```c
/*************************************************************************
* main.c                                                                  *
* Grubmüller Stefan, Marx Clemens                                         *
* May 2020                                                                *
*                                                                         *
* Function: This programm should use the sparkfun sensor and the          *
* proximity engine. At the beginning the LCD will show you the Text:      *
*                                                                         *
* WELCOME                                                                 *
* DEVICE LINKED / DISLINKED                                               *
*                                                                         *
* After sending data via the I2C - so data from the device/cortex can     *
* be transmitted and recieved - the LCD will show you a real time         *
* clock, the distance of a object near the threashold and if it is in     *
* threshold range. Example:                                               *
*                                                                         *
* IN RANGE  20-140                                                        *
* 12:04:31      50                                                        *
*                                                                         *
* This process will run paralell due to interrupts.                       *
* To recieve data from the slave/device you have to configure the         *
* devive registers shown in proximity.c.                                  *

* More information in the scritum by @JosefReisinger and in the           *
* Specifications by @ClemensMarx and @StefanGrubmueller or in the         *
* datasheet by @sparkfun:                                                 *
* https://cdn.sparkfun.com/datasheets/Sensors/Proximity/apds9960.pdf      *
*************************************************************************/


/* ------------------------------- Main ----------------------------------*/
#include "proximity.h"
#include "stdlib.h"
#include <string.h>
#include <stdio.h>
int main()
{
  // Initalisations
  // set_clock_36MHz();           // set system clock to 36MHz
  InitI2CPorts();                 // initialisation of GPIO ports (PB6 = SCL and PB7 = SDA)
  i2c_init(&device, &SCL, &SDA);  // initialisation of I2C (extra library)
  lcd_init();                     // initialisation of LCD
  lcd_clear();                    // clear screen
  uart_init(9600);                // 9600,8,n,1
  uart_clear();                   // send clear string to VT 100 terminal


  // PA1 as Input (external interrupt Pin of sparkfun sensor)
  RCC->APB2ENR |= RCC_APB2ENR_IOPAEN;  // enable clock for GPIOA (APB2 Peripheral clock enable register)
  GPIOA->CRL &= 0xFFFFFF0F;            // set Port Pins PA1 to Pull Up/Down Input mode (50MHz) = Mode 8
  GPIOA->CRL |= 0x00000080;
  GPIOA->ODR |= 0x0002;


  // starting text on LCD
  lcd_set_cursor(0, 0);           // set position on LCD
  lcd_put_string("WELCOME");      // write on LCD
  check_device_con();             // is the device connected?
  wait_ms(2000);                  // wait 2 seconds
  lcd_clear();

  // timer on lcd (real time clock)
  milsek = 0;        // initalise milliseconds
  TIM3_Config();     // start timer 3: Upcounter --> triggers every 0,1s an update interrupt



  start_proximity_engine(); // set of configuration registers for proximity detection

  EXTI_config(a, 1);        // external interrupt pin; triggers when int pin of sensor sends falling edge


  // endless loop
  while (1)
  {
    clock_lcd();

    // read data of PDATA regsiter (0x90)
    char pdata_w[] = {0x9C};
    i2c_write(&device, pdata_w, 1, END_WITHOUT_STOP);
    char pdata_r;
    i2c_read(&device, &pdata_r, 1);

    // output of proximity data
    char buffer_i [8]= {0};           // set and clear buffer
    sprintf(buffer_i, "%d", pdata_r); // proximity data as int
    lcd_set_cursor(1,13);
    lcd_put_string(buffer_i);         // output of proximity data on lcd as int

    //output of range (lower threshold to higher threshold)
    char threshold[2];
    sprintf(threshold, "%d-%d", LOWTHRES, HIGHTHRES);
    lcd_set_cursor(0,9);
    lcd_put_string(threshold);        // output of range on lcd as int

    // check if data of proximity data register is inside range
    if (((unsigned char)pdata_r >= LOWTHRES) && ((unsigned char)pdata_r <= HIGHTHRES))
    {
      char en_reg[2] = {ENABLE_REG, SET_PIEN};          // set PIEN (enable Proximity Interrupt)
      i2c_write(&device, en_reg, 2, END_WITHOUT_STOP);  // set enable register
    }
    wait_ms(500);   // wait to avoid lecking image
    lcd_clear();    // clear screen
  }
}

```