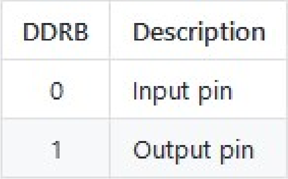
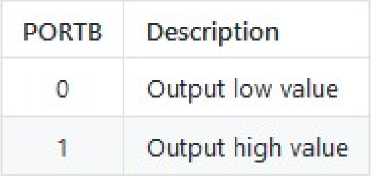
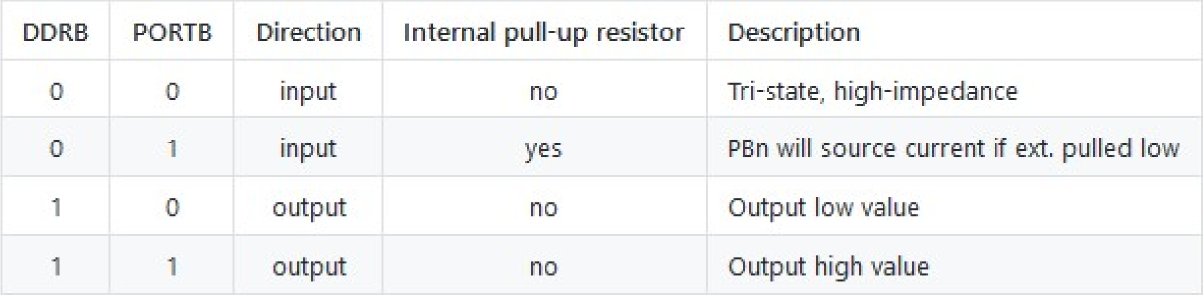
1.



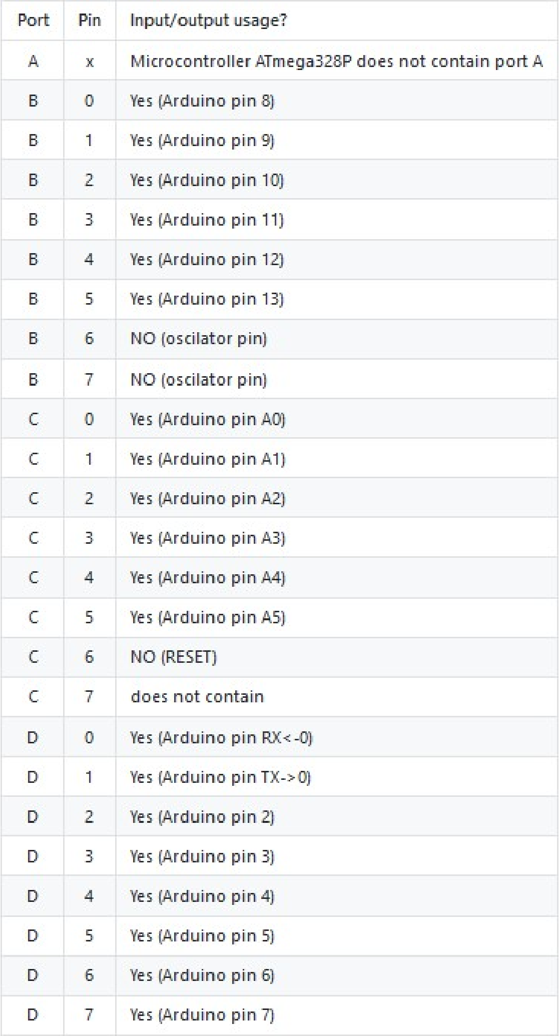
DDRB	Description
0	Input pin
1	Output pin



PORTB	Description
0	Output low value
1	Output high value

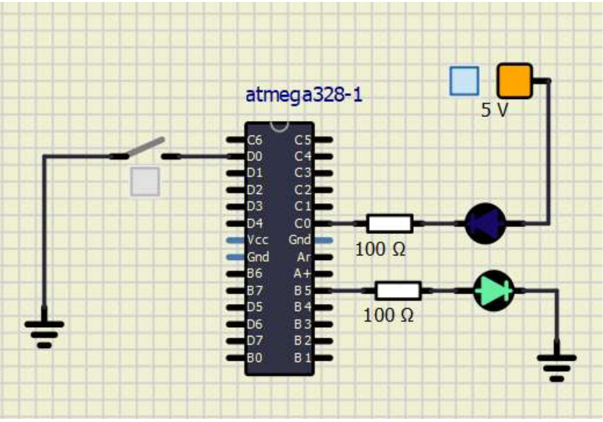


DDRB	PORTB	Direction	Internal pull-up resistor	Description
0	0	input	no	Tri-state, high-impedance
0	1	input	yes	PBn will source current if ext. pulled low
1	0	output	no	Output low value
1	1	output	no	Output high value



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

```
/* Defines -----------------------------------------------------------*/
#define LED_GREEN   PB5     // AVR pin where green LED is connected
#define LED_RED PC0 // AVR pin where red LED is connected
#define PUSH_BUTTON PD0 // AVR pin where pushbutton is connected
#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)
{
    /* GREEN LED */
    DDRB |= (1<<LED_GREEN); // Set pin as output in Data Direction Register...
       PORTB &= ~(1<<LED_GREEN); // ...and turn LED off in Data Register
       /* RED LED */
       DDRC |= (1<<LED_RED); // Set pin as output in Data Direction Register...
       PORTC &= ~(1<<LED_RED); // ...and turn LED off in Data Register
       /* PUSHBUTTON */
       DDRD &= ~(1<<PUSH_BUTTON); // Set pin as input in Data Direction Register...
       PORTD |= (1<<PUSH_BUTTON); // Set pull-up resistor

           // Infinite loop
    while (1)
    {
              if(bit_is_clear(PIND, PUSH_BUTTON)) //if button is pressed blink with
LEDs
              {
        // Pause several milliseconds
        _delay_ms(BLINK_DELAY);
              PORTB ^= (1<<LED_GREEN);
              PORTC ^= (1<<LED_RED);
              }

    }

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



2.

```
/* Defines -----------------------------------------------------------*/

#define BLINK_DELAY 250 //speed of blinking
#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

int main(void)
{
       //LEDs are on pins PC5 to PC0 (6 total) alternative DDRC |=
(1<<PC5)|(1<<PC4)|(1<<PC3)...
       DDRC |= (0b00111111); //set all LEDs as output
       PORTC &= (0b11000000); //set off all LEDs

       uint8_t u = 0b00010000;
    // Infinite loop
    while (1)
    {
              //rewriting PC7 (not used) and PC6 (defined as input with ext. pull up)
does not matter

              //shift lighting LED to left
              for(uint8_t i = 0; i<6; i++)
              {
              PORTC = (1<<i);
              _delay_ms(BLINK_DELAY);
              }
              //shift lighting LED to right
              for(int8_t i = 0; i<4; i++)
              {
                     PORTC = (u>>i);
                     _delay_ms(BLINK_DELAY);
              }
    }
    // Will never reach this
    return 0;
}
```