

TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA
GETALDIĆEVA ULICA 4
10 000 ZAGREB

UGRADBENI RAČUNALNI SUSTAVI
LV02 – UART SERIJSKA KOMUNIKACIJA

Josip Prpić 4.H
08.05.2020
Gajnice, Zagreb

Sadržaj

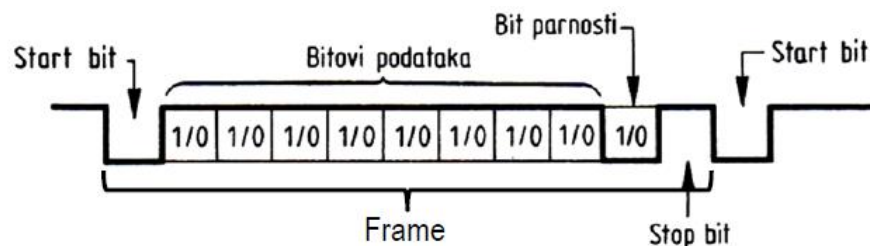
Sadržaj

Sadržaj.....	2
1. Uart serijska komunikacija.....	3
1.1 Osnove serijske komunikacije[1]	3
1.2 Dvosmjerna komunikacija.....	4
2. <i>Terminal emulator program</i>	5
3. LV02 – UART serijska komunikacija prevedena u AVR-C.....	6
3.1 Zadatak 1.....	6
3.2 Zadatak 2.....	7
3.3 Zadatak 3.....	8
3.4 Zadatak 4.....	10
3.5 Zadatak 5.....	12
3.6 Zadatak 6.....	14
3.7 Zadatak 7.....	16
3.8 Zadatak 8.....	17
3.9 Zadatak 9.....	20
3.10. Zadatak 10.....	22
4. Literatura	24

1. Uart serijska komunikacija

1.1 Osnove serijske komunikacije[1]

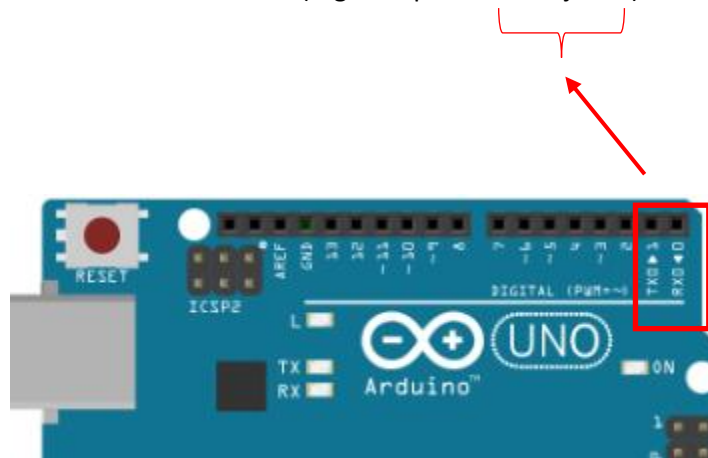
UART (engl. *Universal asynchronous receiver/transmitter*) je serijska komunikacija koja funkcioniра na način da se niz signala prenosi kroz vodič u određenom vremenskom slijedu između pošiljalca (engl. *Transmitter*) i primatelja (engl. *Receiver*). U jednome trenutku moguće je prenijeti samo jedan jedini bit (signal + GND).



Princip rada serijske komunikacije je slijedeći:

- Na predajniku (TX) paralelne podatke pretvaramo u niz serijskih podataka
- Na prijatelju (RX) originalni bajt ponovno grupiramo i vraćamo u prvobitni oblik
- Na taj način se smanjuje broj potrebnih vodiča i pinova, što znatno povećava preglednost sklopa
- Status prikazuje stanje izlaznih i ulaznih registara (engl. *buffer*) što se koristi za kontrolu protoka podataka (engl. *flow control*)

RX i TX pinovi se na arduinu nalaze na D0 i D1 (Digitalni pinovi nula i jedan):

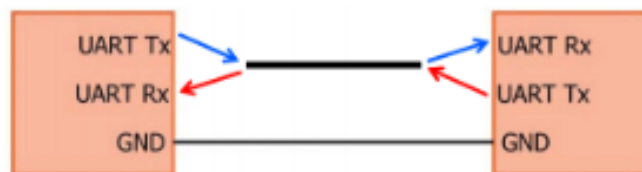


1.2 Dvosmjerna komunikacija

Ovisno o broju vodiča, razlikujemo dvije vrste serijske dvosmjerne komunikacije:

1) Half duplex

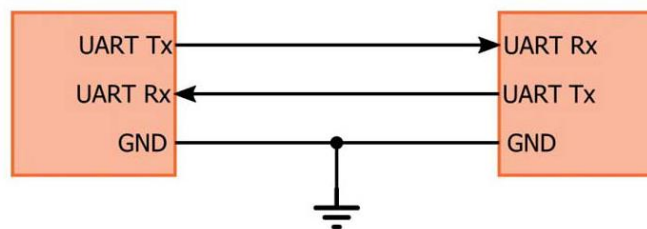
- Half duplex komunikacija omogućava komunikaciju u samo jednom smjeru u jednom trenutku
- Za dvosmjernu komunikaciju potrebno je cijelo vrijeme dijeliti kanal, te je potrebno, na kraju prijenosa, uvijek imati *stop – bit*, koji označava kraj komunikacije u jednome smjeru
- Primjer je I²C komunikacija



Slika 1: Half duplex komunikacija

2) Full duplex komunikacija

- Kako bi oba uređaja mogla funkcionirati i kao predajnik i kao prijatelj, oba moraju posjedovati i *transmitter* i *receiver*
- Komunikacija se može odvijati istovremeno u oba smjera, ali su za to potrebna dva vodiča



Slika 2: Full duplex komunikacija

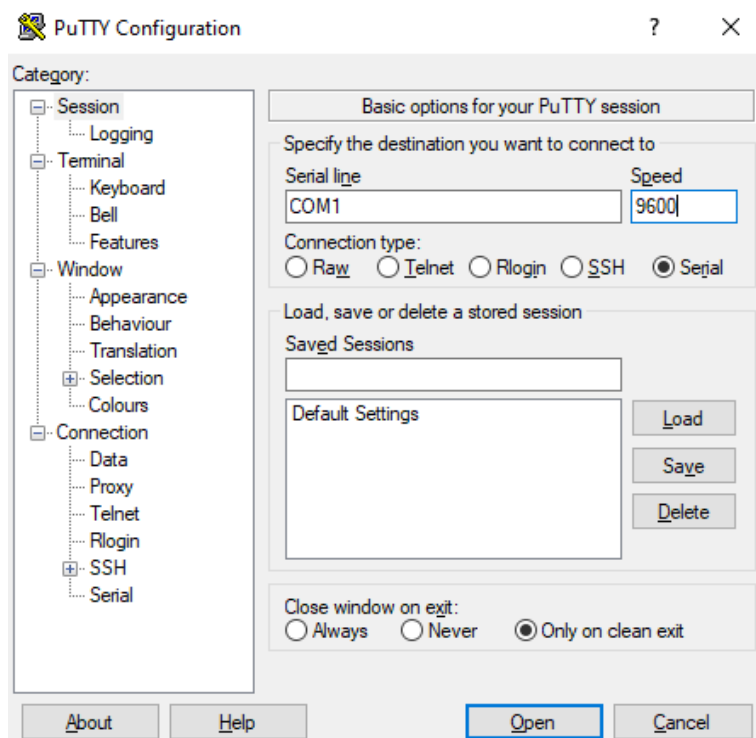
2. Terminal emulator program

Kako bi računalo moglo komunicirati sa vanjskim uređajem potrebno je koristiti neki od *Terminal emulator* programa. Arduino ima svoj ugrađeni Serial Monitor, dok je programčić PuTTY [2] odličan primjer jednostavnog programa koji se može i „bootati“ sa USB pogona.

Kako bi program mogao komunicirati sa vanjskim uređajem, i program i uređaj moraju imati konfiguriranu istu brzinu prijenosa (bitova po sekundi, engl. *Bits per second, BPS*).

Brzine prijenosa mogu biti:

- 9600 bitova po sekundi
- 14400 bitova po sekundi
- 19200 bitova po sekundi
- 38400 bitova po sekundi
- 57600 bitova po sekundi
- 115200 bitova po sekundi



Slika 3: korisničko sučelje programa PuTTY

3. LV02 – UART serijska komunikacija prevedena u AVR-C

3.1 Zadatak 1.

Učitaj program na Arduino. Pomoću tipke Send ili Enter na tipkovnici, šalji znakove serijskom vezom prema Arduino. U serial monitoru pratiti stanje napunjenosti spremnika koju vraća funkcija `Serial.available()`. Provjeri koliki je spremnik, tj. Koliko znakova može prihvatiti spremnik! Što se događa kada se spremnik napuni?

<https://www.xanthium.in/how-to-avr-atmega328p-microcontroller-usart-uart-embedded-programming-avrgcc>

-kod-

3.2 Zadatak 2.

Proširi prethodni zadatak dodavanjem funkcije `Serial.parseInt()` prema priloženom kodu. Preko Serial monitora, šalji različite znakove – slova, brojeve, njihovu kombinaciju, te kombinaciju sa razmakom. U Serial monitoru prati vrijednosti koje vraća funkcija `Serial.parseInt()`.

```
#define FOSC 16000000
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
#include <avr/io.h>
#include<util/delay.h>
void USART_Init (unsigned int ubrr)
{
    /*Set baud rate */
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    /*Enable receiver and transmitter */
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    /* Set frame format: 8data, 2stop bit */
    UCSR0C = (1<<USBS0)|(3<<UCSZ00);
}
void UART_send(unsigned char data)
{
    // wait for transmit buffer to be empty
    while(!(UCSR0A & (1 << UDRE0)));

    // load data into transmit register
    UDR0 = data;
}
unsigned char USART_Receive(void)
{
    /* Wait for data to be received */
    while (!(UCSR0A & (1<<RXC0)))
        ;
    /* Get and return received data from buffer */
    return UDR0;
}
int main(void)
{
    //Serial begin
    USART_Init(MYUBRR);
    while (1)
    {
        //Message displayed
        char message[16]= "Parsed Integer: ";
        char c = USART_Receive();

        for(int i=0; i<16;i++)
        {
            UART_send(message[i]);
        }
        if(c >= '0' && c<='9') UART_send(c);
        else UART_send('0');
    }
}
```

```

        UART_send('\n');
        _delay_ms(5000);
    }
}

```

3.3 Zadatak 3.

Modificiraj prethodni zadatak dodavanjem funkcije `Serial.read()` prema priloženom kodu. Preko Serial monitora, šalji različite znakove – slova, brojeve, njihovu kombinaciju, te kombinaciju sa razmakom. U Serial monitoru prati vrijednosti koje vraća funkcija `Serial.Read()`.

```

#define FOSC 16000000
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
#include <avr/io.h>
#include<util/delay.h>
void USART_Init (unsigned int ubrr)
{
    /*Set baud rate */
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    /*Enable receiver and transmitter */
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    /* Set frame format: 8data, 2stop bit */
    UCSR0C = (1<<USBS0)|(3<<UCSZ00);
}
void UART_send(unsigned char data)
{
    // wait for transmit buffer to be empty
    while(!(UCSR0A & (1 << UDRE0)));

    // load data into transmit register
    UDR0 = data;
}
unsigned char USART_Receive(void)
{
    /* Wait for data to be received */
    while (!(UCSR0A & (1<<RXC0)))
        ;
    /* Get and return received data from buffer */
    return UDR0;
}
int main(void)
{
    //Serial begin
    USART_Init(MYUBRR);
    //Initial message
    char initMessage[40] = "Upisi znak u polje iznad i pritisni SEND";
    for (int i=0; i<40; i++)
    {
        UART_send(initMessage[i]);
    }
    UART_send('\n');
    while (1)
    {
        char c = USART_Receive();
    }
}

```



```
int val = c;
char str[2];
sprintf(str,"%d",val);

if(str[0]!='1' || str[1]!='0')
{
    char txt[14] = "Upisani znak: ";
    char txt2[20] = ", ASCII vrijednost: ";
    for (int i=0; i<14; i++)
    {
        UART_send(txt[i]);
    }
    UART_send(c);
    for (int i=0; i<20; i++)
    {
        UART_send(txt2[i]);
    }
    UART_send(str[0]);
    UART_send(str[1]);
    UART_send('\n');
}
}
```

3.4 Zadatak 4.

Ponovi testove iz prethodnog zadatka koristeći Serial monitor i prati što vraća funkcija `Serial.read()` u ovisnosti o vrijednosti podešenoj u padajućem izborniku: „No line ending“, „Newline“, „Carriage return“, „both NL & CR“. Usporedi primljene ASCII vrijednosti sa danom ASCII tablicom (ASCII tablica se nalazi u poglavlju Prilozi).

```
#define FOSC 16000000
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
#include <avr/io.h>
#include<util/delay.h>
void USART_Init (unsigned int ubrr)
{
    /*Set baud rate */
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    /*Enable receiver and transmitter */
    UCSRB = (1<<RXEN0)|(1<<TXEN0);
    /* Set frame format: 8data, 2stop bit */
    UCSRC = (1<<USBS0)|(3<<UCSZ00);
}
void UART_send(unsigned char data)
{
    // wait for transmit buffer to be empty
    while(!(UCSR0A & (1 << UDRE0)));

    // load data into transmit register
    UDR0 = data;
}
unsigned char USART_Receive(void)
{
    /* Wait for data to be received */
    while (!(UCSR0A & (1<<RXC0)))
    ;
    /* Get and return received data from buffer */
    return UDR0;
}
int main(void)
{
    //Serial begin
    USART_Init(MYUBRR);
    //Initial message
    char initMessage[40] = "Upisi znak u polje iznad i pritisni SEND";
    for (int i=0; i<40; i++)
    {
        UART_send(initMessage[i]);
    }
    UART_send('\n');

    while (1)
    {
        char c = USART_Receive();
        int val = c;
```

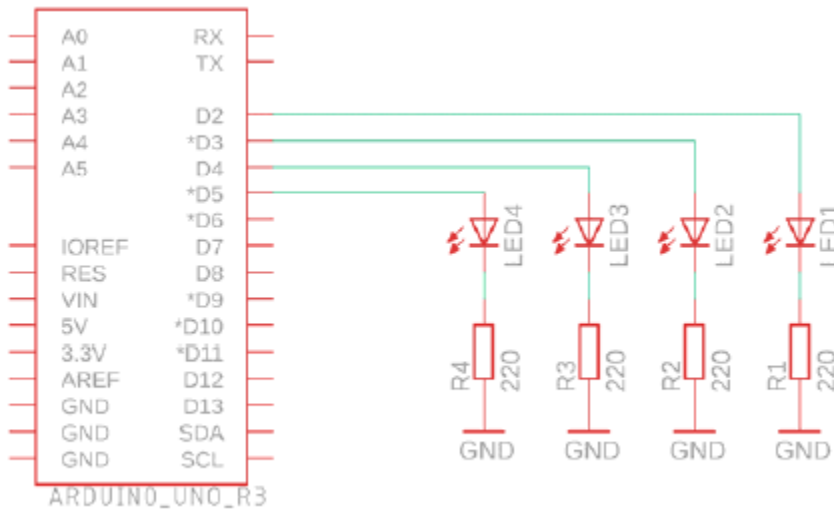
```
char str[2];
sprintf(str,"%d",val);

if(str[0]!='1' || str[1]!='0')
{
    char txt[14] = "Upisani znak: ";
    char txt2[20] = ", ASCII vrijednost: ";

    for (int i=0; i<14; i++)
    {
        UART_send(txt[i]);
    }
    UART_send(c);
    for (int i=0; i<20; i++)
    {
        UART_send(txt2[i]);
    }
    UART_send(str[0]);
    UART_send(str[1]);
    UART_send('\n');
}
}
```

3.5 Zadatak 5.

Spoji 4 LED diode prema shemi. Koristeći Serial monitor omogućiti uključivanje ili isključivanje LED diode priključene na pin 2.



```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#define BAUDRATE 9600
#define BAUD_PRESCALER (((F_CPU / (BAUDRATE * 16UL))) - 1)

//deklaracije funkcija
void USART_init(void);
unsigned char USART_receive(void);
void USART_send( unsigned char data);
void USART_putstring(char* StringPtr);

char String[]="";
unsigned char data;
int main(void){
    DDRD |= 0b00000100;
    PORTD=0x00;
    USART_init();
    while(1){
        data=USART_receive();
        USART_send(data);
        if (data=='1')
        {
            PORTD|=(1<<PORTD2);
            _delay_ms(300);
        }
        else
        {
            PORTD&=~(1<<PORTD2);
        }
    }
}
```

```

    }
}
return 0;
}

// Deklaracija dodatnih funckija

void USART_init(void){

    UBRR0H = (uint8_t)(BAUD_PRESCALLER>>8); // Baud Rate Register upper 4 bits
    UBRR0L = (uint8_t)(BAUD_PRESCALLER); // Baud Rate Register lower 4 bits
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    UCSR0C = (3<<UCSZ00); //Set frame
}
unsigned char USART_receive(void){

    while(!(UCSR0A & (1<<RXC0)));
    return UDR0;
}
void USART_send( unsigned char data)
{
    while(!(UCSR0A & (1<<UDRE0)));
    UDR0 = data;

}

void USART_putstring(char* StringPtr){

    while(*StringPtr != 0x00){
        USART_send(*StringPtr);
        StringPtr++;}

}

```

3.6 Zadatak 6.

Koristeći Serial monitor omogućiti uključivanje i isključivanje svake LED diode zasebno spojene na pinove 2, 3, 4, 5. Naredbu iz Serial monitora slati u obliku „2 1“ gdje prva vrijednost predstavlja broj pina, a druga vrijednost (0 = LOW, 1 = HIGH).

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL))) - 1)

//Deklaracija funkcija
void USART_init(void);
unsigned char USART_receive(void);
void USART_send( unsigned char data);
void USART_putstring(char* StringPtr);

char String[]="";
unsigned char data,a;
int main(void){
    DDRD |= 0b00111100;
    PORTD=0x00;
    USART_init();
    while(1){
        data=USART_receive();
        USART_send(data);

        if (data=='2 1')
        {
            PORTD|=(1<<PORTD2);
        }
        if (data == '2 0')
        {
            PORTD&=~(1<<PORTD2);
        }
        if (data=='3 1')
        {
            PORTD|=(1<<PORTD3);
        }
        if (data == '3 0')
        {
            PORTD&=~(1<<PORTD3);
        }
        if (data=='4 1')
        {
            PORTD|=(1<<PORTD4);
        }
        if (data == '4 0')
        {
            PORTD&=~(1<<PORTD4);
        }
    }
}
```

```

    }
    if (data=='5 1')
    {
        PORTD|=(1<<PORTD5);
    }

    if (data == '5 0')
    {
        PORTD&=~(1<<PORTD5);
    }
}
return 0;
}

// Deklaracija korištenih funkcija
void USART_init(void){

    UBRR0H = (uint8_t)(BAUD_PRESCALLER>>8); // Baud Rate Register upper 4 bits
    UBRR0L = (uint8_t)(BAUD_PRESCALLER); // Baud Rate Register lower 4 bits
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    UCSR0C = (3<<UCSZ00); //Set frame
}

unsigned char USART_receive(void){

    while(!(UCSR0A & (1<<RXC0)));
    return UDR0;
}
void USART_send( unsigned char data)
{
    while(!(UCSR0A & (1<<UDRE0)));
    UDR0 = data;
}
void USART_putstring(char* StringPtr){

    while(*StringPtr != 0x00){
        USART_send(*StringPtr);
        StringPtr++;
    }
}

```

3.7 Zadatak 7.

Koristeći Serial monitor omogućiti promjenu svjetline LED diode na pinovima 3 i 5. Umjesto vrijednosti 0 = LOW i 1 = HIGH, u kodu prihvaćati vrijednosti 0 – 255. Naredbu iz Serial monitora slati u obliku „3 255“, „3 127“ ili slično [6].

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#define BAUDRATE 9600
#define BAUD_PRESCALER (((F_CPU / (BAUDRATE * 16UL))) - 1)

//Deklaracija funkcija
void USART_init(void);
unsigned char USART_receive(void);
void USART_send( unsigned char data);
void USART_putstring(char* StringPtr);

char String[]="";
unsigned char data;
uint8_t brightness;

int main(void)
{
    DDRD |= 0b00111100;
    PORTD=0x00;

    USART_init(); //inicijalizacija serijske komunikacije

    while (1)
    {
        data=USART_receive();
        USART_send(data);

        brightness=(data%100 - '0')*100 + (data%10 - '0')*10 + (data - '0');

        OCR2B = brightness;

        _delay_ms(10);
    }
}

void pwm_init() //funkcija za promjenu svjetine
{
    // initialize timer0 in PWM mode
    TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM21) | _BV(WGM20);
    TCCR2B = _BV(WGM22) | _BV(CS20);
}

//funkcije potrebne za UART komunikaciju

void USART_init(void){
```



```

    UBR0H = (uint8_t)(BAUD_PRESCALER>>8); // Baud Rate Register upper 4 bits
    UBR0L = (uint8_t)(BAUD_PRESCALER); // Baud Rate Register lower 4 bits
    UCSRB = (1<<RXEN)|(1<<TXEN);
    UCSRC = (3<<UCSZ00); //Set frame
}

unsigned char USART_receive(void){
    while(!(UCSR0A & (1<<RXIF)));
    return UDR0;
}

void USART_send( unsigned char data)
{
    while(!(UCSR0A & (1<<UDIF)));
    UDR0 = data;
}

void USART_putstr(char* StringPtr){
    while(*StringPtr != 0x00){
        USART_send(*StringPtr);
        StringPtr++;
    }
}

```

3.8 Zadatak 8.

Doradi program tako da ovisno o primljenoj naredbi, možeš LED na pinovima 3 i 5 uključiti naredbom digitalWrite() ili AnalogWrite(). Umjesto naredbe koristiti broj 10 za digitalWrite(), a broj 11 za analogWrite().

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL))) - 1)

//Deklaracija funkcija
void USART_init(void);
unsigned char USART_receive(void);
void USART_send( unsigned char data);
void USART_putstring(char* StringPtr);

char String[]="";
unsigned char data;
uint8_t brightness;
uint8_t status;

int main(void)
{
    DDRD |= 0b00111100;
    PORTD=0x00;
    USART_init(); //inicijalizacija serijske komunikacije
    while (1)
    {
        data=USART_receive();
        USART_send(data);
        //broj '10' ili '11' može se nalaziti između 8. (ako je razina svjetlosti
        troznamenkasta) i 5. mjesta (ako je razina svjetlosti jednoznamenkasta) u stringu
        if((((data%10000000)*10 + (data%1000000))|(((data%1000000)*10 +
        (data%100000))|(((data%100000)*10 + (data%10000)) == '11' ){

            brightness=(data%100 - '0')*100 + (data%10 - '0')*10 + (data - '0');

            OCR2B = brightness;

            _delay_ms(10);
        }else{
            PORTD |= 0b00001000;
        }
    }
}

void pwm_init() //funkcija za promjenu svjetline
{
    // initialize timer0 in PWM mode
    TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM21) | _BV(WGM20);
    TCCR2B = _BV(WGM22) | _BV(CS20);
}

//funkcije potrebne za UART komunikaciju
```

```
void USART_init(void){
    UBRRH = (uint8_t)(BAUD_PRESCALLER>>8); // Baud Rate Register upper 4 bits
    UBRRL = (uint8_t)(BAUD_PRESCALLER); // Baud Rate Register lower 4 bits
    UCSRB = (1<<RXEN0)|(1<<TXEN0);
    UCSRC = (3<<UCSZ00); //Set frame
}

unsigned char USART_receive(void){
    while(!(UCSR0A & (1<<RXC0)));
    return UDR0;
}

void USART_send( unsigned char data)
{
    while(!(UCSR0A & (1<<UDRE0)));
    UDR0 = data;
}

void USART_putstr(char* StringPtr){
    while(*StringPtr != 0x00){
        USART_send(*StringPtr);
        StringPtr++;}
}
```

3.9 Zadatak 9.

Doradi program tako da uz odgovarajuću naredbu, preko Serial monitora, na Serial monitor ispisuješ niz znakova, npr. Svoje ime i prezime. Umjesto naredbe za tekst koristiti broj 12. Završetak teksta treba označiti znakom „#“. Radi kompatibilnosti sa prethodnim zadacima, naredbu slati u obliku „12 0 0 Ime Prezime#“.

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#define BAUDRATE 9600
#define BAUD_PRESCALER (((F_CPU / (BAUDRATE * 16UL))) - 1)

//Deklaracija funkcija
void USART_init(void);
unsigned char USART_receive(void);
void USART_send( unsigned char data);
void USART_putstring(char* StringPtr);

char String[]="";
unsigned char data;
uint8_t brightness;
uint8_t status;

int main(void)
{
    DDRD |= 0b00111100;
    PORTD=0x00;

    USART_init(); //inicijalizacija serijske komunikacije

    while (1)
    {
        data=USART_receive();
        USART_send(data);

        //znak programu da ispisuje string je zadnji karakter u stringu, te se
        naredbom 'mod' (oznaka: %) može dobiti zadnji znak stringa
        if( data%10=='#'){
            uart_write(data);
        }
    }
}

void pwm_init() //funkcija za promjenu svjetine
{
    // initialize timer0 in PWM mode
    TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM21) | _BV(WGM20);
    TCCR2B = _BV(WGM22) | _BV(CS20);
}
```

//funkcije potrebne za UART komunikaciju

```
void USART_init(void){
    UBRR0H = (uint8_t)(BAUD_PRESCALLER>>8); // Baud Rate Register upper 4 bits
    UBRR0L = (uint8_t)(BAUD_PRESCALLER); // Baud Rate Register lower 4 bits
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    UCSR0C = (3<<UCSZ00); //Set frame
}

unsigned char USART_receive(void){
    while(!(UCSR0A & (1<<RXC0)));
    return UDR0;
}

void USART_send( unsigned char data)
{
    while(!(UCSR0A & (1<<UDRE0)));
    UDR0 = data;
}

void USART_putstring(char* StringPtr){
    while(*StringPtr != 0x00){
        USART_send(*StringPtr);
        StringPtr++;}
}

void uart_putchar(char c)
{
    while (!(UCSR0A & (1 << UDRE0)));
    UDR0 = c;
}

void uart_write(char* c)
{
    for (int i = 0; c[i] != '\0'; i++) {
        uart_putchar(c[i]);
    }
}
```

3.10. Zadatak 10.

Doradi spoj dodavanjem tipkala na pin 7 Arduina. Pritiskom na tipkalo, potrebno je uključivati i isključivati LED na pinu 2. Koristiti odgovarajuću biblioteku JC_button-master[3]. Pritiskom na tipkalo potrebno je uključiti LED. Ponovnim pritiskom na tipku LED se isključuje . Držanje pritisnute tipke ne mijenja stanje LED.

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL))) - 1)

//Deklaracija funkcija
void USART_init(void);
unsigned char USART_receive(void);
void USART_send( unsigned char data);
void USART_putstring(char* StringPtr);
int led_state=0;
int main(void){
    DDRD |= 0b00000100; //pinMode(2, OUTPUT);
    DDRD &= 0b01111111; //pinMode(7, INPUT_PULLUP), Postavljanje registra DDR u 0 za
pin 7;
    PORTD |= 0b11000000; // Postavljanje registra PORTD u 1 za pin 7
    USART_init();
    while(1){
        if ((PIND & 0b10000000) >> 7 == 0){
            PORTD |= 0b00000100;
            led_state=1; //ako je varijabla led_state u 1, znači da je LED dioda
upaljena
        }
        if(led_state==1){
            if ((PIND & 0b10000000) >> 7 == 0)
            {
                PORTD &= 0b01111111;
                led_state=0; //ako je varijabla led_state u 0, znači da je LED
dioda ugašena
            }
        }
    }
    return 0;
}

void USART_init(void){
    UBRR0H = (uint8_t)(BAUD_PRESCALLER>>8); // Baud Rate Register upper 4 bits
    UBRR0L = (uint8_t)(BAUD_PRESCALLER); // Baud Rate Register lower 4 bits
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    UCSR0C = (3<<UCSZ00); //Set frame
}

unsigned char USART_receive(void){
```

```
        while(!(UCSR0A & (1<<RXC0)));
        return UDR0;
    }

    void USART_send( unsigned char data)
    {
        while(!(UCSR0A & (1<<UDRE0)));
        UDR0 = data;
    }

    void USART_putstr(char* StringPtr){
        while(*StringPtr != 0x00){
            USART_send(*StringPtr);
            StringPtr++;}
    }
```

4. Literatura

1. UART serijska komunikacija – PowerPoint prezentacija, Dumančić, Z., 01.Serijska komunikacija UART, 18.09.2019. (pregledano 08.05.2020.)
2. Korisnički program PuTTY, <https://putty.org/> (pregledano 08.05.2020.)
3. JC_button-master biblioteka, https://github.com/JChristensen/JC_Button (pregledano 24.04.2020.)
4. Pomoć pri pisanju programa: Web stranica i forum *AVR Freaks*, <https://www.avrfreaks.net/>
5. Program za pisanje kodova Atmel Studio 7, <https://www.microchip.com/mplab/avr-support/atmel-studio-7>
6. PWM i svjetlina LED diode, <https://sites.google.com/site/qeewiki/books/avr-guide/pwm-on-the-atmega328>