amwellius / Digital-electronics-2 Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

Digital-electronics-2 / Labs / 02-leds / README.md

amwellius Update README.md

1 contributor

Lab 2: Samuel Košík

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

GitHub Link

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	Pxn will source current if ext. pulled low.
1	0	Output	No	Low Output
1	1	Output	No	High Output

Port	Pin	Input/output usage?
A	x	Microcontroller ATmega328P does not contain port A
B	0	Yes (Arduino pin 8)
	1	pin 9
	2	pin 10
	3	pin 11
	4	pin 12
	5	pin 13
	6	NO
	7	NO
C	0	Yes (Arduino pin A0)
	1	pin A1
	2	pin A2
	3	pin A3
	4	pin A4
	5	pin A5
	6	NO
	7	NO
D	0	Yes (Arduino pin RX<-0)
	1	TX ->0
	2	pin 2
	3	pin 3

Port	Pin	Input/output usage?
	4	pin 4
	5	pin 5
	6	pin 6
	7	pin 7

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:

```c
int main(void)
{

    DDRB = DDRB | (1<<LED_GREEN);
    DDRC = DDRC | (1<<LED_RED);

    // Infinite loop
    while (1)
    {
        // Pause several milliseconds
        _delay_ms(BLINK_DELAY);
        PORTC = PORTC ^ (1<<LED_RED);       //LED ON (toogle OFF)
        _delay_ms(BLINK_DELAY);
        PORTB = PORTB ^ (1<<LED_GREEN);     //LED ON (toogle OFF)

    }

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

Push button

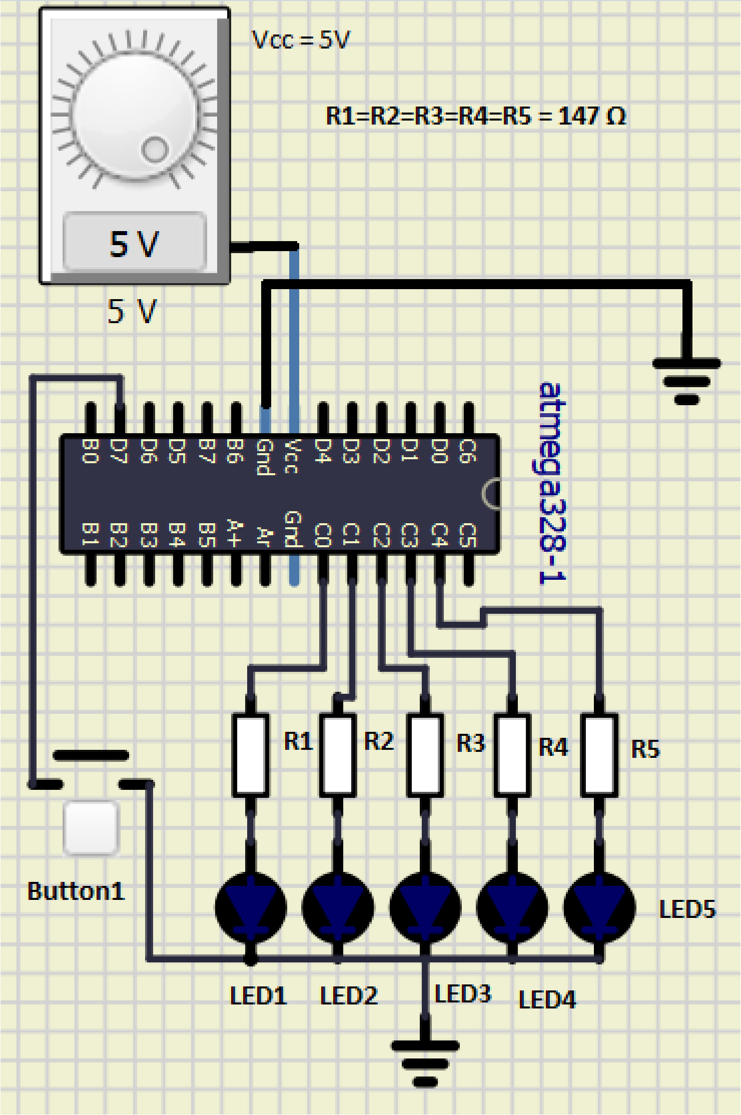
1. Part of the C code listing with syntax highlighting, which toggles LEDs only if push button is pressed. Otherwise, the value of the LEDs does not change. Let the push button is connected to port D:

```c
int main(void)
{
    DDRB = DDRB | (1<<LED_GREEN);
    DDRC = DDRC | (1<<LED_RED);

        // Configure Push button at port D and enable internal pull-up resistor
        DDRD = DDRD & ~(1<<BUTTON);
        PORTD = PORTD | (1<<BUTTON);        //pullup enabled
    // Infinite loop
    while (1)
    {
        if(bit_is_clear(PIND, PD7))
        {
            _delay_ms(BLINK_DELAY);
            //led toogling only if button is pressed
            PORTC = PORTC ^ (1<<LED_RED);       //LED ON (toogle OFF)
            PORTB = PORTB ^ (1<<LED_GREEN);     //LED ON (toogle OFF)
        }

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

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

119 lines (93 sloc) | 3.02 KB



Vcc = 5V
R1=R2=R3=R4=R5 = 147 Ω
5 V
5 V
atmega328-1
B0
D7
D6
D5
B7
B6
Gnd
Vcc
D4
D3
D2
D1
D0
C6
B1
B2
B3
B4
B5
A+
Ar
Gnd
C0
C1
C2
C3
C4
C5
R1
R2
R3
R4
R5
Button1
LED1
LED2
LED3
LED4
LED5