#include <stdint.h>

#include <stdbool.h>

#include "stm32f10x.h"

int main() {

//------------------UART----------------

// Включаем тактирование для USART1 и GPIOA

RCC->APB2ENR |= RCC\_APB2ENR\_USART1EN; // USART1

RCC->APB2ENR |= RCC\_APB2ENR\_IOPAEN; // GPIOA

// Настройка GPIO для USART1 Tx (PA9) и Rx (PA10)

GPIOA->CRH &= ~GPIO\_CRH\_CNF9;

GPIOA->CRH |= GPIO\_CRH\_CNF9\_1 | GPIO\_CRH\_MODE9;

GPIOA->CRH &= ~GPIO\_CRH\_CNF10;

GPIOA->CRH |= GPIO\_CRH\_CNF10\_1;

GPIOA->BSRR |= GPIO\_BSRR\_BS10;

// Настройка USART1

USART1->BRR = 2500; // Настройка скорости передачи: 9600 при тактовой частоте 24 MHz

USART1->CR1 = USART\_CR1\_UE | USART\_CR1\_TE | USART\_CR1\_RE; // Включить USART, передатчик и приемник

//--------------------GPIO--------------

RCC->APB2ENR |= RCC\_APB2ENR\_IOPCEN; // Включаем тактирование для GPIOC

GPIOC->CRH |= GPIO\_CRH\_MODE8; // Настройка PC8 в режим выхода на 50 MHz

GPIOC->CRH &= ~GPIO\_CRH\_CNF8; // Сброс битов конфигурации

GPIOC->CRH |= GPIO\_CRH\_CNF8\_1; // Альтернативная функция push-pull

GPIOC->CRH |= GPIO\_CRH\_MODE9; // Настройка PC9 в режим выхода на 50 MHz

GPIOC->CRH &= ~GPIO\_CRH\_CNF9; // Сброс битов конфигурации

GPIOC->CRH |= GPIO\_CRH\_CNF9\_1; // Альтернативная функция push-pull

//---------------------таймер------------

RCC->APB1ENR |= RCC\_APB1ENR\_TIM3EN; // Включаем тактирование для TIM3

RCC->APB2ENR |= RCC\_APB2ENR\_AFIOEN; // Включаем тактирование для AFIO

AFIO->MAPR |= AFIO\_MAPR\_TIM3\_REMAP; // Переназначаем пины для TIM3

TIM3->PSC = 240; // Предделитель

TIM3->ARR = 100; // Автоматическая перезагрузка

// Выбор режима ШИМ для канала 3

TIM3->CCMR2 |= TIM\_CCMR2\_OC3M\_1 | TIM\_CCMR2\_OC3M\_2;

TIM3->CCMR2 |= TIM\_CCMR2\_OC4M\_1 | TIM\_CCMR2\_OC4M\_2;

// Разрешение события сравнения для канала 3 и 4

TIM3->CCER |= TIM\_CCER\_CC3E;

TIM3->CCER |= TIM\_CCER\_CC4E;

// Предзагрузка

TIM3->CR1 |= TIM\_CR1\_ARPE;

// Включение таймера

TIM3->CR1 |= TIM\_CR1\_CEN;

//--------------------------------------------------

char buffer[3] = {0};

uint8\_t buffer\_index = 0;

while (1) {

if (USART1->SR & USART\_SR\_RXNE) {

char received\_char = USART1->DR;

while (!(USART1->SR & USART\_SR\_TXE));

USART1->DR = received\_char;

if (received\_char == 'b' || buffer\_index > 0 || received\_char == 'g') {

buffer[buffer\_index++] = received\_char;

if (buffer\_index == 3) {

// Проверка на правильность команды 'bXX'

if (buffer[0] == 'b' && buffer[1] >= '0' && buffer[1] <= '9' && buffer[2] >= '0' && buffer[2] <= '9') {

uint8\_t brightness = (buffer[1] - '0') \* 10 + (buffer[2] - '0');

TIM3->CCR3 = brightness;

}

else if (buffer[0] == 'g' && buffer[1] >= '0' && buffer[1] <= '9' && buffer[2] >= '0' && buffer[2] <= '9') {

uint8\_t brightness = (buffer[1] - '0') \* 10 + (buffer[2] - '0');

TIM3->CCR4 = brightness;

}

buffer\_index = 0;

}

}

}

}

}