



xbarto0c Update Assignment.md



1 contributor



99 lines (79 sloc) | 3.14 KB



Lab 2: Jan Bartoň

Link to your Digital-electronics-2 GitHub repository:

<https://github.com/xbarto0c/Digital-electronics-2>

Active-low and active-high LEDs

1. Complete tables according to the AVR manual.

DDRB	Description
0	Input pin
1	Output pin

PORTB	Description
0	Output low value
1	Output high value

DDRB	PORTB	Direction	Internal pull-up resistor	Description
0	0	input	no	Tri-state, high-impedance

DDRB	PORTB	Direction	Internal pull-up resistor	Description
0	1	input	yes	Pin set as input, with a pullup resistor attached
1	0	output	no	Output Low (sink)
1	1	output	no	Output High (Source)

2. Part of the C code listing with syntax highlighting, which blinks alternately with a pair of LEDs; let one LED is connected to port B and the other to port C:

```
int main(void)
{
    // Green LED at port B
    // Set pin as output in Data Direction Register...
    DDRB = DDRB | (1<<LED_GREEN);
    // ...and turn LED off in Data Register
    PORTB = PORTB & ~(1<<LED_GREEN);

    // Configure the second LED at port C
    // WRITE YOUR CODE HERE
    DDRC = DDRC | (1<<LED_BLUE);      // Set the pin as output
    PORTC = ~(PORTC & (1<<LED_BLUE));  // Set the LED pin "HIGH" (LED off)
    // Infinite loop
    while (1)
    {
        // Pause several milliseconds
        _delay_ms(BLINK_DELAY);
        // WRITE YOUR CODE HERE
        PORTB = PORTB ^ (1<<LED_GREEN);
        _delay_ms(BLINK_DELAY);
        PORTC = PORTC ^ (1<<LED_BLUE);
        _delay_ms(BLINK_DELAY);
        PORTB = PORTB ^ (1<<LED_GREEN);
        _delay_ms(BLINK_DELAY);
        PORTC = PORTC ^ (1<<LED_BLUE);
    }

    // Will never reach this
    return 0;
}
```

Push button

1. 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. Let the push button is connected to port D:

```

// Configure Push button at port D and enable internal pull-up resistor
// WRITE YOUR CODE HERE
DDRD = DDRD | (0<<PUSHBUTTON); // Set the pin as input
PORTD = PORTD | (1<<PUSHBUTTON); // Attach pullup resistor to the pushbutton
// Infinite loop
while (1)
{
    // WRITE YOUR CODE HERE
    if(bit_is_clear(PIND, PUSHBUTTON)) // Reading the input state of the push
        // if low, blink the LEDs
    {
        _delay_ms(BLINK_DELAY); // Blink the LEDs in an alternating manner
        PORTB = PORTB ^ (1<<LED_GREEN);
        _delay_ms(BLINK_DELAY);
        PORTC = PORTC ^ (1<<LED_BLUE);
        _delay_ms(BLINK_DELAY);
        PORTB = PORTB ^ (1<<LED_GREEN);
        _delay_ms(BLINK_DELAY);
        PORTC = PORTC ^ (1<<LED_BLUE);
        loop_until_bit_is_clear(PIND, PUSHBUTTON); // Pushbutton debounce
    }
}

```

Knight Rider

1. Scheme of Knight Rider application, i.e. connection of AVR device, five LEDs, resistors, one push button, and supply voltage. The image can be drawn on a computer or by hand. Always name all components and their values!

