```c
/*****************************************************************************
* proximity.h                                                                *
* Grubmüller Stefan, Marx Clemens                                            *
* May 2020                                                                   *
*                                                                            *
* Function: This header should define all the functions and variables        *
* being used in different files.                                             *
*                                                                            *
* 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         *
*****************************************************************************/

#ifndef __PROXIMITY_H__
#define __PROXIMITY_H__

/* ---------------------------Includes ---------------------------------*/

#include <stm32f10x.h>
#include "armv10_std.h"
#include "i2c.h"                    // I2C library by Jakob Pachtrog
//#include <stm32f10x_i2c.h>        // I2C default library


/* -------------------------- defines ----------------------------------*/
// bitbanding for SDA (PB7)
#define GPIOB_IDR GPIOB_BASE + 2*sizeof(uint32_t)
#define SDA_IN *((volatile unsigned long *)(BITBAND_PERI(GPIOB_IDR,7))) //PB7 - Input

#define ENABLE_REG          0x80      // enable register
#define PERS_REG            0x8C      // persistance register
#define LOWTHRES_REG        0x89      // lower threashold
#define HIGHTHRES_REG       0x8B      // higher threashold
#define PROX_PULSE_REG      0x8E      // proximity pulse register
#define CONTROL_REG1        0x8F      // control register one
#define CONTROL_REG2        0x90      // control register two
#define STAT_REG            0x93      // status register
#define UPRIGHT_OFFSET_REG  0x9D      // proximity offset UP / RIGHT register
#define DLEFT_OFFSET_REG    0x9E      // proximity offset UP / RIGHT register
#define CONF_REG3           0x9F      // configuration register three
#define PROX_INT_CLEAR      0xE5      // proximity interrupt clear
#define CLEAR_ALL_INT       0xE7      // clear all non-gesture interrupts
#define SET_PIEN            0x25      // set Proximity interrupt enable
#define DEL_PIEN            0x05      // delete PIEN bit (PIEN = 0)

// Threshold- Low to High
// 0xFF...very near
// 0x00...very far
#define LOWTHRES            0x10      // lower threashold for proximity
#define HIGHTHRES           0xAC      // higher threashold for proximity

// Light Intensity
// Very Dark to very Bright
// Various Gains
#define GAIN_x1             0x00      // very bright conditions (GAIN = 1)
#define GAIN_x2             0x04      // not that bright but also not dark ~  (GAIN = 2)
#define GAIN_x4             0x08      // need of light (normal room condtions in evening) (GAIN = 4)
#define GAIN_x8             0x0C      // dark conditions (GAIN = 8)

// activates or deactivates the LED Boost option of sensor
#define LED_BOOST_ON        0xA0      // additional current up to 200%
#define LED_BOOST_OFF       0x80      // non addtional current

extern int milsek;


/* ------------------------- Prototypes ------------------------------*/

// init ports (PA7 Open Drain)
void InitI2CPorts(void);

// interrupts
void NVIC_init(char position, char priority);
// Nestet Vector Interrupt Controller
void NVIC_init(char position, char priority);
// timer
```

```
78  void TIM3_Config(void);
79
80  // real time clock on lcd
81  void clock_lcd(void);
82  // check connection of device
83  void check_device_con(void);
84
85  // start proximity engine due to setting the register bits
86  void start_proximity_engine(void);
87  // reads data out of the Proximity data register (0x9C)
88  void read_data(void);
89
90  // transfer pins for function of EXTI_config
91  typedef enum { a, b, c, d, e, f, } pin;
92  // external interrupt
93  void EXTI_config(pin p, int n);
94
95  /* --------------------------- structures ---------------------------------*/
96
97  extern I2C_Device device;
98
99  extern I2C_PIN_CONF SCL;    // defenition of SCL (I2C1)
100
101 extern I2C_PIN_CONF SDA;    // defenition of SDA (I2C1)
102
103 #endif
104
```