

*Table for DDRB*

DDRB	Description
0	Input pin
1	Output pin

*Table for PORTB*

PORTB	Description
0	Output low value
1	Output high value

*Table for the combination of DDRB and PORTB*

DDRB	PORTB	Direction	Internal pull-up resistor	Description
0	0	Input	No	Tri-state, high impedance
0	1	Input	Yes	Pxn will source current if ext. pulled low
1	0	Output	No	Output low (Sink)
1	1	Output	No	Output High (Source)

*Table with input/output pins available on ATmega328p*

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

```

//                                     Knight Rider Style
/* Name: Nadir Osman Al-Wattar*/

/*****
 *
 * Alternately toggle two LEDs when a push button is pressed.
 * ATmega328P (Arduino Uno), 16 MHz, AVR 8-bit Toolchain 3.6.2
 *
 * Copyright (c) 2018-2020 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_RED      PC0
#define LED_GREEN    PB5
#define LED_YELLOW   PB4
#define LED_ORANGE   PB3
#define LED_BLUE     PB2
#define BTN          PD0

#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

/* Functions -----*/
/**
 * Main function where the program execution begins. Toggle two LEDs
 * when a push button is pressed.
 */
int main(void)
{
    DDRB = DDRB | (1<<LED_GREEN);
    DDRB = DDRB | (1<<LED_ORANGE);
    DDRB = DDRB | (1<<LED_BLUE);
    DDRB = DDRB | (1<<LED_YELLOW);
    DDRC = DDRC | (1 << LED_RED );

    PORTB = PORTB & ~(1<<LED_GREEN);
    PORTB = PORTB & ~(1<<LED_ORANGE);
    PORTB = PORTB & ~(1<<LED_BLUE);
    PORTB = PORTB & ~(1<<LED_YELLOW);
    PORTC = PORTC & ~(1<<LED_RED);

    DDRD = DDRD & ~(1<<BTN); //Turn OFF
    PORTD = PORTD | (1<<BTN);

    // Infinite loop

    while (1)
    {

```

```

        // Pause several milliseconds

//1
if (bit_is_clear(PIND, BTN))
{
    PORTC=PORTC ^ (1<<LED_RED);
    _delay_ms(BLINK_DELAY);
    PORTC = PORTC & ~(1<<LED_RED);
}

//2
if (bit_is_clear(PIND, BTN))
{
    PORTB=PORTB ^ (1<<LED_GREEN);
    _delay_ms(BLINK_DELAY);
    PORTB = PORTB & ~(1<<LED_GREEN);
}

//3
if (bit_is_clear(PIND, BTN))
{
    PORTB=PORTB ^ (1<< LED_YELLOW);
    _delay_ms(BLINK_DELAY);
    PORTB = PORTB & ~(1<<LED_YELLOW);
}

//4
if (bit_is_clear(PIND, BTN))
{
    PORTB=PORTB ^ (1<< LED_ORANGE);
    _delay_ms(BLINK_DELAY);
    PORTB = PORTB & ~(1<<LED_ORANGE);
}

//5
if (bit_is_clear(PIND, BTN))
{
    PORTB=PORTB ^ (1<< LED_BLUE);
    _delay_ms(BLINK_DELAY);
    PORTB = PORTB & ~(1<<LED_BLUE);
}

//4
if (bit_is_clear(PIND, BTN))
{
    PORTB=PORTB ^ (1<< LED_ORANGE);
    _delay_ms(BLINK_DELAY);
    PORTB = PORTB & ~(1<<LED_ORANGE);
}

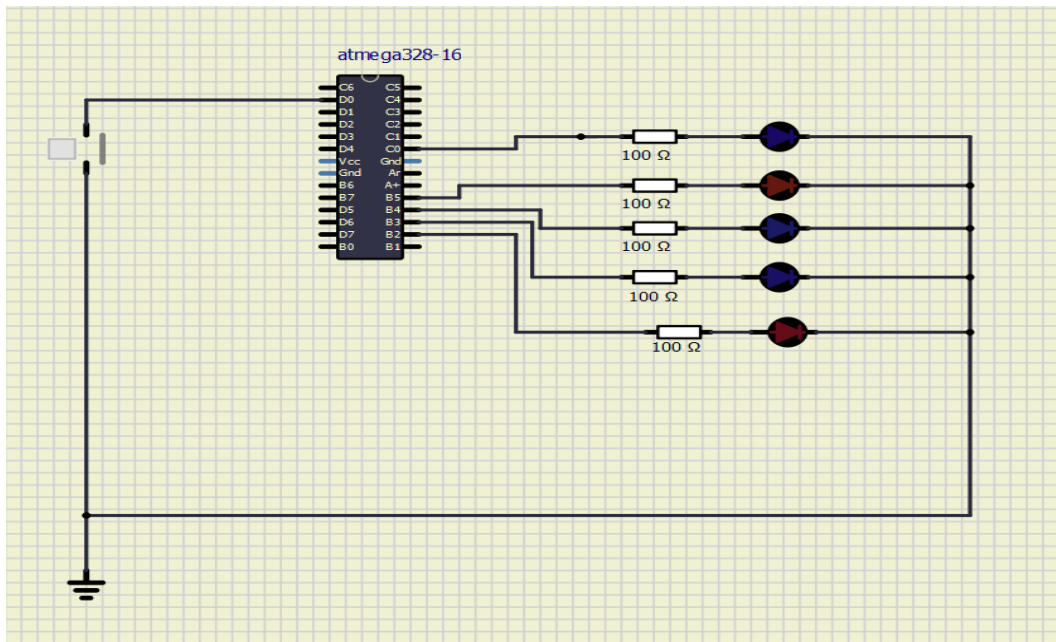
//3
if (bit_is_clear(PIND, BTN))
{
    PORTB=PORTB ^ (1<< LED_YELLOW);
    _delay_ms(BLINK_DELAY);
    PORTB = PORTB & ~(1<<LED_YELLOW);
}

//2
if (bit_is_clear(PIND, BTN))
{
    PORTB=PORTB ^ (1<<LED_GREEN);
    _delay_ms(BLINK_DELAY);
    PORTB = PORTB & ~(1<<LED_GREEN);
}

_delay_ms(50);

    }
    // Will never reach this
    return 0;
}

```



```
// Two LEDs and Push button

/* Defines -----*/
#define LED_GREEN PB5 // AVR pin where green LED is connected
#define LED_RED PC0
#define BTN PD0
#define BLINK_DELAY 250
#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

/* Functions -----*/
/**
 * Main function where the program execution begins. Toggle two LEDs
 * when a push button is pressed.
 */
int main(void)
{
    /* GREEN LED */
    // 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);

    /*RED LED*/
    DDRC = DDRC | (1<<LED_RED);
    PORTC = PORTC & ~(1<<LED_RED);

    DDRD = DDRD & ~(1<<BTN); //Turn OFF
    PORTD = PORTD | (1<<BTN);

    // Infinite loop
    while (1)
    {
        // Pause several milliseconds
    }
}
```

```

_delay_ms(BLINK_DELAY);

// WRITE YOUR CODE HERE
if (bit_is_clear(PIND, BTN))
{

// WRITE YOU CODE HERE
PORTB = PORTB ^ (1<<LED_GREEN);
PORTC = PORTC ^ (1<<LED_RED);
}

}

// Will never reach this
return 0;
}

```

