

 mariogolbi Create Lab-3.md



1 contributor

# Lab 3: MARIO GOLBANO CORZO

[Link to my Digital-electronics-2 GitHub repository](#)

## Data types in C

1. Complete table.

Data type	Number of bits	Range	Description
uint8_t	8	0, 1, ..., 255	Unsigned 8-bit integer
int8_t	8	-128 .. 127	Signed 8 bit integer
uint16_t	16	0 .. 65,535	Unsigned 16 bit integer
int16_t	16	-32,768 .. 32,767	Signed 16 bits integer
float	32	-3.4e+38, ..., 3.4e+38	Single-precision floating-point

Data type	Number of bits	Range	Description
void	32 for 32bits systems and 64 for 64 bits systems	-	No data type but a non-return function or an pointer to something unspecified

## GPIO library

1. In your words, describe the difference between the declaration and the definition of the function in C.

- Function declaration: The function declaration only refers to the prototype of the function where we should specify the name of the function, the return types and the parameters, and then it calls the function. There is no body of the function.
- Function definition: The function definition also includes the body of the function. We could say it is the function itself, due to the fact that in it we can find all the elements of the function.

2. Part of the C code listing with syntax highlighting, which toggles LEDs only if push button is pressed. Otherwise, the value of the LEDs does not change. Use function from your GPIO library. Let the push button is connected to port D:

```

/*****
 *
 * Alternately toggle two LEDs when a push button is pressed. Use
 * functions from GPIO library.
 * ATmega328P (Arduino Uno), 16 MHz, AVR 8-bit Toolchain 3.6.2
 *
 * Copyright (c) 2019-Present Tomas Fryza
 * Dept. of Radio Electronics, Brno University of Technology, Czechia
 * This work is licensed under the terms of the MIT license.
 *
 *****/

/* Defines -----*/
#define LED_GREEN PB5 // AVR pin where green LED is connected
#define LED_2 PC5
#define PUSH_BUTTON PD5
#define BLINK_DELAY 500
#ifndef F_CPU
# define F_CPU 16000000 // CPU frequency in Hz required for delay
#endif

/* Includes -----*/
#include <util/delay.h> // Functions for busy-wait delay loops
#include <avr/io.h> // AVR device-specific IO definitions
#include "gpio.h" // GPIO library for AVR-GCC

```



```

/* Function definitions -----*/
/*****
 * Function: Main function where the program execution begins
 * Purpose: Toggle two LEDs when a push button is pressed. Functions
 *          from user-defined GPIO library is used.
 */

```

☰ 100 lines (78 sloc) | 4.08 KB

...

```

int main(void)
{
    // Green LED at port B
    GPIO_config_output(&DDRB, LED_GREEN);
    GPIO_write_low(&PORTB, LED_GREEN);

    // Configure the second LED at port C
    GPIO_config_output(&DDRC, LED_2);
    GPIO_write_low(&PORTC, LED_2);

    // Configure Push button at port D and enable internal pull-up resistor
    GPIO_config_output(&DDRD, PUSH_BUTTON);
    GPIO_config_input_pullup(&PORTD, PUSH_BUTTON);

    // Infinite loop
    while (1)
    {
        // Pause several milliseconds
        _delay_ms(BLINK_DELAY);

        if (GPIO_read(&PORTD, PUSH_BUTTON)){
            GPIO_toggle(&DDRB, LED_GREEN);
            GPIO_toggle(&DDRB, LED_2);
        }
    }

    // Will never reach this
    return 0;
}

```

## Traffic light

1. Scheme of traffic light application with one red/yellow/green light for cars and one red/green light for pedestrians. Connect AVR device, LEDs, resistors, one push button (for pedestrians), and supply voltage. The image can be drawn on a computer or by hand. Always name all components and their values!

