**Code For Wiper System :**

/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file : main.c

\* @author : Auto-generated by STM32CubeIDE

\* @brief : Main program body

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @attention

\*

\* Copyright (c) 2022 STMicroelectronics.

\* All rights reserved.

\*

\* This software is licensed under terms that can be found in the LICENSE file

\* in the root directory of this software component.

\* If no LICENSE file comes with this software, it is provided AS-IS.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#include <stdint.h>

#include "stm324xx.h"

#include "gpio.h"

#include "rcc.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\* Header files \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#if !defined(\_SOFT\_FP) && defined(\_ARM\_FP)

#warning "FPU is not initialized, but the project is compiling for an FPU. Please initialize the FPU before use."

#endif

/\* RCC (Reset and clock Control) Registers \*/

#define RCC\_BASE\_ADDR 0x40023800

#define RCC\_AHB1ENR\_OFFSET 0x30

#define RCC\_AHB1ENR \*(volatile unsigned int \*)(RCC\_BASE\_ADDR + RCC\_AHB1ENR\_OFFSET)

/\* GPIO Registers \*/

#define GPIOD\_BASE\_ADDR 0x40020C00

#define GPIOD\_MODER\_OFFSET 0x00

#define GPIOD\_MODER \*(volatile unsigned int \*)(GPIOD\_BASE\_ADDR + GPIOD\_MODER\_OFFSET)

#define GPIOD\_ODR\_OFFSET 0x14

#define GPIOD\_ODR \*(volatile unsigned int \*)(GPIOD\_BASE\_ADDR + GPIOD\_ODR\_OFFSET)

static volatile int flag;

static volatile int SCount = 0, count=0; //Global variables

//Scount is number of times the switch is pressed

//count is simple variable to count the loop

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*Interrupt handler \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void exti0\_irqhandler(void)

{

if(EXTI->PR & (1 << PIN\_0))

{

flag = 1;

EXTI->PR = EXTI->PR | (1 << PIN\_0);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*Code to Turn on and off Red LED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

long delay = 0xFFFFFFF;

static void on\_red\_led(void)

{

volatile long i;

GPIOD\_ODR = GPIOD\_ODR | 0x00001000;

}

static void off\_red\_led(void)

{

volatile long i;

GPIOD\_ODR = GPIOD\_ODR & ~(0x00001000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\* GPIO Configuration \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static void init\_config(void)

{

config\_rcc(GPIOD);

config\_gpiox(GPIOD, PIN\_12, GPIO\_OUTPUT\_PP, GPIO\_SPEED\_VERY\_HIGH); //Green LED

config\_rcc(GPIOD);

config\_gpiox(GPIOD, PIN\_14, GPIO\_OUTPUT\_PP, GPIO\_SPEED\_VERY\_HIGH); //Red LED

config\_rcc(GPIOD);

config\_gpiox(GPIOD, PIN\_13, GPIO\_OUTPUT\_PP, GPIO\_SPEED\_VERY\_HIGH); //Orange LED

config\_rcc(GPIOD);

config\_gpiox(GPIOD, PIN\_15, GPIO\_OUTPUT\_PP, GPIO\_SPEED\_VERY\_HIGH); //Blue LED

config\_rcc(GPIOA);

config\_gpiox(GPIOA, PIN\_0, GPIO\_INPUT, GPIO\_SPEED\_VERY\_HIGH); //Push Button

config\_gpio\_irq\_priority(IRQ\_NO\_EXTI0, NVIC\_IRQ\_PR\_LVL\_0); //Interrupt Handler

config\_gpio\_interrupt(IRQ\_NO\_EXTI0, ENABLE);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\* Code to Turn on and off Blue LED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static void LED\_1\_ON()

{

/\* Setting PD15 (Pin 15 of PORTD) as General Purpose Output \*/

GPIOD\_MODER = GPIOD\_MODER | 0x01000000000;

GPIOD\_ODR = GPIOD\_ODR | 0x00001000;

}

static void LED\_1\_OFF()

{

/\* Setting PD15 (Pin 15 of PORTD) as General Purpose Output \*/

GPIOD\_MODER = GPIOD\_MODER | 0x01000000000;

GPIOD\_ODR = GPIOD\_ODR & ~0x00001000;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\* Code to Turn on and off Green LED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static void LED\_2\_ON()

{

/\* Setting PD12 (Pin 12 of PORTD) as General Purpose Output \*/

GPIOD\_MODER = GPIOD\_MODER | 0x01000000;

GPIOD\_ODR = GPIOD\_ODR | 0x00001000;

}

static void LED\_2\_OFF()

{

/\* Setting PD12 (Pin 12 of PORTD) as General Purpose Output \*/

GPIOD\_MODER = GPIOD\_MODER | 0x01000000;

GPIOD\_ODR = GPIOD\_ODR & ~0x00001000;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\* Code to Turn on and off Orange LED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static void LED\_3\_ON()

{

/\* Setting PD13 (Pin 13 of PORTD) as General Purpose Output \*/

GPIOD\_MODER = GPIOD\_MODER | 0x010000000;

GPIOD\_ODR = GPIOD\_ODR | 0x00001000;

}

static void LED\_3\_OFF()

{

/\* Setting PD13 (Pin 13 of PORTD) as General Purpose Output \*/

GPIOD\_MODER = GPIOD\_MODER | 0x010000000;

GPIOD\_ODR = GPIOD\_ODR & ~0x00001000;

}

static void Delay(int x) // Delay function for LED

{

for(int i=0;i<x;i++);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\* Main Function \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main(void)

{

int key;

long int A; // variable to check for button pressed duration

init\_config(); // Configuration Function for GPIO and interrupt

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\* Logic for Wiper System \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while (1)

{

key = gpiox\_read\_pin(GPIOA, PIN\_0); // Polling

if(key == 1)

{

A = 0;

long int j;

for(long int j=0;j<=5000000;j++)

{

A++;

}

if(A>2000000) //1uSec = 1 clock hence 2 Sec = 2000000 Cycle

{

on\_red\_led(); //Turn on Red LED

if (flag == 1) // Set in ISR

{

flag = 0;

count++;

SCount=count % 4;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\* Wiper Moving at 1 Hz \*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while(SCount==1)

{

LED\_1\_ON();

Delay(10000);

LED\_1\_OFF();

LED\_2\_ON();

Delay(10000);

LED\_2\_OFF();

LED\_3\_ON();

Delay(10000);

LED\_3\_OFF();

LED\_2\_ON();

Delay(10000);

LED\_2\_OFF();

LED\_1\_ON();

Delay(10000);

LED\_1\_OFF();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\* Wiper Moving at 4 Hz \*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while(SCount==2)

{

LED\_1\_ON();

Delay(2500);

LED\_1\_OFF();

LED\_2\_ON();

Delay(2500);

LED\_2\_OFF();

LED\_3\_ON();

Delay(2500);

LED\_3\_OFF();

LED\_2\_ON();

Delay(2500);

LED\_2\_OFF();

LED\_1\_ON();

Delay(2500);

LED\_1\_OFF();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\* Wiper Moving at 8 Hz \*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while(SCount==3)

{

LED\_1\_ON();

Delay(1250);

LED\_1\_OFF();

LED\_2\_ON();

Delay(1250);

LED\_2\_OFF();

LED\_3\_ON();

Delay(1250);

LED\_3\_OFF();

LED\_2\_ON();

Delay(1250);

LED\_2\_OFF();

LED\_1\_ON();

Delay(1250);

LED\_1\_OFF();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\* Wiper Movement stop \*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

while(SCount==0)

{

LED\_1\_OFF();

LED\_2\_OFF();

LED\_3\_OFF();

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* Wiper System turn Off when switch pressed for less than 2 sec \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

else if(A<2000000)

{

off\_red\_led();

LED\_1\_OFF();

LED\_2\_OFF();

LED\_3\_OFF();

}

}

return 0;

}