Assignment lab 3 Amaury Menneteau

```
# Lab 3: YOUR_FIRSTNAME FAMILYNAME

Link to your `Digital-electronics-2` GitHub repository:

   [https://github.com/...](https://github.com/...)


### 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          | -137,…,138     | Signed integer 8
bits|
| `uint16_t`    |         16         |    65535       | Unsigned integer 16
bits    |
| `int16_t`     |         16         |   -32767,…,32768| Signed integer 16
bits |
| `float`       |         32         | -3.4e+38, ..., 3.4e+38 | Single-
precision floating-point |
| `void`        |         2          |  0,1        | a Type with no
numerical  value|


### GPIO library

   1. In your words, describe the difference between the declaration and
      the definition of the function in C.
   2.
   * Function declaration : the line where name, arguments and data type
are defined

   * Function definition : the code that define what the function will do

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:


```c
    #define LED_GREEN   PB5     // AVR pin where green LED is connected
#define LED_RED  PC6
#define PUSH_BUTTON PD7
#define BLINK_DELAY 500
#ifndef F_CPU
# define F_CPU 16000000     // CPU frequency in Hz required for delay
#endif

#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

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

    GPIO_config_output(&DDRC, LED_RED);
    GPIO_write_low(&PORTC, LED_RED);

    // Configure Push button at port D and enable internal pull-up resistor

    GPIO_config_input_pullup(&DDRD,PUSH_BUTTON)

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

       if (bit_is_set(PIND, 7)) {
          if (bit_is_set(PINB,5)){
             GPIO_write_low (&DDRB,LED_GREEN);
              GPIO_writ_high(&DDRC,LED_RED);
            }
           else {
             GPIO_write_low(&DDRC,LED_RED);
              GPIO_writ_high(&DDRB,LED_GREEN);
            }
         }
    }

    // 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!

```
   ![your figure]()
```