

# KONTROLLERITE PROGRAMMEERIMINE KASUTADES ARENDUSPLAATI DIGILENT BASYS MX3

## 1 Sissejuhatus programmeerimis keelde C

### Teek

```
1. #include <stdio.h> //printf(), Kommentaar
```

Saame lisada teiste loodud koode(funktsioone) läbi teekide(*libraries*).

### Prototüübid

```
1. //prototüübid
2. int arvutaSumma(int arvudeNimekiri[], int punktideArv);
3. int vagaSuurPunktiSumma(int arvudeNimekiri[], int punktideArv);
4. void prindiPunktid(int punktiSumma);
```

Enda koostatud koodi osade ehk funktsioonide kirjeldused.

### Andmetüübid

Liik	Tüüp
Märk(üksik täht/karakter)	char
Täisarv	int
Komaga arv	float/double
Tõeväärtus(true/false)	bool

## Funktsioonid

```
andmetüüp1 funk_nimetus(andmetüüp parameetri_nimetus, andmetüüp teine_param,...)
{
    //muutuja deklaratsioonid
    andmetüüp1 tulem;

    //tegevused

    return tulem; //funk-ni väärtuse tagastamine(PS v.a. void andmetüübi puhul)
}
```

### Tagastamisega

```
1. int arvutaSumma(int arvudeNimekiri[], int punktideArv)
2. {
3.     int i; //loendur
4.     int arvudKokku = 0; //summa
5.     for(i = 0; i < punktideArv; i++)
6.     {
7.         arvudKokku = arvudKokku + arvudeNimekiri[i];
8.     }
9.
10.     return arvudKokku;
11. }
```

### Tagastamiseta

```
1. void prindiPunktd(int punktiSumma)
2. {
3.     printf("Punktide summa on %d\n", punktiSumma);
4. }
```

## Muutujate deklareerimine ja kasutamine

andmetüüp nimetus;

```
1. //muutujate deklareerimine
2. char taht = 'A';
3. int number = 0, nimekiri[6] = {3, 36, 1, 1, 2, 6};
4. int punktideSumma;
5. //muutujatele väärtuste omistamine
6. number = 6;
7. nimekiri[0] = 5; // jada esimene arv indeksiga 0
8. nimekiri[3] = 2; //jada neljas arv indeksiga 3
```

	0.	1.	2.	3.	4.	5.
Arvude jada	3	36	1	1	2	6

## Operatsioonid

### Aritmeetilised

Operaator	Kirjeldus
+	Liitmine
-	Lahutamine
*	Korrutamine
/	Jagamine
++	Inkrementeerimine
--	Dekrementeerimine

### Võrdlused

Operaator	Kirjeldus
==	Võrdlus
!=	Ei ole võrdsed
>	Suurem kui
<	Väiksem Kui
>=	Suurem või võrdne
<=	Väiksem või võrdne

## Tsükliid

**for** (algavaldis; tingimus; lõpptegevus)

```
{  
    Tegevus;  
}
```

```
1. int i; //loendur  
2. int arvudKokku = 0; //summa  
3. for(i = 0; i < punktideArv; i++)  
4. {  
5.     arvudKokku = arvudKokku + arvudeNimekiri[i];  
6. }
```

**while** (tingimus)

```
{  
    tegevus;//tehakse, kuniks tingimus osutub valeks  
}
```

```
1. while(kontrollime)  
2. {  
3.     if(arvudeNimekiri[i] > 30) //Kui tulemus on suurem kui 30  
4.     {  
5.         tulem = 1;  
6.         break; // väljutakse tsüklist  
7.     }  
8.     }else  
9.     {  
10.        i++; // i = i + 1;  
11.    }  
12.  
13.    if( i == punktideArv)  
14.    {  
15.        break;//väljutakse tsüklist  
16.    }  
17. }
```

## Tingimuslaused

```
if (tingimus) // kui tingimus on õige
{
    tegevus;
} else
{
    teine tegevus;
}
```

```
1. if (jada[i] < 0)
2. {
3.     tulem = 1;
4.     break; // väljutakse tsüklist
5.
6. } else
7. {
8.     i++; // i = i + 1;
9. }
```

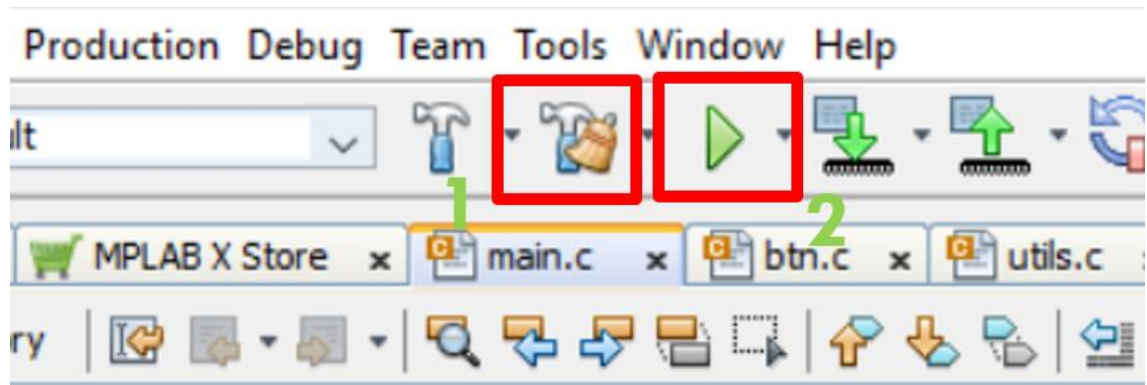
```
switch (avaldis)
{
    case väärtus1:
        tegevus1;
        break;
    case väärtus2:
        tegevus2;
        break;
    default:
        muu tegevus;
}
```

```
1. switch (vagaSuurPunktiSumma(nimekiri, number))
2. {
3.     case 0:
4.         printf("Punktisummad olid väiksed.\n");
5.         break;
6.     case 1:
7.         printf("Leidus suur punktisumma.\n");
8.         break;
9.     default:
10.        printf("Miski läks valesti!\n");
11.        break;
12. }
```

Rohkem infot programmeerimis keele C kohta lingilt:  
<https://www.tutorialspoint.com/cprogramming/>

## 2 MPLAB kasutamine

1. Programmi koodi kompileerimine ehk „Clean and Build“
2. Jooksuta programm plaadile ehk Run



Prototypes

```
id init(void);  
id sesamOpen();  
id toggleRGB();  
- switchSelected();
```

Rohkem infot arendusplaadi manuaalist:

<https://reference.digilentinc.com/reference/microprocessor/basys-mx3/reference-manual>

### 3 Ülesanne

#### 1. Põhiülesanne – Seifi avamine PIN koodiga

Eesolevasse programmikoodi **main()** funktsiooni tuleb lisada vastavalt teisi selle funktsiooni allpool kirjeldatud funktsioone (nt `toggleRGB()`), nii et arendusplaadil saaks lülititega (*switches*) sisestada PIN koodi, mis kujutatakse seitsme segmenti ekraanil, ja see seejärel kontrollida PIN koodi õigsust.

Ülesande lahendus videol: <https://youtu.be/DQvokYZsz-s>

#### 2. Lisaülesanne I

Lisada funktsioon **LEDToggleBySwitch()** vastavasse funktsiooni selliselt, et iga kord, kui lüliti lükatakse sisse, läheb arendusplaadil sama numbriga LED tuluks põlema.

Ülesande lahendus videol: <https://youtu.be/xeWCDBpGnmc>

#### 3. Lisaülesanne II

Lisada funktsioon **BTNPressed()** vastavasse funktsiooni selliselt, et iga kord, kui PIN kood on segmentide ekraanile kuvatud, siis alles arendusplaadil oleva nupu vajutamisel kontrollitakse, kas PIN kood on õige.

Ülesande lahendus videol: <https://youtu.be/ymHN19WM5hA>

Ülesande programmikood leitav lingilt:

<https://raw.githubusercontent.com/Lapikud/basys-mx3-koolitus/master/main.c>

## Funktsioonide kirjeldused

```
1. int main(void)
2. {
3.     int pin[4] = {0, 1, 2, 3};
4.
5.     init();
6.
7.     //oma koodi jupp lisada siia
8.
9.     return 0;
10. }
```

```
1. /* -----
   */
2. /***   init
3. **      Parametrid:
4. **      Tagastatav väärtus:
5. **      Kirjeldus:
6. **              Initsialiseeritakse LED, LCD, SWT, BTN, RGBLE
7. **              and SSD(Seitsme segmendi ekraan) mooduleid.
8. **              Vajalik riistvara ettevalmistamiseks.
9. **
10. */
11. void init(void)
```

```
1. /* -----
   */
2. /***   sesameOpen
3. **      Parametrid:
4. **      Tagastatav väärtus:
5. **      Kirjeldus:
6. **              LCD ekraanile kuvatakse kinnitus, et PIN kood
7. **              on korras
8. **
9. */
9. void sesameOpen()
```

```
1. /* -----
   */
2. /***   toggleRGB
3. **      Parametrid:
4. **      Tagastatav väärtus:
5. **      Kirjeldus:
6. **              RGB LED muudab värve;
7. **              NB! See funktsioon ei lõpeta oma tööd!
8. **
9. */
10. void toggleRGB()
```



```

1.  /* -----
   */
2.  /**      switchSelected
3.  **      Parameetrid:
4.  **      Tagastatav väärtus:
5.  **          int i - Lüliti number, mis on sisse lülitatud
6.  **      Kirjeldus:
7.  **          Tagastab väärtuse siis, kui lüliti on sisse
8.  **          lülitatud. Kui sisse lülitatakse mitu lüliti,
9.  **          tagastatakse esimesena registreeritu lüliti
10. **          väärtuse
11. **
12. */
13. int switchSelected()

```

```

1.  /* -----
   */
2.  /**      checkPin
3.  **      Parameetrid:
4.  **          int sentPin    - PIN kood, mida kasutaja s
5.  **                      sisestab
6.  **          int pin        - Õige PIN kood
7.  **      Tagastatav väärtus:
8.  **          int            - Õige(true) või väär(false)
9.  **                      0- kui sentPin on erinev pin-ist
10. **                      1- kui sentPin on sama kui pin
11. **      Kirjeldus:
12. **          Vörreldakse omavahel kahte neljanumbrilist arvu
13. **          Tagastatakse 1 või 0 olenevalt tulemusest
14. **
15. */
16. int checkPin(int sentPin[], int pin[])

```

```

1.  /* -----
   */
2.  /**      displaySegment
3.  **      Parameetrid:
4.  **          int enteredPin[]- kaustaja sisestatud PIN kood
5.  **      Tagastatav väärtus:
6.  **      Kirjeldus:
7.  **          Sisestatud PIN kood kuvatakse
8.  **          SSD-le ehk seitsme segmendilisele ekraanile
9.  **          Its mapped from right to left
10. **
11. */
12. void displaySegment(int enteredPin[])

```

```

1.  /* -----
    */
2.  /**      swtChanged
3.  **
4.  **      Parameetrid:
5.  **
6.  **      Tagastatv väärtus:
7.  **          int      - Boolean (true (1) või false(0))
8.  **                      0 kui ühtegi lülitit pole sisse
9.  **                      lülitatud
10. **                      1 kui vähemalt 1 lülititest on sisse
11. **                      lülitatud
12. **      Kirjeldus:
13. **          Näitab, kas ühtegi lülitit on sisse lülitatud
14. **
15. */
16. int swtChanged()

```

```

1.  /* -----
    */
2.  /**      checkSafe
3.  **
4.  **      Parameetrid:
5.  **          int pin[] - Kasutaja poolt sisestatud PIN kood
6.  **                      massiivina
7.  **
8.  **      Tagastatav väärtus:
9.  **          int result - näitab, kas seif on lahti tehtud
10. **                      ehk kas sisestatud PIN kood on õige
11. **                      Kui sisestatud PIN kood osutub
12. **                      õigeks, siis tagastatakse väärtus 1
13. **
14. **      Kirjeldus:
15. **          Laseb kasutajal sisestada PIN koodi lülitite
16. **          kaudu. Juhised sisestamiseks kuvatakse LCD
17. **          ekraanle.
18. **          Funktsioon tagastab väärtuse 1 siis, kui
19. **          sisestatud PIN kood on õige.
20. **
21. */
22. int checkSafe(int pin[])

```

```

1.  /* -----
   */
2.  /**      LEDTogglBySwitch
3.  **
4.  **      Parameetrid:
5.  **      Tagastatav väärtus:
6.  **      Kirjeldus:
7.  **          Lüliti sisse lülitamisel, pannakse põlema
8.  **          lüliti numbrile vastav LED
9.  **          Näiteks: kui lüliti SWT0 on sisse lülitatud,
10. **          siis läheb LED pirn LD0 põlema
11. **
12. */
13. void LEDTogglBySwitch()

```

```

1.  /* -----
   */
2.  /**      BTNPressed
3.  **
4.  **      Parameetrid:
5.  **          unsigned char btnLetter - Nupu tähistus plaadil
6.  **                                     Nt. 'R' ; 'L' ; 'U'; 'D'; 'C'
7.  **
8.  **      Tagastatav väärtus:
9.  **          1 - määratud nupp on alla vajutatud
10. **
11. **      Kirjeldus:
12. **          Kui määratud nupp on alla vajutatud, tagastab
13. **          funktsioon väärtuse 1.
14. **          Vastasel juhul programm ei lõpeta tööd
15. **
16. */
17. int BTNPressed(unsigned char btnLetter)

```