```c
/* Name: David (DongYun) Kim
 * SID: 200405213
 * Course: ENEL351
 * Description: ENEL351 Project - Smart Parking System
 * File name: main.c
 */

#include "stm32f10x.h"
#include "timer.h"
#include "gpio.h"
#include "pwm.h"
#include "timer.h"
#include "i2c.h"
#include "i2c_lcd_driver.h"
#include "adc.h"
#include <stdio.h>
#include <string.h>
#include <math.h>

int main(void) {
  uint8_t my_lcd_addr = 0x27;
  char welcome[17] = " WELCOME TO SPS ";
  int openSpots = 5; // Initialize counter for open parking spots

  SystemInit(); // 72 MHz system clock
  LD2_init(); // Green LED
  tim2_init(); // Delay function timer
  pwminit();
  i2c_init(); //init I2C1 in standard mode
  i2c_enable();
  lcd_init(my_lcd_addr); // Send the initialization commands to the i2c LCD
  adc1_init();

  while (1) {

  /***IR Sensor at the Entry***/
    if ((GPIOB -> IDR & GPIO_IDR_IDR9) == 0) // Car entry detected
    {
      if (openSpots > 0) {
        openSpots--; // Decrease the open spots if parking is not full
        change_CH1_DC(100); // Open the gate
        delay_ms(2000);
        change_CH1_DC(190); // Return servo to initial position
      } else {
        // If parking is already full and another car tries to enter, operate the servo
        openSpots--; // Decrease the open spots if parking is not full
        delay_ms(2000);
      }
    }


  /***IR Sensor at the Exit***/
    if ((GPIOB -> IDR & GPIO_IDR_IDR8) == 0) // Car exit detected
    {
      change_CH2_DC(190); // Open the gate
      delay_ms(2000);
      change_CH2_DC(100); // Return servo to initial position
      openSpots = (openSpots < 5) ? openSpots + 1 : 5; // Increase only if less than total
spots
    }


  /*** Sharp GP2Y0A41SK0F Analog Sensor ***/

  int adc_val = adc1_acquire();

  // Conversion of adc value to voltage, as working with a 12-bit ADC
  // 0xFFF is the maximum value (Decimal=4095, Voltage=3.3)
  // Examle in lab, we used 0xC00 for 2.465V and 0xFFF for 3.3V
    float voltage = (adc_val / 4095.0f) * 3.3f;

  // Empirical formula to convert voltage to distance, 13 * voltage^-1
  // Reference from, https://www.instructables.com/How-to-Use-the-Sharp-IR-Sensor-GP2Y0A41SK0F-
Arduin/
  int distance = (int)(13 * pow(voltage, -1));

  // Smart Parking Assitant where the analog sensor helps
  if (distance == 4) {
        GPIOC->ODR |= GPIO_ODR_ODR8;
    } else {
        GPIOC->ODR &= ~GPIO_ODR_ODR8;
    }

    if (distance >= 5 && distance < 8) {
        GPIOC->ODR |= GPIO_ODR_ODR6;
    } else {
        GPIOC->ODR &= ~GPIO_ODR_ODR6;
    }

    if (distance >= 8 && distance < 15) {
        GPIOC->ODR |= GPIO_ODR_ODR12;
    } else {
        GPIOC->ODR &= ~GPIO_ODR_ODR12;
    }

    if (distance >= 15 && distance < 18) {
        GPIOC->ODR |= GPIO_ODR_ODR10;
    } else {
        GPIOC->ODR &= ~GPIO_ODR_ODR10;
    }

  /*** I2C LCD 16x2 ***/
    char displayStr[17];

    if (openSpots <= 0) {
      // If no open spots are available, indicate the parking is full
      sprintf(displayStr, "PARKING IS FULL");
      openSpots = 0;
    } else {
      // Otherwise, display the number of open spots
      sprintf(displayStr, "  Open Spot: %d ", openSpots);
    }

    lcd_write_cmd(my_lcd_addr, LCD_LN1); // Position cursor at beginning of line 1
    stringToLCD(my_lcd_addr, welcome); // Output a welcome message
    lcd_write_cmd(my_lcd_addr, LCD_LN2); // Position cursor at beginning of line 2
    stringToLCD(my_lcd_addr, displayStr); // Display the appropriate message based on the
number of open spots
    lcd_write_cmd(my_lcd_addr, LCD_DISPLAYCONTROL | LCD_DISPLAYON | LCD_CURSOROFF |
LCD_BLINKOFF);
  }
}
```