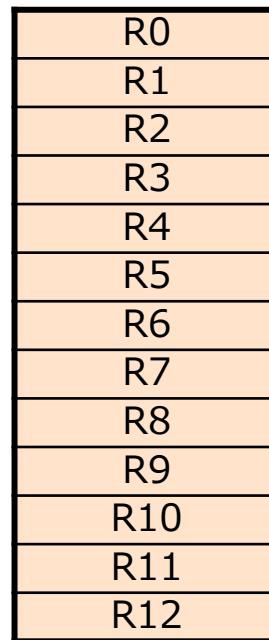
セキュアステートの移行

- セキュアステートの移行は、関数呼び出し
- 非セキュアステートからセキュアステートへ遷移は、NSC(非セキュアコール可能)領域のSG命令(セキュアゲートウェイ命令)を必ず最初に行うことが条件

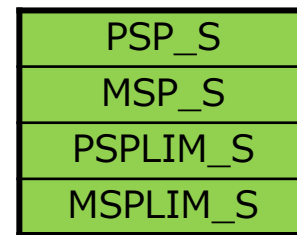


汎用レジスタ構成の違い

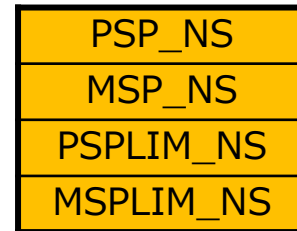
- セキュアステートと非セキュアステートでレジスタがバンク化
- レジスタR0-R7
 - ✓ すべての命令がアクセス
- レジスタR8-R12
 - ✓ 一部16ビット命令がアクセスと32ビット命令がアクセス
- R13はスタック・ポインタ (SP)
 - ✓ セキュアステートでバンク化
- R14はリンク・レジスタ (LR)
- R15はプログラム・カウンタ (PC)



セキュアステート

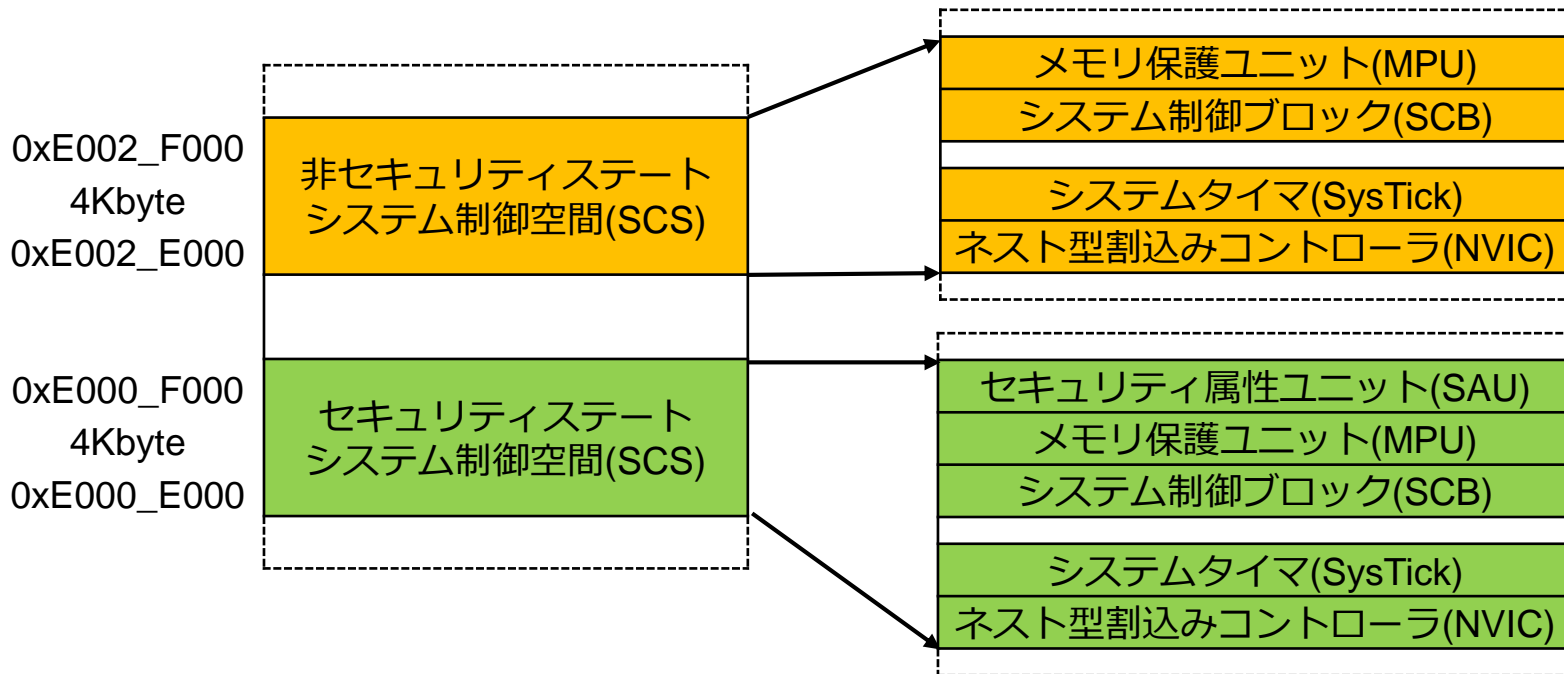


非セキュアステート



システム制御空間(SCS)

- システム制御レジスタは、セキュアステート毎に設定



始まりは、リセットから...でも


- 例外ベクタは、セキュアステート毎に設定

例外番号	セキュアステート	非セキュアステート	優先度	セキュアステート
16...	割込み#0 - N	割込み#0 - N	設定可能	設定可能
15	SysTick	SysTick	設定可能	バンク化
14	PendSV	PendSV	設定可能	バンク化
12	デバッグモニタ	デバッグモニタ	設定可能	設定可能
11	SVCall	SVCall	設定可能	バンク化
7	セキュアフォールト	-	設定可能	セキュアステート
6	用法フォールト	用法フォールト	設定可能	バンク化
5	バスフォールト	バスフォールト	設定可能	設定可能
4	メモリ管理フォールト	メモリ管理フォールト	設定可能	バンク化
3	-	非セキュアハードフォルト	-1	非セキュアステート
3	セキュアハードフォルト	-	-3または-1	セキュアステート
2	NMI	-	-2	設定可能
1	リセット	-	-4	セキュアステート

※:Armv8-Mメインライン (TrustZone搭載時)

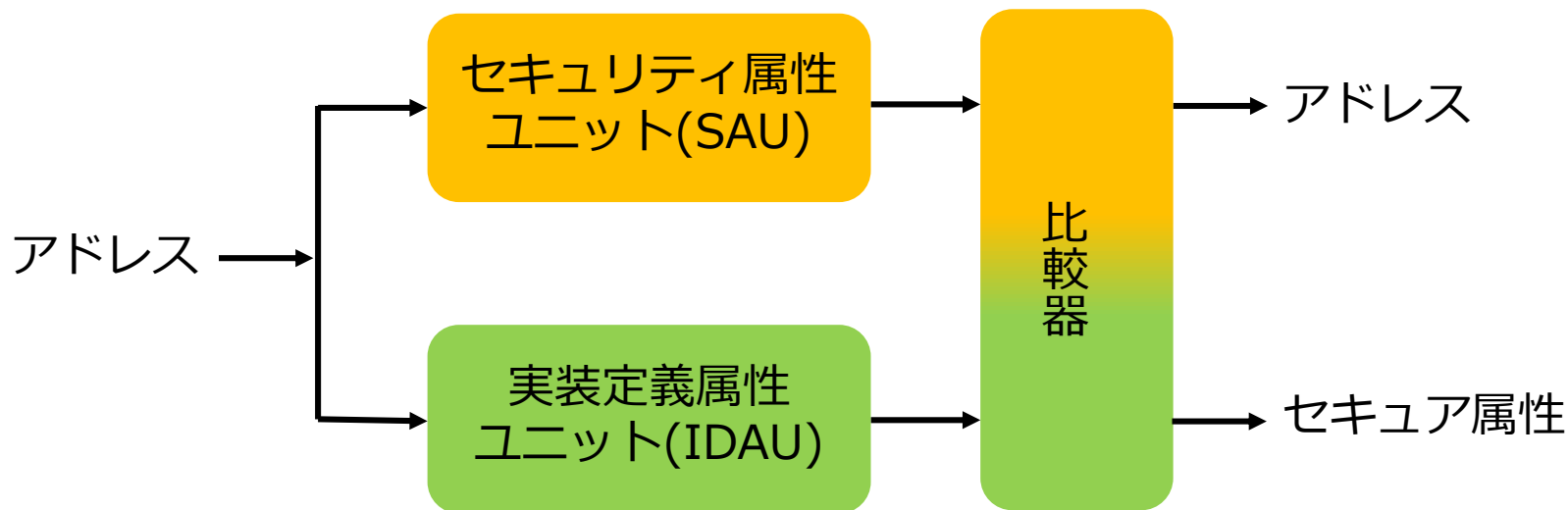
セキュアステート毎のメモリアクセス

- セキュアステートは、セキュアアドレス空間アクセス
- 非セキュアステートは、非セキュアアドレス空間アクセス
- NSC (非セキュアコール可能)領域は、SG (セキュアゲートウェイ)命令を実行する場合に限り、非セキュアステートからの実行可能

ステート	セキュア アドレス空間	NSC領域	非セキュア アドレス空間
セキュアステート	○	○	×
非セキュアステート	×		○

セキュアステートの設定方法

- セキュリティ属性ユニット(SAU)と実装定義属性ユニット(IDAU)で選択された、最も高いセキュア属性を使用



セキュリティを意識したソフトウェアで 必要な事は？



- セキュアステートの処理は小さく、最低限！
- セキュアステート関数コールは、最低限！

セキュアステート

高信頼性RTOSなど
(ハンドラモード：特権)

高信頼性アプリケーション
(スレッドモード：非特権)

非セキュアステート

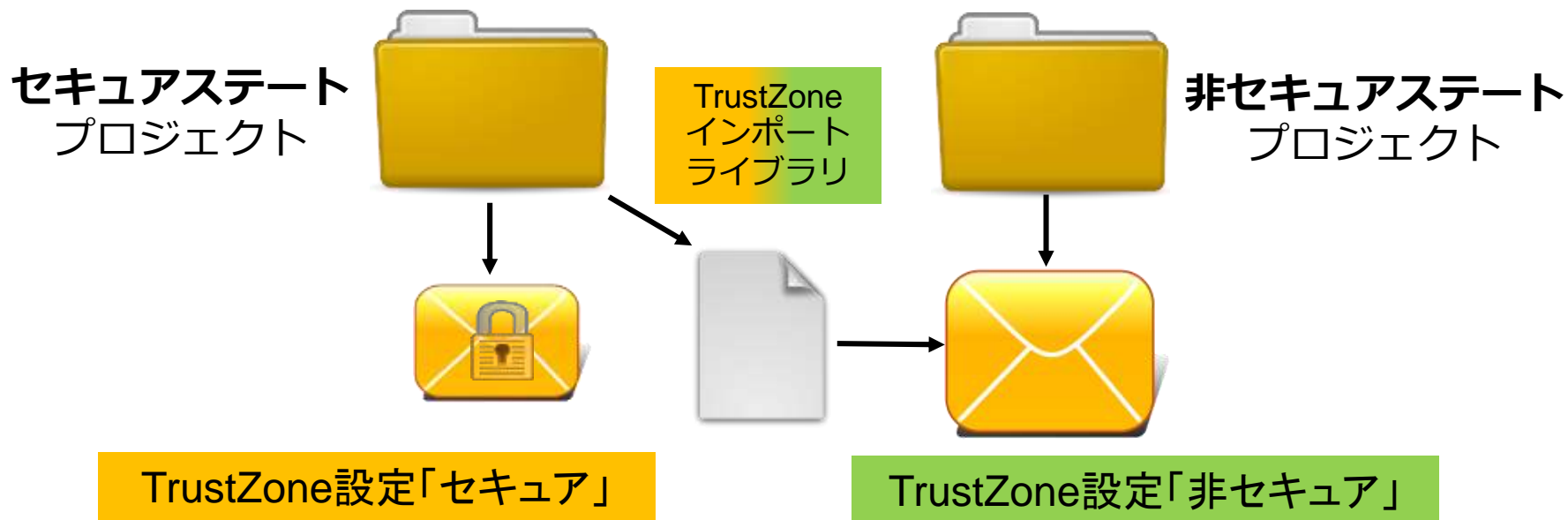
RTOSなど
(ハンドラモード：特権)

アプリケーション
(スレッドモード：非特権)

統合開発環境IAR Embedded Workbench® for Arm で開発する場合は？



- セキュアステート毎にアプリケーションを作成



非セキュアステートから セキュアステートの関数呼び出しは

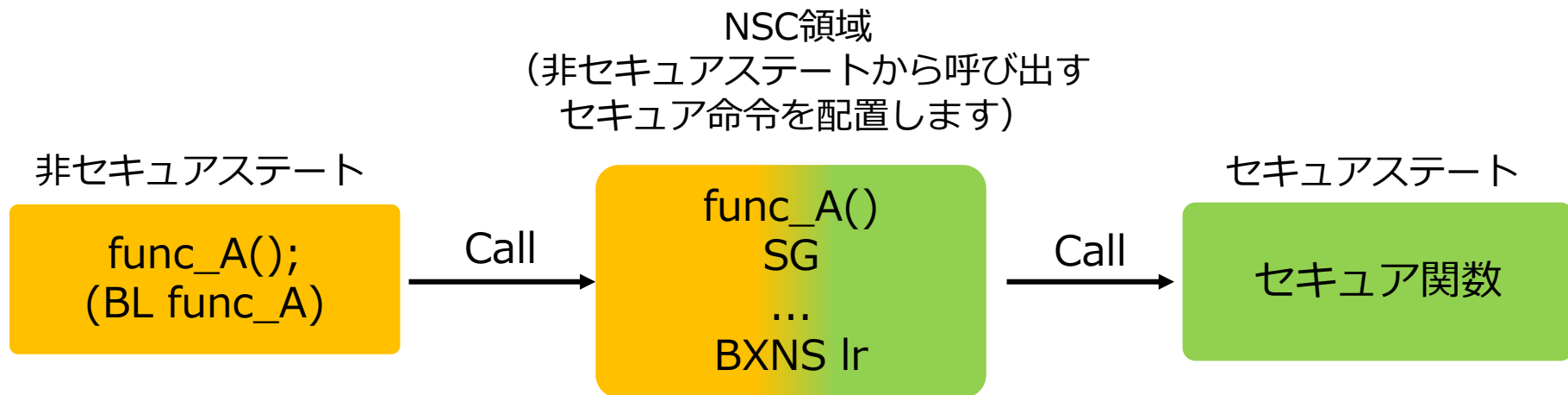


- 非セキュアステートからセキュアステート関数の実行時、セキュアステート関数の配置情報が必要
- セキュアステートプログラムをビルド時に、**TrustZoneインポートライブラリ**が作成
- 非セキュアステートは、**TrustZoneインポートライブラリ**で、セキュアステート関数を利用



セキュリティステートの呼び出しは？

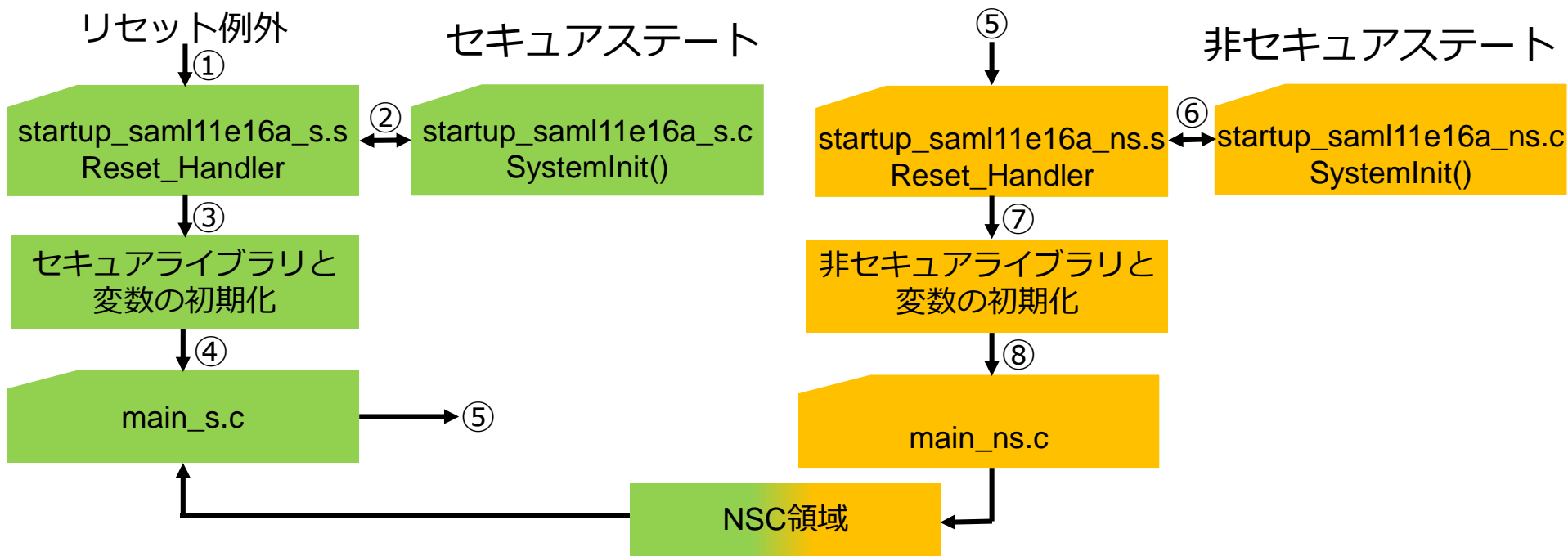
- 非セキュアステートからセキュアステートで実行は、関数に「`__cmse_nonsecure_entry`」を設定
- コンパイラがNSC (非セキュアコール可能)領域にSG (セキュアゲートウェイ)命令を最初に実行するコード生成



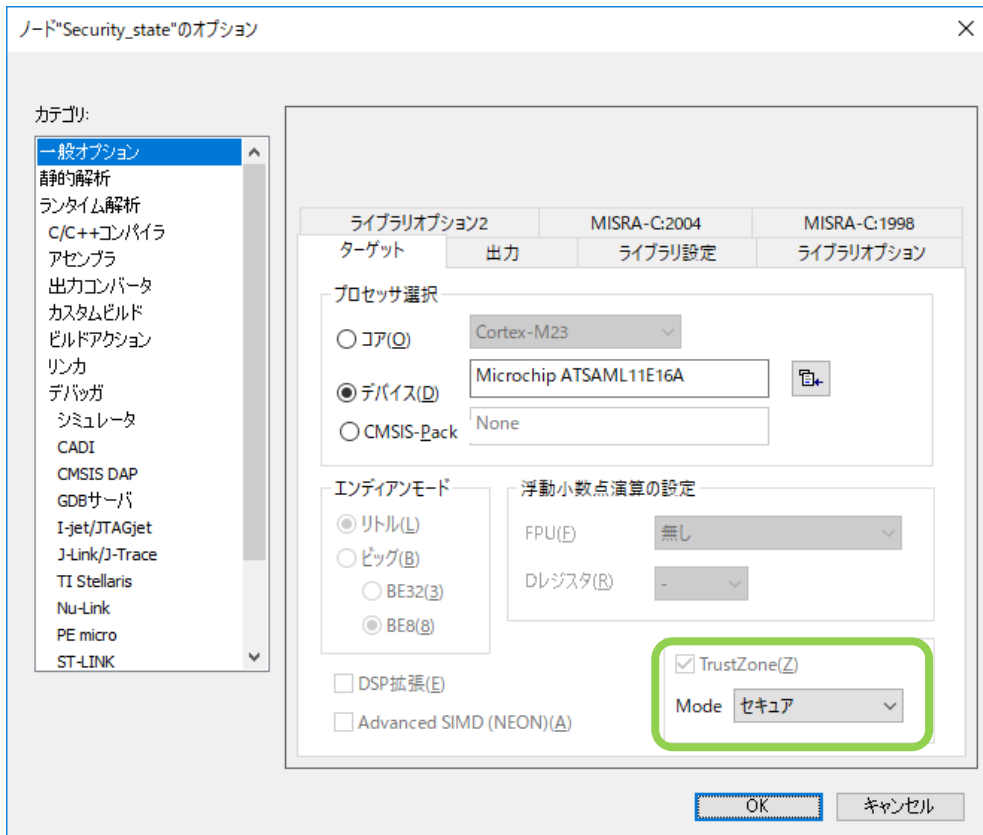
セキュリティを意識した起動シーケンス



- セキュアステートのリセット例外から実行開始

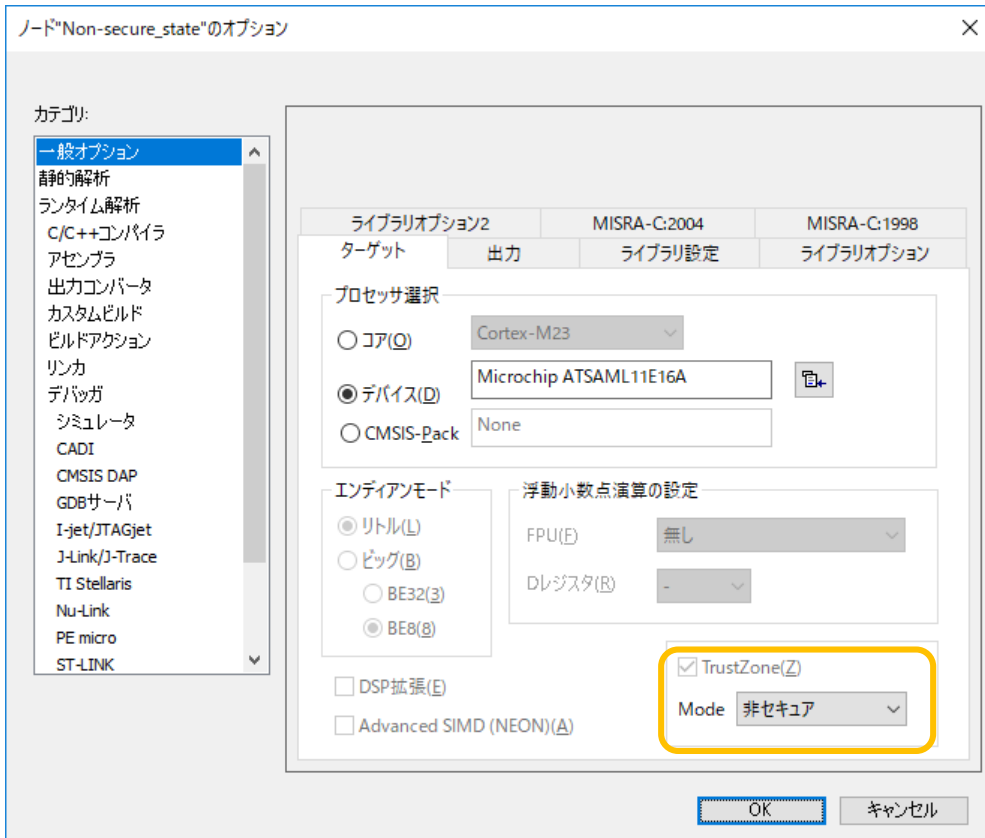


セキュアステートアプリケーション設定



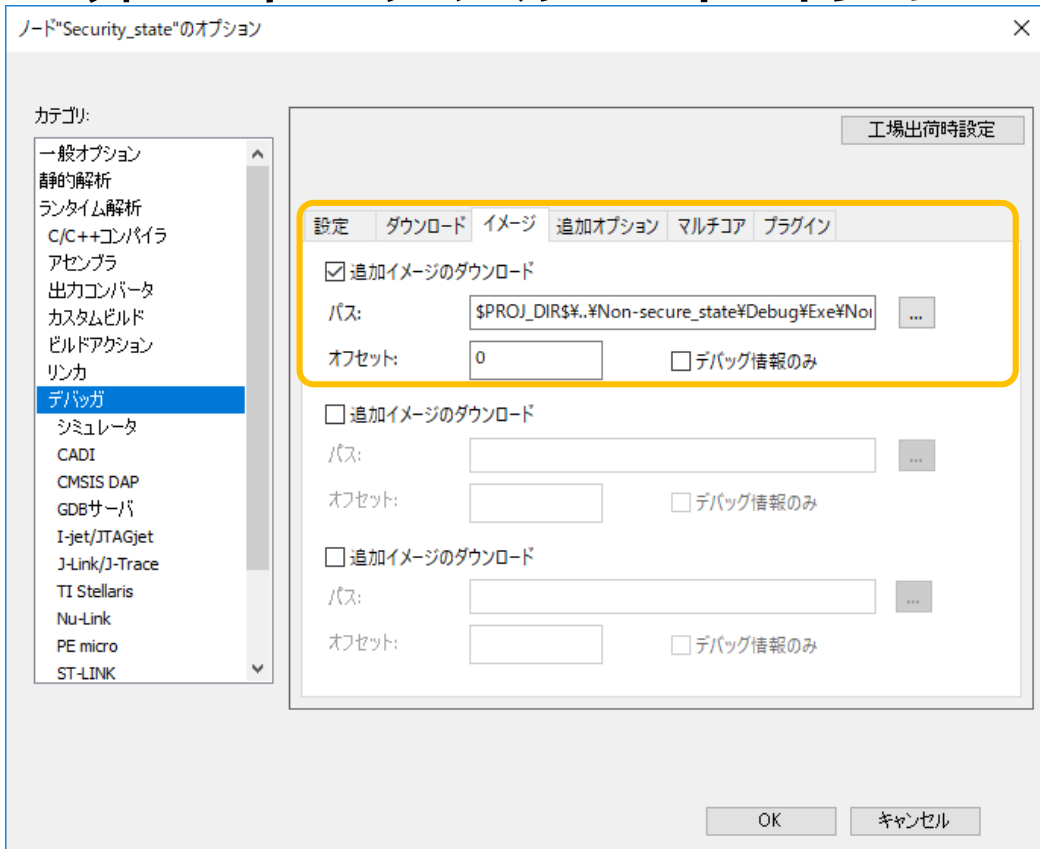
- アプリケーションプログラムのセキュア属性は、「オプション」設定で設定
- 「一般オプション」 - 「ターゲット」を選択
- TrustZone(Z)の「セキュア」に設定

非セキュアステートアプリケーション設定



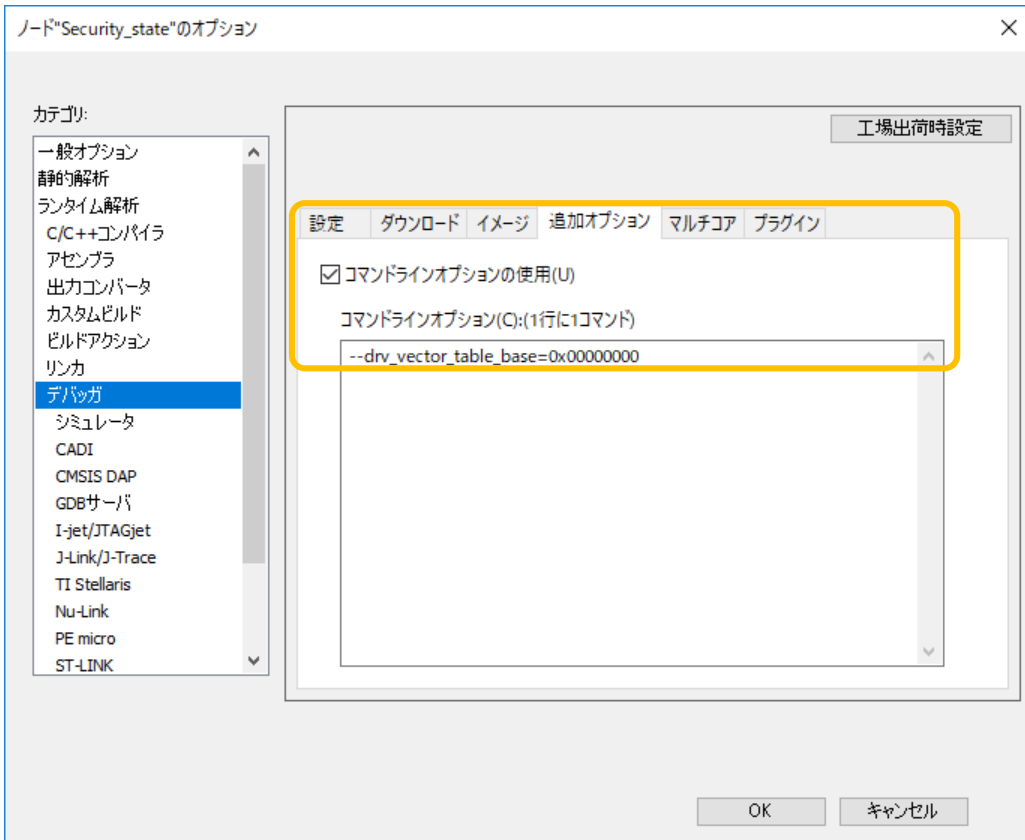
- アプリケーションプログラムのセキュア属性は、「オプション」設定で設定
- 「一般オプション」 - 「ターゲット」を選択
- TrustZone(Z)の「非セキュア」に設定

非セキュアステートオブジェクト選択



- 「デバッガ」 - 「イメージ」 を選択
- 「追加イメージのダウンロード」を有効にし、**非セキュアステートのオブジェクト**を選択

例外ベクタ選択



- 「デバッガ」 - 「追加オプション」を選択
- 「コマンドラインオプションの使用 (U)」を有効にし、「コマンドラインオプション」に、「**--drv_vector_table_base=0x00000000**」を設定

デバッグの開始は、セキュアステート



- セキュアステートからデバッグを開始

The screenshot displays the IAR Embedded Workbench IDE interface for the ATSAM111E16A project. The main window shows assembly code for the 'Security_state - Debug' section. The code includes various handlers and initialization routines, with the 'Reset_Handler' function highlighted in green. The assembly code is as follows:

```
79      DCD      TC2_Handler           ; 36 Basic Timer Counter
80      DCD      ADC_OTHER_Handler    ; 37 Analog Digital Converter
81      DCD      ADC_READY_Handler    ; 38 Analog Digital Converter
82      DCD      AC_Handler           ; 39 Analog Comparators
83      DCD      DAC_EMPTY_Handler    ; 40 Digital Analog Converter
84      DCD      DAC_EMPTY_Handler    ; 41 Digital Analog Converter
85      DCD      PTC_Handler          ; 42 Peripheral Touch Controller
86      DCD      TRNG_Handler         ; 43 True Random Generator
87      DCD      TRAM_Handler         ; 44 TrustRAM
88
89      ; リセットハンドラ
90
91      THUMB
92      SECTION ,text:CODE:ROOT(2)    ; 4 bytes aligned
93
94      Reset_Handler
95      LDR      R0, =svstestinit
96      BLX     R0
97      LDR      R0, =_main
98      BLX     R0
99
100     PUBWEAK NMI_Handler
101     B       NMI_Handler
102
103     PUBWEAK HardFault_Handler
104     B       HardFault_Handler
105
106
```

The right-hand pane shows the 'Registers' window, displaying the state of various registers. The 'Reset_Handler' register is highlighted in green, showing its value as 0x40000000. Other registers shown include SYSTEM_CLOCK, SYSTEMCORECLOCK, and SYSTEMCORECLOCK.

The bottom pane shows the 'Log' window, displaying the following debug messages:

```
05
Thu Nov 01, 2018 14:55:54 Debug resources: 4 instruction comparators, 2 data watchpoints.
Thu Nov 01, 2018 14:55:54 ソフトウェアが実行されました。
Thu Nov 01, 2018 14:55:54 LowPowerResetSoftwareDelay=200
Thu Nov 01, 2018 14:55:54 Ultra LowPowerModeSetup: SoftwareReset
Thu Nov 01, 2018 14:55:54 リセットハンドラ
Thu Nov 01, 2018 14:55:54 INFO: Configuring trace using 'AutotraceInject=100%' setting.
Thu Nov 01, 2018 14:55:54 INFO: CoreView: current when J-link is not powering the target.
Thu Nov 01, 2018 14:55:54 MultiCore: Synchronous core execution DISABLED.
```

まとめ



- セキュリティ対策は、製品開発全体で考える事が必要で、セキュアマイコンの導入はその中で必須の1ステップ
- セキュアステートと非セキュアステートを活用したプログラミングスキルが必要
- 今後の組込み開発には、セキュリティの実装を行うことが必要不可欠

セキュアマイコンの開発を体験！



- 展示コーナーでは、最新版IAR Embedded Workbench® for Arm(EWARM)とMicrochip社製SAML11(Cortex-M23)プロセッサのデモを用意
- 最新のArmv8-Mシリーズに対応
 - ✓ STマイクロエレクトロニクス社製
 - STM32L5シリーズ(Cortex-M33)
 - ✓ NXP社製
 - LPC5500/i.MX RT600シリーズ(Cortex-M33)
 - ✓ Nuvoton社製
 - M2351(Cortex-M23)

IAR DevCon Tokyo

#IARdevcon