

```

#include "main.h"
#include "stm32f3xx_hal_tim.h"
#include "stm32f3xx_hal_adc.h"

// TIMインスタンス
TIM_HandleTypeDef htim1;

// ADCインスタンス
ADC_HandleTypeDef hadc1;

// GPIOポートとピンの設定
#define A_PHASE_PORT GPIOA
#define A_PHASE_PIN GPIO_PIN_8

#define B_PHASE_PORT GPIOA
#define B_PHASE_PIN GPIO_PIN_9

#define C_PHASE_PORT GPIOA
#define C_PHASE_PIN GPIO_PIN_10

#define ENABLE_A_PORT GPIOC
#define ENABLE_A_PIN GPIO_PIN_10

#define ENABLE_B_PORT GPIOC
#define ENABLE_B_PIN GPIO_PIN_11

#define ENABLE_C_PORT GPIOC
#define ENABLE_C_PIN GPIO_PIN_12

// モータードライバ初期化
void motorDriverInit() {
    // GPIOポートのクロック有効化
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOC_CLK_ENABLE();

    // GPIOピンの設定
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_HIGH;

    GPIO_InitStructure.Pin = A_PHASE_PIN;
    GPIO_InitStructure.Alternate = GPIO_AF2_TIM1; // タイマー1をオルタネート機能に割り当て
る
    HAL_GPIO_Init(A_PHASE_PORT, &GPIO_InitStructure);

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

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

    GPIO_InitStructure.Pin = ENABLE_A_PIN | ENABLE_B_PIN | ENABLE_C_PIN;

```

```

GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

// タイマークロックの有効化
__HAL_RCC_TIM1_CLK_ENABLE();
}

// PWM初期化
void pwmInit() {
// タイマー設定
TIM_HandleTypeDef htim1;
htim1.Instance = TIM1;
htim1.Init.Prescaler = 0;
htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
htim1.Init.Period = 19; // PWM周期 (50kHz)
htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
HAL_TIM_PWM_Init(&htim1);

// PWMチャンネル設定
TIM_OC_InitTypeDef sConfigOC;
sConfigOC.OCMode = TIM_OCMODE_PWM1;
sConfigOC.Pulse = 0; // デューティサイクルの初期値 (0%)
sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;

HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_1);
HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_2);
HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_3);

// PWM出力開始
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_2);
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_3);
}

// ADC初期化
void adcInit() {
ADC_ChannelConfTypeDef sConfig;

// ADCクロックの有効化
__HAL_RCC_ADC1_CLK_ENABLE();

// ADC設定
hadc1.Instance = ADC1;
hadc1.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV1;
hadc1.Init.Resolution = ADC_RESOLUTION_12B;
hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;
hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
hadc1.Init.LowPowerAutoWait = DISABLE;
hadc1.Init.LowPowerAutoPowerOff = DISABLE;

```

```

hadc1.Init.ContinuousConvMode = DISABLE;
hadc1.Init.DiscontinuousConvMode = DISABLE;
hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
hadc1.Init.DMAContinuousRequests = DISABLE;
hadc1.Init.Overrun = ADC_OVR_DATA_OVERWRITTEN;
HAL_ADC_Init(&hadc1);

// ADCチャンネル設定
sConfig.Channel = ADC_CHANNEL_0;
sConfig.Rank = ADC_REGULAR_RANK_1;
sConfig.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;
sConfig.SingleDiff = ADC_SINGLE_ENDED;
sConfig.OffsetNumber = ADC_OFFSET_NONE;
sConfig.Offset = 0;
HAL_ADC_ConfigChannel(&hadc1, &sConfig);
}

// ADCから値を読み取る
uint16_t readADC() {
    uint16_t adcValue = 0;

    // ADC変換開始
    HAL_ADC_Start(&hadc1);

    // 変換完了まで待機
    if (HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY) == HAL_OK) {
        // 変換値の読み取り
        adcValue = HAL_ADC_GetValue(&hadc1);
    }

    // ADC変換停止
    HAL_ADC_Stop(&hadc1);

    return adcValue;
}

// 3相のPWMデューティサイクルとモータイネーブルを設定
void setPWMEnable(uint16_t dutyCycleA, uint16_t dutyCycleB, uint16_t dutyCycleC,
GPIO_PinState enableAState, GPIO_PinState enableBState, GPIO_PinState
enableCState) {
    TIM1->CCR1 = dutyCycleA; // チャンネル1のPWMデューティサイクル
    TIM1->CCR2 = dutyCycleB; // チャンネル2のPWMデューティサイクル
    TIM1->CCR3 = dutyCycleC; // チャンネル3のPWMデューティサイクル

    HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, enableAState);
    HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, enableBState);
    HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, enableCState);
}

int main(void)
{
    HAL_Init();

```

```

// モータードライバの初期化
motorDriverInit();

// PWMの初期化
pwmInit();

// ADCの初期化
adcInit();

// メインループ
while (1) {
    // ADCから値を読み取る
    uint16_t adcValue = readADC();

    // PWMデューティサイクルを設定
    uint16_t dutyCycle = (adcValue * 19) / 4095; // ADC値をPWM周期に変換
    setPWMEnable(dutyCycle, dutyCycle, dutyCycle, GPIO_PIN_SET,
GPIO_PIN_SET, GPIO_PIN_SET);

    // 一定時間待機
    HAL_Delay(10);
}
}

```