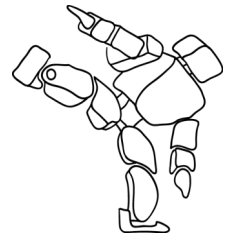


Programmeerimine C keeles AVR mikrokontrolleri baasil

Peeter Salong

Sisujuht



Mõisted

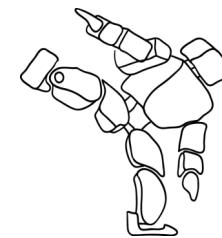
Programmi elutsükkel

C keel

- Näiteprogramm
- Süntaksireeglid
- Nimed
- Andmetüübid
- Muutujate deklareerimine ja väärtuse omistamine
- Avaldised, operaatorid
- Skoop
- Laused
- Funktsioonid
- Failid
- Koodi kirjutamise soovitused

Praktiline programmeerimine

Mõisted

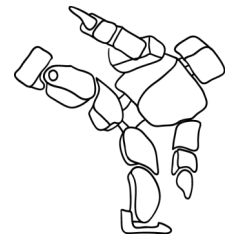


Arvutiprogramm - kindlate reeglite järgi kirja pandud instruksioonide kogum, mis on mõeldud arvuti protsessorile täitmiseks.

Programmeerimiskeel - tehiskeel arvutile instruksioonide kirja panemiseks. Keelel on kindel süntaks (lauseehitus) ja semantika (tähendus). Keele abil kirja pandud lauseid kutsutakse lähtekoodiks.

Kompilaator - programm, mis muudab inimesele arusaadava lähtekoodi arvutile arusaadavaks masinkoodiks. Meie kasutame **AVR-GCC** kompilaatorit.

Programmi elutsükkel



Lähtekoodi kirjutamine

PC



AVR GCC kompilaator



Hex-fail

- C keel
- IDE (AVR Studio, KontrollerLab) või tekstiredaktor

- objektifailid
- linkimine
- masinkood

```
00000000: 12 00 00 EA 14 F0 9F E5 14 F0 9F E5 14 F0 9F E5
00000010: 14 F0 9F E5 14 F0 9F E5 14 F0 9F E5 14 F0 9F E5
00000020: 38 00 00 25 00 00 00 00 00 00 00 00 00 00 00
00000030: 40 00 00 00 4C 00 00 00 4E 4E 4E 4E 4E 4E 4E
00000040: FE FF FF EA FE FF FF EA FE FF FF EA FE FF FF EA
00000050: C0 9F E5 20 F0 21 E3 00 00 A0 E1 04 00 40 E2
00000060: 07 F0 21 E3 00 00 A0 E1 04 00 40 E2 01 F0 21 E3
00000070: 00 00 A0 E1 04 00 40 E2 02 F0 21 E3 00 00 A0 E1
00000080: 00 00 A0 E1 04 00 40 E2 03 F0 21 E3 00 00 A0 E1
00000090: DF F0 21 E3 00 00 A0 E1 04 00 40 E2 04 00 40 E2
000000A0: 7C 3F 4E 05 00 00 00 00 00 00 00 00 00 00 00
000000B0: F0 FF 3A 00 00 A0 E1 04 00 40 E2 05 00 40 E2
000000C0: 02 00 01 E3 04 00 01 04 00 00 FF 3A 00 00 00
000000D0: 0C 10 9F E5 01 00 00 E1 00 00 00 0A 00 00 EA
000000E0: 03 00 00 E5 0F 82 A0 E1 00 00 00 A0 E1 00 00 EA
000000F0: FF FF FF EA 00 00 A0 E1 00 00 00 A0 E1 00 00 EA
00000100: 00 00 A0 E1 00 00 70 A0 E1 00 A0 9F E5 0F EA A0 E1
00000110: 1A FF FF E1 FF FF EA 00 00 40 4C 4E 4E 4E 4E
00000120: FF FF FF FF 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
00000130: 4C 0E 00 00 4C 1E 00 00 00 00 00 00 04 E0 4C E2
00000140: FF FF FF FF 00 10 4F E1 00 00 20 E1 4A 31 9F E5
00000150: 80 00 01 40 31 9F E5 80 00 03 E1 3C 31 9F E5
00000160: 00 A0 93 E5 00 F1 9F 07 00 00 EA 0C 00 00 00
00000170: 80 03 01 31 00 00 EA 00 00 02 E2 02 00 40 E2
00000180: 4C 02 00 00 C4 01 00 00 4C 02 00 00 00 00 00
00000190: 4C 02 00 00 4C 02 00 00 4C 02 00 00 4C 02 00
000001A0: 4C 02 00 00 C4 01 00 00 4C 02 00 00 4C 02 00
000001B0: 24 00 00 EA E4 10 9F E5 00 10 41 B2 01 10 02 E1
000001C0: 00 00 01 E3 00 9F E5 01 10 41 B2 01 1C E0 81
000001D0: 00 00 0E E4 C0 30 9F E5 01 1C E0 01 03 00 E7
000001E0: 01 10 01 B2 01 38 A0 E1 23 38 A0 E1 14 20 00 E5
```

•kogu kood Hex arvudena

Mikrokontroller

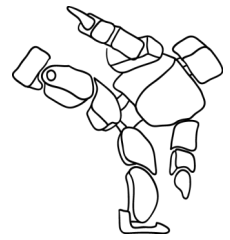


Programmaator



•Programmeerib mikrokontrollerit Hex-failiga

C keele näiteprogramm



```
#include <avr/io.h>
```

```
#include <stdio.h>
```

```
int main(void)
```

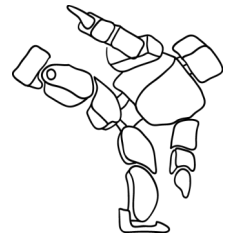
```
{
```

```
    printf("Hello world!");
```

```
    return 0;
```

```
}
```

C keel



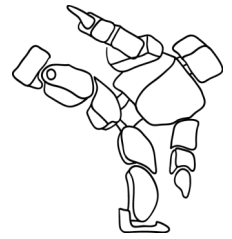
Programm koosneb:

- Funktsioonidest (function) - tehted
- Muutujatest (variable) - andmed
- Makrodest (macro) - programmi lähtekood esitatud ühe avaldisena

Üldist

- Olulisim funktsioon: `main`
- Funktsioone ja muutujaid pannakse kirja lausetena (tinglikult võib öelda, et funktsioon on liitlause, mis koosneb lihtlausest).
- Programmi töö toimub järjestikuliste instruksioonide täitmise teel

C keel - süntaks



Funktsiooni sisu kirjutatakse looksulgude vahele { }

Iga lihtlause lõpeb semikooloniga ‘;’

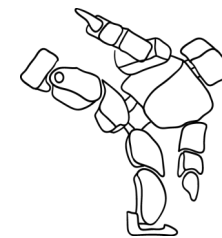
Valged sümbolid (tühimik):

- Tühik
- Tabulaator
- Reavahetus

Eralduse tüübid:

- Kohustuslik eraldus – vähemalt 1 tühimik
- Vabatahtlik eraldus – tühimik võib puududa

C keel - Nimed



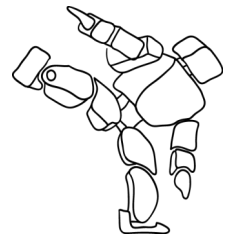
Suuri ja väikeseid tähti eristatakse

Nimi võib sisaldada ainult suuri ja väikeseid ladinatähti, numbreid ja alakriipsusid.

Nimi ei või alata numbri sümboliga

Näited: name, S66m , 6S6m, _, jUhAn

C keel - Andmetüübid



Baas-andmetüübid:

```
int, char, short, long, float
```

Massiivid:

```
char[], int[][]
```

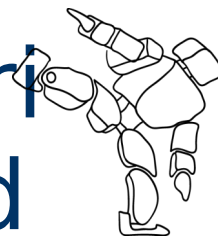
Struktuurid:

```
struct { char, int }
```

Ühendtüüp (jagatud mälu):

```
union { char[2], short }
```

C keel – AVR-GCC kompilaatori baas-andmetüüpide suurused

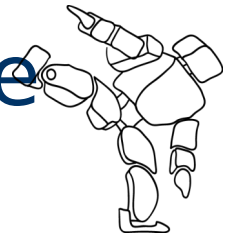


NB! Samad andmetüübid ei pruugi omada samu arvvaahemikke teiste kompilaatoritega

Nimetus	Bitte	Minimaalne	Maksimaalne
(signed) char	8	-128	127
unsigned char	8	0	255
(signed) int	16	-32768	32767
unsigned int	16	0	65535
(signed) long	32	-2 147 483 648	2 147 483 647
unsigned long	32	0	4 294 967 296

* AVR mikrokontrolleril: short = int

C keel – Muutujate deklareerimine ja väärtuse omistamine



```
unsigned int intValue;
```

```
intValue = 145;
```

```
char charValue = 150;
```

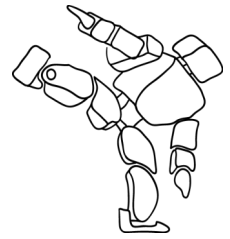
Andmetüübi nimi

Muutuja nimi

Väärtuse omistamine

```
charValue > intValue ?
```

C keel - Massiivid ja struktuurid



```
unsigned char array[3] // Ühemõõtmeline massiiv  
unsigned int table[10][20]; // Kahemõõtmeline massiiv
```

```
// Struktuurne muutuja
```

```
struct {  
    int speed;  
    unsigned int horsepower;  
} car;
```

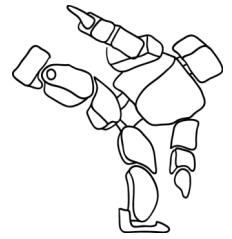
```
// Struktuurne andmetüüp
```

```
typedef struct {  
    int x, y, r;  
    char colour[20];  
} circle;
```

```
circle circlesOnPicture[20]; // Massiiv struktuurist
```

Paar näidet õpilastelt

C keel – Avaldised



// Avaldis

[operand1] [operaator] [operand2]

Operand on konstant, muutuja või funktsioon.

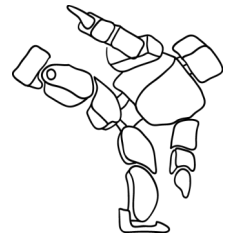
Operand võib ka ise omakorda avaldis olla.

Avaldiste väärtusi saab omistada muutujatele, anda funktsiooni argumentideks või tagastada funktsioonis.

```
int y = 17 - 5 * 2;
```

```
int x = doubleTheValue(10 / 3);
```

Operaatorid (tehted)



Aritmeetika:

+ - / * %

Loogikaavaldised:

! && ||

Bitiloogika:

~ & | ^ << >>

Võrdlus:

== != < > <= >=

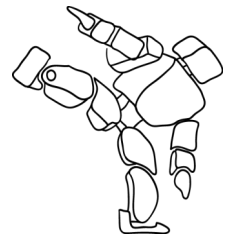
Omistus:

=

Arvutus-omistus:

+= -= *= /= |=

C keel - Skoop



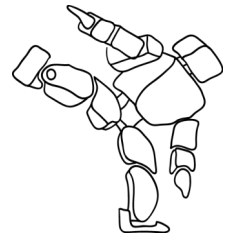
Avalik skoop on üle kõigi lähtefailide
ühes projektis

Failisisene skoop on ühe faili piires

Kohalik skoop on loogsulgude vahel



C keel – skoop näide

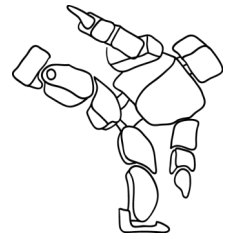


```
int i = 0;
```

```
int function()  
{  
    int i = 3;  
    return i;  
}
```

```
int main(void)  
{  
    i = function();  
    return 0;  
}
```


C keel - laused



Tingimuslause

if *[(avaldis)]* { *[lause]* }

if, *if-else*, *if-elseif-else*

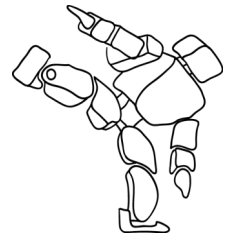
Mis on tõene avaldis? Mis on väär?

Valikulause

switch *[(avaldis)]* { *case* *[väärtus]*: *break*; }

switch

C keel - laused



Tsüklilised laused

for ([alguslause]; [avaldis]; [lõpulause]) {[lause]}

while ([avaldis]) {[lause]}

do{[lause]} *while*([avaldis])

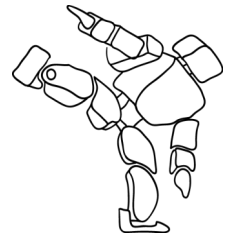
Tsükli suunajad

break; - lõpetab kogu tsükli

continue; - lõpetab hetke iteratsiooni



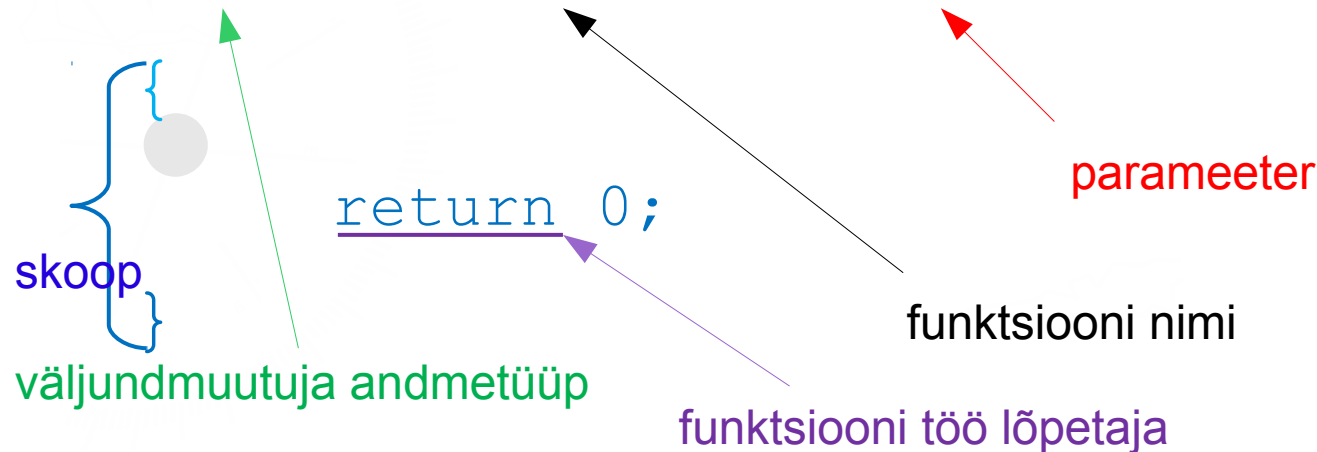
C keel -funktsioonid



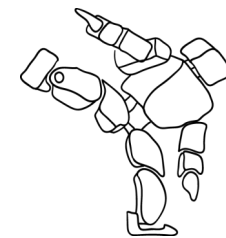
Funktsioon – instruksioonide kogum, mis täidab kindlat programmi osa ning mis omab omaette skoopi.

Funktsiooni definitsioon:

```
int power (int base, int i)
```



C keel - funktsioonid



Funktsiooni deklaratsioon, definitsioon ja väljakutsumine

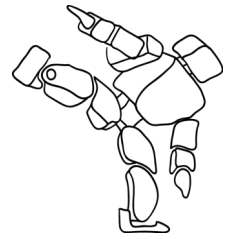
```
#include <avr/io.h>

/* funktsiooni "power" deklaratsioon*/
int power(int base, int n);

int main(void)
{
    int val;
    // funktsiooni "power" väljakutsumine
    val = power(2, 5);
}

// funktsiooni "power" definitsioon
int power(int base, int n)
{
    int p;
    for (p = 1; n > 0; --n)
    {
        p = p * base;
    }
    return p;
}
```

C keel - sprintf



sprintf – funktsioon, mis salvestab andmed tekstiks etteantud formaadi järgi

```
int sprintf (char * str, const char * format, ...);
```

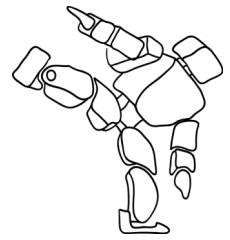
Vajab #include <stdio.h> faili algusse

[Detailne info](#)

Kirjutab numbri 6 tekstis kümnendsüsteemi numbrina

```
char text[5];  
sprintf(text, "%d", 6);
```

Failid



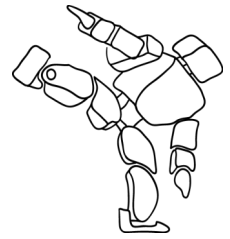
*.c fail – lähtekoodi fail, mis sisaldab funktsioonide definitsioone (sisu).

*.h fail – päise fail, mis sisaldab globaalsete muutujate ja funktsioonide deklaratsioone

Failidevaheline skoop

extern

C keel - soovitus



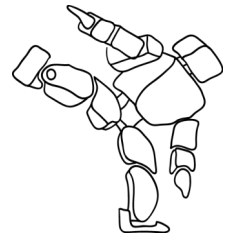
Kumb on arusaadavam?

```
int power(int base, int n)
{
    int p;
    for (p = 1; n > 0; --n)
    {
        p = p * base;
    }
    return p;
}
```

```
int power(int base, int n)
{
    int p;
    for (p = 1; n > 0; --n)
    {
        p = p * base;
    }
    return p;
}
```

Kood kirjutada treppi!

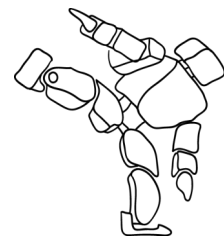
C keel - soovitused



```
int power(int base, int n)
{
    int p;
    for (p = 1; n > 0; --n)
    {
        p=p*base;
    }
    return p;
}
```

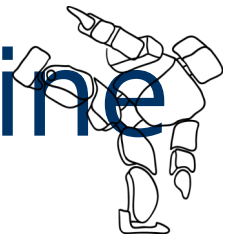
Hoida ühtset stiili!

LCD ekraan



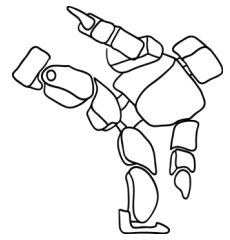
- Kirjutame ekraanile praeguse aastaarvu

Praktiline programmeerimine



- Kirjuta oma eesnimi LCD ekraani teisele reale
- Kirjutada programm, mis kutsub välja funktsiooni ühe valitud aritmeetilise tehte sooritamiseks. Funktsiooni parameetrid on 2 täisarvu. Funktsioon väljastab tehte tulemuse. Näita tulemust LCD ekraanil.
- Kirjutada funktsioon, mis võimaldab teostada kahe täisarvuga kõiki viite tehet (+, -, *, /, %). Funktsiooni parameetrid on 2 arvu ning valitav tehe. Väljundiks on kahe arvu tulemus. Näita kõiki tulemusi korraga LCD ekraanil.

Iseisev harjutamine



Kirjuta funktsioon, mis konverteerib kümne süsteemi arvu kahend süsteemi arvuk.



Tänan!

