```
#include "main.h"
#include "stm32f3xx_hal_tim.h"

UART_HandleTypeDef huart2;

void SystemClock_Config(void);
#define LED_PIN     GPIO_PIN_5
#define LED_PORT    GPIOA

// TIMインスタンス
TIM_HandleTypeDef htim1;

int main(void)
{
        uint16_t adcValueOld=0;
        uint16_t adcValue;
        uint16_t dutyCycle;

  HAL_Init();

  SystemClock_Config();
  // GPIOポートのクロックを有効化
  __HAL_RCC_GPIOA_CLK_ENABLE();

  // GPIOの設定
  GPIO_InitTypeDef GPIO_InitStruct;
  GPIO_InitStruct.Pin = LED_PIN;
  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
  GPIO_InitStruct.Alternate = GPIO_AF1_TIM2; // TIM2のAlternate Functionを選択
  HAL_GPIO_Init(LED_PORT, &GPIO_InitStruct);

  // TIM2のクロックを有効化
  __HAL_RCC_TIM2_CLK_ENABLE();

  // TIM2の設定
  TIM_HandleTypeDef htim2;
  htim2.Instance = TIM2;
  htim2.Init.Prescaler = 1600 - 1;         // プリスケーラの設定
  htim2.Init.Period = 10000 - 1;           // カウンタの最大値の設定
  htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
  HAL_TIM_PWM_Init(&htim2);

  // PWM出力の設定
  TIM_OC_InitTypeDef sConfig;
  sConfig.OCMode = TIM_OCMODE_PWM1;
  sConfig.Pulse = 5000;                    // パルス幅の初期値 (0 ~ 10000)
  sConfig.OCPolarity = TIM_OCPOLARITY_HIGH;
  sConfig.OCFastMode = TIM_OCFAST_DISABLE;
  HAL_TIM_PWM_ConfigChannel(&htim2, &sConfig, TIM_CHANNEL_1);
```

```
  // PWM出力の開始
  HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);

  while (1) {
      // パルス幅を変化させる
      for (int dutyCycle = 0; dutyCycle <= 10000; dutyCycle += 1000) {
          htim2.Instance->CCR1 = dutyCycle; // パルス幅の設定
          HAL_Delay(1000);
      }
  }
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
  RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL16;

  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                                |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

}
```