```c
/***********************************************************************
 *
 * GPIO library for AVR-GCC.
 * ATmega328P (Arduino Uno), 16 MHz, AVR 8-bit Toolchain 3.6.2
 *
 * Copyright (c) 2019-2020 Tomas Fryza
 * Dept. of Radio Electronics, Brno University of Technology, Czechia
 * This work is licensed under the terms of the MIT license.
 *
 **********************************************************************/

/* Includes ----------------------------------------------------------*/
#include "gpio.h"

/* Function definitions ----------------------------------------------*/
void GPIO_config_output(volatile uint8_t *reg_name, uint8_t pin_num)
{
    //na danem bitu adresy nastavi 1
    *reg_name |= (1<<pin_num);
}

/*-----------------------------------------------------------------------*/
/* GPIO_config_input_nopull */
void GPIO_config_input_nopull(volatile uint8_t *reg_name, uint8_t pin_num)
{
    //(inp nopull)
    //na danem bitu adresy nastavi 0
    //na danem bitu o jednu vyssi adresy nastavi 0
    *reg_name &= ~(1<<pin_num);  // Data Direction Register
    *reg_name++;                    // Change pointer to Data Register
    *reg_name &= ~(1<<pin_num);   // Data Register
}

/*-----------------------------------------------------------------------*/
void GPIO_config_input_pullup(volatile uint8_t *reg_name, uint8_t pin_num)
{
    //(inp pull)
    //na danem bitu adresy nastavi 0
    //na danem bitu o jednu vyssi adresy nastavi 1
    *reg_name &= ~(1<<pin_num);  // Data Direction Register
    *reg_name++;                    // Change pointer to Data Register
    *reg_name |= (1<<pin_num);   // Data Register
}

/*-----------------------------------------------------------------------*/
void GPIO_write_low(volatile uint8_t *reg_name, uint8_t pin_num)
{
    //na danem bitu adresy nastavi 0
    *reg_name &= ~(1<<pin_num);
}

/*-----------------------------------------------------------------------*/
/* GPIO_write_high */
```

```
54  void GPIO_write_high(volatile uint8_t *reg_name, uint8_t pin_num)
55  {
56      //na danem bitu adresy nastavi 1
57      *reg_name |= (1<<pin_num);
58  }
59
60  /*-------------------------------------------------------------------*/
61  /* GPIO_toggle */
62  void GPIO_toggle(volatile uint8_t *reg_name, uint8_t pin_num)
63  {
64      //na danem bitu adresy nastavi negaci bitu
65      *reg_name ^= (1<<pin_num);
66  }
67
68  /*-------------------------------------------------------------------*/
69  /* GPIO_read */
70  uint8_t GPIO_read(volatile uint8_t *reg_name, uint8_t pin_num)
71  {
72      // kdyz je dany bit na dane adrese 0
73      // tak se vrati 0 jinak 1
74      if(bit_is_clear(*reg_name, pin_num))
75          return 0;
76      else
77          return 1;
78  }
```