











#include "stm32f4xx.h"

int data;

enum PORT {A, B, C, D, E, F, G, H, I};

void SetAltFunc(GPIO\_TypeDef\* Port, int Channel, int AF)

{

Port->MODER &= ~(3<<(2\*Channel));

Port->MODER |= 2<<(2\*Channel);

if(Channel<8)

{

Port->AFR[0] &= ~(15<<4\*Channel);

Port->AFR[0] |= AF<<(4\*Channel);

}

else

{

Port->AFR[1] &= ~(15<<4\*(Channel-8));

Port->AFR[1] |= AF<<(4\*(Channel-8));

}

}

extern "C" void EXTI0\_IRQHandler() // ???????? ??????? ??? EXTI0

{

EXTI->PR = 1<<0; // ????? ??? ?????????? ?????????? (?????????? 0 ??????????)

if (GPIOA->IDR & (1<<0)) // ???????? ???????? ????????? ??????? ? PA0

{

TIM4->CCR1 = 500; // ?????????? PD12 = 100\*500/1000 = 50% (RED)

TIM4->CCR3 = 0; // ?????????? PD14 = 100\*500/1000 = 75% (BLUE)

}

else

{

TIM4->CCR1 = 0; // ?????????? PD12 = 100\*500/1000 = 50% (RED)

TIM4->CCR3 = 500; // ?????????? PD14 = 100\*500/1000 = 75% (BLUE)

}

}

void SetEXTI(PORT Port, int Channel, bool Rise, bool Fall)

{

SYSCFG->EXTICR[Channel/4] &= ~(15<<(4\*(Channel%4)));

SYSCFG->EXTICR[Channel/4] |= Port<<(4\*(Channel%4));

EXTI->IMR |= 1<<Channel;

if(Rise) EXTI->RTSR |= 1<<Channel;

else EXTI->RTSR &= ~(1<<Channel);

if(Fall) EXTI->FTSR |= 1<<Channel;

else EXTI->FTSR &= ~(1<<Channel);

}

int main()

{

RCC->APB1ENR |= RCC\_APB1ENR\_TIM4EN;

RCC->AHB1ENR |= RCC\_AHB1ENR\_GPIODEN;

SetAltFunc(GPIOD, 12, 2);

SetAltFunc(GPIOD, 13, 2);

SetAltFunc(GPIOD, 14, 2);

TIM4->CR2 = TIM\_CR2\_MMS\_0 | TIM\_CR2\_MMS\_1;

TIM4->ARR = 1000 - 1;

TIM4->PSC = (84000000/1000/2000)-1;

TIM4->CCMR1 = TIM\_CCMR1\_OC1M\_1 | TIM\_CCMR1\_OC1M\_2 | TIM\_CCMR1\_OC1PE; // ????? CCR1

TIM4->CCMR1 |= TIM\_CCMR1\_OC2M\_1 | TIM\_CCMR1\_OC2M\_2 | TIM\_CCMR1\_OC2PE; // ????? CCR2

TIM4->CCMR2 = TIM\_CCMR2\_OC3M\_1 | TIM\_CCMR2\_OC3M\_2 | TIM\_CCMR2\_OC3PE; // ????? CCR3

TIM4->CCER = TIM\_CCER\_CC1E; // ????? PD12 ???????

TIM4->CCER |= TIM\_CCER\_CC2E; // ????? PD13 ???????

TIM4->CCER |= TIM\_CCER\_CC3E; // ????? PD14 ???????

TIM4->CR1 = TIM\_CR1\_CEN;

TIM4->CCR1 = 400; // ?????????? PD12 = 100\*500/1000 = 50% (RED)

TIM4->CCR2 = 800; // ?????????? PD13 = 100\*500/1000 = 25% (GREEN)

RCC->AHB1ENR |= RCC\_AHB1ENR\_GPIOAEN; // ???? A ????????????

GPIOA->MODER &= ~GPIO\_MODER\_MODER0;

RCC->APB2ENR |= RCC\_APB2ENR\_SYSCFGEN;

SetEXTI(PORT::A, 0, true, true);

NVIC\_SetPriority(EXTI0\_IRQn, 0); // ?????? ????????? ??????????

NVIC\_EnableIRQ(EXTI0\_IRQn);

while(true){}

}