

# C keele kiirkursus

Tutvustus

Peeter Salong



TTÜ Robotiklubi

# Sisujuht

- Mõisted
- Taustaprotsess
- Projekti loomine
- C keel
  - Näiteprogramm
  - Süntaksireeglid
  - Nimed ja andmetüübid
  - Muutujate deklareerimine ja väärtuse omistamine
  - Avaldised, operaatorid
  - Funktsioonid
  - Skoop
  - Laused
  - Failid
  - Koodi kirjutamise soovitused



# 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.



# Taustaprotsess

Käskood	Operant/Operandid
<b>Masinkood</b> <i>Binary machine language</i>	0011011100111001 1100011001010011 1111010000010101 1111000001110010
<b>Assembler keel</b> <i>Assembly language</i>	ADD a,b,c MOV d, M XOR c,d
<b>Kõrgtaseme keel</b> <i>High-level language</i>	If n<100 then a:= b else v[i]:=k[j-1] end;



# Projekti loomine

- Eclipse seadistamine ja uue projekti loomine  
<http://www.scienceprog.com/setup-avr-eclipse-plugin-to-work-with-winavr/>
- **File->New->C Project**. Võta **Empty Project** valikus **AVR Cross Target Applications** ja sisesta projekti nimi. Muuda vajadusel projekti asukohta.
- **Select Configuration** aknas vali lihtsalt **Next**
- **HW target properties** aknas
  - **MCU type:** ATmega328P
  - **MCU frequency:** 20 000000 (20 Mhz)
- **File->New->Source File** ja kirjuta faili nimi (main.c)



# C keele näiteprogramm

```
#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).



# 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 arvvahemikke 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



TTÜ Robotiklubi

# 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
```



# 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 -funktsioonid

- Funktsioon – instruksioonide kogum, mis täidab kindlat programmi osa ning mis omab omaette skoopi.
- Funktsiooni definitsioon:

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

```
{
```

```
    return 0;
```

```
}
```

parameeter

funktsiooni nimi

funktsiooni töö lõpetaja

väljundmuutuja andmetüüp

skoop





# 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 - 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*

- 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



# 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 - include

- #include < > ja #include " " vahe?



# 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!**



# Täna!



TTÜ Robotiklubi