```c
#include "stm32f10x.h"
#include "LCDlab.h"

void CMD2LCD(uint8_t data)
{
  GPIOB->BSRR = LCD_CM_ENA; //RS low, E high
  // GPIOC->ODR = data; //BAD: may affect upper bits on port C
  GPIOC->ODR &= 0xFF00; //GOOD: clears the low bits without affecting high bits
  GPIOC->ODR |= data; //GOOD: only affects lowest 8 bits of Port C
  delay(8000);
  GPIOB->BSRR = LCD_CM_DIS; //RS low, E low
  delay(80000);
}

void DAT2LCD(uint8_t data)
{
  GPIOB->BSRR = LCD_DM_ENA; //RS low, E high
  // GPIOC->ODR = data; //BAD: may affect upper bits on port C
  GPIOC->ODR &= 0xFF00; //GOOD: clears the low bits without affecting high bits
  GPIOC->ODR |= data; //GOOD: only affects lowest 8 bits of Port C
  delay(8000);
  GPIOB->BSRR = LCD_DM_DIS; //RS low, E low
  delay(80000);
}

/*
* Name:         void delay()
* Paramaters:   32 bit delay value, ( a value of 6000
                gives approximately 1 mS of delay)
* Description:  This function creates a delay
*/
void delay(uint32_t count)
{
    int i=0;
    for(i=0; i< count; ++i)
    {
    }
}

void LCD_IO_INIT()
  {
  //Sets all ports needed in the enable register
  RCC->APB2ENR |= RCC_APB2ENR_IOPAEN | RCC_APB2ENR_IOPBEN | RCC_APB2ENR_IOPCEN;

  GPIOC->CRL |= GPIO_CRL_MODE7 | GPIO_CRL_MODE6 | GPIO_CRL_MODE5 | GPIO_CRL_MODE4 | GPIO_CRL_MODE3 | GPIO_CRL_MODE2| GPIO_CRL_MODE1 | GPIO_CRL_MODE0;
  GPIOC->CRL &= ~GPIO_CRL_CNF7 & ~GPIO_CRL_CNF6 & ~GPIO_CRL_CNF5 & ~GPIO_CRL_CNF4 & ~GPIO_CRL_CNF3 & ~GPIO_CRL_CNF2 & ~GPIO_CRL_CNF1 & ~GPIO_CRL_CNF0;

  GPIOB->CRL |=  GPIO_CRL_MODE5 | GPIO_CRL_MODE1 | GPIO_CRL_MODE0;
  GPIOB->CRL &= ~GPIO_CRL_CNF5 & ~GPIO_CRL_CNF1 & ~GPIO_CRL_CNF0;

}

//uint16_t dipswitchRead()
//{
//
//  uint16_t sw_val;
//  //Sets the GPIOs needed for switches to be read. The final result for this equation will be a 4-bit number corresponding to the values on the switches.
//  sw_val= (((GPIOA-> IDR & (GPIO_IDR_IDR6|GPIO_IDR_IDR7))>>6 | ((GPIOC->IDR&(GPIO_IDR_IDR10|GPIO_IDR_IDR11))>>8)) & 0x000F);
//
//  return (sw_val);
//}

void LCD_INIT()
{
    delay(90000);
    CMD2LCD(LCD_8B2L);
    CMD2LCD(LCD_8B2L);
    CMD2LCD(LCD_8B2L);
    CMD2LCD(LCD_8B2L);
    CMD2LCD(LCD_DCB);
    CMD2LCD(LCD_CLR);
    CMD2LCD(LCD_MCR);
}

void StartLCD()
{
    //char sw1, sw2, sw3, sw4;
    char start[15] = "Start Procedure";
    char condition[16] = "Coast Clear.....";
    char space = ' ';
    //uint16_t dipSwitch = dipswitchRead();

    CMD2LCD(LCD_LN1);
    for(int i = 0;i < 15; i++)
    {
      DAT2LCD(start[i]);
    }

    CMD2LCD(LCD_LN2);

    for(int i = 0;i < 16; i++)
    {
      DAT2LCD(condition[i]);
    }
}

void LeftSensorTriggered()
{
    char start[16] = "Left Sensor Trig";
    char condition[16] = "Turning Right...";

    CMD2LCD(LCD_LN1);
    for(int i = 0;i < 16; i++)
    {
      DAT2LCD(start[i]);
    }

    CMD2LCD(LCD_LN2);

    for(int i = 0;i < 16; i++)
    {
      DAT2LCD(condition[i]);
    }
}

void RightSensorTriggered()
{
    char start[15] = "Right Snsr Trig";
    char condition[16] = "Turning Left...";

    CMD2LCD(LCD_LN1);
    for(int i = 0;i < 15; i++)
    {
      DAT2LCD(start[i]);
    }

    CMD2LCD(LCD_LN2);

    for(int i = 0;i < 16; i++)
    {
      DAT2LCD(condition[i]);
    }
}

void BothSensorTriggered()
{
    char start[15] = "Both Snsrs Trig";
    char condition[16] = "Moving back.....";

    CMD2LCD(LCD_LN1);
    for(int i = 0;i < 15; i++)
    {
      DAT2LCD(start[i]);
    }

    CMD2LCD(LCD_LN2);

    for(int i = 0;i < 16; i++)
    {
      DAT2LCD(condition[i]);
    }
}

void FlameSensorTrig()
{
    char start[15] = "FLAME DETECTED!";
    char condition[16] = "EXTINGUISHING...";

    CMD2LCD(LCD_LN1);
    for(int i = 0;i < 15; i++)
    {
      DAT2LCD(start[i]);
    }

    CMD2LCD(LCD_LN2);

    for(int i = 0;i < 16; i++)
    {
      DAT2LCD(condition[i]);
    }
}
```