

---

# AUTOSARメソドロジーに従った車載 アプリケーションソフトウェア開発

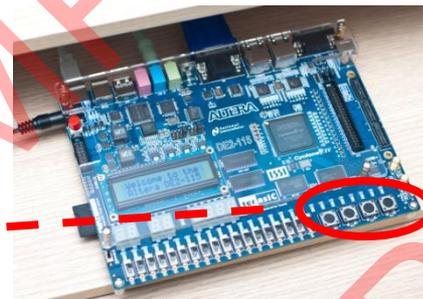
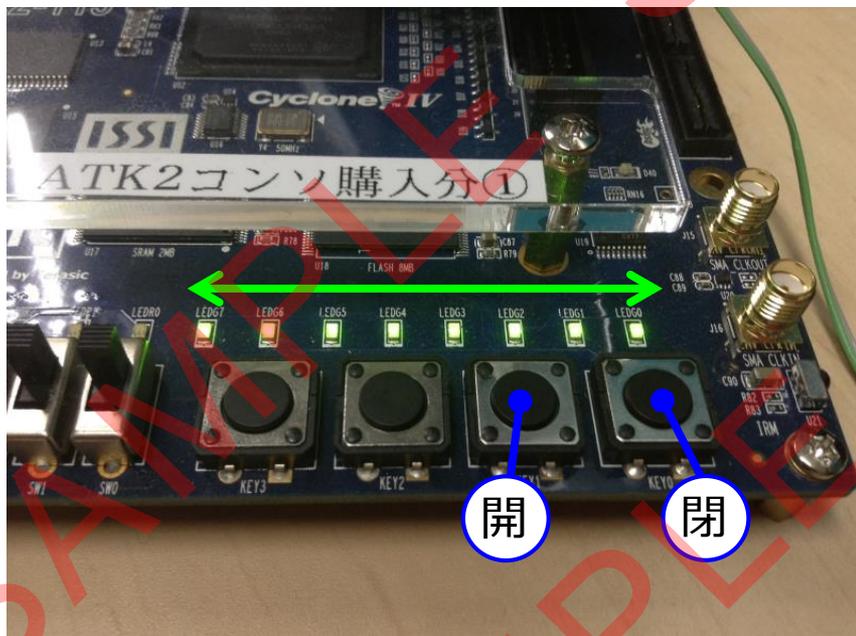
# 目的

---

- メソッドロジや仕組みは理解したものの、実際にAUTOSARを使って開発を行うイメージが分かりにくいいため、車載システムを意識したサンプルアプリケーションを想定し、AUTOSARメソッドロジに従って、実際にアプリケーションを開発してみる
- 車載システムの基本は、以下の周期動作である
  - センサからデータを読み取る
  - 何らかの計算を行う
  - 計算結果を元にアクチュエータを制御する
- 車載システムを構成するセンサ、アクチュエータを用意するのは困難であるので、模擬するものとして、評価ボード上のスイッチやLEDを、センサ、アクチュエータと見なす

# 題材とするアプリケーション

## パワーウィンドウを模したシステム



DE2-115

- 窓の開き具合をLEDで模擬
- 開閉スイッチをKEYで模擬
- RTEとOSのみを使用する



# OSのみで実現する場合

```
#define LED_G_BASE 0x08000420U
#define KEY_BASE 0x08000400U
#define KEY_0 0x01U
#define KEY_1 0x02U

TASK(TASK_500ms) {
    uint32 key;
    boolean up = FALSE;
    boolean down = FALSE;
    static uint8 now = 0;

    key = sil_rew_iop((void *)KEY_BASE);

    if ((key & KEY_0) == 0) {
        up = TRUE;
    }
    if ((key & KEY_1) == 0) {
        down = TRUE;
    }

    if ((down == TRUE) && (now < 0xFF)) {
        now = now << 1;
        now++;
    }
    else if ((up == TRUE) && (now > 0x00)) {
        now = now >> 1;
    }

    sil_wrw_iop((void *)LED_G_BASE, now);

    TerminateTask();
}
```

→ HWに依存したコードが含まれる

→ アラームによりタスクを  
500ms毎に起動

KEY1, KEY2の状態から要求を取得  
※sil\_xxx\_iop(): デバイスレジスタ操作関数

LED表示用データ作成

→ LED表示

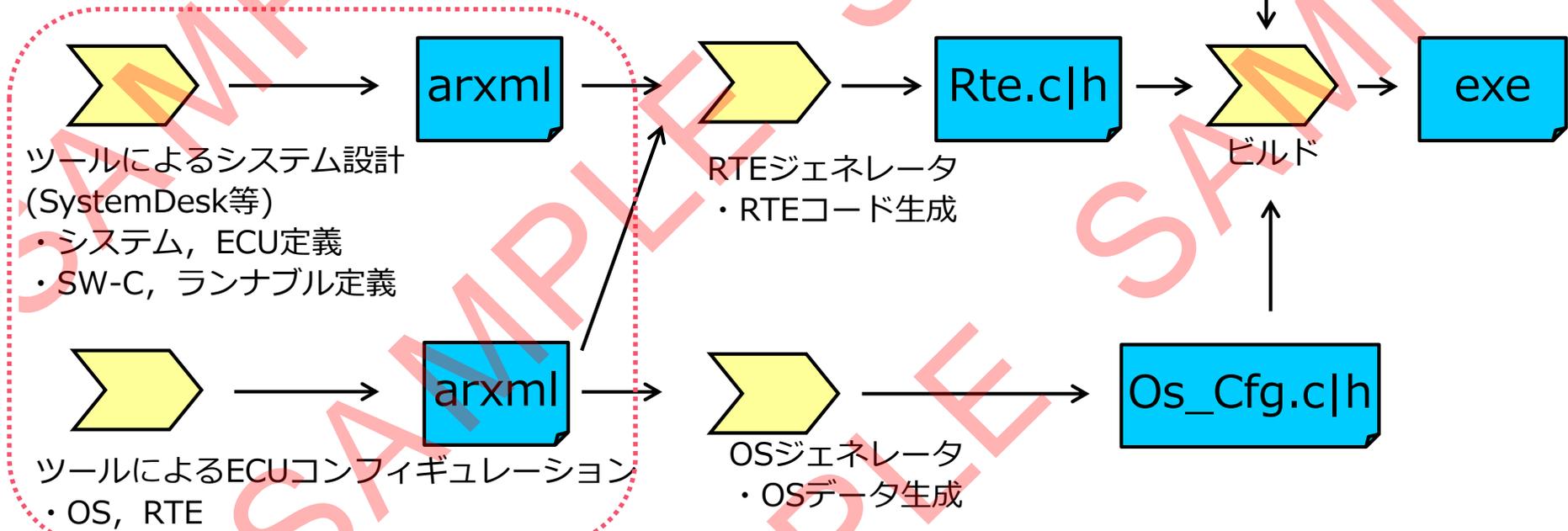
これをAUTOSARで  
実現してみる!

# 開発作業フロー

## <作成担当者>

- BSW開発者
- SW-C開発者
- ECUインテグレータ
- ツール、ジェネレータ

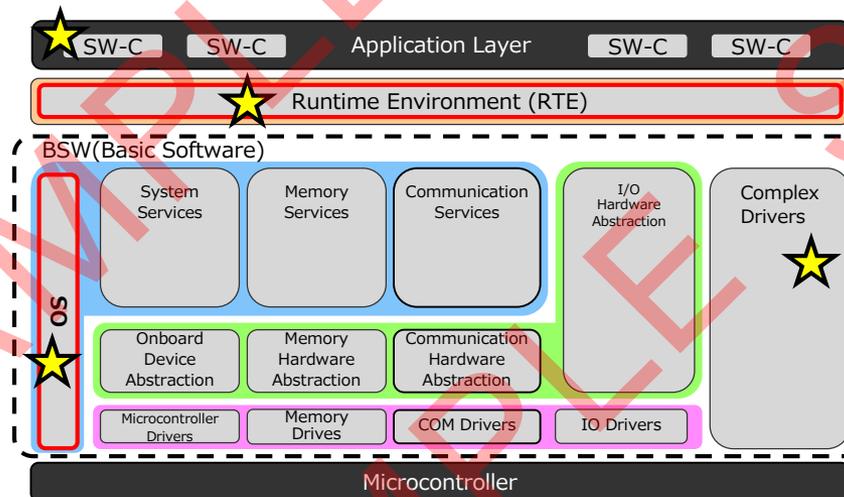
- OSソースコード
- ランナブル実装
- ECUインテグレーションコード (後述)



1つのツール, 1つのarxmlの場合もある

# SW-C設計

- アプリケーションSW-C
  - パワーウィンドウ制御(PowerWindowController)
- センサ/アクチュエータSW-C
  - センサ：ボタン(SensorButton)
- CDD(ComplexDeviceDriver)SW-C
  - デバイス：ボタン(CddButton)
  - デバイス：ウィンドウ(CddWindow)



# SystemDeskによる設計画面

