

```

#include "main.h"
#include "stm32f3xx_hal_tim.h"
#include "stm32f3xx_hal_adc.h"
UART_HandleTypeDef huart2;

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART2_UART_Init(void);
/* USER CODE BEGIN PFP */
// TIMインスタンス
TIM_HandleTypeDef htim1;

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

void motorInit() {
    // GPIOのクロックを有効化
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOC_CLK_ENABLE();

    // GPIO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.Alternate = GPIO_AF6_TIM1;

    GPIO_InitStructure.Pin = A_PHASE_PIN;
    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_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;

GPIO_InitStruct.Pin = ENABLE_A_PIN;
HAL_GPIO_Init(ENABLE_A_PORT, &GPIO_InitStruct);

GPIO_InitStruct.Pin = ENABLE_B_PIN;
HAL_GPIO_Init(ENABLE_B_PORT, &GPIO_InitStruct);

GPIO_InitStruct.Pin = ENABLE_C_PIN;
HAL_GPIO_Init(ENABLE_C_PORT, &GPIO_InitStruct);

GPIO_InitStruct.Pin = GPIO_PIN_5;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

// モータードライバのイネーブルピンをリセット状態に設定
HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_RESET);
HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_RESET);
HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_RESET);
}

void PWM_Init() {
    // タイマーのクロックを有効化
    __HAL_RCC_TIM1_CLK_ENABLE();

    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 0;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 999; // 50kHzの周期に設定
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
    HAL_TIM_PWM_Init(&htim1);

    TIM_OC_InitTypeDef sConfigOC;
    sConfigOC.OCMode = TIM_OCMODE_PWM1;
    sConfigOC.Pulse = 500; // 50%のデューティ比に設定
    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);

    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_2);
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_3);
}

void setPhase(uint8_t phase, uint32_t duty) {
    switch(phase) {

```

```

case 0:
    TIM1->CCR1 = 0;
    TIM1->CCR2 = 0;
    TIM1->CCR3 = 0;
    HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_RESET);
    break;
case 1:
    TIM1->CCR1 = duty;
    TIM1->CCR2 = 0;
    TIM1->CCR3 = 0;
    HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_SET);
    HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_SET);
    HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_RESET);
    break;
case 2:
    TIM1->CCR1 = duty;
    TIM1->CCR2 = 0;
    TIM1->CCR3 = 0;
    HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_SET);
    HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_SET);
    break;
case 3:
    TIM1->CCR1 = 0;
    TIM1->CCR2 = duty;
    TIM1->CCR3 = 0;
    HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_SET);
    HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_SET);
    break;
case 4:
    TIM1->CCR1 = 0;
    TIM1->CCR2 = duty;
    TIM1->CCR3 = 0;
    HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_SET);
    HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_SET);
    HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_RESET);
    break;
case 5:
    TIM1->CCR1 = 0;
    TIM1->CCR2 = 0;
    TIM1->CCR3 = duty;
    HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_SET);
    HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_SET);
    break;
case 6:
    TIM1->CCR1 = 0;
    TIM1->CCR2 = 0;
    TIM1->CCR3 = duty;
    HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_RESET);

```

```

        HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_SET);
        HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_SET);
        break;
    default:
        HAL_GPIO_WritePin(ENABLE_A_PORT, ENABLE_A_PIN, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(ENABLE_B_PORT, ENABLE_B_PIN, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(ENABLE_C_PORT, ENABLE_C_PIN, GPIO_PIN_RESET);
        break;
    }
}
#endif
void ADC_Init() {
    // ADCのクロックを有効化
    __HAL_RCC_ADC1_CLK_ENABLE();
    // GPIOのクロックを有効化
    __HAL_RCC_GPIOB_CLK_ENABLE();

    // GPIOの設定
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.Pin = GPIO_PIN_1;
    GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);

    // ADC設定
    ADC_ChannelConfTypeDef sConfig;
    hadc1.Instance = ADC1;
    hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV1;
    hadc1.Init.Resolution = ADC_RESOLUTION_12B;
    hadc1.Init.ScanConvMode = DISABLE;
    hadc1.Init.ContinuousConvMode = ENABLE;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.NbrOfConversion = 1;
    HAL_ADC_Init(&hadc1);

    // ADCチャンネル設定
    sConfig.Channel = ADC_CHANNEL_9;
    sConfig.Rank = 1;
    sConfig.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;
    HAL_ADC_ConfigChannel(&hadc1, &sConfig);
}
#else
void ADC_Init() {
    // ADCのクロックを有効化
    __HAL_RCC_ADC1_CLK_ENABLE();
    // GPIOのクロックを有効化
    __HAL_RCC_GPIOB_CLK_ENABLE();

    // GPIOの設定
    GPIO_InitTypeDef GPIO_InitStructure;

```

```

GPIO_InitStruct.Pin = GPIO_PIN_2;
GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

// ADC設定
ADC_ChannelConfTypeDef sConfig;
hadc1.Instance = ADC1;
hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV1;
hadc1.Init.Resolution = ADC_RESOLUTION_12B;
hadc1.Init.ScanConvMode = DISABLE;
hadc1.Init.ContinuousConvMode = ENABLE;
hadc1.Init.DiscontinuousConvMode = DISABLE;
hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
hadc1.Init.NbrOfConversion = 1;
HAL_ADC_Init(&hadc1);

// ADCチャンネル設定
sConfig.Channel = ADC_CHANNEL_12;// for PC2
sConfig.Rank = 1;
sConfig.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;
HAL_ADC_ConfigChannel(&hadc1, &sConfig);
}
#endif
// ADC値の取得
uint16_t readADC() {
    // ADC変換開始
    HAL_ADC_Start(&hadc1);

    // 変換完了待ち
    HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);

    // ADC値の取得
    uint16_t adcValue = HAL_ADC_GetValue(&hadc1);

    return adcValue;
}

int main(void)
{
    HAL_Init();

    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_USART2_UART_Init();

    //モータの初期化
    motorInit();
    //PWMの初期化

```

```
PWM_Init();  
//ADCの初期化  
ADC_Init();  
  
while (1) {  
    uint16_t adcValue = readADC();  
    uint32_t duty = (adcValue * 999) / 4095; // ADC値をPWM周期に変換  
    for (uint8_t i = 0; i <= 6; i++) {  
        setPhase(i,duty);  
        HAL_Delay(2); // デレイ時間を適宜調整  
    }  
}  
}
```