

```

#include "main.h"

// GPIOポートとピンの設定
#define A_PHASE_PORT GPIOA
#define A_PHASE_PIN GPIO_PIN_15

#define B_PHASE_PORT GPIOB
#define B_PHASE_PIN GPIO_PIN_3

#define C_PHASE_PORT GPIOB
#define C_PHASE_PIN GPIO_PIN_10

// ホールセンサの状態を示すフラグ
volatile uint8_t A_Phase_Flag = 0;
volatile uint8_t B_Phase_Flag = 0;
volatile uint8_t C_Phase_Flag = 0;

// 割り込みハンドラ関数の宣言
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin);

int main(void) {
    // GPIOのクロックを有効化
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    // GPIOの設定
    GPIO_InitTypeDef GPIO_InitStruct;
    GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING_FALLING; // 割り込みを立ち上がりエッジ
    と立ち下がりエッジで検出
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;

    GPIO_InitStruct.Pin = A_PHASE_PIN;
    HAL_GPIO_Init(A_PHASE_PORT, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = B_PHASE_PIN;
    HAL_GPIO_Init(B_PHASE_PORT, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = C_PHASE_PIN;
    HAL_GPIO_Init(C_PHASE_PORT, &GPIO_InitStruct);

    // 割り込み優先度の設定
    HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
    HAL_NVIC_SetPriority(EXTI3_IRQn, 0, 0);
    HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);

    // 割り込み処理の有効化
    HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
    HAL_NVIC_EnableIRQ(EXTI3_IRQn);
    HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);

    while (1) {
        // メイン処理

```

```

    if (A_Phase_Flag) {
        // A相の割り込みが発生した場合の処理
        // ...
        A_Phase_Flag = 0; // フラグをクリア
    }

    if (B_Phase_Flag) {
        // B相の割り込みが発生した場合の処理
        // ...
        B_Phase_Flag = 0; // フラグをクリア
    }

    if (C_Phase_Flag) {
        // C相の割り込みが発生した場合の処理
        // ...
        C_Phase_Flag = 0; // フラグをクリア
    }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
    if (GPIO_Pin == A_PHASE_PIN) {
        // A相の割り込み処理
        if (HAL_GPIO_ReadPin(A_PHASE_PORT, A_PHASE_PIN) == GPIO_PIN_SET) {
            // 立ち上がりエッジの処理
            A_Phase_Flag = 1;
        } else {
            // 立ち下がりエッジの処理
            A_Phase_Flag = 2;
        }
    } else if (GPIO_Pin == B_PHASE_PIN) {
        // B相の割り込み処理
        if (HAL_GPIO_ReadPin(B_PHASE_PORT, B_PHASE_PIN) == GPIO_PIN_SET) {
            // 立ち上がりエッジの処理
            B_Phase_Flag = 1;
        } else {
            // 立ち下がりエッジの処理
            B_Phase_Flag = 2;
        }
    } else if (GPIO_Pin == C_PHASE_PIN) {
        // C相の割り込み処理
        if (HAL_GPIO_ReadPin(C_PHASE_PORT, C_PHASE_PIN) == GPIO_PIN_SET) {
            // 立ち上がりエッジの処理
            C_Phase_Flag = 1;
        } else {
            // 立ち下がりエッジの処理
            C_Phase_Flag = 2;
        }
    }
}

void EXTI15_10_IRQHandler(void) {
    HAL_GPIO_EXTI_IRQHandler(A_PHASE_PIN);
}

```

```
    HAL_GPIO_EXTI_IRQHandler(C_PHASE_PIN);  
}  
  
void EXTI3_IRQHandler(void) {  
    HAL_GPIO_EXTI_IRQHandler(B_PHASE_PIN);  
}
```