

# Bài 14: AUTOSAR Classic

Nguyễn Hoàng Anh

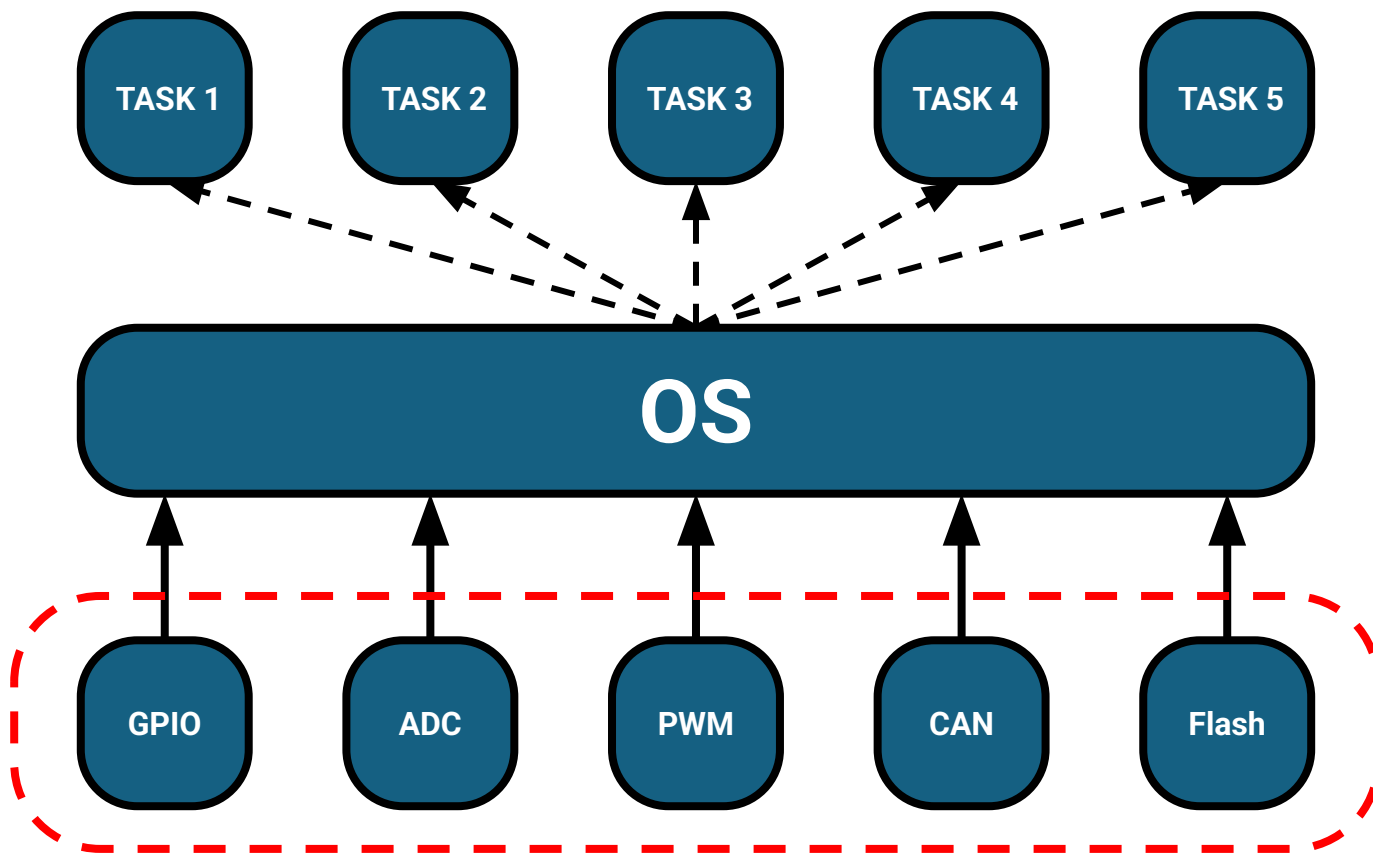
## Mục tiêu của Task1 – Torque Control

Điều chỉnh mô-men xoắn của động cơ dựa trên:

- Khởi động các cảm biến
- Vị trí bàn đạp ga (throttle)
- Tốc độ xe (speed)
- Tải trọng (load)
- So sánh với mô-men thực tế để điều chỉnh

```
// Task 1: Điều khiển mô-men xoắn
uint16_t readThrottleSensor();
SensorData_t readSensorData();
uint16_t calculateTorque(SensorData_t data);
void setTorquePWM(uint16_t torque);
void torqueControlLoop();
```

```
// Task 2: Điều khiển tốc độ động cơ
uint16_t readSpeedSensor();
uint16_t getTargetSpeed();
int16_t compareSpeed(SpeedData_t speedData);
void adjustMotorPWM(int16_t speed_error);
void speedControlLoop();
```



**Phần cứng**

# Khái niệm

**AUTOSAR (AUTomotive Open System ARchitecture)** là một tiêu chuẩn toàn cầu cho phát triển phần mềm ô tô với mục tiêu là chuẩn hóa kiến trúc phần mềm cho các hệ thống điều khiển điện tử (ECU) trong ô tô, nhằm tăng tính khả chuyển, khả mở rộng và giảm chi phí phát triển.

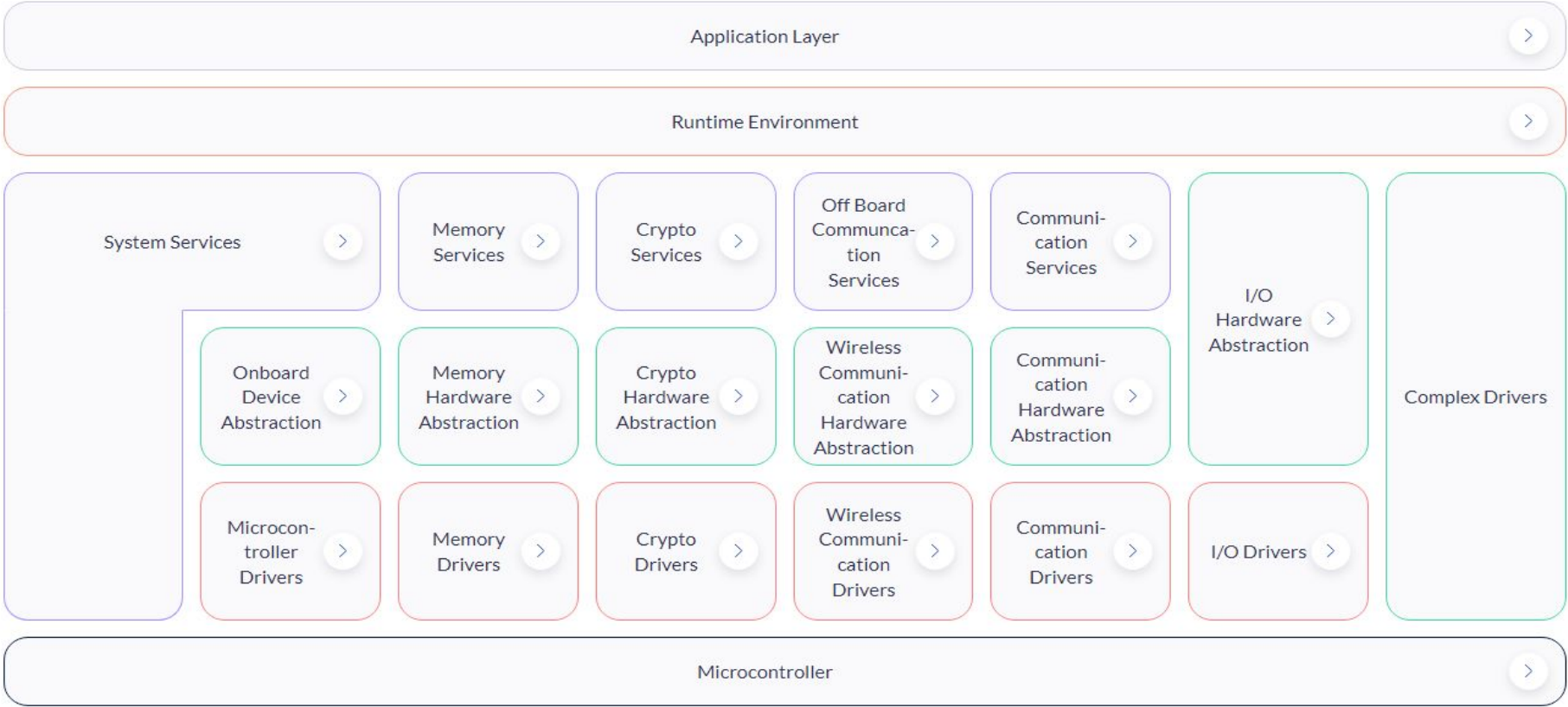
## Không sử dụng AUTOSAR

- ❖ Thiếu sự đồng nhất giữa các task của ECU.
- ❖ Khả năng tái sử dụng thấp.
- ❖ Quản lý lỗi và bảo trì phức tạp.
- ❖ Hệ thống thiếu linh hoạt, mở rộng và phát triển hệ thống khó khăn.
- ❖ Làm việc nhóm khó khăn.

## Có sử dụng AUTOSAR

- ❖ Có sẵn các tiêu chuẩn để dựa vào.
- ❖ Khả năng tái sử dụng phần mềm cao với các dự án khác nhau.
- ❖ Dễ dàng thay đổi để tương thích với các dòng MCU khác nhau.
- ❖ Phần mềm và phần cứng được tách biệt với nhau.
- ❖ Dễ quản lý và bảo trì phần mềm.

# Kiến trúc Autosar Classic



# Kiến trúc Autosar Classic

Kiến trúc phân lớp của AUTOSAR:

- **Application Layer:** Bao gồm các thành phần phần mềm ứng dụng, thực hiện các chức năng cụ thể của xe (như kiểm soát động cơ, phanh, v.v).
- **Runtime Environment (RTE):** Là lớp trung gian giữa phần mềm ứng dụng và phần mềm cơ bản, giúp phần mềm ứng dụng có thể giao tiếp với nhau một cách chuẩn hóa (liên kết SWC và BSW).
- **Basic Software (BSW):** Là phần mềm cơ bản, bao gồm các thành phần phần mềm tiêu chuẩn để quản lý các chức năng hệ thống, giao tiếp và điều khiển phần cứng.

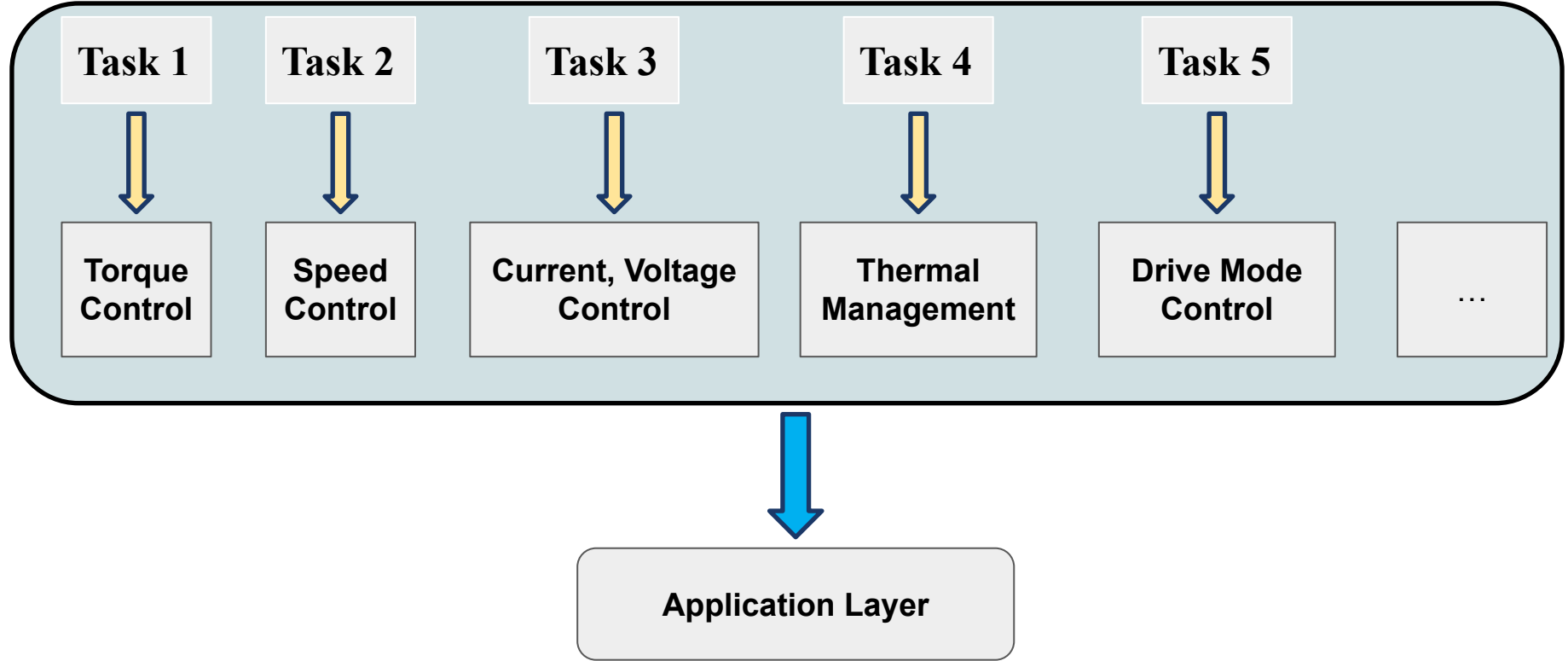
# Application layer

## **Application layer (lớp ứng dụng):**

- Bao gồm nhiều khối phần mềm ứng dụng (Software Component - SWC). Mỗi SWC thực hiện 1 chức năng cụ thể trong hệ thống ECU.
- SWC chỉ quan tâm đến logic (tính toán số học, khởi động thế nào), không cần quan tâm đến phần cứng. Tuy nhiên, SWC vẫn có thể giao tiếp với nhau và giao tiếp với phần cứng thông qua RTE.



# Application layer



# Runtime Environment (RTE)

- **RTE** đóng vai trò quan trọng trong việc kết nối Software Components (SWC) và Basic Software (BSW) thông qua một kiến trúc trừu tượng.
- Một số chức năng của RTE:
  - **Truyền thông tin giữa các SWCs:** RTE cung cấp cơ chế truyền thông để các thành phần phần mềm (SWCs) có thể trao đổi dữ liệu hoặc gọi dịch vụ với nhau mà không cần biết chi tiết về các phần còn lại của hệ thống.
  - **Kết nối với BSW:** RTE cung cấp giao diện để các SWCs có thể tương tác với BSW. Điều này giúp các SWCs có thể sử dụng các dịch vụ hoặc điều khiển phần cứng một cách dễ dàng.
  - Hỗ trợ việc lập lịch và điều phối thực thi của các SWCs theo các sự kiện hoặc chu kỳ định sẵn.

# Một số quy tắc đặt tên API trong RTE

[https://docs.google.com/document/d/1UUhyHBv\\_VDxr7ftPmI7mypk7d7SJ6Bpk\\_OKNKVB6cPY/edit?tab=t.0#heading=h.mvuxkcsis0va](https://docs.google.com/document/d/1UUhyHBv_VDxr7ftPmI7mypk7d7SJ6Bpk_OKNKVB6cPY/edit?tab=t.0#heading=h.mvuxkcsis0va)

# Basic Software (BSW)

- **Basic Software (BSW)** là một trong ba thành phần chính của kiến trúc AUTOSAR, đóng vai trò nền tảng để hỗ trợ phần mềm ứng dụng (SWC) hoạt động trên phần cứng. BSW cung cấp các dịch vụ cơ bản như quản lý phần cứng, giao tiếp, chẩn đoán, và các dịch vụ hệ thống.
- BSW được chia thành 3 lớp chính:
  - **Service Layer.**
  - **Hardware Abstraction Layer.**
  - **Microcontroller Abstraction Layer - MCAL.**

# Basic Software (BSW)

## **System Service:**

Đây là lớp cao nhất trong BSW, cung cấp các dịch vụ hệ thống và tiện ích cho các phần mềm ứng dụng (SWC) và các lớp khác của BSW. Các dịch vụ này bao gồm quản lý thời gian thực, chẩn đoán, quản lý lỗi, quản lý nguồn, v.v.

# Basic Software (BSW)

## **Hardware Abstraction Layer:**

Lớp này cung cấp một giao diện trừu tượng cho tất cả các thiết bị ngoại vi và phần cứng cụ thể của ECU. Nó ẩn đi sự khác biệt về phần cứng của các thiết bị ngoại vi khác nhau và cung cấp một giao diện tiêu chuẩn cho các lớp bên trên (Service Layer và SWC).

# Basic Software (BSW)

## **MCAL:**

Đây là lớp thấp nhất trong BSW, cung cấp giao diện trừu tượng để tương tác trực tiếp với các thành phần phần cứng của vi điều khiển, chẳng hạn như bộ xử lý trung tâm (CPU), các thiết bị ngoại vi tích hợp (như ADC, PWM, UART), và các bộ định thời (timer).

# Tóm tắt

- **SWCs ↔ RTE**

- **SWC** gọi các API được cung cấp bởi **RTE**.
- **RTE**
  - Nhận yêu cầu từ SWC
  - Gửi dữ liệu tới các module dịch vụ của BSW (trong đó có thể đi xuống HAL/MCAL)
  - Cũng nhận dữ liệu từ phần cứng (thông qua HAL/MCAL) và chuyển ngược lên SWC

- **RTE ↔ HAL**

- RTE không giao tiếp trực tiếp với HAL, mà thông qua System Services (trong BSW).
- Khi RTE nhận yêu cầu từ SWC, nó gọi module phù hợp trong System Services, và các service này sẽ sử dụng HAL nếu cần truy cập phần cứng



# Một số quy tắc code trong AUTOSAR Classic

<https://docs.google.com/document/d/166GBPCrXjJhhts93tDavctXgox14eo6wkW-JRMpMOtE/edit?tab=t.0#heading=h.8e77vxa6kmy2>

# Source Code

[https://github.com/HalaAcademy/Autosar-Classic/tree/main/ECU\\_Engine\\_Control\\_Project](https://github.com/HalaAcademy/Autosar-Classic/tree/main/ECU_Engine_Control_Project)