Digital-electronics-2 / Labs / 03-gpio / Assignment.md

🔷 UTAN25 Update Assignment.md   ↻

👥 **1 contributor**

☰ 61 lines (43 sloc) │ 2.74 KB   ...

Lab 3: Unai Telletxea

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

https://github.com/UTAN25/Digital-electronics-2/

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 integrer
uint16_t	16	0,...,65535	Unsigned 8-bit integer
int16_t	16	-32768,...,32767	Signed 16-bit integrer
float	32	-3.4e+38, ..., 3.4e+38	Single-precision floating-point
void	0	-	Nothing

GPIO library

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

   - Function declaration specifies the name of the functions, which parameters will be needed and the type of return the program should expect.
   - Function definition is the body of the function which tells the program what should the function do with the given parameters.

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
// Configure Push button at port D and enable internal pull-up resistor
GPIO_config_input_pullup(&DDRD, PUSH); //bottom is connected to GND

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

    while(!GPIO_read(&PORTD,PUSH))              //Starts an internal loop while
    {
        GPIO_write_high(&PORTB, LED_2);          //Sets LED2 ON and LED_GREEN
        GPIO_write_low(&PORTB, LED_GREEN);
        _delay_ms(BLINK_DELAY);

        if(!GPIO_read(&PORTD,PUSH))              //Ensures that the bottom is
        {                                        //If it is not it will jump
            GPIO_write_low(&PORTB, LED_2);
            GPIO_write_high(&PORTB, LED_GREEN);    //Sets LED_GREEN ON and L
            _delay_ms(BLINK_DELAY);
        }
    }
}
```

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!

Traffic — atmega 328P AVR — Resistor — LED — Pedestrian

RED — 100Ω — PD7 — PB5 — 100Ω — LED RED
YELLO — 100Ω — PD6 — PB4 — 100Ω — LED GREEN
GREEN — 100Ω — PD5 — PB3

Vcc — GND — push (pull up)

5V — Power