

Prototip funkcije	Opis funkcije
<code>void InitIO(void);</code>	Inicijalizacija ulaza i izlaza
<code>void InitOut(unsigned char ch);</code>	Inicijalizacija izlaza „ch“
<code>void SetOutHigh(unsigned char ch);</code>	Postavljanje izlaza „ch“ u logički 1
<code>void SetOutLow(unsigned char ch);</code>	Postavljanje izlaza „ch“ u logički 0
<code>void SetOutToggle(unsigned char ch);</code>	Promjena stanja izlaza „ch“
<code>unsigned char Relay1State(void);</code>	Čitanje stanja releja 1
<code>unsigned char Relay2State(void);</code>	Čitanje stanja releja 2
<code>unsigned char Relay3State(void);</code>	Čitanje stanja releja 3
<code>unsigned char Relay4State(void);</code>	Čitanje stanja releja 4
<code>void Relay1On(void);</code>	Uključivanje releja 1
<code>void Relay2On(void);</code>	Uključivanje releja 2
<code>void Relay3On(void);</code>	Uključivanje releja 3
<code>void Relay4On(void);</code>	Uključivanje releja 4
<code>void Relay1Off(void);</code>	Isključivanje releja 1
<code>void Relay2Off(void);</code>	Isključivanje releja 2
<code>void Relay3Off(void);</code>	Isključivanje releja 3
<code>void Relay4Off(void);</code>	Isključivanje releja 4
<code>void Relay1Toggle(void);</code>	Promjena stanja releja 1
<code>void Relay2Toggle(void);</code>	Promjena stanja releja 2
<code>void Relay3Toggle(void);</code>	Promjena stanja releja 3
<code>void Relay4Toggle(void);</code>	Promjena stanja releja 4
<code>unsigned char Input1(void);</code>	Čitanje ulaza 1 i tipkala 1
<code>unsigned char Input2(void);</code>	Čitanje ulaza 2 i tipkala 2
<code>unsigned char Input3(void);</code>	Čitanje ulaza 3 i tipkala 3
<code>unsigned char Input4(void);</code>	Čitanje ulaza 4 i tipkala 4
<code>unsigned char InputIR1(void);</code>	Čitanje ulaza IR1
<code>unsigned char InputIR2(void);</code>	Čitanje ulaza IR2

<code>unsigned char InputDS1(void);</code>	Čitanje ulaza DS1
<code>unsigned char InputSERVO_1(void);</code>	Čitanje ulaza SERVO_1
<code>unsigned char InputSERVO_2(void);</code>	Čitanje ulaza SERVO_2
<code>unsigned char InputPWM_1(void);</code>	Čitanje ulaza PWM_1
<code>unsigned char InputPWM_2(void);</code>	Čitanje ulaza PWM_2
<code>int InitServo(unsigned char ch);</code>	Inicijalizacija servo motora
<code>int StopServo(unsigned char ch);</code>	Zaustavljanje servo motora
<code>int GetServoState(unsigned char ch);</code>	Čitanje stanja servo motora
<code>int SetServo(unsigned char ch, int position);</code>	Postavljanje pozicije servo motora
<code>int GetServo(unsigned char ch);</code>	Čitanje pozicije servo motora
<code>int ConfigServoSpeed(unsigned char ch, unsigned char dir,                                 unsigned char speed, int position);</code>	Konfiguracija brzina servo motora
<code>int SetServoSpeed(unsigned char ch, unsigned char dir,                                 unsigned char speed);</code>	Postavljanje servo motora na konfiguriranu brzinu
<code>void InitDcMotor(void);</code>	Inicijalizacija DC motora
<code>void StopDcMotor(void);</code>	Zaustavljanje DC motora
<code>unsigned char GetDcMotorState(void);</code>	Čitanje stanja DC motora
<code>int SetRightDcMotor(unsigned char direction,                                 unsigned char speed);</code>	Postavljanje brzine i smjera desnog DC motora
<code>int SetLeftDcMotor(unsigned char direction,                                 unsigned char speed);</code>	Postavljanje brzine i smjera lijevog DC motora
<code>int GetRightDcMotor(unsigned char* direction,                                 unsigned char* speed);</code>	Čitanje brzine i smjera desnog DC motora
<code>int GetLeftDcMotor(unsigned char* direction,                                 unsigned char* speed);</code>	Čitanje brzine i smjera lijevog DC motora
<code>int SetDcMotorAcceleration(int acc);</code>	Postavljanje akceleracije DC motora
<code>int GetDcMotorAcceleration(void);</code>	Čitanje akceleracije DC motora
<code>void InitTimer(void);</code>	Inicijalizacija hardverskog brojača
<code>int InitDs1820(unsigned char ch);</code>	Inicijalizacija temperaturnog senzora DS1820.
<code>int GetDs1820(unsigned char ch);</code>	Čitanje temperaturnog senzora DS1820

<code>int InitAdc(unsigned char ch);</code>	Inicijalizacija AD pretvarača
<code>int StopAdc(unsigned char ch);</code>	Zaustavljanje AD pretvarača
<code>int GetAdcState(unsigned char ch);</code>	Čitanje stanja AD pretvarača
<code>int GetAdc(unsigned char ch);</code>	Čitanje AD pretvarača
<code>int InitUart(unsigned long baudrate);</code>	Inicijalizacija RS232 komunikacije
<code>int UartPutChar(unsigned char c);</code>	Slanje jednog byte-a putem RS232 ili bluetooth komunikacije
<code>unsigned char UartGetChar(void);</code>	Čitanje jednog byte-a iz dolaznog RS232 ili bluetooth međuspremnik
<code>unsigned char UartTestHit(void);</code>	Čitanje stanja dolaznog RS232 ili bluetooth međuspremnik
<code>void UartFlushRx(void);</code>	Brisanje dolaznog RS232 ili bluetooth međuspremnik
<code>void UartFlushTx(void);</code>	Brisanje odlaznog međuspremnik
<code>void InitBluetooth(void);</code>	Inicijalizacija bluetooth komunikacije
<code>int BluetoothChangeName(char* new_name);</code>	Mjenjanje imena bluetooth modula
<code>int BluetoothChangePasskey(char* new_passkey);</code>	Mjenjanje lozinke bluetooth modula
<code>int BluetoothChangeNameAndPasskey( char* new_name, char* new_passkey);</code>	Mjenjanje lozinke i imena bluetooth modula
<code>char* BluetoothGetName(void);</code>	Čitanje imena bluetooth modula
<code>char* BluetoothGetPasskey(void);</code>	Čitanje lozinke bluetooth modula
<code>unsigned char BluetoothConnection(void);</code>	Čitanje stanja bluetooth veze
<code>unsigned char BluetoothT1(void);</code>	Čitanje T1 tipke bluetooth modula
<code>void BluetoothLedOn(void);</code>	Uključivanje STATUS LED-a bluetooth modula
<code>void BluetoothLedOff(void);</code>	Isključivanje STATUS LED-a bluetooth modula
<code>void BluetoothLedToggle(void);</code>	Promjena stanja STATUS LED-a bluetooth modula
<code>void InitBuzzer(void);</code>	Inicijalizacija buzzer-a na portu DS1
<code>void PlayNote(unsigned char note, unsigned char duration);</code>	Sviranje note buzzer-om

Funkcija inicijalizira ulaze i izlaze EMoR kontrolera i poziva se prije drugih inicijalizacija u programu. Nakon izvršavanja funkcije releji su postavljeni kao izlazi, a SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2 i DS1 kao TTL ulazi. Inicijalizacija servo motora, analognog-digitalnog konvertera, DC motora, temperaturnog senzora može promijeniti postavke ovih ulaza, te je zbog toga nužno prvo izvršiti **InitIO()**; a nakon toga ostale inicijalizacije u programu.

```
void InitIO(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    while(1){                        /* beskonačna petlja */
    }
}
```

Funkcija inicijalizira izlaze EMoRo kontrolera na portovima SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2 i DS1. Prije inicijalizacije izlaza potrebno je izvršiti inicijalizaciju ulaza i izlaza **InitIO()**; funkcijom. Prilikom inicijalizacije izlaza na pojedinim izlazima potrebno je obratiti pozornost da se ostale inicijalizacije vezane za pojedini port ne mogu koristiti.

(Prilikom poziva **InitOut(DS\_1)**; port „DS1“ biti će inicijaliziran kao izlaz, te se njime može upravljati **SetOutHigh(DS1)**;, **SetOutLow(DS1)**;, **SetOutToggle(DS1)**; funkcijama, ali DS1 port ne možemo koristiti za upravljanje servo motorom, čitanje analogne vrijednosti ili bilo kojom drugom funkcionalnosti vezanom za DS1 port)

```
void InitOut(unsigned char ch);
```

**Parametri:**

**unsigned char** ch – izlaz [IR1, IR2, DS1, SERVO\_1, SERVO\_2, PWM\_1, PWM\_2]

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitOut(SERVO_1);               /* inicijaliziraj izlaz na portu SERVO_1 */
    InitOut(SERVO_2);               /* inicijaliziraj izlaz na portu SERVO_2 */
    InitOut(PWM_1);                 /* inicijaliziraj izlaz na portu PWM_1 */
    InitOut(PWM_2);                 /* inicijaliziraj izlaz na portu PWM_2 */
    InitOut(IR1);                   /* inicijaliziraj izlaz na portu IR1 */
    InitOut(IR2);                   /* inicijaliziraj izlaz na portu IR2 */
    InitOut(DS1);                   /* inicijaliziraj izlaz na portu DS1 */
    while(1){                        /* beskonačna petlja */
    }
}
```

Funkcija postavlja logički „1“ na izlaz „ch“. Prije postavljanja izlaza na logički „1“ potrebno je izvršiti inicijalizaciju ulaza i izlaza **InitIO()**; funkcijom, te inicijalizaciju izlaza **InitOut()**; funkcijom.

```
void SetOutHigh(unsigned char ch);
```

**Parametri:**

`unsigned char ch` – izlaz [IR1, IR2, DS1, SERVO\_1, SERVO\_2, PWM\_1, PWM\_2]

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitOut(SERVO_1);                /* inicijaliziraj izlaz na portu SERVO_1 */
    InitOut(SERVO_2);                /* inicijaliziraj izlaz na portu SERVO_2 */
    InitOut(PWM_1);                  /* inicijaliziraj izlaz na portu PWM_1 */
    InitOut(PWM_2);                  /* inicijaliziraj izlaz na portu PWM_2 */
    InitOut(IR1);                    /* inicijaliziraj izlaz na portu IR1 */
    InitOut(IR2);                    /* inicijaliziraj izlaz na portu IR2 */
    InitOut(DS1);                    /* inicijaliziraj izlaz na portu DS1 */

    while(1){                        /* beskonačna petlja */
        SetOutHigh(SERVO_1);         /* postavi u logički 1 SERVO_1 izlaz */
        SetOutHigh(SERVO_2);         /* postavi u logički 1 SERVO_2 izlaz */
        SetOutHigh(PWM_1);           /* postavi u logički 1 PWM_1 izlaz */
        SetOutHigh(PWM_2);           /* postavi u logički 1 PWM_2 izlaz */
        SetOutHigh(IR1);             /* postavi u logički 1 IR1 izlaz */
        SetOutHigh(IR2);             /* postavi u logički 1 IR2 izlaz */
        SetOutHigh(DS1);             /* postavi u logički 1 DS1 izlaz */
    }
}
```

Funkcija postavlja logički „0“ na izlaz „ch“. Prije postavljanja izlaza na logički „0“ potrebno je izvršiti inicijalizaciju ulaza i izlaza **InitIO()**; funkcijom, te inicijalizaciju izlaza **InitOut()**; funkcijom.

```
void SetOutLow(unsigned char ch);
```

**Parametri:**

`unsigned char ch` – izlaz [IR1, IR2, DS1, SERVO\_1, SERVO\_2, PWM\_1, PWM\_2]

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitOut(SERVO_1);                /* inicijaliziraj izlaz na portu SERVO_1 */
    InitOut(SERVO_2);                /* inicijaliziraj izlaz na portu SERVO_2 */
    InitOut(PWM_1);                  /* inicijaliziraj izlaz na portu PWM_1 */
}
```

```

InitOut(PWM_2);          /* inicijaliziraj izlaz na portu PWM_2          */
InitOut(IR1);            /* inicijaliziraj izlaz na portu IR1          */
InitOut(IR2);            /* inicijaliziraj izlaz na portu IR2          */
InitOut(DS1);            /* inicijaliziraj izlaz na portu DS1          */

while(1){                /* beskonačna petlja          */
    SetOutLow(SERVO_1);   /* postavi u logički 0 SERVO_1 izlaz        */
    SetOutLow(SERVO_2);   /* postavi u logički 0 SERVO_2 izlaz        */
    SetOutLow(PWM_1);     /* postavi u logički 0 PWM_1 izlaz         */
    SetOutLow(PWM_2);     /* postavi u logički 0 PWM_2 izlaz         */
    SetOutLow(IR1);       /* postavi u logički 0 IR1 izlaz           */
    SetOutLow(IR2);       /* postavi u logički 0 IR2 izlaz           */
    SetOutLow(DS1);       /* postavi u logički 0 DS1 izlaz           */
}
}

```

Funkcija mijenja stanje izlaza „ch“. Prije promjene trenutnog stanja „ch“ izlaza potrebno je izvršiti inicijalizaciju ulaza i izlaza **InitIO()**; funkcijom, te inicijalizaciju izlaza **InitOut()**; funkcijom.

```
void SetOutToggle(unsigned char ch);
```

#### Parametri:

**unsigned char** ch - izlaz [IR1, IR2, DS1, SERVO\_1, SERVO\_2, PWM\_1, PWM\_2]

#### Povratna vrijednost:

Nema

#### Podržani moduli:

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

#### Primjer:

```

#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke          */

int main(void){           /* glavni program          */
    InitIO();              /* inicijaliziraj ulaze i izlaze          */
    InitOut(SERVO_1);      /* inicijaliziraj izlaz na portu SERVO_1  */
    InitOut(SERVO_2);      /* inicijaliziraj izlaz na portu SERVO_2  */
    InitOut(PWM_1);        /* inicijaliziraj izlaz na portu PWM_1    */
    InitOut(PWM_2);        /* inicijaliziraj izlaz na portu PWM_2    */
    InitOut(IR1);          /* inicijaliziraj izlaz na portu IR1      */
    InitOut(IR2);          /* inicijaliziraj izlaz na portu IR2      */
    InitOut(DS1);          /* inicijaliziraj izlaz na portu DS1      */

    while(1){              /* beskonačna petlja          */
        SetOutToggle(SERVO_1); /* promijeni stanje SERVO_1 izlaza        */
        SetOutToggle(SERVO_2); /* promijeni stanje SERVO_2 izlaza        */
        SetOutToggle(PWM_1);  /* promijeni stanje PWM_1 izlaza         */
        SetOutToggle(PWM_2);  /* promijeni stanje PWM_2 izlaza         */
        SetOutToggle(IR1);    /* promijeni stanje IR1 izlaza           */
        SetOutToggle(IR2);    /* promijeni stanje IR2 izlaza           */
        SetOutToggle(DS1);    /* promijeni stanje DS1 izlaza           */
    }
}

```

Funkcije čitaju stanje relejnih izlaza. Za čitanje relejnih izlaza potrebno je inicijalizirati ulaze i izlaze kontrolera **InitIO()**; funkcijom.

```

unsigned char Relay1State(void);
unsigned char Relay2State(void);
unsigned char Relay3State(void);

```

```
unsigned char Relay4State(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

unsigned char – Stanje relejnog izlaza. [0 - relej isključen, 1 - relej uključen]

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    while(1){ /* beskonačna petlja */
        if(Relay1State() == 1) Relay10ff(); /* relej 1 uključen? Isključi relej */
        if(Relay2State() == 1) Relay20ff(); /* relej 2 uključen? Isključi relej */
        if(Relay3State() == 1) Relay30ff(); /* relej 3 uključen? Isključi relej */
        if(Relay4State() == 1) Relay40ff(); /* relej 4 uključen? Isključi relej */
    }
}
```

Funkcije uključuju relejne izlaze. Za uključivanje relejnih izlaza potrebno je inicijalizirati ulaze i izlaze kontrolera **InitIO()** funkcijom.

```
void Relay1On(void);
void Relay2On(void);
void Relay3On(void);
void Relay4On(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    while(1){ /* beskonačna petlja */
        Relay1On(); /* uključi relej 1 */
        Relay2On(); /* uključi relej 2 */
        Relay3On(); /* uključi relej 3 */
        Relay4On(); /* uključi relej 4 */
    }
}
```

Funkcije isključuju relejne izlaze. Za isključivanje relejnih izlaza potrebno je inicijalizirati ulaze i izlaze kontrolera **InitIO()** funkcijom.

```
void Relay10ff(void);
void Relay20ff(void);
void Relay30ff(void);
```

```
void Relay4Off(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>      /* dodjeljivanje emoro biblioteke */

int main(void){                  /* glavni program */
    InitIO();                    /* inicijaliziraj ulaze i izlaze */
    while(1){                    /* beskonačna petlja */
        Relay1Off();             /* isključi relej 1 */
        Relay2Off();             /* isključi relej 2 */
        Relay3Off();             /* isključi relej 3 */
        Relay4Off();             /* isključi relej 4 */
    }
}
```

Funkcije mijenjaju stanje relejnih izlaza. Za mijenjanje stanja relejnih izlaza potrebno je inicijalizirati ulaze i izlaze kontrolera **InitIO()**; funkcijom. Pozivom funkcije uključeni relejni izlaz će se isključiti, a isključeni relejni izlaz uključiti.

```
void Relay1Toggle(void);
void Relay2Toggle(void);
void Relay3Toggle(void);
void Relay4Toggle(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>      /* dodjeljivanje emoro biblioteke */

int main(void){                  /* glavni program */
    InitIO();                    /* inicijaliziraj ulaze i izlaze */
    while(1){                    /* beskonačna petlja */
        Relay1Toggle();          /* promijeni stanje releja 1 */
        Relay2Toggle();          /* promijeni stanje releja 2 */
        Relay3Toggle();          /* promijeni stanje releja 3 */
        Relay4Toggle();          /* promijeni stanje releja 4 */
    }
}
```

Funkcije čitaju ulaze (1, 2, 3, 4) i tipke (1, 2, 3, 4). Za čitanje ulaza potrebno je inicijalizirati ulaze i izlaze kontrolera **InitIO()**; funkcijom. LED svakog ulaza prikazuje trenutno stanje ulaza. Ako LED svijetli ulaz je aktivan (0V), te funkcija vraća vrijednost „1“. Ako LED ne svijetli ulaz je neaktivan (5V), te funkcija vraća vrijednost „0“.

```
unsigned char Input1(void);
unsigned char Input2(void);
```



```
unsigned char Input3(void);
unsigned char Input4(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

**unsigned char** - Stanje ulaza (ulazi 1 - 4)  
(0) - puštena tipka ili neaktivan ulaz. (Input High - 5V)  
(1) - pritisnuta tipka ili aktiviran ulaz. (Input Low - 0V)

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    while(1){ /* beskonačna petlja */
        if(Input1() == 1) Relay1On(); /* ulaz 1 aktivan (0V), uključi relej 1 */
        if(Input2() == 1) Relay2On(); /* ulaz 2 aktivan (0V), uključi relej 2 */
        if(Input3() == 1) Relay3On(); /* ulaz 3 aktivan (0V), uključi relej 3 */
        if(Input4() == 1) Relay4On(); /* ulaz 4 aktivan (0V), uključi relej 4 */
    }
}
```

Funkcije čitaju ulaze IR1, IR2, DS1, SERVO\_1, SERVO\_2, PWM\_1 i PWM\_2. Za čitanje ulaza potrebno je inicijalizirati ulaze i izlaze kontrolera **InitIO()**; funkcijom. LED ulaza IR1 i IR2 prikazuje trenutno stanje ulaza. Ako LED svijetli ulaz je aktivan (0V), te funkcija vraća vrijednost „1“. Ako LED ne svijetli ulaz je neaktivan (5V), te funkcija vraća vrijednost „0“.

Navedene ulaze moguće je čitati samo u slučaju da na njima nije inicijalizirana druga funkcionalnost. Ulaze mogu koristiti upravljački programi servo motora, analogno-digitalnog konvertera, DC motora i u slučaju da je ovim portovima inicijalizirana druga funkcionalnost, ne mogu se čitati kao TTL ulazi.

(npr. Izvršavanjem **InitServo(SERVO\_1)**; **InitAdc(DS1)**; SERVO\_1 i DS1 će biti konfigurirani za funkcionalnost servo motora i analognog ulaza, te se funkcije **InputSERVO\_1()**; i **InputDS1()**; ne mogu koristiti).

```
unsigned char InputIR1(void);
unsigned char InputIR2(void);
unsigned char InputDS1(void);
unsigned char InputSERVO_1(void);
unsigned char InputSERVO_2(void);
unsigned char InputPWM_1(void);
unsigned char InputPWM_2(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

**unsigned char** - Stanje ulaza [IR1, IR2, DS1, SERVO\_1, SERVO\_2, PWM\_1, PWM\_2]  
(0) - ulaz neaktivan. (Input High - 5V)  
(1) - ulaz aktiviran. (Input Low - 0V)

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
}
```

```

while(1){
    if(InputIR1()==1)    Relay1On();    /* ulaz IR1 aktivan, uključi relej 1 */
    if(InputIR2()==1)    Relay2On();    /* ulaz IR2 aktivan, uključi relej 2 */
    if(InputDS1()==1)    Relay3On();    /* ulaz DS1 aktivan, uključi relej 3 */
    if(InputSERVO_1()==1) Relay4On();    /* ulaz SERVO_1 aktivan, uključi relej 4 */
    if(InputSERVO_2()==1) Relay1Off();    /* ulaz SERVO_2 aktivan, isključi relej 1 */
    if(InputPWM_1()==1)   Relay2Off();    /* ulaz PWM_1 aktivan, isključi relej 2 */
    if(InputPWM_2()==1)   Relay3Off();    /* ulaz PWM_2 aktivan, isključi relej 3 */
}
}

```

Funkcija inicijalizira izlaz servo motora na kojemu upravljački program generira servo pulse . Kontroler EMoRo podržava 7 servo motora na izlazima SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2 i DS1, a za upravljanje servo motorima potrebno je izvršiti inicijalizaciju servo izlaza. Nakon inicijalizacije servo motora zadano stanje izlaza postavljeno je na vrijednost 1500 što odgovara impulsu 1.5 mS. Treba obratiti pozornost da se nakon inicijalizacije servo motora na ovim portovima ne može koristiti inicijalizacija za drugu funkcionalnost, kao niti funkcije koje čitaju port kao ulaz.

(npr. Pozivom **InitServo(IR1)**; IR1 port konfiguriran je kao servo izlaz, te se ne može koristiti funkcija čitanja ulaza **InputIR1()** , kao niti inicijalizacija drugih funkcionalnosti na IR1 portu (ADC, DS1820...)

DC motor driver koristi zajedničke portove SERVO\_2, PWM\_1, PWM\_2, i u slučaju da se jedan od navedenih izlaza konfigurira kao servo motor tada se DC motori ne mogu koristiti.

Inicijalizirani upravljački program servo motora generira servo pulse u intervalu [0,5 mS – 2,5 mS] svakih 20mS.

```
int InitServo(unsigned char ch);
```

#### Parametri:

**unsigned char** ch – servo motor [SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2, DS1]

#### Povratna vrijednost:

**int** - Rezultat funkcije:

(0) – Servo izlaz uspješno inicijaliziran

(-1) – Greška: Parametar servo motora „ch“ nije u intervalu [1-7]

#### Podržani moduli:

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

#### Primjer:

```

#include <inovatic\emoro.h>    /* dodjeljivanje emoro biblioteke */

int main(void){
    InitIO();                  /* glavni program */
    InitServo(SERVO_1);        /* inicijaliziraj ulaze i izlaze */
    InitServo(SERVO_2);        /* inicijaliziraj servo motor na portu SERVO_1 */
    InitServo(PWM_1);          /* inicijaliziraj servo motor na portu SERVO_2 */
    InitServo(PWM_2);          /* inicijaliziraj servo motor na portu PWM_1 */
    InitServo(IR1);            /* inicijaliziraj servo motor na portu PWM_2 */
    InitServo(IR2);            /* inicijaliziraj servo motor na portu IR1 */
    InitServo(DS1);            /* inicijaliziraj servo motor na portu IR2 */
    while(1){                  /* inicijaliziraj servo motor na portu DS1 */
        /* beskonačna petlja */
    }
}

```

Funkcija isključuje izlaz servo motora i konfigurira port kao ulaz.

```
int StopServo(unsigned char ch);
```

#### Parametri:

**unsigned char** ch – servo motor [SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2, DS1]

**Povratna vrijednost:**

**int** - Rezultat funkcije:  
 (0) - Servo izlaz uspješno zaustavljen  
 (-1) - Greška: Parametar servo motora „ch“ nije u intervalu [1-7]

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitServo(SERVO_1);              /* inicijaliziraj servo motor na portu SERVO_1 */
    SetServo(SERVO_1, 1000);         /* postavi SERVO_1 na poziciju 1000 (1mS) */
    while(1){                        /* beskonačna petlja */
        if(Input1() == 1){           /* pritisnuta tipka 1? */
            StopServo(SERVO_1);      /* zaustavi servo motor SERVO_1 */
        }
    }
}
```

Funkcija čita stanje servo izlaza (Servo izlaz pokrenut, servo izlaz zaustavljen).

```
int GetServoState(unsigned char ch);
```

**Parametri:**

**unsigned char** ch - servo motor [SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2, DS1]

**Povratna vrijednost:**

**int** - Rezultat funkcije:  
 (1) - Servo izlaz je inicijaliziran (Servo pokrenut)  
 (0) - Servo izlaz nije inicijaliziran (Servo zaustavljen)  
 (-1) - Greška: Parametar servo motora „ch“ nije u intervalu [1-7]

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitServo(SERVO_1);              /* inicijaliziraj servo motor SERVO_1 */
    while(1){                        /* beskonačna petlja */

        /* pritisnuta tipka ulaza 1, a SERVO_1 je inicijaliziran? */
        if((Input1()==1) && (GetServoState(SERVO_1)==1))
            StopServo(SERVO_1);      /* zaustavi servo na portu SERVO_1 */

        /* puštena tipka ulaza 1, a SERVO_1 nije inicijaliziran? */
        if((Input1()==0) && (GetServoState(SERVO_1)==0))
            InitServo(SERVO_1);      /* inicijaliziraj servo motor SERVO_1 */
    }
}
```

Funkcija postavlja novu poziciju servo motora.

```
int SetServo(unsigned char ch, int position);
```

**Parametri:**

`unsigned char ch` - servo motor [SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2, DS1]  
`int position` - nova pozicija servo motora [500 - 2500] uS

**Povratna vrijednost:**

`int` - Rezultat funkcije:  
(0) - Servo izlaz uspješno postavljen na novu vrijednost  
(-1) - Greška: Pozicija nije u intervalu [500-2500]  
(-2) - Greška: Parametar servo motora „ch“ nije u intervalu [1-7]  
(-3) - Greška: Servo motor nije inicijaliziran

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitServo(SERVO_1);              /* inicijaliziraj servo motor SERVO_1 */
    while(1){                        /* beskonačna petlja */
        SetServo(SERVO_1, 1000);     /* postavi motor SERVO_1 pulse 1000 uS (1mS) */
    }
}
```

Funkcija vraća poziciju servo motora.

```
int GetServo(unsigned char ch);
```

**Parametri:**

`unsigned char ch` - servo motor [SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2, DS1]

**Povratna vrijednost:**

`int` - Rezultat funkcije:  
(500-2500) - Pozicija servo motora  
(-1) - Greška: Parametar servo motora „ch“ nije u intervalu [1-7]  
(-2) - Greška: Servo motor nije inicijaliziran

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitServo(SERVO_1);              /* inicijaliziraj servo motor SERVO_1 */
    InitServo(SERVO_2);              /* inicijaliziraj servo motor SERVO_2 */
    while(1){                        /* beskonačna petlja */

        /* postavi SERVO_2 na poziciju 2000 (2mS pulse) */
        SetServo(SERVO_2, 2000);

        /* postavi SERVO_1 na poziciju na kojoj se nalazi SERVO_2 */
        SetServo(SERVO_1, GetServo(SERVO_2));
    }
}
```

Funkcija konfigurira 5 servo brzina za smjer „FORWARD“ i 5 servo brzina za smjer „REVERSE“, te ih pohranjuje u eeprom memoriju kako bi brzine servo motora ostale pohranjene i nakon isključivanja napona napajanja.

Funkcija **ConfigServoSpeed()**; predviđena je za pohranjivanje brzina servo motora sa konstantom rotacijom, ali se može koristiti i za pohranjivanje vrijednosti pulsa pozicijskih servo motora.

Servo motori postavljaju se na konfigurirane brzine funkcijom **SetServoSpeed()**;

```
int ConfigServoSpeed(unsigned char ch, unsigned char dir, unsigned char speed,
                    int position);
```

#### Parametri:

unsigned char ch	- servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]
unsigned char dir	- smjer servo motora [FORWARD, REVERSE]
unsigned char speed	- konfiguracijska brzina [1-5]
int position	- brzina servo motora [500 - 2500] uS

#### Povratna vrijednost:

int	- Rezultat funkcije:
(0)	- Servo brzina uspješno konfigurirana
(-1)	- Greška: Parametar servo motora „ch“ nije u intervalu [1-7]
(-2)	- Greška: Brzina „speed“ nije u intervalu [1-5]
(-3)	- Greška: Smjer nije u intervalu [FORWARD, REVERSE]
(-4)	- Greška: Pozicija nije u intervalu [500-2500]

#### Podržani moduli:

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

#### Primjer:

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitServo(SERVO_1);              /* inicijaliziraj servo motor SERVO_1 */
    InitServo(SERVO_2);              /* inicijaliziraj servo motor SERVO_2 */
    while(1){                        /* beskonačna petlja */
        /* konfiguriraj 5 brzina FORWARD za motor SERVO_1 */
        ConfigServoSpeed(SERVO_1, FORWARD, 1, 1450);
        ConfigServoSpeed(SERVO_1, FORWARD, 2, 1400);
        ConfigServoSpeed(SERVO_1, FORWARD, 3, 1350);
        ConfigServoSpeed(SERVO_1, FORWARD, 4, 1300);
        ConfigServoSpeed(SERVO_1, FORWARD, 5, 1000);

        /* konfiguriraj 5 brzina FORWARD za motor SERVO_2 */
        ConfigServoSpeed(SERVO_2, FORWARD, 1, 1550);
        ConfigServoSpeed(SERVO_2, FORWARD, 2, 1600);
        ConfigServoSpeed(SERVO_2, FORWARD, 3, 1650);
        ConfigServoSpeed(SERVO_2, FORWARD, 4, 1700);
        ConfigServoSpeed(SERVO_2, FORWARD, 5, 2000);
    }
}
```

Funkcija postavlja servo motor na konfigurirane brzine. Za uspješno postavljanje brzina servo motora potrebno je konfigurirati brzine **ConfigServoSpeed()**; funkcijom. Konfiguraciju brzina potrebno je izvršiti samo jednom jer se brzine pohranjuju u eeprom memoriju i ostaju pohranjene i nakon isključivanja kontrolera sa napona napajanja. Prije postavljanja servo motora na konfiguriranu brzinu potrebno je inicijalizirati servo izlaz **InitServo()**; funkcijom.

```
int SetServoSpeed(unsigned char ch, unsigned char dir, unsigned char speed);
```

#### Parametri:

unsigned char ch	- servo motor [SERVO_1, SERVO_2, PWM_1, PWM_2, IR1, IR2, DS1]
unsigned char dir	- smjer servo motora [FORWARD, REVERSE, STOP]
unsigned char speed	- brzina servo motora [1-5]

**Povratna vrijednost:**

**int** - Rezultat funkcije:

- (0) - Servo brzina uspješno postavljena
- (-1) - Greška: Servo brzina „speed“ nije konfigurirana.
- (-2) - Greška: Parametar servo motora „ch“ nije u intervalu [1-7]
- (-3) - Greška: Servo motor nije inicijaliziran
- (-4) - Greška: Servo brzina „speed“ nije u intervalu [1-5]
- (-5) - Greška: Smjer motora „dir“ nije u intervalu [FORWARD, REVERSE, STOP]

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>    /* dodjeljivanje emoro biblioteke */

int main(void){                /* glavni program */
    InitIO();                   /* inicijaliziraj ulaze i izlaze */
    InitServo(SERVO_1);         /* inicijaliziraj servo motor SERVO_1 */
    InitServo(SERVO_2);         /* inicijaliziraj servo motor SERVO_2 */
    while(1){                   /* beskonačna petlja */
        /* postavi SERVO_1 brzinom 1 FORWARD */
        SetServoSpeed(SERVO_1, FORWARD, 1);

        /* postavi SERVO_2 brzinom 1 FORWARD */
        SetServoSpeed(SERVO_2, FORWARD, 1);
    }
}
```

Funkcija inicijalizira izlaze DC motora Motor-R i Motor-L. Pozivom funkcije **InitDcMotor()**; SERVO\_2, PWM\_1 i PWM\_2 port konfigurirani su kao izlaz, te ih koristi upravljački program DC motora. Potrebno je obratiti pozornost da se navedeni portovi ne mogu konfigurirati za drugu funkcionalnost, niti čitati kao ulazi.

(npr. Inicijalizacijom DC motora SERVO\_2, PWM\_1 i PWM\_2 portovi ne mogu se koristiti kao ulazi ili servo motor, ali se uz inicijalizaciju DC motora mogu inicijalizirati servo motori na portovima SERVO\_1, IR1, IR2 i DS1).

Za korištenje DC motora potrebno je odspojiti hardver sa portova SERVO\_2, PWM\_1 i PWM\_2. Nakon inicijalizacije DC motori su postavljeni na brzinu „0“.

```
void InitDcMotor(void);
```

**Parametri:**

nema

**Povratna vrijednost:**

nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>    /* dodjeljivanje emoro biblioteke */

int main(void){                /* glavni program */
    InitIO();                   /* inicijaliziraj ulaze i izlaze */
    InitDcMotor();              /* inicijaliziraj DC motore Motor-L i Motor-R */
    while(1){                   /* beskonačna petlja */
    }
}
```

Funkcija zaustavlja upravljački program DC motora.

```
void StopDcMotor(void);
```

**Parametri:**

nema

**Povratna vrijednost:**

nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitDcMotor();                   /* inicijaliziraj DC motore Motor-L i Motor-R */
    SetLeftDcMotor(FORWARD, 170);    /* postavi lijevi motor MAX brzinom naprijed */
    SetRightDcMotor(FORWARD, 170);   /* postavi desni motor MAX brzinom naprijed */

    while(1){                        /* beskonačna petlja */
        if(Input1() == 1)            /* pritisnuta tipka 1? */
            StopDcMotor();           /* zaustavi upravljački program DC motora */
    }
}
```

Funkcija čita stanje upravljačkog programa DC motora. (DC motor driver pokrenut, DC motor driver zaustavljen).

```
unsigned char GetDcMotorState(void);
```

**Parametri:**

nema

**Povratna vrijednost:**

unsigned char – Stanje DC motor driver-a:

(1) – DC motor driver inicijaliziran (DC motori pokrenuti)

(0) – DC motor driver nije inicijaliziran (DC motori zaustavljeni)

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */

    while(1){                        /* beskonačna petlja */

        /* pritisnuta tipka ulaza 1 i DC motori su inicijalizirani? */
        if((Input1()==1) && (GetDcMotorState() == 1))
            StopDcMotor();           /* zaustavi DC motore */

        /* puštena tipka ulaza 1 i DC motori nisu inicijalizirani? */
        if((Input1()==0) && (GetDcMotorState() == 0))
            InitDcMotor();           /* inicijaliziraj DC motore */
    }
}
```

Funkcija postavlja novu brzinu desnog DC motora. Prije postavljanja brzine DC motora potrebno je inicijalizirati DC motore `InitDcMotor()`; funkcijom.

```
int SetRightDcMotor(unsigned char direction, unsigned char speed);
```

**Parametri:**

`unsigned char direction` - smjer DC motora [FORWARD, REVERSE, STOP]  
`unsigned char speed` - brzina DC motora [0 - 170]

**Povratna vrijednost:**

`int` - Rezultat funkcije:  
(0) - Uspješno postavljena brzina i smjer desnog DC motora  
(-1) - Greška: Brzina motora nije u intervalu [0-170]  
(-2) - Greška: Smjer motora nije u intervalu [FORWARD, REVERSE, STOP]  
(-3) - Greška: DC motori nisu inicijalizirani

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    InitDcMotor(); /* inicijaliziraj DC motore Motor-L i Motor-R */
    while(1){ /* beskonačna petlja */
        SetRightDcMotor(FORWARD, 170); /* postavi desni DC motor MAX brzinom naprijed */
        _delay_ms(1000); /* čekaj 1000 ms */
        SetRightDcMotor(REVERSE, 170); /* postavi desni DC motor MAX brzinom nazad */
        _delay_ms(1000); /* čekaj 1000 ms */
    }
}
```

Funkcija postavlja novu brzinu lijevog DC motora. Prije postavljanja brzine DC motora potrebno je inicijalizirati DC motore `InitDcMotor()`; funkcijom.

```
int SetLeftDcMotor(unsigned char direction, unsigned char speed);
```

**Parametri:**

`unsigned char direction` - smjer DC motora [FORWARD, REVERSE, STOP]  
`unsigned char speed` - brzina DC motora [0 - 170]

**Povratna vrijednost:**

`int` - Rezultat funkcije:  
(0) - Uspješno postavljena brzina i smjer lijevog DC motora  
(-1) - Greška: Brzina motora nije u intervalu [0-170]  
(-2) - Greška: Smjer motora nije u intervalu [FORWARD, REVERSE, STOP]  
(-3) - Greška: DC motori nisu inicijalizirani

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    InitDcMotor(); /* inicijaliziraj DC motore Motor-L i Motor-R */
    while(1){ /* beskonačna petlja */
        SetLeftDcMotor(FORWARD, 170); /* postavi lijevi DC motor MAX brzinom naprijed */
    }
}
```



```

    _delay_ms(1000);          /* čekaj 1000 ms */
    SetLeftDcMotor(REVERSE, 170); /* postavi lijevi DC motor MAX brzinom nazad */
    _delay_ms(1000);          /* čekaj 1000 ms */
}
}

```

Funkcija čita postavljenu brzinu i smjer desnog DC motora. Prije čitanja brzine i smjera DC motora potrebno je inicijalizirati DC motore **InitDcMotor()**; funkcijom.

```
int GetRightDcMotor(unsigned char* direction, unsigned char* speed);
```

**Parametri:**

**unsigned char\*** direction – smjer DC motora [FORWARD, REVERSE, STOP]  
**unsigned char\*** speed – brzina DC motora [0 - 170]

**Povratna vrijednost:**

**int** – Rezultat funkcije:  
 (0) – Uspješno pročitana brzina i smjer desnog DC motora  
 (-1) – Greška: DC motori nisu inicijalizirani

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

unsigned char my_speed, my_direction;

int main(void){                     /* glavni program */
    InitIO();                       /* inicijaliziraj ulaze i izlaze */
    InitDcMotor();                  /* inicijaliziraj DC motore Motor-L i Motor-R */
    SetRightDcMotor(FORWARD, 100);  /* postavi desni DC motor na brzinu 100 naprijed */
    while(1){                       /* beskonačna petlja */

        GetRightDcMotor(&my_direction, &my_speed); /* čitaj smjer i brzinu desnog DC motora */

        SetLeftDcMotor(my_direction, my_speed);    /* postavi lijevi DC motor na brzinu i smjer desnog */
    }
}

```

Funkcija čita postavljenu brzinu i smjer lijevog DC motora. Prije čitanja brzine i smjera DC motora potrebno je inicijalizirati DC motore **InitDcMotor()**; funkcijom.

```
int GetLeftDcMotor(unsigned char* direction, unsigned char* speed);
```

**Parametri:**

**unsigned char\*** direction – smjer DC motora [FORWARD, REVERSE, STOP]  
**unsigned char\*** speed – brzina DC motora [0 - 170]

**Povratna vrijednost:**

**int** – Rezultat funkcije:  
 (0) – Uspješno pročitana brzina i smjer lijevog DC motora  
 (-1) – Greška: DC motori nisu inicijalizirani

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

```

```

unsigned char my_speed, my_direction;

int main(void){
    InitIO();
    InitDcMotor();
    SetLeftDcMotor(FORWARD, 100);
    while(1){

        GetLeftDcMotor(&my_direction, &my_speed);
        SetRightDcMotor(my_direction, my_speed);
    }
}

```

Funkcija postavlja novu akceleraciju DC motora. Prije postavljanja akceleracije potrebno je inicijalizirati DC motore **InitDcMotor()**; funkcijom. Upravljački program DC motora ne dozvoljava naglu promjenu smjera i brzine zbog zaštite DC motora, reduktora i kontrolera, te će ubrzavati motore sukladno podešenoj akceleraciji. Zadana akceleracija (acc) nakon inicijalizacije DC motora je vrijednost 128, a računa se prema formuli:

$$Akceleracija(\mu s) = 32 \times acc$$

Zadana akceleracija DC motora nakon inicijalizacije (acc=128):

$$\begin{aligned} \text{Promjena brzine svakih:} & 32(\mu s) \times 128 = 4096(\mu s) \\ \text{Maksimalnu brzinu DC motori će postići za:} & 4,096(ms) \times 170 = 969,32(ms) \end{aligned}$$

Maksimalna akceleracija DC motora (acc=32):

$$\begin{aligned} \text{Promjena brzine svakih:} & 32(\mu s) \times 32 = 1024(\mu s) \\ \text{Maksimalnu brzinu DC motori će postići za:} & 1,024(ms) \times 170 = 174,08(ms) \end{aligned}$$

Minimalna akceleracija DC motora (acc=1024):

$$\begin{aligned} \text{Promjena brzine svakih:} & 32(\mu s) \times 1024 = 32768(\mu s) \\ \text{Maksimalnu brzinu DC motori će postići za:} & 32,768(ms) \times 170 = 5570,56(ms) \end{aligned}$$

```
int SetDcMotorAcceleration(int acc);
```

#### Parametri:

**int** acc – akceleracija DC motora [32 - 1023]

#### Povratna vrijednost:

**int** – Rezultat funkcije:  
 (0) – Uspješno postavljena nova akceleracija DC motora  
 (-1) – Greška: DC motori nisu inicijalizirani

#### Podržani moduli:

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

#### Primjer:

```

#include <inovatic\emoro.h>
/* dodjeljivanje emoro biblioteke */

int main(void){
    InitIO();
    InitDcMotor();
    SetDcMotorAcceleration(32);
    SetLeftDcMotor(FORWARD, 170);
    while(1){

        /* lijevi motor će dostići maksimalnu brzinu nakon: 1,024 [mS] * 170 [speed] = 174,08 mS */
    }
}

```

```
}
```

Funkcija čita akceleraciju DC motora. Prije čitanja akceleracije potrebno je inicijalizirati DC motore **InitDcMotor()**; funkcijom.

```
int GetDcMotorAcceleration(void);
```

**Parametri:**

nema

**Povratna vrijednost:**

**int** - Rezultat funkcije:  
(32-1023) - Akceleracija DC motora [32-1023]  
(-1) - Greška: DC motori nisu inicijalizirani

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    InitDcMotor(); /* inicijaliziraj DC motore Motor-L i Motor-R */
    if(GetDcMotorAcceleration() != 32){ /* akceleracija nije 32? (max) */
        SetDcMotorAcceleration(32); /* postavi novu akceleraciju: 32uS * 32 = 1024 uS */
    }
    SetLeftDcMotor(FORWARD, 170); /* postavi lijevi DC motor na maksimalnu brzinu */
    while(1){ /* beskonačna petlja */

        /* lijevi motor će dostići maksimalnu brzinu nakon: 1,024 [mS] * 170 [speed] = 174,08 mS */
    }
}
```

Funkcija inicijalizira hardverski brojač koji povećava 10 varijabla svakih 1mS. Varijable brojača su „**unsigned int**“ i mogu sadržavati vrijednosti [0 - 65535]. U slučaju inicijalizacije brojača DC motori se ne mogu koristiti jer njihova inicijalizacija, zbog interne arhitekture mikrokontrolera, mora isključiti hardverski brojač.

```
void InitTimer(void);
```

**Varijable brojača:**

```
extern volatile unsigned int timer_1; extern volatile unsigned int timer_2;
extern volatile unsigned int timer_3; extern volatile unsigned int timer_4;
extern volatile unsigned int timer_5; extern volatile unsigned int timer_6;
extern volatile unsigned int timer_7; extern volatile unsigned int timer_8;
extern volatile unsigned int timer_9; extern volatile unsigned int timer_10;
```

**Parametri:**

nema

**Povratna vrijednost:**

nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
}
```

```

InitTimer();                                /* inicijaliziraj brojač (1mS)          */
while(1){                                    /* beskonačna petlja                    */
    if(timer_1 >= 1000){                     /* timer_1 je nabrojao 1000 mS? (1 sekunda) */
        Relay1Toggle();                     /* okreni stanje releja 1              */
        timer_1 = 0;                        /* postavi timer_1 na „0“              */
    }
}
}

```

Funkcija inicijalizira temperaturni senzor DS1820. Kontroler EMoRo podržava 7 temperaturnih senzora na portovima SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2 i DS1.

Upravljački program DC motora koristi zajedničke portove SERVO\_2, PWM\_1 i PWM\_2, te u slučaju inicijalizacije temperaturnog senzora na jednom od navedenih portova, DC motori se ne mogu koristiti.

```
int InitDs1820(unsigned char ch);
```

#### Parametri:

`unsigned char ch` – senzor DS1820 [SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2, DS1]

#### Povratna vrijednost:

`int` – Rezultat funkcije:  
 (0) – Uspješno inicijaliziran temperaturni senzor DS1820  
 (-1) – Greška: Senzor nedostupan na „ch“ portu

#### Podržani moduli:

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

#### Primjer:

```

#include <inovatic\emoro.h>                  /* dodjeljivanje emoro biblioteke      */

int main(void){                             /* glavni program                       */
    InitIO();                               /* inicijaliziraj ulaze i izlaze        */
    InitDs1820(IR1);                        /* inicijaliziraj DS1820 na portu IR1   */
    while(1){                               /* beskonačna petlja                    */
    }
}

```

Funkcija čita temperaturni senzor DS1820 sa razlučivosti od 0,5 °C. Prije čitanja temperature potrebno je inicijalizirati temperaturni senzor `InitDs1820()`; funkcijom.

```
int GetDs1820(unsigned char ch);
```

#### Parametri:

`unsigned char ch` – senzor DS1820 [SERVO\_1, SERVO\_2, PWM\_1, PWM\_2, IR1, IR2, DS1]

#### Povratna vrijednost:

`int` – Rezultat funkcije:  
 (-550 - +1250) – temperatura \* 10 (-55°C - +125°C)  
 (20000) – Greška: Senzor nedostupan na „ch“ portu

#### Podržani moduli:

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

#### Primjer:

```

#include <inovatic\emoro.h>                  /* dodjeljivanje emoro biblioteke      */

int main(void){                             /* glavni program                       */
    InitIO();                               /* inicijaliziraj ulaze i izlaze        */
    InitDs1820(IR1);                        /* inicijaliziraj DS1820 na portu IR1   */
}

```

```

while(1){
    if(GetDs1820(IR1) > 305)          /* beskonačna petlja          */
        Relay1On();                 /* temperatura je veća od 30,5 °C? */
    else                             /* uključi relej 1             */
        Relay1Off();                /* temperatura je manja ili jednaka 30,5 °C */
}
}

```

Funkcija inicijalizira analogno-digitalni pretvarač (ADC). Kontroler EMoRo podržava 3 analogno-digitalna pretvarača na ulazima IR1, IR2 i DS1, i prije čitanja ADC-a potrebno je izvršiti inicijalizaciju **InitAdc()**; funkcijom.

```
int InitAdc(unsigned char ch);
```

**Parametri:**

`unsigned char ch` – ADC [IR1, IR2, DS1]

**Povratna vrijednost:**

`int` – Rezultat funkcije:  
 (0) – Uspješno inicijaliziran analogno digitalni pretvarač  
 (-1) – Greška: Parametar „ch“ nije u intervalu [IR1, IR2, DS1]

**Podržani moduli:**

EMoRo RS232, EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                     /* glavni program                  */
    InitIO();                        /* inicijaliziraj ulaze i izlaze   */
    InitAdc(IR1);                    /* inicijaliziraj ADC na portu IR1 */
    InitAdc(IR2);                    /* inicijaliziraj ADC na portu IR2 */
    InitAdc(DS1);                    /* inicijaliziraj ADC na portu DS1 */
    while(1){                        /* beskonačna petlja              */
    }
}

```

Funkcija zaustavlja analogno-digitalni pretvarač (ADC).

```
int StopAdc(unsigned char ch);
```

**Parametri:**

`unsigned char ch` – ADC [IR1, IR2, DS1]

**Povratna vrijednost:**

`int` – Rezultat funkcije:  
 (0) – ADC port uspješno zaustavljen  
 (-1) – Greška: Parametar „ch“ nije u intervalu [IR1, IR2, DS1]

**Podržani moduli:**

EMoRo RS232, EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                     /* glavni program                  */
    InitIO();                        /* inicijaliziraj ulaze i izlaze   */
    InitAdc(DS1);                    /* inicijaliziraj ADC na portu DS1 */

    while(1){                        /* beskonačna petlja              */
        if(Input1() == 1)            /* pritisnuta tipka 1?            */

```

```

        StopAdc(DS1);                /* zaustavi ADC na portu DS1          */
    }
}

```

Funkcija čita stanje analogno-digitalnog pretvarača (ADC pokrenut, ADC zaustavljen).

```
int GetAdcState(unsigned char ch);
```

**Parametri:**

unsigned char ch – ADC [IR1, IR2, DS1]

**Povratna vrijednost:**

int – Rezultat funkcije:  
 (1) – ADC ulaz je inicijaliziran (ADC pokrenut)  
 (0) – ADC ulaz nije inicijaliziran (ADC zaustavljen)  
 (-1) – Greška: Parametar „ch“ nije u intervalu [IR1, IR2, DS1]

**Podržani moduli:**

EMoRo RS232, EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>    /* dodjeljivanje emoro biblioteke */

int main(void){               /* glavni program */
    InitIO();                  /* inicijaliziraj ulaze i izlaze */

    while(1){                  /* beskonačna petlja */

        /* pritisnuta tipka ulaza 1 i ADC ulaz IR1 je inicijaliziran? */
        if((Input1()==1) && (GetAdcState(IR1) == 1))
            StopAdc(IR1);      /* zaustavi ADC na ulazu IR1 */

        /* puštena tipka ulaza 1 i ADC ulaz IR1 nije inicijaliziran? */
        if((Input1()==0) && (GetAdcState(IR1) == 0))
            InitAdc(IR1);      /* inicijaliziraj ADC na ulazu IR1 */
    }
}

```

Funkcija čita analogno-digitalni pretvarač. Prije čitanja ADC-a potrebno je izvršiti inicijalizaciju pretvarača **InitAdc()**; funkcijom. Napon na ulazu pretvarača računa se formulom  $U = (5/1024) \times GetAdc(ch)$

```
int GetAdc(unsigned char ch);
```

**Parametri:**

unsigned char ch – ADC [IR1, IR2, DS1]

**Povratna vrijednost:**

int – Rezultat funkcije:  
 (0-1023) – Vrijednost analogno-digitalnog pretvarača [0-1023]  
 (-1) – Greška: Parametar „ch“ nije u intervalu [IR1, IR2, DS1]  
 (-2) – Greška: ADC nije inicijaliziran

**Podržani moduli:**

EMoRo RS232, EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>    /* dodjeljivanje emoro biblioteke */

int main(void){               /* glavni program */
    InitIO();                  /* inicijaliziraj ulaze i izlaze */
}

```

```

InitAdc(DS1);                                /* inicijaliziraj ADC na portu DS1 */
while(1){                                    /* beskonačna petlja */
    if(GetAdc(DS1) > 256)                    /* napon na ulazu DS1 je veći od 1.25 V? (5/1024)*256 */
        Relay1On();                        /* uključi relej 1 */
    else                                    /* napon je manji ili jednak 1.25V? */
        Relay1Off();                       /* isključi relej 1 */
}
}

```

Funkcija inicijalizira serijsku komunikaciju (8 data bit-ova, 1 stop bit), te podržava brzinu komunikacije u intervalu [1200-57600] bps. Za uspostavljanje komunikacije mikrokontrolerskog modula EMoRo RS232 potrebno je inicijalizirati serijsku komunikaciju **InitUart()**; funkcijom, a za uspostavljanje komunikacije EMoRo Bluetooth modula potrebno je inicijalizirati Bluetooth komunikaciju **InitBluetooth()**; funkcijom.

```
int InitUart(unsigned long baudrate);
```

**Parametri:**

**unsigned long** baudrate – brzina komunikacije [1200-57600] bps

**Povratna vrijednost:**

**int** – Rezultat funkcije:  
 (0) – Serijska komunikacija uspješno inicijalizirana  
 (-1) – Greška: Parametar „baudrate“ nije u intervalu [1200-57600]

**Podržani moduli:**

EMoRo RS232, EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>                /* dodjeljivanje emoro biblioteke */
int main(void){                            /* glavni program */
    InitIO();                             /* inicijaliziraj ulaze i izlaze */
    InitUart(38400);                      /* inicijaliziraj serijsku komunikaciju. (brzina 38400 bps) */
    while(1){                             /* beskonačna petlja */
        printf("Hello World\r\n");        /* pošalji pozdravnu poruku putem RS232 veze */
    }
}

```

Funkcija šalje 1 byte putem RS232 ili Bluetooth komunikacije. Za slanje jednog byte-a sa EMoRo RS232 mikrokontrolerskog modula potrebno je inicijalizirati komunikaciju **InitUart()**; funkcijom, a za slanje byte-a putem bluetooth veze potrebno je inicijalizirati komunikaciju **InitBluetooth()**; funkcijom.

```
int UartPutChar(unsigned char c);
```

**Parametri:**

**unsigned char** c – podatak za slanje [0-255]

**Povratna vrijednost:**

**int** – Rezultat funkcije:  
 (0) – Podatak uspješno pohranjen u međuspremnik  
 (-1) – Greška: Komunikacija nije inicijalizirana

**Podržani moduli:**

EMoRo RS232, EMoRo Bluetooth

**Primjer za EMoRo RS232:**

```

#include <inovatic\emoro.h>                /* dodjeljivanje emoro biblioteke */
int main(void){                            /* glavni program */

```

```

InitIO(); /* inicijaliziraj ulaze i izlaze */
InitUart(38400); /* inicijaliziraj serijsku komunikaciju. (brzina 38400 bps) */
while(1){ /* beskonačna petlja */
    UartPutChar(170); /* pošalji broj 170 serijskom komunikacijom */
}
}

```

#### Primjer za EMOro Bluetooth:

```

#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    InitBluetooth(); /* inicijaliziraj bluetooth komunikaciju. */
    while(1){ /* beskonačna petlja */
        UartPutChar(170); /* pošalji broj 170 serijskom komunikacijom */
    }
}

```

Funkcija čita 1 byte iz međuspremnika bluetooth ili RS232 komunikacije. Za čitanje jednog byte-a iz EMOro RS232 međuspremnika potrebno je inicijalizirati komunikaciju **InitUart()**; funkcijom, a za čitanje jednog byte-a iz bluetooth međuspremnika potrebno je inicijalizirati komunikaciju **InitBluetooth()**; funkcijom.

**UartGetChar()**; funkcija kao povratnu vrijednost uvijek vraća 1 byte, te je prije čitanja podatka iz međuspremnika potrebno provjeriti postoji li primljeni podatak u međuspremniku funkcijom **UartTestHit()**;

```
unsigned char UartGetChar(void);
```

#### Parametri:

Nema

#### Povratna vrijednost:

**unsigned char** - primljeni podatak [0-255]

#### Podržani moduli:

EMoRo RS232, EMOro Bluetooth

#### Primjer za EMOro RS232:

```

#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

unsigned char my_data;

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    InitUart(38400); /* inicijaliziraj serijsku komunikaciju. (brzina 38400 bps) */
    while(1){ /* beskonačna petlja */
        if(UartTestHit() > 0){ /* provjeri ima li primljenih podataka u međuspremniku */
            my_data = UartGetChar(); /* pročitaj 1 primljeni byte i pohrani ga u my_data varijablu */
        }
    }
}

```

#### Primjer za EMOro Bluetooth:

```

#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

unsigned char my_data;

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    InitBluetooth(); /* inicijaliziraj bluetooth komunikaciju. */
    while(1){ /* beskonačna petlja */
        if(UartTestHit() > 0){ /* provjeri ima li primljenih podataka u međuspremniku */

```



```

        my_data = UartGetChar(); /* pročitaj 1 primljeni byte i pohrani ga u my_data varijablu */
    }
}

```

Funkcija provjerava međuspremnik primljenih podataka.

```
unsigned char UartTestHit(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

unsigned char - (0) - međuspremnik primljenih podataka je prazan  
 (1) - međuspremnik primljenih podataka nije prazan

**Podržani moduli:**

EMoRo RS232, EMoRo Bluetooth

**Primjer za EMoRo RS232:**

```

#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    InitUart(38400); /* inicijaliziraj serijsku komunikaciju. (brzina 38400 bps) */
    while(1){ /* beskonačna petlja */
        if(UartTestHit() > 0){ /* provjeri ima li primljenih podataka u međuspremniku */

            /* pročitaj 1 primljeni byte i vrati ga nazad pošiljatelju */
            UartPutChar( UartGetChar() );
        }
    }
}

```

**Primjer za EMoRo Bluetooth:**

```

#include <inovatic\emoro.h> /* dodjeljivanje emoro biblioteke */

int main(void){ /* glavni program */
    InitIO(); /* inicijaliziraj ulaze i izlaze */
    InitBluetooth(); /* inicijaliziraj bluetooth komunikaciju. */
    while(1){ /* beskonačna petlja */
        if(UartTestHit() > 0){ /* provjeri ima li primljenih podataka u međuspremniku */

            /* pročitaj 1 primljeni byte i vrati ga nazad pošiljatelju */
            UartPutChar( UartGetChar() );
        }
    }
}

```

Funkcija briše sve dolazne podatke iz međuspremnika RS232 ili bluetooth komunikacije.

```
void UartFlushRx(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo Bluetooth

**Primjer za EMOro RS232:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitUart(38400);                 /* inicijaliziraj serijsku komunikaciju. (brzina 38400 bps) */
    while(1){                        /* beskonačna petlja */
        UartPutChar(170);             /* pošalji broj 170 serijskom komunikacijom */
        UartFlushRx();               /* obriši sve primljene podatke iz dolaznog međuspremnik */
    }
}
```

**Primjer za EMOro Bluetooth:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                 /* inicijaliziraj bluetooth komunikaciju. */
    while(1){                        /* beskonačna petlja */
        UartPutChar(170);             /* pošalji broj 170 serijskom komunikacijom */
        UartFlushRx();               /* obriši sve primljene podatke iz dolaznog međuspremnik */
    }
}
```

Funkcija briše sve odlazne podatke iz međuspremnik RS232 ili bluetooth komunikacije.

```
void UartFlushTx(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMOro Bluetooth

**Primjer za EMOro RS232:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitUart(38400);                 /* inicijaliziraj serijsku komunikaciju. (brzina 38400 bps) */
    while(1){                        /* beskonačna petlja */
        UartPutChar(170);             /* pošalji broj 170 serijskom komunikacijom */
        UartPutChar(170);             /* pošalji broj 170 serijskom komunikacijom */
        UartFlushTx();               /* obriši sve podatke iz odlaznog međuspremnik */
        _delay_ms(1000);              /* čekaj 1000 mS */
    }
}
```

**Primjer za EMOro Bluetooth:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                 /* inicijaliziraj bluetooth komunikaciju. */
    while(1){                        /* beskonačna petlja */
        UartPutChar(170);             /* pošalji broj 170 serijskom komunikacijom */
        UartPutChar(170);             /* pošalji broj 170 serijskom komunikacijom */
    }
}
```

```

    UartFlushTx();          /* obriši sve podatke iz odlaznog međuspremnika */
    _delay_ms(1000);        /* čekaj 1000 mS */
}

```

Funkcija inicijalizira bluetooth komunikaciju.

```
void InitBluetooth(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                     /* glavni program */
    InitIO();                       /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                /* inicijaliziraj bluetooth komunikaciju. */
    while(1){                       /* beskonačna petlja */
        printf("Hello World\r\n"); /* pošalji pozdravnu poruku putem bluetooth veze */
    }
}

```

Funkcija mijenja ime bluetooth modula. Za promjenu imena bluetooth modula potrebno je izvršiti inicijalizaciju **InitBluetooth()**; funkcijom.

```
int BluetoothChangeName(char* new_name);
```

**Parametri:**

char\* new\_name - Pokazivač na prvi član novog imena.

**Povratna vrijednost:**

int - (0) - Bluetooth ime uspješno promijenjeno  
 (-1) - Greška: Bluetooth modul ne podržava znakove novog imena

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```

#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                     /* glavni program */
    InitIO();                       /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                /* inicijaliziraj bluetooth komunikaciju. */
    BluetoothChangeName("My_New_Name"); /* promjeni ime bluetooth modula u „My_New_Name“ */
    while(1){                       /* beskonačna petlja */
    }
}

```

Funkcija mijenja lozinku bluetooth modula. Za promjenu lozinke bluetooth modula potrebno je izvršiti inicijalizaciju **InitBluetooth()**; funkcijom.

```
int BluetoothChangePasskey(char* new_passkey);
```

**Parametri:**

`char*` new\_passkey - Pokazivač na prvi član nove lozinke.

**Povratna vrijednost:**

`int` - (0) - Bluetooth lozinka uspješno promijenjena  
 (-1) - Greška: Bluetooth modul ne podržava znakove nove lozinke

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>           /* dodjeljivanje emoro biblioteke */

int main(void){                       /* glavni program */
    InitIO();                         /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                 /* inicijaliziraj bluetooth komunikaciju. */
    BluetoothChangePasskey("2323");  /* promjeni lozinku bluetooth modula u „2323“ */
    while(1){                        /* beskonačna petlja */
    }
}
```

Funkcija mijenja lozinku i ime bluetooth modula. Za promjenu lozinke i imena bluetooth modula potrebno je izvršiti inicijalizaciju **InitBluetooth()**; funkcijom.

```
int BluetoothChangeNameAndPasskey(char* new_name, char* new_passkey);
```

**Parametri:**

`char*` new\_name - Pokazivač na prvi član novog imena.  
`char*` new\_passkey - Pokazivač na prvi član nove lozinke.

**Povratna vrijednost:**

`int` - (0) - Bluetooth ime i lozinka uspješno promijenjeni.  
 (-1) - Greška: Bluetooth modul ne podržava znakove novog imena ili lozinke

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>           /* dodjeljivanje emoro biblioteke */

int main(void){                       /* glavni program */
    InitIO();                         /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                 /* inicijaliziraj bluetooth komunikaciju. */

    /* promjeni ime bluetooth modula u „My_New_Name“ i lozinku u „5555“ */
    BluetoothChangeNameAndPasskey("My_New_Name", "5555");
    while(1){                        /* beskonačna petlja */
    }
}
```

Funkcija čita ime bluetooth modula. Za čitanje imena bluetooth modula potrebno je izvršiti inicijalizaciju **InitBluetooth()**; funkcijom.

```
char* BluetoothGetName(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

`char*` - Pokazivač na ime bluetooth modula

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

char *module_name;

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                /* inicijaliziraj bluetooth komunikaciju. */
    module_name = BluetoothGetName(); /* čitaj pokazivač na ime bluetooth modula */
    while(1){                        /* beskonačna petlja */
    }
}
```

Funkcija čita lozinku bluetooth modula. Za čitanje lozinke bluetooth modula potrebno je izvršiti inicijalizaciju **InitBluetooth()**; funkcijom.

```
char* BluetoothGetPasskey(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

`char*` - Pokazivač na lozinku bluetooth modula

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

char *module_passkey;

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                /* inicijaliziraj bluetooth komunikaciju. */
    module_passkey = BluetoothGetPasskey(); /* čitaj pokazivač na lozinku bluetooth modula */
    while(1){                        /* beskonačna petlja */
    }
}
```

Funkcija vraća stanje bluetooth veze (Bluetooth spojen, Bluetooth nije spojen). Za čitanje bluetooth veze potrebno je izvršiti inicijalizaciju **InitBluetooth()**; funkcijom.

```
unsigned char BluetoothConnection(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

`unsigned char` - (1) - Bluetooth spojen  
(0) - Bluetooth odspojen

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                 /* inicijaliziraj bluetooth komunikaciju. */

    while(1){                        /* beskonačna petlja */

        while(BluetoothConnection() == 0); /* čekaj povezivanje bluetooth master-a */
        printf("Hello World\r\n");        /* pošalji pozdravnu poruku putem bluetooth veze */
        while(BluetoothConnection() == 1); /* čekaj odspajanje bluetooth master-a */
    }
}
```

Funkcija čita tipku bluetooth modula T1. Za čitanje stanja tipke potrebno je inicijalizirati bluetooth **InitBluetooth()**; funkcijom.

```
unsigned char BluetoothT1(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

**unsigned char** – Stanje tipke T1  
 (0) – puštena tipka  
 (1) – pritisnuta tipka

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                 /* inicijaliziraj bluetooth komunikaciju */

    while(BluetoothT1() == 0);        /* čekaj dok tipka T1 ne bude pritisnuta */
    Relay1On();                       /* uključi relej 1 */

    while(1){                        /* beskonačna petlja */
    }
}
```

Funkcije upravljaju STATUS LED-om bluetooth modula. Prije uključivanja, isključivanja ili promjene stanja STATUS LED-a potrebno je inicijalizirati bluetooth modul **InitBluetooth()**; funkcijom.

```
void BluetoothLedOn(void);
void BluetoothLedOff(void);
void BluetoothLedToggle(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitBluetooth();                 /* inicijaliziraj bluetooth komunikaciju */

    while(1){                        /* beskonačna petlja */
        BluetoothLedOn();            /* uključi bluetooth STATUS LED */
        _delay_ms(500);              /* čekaj 500 mS */

        BluetoothLedOff();           /* isključi bluetooth STATUS LED */
        _delay_ms(500);              /* čekaj 500 mS */

        BluetoothLedToggle();        /* promjeni stanje bluetooth STATUS LED-a */
        _delay_ms(500);              /* čekaj 500 mS */
    }
}
```

Funkcija inicijalizira buzzer na portu DS1. Na mikrokontrolersom modulu EMoRo USB prilikom korištenja buzzer-a ne mogu se koristiti DC motori jer upravljački program buzzer-a koristi zajednički hardverski brojač TIMER0.

```
void InitBuzzer(void);
```

**Parametri:**

Nema

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke */

int main(void){                      /* glavni program */
    InitIO();                        /* inicijaliziraj ulaze i izlaze */
    InitBuzzer();                   /* inicijaliziraj buzzer na portu DS1 */

    while(1){                        /* beskonačna petlja */
    }
}
```

Funkcija buzzer-om svira ton. Buzzer je potrebno spojiti na port DS1. Na mikrokontrolersom modulu EMoRo USB prilikom korištenja buzzer-a ne mogu se koristiti DC motori jer upravljački program buzzer-a koristi zajednički hardverski brojač TIMER0.

```
void PlayNote(unsigned char note, unsigned char duration);
```

**Parametri:**

**unsigned char note** - Nota [0-255] - MACRO naredbe tonova nalaze se u header datoteci. buzzer\_driver.h  
**unsigned char duration** - Trajanje note [0-255] (Vrijeme trajanja = duration\*25mS)

**Povratna vrijednost:**

Nema

**Podržani moduli:**

EMoRo RS232, EMoRo USB, EMoRo Bluetooth

**Primjer:**

```
#include <inovatic\emoro.h>          /* dodjeljivanje emoro biblioteke          */

int main(void){                      /* glavni program                      */
    InitIO();                        /* inicijaliziraj ulaze i izlaze      */
    InitBuzzer();                    /* inicijaliziraj buzzer na portu DS1 */

    PlayNote(C2, 3);                 /* Sviraj C2 u trajanju 3*25=75mS    */
    PlayNote(E2, 3);                 /* Sviraj E2 u trajanju 3*25=75mS    */
    PlayNote(G2, 3);                 /* Sviraj G2 u trajanju 3*25=75mS    */
    PlayNote(C3, 3);                 /* Sviraj C3 u trajanju 3*25=75mS    */
    while(1){                        /* beskonačna petlja                  */

    }
}
```