Lab 2: Control of GPIO, LED, push button– David Garcia Torre

1. Tables for DDRB, PORTB, and their combination.

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	No	Tri-state High-impedance
0	1	input	Yes	Pxn Will source current if ext. pulled low
1	0	output	No	Output Low
1	1	output	No	Output High

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	No
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)

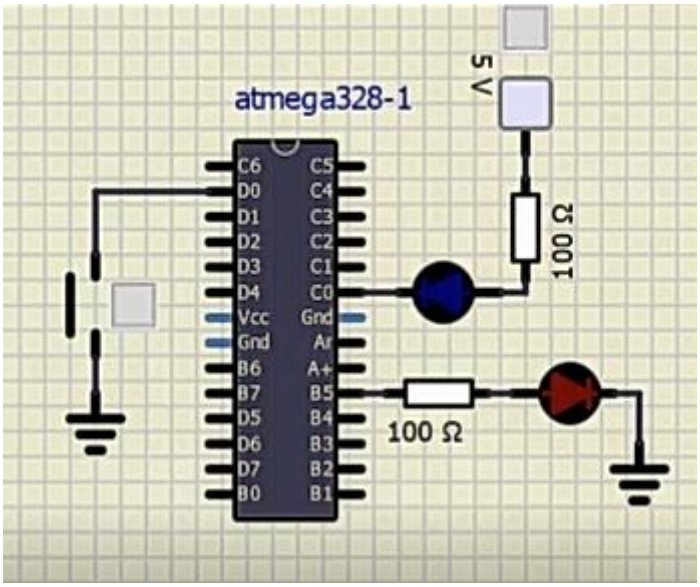
7 Yes (Arduino pin 7)

Listing of C code with two LEDs and a push button

```
/*
 * lab2.c

 * Author : TheGT23
 *
 * 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_GREEN   PB5     // AVR pin where green LED is connected
#define LED_RED   PC0     // AVR pin where red LED is connected
#define BUTTON   PD0     // AVR pin where the button is connected

#define SHORT_DELAY 500      // Delay in milliseconds
#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 one LED
 * and use function from the delay library.
 */

int main(void){

    // Set pin as output in Data Direction Register
    // DDRB = DDRB or 0010 0000
    DDRB = DDRB | (1<<LED_GREEN);

    // Set pin LOW in Data Register (LED off)
    // PORTB = PORTB and 1101 1111
    PORTB = PORTB & ~(1<<LED_GREEN);


       // Set pin as output in Data Direction Register
       // DDRC = DDRC or 0010 0000
       DDRC = DDRC | (1<<LED_RED);


       // Set pin LOW in Data Register (LED off)
       // PORTC = PORTC and 1101 1111
       PORTC = PORTC & ~ (1<<LED_RED);
```

```
    /*PUSH BUTTON*/

    DDRD = DDRD & ~(1<<BUTTON);  // input
    PORTD = PORTD | (1<<BUTTON); // enable internal pull up

  // Infinite loop

  while (1) {

      // Pause several milliseconds
      _delay_ms(SHORT_DELAY);

          if(bit_is_clear(PIND,BUTTON)){

              // Invert LED in Data Register
              // PORTB = PORTB xor 0010 0000
              PORTB = PORTB ^ (1<<LED_GREEN);
              PORTC = PORTC ^ (1<<LED_RED);

          }

  }

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

Screenshot of SimulIDE circuit



Knight Rider application. Submit:

- Listing of C code.

```
/*****************************************************************
 * Author: David Garcia Torre
 *****************************************************************/

/* Defines -----------------------------------------------------------*/
#define LED_RED1    PC0     // AVR pin where red LED 1 is connected
#define LED_RED2    PC1     // AVR pin where red LED 2 is connected
#define LED_RED3    PC2     // AVR pin where red LED 3 is connected
#define LED_RED4    PC3     // AVR pin where red LED 4 is connected
#define LED_RED5    PC4     // AVR pin where red LED 5 is connected
#define LED_RED6    PC5     // AVR pin where red LED 6 is connected
#define BUTTON          PD0     // AVR pin where the button is connected

#define SHORT_DELAY 250      // Delay in milliseconds

#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.
 */

int leds[] ={LED_RED1,LED_RED2,LED_RED3,LED_RED4,LED_RED5,LED_RED6};
int a=0,b=0;

int main(void)
{


       // Set pin as output in Data Direction Register

       DDRC = DDRC | (1<<LED_RED1);
       DDRC = DDRC | (1<<LED_RED2);
       DDRC = DDRC | (1<<LED_RED3);
       DDRC = DDRC | (1<<LED_RED4);
       DDRC = DDRC | (1<<LED_RED5);
       DDRC = DDRC | (1<<LED_RED6);


       // Set pin LOW in Data Register (LED off)

       PORTC = PORTC | (1<<LED_RED1);
       PORTC = PORTC | (1<<LED_RED2);
       PORTC = PORTC | (1<<LED_RED3);
       PORTC = PORTC | (1<<LED_RED4);
       PORTC = PORTC | (1<<LED_RED5);
       PORTC = PORTC | (1<<LED_RED6);
```

```
        /*PUSH BUTTON*/

        DDRD = DDRD & ~(1<<BUTTON);  // input
        PORTD = PORTD | (1<<BUTTON); // enable internal pull up

    // Infinite loop

    for (;;){

        // Pause several milliseconds
        _delay_ms(SHORT_DELAY);

              PORTC = PORTC | (1<<leds[a]);

              if(bit_is_clear(PIND,BUTTON)){ //we select the direction with this
if

                 if(a == 5){

                        b = 1;
                        PORTC = PORTC | (1<<leds[5]);

                 }else if(a == 0){

                        b = 0;
                        PORTC = PORTC | (1<<leds[0]);

                 } // we rest one to a unless if b is 0 that we add one

                 if(b == 0){

                        a++;

                 }else{

                        a--;

                 }

              PORTC = PORTC & ~(1<<leds[a]);

              }

              }
    return 0;
}
```