```
#include "stm32f4xx.h"
#include "stm32f4_discovery.h"
#include "stm32f4xx_conf.h"
#include "stdio.h"

#define PWM_FREQUENCY 10000

uint16_t motoracisi = 0;
char lcdHafiza[16];

TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
TIM_OCInitTypeDef TIM_OCInitStructure;

GPIO_InitTypeDef GPIO_InitStructure;
GPIO_TypeDef *RS_PORT = GPIOA; // RS- PA6
GPIO_TypeDef *E_PORT = GPIOA; // E -PA7
GPIO_TypeDef *D4_PORT = GPIOB; // D4 -PB6
GPIO_TypeDef *D5_PORT = GPIOB; // D5 - PB7
GPIO_TypeDef *D6_PORT = GPIOB; // D6 -PB8
GPIO_TypeDef *D7_PORT = GPIOB; // D7 -PB9

GPIO_TypeDef *BUTTON1_PORT = GPIOA; // 1. buton - A1
GPIO_TypeDef *BUTTON2_PORT = GPIOA; // 2. buton - A2

GPIO_TypeDef *MAVİ_LED_PORT = GPIOD; // Mavi -PD12
GPIO_TypeDef *TURUNCU_LED_PORT = GPIOD; // Turuncu -PD13 üzerinde

void PWM_Config(void)
{
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);
  RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
```

```
 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;

 TIM_TimeBaseStructure.TIM_Prescaler = (uint16_t)((SystemCoreClock / 2) / PWM_FREQUENCY) - 1;
 TIM_TimeBaseStructure.TIM_Period = 100 - 1;
 TIM_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1;
 TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
 TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure);

 TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
 TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
 TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
 TIM_OCInitStructure.TIM_Pulse = 0;
 TIM_OC1Init(TIM4, &TIM_OCInitStructure);

 TIM_OC1PreloadConfig(TIM4, TIM_OCPreload_Enable);

 TIM_Cmd(TIM4, ENABLE);
}

void LCD-GPIO_Ayarla(void)
{
 RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA | RCC_AHB1Periph_GPIOB, ENABLE);

 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7;
 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7 | GPIO_Pin_8 | GPIO_Pin_9;
}

LCD_Init();

void Button_Ayarla(void)
```

```
{
  RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 | GPIO_Pin_2;
}

uint8_t Button1_Bas(void)
{
  return GPIO_ReadInputDataBit(BUTTON1_PORT, GPIO_Pin_1) == Bit_RESET;
}

uint8_t Button2_Bas(void)
{
  return GPIO_ReadInputDataBit(BUTTON2_PORT, GPIO_Pin_2) == Bit_RESET;
}



void UpdateMotoracisi(int8_t aci)
{
  motoracisi += aci;
  if (motoracisi > 100) {
    motoracisi = 100;
  } else if (motoracisi < 0) {
    motoracisi = 0;
  }

  TIM_OCInitStructure.TIM_Pulse = (uint16_t)((motoracisi / 100.0) *
TIM_TimeBaseStructure.TIM_Period);
  TIM_OC1Init(TIM4, &TIM_OCInitStructure);
}
```

```
int main()
{
   PWM_Config();
  LCD-GPIO_Ayarla();
  LCD_Init();
  Button_Ayarla();

GPIO_Write(TURUNCU_LED_PORT, GPIO_Pin_13,0);
GPIO_Write(TURUNCU_LED_PORT, GPIO_Pin_13,1);

  while (1) {
    if (Button1_Bas()) {
uint8_t   aci2=UpdateMotoracisi(10);
GPIO_Write(TURUNCU_LED_PORT, GPIO_Pin_13,0);
GPIO_Write(TURUNCU_LED_PORT, GPIO_Pin_13,1);
sprintf(lcdHafiza, aci2);
  LCD_Write(1,1,aci2);
  LCD_Write(2,6,lcdHafiza);

     while (Button1_Bas());
    }

    if (Button2_Bas()) {
uint8_t aci2=  UpdateMotoracisi(-10); // Motor açısını %10 azalt
     GPIO_Write(TURUNCU_LED_PORT, GPIO_Pin_13,0);
     GPIO_Write(TURUNCU_LED_PORT, GPIO_Pin_13,1);
sprintf(lcdHafiza, aci2);
  LCD_Write(1,1,Motorun Acisi:');
  LCD_Write(2,6,lcdHafiza);
     while (Button2_Bas());
```

```
    }
  }
}
```