### **1. File IoHwAb\_ThrottleSensor.h**

| #ifndef IOHWAB\_THROTTLESENSOR\_H #define IOHWAB\_THROTTLESENSOR\_H  #include "Std\_Types.h"  // Cấu hình cho cảm biến bàn đạp ga typedef struct {  uint8\_t ThrottleSensor\_Channel; // Kênh ADC để đọc giá trị từ cảm biến bàn đạp ga } ThrottleSensor\_ConfigType;  // Prototype cho hàm khởi tạo cảm biến bàn đạp ga Std\_ReturnType IoHwAb\_ThrottleSensor\_Init(const ThrottleSensor\_ConfigType\* ConfigPtr);  // Prototype cho hàm đọc giá trị từ cảm biến bàn đạp ga Std\_ReturnType IoHwAb\_ThrottleSensor\_Read(float\* ThrottlePosition);  #endif /\* IOHWAB\_THROTTLESENSOR\_H \*/ |
| --- |

### **2. File IoHwAb\_ThrottleSensor.c**

| #include "IoHwAb\_ThrottleSensor.h" #include "Adc.h" // Gọi API từ MCAL để đọc giá trị từ ADC #include "Dio.h" // Gọi API từ MCAL để kiểm tra trạng thái DIO nếu cần #include <stdio.h> #include <stdlib.h>  // Địa chỉ hoặc kênh của cảm biến bàn đạp ga sử dụng ADC #define THROTTLE\_SENSOR\_ADC\_CHANNEL 0 // Giả sử cảm biến bàn đạp ga sử dụng kênh ADC 0  // Định nghĩa phạm vi giá trị của cảm biến bàn đạp ga (giá trị thô) #define THROTTLE\_SENSOR\_MIN\_RAW\_VALUE 0 // Giá trị ADC tối thiểu #define THROTTLE\_SENSOR\_MAX\_RAW\_VALUE 1023 // Giá trị ADC tối đa  // Định nghĩa phạm vi giá trị của bàn đạp ga sau khi chuyển đổi #define THROTTLE\_POSITION\_MIN 0.0f // Bàn đạp ga hoàn toàn thả #define THROTTLE\_POSITION\_MAX 1.0f // Bàn đạp ga hoàn toàn nhấn  // Giả lập cấu hình của cảm biến bàn đạp ga static ThrottleSensor\_ConfigType ThrottleSensor\_CurrentConfig;  // Hàm khởi tạo cảm biến bàn đạp ga với cấu hình Std\_ReturnType IoHwAb\_ThrottleSensor\_Init(const ThrottleSensor\_ConfigType\* ConfigPtr) {  if (ConfigPtr == NULL) {  printf("Error: Null configuration pointer passed to IoHwAb\_ThrottleSensor\_Init.\n");  return E\_NOT\_OK;  }   // Lưu cấu hình cảm biến bàn đạp ga vào biến toàn cục  ThrottleSensor\_CurrentConfig.ThrottleSensor\_Channel = ConfigPtr->ThrottleSensor\_Channel;   // Gọi API từ MCAL để khởi tạo ADC  Adc\_ConfigType adcConfig;  adcConfig.Adc\_Channel = ThrottleSensor\_CurrentConfig.ThrottleSensor\_Channel;  Adc\_Init(&adcConfig);   // Gọi API từ MCAL để khởi tạo DIO nếu cần  Dio\_Init();   // In ra thông tin cấu hình của cảm biến bàn đạp ga  printf("Throttle Sensor Initialized with ADC Channel %d\n", ThrottleSensor\_CurrentConfig.ThrottleSensor\_Channel);   return E\_OK; }  // Hàm đọc giá trị bàn đạp ga Std\_ReturnType IoHwAb\_ThrottleSensor\_Read(float\* ThrottlePosition) {  if (ThrottlePosition == NULL) {  return E\_NOT\_OK; // Kiểm tra con trỏ NULL  }   // Đọc giá trị ADC từ kênh cảm biến bàn đạp ga  uint16\_t raw\_adc\_value = 0;  if (Adc\_ReadChannel(ThrottleSensor\_CurrentConfig.ThrottleSensor\_Channel, &raw\_adc\_value) != E\_OK) {  printf("Error: Failed to read ADC value.\n");  return E\_NOT\_OK;  }   // Chuyển đổi giá trị thô của ADC sang phạm vi từ 0.0 đến 1.0  \*ThrottlePosition = ((float)(raw\_adc\_value - THROTTLE\_SENSOR\_MIN\_RAW\_VALUE) /   (THROTTLE\_SENSOR\_MAX\_RAW\_VALUE - THROTTLE\_SENSOR\_MIN\_RAW\_VALUE));   // Đảm bảo giá trị nằm trong phạm vi từ 0.0 đến 1.0  if (\*ThrottlePosition < THROTTLE\_POSITION\_MIN) {  \*ThrottlePosition = THROTTLE\_POSITION\_MIN;  } else if (\*ThrottlePosition > THROTTLE\_POSITION\_MAX) {  \*ThrottlePosition = THROTTLE\_POSITION\_MAX;  }   // In ra giá trị bàn đạp ga sau khi chuyển đổi  printf("Reading Throttle Sensor (ADC Channel %d): Throttle Position = %.2f\n",  ThrottleSensor\_CurrentConfig.ThrottleSensor\_Channel, \*ThrottlePosition);   return E\_OK; } |
| --- |

### **3. File Adc.h (MCAL Layer)**

| #ifndef ADC\_H #define ADC\_H  #include "Std\_Types.h"  // Cấu trúc cấu hình cho ADC typedef struct {  uint8\_t Adc\_Channel; // Kênh ADC } Adc\_ConfigType;  // Prototype cho hàm khởi tạo ADC void Adc\_Init(const Adc\_ConfigType\* ConfigPtr);  // Prototype cho hàm đọc giá trị từ kênh ADC Std\_ReturnType Adc\_ReadChannel(uint8\_t Channel, uint16\_t\* ValuePtr);  #endif /\* ADC\_H \*/ |
| --- |

### **4. File Adc.c (MCAL Layer)**

| #include "Adc.h" #include <stdio.h> #include <stdlib.h>  // Giả lập khởi tạo ADC void Adc\_Init(const Adc\_ConfigType\* ConfigPtr) {  printf("ADC Initialized for Channel %d\n", ConfigPtr->Adc\_Channel); }  // Giả lập đọc giá trị từ kênh ADC Std\_ReturnType Adc\_ReadChannel(uint8\_t Channel, uint16\_t\* ValuePtr) {  // Giả lập giá trị ADC ngẫu nhiên từ 0 đến 1023  \*ValuePtr = rand() % 1024;  printf("ADC Channel %d Read: Value = %d\n", Channel, \*ValuePtr);  return E\_OK; } |
| --- |

### **5. File Dio.h (MCAL Layer)**

| #ifndef DIO\_H #define DIO\_H  #include "Std\_Types.h"  // Prototype cho hàm khởi tạo DIO void Dio\_Init(void);  #endif /\* DIO\_H \*/ |
| --- |

### **6. File Dio.c (MCAL Layer)**

| #include "Dio.h" #include <stdio.h>  // Giả lập khởi tạo DIO void Dio\_Init(void) {  printf("DIO Initialized.\n"); } |
| --- |

### **7. Ví dụ Cấu hình và Sử dụng trong Hệ thống**

#### **File main.c**

| #include "IoHwAb\_ThrottleSensor.h" #include <stdio.h>  int main(void) {  // Cấu hình cảm biến bàn đạp ga  ThrottleSensor\_ConfigType throttleSensorConfig = {  .ThrottleSensor\_Channel = THROTTLE\_SENSOR\_ADC\_CHANNEL // Kênh ADC 0  };   // Khởi tạo cảm biến bàn đạp ga  if (IoHwAb\_ThrottleSensor\_Init(&throttleSensorConfig) != E\_OK) {  printf("Failed to initialize Throttle Sensor.\n");  return -1;  }   // Đọc giá trị bàn đạp ga  float throttlePosition = 0;  if (IoHwAb\_ThrottleSensor\_Read(&throttlePosition) == E\_OK) {  printf("Current Throttle Position: %.2f\n", throttlePosition);  } else {  printf("Failed to read Throttle Sensor.\n");  }   return 0; } |
| --- |

### **8. Giải thích Luồng Hoạt Động**

1. **Khởi tạo cảm biến bàn đạp ga**:
   * Hàm IoHwAb\_ThrottleSensor\_Init được gọi để khởi tạo cảm biến bàn đạp ga. Cấu hình bao gồm kênh ADC được sử dụng để đọc giá trị từ cảm biến.
   * **API Adc\_Init** từ MCAL được gọi để khởi tạo ADC, và **Dio\_Init** cũng được gọi để khởi tạo giao tiếp số (nếu cần thiết).
2. **Đọc giá trị bàn đạp ga**:
   * Khi cần đọc giá trị bàn đạp ga, hàm IoHwAb\_ThrottleSensor\_Read được gọi. **API Adc\_ReadChannel** được sử dụng để đọc giá trị ADC từ kênh đã cấu hình.
   * Giá trị thô từ ADC được chuyển đổi thành giá trị bàn đạp ga với phạm vi từ **0.0** (hoàn toàn thả) đến **1.0** (hoàn toàn nhấn).
3. **Kết quả**:
   * Giá trị bàn đạp ga sẽ được in ra màn hình, cho phép người dùng biết được trạng thái hiện tại của bàn đạp ga.

### **9. Kết quả Mô phỏng**

Khi chạy chương trình trên một hệ thống giả lập, đầu ra có thể như sau:

| ADC Initialized for Channel 0 DIO Initialized. Throttle Sensor Initialized with ADC Channel 0 ADC Channel 0 Read: Value = 780 Reading Throttle Sensor (ADC Channel 0): Throttle Position = 0.76 Current Throttle Position: 0.76 |
| --- |