## **Thiết kế Module: Input/Output Hardware Abstraction (IoHwAb)**

### **1. Giới thiệu**

#### **1.1 Mục đích**

Tài liệu này mô tả chi tiết thiết kế và chức năng của module **IoHwAb (Input/Output Hardware Abstraction)** trong lớp **ECU Abstraction Layer** của kiến trúc **AUTOSAR Classic**. Module này cung cấp một giao diện chung giữa phần mềm ứng dụng và phần cứng thông qua các thiết bị như cảm biến, bộ truyền động, đèn LED, công tắc và các thiết bị ngoại vi khác. Mục tiêu là cung cấp một lớp trừu tượng hóa phần cứng giúp cho phần mềm ứng dụng không bị phụ thuộc vào cấu trúc phần cứng cụ thể.

#### **1.2 Phạm vi**

Module **IoHwAb** chịu trách nhiệm quản lý giao tiếp với các thiết bị đầu vào và đầu ra (I/O) trong hệ thống ECU, bao gồm các thiết bị tương tự (Analog) và kỹ thuật số (Digital). Module tương tác với lớp **MCAL (Microcontroller Abstraction Layer)**, nơi chứa các driver phần cứng cụ thể của vi điều khiển, như ADC, PWM, DIO, v.v.

### **2. Kiến trúc và Vị trí của IoHwAb trong AUTOSAR**

#### **2.1 Vị trí của IoHwAb trong Kiến trúc AUTOSAR**

Module **IoHwAb** thuộc lớp **ECU Abstraction Layer** của AUTOSAR, nằm giữa lớp **MCAL** và lớp **Application Layer**. IoHwAb cung cấp các API tiêu chuẩn cho lớp ứng dụng thông qua **Runtime Environment (RTE)**, cho phép truy cập các thiết bị ngoại vi như cảm biến và bộ truyền động mà không cần phải biết chi tiết phần cứng cụ thể.

Cấu trúc lớp:

| +---------------------------------------+ | Application Layer | +---------------------------------------+ | Runtime Environment (RTE) | +---------------------------------------+ | ECU Abstraction Layer (IoHwAb) | | +----------------------------------+ | | | IoHwAb Module | | | +----------------------------------+ | +---------------------------------------+ | Microcontroller Abstraction Layer | | +----------------------------------+ | | | MCAL Drivers (ADC, DIO) | | +---------------------------------------+ | Hardware | +---------------------------------------+ |
| --- |

#### **2.2 Mô tả chức năng chính của IoHwAb**

* **Trừu tượng hóa phần cứng**: Cung cấp giao diện trừu tượng hóa cho phần mềm ứng dụng tương tác với các thiết bị I/O, cho phép phần mềm ứng dụng dễ dàng giao tiếp với phần cứng mà không bị ràng buộc bởi cấu trúc phần cứng cụ thể.
* **Quản lý cảm biến**: Đọc giá trị từ cảm biến (như cảm biến nhiệt độ, cảm biến tải trọng) thông qua các giao diện tiêu chuẩn.
* **Quản lý bộ truyền động**: Điều khiển các thiết bị như đèn LED, relay, động cơ thông qua các driver phần cứng.
* **Chuyển đổi tín hiệu**: Sử dụng các driver MCAL như ADC (Analog-to-Digital Converter) để chuyển đổi tín hiệu tương tự thành tín hiệu số.

### **3. Thiết kế chi tiết Module IoHwAb**

#### **3.1 Cấu trúc Dữ liệu**

**IoHwAb\_ConfigType**

* Mô tả cấu hình chung cho các thiết bị đầu vào/đầu ra được quản lý bởi module IoHwAb. Các cấu trúc dữ liệu này được sử dụng để khởi tạo các thiết bị và lưu trữ thông tin cấu hình.

| typedef struct {  uint8 ChannelId; // Kênh phần cứng của thiết bị  uint16 MaxValue; // Giá trị tối đa của tín hiệu (cho các thiết bị như cảm biến tải trọng) } IoHwAb\_ConfigType; |
| --- |

#### **3.2 Các API chính của IoHwAb**

**IoHwAb\_Init**

* **Mô tả**: Khởi tạo tất cả các thiết bị I/O với cấu hình đã được cung cấp. Cấu hình này bao gồm các thông tin về kênh ADC hoặc DIO, và các thông số hoạt động khác.

**Prototype**:

| void IoHwAb\_Init(const IoHwAb\_ConfigType\* ConfigPtr); |
| --- |

**IoHwAb\_ReadInput**

* **Mô tả**: Đọc giá trị từ một thiết bị đầu vào như cảm biến thông qua ADC.

**Prototype**:

| Std\_ReturnType IoHwAb\_ReadInput(uint8 ChannelId, uint16\* ValuePtr); |
| --- |

**IoHwAb\_WriteOutput**

* **Mô tả**: Ghi giá trị tới một thiết bị đầu ra, chẳng hạn như bật/tắt một đèn LED hoặc điều khiển một động cơ thông qua DIO hoặc PWM.

**Prototype**:

| Std\_ReturnType IoHwAb\_WriteOutput(uint8 ChannelId, uint16 Value); |
| --- |

#### **3.3 Luồng Dữ liệu IoHwAb**

1. **Khởi tạo**:
   * Lớp ứng dụng hoặc RTE gọi hàm IoHwAb\_Init để khởi tạo tất cả các thiết bị đầu vào/đầu ra được cấu hình.
   * Cấu hình bao gồm việc ánh xạ các kênh ADC hoặc DIO tới từng thiết bị ngoại vi (như cảm biến tải trọng, cảm biến nhiệt độ, đèn LED, động cơ).
2. **Đọc giá trị cảm biến**:
   * Ứng dụng gọi hàm IoHwAb\_ReadInput để đọc giá trị từ một cảm biến thông qua ADC.
   * Giá trị ADC sau đó được chuyển đổi thành giá trị tín hiệu thực tế (ví dụ: kg cho cảm biến tải trọng).
3. **Ghi giá trị ra đầu ra**:
   * Ứng dụng gọi hàm IoHwAb\_WriteOutput để điều khiển một thiết bị đầu ra, ví dụ như điều khiển một đèn LED bật/tắt hoặc điều chỉnh độ sáng của đèn.

#### **3.4 Sơ đồ Tương tác API**

| Application Layer IoHwAb MCAL +-------------------+ +-------------------------+ +--------------------+ | | | | | | | IoHwAb\_Init | ---> | IoHwAb\_Init | ---> | Adc\_Init, Dio\_Init | | | | | | | +-------------------+ +-------------------------+ +--------------------+  +-------------------+ +-------------------------+ +--------------------+ | | | | | | | IoHwAb\_ReadInput | ---> | IoHwAb\_ReadInput | ---> | Adc\_ReadChannel | | | | | | | +-------------------+ +-------------------------+ +--------------------+  +-------------------+ +-------------------------+ +--------------------+ | | | | | | | IoHwAb\_WriteOutput| ---> | IoHwAb\_WriteOutput | ---> | Dio\_WriteChannel | | | | | | | +-------------------+ +-------------------------+ +--------------------+ |
| --- |

### **4. Cấu hình IoHwAb**

#### **4.1 Cấu hình compile-time**

Cấu hình **IoHwAb** được định nghĩa trong thời gian biên dịch (compile-time). Cấu hình này bao gồm việc ánh xạ các kênh phần cứng (ADC, DIO) tới các thiết bị I/O cụ thể.

Ví dụ:

| const IoHwAb\_ConfigType IoHwAb\_Configurations[] = {  { .ChannelId = 1, .MaxValue = 1000 }, // Cấu hình cho cảm biến tải trọng  { .ChannelId = 2, .MaxValue = 5 } // Cấu hình cho cảm biến nhiệt độ }; |
| --- |

#### **4.2 Cách sử dụng**

Trong hàm chính của ứng dụng, bạn có thể khởi tạo và sử dụng module **IoHwAb** như sau:

| int main(void) {  IoHwAb\_Init(IoHwAb\_Configurations);   uint16 loadValue;  if (IoHwAb\_ReadInput(1, &loadValue) == E\_OK) {  printf("Load Sensor Value: %d\n", loadValue);  }   IoHwAb\_WriteOutput(2, 1); // Bật đèn LED  return 0; } |
| --- |

### **5. Kiểm thử và Xác minh**

#### **5.1 Kiểm thử**

* **Khởi tạo thiết bị**: Kiểm tra xem tất cả các thiết bị I/O có được khởi tạo đúng cấu hình không.
* **Đọc giá trị cảm biến**: Kiểm tra giá trị đọc từ các thiết bị cảm biến (ví dụ: kiểm tra giá trị ADC từ cảm biến tải trọng).
* **Điều khiển đầu ra**: Kiểm tra việc điều khiển các thiết bị đầu ra như đèn LED hoặc relay có hoạt động đúng không.

#### **5.2 Xác minh**

* Giá trị đầu vào/đầu ra phải được kiểm tra và xác minh thông qua các thiết bị phần cứng thật hoặc thông qua việc giả lập giá trị tín hiệu từ ADC và DIO.

### **6. Yêu cầu và Ràng buộc**

#### **6.1 Yêu cầu Phần cứng**

* **ADC**: Cần thiết để chuyển đổi tín hiệu tương tự từ cảm biến sang tín hiệu số.
* **DIO (Digital Input/Output)**: Cần thiết để điều khiển các thiết bị đầu ra như đèn LED, relay hoặc động cơ.

#### **6.2 Yêu cầu Phần mềm**

* **MCAL Drivers**: Cần có các driver MCAL cho ADC, DIO, PWM để module **IoHwAb** có thể hoạt động đúng chức năng.
* **RTE (Runtime Environment)**: Nếu ứng dụng sử dụng RTE, các hàm API của **IoHwAb** sẽ được gọi thông qua RTE.

#### **6.3 Giới hạn**

* Mỗi thiết bị I/O chỉ được ánh xạ tới một kênh phần cứng (ADC hoặc DIO) duy nhất trong một thời điểm.