```
/*
* Title: main.c
* Author: Noah Rowbotham
* Date: Apr. 8th, 2020
* Lab: ENEL 387 Project
*/

#include "clock.h"
#include "gpio.h"
#include "subsystemSetup.h"
#include "motors.h"
#include "coolCrap.h"
#include "ir_sensors.h"
#include "pwm.h"

void beepBuzzer_QuarterSec(void);
void beepBuzzer_OneSec(void);
void multiBeep(int);
void regularTraversal(void);
void theEnd(void);
void countDownToStart(void);
void homeSound(void);

int main()
{
        initPLL_24MHz();
        initAPB2_GPIO();
        initAPB2_AFIO();
        initAPB2_TIM1();

        configAFIO_Output('A', 8);      //PWM Channel

        configGPIO_Output('B', 10);     //Ultrasonic Trigger
        configGPIO_Output('B', 11);     //Ultrasonic Echo
        configGPIO_Output('B', 12);     //Motor
        configGPIO_Output('B', 13); //Motor
        configGPIO_Output('B', 14); //Motor
        configGPIO_Output('B', 15); //Motor
        configGPIO_Output('C', 6);      //Buzzer

        configGPIO_Input('B', 6);       //Right IR Sensor
        configGPIO_Input('B', 7);       //Left IR Sensor

        uint8_t x = 0x00;
        uint32_t divider = 2399;
        uint32_t reload = 100;
        uint32_t cmp = 95;

        initTIM1(divider, reload, cmp);
```

```
        multiBeep(3);

        while(x == 0x00) { x = getUSER_PB(); }

        countDownToStart();

        regularTraversal();

        return 0;
}

void regularTraversal(void)
{
        bool left = false;
        bool right = false;
        while(1)
        {
                left = leftIR_Triggered();
                right = rightIR_Triggered();

                if (left && right)
                {
                        theEnd();
                        break;
                }
                else if (left)
                {
                        setMotionForward();
                        delay(850000);
                        stopMotion();
                        delay(1500000);

                        left = false;
                        while(!left)
                        {
                                turnLeft();
                                left = leftIR_Triggered();
                        }

                        left = leftIR_Triggered();
                        while(left)
                        {
                                turnLeft();
                                left = leftIR_Triggered();
                        }

                        stopMotion();
                }
                else if (right)
                        turnRight();
```

```
                else
                        setMotionForward();
        }
}



int testForTarget(void)
{
        int targetNum = 0;

        int giveUpCountTime = 1500000;
        int timeCount = 0;

        int giveUpHomeTime = 6000000;
        int timeHome = 0;

        bool notCounted = false;
        bool left = true;
        bool right = true;

        updatePWM(25);
        setMotionForward();

        while(1)
        {
                left = leftIR_Triggered();
                right = rightIR_Triggered();

                if (!left && !right) //white stripe OR more course
                {
                        notCounted = true;

                        while(notCounted)
                        {
                                left = leftIR_Triggered();
                                right = rightIR_Triggered();

                                if (left && right) //black stripe, therefore it was
a white stripe
                                {
                                        notCounted = false;
                                        targetNum++;
                                }

                                timeCount++;

                                if (timeCount == giveUpCountTime) //more course
                                        return targetNum;
```

```
                        }
                }

                if (left && right) //black stripe OR home
                {
                        while(left && right)
                        {
                                left = leftIR_Triggered();
                                right = rightIR_Triggered();

                                timeHome++;

                                if (timeHome == giveUpHomeTime)
                                        theEnd();
                        }
                }
        }
}
void theEnd(void)
{
        stopMotion();
        homeSound();
}



void beepBuzzer_QuarterSec(void)
{
        uint32_t output = 0x00000000;

        output = (GPIOC->ODR & 0xFFFFFFBF); //mask odr to clear PC6
        output |= 0x00000040;
        GPIOC->ODR = output;

        delay(1500000);

        output = 0x00000000;

        output = (GPIOC->ODR & 0xFFFFFFBF); //mask odr to clear PC6
        output |= 0x00000000;
        GPIOC->ODR = output;
}
void beepBuzzer_OneSec(void)
{
        uint32_t output = 0x00000000;

        output = (GPIOC->ODR & 0xFFFFFFBF); //mask odr to clear PC6
        output |= 0x00000040;
        GPIOC->ODR = output;
```

```
        delay(6000000);

        output = 0x00000000;

        output = (GPIOC->ODR & 0xFFFFFFBF); //mask odr to clear PC6
        output |= 0x00000000;
        GPIOC->ODR = output;
}
void multiBeep(int numBeeps)
{
        for (int i = 0; i < numBeeps; i++)
        {
                beepBuzzer_QuarterSec();
                delay(750000);
        }
}
void countDownToStart(void)
{
        for (int i = 0; i < 5; i++)
        {
                beepBuzzer_QuarterSec();
                delay(6000000);
        }
}
void homeSound(void)
{
        beepBuzzer_QuarterSec();
        delay(750000);
        beepBuzzer_QuarterSec();
        delay(750000);
        beepBuzzer_OneSec();
        delay(750000);
        beepBuzzer_QuarterSec();
        delay(750000);
        beepBuzzer_QuarterSec();
        delay(750000);
        beepBuzzer_OneSec();
        delay(750000);
        beepBuzzer_OneSec();
}
```